University of Southampton

Faculty of Engineering and Physical Sciences

School of Engineering

Improving the applicability of genetic algorithms to real problems

by

Przemyslaw Andrzej Grudniewski

ORCID ID 0000-0003-0635-3125

Thesis for the degree of Doctorate of Philosophy

January 2021

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
School of Engineering

Thesis for the degree of Doctorate of Philosophy

Improving the applicability of genetic algorithms to real problems

by Przemyslaw Andrzej Grudniewski

The current state-of-the-art of genetic algorithms is dominated by high-performing specialist-solvers with fast convergence. These algorithms require prior knowledge about the characteristics of the optimised problem to operate effectively, although such information is not available in most real cases. Most of these algorithms are only tested on a narrow range of similar benchmarking problems with lower complexity than the real cases. This leads to the promotion of high-performing strategies for these cases, but which might prove to be ineffective on practical applications. This hypothesis is supported by a low uptake of the current specialist/convergence algorithms on real-world cases; NSGA-II remains the most popular algorithm despite being developed in 2002. It is suggested that this is due to its uniformly good performance across a wide range of problems with distinct characteristics, indicating high generality, and a high diversity retention across iterations.

To assess if increasing the generality and diversity of the search improves the performance on real problems, the Multi-Level Selection Genetic Algorithm (MLSGA) is extended to develop a "diversity-first" general-solver genetic algorithm. It is selected as it shows high promise for the diversity-oriented methodology. Firstly, the reasons behind why it exhibits high diversity are investigated, as it is shown that the collective-level mechanisms create additional evolutionary pressure, while the fitness separation approach leads to collectives targeting different regions of the search space. This creates unique region-based search which leads to retention of higher diversity of solutions between generations. Secondly, as the MLSGA is exhibiting a poor convergence, the algorithm is combined with the current state-of-the-art algorithms in the hybrid approach (MLSGA-hybrid) to offset this problem. MLSGA-hybrid focuses on increasing the convergence of the search, over the original MLSGA algorithm, while retaining its emphasis on the diversity. The results demonstrate that this improvement leads to top performance on a range of problems. This is particularly the case on constrained problems indicating that the diversity has been retained. Thirdly, the co-evolutionary variant is introduced and tested (cMLSGA), which combines multiple evolutionary algorithms to improve the generality of the method. To validate the performance of the "diversity-first" general-solver approach, the algorithm is tested on 100 benchmarking problems and compared with top algorithms from the current state-of-the-art. It is shown that cMLSGA is the best general-solver, due to the most robust performance across the evaluated cases, while maintaining a higher focus on problems where elevated diversity of the search is preferred, such as discontinuous, constrained and biased cases. Finally, the cMLSGA approach is benchmarked on 3 engineering cases with a wide range of diverse characteristics and compared with other leading genetic algorithms. It is shown that the convergence-oriented solvers are ineffective for real-world applications due to higher complexity of practical problems, whereas performance of specialist-solvers is low due differences between real-world cases and benchmarking functions they are adjusted to. According to that, the "diversity-first" genetic algorithms with a high generality are preferred and there should be more focus on algorithms with these characteristics in the future.

Table of Contents

Table of Tables

Table of Figures

Research Thesis: Declaration of Authorship

Print name: PRZEMYSLAW ANDRZEJ GRUDNIEWSKI

Title of thesis: Improving the applicability of genetic algorithms to real problems

I declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:

P.A. Grudniewski* and A. J. Sobey, "Multi-Level Selection Genetic Algorithm applied to CEC'09 test instances", *2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastian*, pp. 1613-1620, 2017.

A. J. Sobey* and P.A. Grudniewski, "Re-inspiring the genetic algorithm with multi-level selection theory: multi-level selection genetic algorithm (MLSGA)", *Bioinspiration and Biomimetics*, vol. 13, no. 5, pp. 1-13, 2018.

P. A. Grudniewski* and A. J. Sobey, "Behaviour of Multi-Level Selection Genetic Algorithm (MLSGA) using different individual-level selection mechanisms", *Swarm and Evolutionary Computation*, vol. 44, no. September 2018, pp. 852–862, 2018.

P. A. Grudniewski* and A. J. Sobey "Do general Genetic Algorithms provide benefits when solving real problems?", *2019 IEEE Congress on Evolutionary Computation (CEC), Wellington*, 2019.

P. A. Grudniewski* and A. J. Sobey, "cMLSGA: co-evolutionary Multi-Level Selection Genetic Algorithm", *Natural Computing*, 2021, Under review.

Signature:... Date: 13.01.2021

Acknowledgements

I would like to thank my supervisors, Dr Adam Sobey and Prof. Ajit Shenoi for their guidance, support, and time during all these years.

I would like to thank my parents, Tomasz and Alina Grudniewski, and my siblings, Aleksandra and Patryk, for their constant support.

I would like to thank my partner, Meg O'Callaghan, for constant encouragement.

I would like to thank my friends for their support and encouragement. Thanks especially to my colleagues' Dr Jeanne Blanchard, Dr Apostolos Grammatikopoulos and Amy Parkes for their advice regarding English writing while producing this thesis.

Finally, I would like to thank Lloyd Register Foundation for sponsoring this project.

Abbreviations

BCE	- Bi-Criterion Evolution algorithm
cMLSGA	- co-evolutionary Multi-Level Selection Genetic Algorithm
DMOP	- Dynamic Multi-Objective Problem
GA	- Genetic Algorithm
HEIA	- Hybrid Evolutionary Immune Algorithm
HV	- Hyper Volume
IBEA	- Indicator Based Evolutionary Algorithm
IGD	- Inverted Generational Distance
MLS	- Multi-Level Selection
MLSGA	- Multi-Level Selection Genetic Algorithm
MOEA/D	- Multi-Objective Evolutionary Algorithm based on Decomposition
MTS	- Multiple Trajectory Search algorithm
NSGA	- Non-dominated Sorting Genetic Algorithm

Chapter 1 Introduction

1.1 Background

It is predicted that the development of Artificial Intelligence will have a significant impact on the global economy, resulting in an increase of the global GDP by over US$ 13 trillion in the next 10 years [2]. An important subfield of the artificial intelligence are genetic algorithms which are inspired by Darwinian theory of evolution [3]. Up to 2017, Coello [4] has collected over 10000 references relating to the genetic algorithms' development showing a wide interest in this field. Genetic Algorithms (GA) have found applications over a diverse range of industrial and scientific fields, due to their ability to accurately approximate the set of solutions on multimodal problems with multi-objectives and large search spaces. Examples include: medical and biological data classification for Parkinson, diabetes and cancer sicknesses diagnosis [5,6]; adjustment of beam-angle in radiotherapy [7]; DNA sequencing [8] machine learning and artificial intelligence for intelligent stock trading and market forecasting [9]; control systems for greenhouse gases and fuel consumption reduction [10]; design of engineering artefacts, such as space satellite antenna [1];training of neural network for feature selection [11]; and job shop scheduling [12]. Utilisation of genetic algorithms often leads to unusual, but more effective solutions that do not follow the field-related experience, with example in Fig. 1.1, further showing their usefulness.

One of the first ideas of utilisation of evolutionary-like mechanisms into the computational field come as early as 1950s [13]. This was followed by first developments of GAs in the 1960s to study evolution [14–17], before their rapid development in the 1980s and 90s, as they were found to be successful global optimisers [18–25].

In the current state-of-the-art three main branches of algorithms can be recognised: niching methods [26]; the decomposition methods [27]; and hybrid and co-evolutionary approaches [28, 29]. Most of the algorithms from both groups promote, a "convergence-first diversity-second" approach. The main search mechanisms focus on the convergence, achieving the global optimum, while the exploration of the different regions of the search space, diversity, is achieved by secondary mechanisms. The performance of such an approach has been successfully demonstrated on many multi-objective problems with gradually increasing complexity [30–33].

FIGURE 1.1: The 2006 NASA ST5 spacecraft antenna [1]

However, the uptake of genetic algorithms into scientific and engineering applications does not utilise the state-of-the-art solutions from evolutionary computation. The most commonly applied genetic algorithm is NSGA-II developed in 2002 [34] with 14,800 results in the Google Scholar [1] and 30785 citations. The MOEA/D developed in 2007 [35] comes second with 3,250 results in the Google Scholar and 4161 citations[2]. The modern algorithms are significantly less often applied: HEIA (2016) [36] with 81 results in Google Scholar and 69 citations; MOEA/D-MSF (2018) [32] with 2 results and 27 citations; MTS (2007) with 69 results and 40 citations; NSGA-III (2014) [37] with 1500 results and 1792 citations; despite the fact that those algorithms has been shown to outperform both NSGA-II and MOEA/D in multiple comparative benchmarks and competitions [32, 33, 36–39]. It is possible that this is simply due to a certain lag between an academic science and real-world applications, but it is also possible that the benchmarking problems in the evolutionary computation literature do not represent real-world optimisation, leading to genetic algorithms that perform well on these problems, but ineffective in practical applications.

Many of practical problems contains multiple constraints, resulting in discontinuities of the search space, and thus the complexity of their characteristics is significantly higher than of commonly utilised benchmarking sets, making the choice of suitable genetic algorithm non-trivial. Therefore, it is suggested that real-world problems may require a higher diversity and generality of the search, than the current state-of-the-art is providing, and thus this possibility should be explored. It is supported by a high popularity of NSGA-II algorithm. NSGA-II, despite not being the top performer in evolutionary computation problems, is showing a uniform performance across wide range of cases with diverse characteristics where is able to provide a high diversity of obtained solutions [32–34, 36, 40–44]. Therefore, indicating its high generality and diversity focus in comparison to the rest of alternatives in the genetic algorithms field.

[1]All Google Scholar searches have been conducted on 03.01.2020 with the term "*Name_of_the_GA* application" for the 2017-2019 range.

[2]As for 03.01.2020, according to Google Scholar

A potential candidate for the diversity-oriented methodology is Multi-Level Selection Genetic Algorithm (MLSGA) originally proposed by [45]. It has been shown that this approach utilises a unique sub-population-based search strategy resulting in additional evolutionary pressure, which in result encourages the exploration of a particular region of the search space [45]. In total, three Multi-Level Selection (MLS) strategies have been introduced, each with a different preferred region. It is suggested that this strategy can be further modified to simultaneously target multiple regions of the search space leading to a higher overall diversity of exploration in result. This is supported by other research on sub-population-based approaches, as a valid strategy of improving the diversity [46, 47]. However, unlike other sub-population-based methods, which usually utilise a decomposition of the search space, MLSGA does not require extensive hyper-parameter tuning to operate. Sensitivity to hyper-parameters is disadvantageous for real-world applications, due to complexity of those problems and not knowing the location of the global optimum, making the tuning process expensive or even impossible. Furthermore, it is suggested that the generality of MLSGA can be improved by utilisation of co-evolutionary approach, where different sub-populations will use distinct evolutionary algorithm. By utilising a range of search strategies with different areas of applications there is a lower chance that the overall methodology will be ineffective on a particular problem, and thus increasing its robustness.

1.2 Aim and objectives

The aim of the project is to investigate whether genetic algorithm designed for high diversity of the search and a high generality over a wide set of evolutionary computation problems will improve its performance on the real-world problems. This aim will be met through a number of objectives:

- Literature review of the state-of-the-art genetic algorithms and currently utilised benchmarking sets with emphasis on trends in development, their strengths and the flaws.
- Understanding the principles of working of the previously developed Multi-Level Selection Genetic Algorithm:
  - Investigating if the diversity increase from the MLSGA can be retained, while improving its convergence through implementation of the current state-of-the-art mechanisms.
  - Investigating if the generality of the search can be enhanced through co-evolutionary approach.
- Validation of the applicability of the developed methodology:
  - Demonstrating the effectiveness on diverse, with various dominant characteristics, set of benchmarking functions and selected real-world problems.
  - Comparison to the current state-of-the-art.

1.3 Research novelty

Main novelty of this study is to demonstrate the importance of diversity and generality as opposed to convergence by development and understanding of the genetic algorithm with these properties. That knowledge will be used to suggest why the uptake of modern genetic algorithms in real-world applications is slow and to propose improvements to the existing methodologies to make them more suited for the needs of industry.

1.4 Outline of the study

The reminder of this work is organised as follows. The state-of-the-art literature on genetic algorithms and the benchmarking test functions is reviewed and discussed in Chapter 2. In Chapter 3 the principles of working of Multi-Level Selection Genetic Algorithm and its high potential for diversity-oriented methodology are investigated. Chapter 4 presents extension of the original methodology with the current state-of-the-art mechanisms in order to improve its convergence while maintaining a high diversity of solutions, resulting in the hybrid methodology. The co-evolutionary approach is introduced in Chapter 5, where multiple evolutionary algorithms are combined within the MLSGA, in order to maximise the generality. The performance of the MLSGA is validated and compared with the current state-of-the-art on a range of real cases in Chapter 6. This thesis is concluded, with discussion including the indication of potential limitations and the future work in Chapter 7.

1.5 List of publications

P.A. Grudniewski* and A. J. Sobey, "Multi-Level Selection Genetic Algorithm applied to CEC'09 test instances", *2017 IEEE Congress on Evolutionary Computation (CEC), San Sebastian*, pp. 1613-1620, 2017.[1]

U. Mutlu*, P. A. Grudniewski, A. J. Sobey and J. I. R. Blake, "Selecting an Optimisation Methodology in the Context of Structural Design for Leisure Boats", *RINA International Conference on Design and Construction of Super & Mega Yachts 2017*, 2017.

A. J. Sobey* and P.A. Grudniewski, "Re-inspiring the genetic algorithm with multi-level selection theory: multi-level selection genetic algorithm (MLSGA)", *Bioinspiration and Biomimetics*, vol. 13, no. 5, pp. 1-13, 2018.[1]

P. A. Grudniewski* and A. J. Sobey, "Behaviour of Multi-Level Selection Genetic Algorithm (MLSGA) using different individual-level selection mechanisms", *Swarm and Evolutionary Computation*, vol. 44, no. September 2018, pp. 852-862, 2018.[1]

* Main author
[1] Peer-reviewed

P. A. Grudniewski* and A. J. Sobey "Do general Genetic Algorithms provide benefits when solving real problems?", *2019 IEEE Congress on Evolutionary Computation (CEC2019), Wellington*, 2019.[1]

A. J. Sobey*, J. Blanchard, P. A. Grudniewski and T. Savasta, "There's no Free Lunch: A Study of Genetic Algorithm Use in Maritime Applications", *18th Conference on Computer Applications and Information Technology in the Maritime Industries (COMPIT)*, 2019.

J. W. Steer*, P. A. Grudniewski, M. Browne, P. R. Worsley, A. J. Sobey, and A. S. Dickinson, "Predictive prosthetic socket design: part 2—generating person-specific candidate designs using multi-objective genetic algorithms," *Biomechanics and Modelling in Mechanobiology*, 2019.[1]

P. A. Grudniewski* and A. J. Sobey, "Benchmarking the performance of genetic algorithms on constrained dynamic problems", *Natural Computing*, 2020.[1]

P. A. Grudniewski* and A. J. Sobey, "cMLSGA: co-evolutionary Multi-Level Selection Genetic Algorithm", *Natural Computing*, 2021, Under review.

Chapter 2 Literature review

2.1 Genetic algorithms

2.1.1 The history of genetic algorithms

In 1950 Turing proposed the idea of evolutionary-based machine learning, where machines could self-develop and form unique responses after being injected with knowledge and ideas [13]. However, the first numerical testing of evolutionary computing was conducted over a decade later, in 1963, by Barricelli [15]. The primary subject was to evaluate if the evolutionary improvement can be directed to the selected problem and thus produce a desired performance. According to results the individuals at more advanced stage of evolution were of a higher quality than the more primitive ones, due to better abilities in performing necessary tasks and a higher survivability rate. Therefore, Barricelli was the first one to conclude that evolutionary-like mechanisms may lead to overall improvement of individuals.

The first theoretical view of evolutionary inspired genetic algorithms was presented by Holland in 1960s [14, 16, 17]. Later, Holland proved by mathematical analysis that evolutionary reproduction leads to the improvement of population's average quality on a generation-to-generation basis [48]. Thus, further confirming the Barricelli's work. Holland's work led to a growth of the interest in improvement of the genetic algorithms mechanisms and furtherly followed by development of new methodologies. However, the first implementations of evolutionary algorithms for single-objective problems solving in engineering were conducted in late 1960s towards early 1970s by Rechenberg [49] and Schwefel [50], leading to development of evolutionary algorithms branch known as Evolution Strategies (ES).

In 1972 Bosworth et. al introduced the theory of elitism for single-objective cases, as the procedure to improve convergence of solutions by retaining the best individuals between generations [51]. In 1975 De Jong introduced crowding, as the first technique for maintaining the population diversity [52] across the search spaces with single objective. Crowding assumes that individuals in less densely populated regions have higher chance of survival, leading to the expansion into diverse regions of the search space. Both elitism and crowding are successfully utilised in many genetic algorithms that followed their work. They remain useful even in the current state-of-the-art.

Schaffer introduced in 1985 the Vector Evaluated Genetic Algorithm (VEGA) for single- and multi-objective problems, where individuals are grouped based on the best values of each objective, and the parents, for the reproduction process, are selected randomly from these groups [18]. Additionally, the Pareto-optimality concept is introduced for the first time as a form of elitism. In each generation, the offspring population is compared with their parents, and the non-dominated solutions from both groups are selected, which leads to a better convergence as the result. Due to that the VEGA was first GA able to find the Pareto optimal front during a single run, and thus the first multi-objective genetic algorithm. However, due to basing the parents' selection on a single objective only, a strong bias toward extreme values of the objective space could be observed. In this case the individuals with the lowest values of each objective were strongly preferred, and thus there was no incentive for individuals to obtain more "average" values.

In 1989 Goldberg proposed a methodology that become a foundation for the current state-of-the-art niching methods [19]. The Pareto ranking approach, based on Pareto's dominance principle [26], is introduced for the first time for the multi-objective cases, where the population is classified into several sub-groups, based on the individual's dominance relations. The best, non-dominated, points are assigned to the first rank and have a highest chance of being selected. Then, from among remaining individuals the non-dominated solutions are assigned to the second rank. The process repeats until every individual has a rank assigned. Introducing the Pareto ranking allowed removal of the bias towards extreme regions of the search space, occurring in the VEGA, and thus increase overall convergence. The concept of Pareto ranking for multi-objective problems is extended by Fonseca and Fleming to develop the Multi-Objective Genetic Algorithm (MOGA) [20], and further improved to promote convergence by Srinivas and Deb in the Non-dominated Sorting Genetic Algorithm (NSGA) [21]. However, the key breakthrough for the NSGA algorithm was development of the crowding estimator, called fitness sharing by Deb and Goldberg in 1991 [22]. This method expanded De Jong's crowding idea to stronger promote the individuals in a less population-dense regions of the objective space with a lower number of iterations needed to achieve it. However, the significant increase in computational complexity introduced by non-dominated sorting and fitness sharing, and the sensitivity to utilised hyper-parameters limits its usability in practical applications. The concepts of non-dominated sorting, ranking and niching for multi-objective problems have been further studied and followed [23–25, 53–56], and the niching algorithms become the most used branch of the current state-of-the-art genetic algorithms [34, 37].

In Strength Pareto Evolutionary Algorithm (SPEA) introduced in 1999 by Zitzler and Thiele for the multi-objective problems, the front is stored externally, outside of the main population, for the first time [25]. It is used to evaluate the fitness values of the individuals instead of the domination relations inside of the population, unlike the NSGA. The fitness is proportional to the number of points, from the external front, which dominate the individual and the number of individuals dominated by those external points. This approach was introduced to reduce the overall complexity of the algorithm; by eliminating the necessity to compare all individuals with each other to check the dominance relations separately for each rank; and to achieve a better convergence rate. The biggest drawback of this methodology is that the fitness of individuals is based only on the points from the external archive. Therefore, many individuals can potentially have the same rank even if they dominate each other. This resulted in a highly random and ineffective search, and often leading to premature convergence. This issue has been addressed in the SPEA-II [54], where the fitness of individuals is based not only on the relation to the external front, but also on the quality of points that either dominate or are dominated by it. However, the resulting methodology has comparable performance to the NSGA, but with a higher computational complexity.

The idea of a crowding-based selection strategy has been extended by Corne et. al in the multi-objective Pareto Envelope-based Selection Algorithm (PESA) [55] and an improved version of its PESA-II [56], developed for a better convergence and diversity. In both algorithms, the objective space is divided into a number of hyper-boxes, and the fitness of a non-dominated individual is dependent only on the amount of points sharing the box with it. As the points in less crowded areas are always chosen, it is the first methodology that strongly promotes diversity by full integration of the selection mechanisms with the diversity maintenance. Despite the low computational complexity, both PESA and PESA-II are overly sensitive to a number of utilised hyper-parameters reducing its usefulness.

In 1994 Srinivas and Patnaik proposed the Adaptive Genetic Algorithm (AGA), addressing issues with low performance on multimodal, multi-objective problems shown by previously developed methodologies [57]. In this approach, the probabilities of crossover and mutation operations are changing over generations, in order to avoid premature convergence on local optima. Therefore, AGA was the first self-adaptive genetic algorithm. In 1996 this concept is further extended, by Hinterding et. al to develop the Self-Adaptive Genetic Algorithm (SAGA), where the population size adaptation is added, leading to elevated performance on multimodal, multi-objective problems in comparison to AGA [58]. However, the biggest drawbacks of the adaptive-based methodologies are high computational complexity of self-adaptive mechanisms and significant hyper-parameters of those. That is due to the requirement for tuning the algorithm to the shape of search and objective spaces in advance. Therefore, the usability of the adaptive methodologies in real-world is limited.

A different approach to promoting diversity of the search has been proposed by Whitley et. al in the Island Model Genetic Algorithm (IMGA) [46] designed for single-objective problems. In IMGA, the population is divided into the sub-populations, called islands. Each island operates in parallel on the same search space, where individuals are allowed to migrate between groups. As the populations operate semi-independently, they have a higher chance of exploring different regions of search space [46] leading to a better diversity in overall, while the between-groups migration minimises the chance of premature convergence for a single group. The concept of sub-populations is successfully implemented in many single- and multi-objective GA methodologies [59] and has become the foundation for modern decomposition and hierarchical-based algorithms. However, due to the lack of additional diversification mechanisms between the groups in the IMGA, the groups often evaluate the same regions of the search space, especially on more complex problems, leading to an ineffective search.

Furthermore, similarly to adaptive methods the sensitivity to a number of hyper-parameters can be observed.

Similar concepts have been implemented in the micro-GA (MGA) [60] and an adaptive version of its AMGA [61], both designed for multi-objective environments. However, instead of running multiple populations in parallel, a serial approach is utilised, where a single small population is used in a series of runs. After each run, the obtained solutions are stored in the external archive and then used for population initialisation of the next run. These methods were initially developed to reduce the computational time for the practical applications by minimising population sizes and as an approach to maximise the convergence. However, due to strong emphasis on convergence, a low diversity in the final solutions can be observed.

From the presented methods, a high interest in the niching approach can be observed. Most of these methods were developed to improve the convergence of solutions or to lower the computational complexity while maintaining a similar convergence: MOGA, NSGA, SPEA, SPEA-II, PESA-II, AGA, SAGA, MGA, AMGA. With only a few attempts to develop a better diversity preservation mechanism: NSGA, PESA, IMGA. Furthermore, in all of the presented methodologies, except PESA, diversity is maintained by secondary mechanisms resulting in a "convergence-first diversity-second" approach, which is prevalent in the current state-of-the-art.

Here the modern methods are considered to be from NSGA-II [34]. NSGA-II is selected due to a substantial improvement in efficiency and effectiveness of this methodology, in comparison to the previously developed algorithms; as well as its high competitiveness with the current state-of-the-art; and utilisation in a wide range of real-world applications. The leading methodologies are considered to form 3 categories: niching; decomposition-based; and co-evolutionary and hybrid methods. These will be discussed in the following sections. Only the highest performing multi-objective algorithms are reviewed, as illustrated in Fig. 2.1.

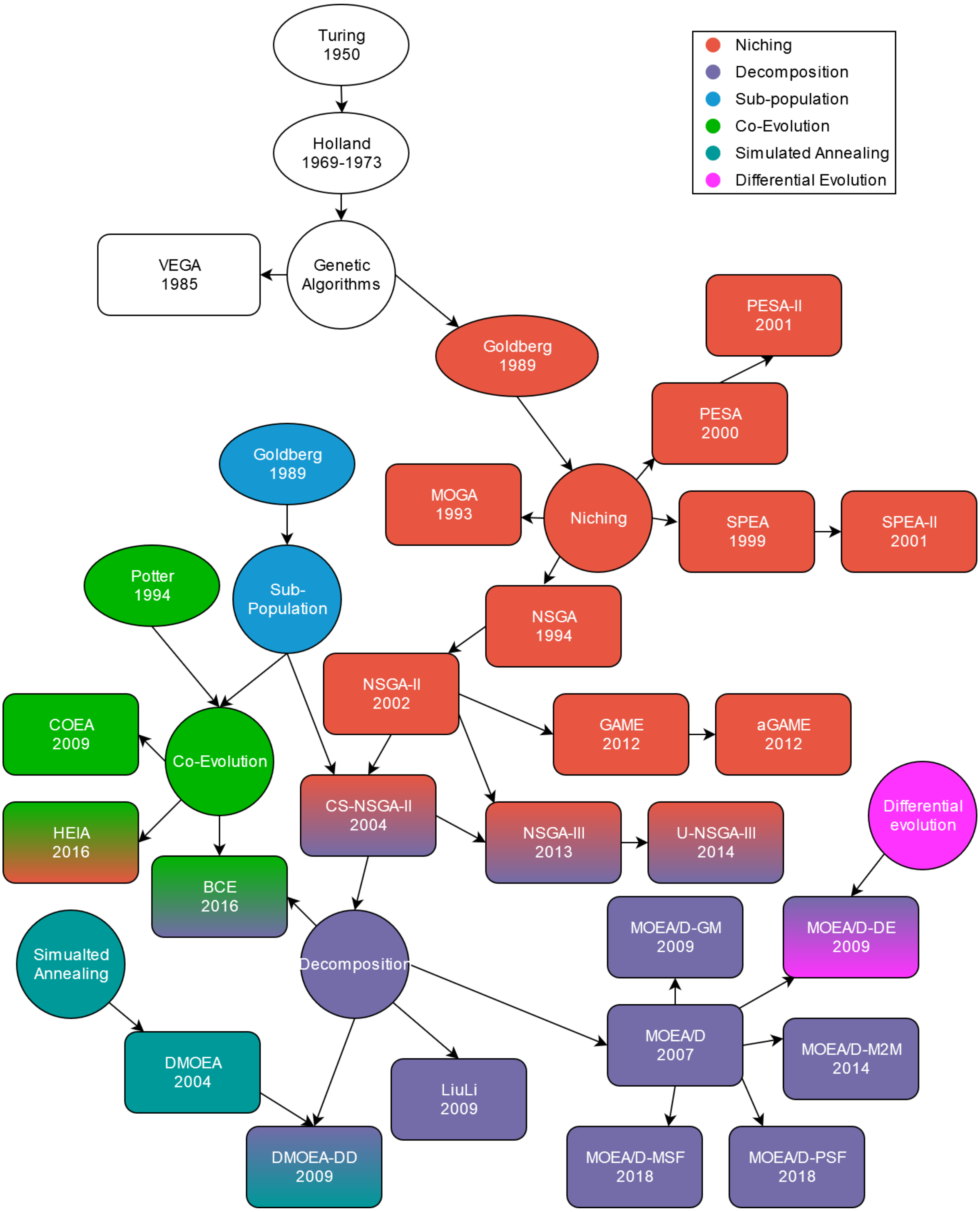


FIGURE 2.1: Development tree of genetic algorithms. The colours represent different approaches in the GAs development.

2.1.2 Niching genetic algorithms

In the current state-of-the-art, niching approach is dominated by improved Non-dominated Sorting Genetic Algorithm (NSGA-II) [34] and the next iteration of it, NSGA-III [37]. In the NSGA-II algorithm, similarly to the original NSGA [21], the population is sorted into

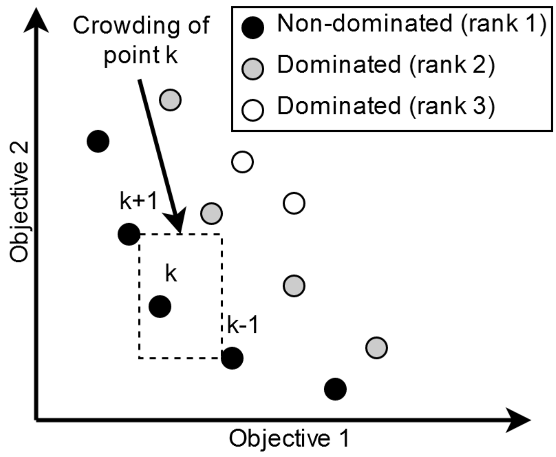


FIGURE 2.2: The principles of working of NSGA-II methodology. Non-dominated sorting and the crowding distance calculation.

several ranks. Individuals in less dominated fronts are assigned a lower rank, and therefore are more likely to be selected for reproduction with prioritisation of individuals in less crowded regions. The principles of this methodology are illustrated in Fig 2.2. The main improvements over the NSGA are the implementation of the Fast Non-Dominated Sorting Approach, allowing the reduction in an overall computational complexity; and improving the crowded-comparison operator to enhance diversity. In the original NSGA each solution was compared with every other solution in order to find the dominance-relations separately for each rank, which required $O(MN^3)$ comparisons in the worst case scenario, where M is the number of objectives and N is the population size. In NSGA-II, the dominance calculation is performed only once for all ranks, allowing reduction of complexity to $O(MN^2)$. In the NSGA, the density of solutions is calculated by using a predefined sharing parameter, which determines the Euclidean range in which two solutions are additively sharing each other fitness. In the NSGA-II the distance between closest solutions is calculated instead, as illustrated in Fig 2.2. This change allows to remove the sharing value, due to its sensitivity to predefined hyper-parameter, and the reduction of the computational complexity of the crowding-distance calculation from $O(MN^2)$ to $O(MNlogN)$. Furthermore, the diversity is strongly promoted by implementation of the crowded-comparison operator into selection procedure, where the individuals in a less crowded areas are always selected from among the same rank. Despite, often being outperformed by more recent methods, the NSGA-II remains the most widely utilised GA and has been successfully applied to diverse branches of science and industry [12,62–69]. The interest can be explained by the fact that NSGA-II shows good performance across the range of multi-objective problems, such as: unconstrained [30,41,70], constrained and with complex Pareto optimal sets [31]; indicating a general-solver approach.. Furthermore, multiple modifications to it have been proposed in order to adjust its performance to the other types, such as dynamic [71], imbalanced [33], large scale [72] and bi-level problems [73].

However, the NSGA-II is ineffective on cases with more than two objectives, due to low selection pressure towards Pareto optimal front and thus a low convergence, in comparison to other methods. The improvements in this aspect have been proposed by Deb and Jain in 2014, resulting in the NSGA-III [37]. Instead of the crowding distance for diversity preservation, a number of the predefined reference points are used, similarly to many decomposition methods [35]. Where the points are used to predict the position of the final front. It is assumed that the ideal solutions will be located on the vectors drawn from these reference points, to the predefined ideal point. This is usually $(0)^M$ for the minimisation problem, where M is the number of objectives. During the reproduction step, the closest individual to each vector is chosen from non-dominated rank, to form a new population. According to the benchmarks, the NSGA-III outperforms the other state-of-the-art algorithms on a range of test instances up to 15 objectives [37, 43, 74, 75] or at least exhibit comparable performance. However, due to utilisation of predefined reference points, the performance of NSGA-III is sensitive to the quality of those points. The positions of reference points have to be adjusted regarding to the shape and geometry of the Pareto optimal front and potential discontinuities in it, similarly to many decomposition methods [32, 47]. Furthermore, due to the small amount of reference points utilised, resulting in small population sizes as the number of individuals is equal to the number of the reference points. Due to that, this methodology is ineffective on one- and two-objective problems. In 2015 the U-NSGA-III is proposed which combines mechanisms from NSGA-II and NSGA-III in order to solve that issue. In this method individuals are selected, based not only on the distance to the reference point, but also on their crowding distance like in the NSGA-II. Furthermore, higher population sizes are allowed due to utilisation of crowding distance as a parameter. Proposed changes lead to performance similar to NSGA-II on one and two-objective problems, but with slightly lowered effectiveness on cases with more objectives in comparison to the NSGA-III. Therefore, U-NSGA-III was successful in unifying the effectiveness of both NSGA-II and NSGA-III, resulting in more universal methodology. However, the sensitivity to reference points has not been addressed in this case.

A similar approach to NSGA-II is utilised in Genetic Algorithm with Multiple ParEto sets (GAME) [76] and an improved self-adaptive version of it (aGAME) [77]. In GAME the population is classified into several Pareto fronts depending on their ranks, and crowding distance mechanisms are used for the diversity preservation. However, in contradiction to NSGA-II, the selection process is less biased towards low-rank individuals. In GAME the rank is used to calculate fitness, instead of being key element of the selection process. Resulting methodology is shown to outperform algorithms such as DMOEA-DD, LiuLi and MTS on multi-objective constrained CF problems [76]. However, the algorithm's behaviour varies on diverse kinds of search spaces due to utilisation of statistically tuned priority parameters for the fitness calculation, resulting in a sensitivity to these parameters. Partial solutions to those problems have been proposed in the self-adaptive version of it, aGAME [77], where those parameters are being automatically adjusted to the search basing on the quality of current population. However, the aGAME introduces additional set of hyper-parameters to manage its self-adaptiveness. Therefore, as the performance evaluation is limited to only 7 constrained function, with no data on other types of problems given. As there is no comparison with the current state-of-the art and no indication of potentially the best hyper-parameters for other kind of problems the generality of both GAME and aGAME is low. According to that, it is likely that the high performance of GAME on constrained problems is due to being adjusted to these cases rather than a better diversity in comparison to other methods.

2.1.3 Decomposition based methods

Unlike the niching methods, which are based on the Pareto-dominance principle, the decomposition-based algorithms operate by splitting the objective or search space into number of subspaces and evaluating each of them separately. This approach originates from the modification to NSGA-II where cone separation of the search space is implemented (CS-NSGA-II), as the mean of improving the efficiency of the genetic algorithms in parallel computing [27]. However, the peak interest in those methods originates in Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) presented by Zhang and Li in 2007 [35]. It was due to significantly higher performance on unconstrained problems such as UF [31], ZDT [30] and DTLZ [40] sets in comparison to other leading methods.

In MOEA/D the multi-objective problem is decomposed into a predefined set of sub-problems, by assigning a distinct weight vector to each individual, and utilising scalarisation method for the fitness calculation. Therefore, each sub-problem has exactly one individual assigned. In the original work three scalarisation methods were proposed: Weighted Sum [78], Tchebycheff (TCH) [78] and penalty-based boundary intersection (PBI) [35]; with Tchebycheff approach proven superior in the performance testing. The vectors remain constants during a single run and the individuals can reproduce only within the same sub-problem or its neighbourhood. The neighbourhood is defined as predefined amount of sub-problems that have the closest weigh vectors based on the Euclidean distances, as illustrated in Fig. 2.3. During the reproduction step the offspring replaces predefined number of worse individuals in the parents'sub-problem or its neighbourhood. Finally, the Pareto optimal front is composed by non-dominated solutions originating from each sub-problem.

Over past decades the range of improvements to different mechanisms of MOEA/D has been proposed: weight generation [79, 80], decomposition [32, 70, 81, 82], Scalarising functions [83, 84], selection [39, 85], reproduction [86, 87] and mutation [88]. In 2016 Trivedi reviewed over 50 different improvements and variants of MOEA/D [47]. However, due to the lack of comprehensive benchmarks between each variant and compatibility tests of distinct improved mechanisms, it is not trivial to choose unquestionably the best methodology. Furthermore, most of them are tested on selected problems with a narrow range of characteristics leading to the family of specialist methodologies, where each member can be used on different kind of problems. From the benchmarks on WFG [41], MOP [32] and UF [31] it can be concluded that the best variants of MOEA/D for two- and three-objective unconstrained cases are MOEA/D-PSF and MOEA/D-MSF [83]. However, as no data is provided on constrained, discontinuous and 4+ objective cases it is not possible to conclude them as the best variants of MOEA/D in overall.

The MOEA/D based methods have been shown to outperform niching and other decomposition methods on unconstrained and dynamic functions, by promoting convergence over diversity [32, 47]. The decomposition removes the necessity of using any diversity maintaining mechanisms, as each sub-problem seek different regions of the search space leading to a stronger focus on convergence. However, due to utilisation of the vector approach based on reference points, low effectiveness can be observed on constrained and discontinuous problems, where gaps on the search and objective spaces can be observed. As it cannot be predicted where the infeasibilities occur, the weight vectors tend to guide the search straight through them, resulting in individuals being stuck on the boundaries of those regions. Thereby resulting in inefficient search. Furthermore, it has been shown that the relationship of the shape and structure between the grid of reference points and the Pareto optimal front can have significant impact on the final performance [32, 47]. In the ideal case, the shape of the grid should be identical to that of global optimum. According to that, those methods usually require prior knowledge about the characteristic of the space in order to adjust the vectors in advance. That knowledge is usually not available on a real-world scientific and engineering cases, which may limit its potential application to real-world problems.

In 2014, the Multi-objective to simple Multi-objective decomposition method (M2M) for MOEA/D framework was proposed in order to balance convergence and diversity [32]. In the MOEA/D-M2M, instead of decomposing the problem into a set of one-objective sub-problems, each with a separate individual assigned to them, the split of the objective space into a set of sub-regions, of the same dimensions, via predefined direction vectors is implemented. Therefore, each sub-population operates on a different sub-region for which the boundaries are defined by the direction vectors. This methodology can achieve better performance than other state-of-the-art algorithms on imbalanced problems [32], [33], but it is outperformed by other methods on biased and unconstrained cases [83] indicating its specialisation. As the decomposition by predefined vectors is applied, similarly to other MOEA/D-based methods, the mechanisms are subject to hyper-parameter tuning, further limiting its generality. Furthermore, as no information is provided on behaviour of this strategy on constrained and highly discontinuous problems and as mostly cases with continuous characteristics have been used for testing, it is not possible to conclude if the high performance of M2M on imbalanced problems is due higher diversity of the search than other methodologies or rather due to implementation of mechanisms tailored to them.

A different way of decomposition was utilised in LiuLi [89] and DMOEA-DD [90] algorithms. In them, the decomposition is applied to the search space instead of the objective space, resulting in higher overall convergence and a better performance on constrained cases [89]. In LiuLi each sub-region has a separate sub-population, called class, assigned to it instead of single individual. Each sub-region has an external archive assigned to it in order to store achieved non-dominated points and thus preserve elite individuals. In the reproduction step, first parent is chosen randomly from among the class, whereas a second one is chosen from its corresponding archive, leading to better convergence. Dynamical Multi-Objective Evolutionary Algorithm with Domain Decomposition technique (DMOEA-DD) [90] combines:

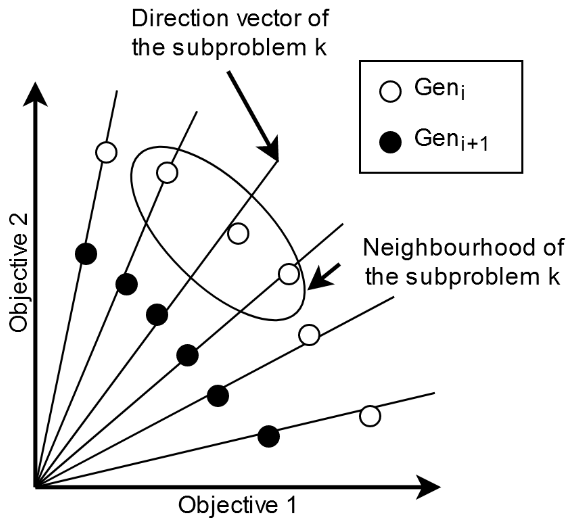


FIGURE 2.3: The principles of working of MOEA/D methodology.

principles of the minimal free energy in thermodynamics for the fitness calculation from its predecessor DMOEA [91], based on the Simulated Annealing [92]; the decomposition of the search space similar to LiuLi; and the crowding distance from the NSGA-II [34].

The effectiveness of both algorithms has been tested on the set of multi-objectives constrained and unconstrained problems with complex POS [31]. According to the final ranking [42], LiuLi [89] and DMOEA-DD are the best genetic algorithms for constrained cases, while being outperformed only by MOEA/D [35] and MTS [93] on the unconstrained problems. However, both algorithms have been evaluated only on the CEC test set [31], with no indication of performance for other types of problems or any following studies. In the presented results, high variation between best and worst performance on different runs can be observed, implying high randomness of the algorithm's behaviour, as at least few independent runs are needed to provide reliable result. Furthermore, both algorithms have non-uniform performance across all tested problems, despite high similarity between them. In addition, due to utilisation of decomposition, a high sensitivity to the hyper-parameters can be observed. According to that, it is not possible to predict the behaviour on other kinds of problems, or even theoretically similar cases, which limits their generality. Furthermore, as those methods has not been evaluated outside of the original publications, it is possible that they are highly ineffective on problems other than the CEC set on which they were originally tested [1].

2.1.4 Hybrid and Co-Evolutionary approaches

Different approach is to combine multiple search strategies, which have distinct advantages and areas of application, via hybridisation or co-evolution. The hybridisation is defined as combining search, update or reproduction mechanisms from different methodologies, whereas

[1] In addition the code of LiuLi and DMOEA-DD is not available online and the publications do not provide enough details to replicate the results

co-evolution is defined as evolving multiple sub-populations simultaneously, potentially each by distinct reproduction mechanisms, with the data exchange introduced between them [29]. According to this definition, each sub-population-based algorithm could be potentially considered as co-evolutionary, even most decomposition methods, which is not accurate. Therefore, in this work these definitions are extended and the hybrid methods are defined as methodologies that utilise distinct mechanisms on the same population in serial way, while co-evolutionary methods combine distinct evolutionary algorithms that operate in parallel on different sub-populations. The simplifications of hybrid and co-evolutionary approaches are shown in Fig. 2.4 and Fig. 2.5 respectively.

Hybrid GAs often incorporate other non-GA evolutionary computation approaches, such as Particle Swarm Optimisation (PSO) [94] or Differential Evolution (DE) [86], in order to adjust the algorithm to characteristics of the search space or to improve the convergence. Examples of adjusting the algorithm include NSGA-II-DE and MOEA/D-DE where typical SBX-crossover is replaced with the DE-based mechanism in order to significantly improve the performance on problems with complicated POS [86]; or the Quantum-inspired GA (QGA) for more effective optimisation of the multi-objective flow shop scheduling [95]. Different approach is utilised in memetic genetic algorithms, where local search methods are incorporated to enhance the convergence rate. One of the first GAs of that type has been proposed by Ishibuchi and Murata in 1998 to increase the convergence on practical problems [96]. Other notable memetic GAs are Multiple Trajectory Search (MTS) [93] and Local Search NSGA-II (NSGA-II-LS) [97]. In MTS four different local-search methods are implemented instead of a single one. Each of them has distinct performance on different structures of the search surfaces, which mitigates the risk of using a single search mechanism. NSGA-II-LS has been shown to increase convergence over the original NSGA-II on unconstrained problems during the CEC'09 competition [31, 42]. MTS was the second-best algorithm across unconstrained

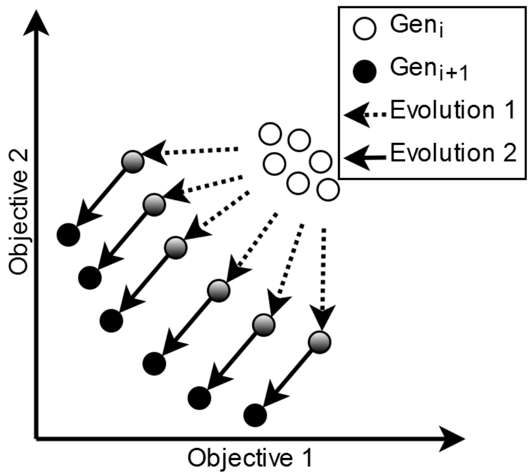


FIGURE 2.4: The simplified methodology of hybrid genetic algorithms

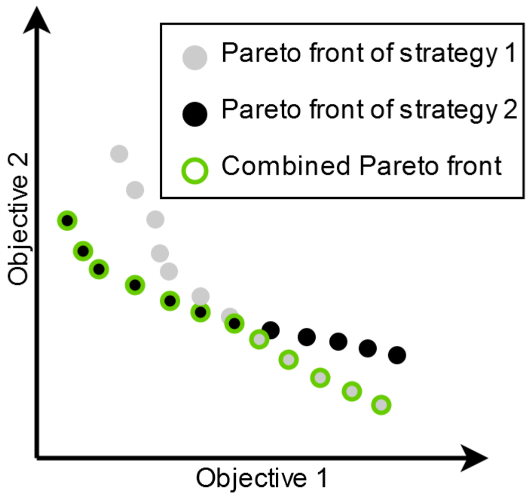


FIGURE 2.5: The simplified methodology of co-evolutionary genetic algorithms

and constrained functions on the same test set. However, robustness of both algorithms is low across the sequence of runs, therefore multiple runs are needed in order to obtain representative results. Furthermore, implemented search methods may be ineffective on the problems with other characteristics than those to which search method was adjusted to. This is possible even if multiple search strategies are utilised. Therefore, in ideal scenario the search strategy should be adjusted in advance to the characteristics of the optimised problem to maximise the performance. According to that, hybrid approach should be considered as a method to specialise the algorithm to specific problem or to increase overall convergence; rather than a mean to improve its generality or diversity.

The co-evolutionary approach was introduced by Potter and De Jong in 1994 [28]. The co-evolutionary methods are usually split into two groups, cooperative [28] and competitive [98], depending on the form of the data exchange between the groups. In the competitive approach, groups compete with each other to form offspring sub-populations [99] or the entire population [36]. In the process, fitter sub-populations have higher chances of mating and producing offspring, and the less-fit sub-populations are encouraged to counter the better ones by quicker adaptation via "arms race" [100]. In the cooperative approach, the individuals are separated into distinct species, e.g. by decomposition of a problem into sub-problems along the search space [28], and they have to collaborate by sharing the genetic information in order to form valid solutions. Originally all groups operated on the same objective space [101], but in more modern methods decomposition is introduced in order to improve convergence [99, 102]. The most significant co-evolutionary algorithms in the current state-of-the-art are: competitive-cooperative CO-Evolutionary Algorithm (COEA) [99], Hybrid Evolutionary Immune Algorithm (HEIA) [36] and Bi-Criterion Evolution algorithm (BCE) [103] due to high performance in comparison to other co-evolutionary methods and high generality of the search. In COEA the problem is decomposed into a set of single-objective problems, where each of them is evaluated by one sub-population. Furthermore, two selection procedures are introduced: competition, where the individuals contest for a place in each group; and cooperation, where distinct species collaborate to create a valid solution. Therefore, cooperation is used to promote convergence, while competition maintains the diversity of gene-pool. The resulting methodology exhibit higher convergence rates than other methods, as shown on the dynamic problems [104]. However, due to decomposition of the search space and solving each variable individually, low performance on discontinuous and irregular problems is observed with poor diversity of the results due to lack of additional diversity preservation mechanisms. Furthermore, COEA has not been tested on constrained and highly discontinuous problems, such as MOP [32], IMB [33] and DAS_CMOP [44], but due to discussed drawbacks a low performance on those cases is expected.

In HEIA, two distinct crossover methods, SBX and DE, are used independently on different sub-populations instead of the problem decomposition. After each generation, the best individuals are moved to the external archive and sub-populations are recreated with the Immune Algorithm based cloning process. Performance of this method has been evaluated on a wide range of unconstrained problems with mostly continuous geometries, such as WFG [41], UF [31], DTLZ [40] and ZDT [30]. The results show comparable performance to the state-of-the-art on all tested cases. However, HEIA is showing significantly better performance on discontinuous cases, such as ZDT3 and UF5-6, and better spread of points along the true Pareto optimal front on all cases, indicating its high focus on diversity of the search. Furthermore, obtained performance is uniform across different test cases indicating high generality of the search strategy. In the BCE instead of implementing different crossover strategies, two distinct fitness indicators are utilised for selection and reproduction: Pareto-based criterion (PC) for convergence and Non-Pareto-based (NPC) for diversity. In PC standard Pareto dominance rule is applied; in NPC the selection is based on the Hyper-volumes [105]. In the performance testing over diverse set of unconstrained problems WFG [41], UF [31], DTLZ [40] and ZDT [30] it has been shown that different search methodologies can be utilised for both criteria. Improvement in the performance over implemented algorithms, MOEA/D and IBEA in this case [103], was observed on average, but not in every tested case. Therefore, the results indicate that combining multiple search strategies under a single methodology leads to better generality in overall. However, both HEIA and BCE are tested only on a limited set of problems, therefore more research has to be conducted regarding the potential of co-evolutionary approach for the general-solver GA.

2.1.5 Methodologies inspired by Multi-Level Selection theories

Another approach to improve the convergence of genetic algorithms is utilised in methodologies inspired by Multi-Level Selection (MLS) theories [106,107]. These sub-population-based algorithms implement an additional group-selection level, where different sub-populations compete with each other for the reproduction in analogous manner to individuals. This generates an additional evolutionary pressure leading to higher convergence rates.

Lenaerts et. al were the first to implement multi-level selection into GAs, by introducing the group selection model [108]. In Multi-Level Evolutionary Algorithm (MLEA), individuals are distributed from the shared pool into isolated sub-populations. In the selection process, first parent is selected from all individuals and its mate is chosen from the same group. As fitter individuals have higher chances of being selected, the groups with “more fit” individuals grow bigger in the process. After a specified number of generations, individuals from all groups are moved to the shared pool, and sub-populations are recreated. This promotes better individuals and results in higher convergence rates. However, no distinct fitness value is assigned to the higher level, and no direct sub-population selection method is implemented. Therefore, this methodology does not introduce an additional selection pressure, but is based on a strong elitism approach, missing the key aspects of the MLS. Due to that and lack of diversity-preservation mechanisms, it is likely that MLEA will exhibit low performance on multi-objective and multimodal problems, which is typical for elitist approaches due to premature convergence [109].

Akbari et. al introduced the colonization-inspired mechanism for interaction between groups of individuals [110, 111]. The between-group dynamics is proposed in which single subpopulation is selected based on its fitness and then is used for generation of two offspring. First offspring replaces the parent population, when the second one has a chance to replace other groups, basing on its fitness and fitness of other groups. The resulting algorithm is efficient on single-objective problems, but no tests on multi-objective instances are provided. The biggest drawback of this mechanism is that the fitness of whole group is based only on the quality of the best individual within it. Therefore, the collective fitness definition and the additional selection levels are implemented in the manner that strongly favours best solutions and therefore leads to loss of many good solutions and thus reduction of the gene pool's diversity. This results in the strong elitism approach which has similar disadvantages as MLEA.

Finally, Wu and Banzhaf presented the hierarchical cooperative evolutionary algorithm based on the multi-level selection [112]. However, in this case the hierarchy is introduced in order to select potential candidates that may cooperatively form a valid solution, similarly to cooperation-based co-evolutionary algorithms, rather than developing an additional evolutionary pressure. Therefore, no separate units of selection are introduced at all levels, even though it is a basic concept of the multi-level selection theory.

All of the algorithms presented in this section are ignoring the key aspects of the theory and do not demonstrate any improvement over current methods. In none of the presented mechanisms, including non-MLS based methodologies, the fitness separation, with an additional collective-based selection pressure, is introduced. Furthermore, in none of those algorithms the potential of an additional evolutionary pressure for a higher diversity is explored, as these focus only on the convergence improvement.

2.1.6 Novelty search

A shift towards diversity-oriented optimisation has been introduced in novelty search algorithms. This methodology has been firstly introduced by Lehman and Stanley in 2008 [113] basing on the NeuroEvolution of Augmenting Topologies (NEAT) method developed by Stanley and Miikkulainen in 2002 [114] for optimisation of topologies of the neural networks. This was followed by a number of publications that explored the importance of novelty search and furtherly developed this method, led mostly by Lehman and Stanley [115–120].

The base principles of working of the novelty search algorithms show resemblance to SPEA [25], SPEA-II [54] and PESA [55] algorithms, where a separate fitness function is introduced to evaluate the quality of individuals, instead of the Pareto dominance principle, but without decomposition of the problem. However, unlike the SPEA, SPEA-II and PESA where the fitness is based on the value of objectives or mutual dominance/non-dominance relations between individuals, a separate factor, called sparseness, is introduced. According to this indicator, the individuals in less populated regions are preferred and thus strongly promoted, leading to a higher diversity of offspring. However, unlike other diversity-inducing indicators, such as the Deb's crowding distance [22, 34], the diversity along the search space is measured and all previously found individuals are considered instead of the current population only.

The novelty search has been shown to be effective on complex deceptive methods where a higher diversity of the search is preferred [113, 115–117]. However, due to discarding the impact of the objective's values in the fitness function, a significantly lower convergence can be observed [119]. This issue has been addressed in 2011 by Mouret [119]. A second fitness value, based on the values of objectives, is introduced, leading to the novelty-based multi-objective algorithm. This methodology has been shown to maintain better balance between convergence and diversity, while strongly promoting the diversity.

However, the biggest disadvantage of novelty search algorithms is in the sparseness indicator used to evaluate diversity. As it is based on the values across the search space, its effectiveness drops exponentially with higher number of variables, due to necessity of evaluating multi-dimensional spaces; and in many kinds of problems the diversity of points on the search space do not translate to better exploration of the objective space or a higher diversity across it, such as biased, imbalanced or discontinuous problems [32, 33]. Therefore, higher diversity of results is not guaranteed. Furthermore, sparseness indicator introduces a number of hyper-parameters, which potentially have to be adjusted to the characteristics of optimised problem, leading to similar issues as in decomposition-based methods; and due to calculation based on all previously find individuals its computational effectiveness is getting exponentially lower with higher number of iterations required to solve the problem reducing its usability for more complex cases.

2.1.7 Diversity in genetic algorithms

In the presented algorithms, strong focus on the convergence-based methodologies can be observed, while the diversity is often maintained by secondary mechanisms, leading to the "convergence-first, diversity-second" approach. In the niching methods, the crowding distance promotes a wide spread of points on the objective space, but more converged solutions are still preferred even if resulting in lowered diversity of the search. In decomposition-based methods, the solutions are selected according to the convergence on predefined weight-vectors, while the co-evolutionary and hybrid methods are developed to usually increase generality and convergence of the search rather than diversity.

Despite the fact that the "convergence-first, diversity-second" methodologies are showing low performance on the diversity-demanding functions [32, 33, 121–124], not much work is done for development of methodologies which would strongly promotes diversity over convergence, "diversity-first, convergence-second" methods. Furthermore, there is no literature that would definitively show that strong convergence is preferred over the diversity while applied to the real-world problems. This indicates a gap in the field of genetic algorithms.

However, the development of genetic algorithms is strongly dictated by established benchmarking functions, as those provide the easiest measurement of the comparative performance. Therefore, it is possible that these functions may be biased towards certain characteristics and thus promote convergence-oriented algorithms. The influence of available benchmarking functions on the genetic algorithms' development is investigated in the following section.

2.2 Test functions and benchmarking

In order to verify the performance of developed methodologies, a number of artificial benchmarking functions have been created. This allowed to operate on problems for which the calculation time is computationally feasible; and for which the final solutions and characteristics of the search area are known in advance, leading to better understanding of the testing algorithm. The complexity of those problems is gradually increasing, but remain significantly lower than most of the practical problems, especially regarding the geometry of the search space and biases towards certain regions; amount of local and global optima; size and shape of the infeasible regions across the search space; and interactions between variables.

Most of the benchmarking problems possess one dominant characteristic such as: imbalanced search space [32, 33]; time dependent Pareto optimal sets and fronts [104, 125, 126]; complex Pareto optimal sets [31]; large quantity of variables [127], defined as 500+ variables, as usually 10-50 are utilised; bi-level [128]; sequence-dependent [129]; constrained [74] and unconstrained [70]. However, as large scale, bi-level and sequence dependent problems, usually are significantly more complex and requires specialist mechanisms in order to effectively solve them, those are not reviewed nor utilised in this work. The most commonly utilised benchmarking functions are reviewed in this section, as summarised in Table 2.1 for unconstrained functions; in Table 2.2 for constrained cases; in Table 2.3 for imbalanced; and in Table 2.4 for unconstrained dynamic problems. Extensive review of all other functions developed up to 2006 can be found in [70]. In addition to the main characteristic, those functions can be described by a number of additional properties, which describe the geometry or complexity of the search space. Those are defined as follows:

Definition 2.4 Convex, concave, linear or mixed: describes the shape of final Pareto optimal front, where in mixed type the shape is a combination of other types.

Definition 2.5 The discontinuous problem contains one or more infeasible regions on the objective space.

Definition 2.6 The degenerate problem contains Pareto optimal front that is of a lower dimension than the objective space in which it is embedded.

Definition 2.7 The biased problems have significant variation in mapping from the Pareto optimal set to the Pareto optimal front. Therefore, some regions on the objective space are preferred.

Definition 2.8 The multimodal problems contain multiple local optima.

Definition 2.9 The deceptive problems are a special case of multimodal problems, where the geometry of the search space favours deceptive local optimum. Therefore, global optimum is localised in the unlikely to find region.

Definition 2.9 The non-separable problem cannot be decomposed into many separated sub-problems that can be solved independently, i.e. variables cannot be optimised independently of one another.

Definition 2.10 The scalable problem can be adjusted to potentially any number of objectives.

Definition 2.11 The variable-linkage problem contains at least one set of variables which numerical values are strongly correlated to each other.

2.2.1 Unconstrained and constrained problems

The first comprehensive set of unconstrained functions is Zitzler-Deb-Thiele (ZDT) set developed in 2000 [30]. A total of six two-objective functions have been presented, including different geometries of the Pareto Fronts, biases and multimodality, as detailed in Table 2.1. However, in all introduced problems, except ZDT6, the first objective function is solely based on a single variable, and, except of ZDT4, Pareto optimal set is for the boundary values of remaining variables with identical parameter value for each of them. As Pareto optimal front consists of almost identical points differentiated only by a value of the first variable, the complexity of the problems is low. Therefore, low diversity of the search is required, as after finding a single point on the global optimum, the Pareto optimal front can be easily expanded by limiting the exploration to the first variable.

The ZDT set has been extended in 2001 by the DTLZ test suite [40]. In this case a total of nine fully scalable test functions are proposed, introducing additional geometries and characteristics of the problems, such as linear, mixed and degenerated fronts, as detailed in Table 2.1. This test suite has similar disadvantages as the ZDT test suite: small parameter domains; global optima which are on the boundaries or centre of the search range and which have identical parameter values for almost every dimension. However, despite these flaws DTLZ remains one of the most popular sets for the multi-objective benchmarking due to its scalability in comparison to the UF test set [31] and the simplicity in defining the true Pareto optimal fronts in comparison to WFG set [41].

In order to mitigate mentioned issues of the DTLZ set, the WFG test problem toolkit has been proposed in 2005 [41]. This toolkit allows to freely create scalable problems with different Pareto optimal front geometries and characteristics of the search space including biases, deceptions, uni/multimodality, degeneration of the Pareto optimal front and non-separability of objectives. In addition, nine exemplar functions are introduced to highlight the possibilities

Table 2.1: Summary of the state-of-the-art unconstrained multi-objective test problems

Name	Objectives	Variables	Problem characteristics
Zitzler-Deb-Thiele set [30]			
ZDT1	2	30	Convex
ZDT2	2	30	Concave, Biased
ZDT3	2	30	Discontinuous
ZDT4	2	10	Convex, Multimodal
ZDT5	2	10	Convex, Multimodal, Binary
ZDT6	2	10	Concave, Multimodal
Deb-Thiele-Laumanns-Zitzler set [40]			
DTLZ1	M	M+4	Linear, Multimodal
DTLZ2	M	M+9	Concave
DTLZ3	M	M+9	Concave, Multimodal
DTLZ4	M	M+9	Concave, Biased
DTLZ5	M	M+9	Concave, Degenerated
DTLZ6	M	M+9	Concave, Degenerated, Biased
DTLZ7	M	M+19	Mixed, Discontinuous, Multimodal
Walking Fish Group set [41]			
WFG1	M	2M+18	Mixed, Biased
WFG2	M	2M+18	Convex, Discontinuous, Non-Separable
WFG3	M	2M+18	Linear, Non-separable, Degenerated
WFG4	M	2M+18	Concave, Multimodal
WFG5	M	2M+18	Concave, Deceptive
WFG6	M	2M+18	Concave, Non-separable
WFG7	M	2M+18	Concave, Biased
WFG8	M	2M+18	Concave, Biased, Non-separable
WFG9	M	2M+18	Concave, Biased, Non-separable, Deceptive
Unconstrained Function set [31]			
UF1	2	30	Convex, Complex Pareto optimal set
UF2	2	30	Convex, Deceptive, Complex Pareto optimal set
UF3	2	30	Convex, Complex Pareto optimal set
UF4	2	30	Concave, Complex Pareto optimal set
UF5	2	30	Linear, Distinct points, Complex Pareto optimal set
UF6	2	30	Linear, Discontinuous, Deceptive, Complex Pareto optimal set
UF7	2	30	Linear, Complex Pareto optimal set
UF8	3	30	Concave, Complex Pareto optimal set
UF9	3	30	Discontinuous, Complex Pareto optimal set
UF10	3	30	Concave, Complex Pareto optimal set

M denotes the number of objectives in scalable problems.

of the toolkit, as detailed in Table 2.1. Due to that, WFG remains the most comprehensive toolkit for multi-objective problems. However, unlike other test sets the final Pareto optimal fronts are not easily computable and have to be obtained from the authors [2]; or to be solved in advance e.g. by running a separate optimisation procedure; limiting its usability.

In 2009 during the Congress of Evolutionary Computation (CEC), [31] presented the test suite that focuses on a more complex Pareto optimal sets, regarding their geometry and structure, and relations to the Pareto optimal front and in-between variables. Total of twelve two-, three- and five-objective unconstrained functions (UF) with various complexity and geometries of

[2] The fronts are not made available online by the authors and are available only from the secondary sources.

TABLE 2.2: Summary of the state-of-the-art constrained multi-objective test problems

Name	Objectives	Variables	Problem characteristics
Constrained Functions set [31]			
CF1	2	10	Linear, Distinct points, Discontinuous, Complex Pareto optimal set
CF2	2	10	Convex, Discontinuous, Complex Pareto optimal set
CF3	2	10	Concave, Discontinuous, Complex Pareto optimal set
CF4	2	10	Linear, Complex Pareto optimal set
CF5	2	10	Linear, Deceptive, Complex Pareto optimal set
CF6	2	10	Mixed, Complex Pareto optimal set
CF7	2	10	Mixed, Deceptive, Complex Pareto optimal set
CF8	3	10	Concave, Discontinuous, Degenerated, Complex Pareto optimal set
CF9	3	10	Concave, Discontinuous, Complex Pareto optimal set
CF10	3	10	Concave, Discontinuous, Complex Pareto optimal set
Deb-Thiele-Laumanns-Zitzler set [40]			
DTLZ8	M	10M	Mixed, Discontinuous, Degenerated, Biased
DTLZ9	M	10M	Concave, Discontinuous, Degenerated
Imbalanced set [33]			
IMB11	2	10	Convex, Imbalanced
IMB12	2	10	Linear, Imbalanced
IMB13	2	10	Concave, Imbalanced
IMB14	3	10	Linear, Imbalanced
Difficulty Adjustable and Scalable Constrained Multi-Objective Optimization Problems set [44]			
DAS-CMOP1	2	30	Concave, Discontinuous
DAS-CMOP2	2	30	Mixed, Continuous
DAS-CMOP3	2	30	Linear, Discontinuous, Multimodal
DAS-CMOP4	2	30	Concave, Discontinuous
DAS-CMOP5	2	30	Mixed, Discontinuous
DAS-CMOP6	2	30	Distinct points, Degenerated
DAS-CMOP7	3	30	Linear, Discontinuous
DAS-CMOP8	3	30	Concave, Discontinuous, Multimodal
DAS-CMOP9	3	30	Concave, Discontinuous

M denotes the number of objectives in scalable problems.

the Pareto optimal set and front has been introduced as detailed in Table 2.1. Importantly, unlike the ZDT set, the objective functions depend on a number of variables and the Pareto optimal set is achieved for non-extreme values of them. This set has been used for the most comprehensive benchmark of the GAs on constrained and unconstrained functions [42] up-to-date. Furthermore, unlike the WFG set this test focuses on the complexity of the Pareto optimal sets rather than Pareto optimal fronts therefore is considered as an alternative to it rather than replacement.

A separate kind of problems are constrained cases where additional conditions must be met by potential solutions in order to consider them feasible. The introduction of constraints leads to occurrence of gaps in the search and objective spaces, as no achievable points exist in those regions. This is illustrated in Fig. 2.6. Due to that, constrained problems require higher diversity of the search in order to avoid infeasible areas and to find the global optimum on highly discontinued spaces. Despite that and the fact that most of real-world

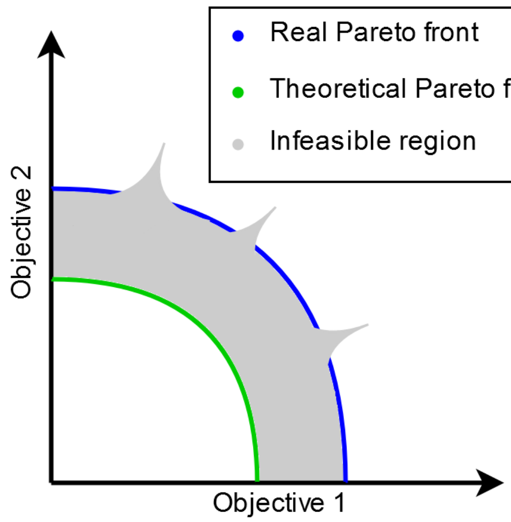


FIGURE 2.6: Infeasible regions and obtainable Pareto optimal fronts on the constrained objective space.

problems contains multiple constraints, less emphasis is given in development and utilisation of this kind of test cases. The most commonly utilised is the constrained functions set (CF) developed in 2009 for the GA competition during the Congress of Evolutionary Computation [31]. In the CF set, total of 10 cases with up to 2 constraints and up to 3 objectives are implemented. Importantly, objective functions are based on the UF set, and thus those contains similar problem characteristics, as detailed in Table 2.2. Similar principle applies to the DTLZ set [40] where 2 scalable cases with up to 2 constraints are included; and for 4 two-objective cases with up to 2 constraints that can be found in the Imbalanced problem set [33]. Therefore, those may be biased towards similar search strategies as their unconstrained counterparts resulting in lowered diversity of available test sets. Furthermore, only 16 constrained cases in total have been introduced in most commonly utilised sets, including imbalanced functions, comparing to the 49 unconstrained problems, showing an imbalance in the problems'development. Only recently, in 2019, this issue has been addressed by DAS-CMOP test suite [44] where 9 test problems with scalable difficulty are introduced. Their difficulty can be adjusted according to 3 parameters: feasibility, convergence and diversity allowing to define on which of these parameters the tested genetic algorithms is focusing on. Therefore, making it easier to understand the principles of working. However, the objective functions are based on the UF and DTLZ sets and thus similar biases may be expected.

2.2.2 Imbalanced functions

Imbalanced problems refer to MOPs where the global optimum consists of multiple parts with varying difficulty to find. Therefore, the Pareto optimal front consists of “easy” and “hard” to find parts, where the “easy-to-find” parts can effectively prevent finding the values of variables that are essential in discovering the other, more difficult, regions. In the various benchmarks [32, 33], it has been shown that those problems may require additional diversity

Table 2.3: Summary of the state-of-the-art imbalanced multi-objective test problems

Name	Objectives	Variables	Problem characteristics
Multi-Objective Problem set [32]			
MOP1	2	10	Convex
MOP2	2	10	Convex
MOP3	2	10	Concave
MOP4	2	10	Mixed, Discontinuous
MOP5	2	10	Convex
MOP6	3	10	Linear
MOP7	3	10	Concave
Imbalanced set [33]			
IMB1	2	10	Convex
IMB2	2	10	Linear
IMB3	2	10	Concave
IMB4	3	10	Linear
IMB5	3	10	Concave
IMB6	3	10	Linear
IMB7	2	10	Convex, Non-separable
IMB8	2	10	Linear, Non-separable
IMB9	2	10	Concave, Non-separable
IMB10	3	10	Linear

enhancing mechanisms, as the typical convergence-first methods fails to maintain necessary diversity due to higher complexity of the search spaces. However, there is no data available regarding if the imbalance is part of many real-world problems, but it is possible due to highly constrained environment of those.

It is relatively new branch of benchmarking functions and only two test sets have been developed, MOP [32] and IMB [33]. Both test sets are summarised in Table 2.3. In MOP suite five two-objective and two three-objective unconstrained problems are included, with various geometries of Pareto optimal front, but without indication of potential modality or variable linkage. For all presented functions, the extreme regions of the Pareto optimal front are preferred, and the simulations shows that most of the algorithms fails to find the points in-between. In the IMB set total of fourteen two- and three-objective functions is included, from which four include constraints. In this case three types of imbalance are introduced: bias; different variable linkages in each region; and constraint isolation where the favoured Pareto optimal front is localised in the infeasible region.

However, both sets are based on ZDT [30] and DTLZ [40] functions, thus they share similar disadvantages, and all introduced problems, except of MOP4, have continuous structures of the Pareto optimal set and front. Furthermore, in all cases the position of favoured and unfavoured regions is dependent on the value of the first variable, deriving the difficulty of the problem to maintaining high diversity of this variable. Therefore, this set is promoting the methods that artificially separate the population and decompose the problem, such as MOEA/D. According to that, the currently developed imbalanced problems may require a higher diversity of the search in theory, but with such a bias towards decomposition methods and specific structure of Pareto optimal sets and fronts, those cannot be considered as impartial for defining the diversity capabilities of tested algorithms. In most of real problems the shape and characteristic of the Pareto optimal front are not know, therefore it is not trivial to hyper-tune the decomposition method to them.

2.2.3 Dynamic problems

In dynamic problems, the geometry and characteristics of the search and objective spaces changes over time. As the problem remain static in between the time steps, it is possible to consider it as set of sub-problems that have to be optimised in series. The major challenge with DMOPs is to find the Pareto optimal front of the current sub-problem in the lowest possible function calls or generations before the environmental change occurs. Therefore, the convergence rate is a crucial aspect of successful DMOP solver, and the diversity of the search is less important [130]. This is shown by a significantly higher performance of the algorithms with high convergence rates, such as MOEA/D or COEA, on these problems [104]. Dynamic functions can be classified into four distinct types based on the dynamics of the objective and decision-variable spaces [131]:

I. Only the Pareto optimal set changes over time

II. Both Pareto optimal set and front are dynamic

III. Only the Pareto optimal front changes over time

IV. Both Pareto optimal set and front are static, although the assigned constraints or objective functions can change over time.

The first comprehensive DMOP benchmarking set was FDA set introduced by Farina et. al [131]. FDA set is based on based on ZDT and DTLZ functions and contains problems of I-III type with various complexity of the change, as detailed in Table 2.4. This set has been widely used in multiple competitive benchmarks [99] indicating its usefulness. However, the FDA set has limited complexity as all included problems are unimodal and have continuous Pareto fronts geometries. Furthermore, due to being based on ZDT and DTLZ functions it shares the same disadvantages as those sets.

The UDF test set was developed in 2014 in order to introduce a higher challenge [130]. This set is based on the UF functions, and thus contains a wider range of problem's characteristics than FDA cases, including multimodality, discontinuity, and non-separability. Similar types of the dynamic change to FDA set are introduced, with the randomness of Pareto optimal front change being introduced for the first time, as detailed in Table 2.4. Randomness of change allows to test the effectiveness in more unpredictable environments. Usually the time changes in artificial problems follow a specific pattern that can be exploited, making the problem easier to solve with front-prediction based methods [132]. Importantly, the

Table 2.4: Summary of the state-of-the-art unconstrained dynamic multi-objective test problems

Name	**Variation/Shift**		**Geometry of the front**	**Type**
	Pareto optimal set	**Pareto optimal front**		
Unconstrained Dynamic Functions set [130]				
UDF1	Vertical shift	Diagonal shift	Continuous	II
UDF2	Curvature change; vertical shift	Diagonal shift	Continuous	II
UDF3	No change	Diagonal shift	Disconnected	III
UDF4	Horizontal shift	Angular shift; curvature variation	Continuous	II
UDF5	Curvature change; vertical shift	Angular shift; curvature variation	Continuous	II
UDF6	No change	Diagonal shift; angular shift	Distinct points	III
UDF7	No Change	Radial shift	Continuous	III
UDF8	Random vertical or horizontal shift	Random diagonal shift; random angular shift or curvature variation	Continuous	II
UDF9	Random curvature change or vertical shift	Random diagonal shift; random angular shift or curvature variation	Continuous	II
Jiang-Yang set [104]				
JY1	Vertical shift	No Change	Continuous	I
JY2	Curvature change	Angular shift; curvature variation	Continuous	II
JY3	Curvature change	Angular shift; curvature variation	Continuous	II
JY4	No change	Discontinuity shift	Discontinuous	III
JY5	No change	Angular shift; curvature variation	Continuous	III
JY6	Vertical shift	No Change	Continuous	I
JY7	Vertical shift	Angular shift; curvature variation	Continuous	II
JY8	No change	Angular shift; curvature variation	Continuous	III
Farina-Deb-Amato set [131]				
FDA1	Vertical shift	No Change	Continuous	I
FDA2	No change	Angular shift; curvature variation	Continuous	III
FDA3	Vertical shift	Vertical shift	Continuous	II

All problems are two-objectives except of UDF7, which is three-objective.

randomness of time change is common in practical problems [104, 132]. Unfortunately, this test set contains only the functions of Type II and III, and only a single discontinuous problem. Therefore, it may promote methods that are adjusted to this kind of change and which prefer continuous search spaces.

The most comprehensive DMOP benchmarking set is JY suite developed in 2017 [104]. This test set not only contains all characteristics of problems implemented in previously discussed sets, but also introduces mixed Pareto optimal front geometry; time-varying multimodality; and novel problem with a mixed type of change. In the mixed type, the environmental change varies between types I-III over time. That approach, similarly to the randomness introduced in the UDF set, allows to track and analyse the performance in more unpredictable environment and to resemble practical problems more closely. Importantly it is the only set that point the similarities of each function to real-world cases and which is not based on previously developed static problems.

2.2.4 Bias of the benchmarking problems

Multiple benchmarking functions have been developed to evaluate the performance of GA methodologies over a range of problems with distinctive characteristics. It can be observed that most of the cases are based on the same frameworks, ZDT or UF; or have similar definition of objectives and thus similar search spaces; or those are developed by the same groups of researchers. This is illustrated in Fig. 2.7. Importantly, most of the presented sets are co-developed by the same authors whom developed two of the most popular methodologies NSGA-II and MOEA/D. Deb, author of NSGA-II [34], co-developed ZDT, DTLZ, DAS-CMOP, IMB and FDA sets, while Zhang, author of MOEA/D [35], took part in development of UF, CF, MOP and DAS-CMOP test problems. This may indicate a development-bias towards those top-performing genetic algorithms, which is supported by high performance of MOEA/D, NSGA-II and NSGA-III on those problems.

Most of the sets are limited to continuous and unconstrained characteristics, despite the facts that many, if not most, optimisation problems have various constraints resulting in discontinuities along the search and objective spaces. Furthermore, discontinuous cases are considered to be significantly more complex to solve [133]. From the 74 discussed static problems, only 22 are discontinuous, and 25 have at least one constraint.

Evaluating the performance and comparing the novel methodologies over a limited range of similar problems may lead to over-development of genetic algorithms that have a strong bias towards certain characteristics, and which are able to effectively solve only those functions.

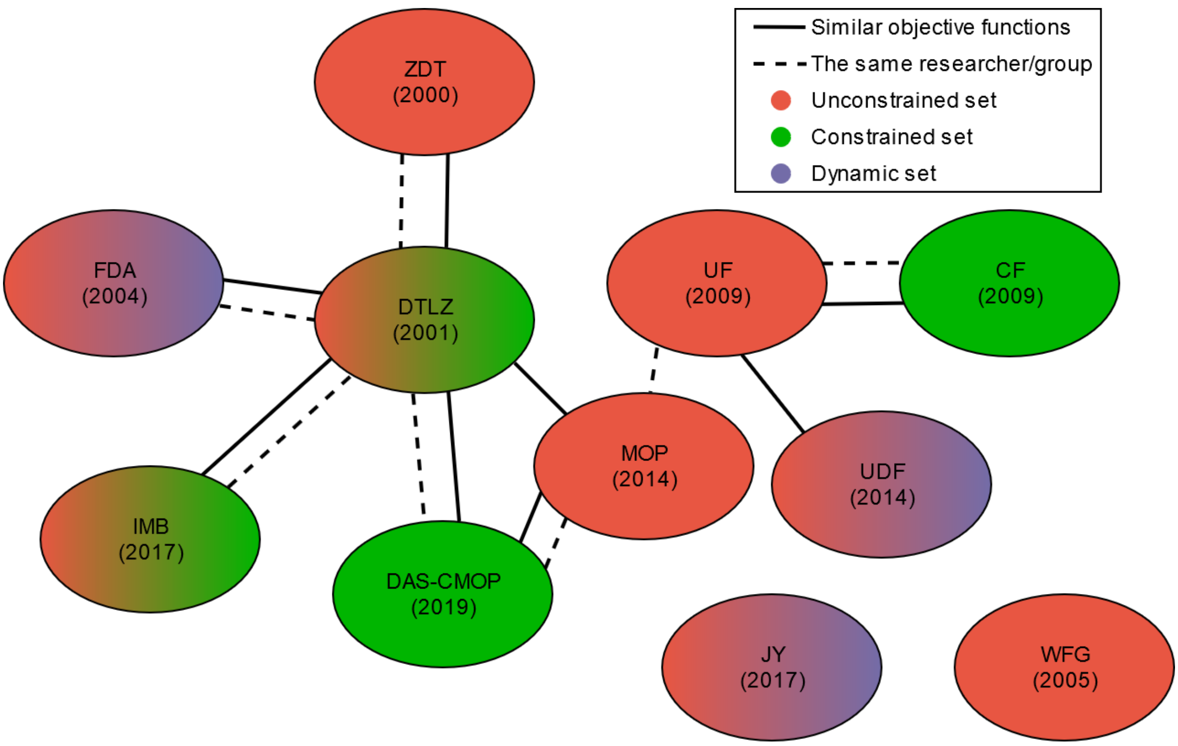


FIGURE 2.7: Relations tree between developed Multi-Objective Problems.

Only recently, it has been noticed that currently existing test sets are promoting strategies focusing predominantly on the convergence and are neglecting the importance of the diversity preservation [32, 33].

Despite described issues with developed artificial test sets, resulting in narrowed range of available characteristics and complexities of the problem, most of the developed GA methodologies are tested only on a few arbitrary-chosen problems instead of all of them. This is detailed in Table 2.5. As it can be observed only NSGA-II and MOEA/D are tested on all presented test sets, except of MOEA/D on MOP, while most of algorithms are tested on 2-4 sets only. Furthermore, higher emphasis is put on unconstrained problems rather than constrained and imbalanced problems, as indicated previously. According to that, most of developed genetic algorithms can be considered as specialist-solvers. However, due to differences between artificial and practical problems, especially regarding their complexity, the question is raised if those specialist-solvers, which are tested on a limited set of problems with narrow range of characteristics, will achieve correct results on real problems.

According to discussed issues, a higher emphasis is needed on the development of general-solver genetic algorithms to produce good quality results on a wide range of problems. Promoting a consistent performance across the cases with divergent characteristics, will allow to maximise the chance of success on real problems as there is no free lunch.

Table 2.5: Summary of the literature on the benchmarking sets used to test genetic algorithms.

Algorithm	Test set									
	Unconstrained				Constrained				Imbalanced	
	ZDT	DTLZ	WFG	UF	CF	DTLZ	IMB	DAS-CMOP	MOP	IMB
NSGA-II	[34, 36]	[36, 40, 43]	[36, 41]	[36, 42]	[42]	[40, 43]	[33]	[44]	[32]	[33]
NSGA-III	[43, 103]	[37, 43, 103, 134]	[37, 103]	[103, 134]		[43, 74]		[44]		
U-NSGA-III	[43]	[43, 134]		[134]		[43]				
GAME					[76]					
aGAME					[77]					
MOEA/D	[35, 36, 103]	[35–37, 103, 134]	[36, 103]	[36, 42, 103, 134]		[74]	[33]	[44]		[33]
MOEA/D-DE				[88]	[88]				[32]	
MOEA/D-M2M			[83]	[83]					[32, 83]	
MOEA/D-PSF			[83]	[83]					[83]	
MOEA/D-MSF			[83]	[83]					[83]	
LiuLi				[42]	[42, 76]					
DMOEA-DD				[42]	[42, 76]					
COEA	[99]	[99]								
HEIA	[36]	[36]	[36]	[36]						
BCE	[103]	[103]	[103]	[103]						
MTS	[93]	[93]	[93]	[42]	[42, 76]					

2.3 Summary of the literature review

In the literature review chapter, the work conducted on multi-objective problems optimisation using genetic algorithms was reviewed. The history and current trends in genetic algorithms development as well as multi objective benchmarking test suites utilised to verify their performance and understand their principles of working were discussed. From this review, the following issues about the current state-of-the-art of genetic algorithms can be highlighted:

- The field is dominated by methodologies promoting convergence over diversity, called "convergence-first diversity-second" GAs, which demonstrate low performance on the diversity-demanding functions [32].
- In many methods, the diversity is maintained by decomposition of the problem or introduction of a diversity evaluating metric, requiring hyper-tuning in advance and extensive knowledge about the characteristic of the search space to work effectively [47]. That information is usually not available for most of real-world applications.
- The commonly utilised benchmarking sets promote specialist approaches. Algorithms are tested on a limited range of problems with similar characteristics.

According to that, two gaps in the genetic algorithms' development can be identified: low interest in the diversity-oriented methodologies; and low emphasis on the generality of search strategies. Furthermore, it is yet to be investigated if the specialist/convergence or the general/diversity approach is preferred for real-world applications. However, due to high popularity of NSGA-II, general-solver GA, in engineering and scientific applications and a low uptake of newer methods, it is suggested that the general/diversity approach is preferred for real-world applications

Possible approach is utilisation of a sub-population strategy, where each group is exploring different regions of the search and objective spaces, resulting in improved diversity. Ideally this methodology should not utilise decomposition or other mechanisms that require extensive hyper-tuning towards the problem's characteristic, to increase its usability for the real-world applications. Another approach could be utilisation of the co-evolution in order to improve the generality of the search. In this case the risk that a single search strategy will be ineffective with the characteristic of a given problem is mitigated. This can be particularly useful for the real cases where the search and objective spaces are not well understood.

Chapter 3 Development of MLSGA

The early version of MLSGA has been developed and initially benchmarked by Paksuttipol and Sobey for solving weighted single-objective problems [135]. However, low performance on multi-objective cases could be observed, due to incorrectly coded collective reproduction mechanisms. Sobey and Grudniewski [45], show that the collective-level selection and reproduction mechanisms are generating additional evolutionary pressure leading to extended diversity of the results. In this chapter the principles behind this phenomenon are further investigated. It is evaluated if this additional selection pressure can be directed into different regions of the objective space by using divergent fitness functions definitions on each level of selection. It is possible that by combining several types of fitness assignment in separate sub-populations, the methodology with disperse region-based search may be achieved leading to even better overall diversity of the search. Importantly, due to utilisation of simple split in the fitness function, rather than forced decomposition of the objective space, no extensive tuning to the characteristic of search space will be needed, unlike weighted approaches or other decomposition-based methods. Therefore, utilisation of such a method will be potentially better for practical applications.

3.1 Levels of Selection

MLSGA is based on the multi-level selection theory, which was re-inspired by Wilson and Sober in 1989 [106] and further reviewed by Okasha in 2006 [107]. It is not implied here that the MLS theory is true explanation of evolutionary processes, as the biological evolution is complex topics, and many divergent theories have been proposed with an attempt to describe it [136, 137]. However, this theory provides a promising inspiration for new GA methodology, as it provides a simpler way of breaking the problem into subsets and to increase the evolutionary pressure. The collective-level mechanisms can be put in parallel with currently existing individual-level evolutionary algorithms leading to more flexibility in the fitness function interpretation and potentially a discovery of new solutions with a higher diversity.

In typical GAs, only one level, called individual-level, is considered. The individuals are chosen for reproduction from among themselves and even if a grouping is introduced, those groups are not subjects for a separate selection procedure. MLS theory proposes the idea that evolution may occur separately at distinct levels of a hierarchical structure, which is normally viewed as nested hierarchy with one level being enclosed within another. For example, in the basic two-level organisation, the individuals form collectives in order to pass some of their survival responsibilities, such as food gathering or danger elimination, to the whole group, increasing their chances of survival. Inside each collective, every individual maintains its personal fitness and the natural selection between them still occurs. Additionally, each collective has a separate fitness value that describes the quality of individuals inside, according to their usefulness for the group. Therefore, different collectives, similarly to the individuals, may compete with each other, e.g. for food sources or occupied space, where better groups survive and further reproduce, whilst less fit sub-populations go extinct. In this model, individuals need to not only improve themselves, to maximise their chances for reproduction, but also maintain the survivability of whole collective.

According to Okasha the levels-of-selection problem is a consequence of three factors [107]:

- The natural selection, according to which fitter individuals produce more offspring than others, which drives the evolution.
- The adaptation of species to the environment in order to increase their chances of survival, such as development of thick furs by polar-regions animals; fangs or claws by predators; or camouflage-like colours of skin.
- The hierarchical organisation mentioned before

According to that, two types of multi-level selection (MLS) have been proposed, determining the way of fitness calculation [107]. In the first type (MLS1), survival of all levels of selections depend on the same evolutionary factor. Collectives and individuals have the same "goals" and focus on improvement of the same parameters. Therefore, the fitness of the group is an aggregate of the individuals inside of it. In the second type (MLS2), the fitness, and thus the selection, is different on each level. The success of whole group depends on different factors than the success of individuals inside. Due to that, the individuals are forced to simultaneously develop properties needed for their own well-being and those necessary for survival of the, even if these are contradictory. This generates an emergent property to selection on each level, leading to competition between them and a higher diversification of individuals.

Basing on those definitions three variants of multi-level selection are here introduced: MLS1, MLS2 and MLS2R. Fitness function definitions used in each MLS type are shown in the Table 3.1, where $f_1(x)$ and $f_2(x)$ are first and second fitness function of the optimised problem respectively.

Table 3.1: Fitness functions used on each level of selection in two-objective optimisation, depending on the MLS type

	MLS1	**MLS2**	**MLS2R**
Individual-level	$\frac{f_1(x)+f_2(x)}{2}$	$f_1(x)$	$f_2(x)$
Collective-level	$\frac{f_1(x)+f_2(x)}{2}$	$f_2(x)$	$f_1(x)$

3.2 Methodology of the MLSGA

3.2.1 Description of mechanisms

In this section the MLSGA mechanisms inspired by the concept of multi-level selection are outlined. The algorithm works as follows with a more detailed description of the MLSGA mechanisms provided below [1]:

Inputs:

- **Multi-objective problem;**
- **Np: Population size;**
- **Nc: Number of collectives;**
- **Ne: Number of eliminated collectives;**
- **Fitness function definitions for each level;**
- **Stopping criterion;**

Output: External non-dominated Population (EP)

Step 1) Initialisation:

Step 1.1) Set $EP = NULL$.

Step 1.2) Randomly generate the initial population P of N_p individuals $\{x_j, \ldots, x_{Np}\}$.

Step 2) Classification:

Step 2.1) Classify the individuals from the initial population P into N_c collectives, $\{C_i, \ldots, C_{Nc}\}$, so each of them contains a separate population $\{P_i, \ldots, P_{Nc}\}$. Classification is based on the search space.

Step 3) Individual-level operations:

[1] The code of MLSGA, and all subsequent implementations of it, is made available online at https://bitbucket.org/Pag1c18/cmlsga

$For \quad i = 1, \ldots, N_c \quad do$

Step 3.1) Individual-level GA's operations: Perform the selection, crossover and mutation procedures with elitism. Selection is based on the fitness function definition from the individual-level.

Step 3.2) Update External Population:

$For \quad j = 1, \ldots, |P_i| \quad do$

Remove from the EP all solutions dominated by x_{ij} (the individual j, from population i). Add individual x_{ij} to the EP if no solutions from EP dominate x_{ij}.

Step 4) Collective-level operations:

Step 4.1) Calculate collective's fitness:

$For \quad i = 1, \ldots, N_c \quad do$

Calculate the fitness of collective C_i as an average fitness of population P_i based on the collective-level fitness definition.

Step 4.2) Collective elimination:

Find collective C_i with the worst fitness value, and store the index of that collective, z.

Store the size of eliminated collective $|P_z|$ as variable s.

Erase sub-population P_z of eliminated collective C_z.

Step 4.3) Collective reproduction:

$For \quad i = 1, \ldots, N_c \quad do$

$if \quad (i \quad != \quad z)$

Copy the best $\frac{s}{(N_c-1)}$ individuals, according to the collective-level fitness definition, from population P_i to P_z.

Step 5) Termination: If the stopping criteria are met, stop and give EP as an output. Otherwise, return to **Step 3)**.

In step 1 the initial population is created. Each individual has every variable assigned randomly according to the search space boundaries for the given problem.

During the Classification, Step 2, the supervised learning classification method is used to assign the collective labels to each individual in the initial population, basing on the distances between them in the decision variable space. In this work the multi-class classification SVM with C parameter, called C-SVC, and with a linear function is used. The utilised code has been taken from LIBSVM open library [138] and the training parameters have been left the same as in the publication. In this method the user predefines a number of label types, and thus the number of collectives, rather than size of each group. This means that, the number of individuals in each collective is highly likely to be divergent. Due to that, the minimum size of 10 individuals and the maximum size of half of the overall population is predefined in order to avoid empty, small or too big collectives as it has been shown in pre-benchmarks to negatively impact the overall performance. Organising the collectives with most similar individuals inside has been shown to be more beneficial over the random initiation after testing using three different classification variants: SVM, k-means clustering and random assignment. Clustering and SVM exhibit similar performances, but the SVM is chosen due to its lower calculation time and a higher robustness. The classification according to the decision variables, instead of the objectives, is utilised in order to retain diversity of solutions inside the collectives.

Inside each collective evolutionary operations are applied to the individuals in the same manner as in typical GA, Step 3. However, with distinction that in this case only a single fitness value is utilised, according to individual-level fitness definitions, as described previously in this Chapter. Therefore, case is treated as a single objective, and concepts such as Pareto-dominance are not needed. In step 3.2 the non-dominated solutions from all collectives are stored in an external population. However, the external population is used as a storage only and is not used for any calculations or the fitness evaluations.

During the collective elimination, Step 4.2, the collective with the worst collective fitness is eliminated and all of the individuals inside are erased. This collective is repopulated in Step 4.3 by copying the best individuals, according to the collective fitness definition, from all of the remaining collectives. This is done in order to maximise the fitness of offspring collectives. Importantly the size of the collective remains the same and this procedure has no effect on parent collectives. Therefore, Step 4.3 is the only step where collectives are able to "communicate" with each other. In between the various levels of selection, the only information passed is the fitness of the individuals, necessary to calculate the collective fitness in Step 4.1.

The algorithm continues in loops from Step 3, unless the termination criterion is met. Here the maximum number of fitness evaluations is utilised, which is the most common approach in the state-of-the-art literature.

3.2.2 Hyper-parameters setting and benchmarking

At the current stage, simple mechanisms are implemented to easily study the impact of proposed strategy without the interference of more complex methods. Therefore, in order to operate on real values, the simulated binary crossover (SBX) [23] and the polynomial mutation [53] are used. Initial operating parameters are based on the best practice in current state-of-the-art. These are listed in Table 3.2.

The benchmarking of GAs on test problems is performed over 30 separate runs with 300,000 function evaluations for each run, following the CEC'09 regulations [31]. Adaptation of best-practice standards will allow reproducibility and continuity of the benchmarking process.

Most of the developed performance metrics are able to signify both convergence and diversity of the solutions simultaneously, but each of them has certain limitations as indicated in [139].

Table 3.2: Initial GA mechanisms and hyper-parameters used within the MLSGA

Step	Parameter	Value
1. Initialisation	Type	Random
	Encoding	Real values 16 decimals
	Pop. Size	800
2. Classification	Method	SVM
	No. Collectives	8
	Collective size limits	min 10 individuals max 1/2 of pop. size
3. Individual-level operations		
Fitness Evaluation	Type	According to Table 3.1
Selection	Type	Roulette wheel
Mating	Crossover type	Real variable SBX
	Crossover rate	1
	Mutation type	Polynomial
	Mutation rate	0.08
Elitism	Rate	0.1
4. Collective-level operations		
Fitness Evaluation	Type	According to Table 3.1
Elimination	No. eliminated collectives	1 every generation
5. Termination	Criterion	300,000 fitness evaluations

In this thesis the Inverted Generational Distance (IGD) [140] and Hyper Volume (HV) [25] are selected as the most widely used metrics in the current state-of-the-art. Both have different emphasis on the quality of solutions regarding the convergence/diversity, where IGD promotes convergence and HV the diversity.

IGD measures the average Euclidean distance between each point in a true Pareto optimal front and the closest solution from the achieved set and is calculated according to eq. 3.1.

$$IGD(A, P^*) = \frac{\sum_{\nu \in P^*} d(\nu, A)}{|P^*|}, \tag{3.1}$$

where P^* is a set of uniformly distributed points along the true Pareto Front in the objective space, A is the approximate set to the Pareto Front being evaluated and $d(\nu, A)$ is the minimum Euclidean distance between the point ν and points in A.

Lower scores for this metric indicate a higher convergence and uniformity of the points, where the ideal value is 0. However, IGD is overly sensitive to the predefined reference set, which ideally should be as big as possible and with perfectly uniform spread of points in order to avoid bias toward certain regions.

HV is a measure of the objective space's volume between a predefined reference point and the obtained solutions and is calculated here according to [141], which provides the fastest and most widely used method for HV calculation. This indicator is to be maximised and has a stronger focus on diversity and edge points, but with slight bias towards the knee points and concave regions. As the results are highly dependable on the predefined reference point, usually it is not possible to compare results coming from different publications, as there is no standard procedure to define this point. Additionally, there is not a "best" value for HV indicator, and thus it cannot be used to define the quality of obtained Pareto optimal front by itself but rather is used to compare the performance of different algorithms. HV metric value of 0 indicates that all of the points obtained from the front are dominated by the reference point or that no feasible solutions have been found.

In this thesis, 1,000,000 uniformly spread reference points are generated as the true Pareto optimal front, for the purposes of IGD calculation. The reference point for HV is defined as $(1.1 * z_1, \ldots, 1.1 * z_m)$ where z_i is the maximum value of the i-th objective of the reference front, and M is the number of objectives.

In all conducted performance comparisons, the average values of the metrics across all runs are utilised and presented in the main part of this work. The more detailed data, including min and max values and standard deviations are attached as supplementary material. Wherever applicable, the Wilcoxon's rank sum test was conducted to assess the statistical significance of the differences between obtained results with the confidence level of $\alpha = 0.05$. The representative graphs are selected according to the closest-to-the-average values of IGD, unless stated otherwise.

3.2.3 Computational complexity and constraint handling

The computational cost of one generation of MLSGA algorithm is determined by the most complex of three main operations: the non-dominated sorting of the fronts; individual reproduction step; or the offspring collective generation. The front selection utilises the fast non-dominated sorting approach from NSGA-II [34] and therefore is bounded by $O(mN^2)$, where m is the number of objectives and N is the size of evaluated population. For the individual reproduction step, as the full replacement of old generation with elitism takes place, only $O(N^2)$ computations are needed at most. Only one of the objectives is considered in the process, therefore m is discarded. For the offspring collective generation, the fitness evaluation requires $O(N)$ computations at most, as the average of single valued fitness of the individuals inside is calculated. The actual collective replacement takes $O(N^2)$ complexity at most, as the best values have to be found.

Therefore, the overall computational complexity of MLSGA is bounded by the non-dominated sorting that requires $O(mN^2)$ calculations at most, which is not higher than most of the solutions in the current state-of-the-art. The complexity of NSGA-II, NSGA-III, MTS is $O(mN^2)$;

in case of MOEA/D it is $O(mNT)$, where T is number of solutions in the neighbourhood and is typically $0.2N$; and $O(N^2)$ in case of BCE and HEIA.

For the constraint handling the constraint-domination principle taken from NSGA-II is adopted, which can be defined as follows [34]:

Definition 3.1 an individual x_1 is said to dominate another individual x_2, if: x_1 is feasible and x_2 is infeasible **or** both x_1 and x_2 are infeasible, and x_1 has a smaller constraint violation (CV) value **or** both x_1 and x_2 are feasible, and x_1 dominates x_2 with the standard fitness domination principle.

This is applied during the selection process whenever two individuals are compared in the constrained problems optimisation. However, at this stage no direct constraint handling is introduced on the collective-level. As the overall fitness of a collective is not affected by infeasibility of the individuals inside of it, collectives can maintain a higher diversity of solutions inside of them.

3.3 Early-stage benchmarks

The effectiveness of developed genetic algorithm is tested over selected two-objective functions with various characteristics. At this stage only problems with two objectives are considered, as they allow to study cases where the behaviour is simple to interpret, while providing enough complexity to replicate a number of real-world problems.

According to that, MLSGA is benchmarked on the unconstrained ZDT1-6 problems [30], where there is a strong imbalance between the complexity of both objective functions; and unconstrained/constrained cases with a higher complexity of the Pareto optimal sets [31], UF1-7 and CF1-7 functions. In this case only the graphical representation of results is presented, as considered to provide more insight into behaviour of different MLS variants than the performance indicators.

3.3.1 Unconstrained ZDT problems with simple characteristics

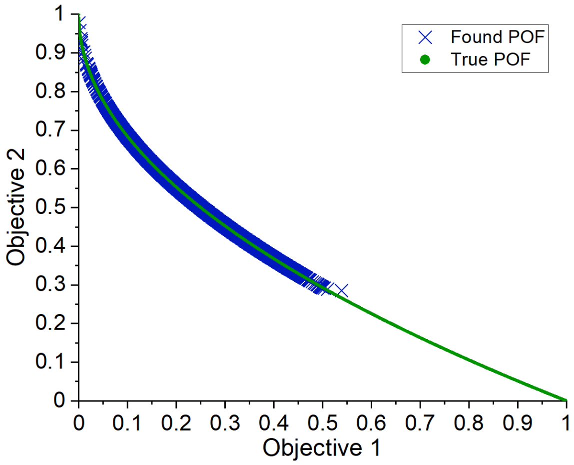
In order to investigate the differences in behaviour between MLSGA variants, the preferred areas of search for each of them are illustrated using plots of the achieved Pareto optimal fronts and the heat maps. The fronts for the ZDT1 test instance are shown in Fig. 3.1. The fronts for the more complex ZDT6 function are illustrated in Fig. 3.2. Heat maps are illustrated in Fig. 3.3 and 3.4 for ZDT1 and ZDT6 problems, respectively. Those two functions are chosen as they illustrate the biggest contrast in behaviour between variants. In addition, the plots for "simple GAs", with sub-population and single population variant, are included to investigate the impact of the population splitting on the final performance and to demonstrate the performance improvement introduced by MLS mechanisms. Simple GA utilises the same selection, reproduction and elitism mechanisms as MLSGA, but without the fitness separation or the collective reproduction.

Heat maps illustrate distribution of the search on the objective space. For that purpose, the concentration of individuals inside each region is calculated by considering all evaluated individuals during a single run. On the maps, the brighter areas, indicates a higher density of individuals, and thus higher focus of the search, with peak shown by green colour; whereas dark areas represents a low concentration of individuals and thus search. The maps are presented for the comparative purposes, with enlarged graphs attached in Appendix G.

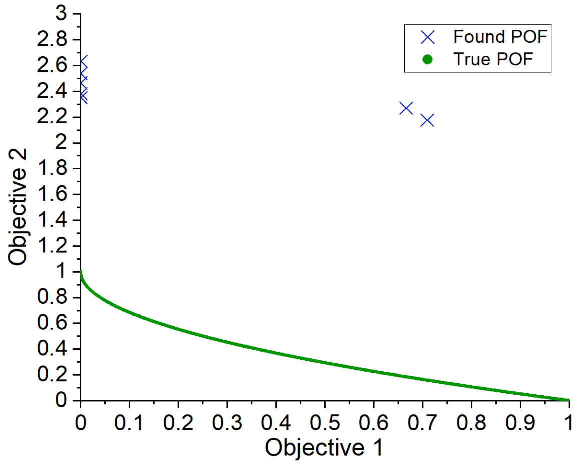
For ZDT1, Pareto optimal fronts in Fig. 3.1 and heat maps in Fig. 3.3, it can be seen that MLS2R is the only variant with a wide area of search and a good coverage near the real front, range of 0-0.65 according to first objective. The search is initially focused on the "left" region of Pareto optimal front, but then it spreads towards lower values of the second objective. MLS1 focuses its search on the central areas with more average fitness, region with 0.3-0.35 values of the first objective. Due to that, its Pareto optimal front is more limited, range of 0.1-0.5 according to first objective. MLS2 stops on the left side of the front and is unable to move further. Comparing the MLSGA variants to the simple GAs, with and without sub-populations, it can be seen that simple GAs obtains comparable results to the MLS1 variant, with a strong bias towards central regions of the objective space. Similarity between those variants was expected as in all of them the selection is based on the averages of both objectives. However, comparing heat maps for those algorithms it can be seen that search pattern of a simple GAs is more dispersed, range of 0.2-0.45 according to the first objective, showing higher focus of MLS1 strategy.

For ZDT6 the Pareto optimal fronts, Fig. 3.2, of all variants are similar, but MLS2 has a significantly better convergence. Comparing the heat maps Fig. 3.4 it can be seen that all MLSGA variants are avoiding central regions of the objective space. This behaviour is caused by the bias towards extreme regions of the objective space, points (0,1) and (1,0), introduced in ZDT6 problem. Similarly, to the previous case, preferred regions of each MLS variant can be observed. In MLS2 the left side of the objective space is explored more often, while in MLS2R the search is shifted towards the right side. MLS1 shows similar search pattern to MLS2R, but with a lowered convergence. Similarly, to ZDT1 function, the single population GA is showing a similar, but more dispersed, search pattern to the MLS1 variant of MLSGA. Interestingly, sub-population GA is showing similar behaviour to the single-population variant and MLS1, but with a better convergence of points on true Pareto optimal front.

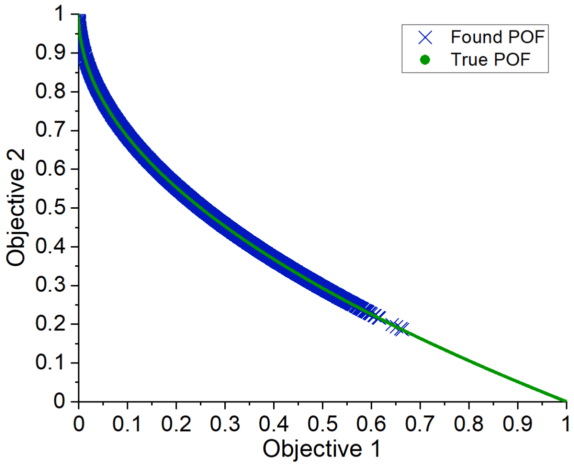
The contrast in behaviour between MLS2 and MLS2R on ZDT1 shows the importance of choosing appropriate fitness function for each level. It is caused by the nature of ZDT 1-4 functions, where first fitness function is simple and is based on a single variable only; and most of the complexity comes from the second objective. This statement is supported by the fact that performance of both MLS2 and MLS2R variants is similar on the ZDT6 function, where both fitness functions have approximately the same level of complexity, but with slight



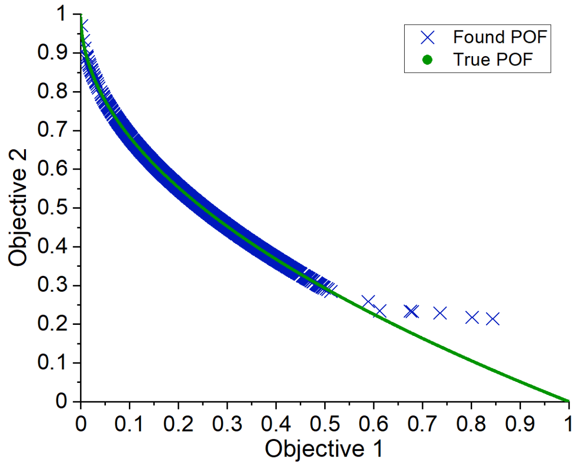
(A) MLS1



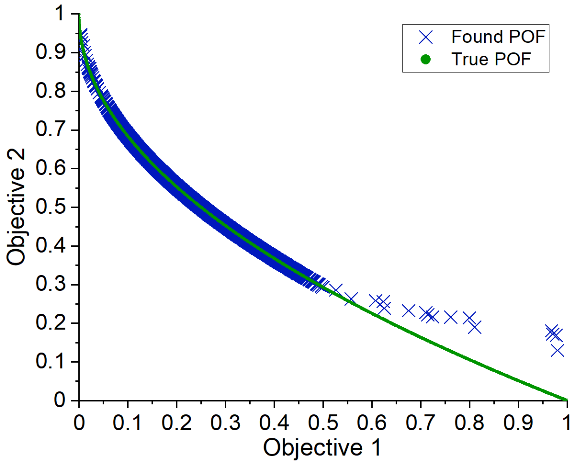
(B) MLS2



(C) MLS2R

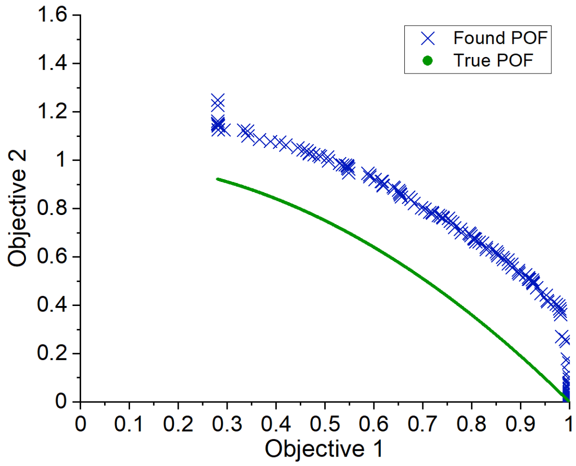


(D) Single population GA

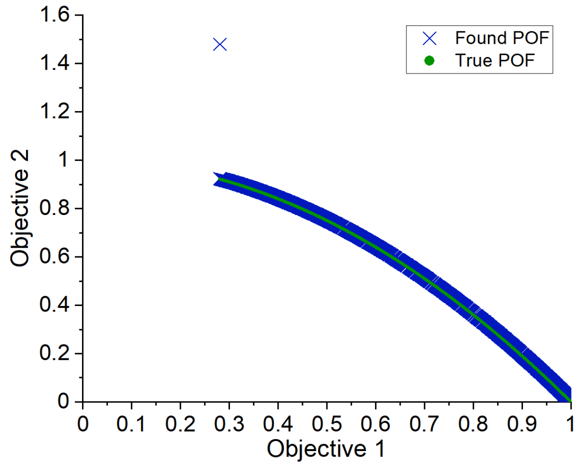


(E) Sub-population GA

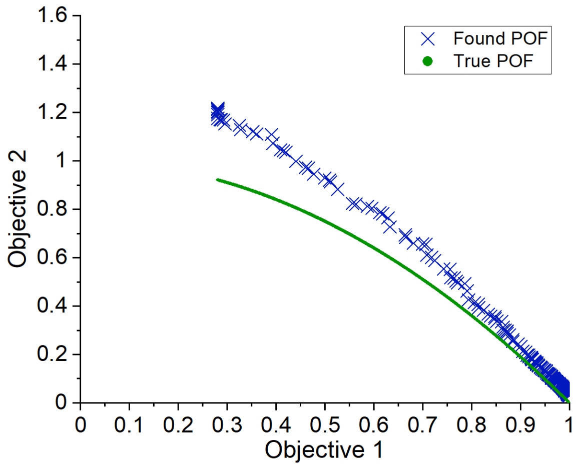
FIGURE 3.1: Comparison of the Pareto optimal fronts achieved by different MLS and simple GA variants on ZDT1 problem.



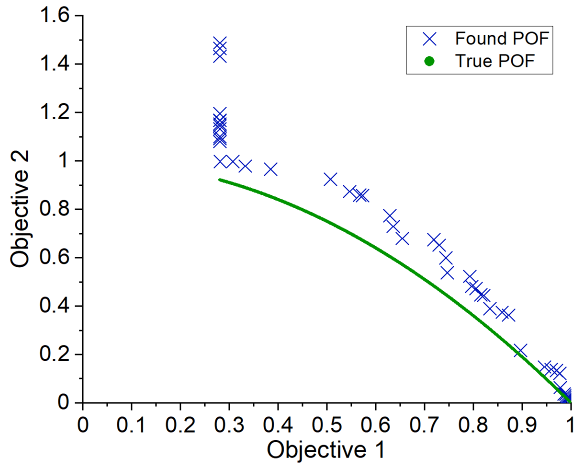
(A) MLS1



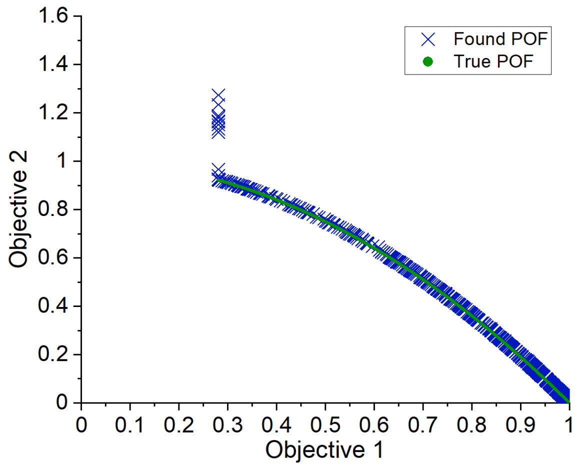
(B) MLS2



(C) MLS2R

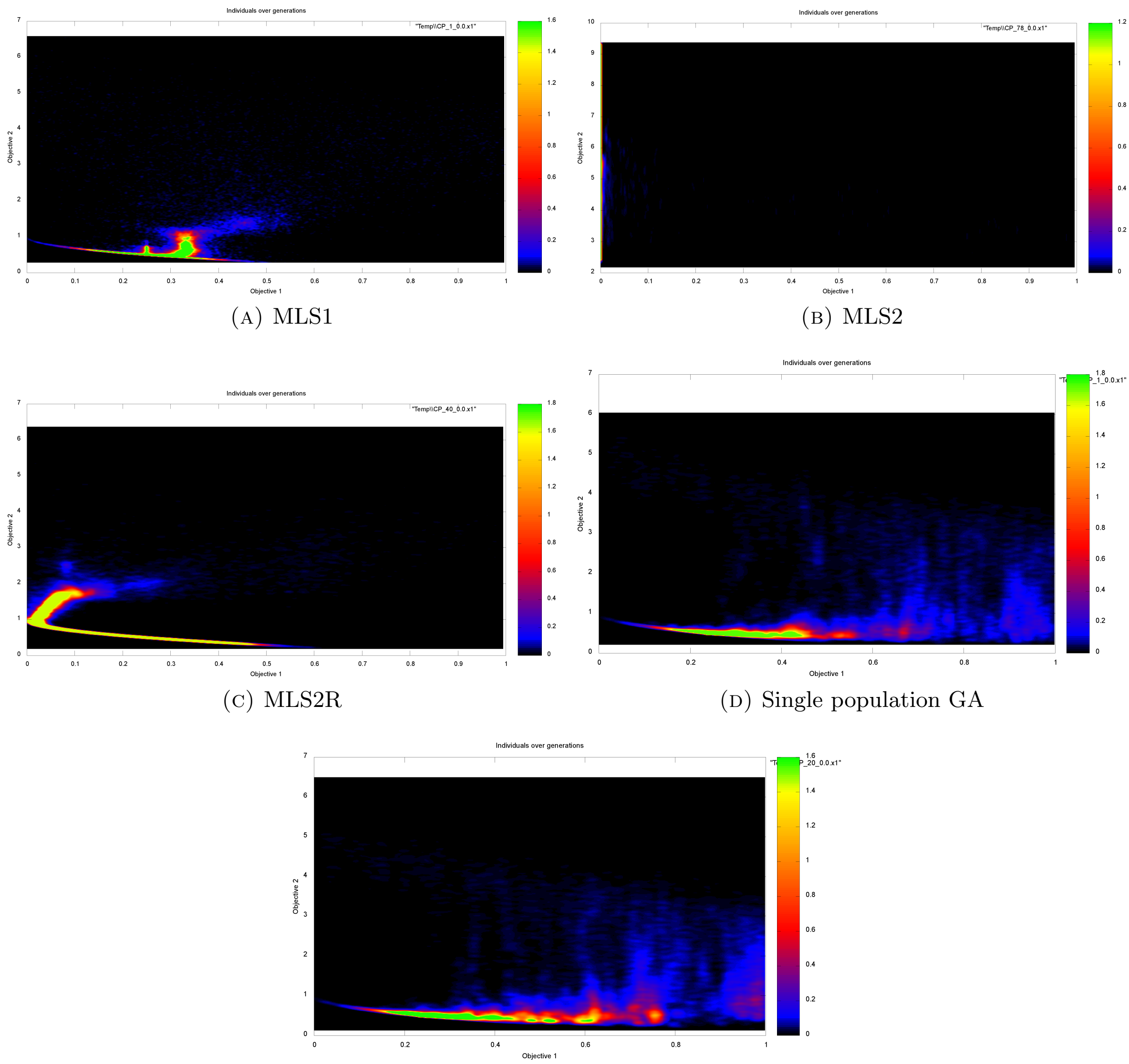


(D) Single population GA



(E) Sub-population GA

FIGURE 3.2: Comparison of the Pareto optimal fronts achieved by different MLS and simple GA variants on ZDT6 problem.



(A) MLS1

(B) MLS2

(C) MLS2R

(D) Single population GA

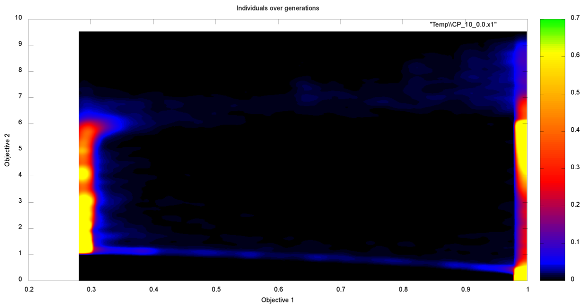
(E) Sub-population GA

FIGURE 3.3: The heat maps of different MLS and simple GA variants on ZDT1 problem.

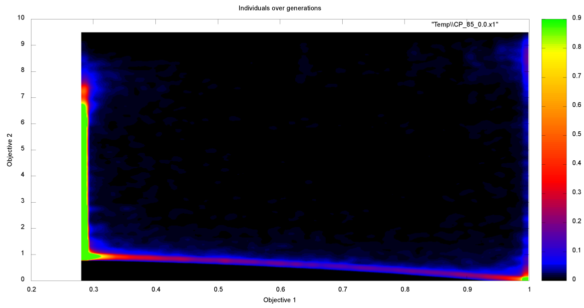
bias towards the first objective, which could explain the better performance of MLS2 and a lowered convergence of MLS1.

3.3.2 Constrained and unconstrained functions with complex Pareto optimal sets

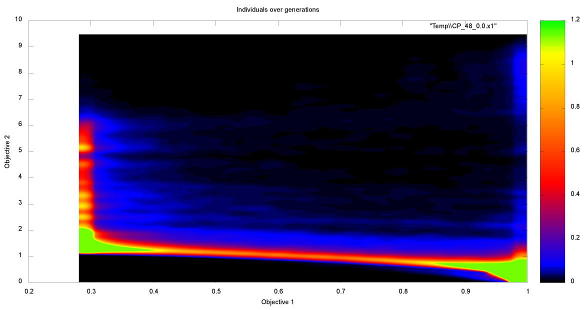
In order to investigate in more details, the differences in behaviour between different MLS variants, the tests are repeated on a more complex unconstrained and constrained problems, UF and CF, respectively. Similarly, to the previous case the preferred areas of search are illustrated in the form of obtained Pareto optimal fronts and the heat maps, which are compared to the results obtained by simple GAs. In this case only CF2 and UF3 problems are shown and discussed, as the cases with most divergent behaviour between MLS variants.



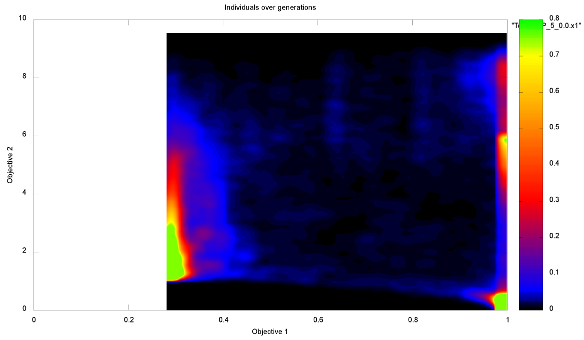
(A) MLS1



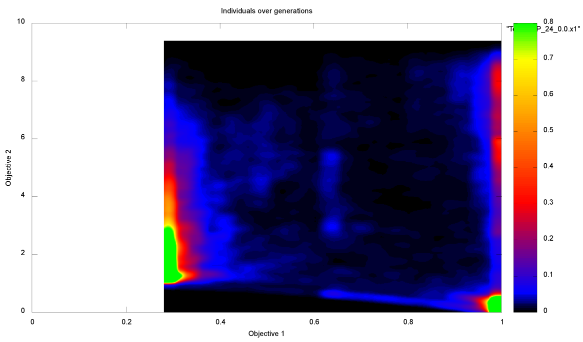
(B) MLS2



(C) MLS2R



(D) Single population GA



(E) Sub-population GA

FIGURE 3.4: The heat maps of different MLS and simple GA variants on ZDT6 problem.

For CF2 test instance, the obtained fronts are shown in Fig. 3.5, and the heat maps are illustrated in Fig. 3.6. The corresponding graphs for UF3 problem are shown in Fig. 3.7 and in Fig. 3.8 respectively.

For CF2 problem it can be observed that all variants are able to converge on the final Pareto optimal front, but neither of them is able to provide a high diversity of the final solutions. However, it can be seen that each MLS variant concentrate the search in different regions of the objective space. MLS2 shifts toward the right side of the true Pareto optimal front, around 0.9 value of the first objective; MLS2R focuses on the left, around value of 0.1; while MLS1 focuses on the area "in between", around value of 0.2 in this case. Similarly, to the ZDT cases, single population simple GA exhibit a similar search pattern to the MLS1, but with more dispersed search. This indicates that implementation of the collective-level

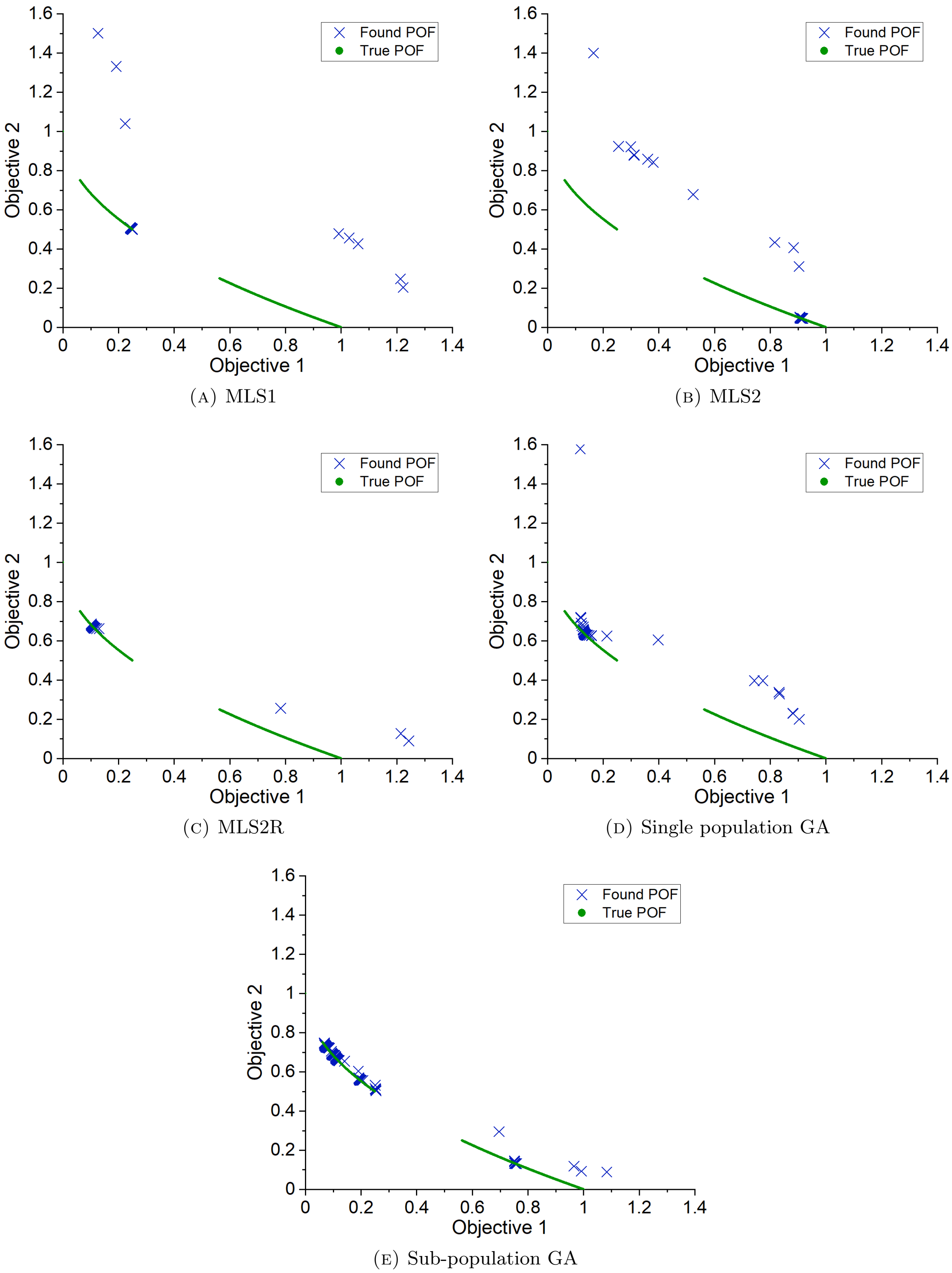
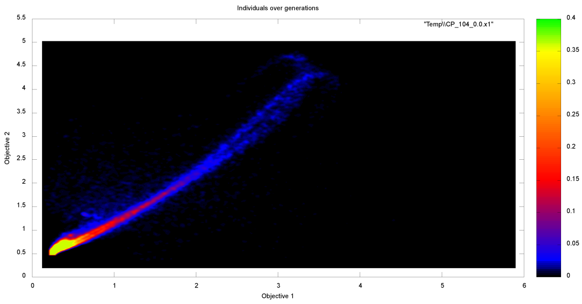
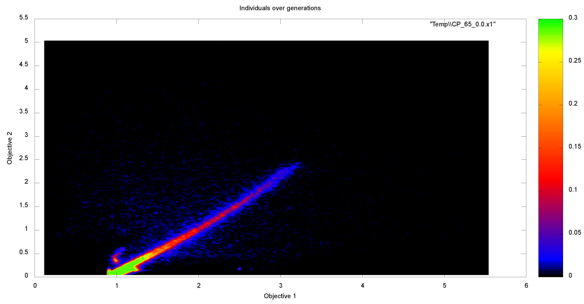


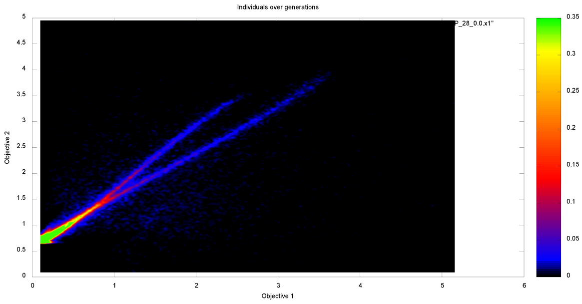
FIGURE 3.5: Comparison of the Pareto optimal fronts achieved by different MLS and simple GA variants on CF2 problem.



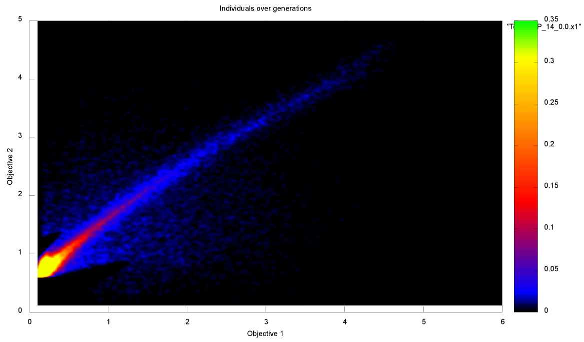
(A) MLS1



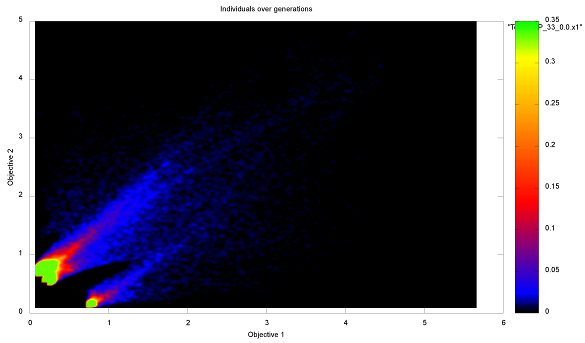
(B) MLS2



(C) MLS2R



(D) Single population GA

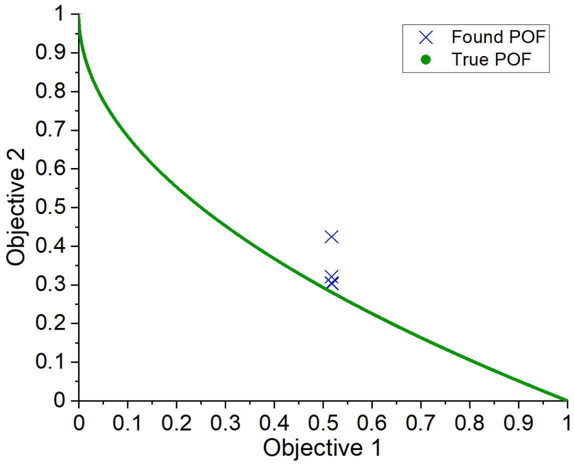


(E) Sub-population GA

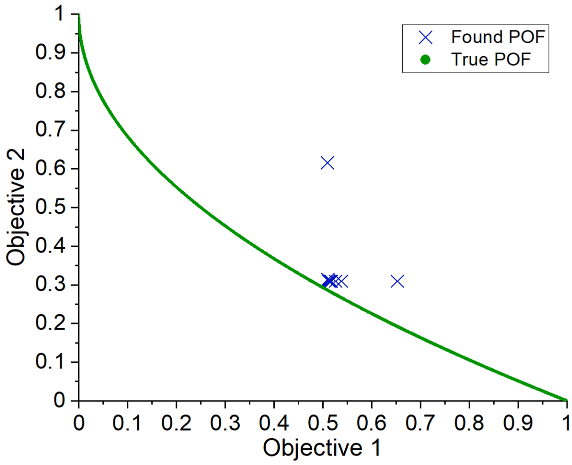
FIGURE 3.6: The heat maps of different MLS and simple GA variants on CF2 problem.

selection with fitness separation can potentially direct the search towards different regions of the objective space.

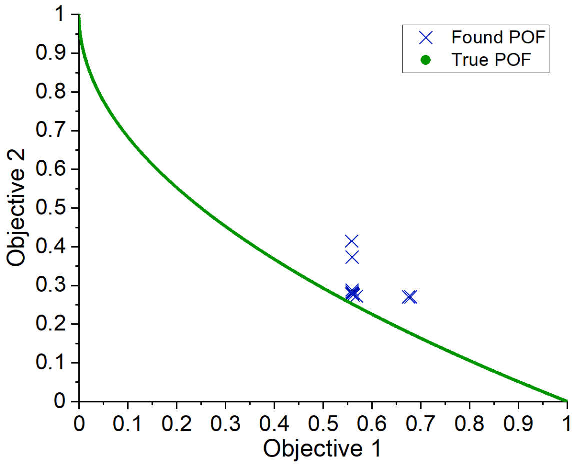
However, this behaviour is not observed for all of the tested problems. One example is UF3. From the presented graphs, with the Pareto optimal fronts shown in Fig. 3.7 and the heat maps in Fig. 3.8, it can be seen that all MLS variants and the simple GAs focus on the same, central, area. A similar principle can be observed for UF6, CF5 and CF7, with corresponding graphs provided in supplementary data. However, in all those cases a poor convergence and a low diversity of the final points is observed. All of these problems are deceptive and have a similar definition of the objectives. Therefore, lack of difference between MLS variants is most likely to be caused by the difficulty of these problems and implementation of ineffective individual-level mechanisms in the current version of MLSGA. Simple search-strategies are more likely to converge on an "easy-to-find" deceptive, local optimum rather than the global



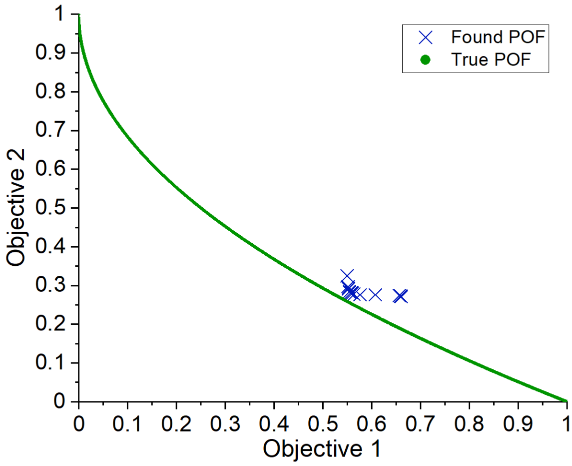
(A) MLS1



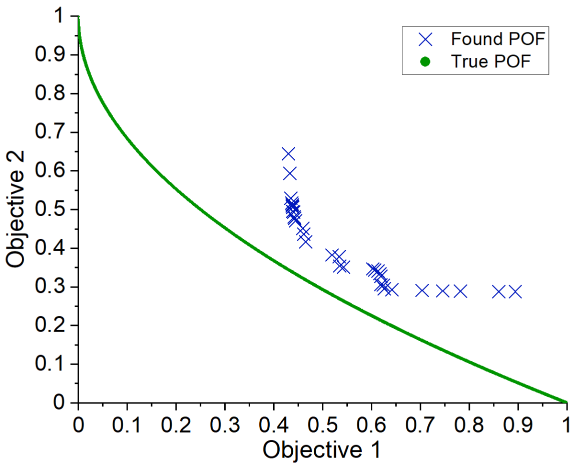
(B) MLS2



(C) MLS2R

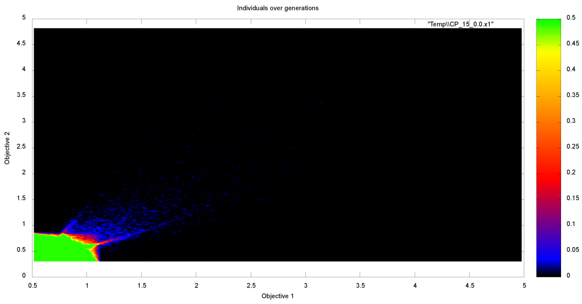


(D) Single population GA

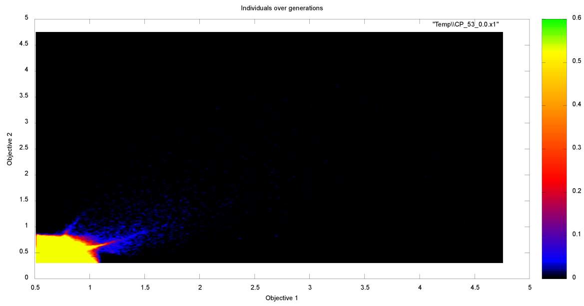


(E) Sub-population GA

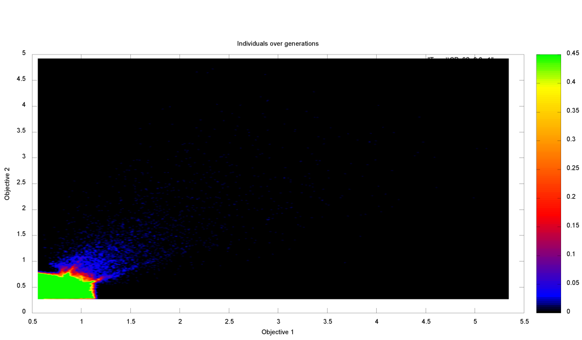
FIGURE 3.7: Comparison of the Pareto optimal fronts achieved by different MLS and simple GA variants on UF3 problem.



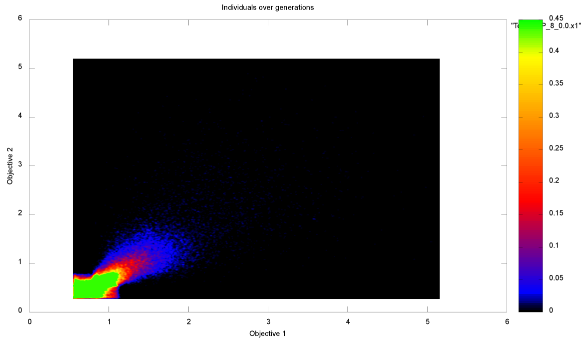
(A) MLS1



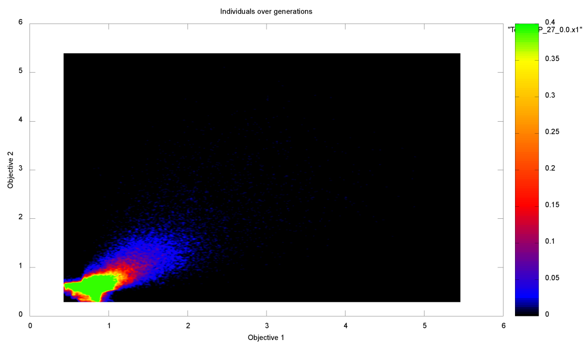
(B) MLS2



(C) MLS2R



(D) Single population GA



(E) Sub-population GA

FIGURE 3.8: The heat maps of different MLS and simple GA variants on UF3 problem.

one [109]. However, as this behaviour is seen only on 4 out of 14 tested functions, simulations with stronger individual-level mechanisms have to be conducted in order to validate if this unusual behaviour is due to limitations of those mechanisms or the MLSGA methodology in overall.

Interestingly, in all tested CF and UF cases the sub-population variant of a simple GA was able to achieve a higher diversity of the final results than other tested algorithms. It is caused by utilisation of sub-population approach without information sharing between the groups. In MLS approach, the eliminated collectives are re-populated by the remaining groups, and thus all groups are likely to reach the same local optimum. Whereas in sub-population simple GA, all groups are fully independent, therefore those are more likely to diversify.

3.4 Combining the MLS strategies

United Multi-Level Selection (MLS-U) variant is introduced to determine if the combination of MLS mechanisms can be used to increase the diversity of the simple genetic algorithm. In MLS-U all three MLS variants are used in parallel where some collectives are being subject to MLS1 type fitness definitions; others to MLS2 type; and remaining groups to MLS2R. The following modification to the overall MLSGA methodology, presented previously in Section 3.2.1, is introduced:

- Add **Step 2.2)** stating: Assign the fitness definitions from types {MLS1, MLS2, MLS2R} to each collective in the following order: $MLS1 \rightarrow MLS2 \rightarrow MLS2R \rightarrow MLS1 \rightarrow$

All hyper-parameters, GA mechanisms, constraint handling and complexity remains the same. The only exception is frequency of the collective elimination, which is decreased to 1 every 5 generation, in order to allow higher differentiation of collectives. Importantly, in conducted pre-benchmarks this change resulted in lowered performance of previously tested MLSGA variants but is preferred with the new variant.

The resulting methodology is benchmarked on the same functions; ZDT1-6, UF1-7 and CF1-7. The performance of MLS-U is compared to the MLS1, MLS2 and MLS2R variants and is presented in Table 3.4 according to HV metric and in Table 3.3 for the IGD values.

From the presented results, MLS-U shows significantly better performance than other variants on most of the tested functions, 14 out of 19 according to the IGD metric and 15 out of 19 for the HV. It can be concluded that combination of different variants has led to an increase in the diversity of obtained points and the convergence toward more regions of true Pareto optimal front on most of the tested cases. Furthermore, the MLS-U is never outperformed by all of the other variants. However, as for most of the tested problems the values of both indicators are far from ideal, it can be predicted that MLS-U was not been able to achieve the true Pareto optimal front on those cases.

The impact on the performance by MLS-U is further verified via comparison of the Pareto optimal fronts achieved by each MLS variant. This is followed by examination of the heat maps generated by each of them in order to better understand the behaviour of MLS-U. The corresponding Pareto optimal fronts are illustrated in Fig. 3.9 for UF2 and in Fig. 3.10 for UF4. Those functions are chosen as representatives of a divergent impact of MLS-U variant on the final performance.

For CF2 it can be seen that, similarly to other MLSGA variants, MLS-U fails to discover all areas of the true Pareto optimal front. It can be seen that MLS-U is mostly covering the regions where at least one of other types, MLS1, MLS2 or MLS2R, is achieving a good convergence; but is unable to discover any new areas where all variants are under-performing.

TABLE 3.3: The comparison of MLS-U performance to the other MLS variants according to average IGD results

Problem	**Variant**			
	MLS1	**MLS2**	**MLS2R**	**MLS-U**
ZDT1	1.17E-01	9.93E-01	1.55E-01	3.17E-02 [+]
ZDT2	7.41E-04	1.89E+0	4.19E-04	9.60E-02 [+][-]
ZDT3	1.15E-01	8.94E-01	4.23E-01	7.76E-03 [+]
ZDT4	7.61E-02	2.81E+01	5.32E-02	2.76E-02 [+]
ZDT6	1.63E-01	3.03E-03	7.58E-02	1.74E-02 [+][-]
UF1	1.87E-01	2.75E-01	1.34E-01	9.58E-02 [+]
UF2	1.36E-01	1.32E-01	8.12E-02	6.03E-02 [+]
UF3	2.95E-01	2.78E-01	2.69E-01	1.47E-01 [+]
UF4	9.30E-02	1.14E-01	1.53E-01	1.48E-01 [-]
UF5	5.11E-01	8.41E-01	8.96E-01	7.70E-01 [+][-]
UF6	3.49E-01	4.64E-01	3.15E-01	2.12E-01 [+]
UF7	4.34E-01	3.86E-01	5.18E-01	2.17E-01 [+]
CF1	9.28E-02	9.35E-02	9.38E-02	7.80E-02 [+]
CF2	2.08E-01	2.83E-01	1.77E-01	9.77E-02 [+]
CF3	1.86E-01	6.50E-01	2.56E-01	1.90E-01 [+]
CF4	3.05E-01	3.57E-01	2.08E-01	1.04E-01 [+]
CF5	4.46E-01	4.63E-01	3.21E-01	2.03E-01 [+]
CF6	2.48E-01	2.38E-01	2.34E-01	8.10E-02 [+]
CF7	3.43E-01	4.39E-01	3.96E-01	1.93E-01 [+]

[+] and [-] indicates if the results of MLS-U are significantly better or worse than the next worse/better variant, respectively. The results of MLS-U are highlighted: in green if it is best variant; in orange if it is outperformed by only one of other variants; and in red if it is exhibiting the worst performance

In the case of CF2 function, neither variant is able to discover the left part of the right Pareto optimal front, range of 0.5-0.8 according to the first objective. It indicates that MLS-U variant may be able to successfully combine the advantages of each type of fitness separation, but is unable to reach beyond their scope, resulting in exploration of more regions simultaneously, and thus a significant increase in diversity, rather than flat improvement in performance or convergence. This was expected, as MLS-U do not provide any new diversity preservation mechanisms but is a simple combination of other developed variants.

Interestingly, on some cases the MLS-U is showing a lower quality of solutions than at least one other MLSGA variant, according to Tables 3.4 and 3.3. In the case of UF4, Fig. 3.10 MLS-U is able to discover more points in extreme parts of the Pareto optimal front, near points (0,1) and (1,0), but with a lowered convergence in other regions. MLS-U combines different variants; therefore, it is logical that reducing the quantity of "good" collectives, MLS1 and MLS2 in this case, will result in worse final solutions, especially if the performance of different "modules" is highly contrasting. Similar observation was made for the other GAs that combines multiple divergent methodologies such as BCE [103] or HEIA [36].

TABLE 3.4: The comparison of MLS-U performance to the other MLS variants according to average HV results

Problem	Variant			
	MLS1	MLS2	MLS2R	MLS-U
ZDT1	0.8302	0.3774	0.8224	0.9068[+]
ZDT2	0.8318	0.0004	0.8331	0.8048[+][-]
ZDT3	1.0897	0.5209	0.7100	1.1861[+]
ZDT4	0.8480	0.0000	0.8943	0.9036[+]
ZDT6	0.7005	0.7547	0.7215	0.7499[+]
UF1	0.7544	0.7125	0.7863	0.8434[+]
UF2	0.8050	0.8067	0.8272	0.8731[+]
UF3	0.6247	0.6211	0.6205	0.7176[+]
UF4	0.7827	0.7659	0.7144	0.7674[-]
UF5	0.4332	0.2906	0.2535	0.3628[+][-]
UF6	0.5550	0.4889	0.5543	0.6592[+]
UF7	0.5634	0.5675	0.5213	0.7019[+]
CF1	0.7811	0.7949	0.7781	0.8344[+]
CF2	0.7523	0.7404	0.7557	0.8430[+]
CF3	0.5518	0.4415	0.4648	0.6357[+]
CF4	0.6261	0.5706	0.6287	0.7495[+]
CF5	0.5397	0.4940	0.5241	0.6108[+]
CF6	0.7245	0.7407	0.7331	0.8298[+]
CF7	0.6198	0.6044	0.6049	0.7364[+]

[+] and [-] indicates if the results of MLS-U are significantly better or worse than the next worse/better variant, respectively. The results of MLS-U are highlighted: in green if it is best variant; in orange if it is outperformed by only one of other variants; and in red if it is exhibiting the worst performance

In order to verify if the MLS-U variant is able to concentrate the search on divergent regions of objective space, rather than initially finding different parts of the true Pareto optimal front and then concentrating the search on single one of them, like MLS1, MLS2 and MLS2R, the heat maps are evaluated to find the preferred areas of search. Those are illustrated in Fig. 3.11 for CF2 case and in Fig. 3.12 for UF4.

From the heat maps it can be concluded that MLS-U is able to uniformly spread the concentration of the search into different regions of the true Pareto optimal front. In case of CF2, Fig. 3.11, 3 regions of interest are clearly visible for the MLS-U and 2 regions in case of UF4, Fig. 3.12. Comparing to the heat maps for MLS1, MLS2 and MLS2R, it can be seen that those regions corresponds to the search patterns of other variants. This may indicate that the most left region is achieved by the MLS2R-type collectives, centre by the MLS1-type and the right side by the MLS2-type groups. From the presented data it can be concluded that MLS-U variant of the MLSGA is showing a high potential for the dispersed, region-based search, as the divergent evolutionary pressures resulting from different collective-types, MLS1, MLS2 and MLS2R, can be maintained under a single methodology.

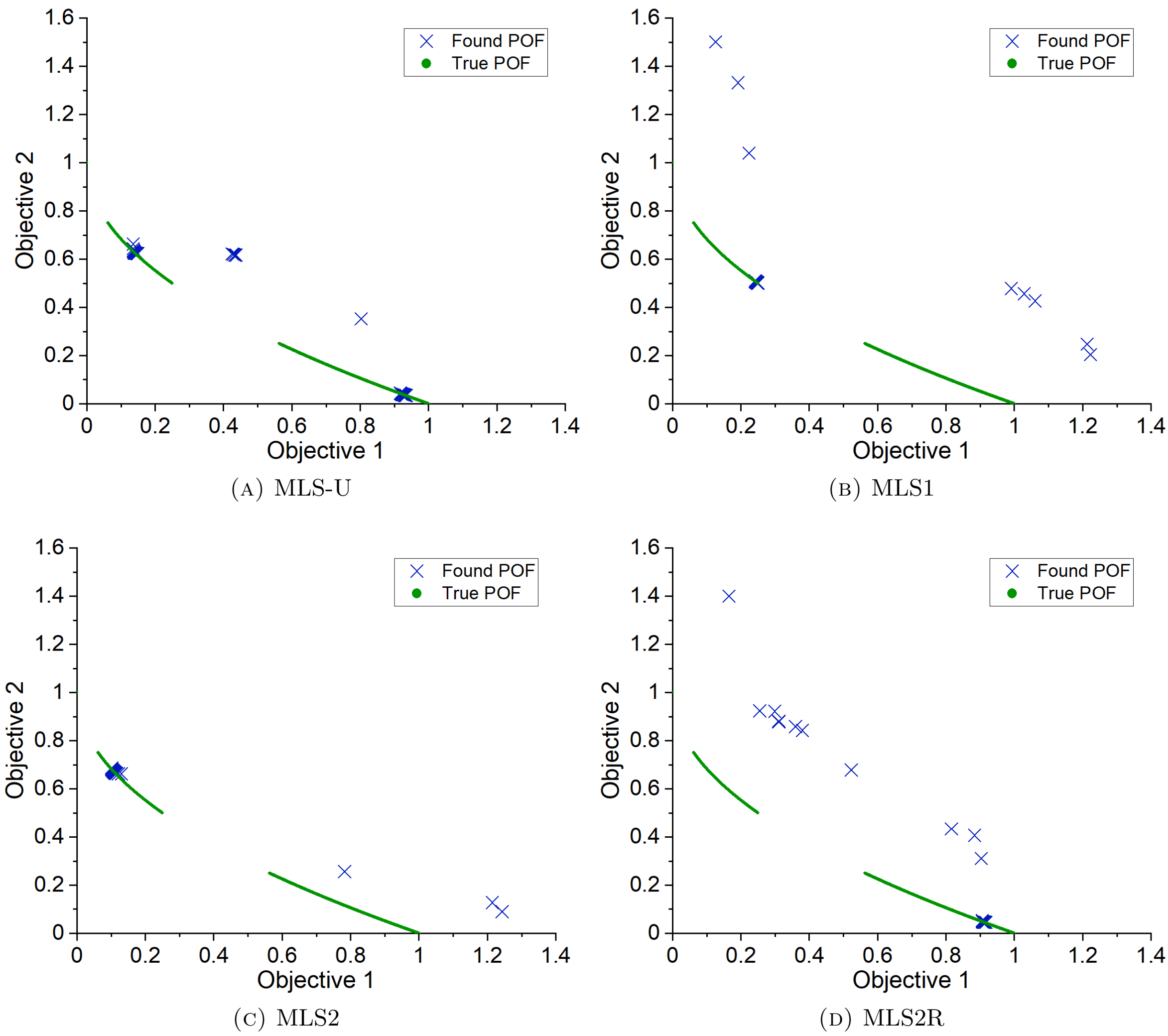
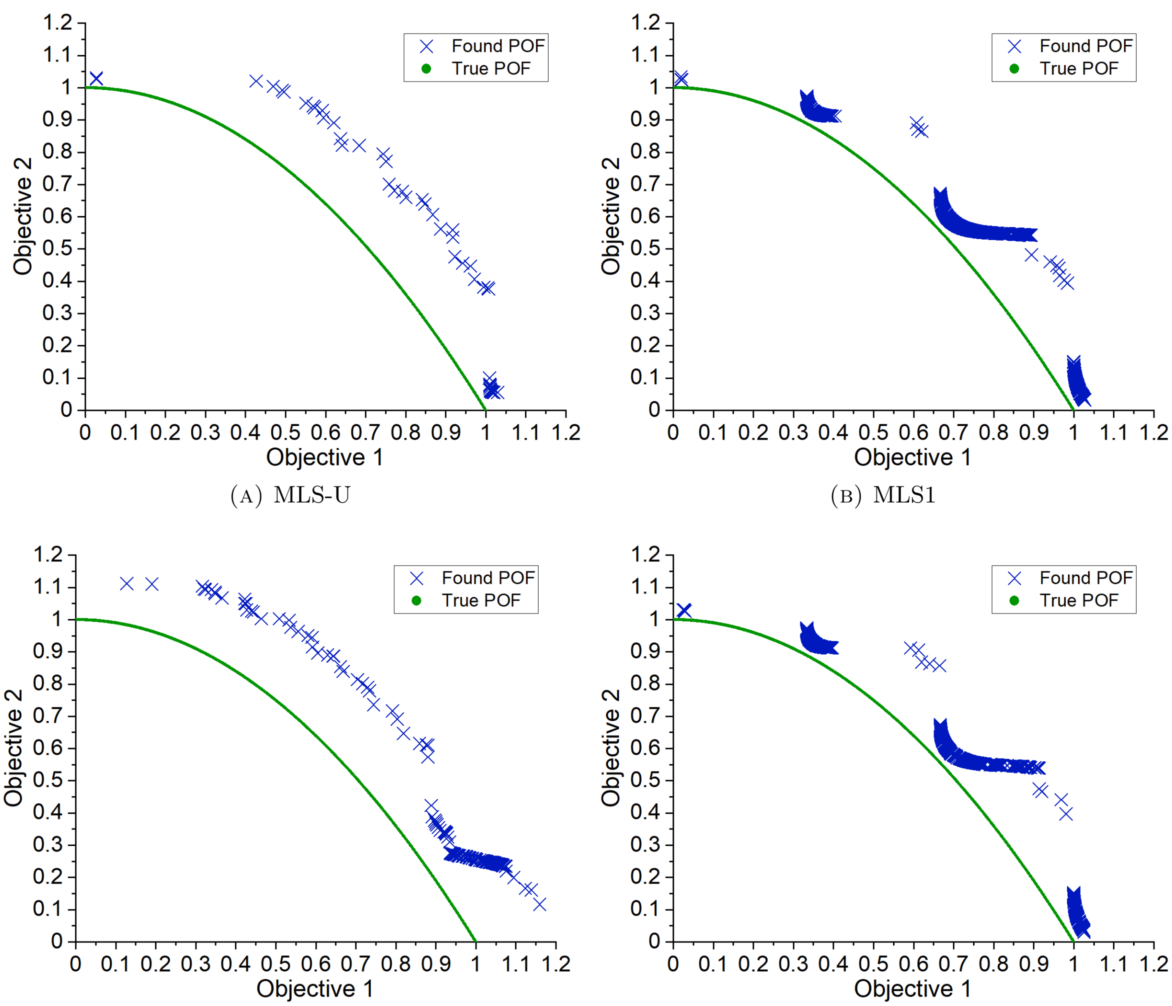


FIGURE 3.9: Comparison of the Pareto optimal fronts achieved by different MLS variants on the CF2 problem.

3.5 Understanding mechanisms of the MLSGA

MLSGA combines multiple search mechanisms, such as sub-population based approach, collective reproduction and fitness separation. It is essential to investigate the impact of each of them on the final behaviour in order to fully understand the principles of working of the MLSGA.

Firstly, MLSGA is compared to the weight-vector based simple GAs with and without sub-populations in order to investigate if MLSGA's sub-region search is similar to simple decomposition methods and if the performance improvement is obtained. In the weight-vector variant without sub-populations, each individual has weights assigned to both objective functions based on the uniformly spread linear weight-vector $V = \{v_1 = (0,1), \ldots, v_i = (\frac{i-1}{n-1}, 1 - \frac{i-1}{n-1}), \ldots, v_n = (1,0)\}$ where n is the number of individuals. In the sub-population version three types of sub-groups are introduced, with following weight-vectors $\{v_1 = (0.001, 0.999), v_2 =$



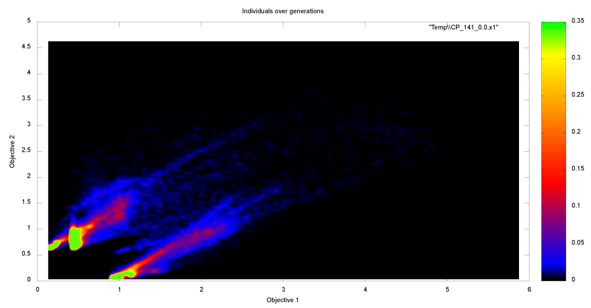
(A) MLS-U (B) MLS1 (C) MLS2 (D) MLS2R

FIGURE 3.10: Comparison of the Pareto optimal fronts achieved by different MLS variants on the UF4 problem.

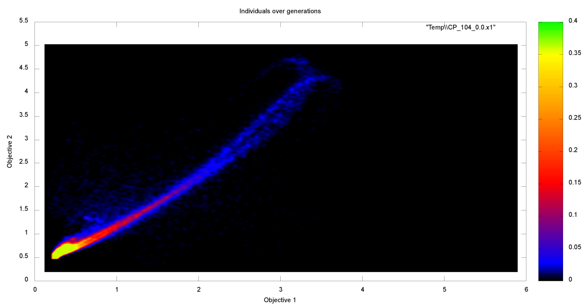
$(0.5, 0.5), v_3 = (0.999, 0.001)\}$. The types assignment to each group follows the same rules as in MLSGA, but sub-populations are not allowed to exchange any information regarding their objectives or variables during the optimisation process. Therefore, every individual in each group has the same weight-vector and mating is restricted to among the individuals of a particular group.

The proposed GA methodologies are simulated, and the average results are compared with the performance of MLS-U variant of MLSGA. Those are presented in Table 3.5 for the IGD indicator and in Table 3.6 for HV.

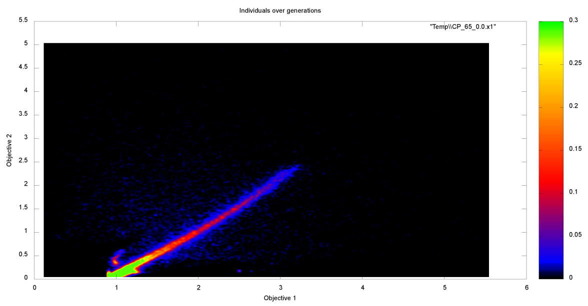
From the presented data it can be seen that MLS-U variant of MLSGA has the best overall performance on most of the tested problems, 12 out of 19 cases for the IGD; and 15 out of 19 cases for the HV; while the single population variant of weight-vector GA always comes last. This is showing the superiority of proposed solution, especially regarding the diversity of achieved solutions due to significantly better HV metrics. Furthermore, in most cases



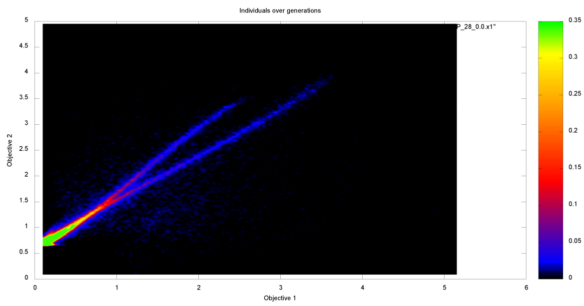
(A) MLS-U



(B) MLS1

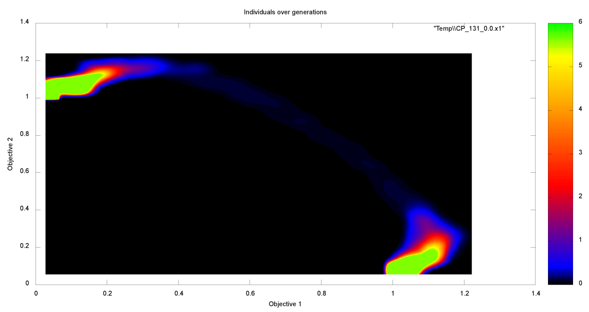


(C) MLS2

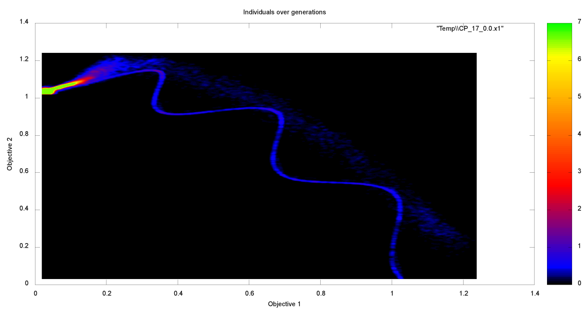


(D) MLS2R

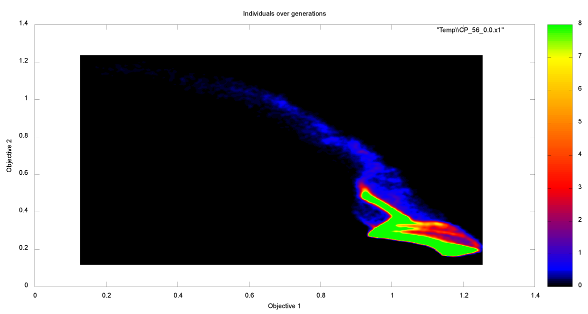
FIGURE 3.11: The heat maps of different MLS variants on the CF2 problem



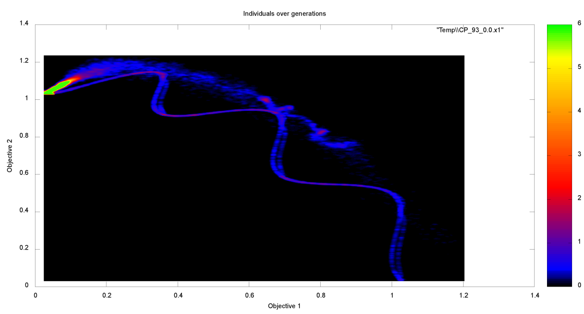
(A) MLS-U



(B) MLS1



(C) MLS2



(D) MLS2R

FIGURE 3.12: The heat maps of different MLS variants on the UF4 problem.

where MLSGA is outperformed by other variants, there is no statistical significance between results, such as UF3, UF4, CF4 and CF5 cases for the IGD indicator, Table 3.5, and UF3 and CF2 for the HV metric, Table 3.6.

The better performance of the sub-population variant over the single-population one, indicates that splitting the population by itself leads to a better performance. Similar principle has been observed for non-weight-vector variants of simple GA, as discussed in Section 3.3 and in the literature [46]. However, as MLS-U is better than both sub-population approaches, this indicates that split in population is not the main reason for the increased performance and diversity of MLSGA. The low performance of single-population weight-vector method, shows that additional mechanisms, such as neighbourhoods of similar weights, should be applied in order to effectively find the best solutions for each weight, as demonstrated by the MOEA/D algorithm [35].

Applying weights to each sub-population separately is shown to yield better results than the single population approach, leading to a comparable performance with MLSGA on some cases. This phenomenon is investigated further, by comparison of the Pareto optimal fronts achieved by MLSGA and the weight-vector GA with sub-populations. These are illustrated in Fig. 3.13 for UF2 and in Fig. 3.14 for CF2. UF2 and CF2 are selected as the representatives for better and worse performance of MLS-U in comparison to the sub-population weight-vector GA, respectively.

On CF2 problem, Fig. 3.14 the weight-vector approach finds more regions of the true Pareto

Table 3.5: The performance of MLS-U compared to the weight-vector variants of simple GA, according to the average IGD metric

Problem	MLS-U	Weight-vector GA	Sub-population weight-vector GA
ZDT1	3.17E-02 [+]	1.43E-01	7.18E-02 [+]
ZDT2	9.60E-02 [+]	6.09E-01	6.09E-01
ZDT3	7.76E-03 [+]	1.21E-01	2.06E-02 [+]
ZDT4	2.76E-02 [+]	1.15E-01	7.07E-02 [+]
ZDT6	1.74E-02 [+]	1.41E-01	3.60E-02 [+]
UF1	9.58E-02	2.50E-01	1.09E-01 [+]
UF2	6.03E-02 [+]	1.21E-01	7.43E-02 [+]
UF3	1.47E-01 [+]	2.77E-01	1.42E-01
UF4	1.48E-01	1.60E-01	1.34E-01
UF5	7.70E-01	8.88E-01[+]	7.63E-01
UF6	2.12E-01 [+]	3.64E-01	1.47E-01 [+]
UF7	2.17E-01 [+]	4.29E-01	3.27E-01 [+]
CF1	7.80E-02 [+]	9.19E-02	8.99E-02
CF2	9.77E-02	2.74E-01	7.38E-02 [+]
CF3	1.90E-01	4.51E-01	2.15E-01
CF4	1.04E-01 [+]	2.29E-01	9.64E-02
CF5	2.03E-01 [+]	3.62E-01	1.92E-01
CF6	8.10E-02 [+]	2.55E-01	1.70E-01 [+]
CF7	1.93E-01 [+]	3.16E-01	9.87E-02 [+]

The best algorithm for each problem is highlighted in green, the second best is highlighted in orange. [+] indicates if the results are significantly better than the next worse variant.

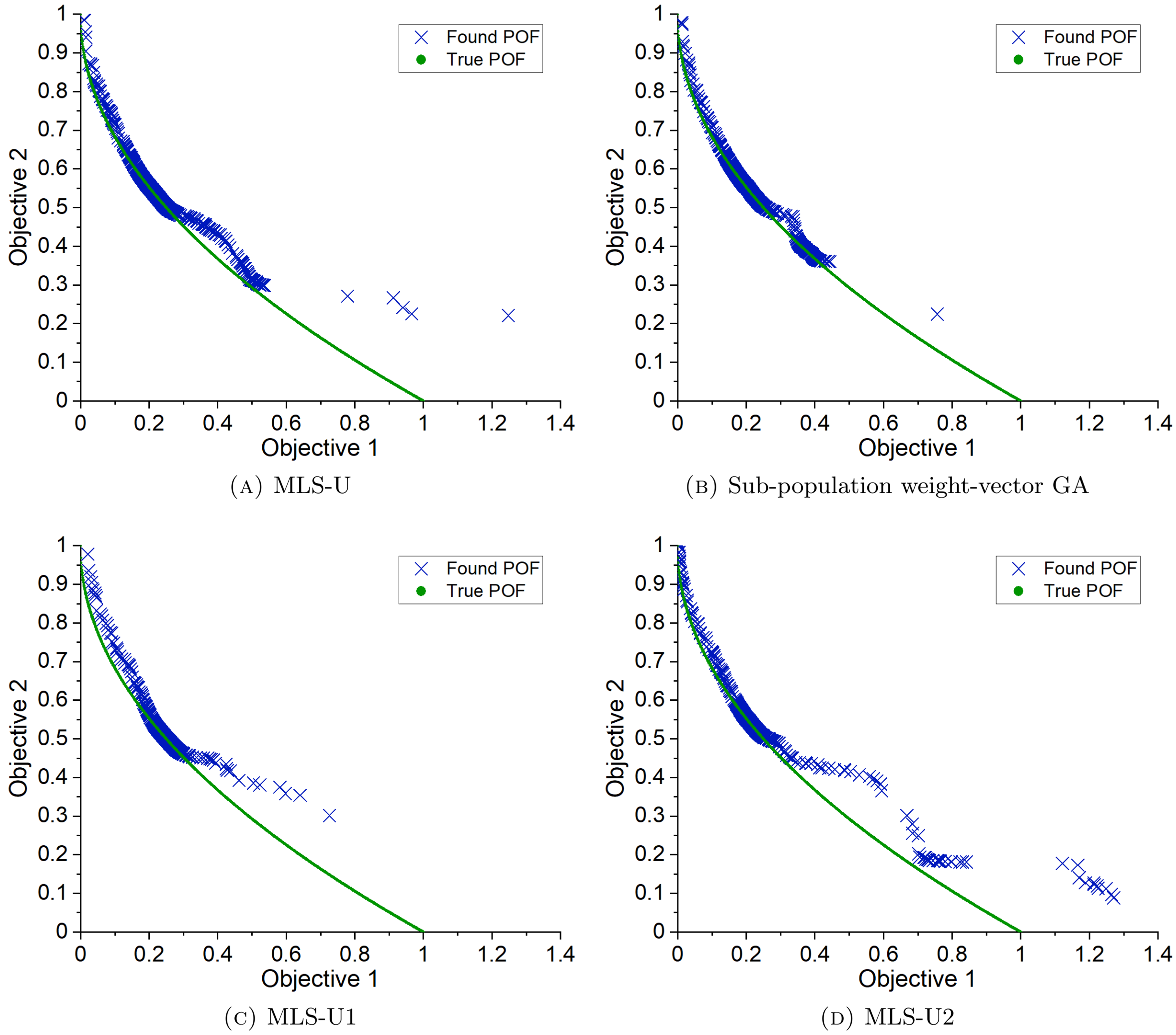
TABLE 3.6: The performance of MLS-U compared to the weight-vector variants of simple GA, according to the average HV metric

Problem	MLS-U	Weight-vector GA	Sub-population weight-vector GA
ZDT1	0.9068 [+]	0.8246	0.8466 [+]
ZDT2	0.8048 [+]	0.5000	0.5000
ZDT3	1.1861	1.0476	1.1812 [+]
ZDT4	0.9036 [+]	0.8363	0.8529
ZDT6	0.7499 [+]	0.7075	0.7434 [+]
UF1	0.8434	0.7192	0.8375 [+]
UF2	0.8731 [+]	0.8123	0.8511[+]
UF3	0.7176 [+]	0.6210	0.7224
UF4	0.7676	0.7318	0.7666 [+]
UF5	0.3628 [+]	0.2420	0.3379 [+]
UF6	0.6592 [+]	0.5335	0.6891 [+]
UF7	0.7019 [+]	0.5733	0.6450 [+]
CF1	0.8344 [+]	0.7833	0.8099
CF2	0.8430 [+]	0.7329	0.8526
CF3	0.6357 [+]	0.4369	0.6231 [+]
CF4	0.7495	0.6309	0.7455 [+]
CF5	0.6108	0.5410	0.6042 [+]
CF6	0.8298[+]	0.7301	0.8029 [+]
CF7	0.7364[+]	0.6574	0.8122 [+]

The best algorithm for each problem is highlighted in green, the second best is highlighted in orange. [+] indicates if the results are significantly better than the next worse variant.

optimal front in comparison to MLSGA, localised on the "left part" of the front. The potential explanation for this phenomenon is that in developed weight-vector variant the sub-populations are fully independent. Therefore, the groups may freely explore different regions of the objective space even if they share the same weights, resulting in more optima found on highly multimodal problems, such as CF2. Whereas in MLSGA the collectives of the same type strongly promote similar individuals due to utilised collective-reproduction mechanisms and thus are more likely to converge on the same optimum as discussed previously.

However, comparing the UF2 case, Fig. 3.13, it can be observed that MLSGA is able to retain a higher diversity on more continuous problems. The potential explanation for this phenomenon is that in the weight-vector approach each individual in a sub-population will try to converge on the same point, as solutions are sharing the same weights. In MLSGA, the collectives can retain a higher diversity of solutions, as long as the average of them is following the values promoted by the collective-level reproduction mechanisms. Therefore, in MLSGA the diversity of each group is not penalised as strongly as in the weight-vector approach. Furthermore, in the collectives of MLS2 and MLS2R type the diversity is additionally encouraged as both levels of selection have different evolutionary goals. It could be suggested that the weight-vector based method may include more than 3 types of groups, which will lead to a higher diversification of sub-populations. However, the same principle can be applied to the MLSGA, as potentially unlimited number of divergent fitness definitions can be used on each level of selection. Furthermore, each additional weight presumes that



(A) MLS-U

(B) Sub-population weight-vector GA

(C) MLS-U1

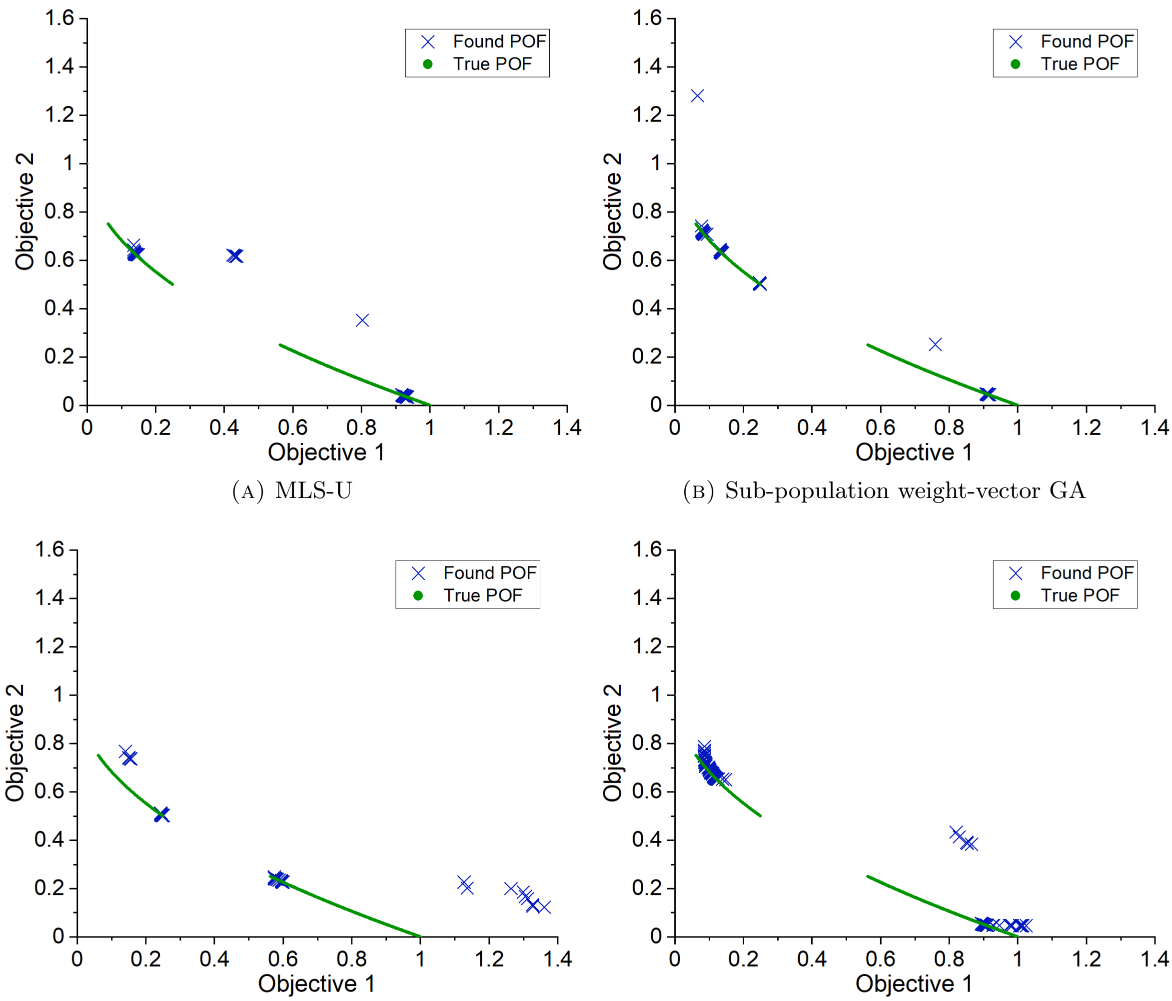
(D) MLS-U2

FIGURE 3.13: Comparison of the Pareto optimal fronts achieved by different MLS-U variants and the sub-population weight-vector GA on the UF2 problem

there are feasible solutions in the regions promoted by that weight. That approach usually requires the extensive knowledge about optimised problem, which is highly problematic for the practical applications as discussed in Chapter 2. This further shows the usefulness of MLS-based methodology.

Secondly, in order to investigate the impact of the fitness separation on each level, the additional variants of MLS-U where the separation is performed only on a single level are introduced. In MLS-U1 variant the influence of divergent individual-level definitions on the region-based search is examined, while for MLS-U2 it is the importance of the collective-level. Corresponding fitness function definitions for each level of selection are presented in Table 3.7. Similarly, to MLS-U, only three types of collectives are introduced, and the same hyper-parameters are used for all simulations.

The Pareto optimal fronts obtained by both variants are compared to the results achieved by MLS-U. These are illustrated for UF2 function in Fig. 3.13 and in Fig. 3.14 for CF2 problem.



(A) MLS-U

(B) Sub-population weight-vector GA

(C) MLS-U1

(D) MLS-U2

FIGURE 3.14: Comparison of the Pareto optimal fronts achieved by different MLS-U variants and the sub-population weight-vector GA on the CF2 problem

TABLE 3.7: Fitness functions used on each level of selection, depending on the collective type, for MLS-U1 and MLS-U2 variants

Type of collective	**1**	**2**	**3**
		MLS-U1	
Individual-level	$\frac{f_1(x)+f_2(x)}{2}$	$f_1(x)$	$f_2(x)$
Collective-level		$\frac{f_1(x)+f_2(x)}{2}$	
		MLS-U2	
Individual-level		$\frac{f_1(x)+f_2(x)}{2}$	
Collective-level	$\frac{f_1(x)+f_2(x)}{2}$	$f_1(x)$	$f_2(x)$

For both cases it can be seen that for MLS-U1 variant the final Pareto optimal front is narrowed to the central regions of the objective space in comparison to the original MLS-U. It indicates that limiting the fitness separation to the individual-level, strongly narrows the final diversity. Therefore, even if the individual reproduction is promoting the extreme regions of the objective space, by using a single objective, the collective-level mechanisms will quickly eliminate the groups that derive from the preferred "average" values. The opposite can be observed for MLS-U2. In the MLS-U2 variant a stronger shift towards "left" and "right" side of the true Pareto optimal front can be observed in comparison to the MLS-U. It indicates that the evolutionary pressure is directed towards the regions with more extreme values of both objectives. Therefore, by utilisation of an average of both objectives for the individual reproduction, the individual-level search is in weaker opposition to the evolutionary pressure developed by the collective-level. This proves that the region-based search of MLSGA is achieved by implementing different fitness definitions on the collective-level. However, the individual-level can be adjusted to promote the extreme regions of the objective space, as in MLS-U2, or more central regions, as in MLS-U, depending on utilised definitions of the fitness functions.

3.6 Multi-level selection theory applied to a genetic algorithm

In this chapter the potential of the fitness separation and collective-level mechanisms for a diversity promoting methodology have been investigated. For that purpose, three divergent MLS types with different fitness definitions on each level of selection are proposed: MLS1, MLS2 and MLS2R. The results show that each of proposed variants is focusing its search on a particular region of the objective space, MLS1 on the centre, while MLS2 and MLS2R on the right and left side, respectively. Therefore, it is proven that the additional evolutionary pressure developed by MLSGA can be successfully targeted into various regions of the objective space.

In the next step the MLS-U variant of MLSGA is proposed in order to combine advantages of each MLS type. Presented results show that the distinct search strategy of each MLS type is maintained under the combined variant, leading to a more dispersed search and thus increase in diversity. It is concluded that this behaviour is due to distinct types of collectives strongly promoting certain values of the objectives in its members, and thus limiting free exploration in other directions. Therefore, MLS1-type collectives push for the average values of both objectives, the central region of Pareto optimal front; MLS2 collectives focuses on minimising the second objective and thus right-hand-side areas; and the groups with MLS2R fitness definitions are promoting the first objective and the left region respectively. This indicates a high potential of MLSGA methodology for the region-based search strategy.

The mechanisms used in MLSGA show some similarities in design to the sub-population and weight-vector approaches. However, according to the conducted benchmarks, MLSGA is outperforming simple sub-population and weight-vector approaches, especially regarding the diversity of obtained solutions. Furthermore, unlike the weight-vector approaches, MLSGA do not fully restrain the free exploration of the objective space by individuals. The collective-level mechanisms only promote certain regions of the objective space, but the individual-level search is not hard-bounded by them, and thus the individuals may retain a higher diversity inside each group.

However, on more complex problems, such as UF and CF sets, the MLSGA is unable to properly approximate the true Pareto optimal front. On most of those cases, MLSGA is able to successfully discover different regions of the Pareto optimal front, but obtained results are often limited to a narrow set of solutions or a poor convergence is observed. It is suggested that this phenomenon is due to utilisation of simple individual-level mechanisms, which are unable to provide a sufficient evolutionary pressure. Furthermore, with currently implemented mechanisms the individuals in the same collective do not have any incentive to diversify among themselves, as each of them is sharing the same goals. Therefore, the MLS mechanisms are able to increase the diversity of the search globally, on the objective space, but are unable to maintain it locally, inside of each collective.

According to that, stronger individual-level reproduction strategies with proper diversity preservation mechanisms have to be implemented into MLSGA, in order to fully evaluate the potential of this approach. This will potentially lead to the methodology, where the collective-level promotes a global diversity, through the discovery of different regions of the objective space, while the individual-level mechanisms are used to obtain a diverse front of solutions inside each group with a proper convergence. However, it is yet to be investigated if the behaviour exhibited by the current version of MLSGA will be maintained with those stronger mechanisms, as this phenomenon may be subject to currently implemented individual-level mechanisms only.

Chapter 4 Hybridising the MLSGA

In this chapter it is investigated if the diversity focus and region-based search shown by the MLSGA will be retained with stronger individual-level mechanisms. For that purpose, selected methodologies, taken from the current state-of-the-art, will be implemented into individual-level of the MLSGA. Performance of developed variants will be validated on benchmarking test sets and then compared with implemented algorithms and previous version of the MLSGA. In an ideal outcome, resulting methodology will strongly promote overall diversity by the collective-level mechanisms, while individual-level mechanisms will maintain convergence and diversity inside each sub-population.

4.1 Methodology of the MLSGA-hybrid

4.1.1 Description of mechanisms

In this study, eight different GAs are chosen as potential individual-level reproduction mechanism for the MLSGA, leading to development of eight distinct hybrid variants of the MLSGA. Following algorithms are selected from the state-of-the-art due to given reasons:

1. U-NSGA-III [43] as a general-solver GA and the combined variant of the most commonly utilised algorithm for two-objective problems: NSGA-II [34]; and an improved version of it for 3+ objective cases: NSGA-III [37];
2. MOEA/D [35] as the best GA for unconstrained and continuous problems according to the CEC'09 comparison [42];
3. MOEA/D-MSF and MOEA/D-PSF [83] as updated versions of the MOEA/D with better performance on a number of test cases.
4. MTS [93] as the best available algorithm for constrained problems and representing distributed search algorithms.
5. IBEA [121] as the most commonly utilised indicator-based GA.

6. BCE [103] and HEIA [36] as the most highly performing co-evolutionary approaches.

Resulting algorithm, denoted as MLSGA-hybrid, works as follows with the changes, in comparison to the previous version of the MLSGA, given in red text colour:

Inputs:

- **Multi-objective problem;**
- **Np: Population size;**
- **Nc: Number of collectives;**
- **Fc: Frequency of collective elimination;**
- **Fitness function definitions for each level;**
- **Individual-level specific parameters e.g. from MOEA/D;**
- **Stopping criterion;**

Output: External non-dominated Population (EP)

Step 1) Initialisation:

Step 1.1) Set $EP = NULL$.

Step 1.2) Randomly generate the initial population P of N_p individuals $\{x_j, \ldots, x_{Np}\}$.

Step 2) Classification:

Step 2.1) Classify the individuals from the initial population P into N_c collectives, $\{C_i, \ldots, C_{Nc}\}$, so that each contains a separate population $\{P_i, \ldots, P_{Nc}\}$. Classification is based on the search space.

Step 2.2) Assign fitness definitions from types {MLS1, MLS2, MLS2R} to each collective in following order $MLS1 \rightarrow MLS2 \rightarrow MLS2R \rightarrow MLS1 \ldots etc.$.

Step 2.3) Perform the individual-level assignment specific to implemented algorithm separately for each collective, as documented in corresponding literature. For example, in case of MOEA/D hybrid, assign the nearest weight vectors λ_i to each individual.

Step 3) Individual-level operations:

$For \quad i = 1, \ldots, N_c \quad do$

Step 3.1) Individual-level GA's operations: Perform the reproduction, improvement and update steps over the sub-population P_i; subject to the hybrid variant e.g. MOEA/D, as documented in corresponding literature.

Step 3.2) Update External Population:

For $j = 1, \ldots, |P_i|$ *do*

Remove from the EP all solutions dominated by x_{ij} (the individual j, from population i). Add individual x_{ij} to the EP if no solutions from the EP dominate it.

Step 4) Collective-level operations:

Step 4.1) Calculate collective's fitness:

For $i = 1, \ldots, N_c$ *do*

Calculate fitness of the collective C_i, as the average fitness of the population P_i based on the collective-level fitness definition.

Step 4.2) Collective elimination:

Find the collective C_i with the worst fitness value and store the index of that collective, z.

Store the size of the eliminated collective $|P_z|$ as the variable s.

Erase the sub-population P_z of the eliminated collective C_z.

Step 4.3) Collective reproduction:

For $i = 1, \ldots, N_c$ *do*

if $(i \ != \ z)$

Copy the best $\frac{s}{(N_c-1)}$ individuals, according to the collective-level fitness definition, from population P_i to P_z.

then

Copy all individual-level parameters, specific to the implemented algorithms from eliminated collective C_z to the new collective, e.g. In the case of the MOEA/D hybrid: assign the weight vector λ_z randomly to population P_z.

Step 5) Termination: If the stopping criteria are met, stop and give EP as an output. Otherwise, return to **Step 3)**.

It can be seen that general methodology of the MLSGA-hybrid is not significantly different than of the MLSGA. The only difference is the change in individual-level mechanisms, whereas classification and collective-reproduction mechanisms remains unmodified. Implemented individual-level mechanisms are recreated from the literature or utilised as is, if the code was provided, without any significant modifications. Detailed principles of working for them can be found in corresponding literature.

However, it has to be considered that some of implemented algorithms includes hard-coded individual-assigned parameters, such as weight vectors in MOEA/D. Therefore, in the sub-population based MLSGA methodology a following strategy is proposed: all parameters are defined before classification step, for the initial population as whole, but these are assigned to the individuals after the classification step; where individuals in the first collective are

TABLE 4.1: Fitness functions definitions on each level of selection for MLSGA-hybrid

Type of collective	**MLS1**	**MLS2**	**MLS2R**
Individual-level	$f_1(x)$ *and* $f_2(x)$		
Collective-level	$\frac{f_1(x)+f_2(x)}{2}$	$f_1(x)$	$f_2(x)$

assigned first, followed by the assignment for the next collectives until every individual has values assigned to it. Inside each collective the assignment is performed randomly. This strategy leads to a higher diversification of sub-populations, as each of them potentially contains different parameters. Pre-tests regarding MOEA/D have shown that if each sub-population has the same weight-vectors, this leads to overlapping of collectives and to highly ineffective search in the process.

Importantly as implemented individual-level mechanisms requires all objectives to operate effectively, the fitness separation introduced in the original MLS-U strategy cannot be utilised. Instead, the strategy similar to the MLS-U2 variant, proposed in a previous chapter, is implemented; where both objectives are always available on the individual-level, as shown in Table 4.1. In previous chapter it has been shown that there is no significant difference between the performance of MLS-U2 and MLS-U, with only change that the MLS-U2 variant leads to a higher diversification of collectives.

4.1.2 Hyper-parameter setting and benchmarking

Implemented individual-level reproduction mechanisms remain unchanged in comparison to the original algorithms. Therefore, the operational parameters related to those mechanisms are directly copied from the corresponding literature, and are detailed in Appendix E. The same parameters are used when simulating original algorithms for the purpose of comparative benchmarks, including population sizes. It is possible that the performance of MLSGA-hybrid could be higher after careful hyper-tuning of those, but the principles of this study is not to find the best setting for each individual-level mechanism.

Different MLSGA parameters: population size, number of collectives, steps between collective reproduction and number of eliminated collectives have been parametrically evaluated and the description of hyper-parameter tuning can be found in the section 4.4 of this chapter. Therefore, only the parameters that give the best performance across all of the problems are used to perform presented simulations. Those parameters are detailed in Table 4.2 and remain constant over all the runs for all tested functions. The distinction between parameter values used MTS and other hybrids can be observed, and to those used by the standard MLSGA. It is due to the fact that most of the algorithms are dependent on developing and maintaining uniform Pareto front each generation. Therefore, the collectives must maintain a high number of individuals for the individual-level mechanisms to be effective; resulting in bigger population sizes in overall. Additionally, frequency of the collective elimination

Table 4.2: Hyper-parameters utilised in the MLSGA-hybrids for benchmarking.

Parameter	MTS	Others
Population size	225	1000
Crossover rate	1	
Mutation rate	0.08	
Number of collectives	6	8
Collective reproduction delay	1	10

has to be lowered to occur once every 10 generations. Otherwise, the collectives do not have enough iterations to properly develop the fronts, leading to a premature elimination of potentially good sub-populations. In contrast to other methodologies, MTS utilises multiple local searches for each individual, requiring a significantly higher number of iterations per generation. That results in a requirement for smaller populations. In this case, the collective reproduction may occur more often, as the search strategy is not dependent on a front propagation.

Similarly, to the previous chapter, the tests are performed over 30 separate runs, with 300,000 function evaluations for each run; and both IGD and HV are used as the performance indicators. The comparative performance is assessed via average IGD and HV values across all runs, and only this data is presented in the main part of this thesis. Additionally, wherever applicable, the Wilcoxon's rank sum test was conducted in order to assess the statistical significance between the results with a confidence level of $\alpha = 0.05$.

4.1.3 Computational complexity and constraint handling

In this case the computational cost of the MLSGA-hybrid is determined by two operations: individual-level reproduction, taken from the embedded algorithm and thus with the same complexity, denoted here as C, and the collective reproduction of MLSGA, which requires $O(mN^2)$ comparisons. Therefore, the overall computational complexity of one generation of MLSGA-hybrid is bounded by C, or $O(mN^2)$ whichever is higher.

The constraint handling strategy remains the same as in the previous version of the MLSGA.

4.2 Improvement over the MLSGA and comparison to the implemented algorithms

The performance is evaluated over 14 different problems: 7 unconstrained functions, UF1-UF7, and 7 constrained cases, CF1-7. Tests on ZDT1-6 functions have been conducted as well, but those are not included in the main part of this thesis, due to no significant differences in comparative performance on those problems. This phenomenon is caused by the relative simplicity of the ZDT set.

The average IGD and HV scores achieved by the MLSGA-hybrids are calculated for each tested problem and compared with the scores obtained by the standard MLSGA and implemented algorithms. The results are presented in Table4.3 for CF1-7 cases, and in Table 4.4 for UF1-7 problems. MTS, MOEA/D-PSF and MOEA/D-MSF, and the corresponding MLSGA-hybrids, are not included in the comparison as their performance shows similar trends to MOEA/D and MLSGA MLSGA_MOEA/D comparative performance.

The constrained results, Table 4.3, show that the MLSGA-hybrids are significantly outperforming implemented algorithms on most of the cases. It is 6 out of 7 cases for U-NSGA-III based variant, 5 cases for the BCE hybrid and all cases for the IBEA one, according to both metrics, and in 5 and 7 cases for MOEA/D according to the IGD and HV, respectively. Interestingly, HEIA is better than MLSGA_HEIA , on 4 out of 7 cases according to the IGD indicator, but only on one of them according to the HV metric. Furthermore, the performance of HEIA is not significantly better on CF6 problem. Similar principle can be observed for the MLSGA_MOEA/D in comparison to MOEA/D. The better performance of the MLSGA-hybrid according to the HV metric, in comparison to IGD-based results, indicates a stronger focus on the diversity of the search rather than convergence.

It can be noticed that the MLSGA leads to a worse performance on CF1 problem with most of the implemented algorithms. However, it can be explained by a geometry of the objective and search spaces in CF1 function. In this case the spaces consist of a number of straight lines, which are uniformly spread, rather than a proper discontinuity. Therefore, the difficulty of this problem can be derived to finding at least one point in each region, and then properly converge on them separately, and thus it is suggested that a high convergence rather than diversity is needed. On the most of other cases, where the MLSGA-hybrids are being outperformed according to the IGD metric, such as CF2 for HEIA and MOEA/D, and CF4 for HEIA, it is most likely due to outstanding performance of the original algorithms, i.e. IGD values lower than 1E-02. Therefore, this indicates that the MLSGA mechanisms leads to lowered convergence in overall.

Table 4.3: The performance of different MLSGA-hybrid variants on CF test set, in comparison to the corresponding implemented algorithms and simple MLSGA according to the IGD and HV metrics

Case	MLSGA	U-NSGA-III		MOEA/D		BCE		HEIA		IBEA	
	Original	**Hybrid**	**Original**	**Hybrid**	**Original**	**Hybrid**	**Original**	**Hybrid**	**Original**	**Hybrid**	**Original**
						IGD					
CF1	7.801E-02	2.211E-02	1.040E-02	7.828E-03	2.140E-01	1.663E-02	1.202E-02	2.755E-03	3.510E-04	3.357E-02	1.522E-01
CF2	9.774E-02	1.059E-02	2.567E-02	4.503E-03	9.460E-04	1.146E-02	2.007E-02	3.762E-03	7.980E-04	1.411E-02	6.083E-02
CF3	1.899E-01	1.491E-01	2.633E-01	2.093E-01	1.755E-01	**5.900E-01**	5.625E-01	9.273E-02	1.718E-01	1.352E-01	6.019E-01
CF4	1.042E-01	2.999E-02	7.288E-02	4.586E-02	1.495E-01	6.207E-02	9.654E-02	1.453E-02	8.233E-03	4.635E-02	3.089E-01
CF5	2.026E-01	9.921E-02	1.812E-01	1.941E-01	2.190E-01	**3.057E-01**	5.591E-01	5.274E-02	9.337E-02	1.098E-01	4.707E-01
CF6	8.102E-02	3.350E-02	5.332E-02	4.021E-02	1.332E-01	6.206E-02	9.383E-02	4.801E-02	3.615E-02	**8.496E-02**	1.407E-01
CF7	1.933E-01	1.167E-01	2.099E-01	**2.146E-01**	1.275E+0	**3.934E-01**	4.431E-01	8.080E-02	1.248E-01	1.299E-01	2.553E+0
						HV					
CF1	0.8344	0.8604	0.8648	0.8642	0.6932	0.8626	0.8642	0.8677	0.8686	**0.8313**	0.7326
CF2	0.8430	0.8962	0.8914	0.8998	0.8905	0.8957	0.8901	0.9027	0.8889	0.8880	0.8374
CF3	0.6357	0.7383	0.6938	0.7081	0.6910	**0.3978**	0.4003	0.7468	0.6889	0.7141	0.4576
CF4	0.7495	0.8176	0.7882	0.7903	0.7170	0.7958	0.7766	0.8266	0.8251	0.7916	0.6276
CF5	0.6108	0.7473	0.7171	0.6634	0.6610	**0.5574**	0.4334	0.7844	0.7273	0.7321	0.5087
CF6	0.8298	0.8776	0.8694	0.8795	0.8360	0.8473	0.8395	0.8637	0.8665	0.8468	0.8127
CF7	0.7364	0.8190	0.7640	**0.7258**	0.6226	**0.5980**	0.5737	0.8317	0.7775	0.8037	0.3730

*Better results between the hybrid and the individual-level algorithm are highlighted in blue, if the results are statistically better, and in green otherwise. The MLSGA-hybrid variants outperformed by simple MLSGA are in **bold**.*

TABLE 4.4: The performance of different MLSGA-hybrid variants on UF test set, in comparison to the corresponding implemented algorithms and simple MLSGA according to the IGD and HV metrics

Case	MLSGA	U-NSGA-III		MOEA/D		BCE		HEIA		IBEA	
	Original	Hybrid	Original	Hybrid	Original	Hybrid	Original	Hybrid	Original	Hybrid	Original
IGD											
UF1	9.583E-02	2.014E-02	6.162E-02	3.595E-02	2.156E-03	8.349E-02	7.260E-02	1.521E-02	1.617E-03	2.682E-02	9.698E-02
UF2	6.031E-02	1.172E-02	1.755E-02	1.466E-02	8.272E-03	4.569E-02	3.914E-02	1.143E-02	3.147E-03	1.891E-02	4.254E-02
UF3	1.475E-01	1.008E-01	7.572E-02	1.054E-01	2.398E-02	**2.352E-01**	1.483E-01	5.255E-02	2.367E-02	1.385E-01	2.249E-01
UF4	1.476E-01	4.326E-02	3.881E-02	7.742E-02	6.432E-02	6.361E-02	5.910E-02	4.342E-02	3.515E-02	5.585E-02	5.402E-02
UF5	7.700E-01	2.037E-01	2.299E-01	6.875E-01	3.898E-01	**1.677E+0**	1.470E+0	2.331E-01	1.900E-01	2.381E-01	3.745E-01
UF6	2.117E-01	6.843E-02	8.959E-02	1.960E-01	1.661E-01	4.961E-01	5.089E-01	7.250E-02	1.321E-01	6.093E-02	2.184E-01
UF7	2.172E-01	1.150E-02	2.819E-02	1.858E-02	3.220E-03	9.223E-02	3.178E-02	8.423E-03	1.811E-03	2.361E-02	2.750E-01
HV											
UF1	0.8434	0.9067	0.8796	0.8829	0.9142	0.8604	0.8586	0.9105	0.9160	0.8935	0.8535
UF2	0.8731	0.9103	0.9034	0.9044	0.9102	0.8870	0.8906	0.9113	0.9154	0.8986	0.8760
UF3	0.7176	0.7986	0.8218	0.8741	0.8900	0.8254	0.8576	0.8864	0.9049	0.7360	0.7008
UF4	0.7676	0.8059	0.8069	0.7790	0.7880	0.7858	0.7921	0.8070	0.8108	0.8015	0.7999
UF5	0.3628	0.7135	0.6791	0.3995	0.5841	**0.0383**	0.0710	0.6932	0.7057	0.6564	0.5795
UF6	0.6592	0.7986	0.7234	0.6596	0.7265	**0.4965**	0.4951	0.7684	0.7049	0.7738	0.6114
UF7	0.7019	0.8686	0.8610	0.8608	0.8711	0.8112	0.8434	0.8713	0.8740	0.8588	0.6792

Better results between the hybrid and the individual-level algorithm are highlighted in blue, if the results are statistically better, and in green otherwise. The MLSGA-hybrid variants outperformed by simple MLSGA are in ***bold****.*

However, the results for unconstrained problems, Table 4.4, are less favourable for MLSGA-hybrids. Corresponding MLSGA-hybrids are outperformed by MOEA/D on all cases, BCE and HEIA on 5 out of 7 and U-NSGA-III on two of them, according to both utilised indicators. This shows that the MLSGA-hybrid is not suited for continuous and unconstrained problems, due to requirement of a high convergence rate rather than diversity on them. This is further supported by a high performance of the MLSGA-hybrids on discontinuous UF6 problem, where only MLSGA_MOEA/D is outperformed by its corresponding algorithm. From the presented data it can be seen that for many unconstrained cases the original algorithms have been able to accurately approximate true Pareto optimal front, i.e. IGD values lower than 1E-02. That further supports the statement that MLSGA mechanisms leads to a decreased convergence, as indicated previously. Logically, implementing additional mechanisms to a highly specialist-solver will lead to a decreased performance, as the additional diversity is not needed, so the computational effort is wasted.

Surprisingly, despite utilisation of a stronger individual-level mechanisms, the MLSGA-hybrids are exhibiting a worse performance than original MLSGA on some of the tested problems. However, on all of those cases the implemented algorithm is achieving a poor quality of results as well, e.g. MOEA/D and BCE on CF7 problem and BCE on UF5 case. Therefore, it is indicating that MLSGA-hybrid is bounded by the performance of implemented individual-level mechanisms. It was expected, as performance of the MLSGA-hybrids is dependent on their ability to properly converge on a global optimum. If the individual-level search strategies do not provide a sufficient convergence, then the MLSGA-based algorithms will not be able to further improve it, due to lack of such mechanisms. A similar principle has been shown previously while evaluating standard MLSGA.

According to that it is shown that the MLSGA is preferred on cases where a higher diversity is essential, such as constrained and discontinuous problems, but leads to a lowered convergence on more continuous cases. However, more tests on a wider set of functions is required in order to further validate this statement.

4.2.1 Extending the benchmarking set by high diversity demanding functions

In order to further investigate the impact of MLSGA mechanisms on the diversity of obtained solutions, the benchmarking test set is extended by imbalanced and biased functions. Following the state-of-the-art literature, 14 functions in total are added to the utilised set, with 11 unconstrained cases, MOP1-5, IMB1-3, IMB7-9 and with 3 constrained imbalanced functions, IMB11-13. Importantly all of these problems, except of MOP4 have continuous search and objectives spaces, which may have impact on the final performance, as it was indicated previously that discontinuous spaces are preferred by the MLSGA. Therefore, due to that, and reasons discussed previously in the literature review, Chapter 2, those may promote certain solutions rather than a higher diversity, as originally suggested in the literature [32, 33].

Furthermore, a set of difficulty adjustable and constrained problems is added in order to tailor functions with desired characteristics [44]. Following the guidelines in [44], the design parameters are set to: (0.5,0,0) for diversity-hard problems; (0,0.5,0) for feasibility-hard; and (0,0,0.5) for convergence hard cases; these are denoted as DAS-CMOP1-6(5), DAS-CMOP1-6(6) and DAS-CMOP1-9(7) respectively.

Results are simulated and the average IGD and HV values achieved by the MLSGA-hybrids are compared to the scores of the corresponding algorithms. These are presented in Table 4.5 for MOP and IMB problems; and for DAS-CMOP cases in Table 4.6 for the IGD indicator, and in Table 4.7 for HV. Similarly, to the previous benchmarks, the results for MTS, MOEA/D-PSF and MOEA/D-MSF, and corresponding MLSGA-hybrids, are not included as those show similar trends to MOEA/D and the MLSGA_MOEA/D comparative performance. Original MLSGA is not included due to inferior performance of this strategy on those cases.

From presented data, Table 4.5 it can be seen that the MLSGA-hybrids are outperforming corresponding algorithms on most of the biased MOP problems, or at least achieve a comparable performance, whereas they are ineffective on the most of imbalanced IMB cases, especially IMB7-13. This indicates that MLSGA's diversity enhancing strategy is effective on biased cases, such as MOP and ZDT; discontinuous and constrained cases, as shown previously; but other kind of diversity preservation is needed for IMB problems.

However, as it has been suggested in the literature review, Chapter 2, those functions may require problem decomposition techniques or a specific reproduction mechanism, rather than a dispersed search. IMB7-9 cases introduce the variable-linkage, therefore adjustments to crossover and mutation strategies are preferred, as suggested in [33,86], whereas in IMB11-13 the constraint isolation is introduced, and thus a different constraint handling strategy have to be utilised, as suggested in [33, 142]. Therefore, the only cases in the IMB set where a dispersed search is preferred are IMB1-3 problems due to introduced bias towards favoured regions of the Pareto optimal front. On those cases the MLSGA mechanisms lead to a better performance for most of the compared algorithms. That is here supported by an outstanding performance of MOEA/D on all IMB problems, especially IMB7-13 as the IGD values are lower than 1E-03, and a high performance of all tested algorithms on IMB7-13 cases; which shows that the diversity enhancement was not needed and thus indicating a relative simplicity of this set. According to this it is suggested that MLSGA do not provide reliable benefits for a highly imbalanced problems, due to a specific structure of these functions. MLSGA may encourage exploration of different regions of objective space, which is beneficial on cases with discontinuous or biased objective spaces, but do not help to maintain a high diversity of variables which is needed for the imbalanced problems such as IMB7-13.

TABLE 4.5: The performance of different MLSGA-hybrid variants on MOP and IMB test sets, in comparison to the corresponding implemented algorithms according to the IGD and HV metrics

Case	U-NSGA-III		MOEA/D		BCE		HEIA		IBEA	
	Hybrid	Original	Hybrid	Original	Hybrid	Original	Hybrid	Original	Hybrid	Original
					IGD					
MOP1	1.794E-01	1.810E-01	1.837E-01	1.805E-01	1.839E-01	1.841E-01	1.781E-01	1.820E-01	1.818E-01	1.865E-01
MOP2	3.550E-01	3.550E-01	3.027E-01	3.411E-01	3.263E-01	3.333E-01	2.035E-01	3.411E-01	3.550E-01	3.553E-01
MOP3	4.102E-01	4.301E-01	4.326E-01	4.858E-01	4.656E-01	4.842E-01	4.157E-01	4.717E-01	4.088E-01	4.085E-01
MOP4	2.535E-01	2.999E-01	2.708E-01	2.643E-01	2.303E-01	2.624E-01	2.244E-01	2.813E-01	2.622E-01	2.990E-01
MOP5	1.870E-01	1.901E-01	3.066E-01	2.559E-01	2.912E-01	2.959E-01	1.880E-01	2.550E-01	2.199E-01	2.357E-01
IMB1	7.085E-02	8.157E-02	1.571E-01	9.513E-02	2.210E-01	2.231E-01	2.278E-02	9.673E-03	2.169E-01	3.375E-01
IMB2	1.173E-01	1.278E-01	1.964E-01	1.157E-01	1.958E-01	1.971E-01	1.649E-01	1.354E-01	1.643E-01	2.244E-01
IMB3	2.127E-01	2.406E-01	2.993E-01	2.228E-01	2.010E-01	2.657E-01	2.942E-01	2.840E-01	2.602E-01	2.974E-01
IMB7	1.626E-03	8.170E-04	7.121E-03	7.930E-04	1.858E-01	2.314E-02	1.285E-03	7.260E-04	1.222E-03	3.146E-03
IMB8	9.050E-04	8.140E-04	8.088E-03	6.480E-04	2.456E-01	4.389E-02	1.156E-03	7.120E-04	1.623E-03	3.445E-03
IMB9	1.001E-03	9.030E-04	9.185E-03	7.700E-04	4.109E-01	1.409E-01	7.580E-04	7.200E-04	5.760E-04	5.529E-03
IMB11	1.078E-01	1.105E-01	4.858E-02	9.990E-04	1.240E-01	9.832E-02	9.401E-02	9.311E-02	1.259E-01	1.693E-01
IMB12	8.124E-02	7.249E-02	2.900E-02	6.280E-04	7.072E-02	5.878E-02	5.863E-02	1.331E-03	1.147E-01	2.519E-01
IMB13	8.735E-02	8.299E-02	6.310E-02	9.540E-04	9.513E-02	6.482E-02	6.518E-02	1.230E-02	1.348E-01	2.956E-01
					HV					
MOP1	0.8473	0.8462	0.8442	0.8463	0.8442	0.8440	0.8479	0.8452	0.8462	0.8426
MOP2	0.7500	0.7500	0.7556	0.7508	0.7530	0.7521	0.7785	0.7527	0.7499	0.7472
MOP3	0.7482	0.7310	0.7300	0.6964	0.7070	0.6989	0.7433	0.7054	0.7500	0.7492
MOP4	0.7956	0.7871	0.7908	0.7924	0.8004	0.7954	0.8063	0.7803	0.7913	0.7423
MOP5	0.8404	0.8393	0.7317	0.7818	0.7638	0.7351	0.8323	0.7877	0.8243	0.8090
IMB1	0.8909	0.8876	0.8626	0.8863	0.8438	0.8432	0.9088	0.9134	0.8455	0.7696
IMB2	0.7878	0.7834	0.7084	0.8306	0.7218	0.7162	0.7306	0.7676	0.7990	0.6809
IMB3	0.7422	0.6756	0.5214	0.7597	0.6843	0.6249	0.5376	0.5686	0.6585	0.5327
IMB7	0.9160	0.9164	0.9138	0.9163	0.8529	0.9075	0.9161	0.9164	0.9161	0.9153
IMB8	0.8746	0.8747	0.8716	0.8747	0.7682	0.8568	0.8745	0.8747	0.8743	0.8735
IMB9	0.8032	0.8034	0.7997	0.8033	0.5770	0.7370	0.8032	0.8032	0.8034	0.8024
IMB11	0.8324	0.8344	0.9031	0.9163	0.8162	0.8409	0.8461	0.8480	0.8207	0.8061
IMB12	0.7487	0.7520	0.8640	0.8747	0.7499	0.7621	0.7617	0.8743	0.7211	0.6553
IMB13	0.6682	0.6696	0.7464	0.8329	0.6604	0.6807	0.6802	0.8021	0.6222	0.5577

Better results between the hybrids and corresponding individual-level algorithm are highlighted in blue, if the results are statistically better, and in green otherwise.

Positive impact of the MLSGA mechanisms on the diversity is shown on the DAS-CMOP test set, Tables 4.6 and 4.7. It can be seen that the MLSGA-hybrids have significantly better performance than corresponding original algorithms on most of the tested diversity- and feasibility-hard problems, but they are outperformed on many convergence-hard cases. In case of the U-NSGA-III-based variant, it is better than implemented algorithm in 11 and 9 cases out of 12 according to the IGD and HV indicators respectively for the diversity- and feasibility-hard problems, but worse in 5 out of 6 cases for the convergence-hard functions. For the HEIA-based variant, it is 11 out of 12 cases, for both indicators; and 3 out of 6 cases for the IGD and 4 out of 6 for the HV indicator correspondingly. For IBEA, and BCE an improvement is observed for all cases but one, DAS-CMOP3(6), even the convergence-hard problems. However, due to lack of the significant differences on most of the convergence-hard cases between the corresponding MLSGA-hybrids and IBEA and BCE algorithms, it is here suggested that this is caused by a high randomness of the search within these methodologies on the DAS-CMOP(7) cases, leading to a low performance. Opposite can be observed for the MOEA/D based algorithms, where the MLSGA search strategy leads to a worse performance on most of tested cases, similarly to the previous benchmarks. According to presented results, such as the lowered performance of the U-NSGA-III and HEIA variants on convergence-hard problems, but significantly better on the diversity- and feasibility-hard cases, the high potential of MLSGA for the diversity-oriented methodology is proven.

Regarding a low performance of the MLSGA_MOEA/D , MLSGA_MOEA/D-PSF , MLSGA_MOEA/D-MSF and MLSGA_MTS it is here suggested that it is due to incompatibility of the MLSGA's collective-level mechanisms with implemented individual-level strategies. Performance of MOEA/D is highly dependent on the quality of predefined weight-vectors, as indicated in [32, 47]. In the MLSGA_MOEA/D hybrid a newly created collective inherits the weight vectors of the eliminated collective, during the collective reproduction step. However, the assignment of solutions to each vector is random and thus is not subject to the value of this weight vector. Therefore, the individuals do not have the best possible weight vector assigned to them. Furthermore, the neighbourhood of solutions is being recreated basing only on the vector's values, rather than actual “quality” of individuals. Therefore, the new neighbourhood does not accurately reflect the relationship between two neighbour individuals. This lowers an overall fitness of individuals in the new collective, and thus leads to a decreased performance.

TABLE 4.6: The performance of different MLSGA-hybrid variants on DAS-CMOP test set, in comparison to the corresponding implemented algorithms according to the IGD metric

Case	U-NSGA-III		MOEA/D		BCE		HEIA		IBEA	
	Hybrid	Original	Hybrid	Original	Hybrid	Original	Hybrid	Original	Hybrid	Original
DAS-CMOP1(5)	3.086E-01	3.685E-01	3.409E-02	3.153E-02	3.686E-01	4.030E-01	6.224E-02	3.398E-01	2.928E-01	4.169E-01
DAS-CMOP2(5)	2.775E-01	3.276E-01	3.466E-02	3.216E-02	3.185E-01	3.540E-01	1.307E-01	3.315E-01	2.691E-01	3.465E-01
DAS-CMOP3(5)	3.298E-01	3.557E-01	3.266E-02	1.297E-01	3.564E-01	5.362E-01	8.641E-02	2.576E-01	3.730E-01	4.835E-01
DAS-CMOP4(5)	5.759E-01	6.273E-01	5.123E+00	4.503E+00	1.390E+01	1.877E+01	8.841E-01	1.158E+00	5.265E-01	9.292E-01
DAS-CMOP5(5)	5.705E-01	6.540E-01	4.792E+00	5.534E+00	1.391E+01	2.047E+01	1.037E+00	9.791E-01	6.052E-01	9.024E-01
DAS-CMOP6(5)	5.627E-01	6.415E-01	5.711E+00	5.569E+00	1.292E+01	2.053E+01	8.453E-01	1.249E+00	6.369E-01	1.190E+00
DAS-CMOP1(6)	7.279E-01	7.650E-01	6.321E-01	6.403E-01	7.429E-01	7.494E-01	6.347E-01	6.393E-01	7.188E-01	7.958E-01
DAS-CMOP2(6)	7.766E-01	8.040E-01	6.509E-01	6.556E-01	7.852E-01	8.125E-01	6.513E-01	6.541E-01	7.619E-01	8.331E-01
DAS-CMOP3(6)	7.055E-01	6.330E-01	6.147E-01	6.144E-01	7.175E-01	6.998E-01	6.185E-01	7.063E-01	7.457E-01	7.004E-01
DAS-CMOP4(6)	1.122E-01	2.749E-01	8.000E+13	9.333E+13	1.000E+14	1.000E+14	6.667E+12	4.667E+13	2.611E-01	5.333E+13
DAS-CMOP5(6)	1.182E-01	2.338E-01	1.000E+14	8.667E+13	1.000E+14	1.000E+14	1.333E+13	3.333E+13	2.211E-01	7.333E+13
DAS-CMOP6(6)	2.095E-01	2.601E-01	1.000E+14	1.000E+14	1.000E+14	1.000E+14	2.667E+13	7.333E+13	1.333E+13	6.667E+13
DAS-CMOP1(7)	5.764E-01	5.821E-01	1.464E-01	1.435E-01	5.940E-01	6.074E-01	5.709E-01	5.624E-01	5.297E-01	6.126E-01
DAS-CMOP2(7)	5.658E-01	4.817E-01	3.476E-02	3.216E-02	3.643E-01	8.075E-01	6.172E-01	5.941E-01	2.548E-01	3.557E-01
DAS-CMOP3(7)	6.988E-01	6.715E-01	2.586E-02	2.216E-02	6.990E-01	7.046E-01	6.700E-01	7.896E-01	2.311E-01	2.928E-01
DAS-CMOP4(7)	1.365E+00	1.206E+00	5.235E+00	6.651E+00	1.153E+01	2.000E+01	1.903E+00	1.935E+00	6.678E-01	1.189E+00
DAS-CMOP5(7)	1.558E+00	1.231E+00	5.622E+00	4.948E+00	1.100E+01	1.927E+01	2.131E+00	2.131E+00	9.242E-01	1.262E+00
DAS-CMOP6(7)	1.429E+00	1.214E+00	7.044E+00	6.944E+00	1.223E+01	1.819E+01	2.026E+00	2.070E+00	9.717E-01	1.602E+00

Better results between the hybrids and corresponding individual-level algorithm are highlighted in blue, if the results are statistically better, and in green otherwise.

Table 4.7: The performance of different MLSGA-hybrid variants on DAS-CMOP test set, in comparison to the corresponding implemented algorithms according to the HV metric

Case	U-NSGA-III		MOEA/D		BCE		HEIA		IBEA	
	Hybrid	**Original**	**Hybrid**	**Original**	**Hybrid**	**Original**	**Hybrid**	**Original**	**Hybrid**	**Original**
DAS-CMOP1(5)	0.4865	0.4544	0.8050	0.7931	0.4490	0.4308	0.7628	0.4804	0.4988	0.4572
DAS-CMOP2(5)	0.6797	0.6515	0.8964	0.8966	0.6474	0.6281	0.8121	0.6525	0.6936	0.6396
DAS-CMOP3(5)	0.6804	0.6412	0.8708	0.7785	0.6032	0.4826	0.8102	0.7226	0.6920	0.6152
DAS-CMOP4(5)	0.4174	0.3658	0	0.1101	0	0	0.2695	0.1933	0.4471	0.3063
DAS-CMOP5(5)	0.5223	0.4862	0.0386	0.0006	0	0	0.3186	0.3588	0.4955	0.3861
DAS-CMOP6(5)	0.4961	0.4410	0	0.0002	0	0	0.3526	0.2172	0.4277	0.1645
DAS-CMOP1(6)	0.5764	0.5567	0.8846	0.8856	0.5553	0.5240	0.8854	0.8848	0.6355	0.5567
DAS-CMOP2(6)	0.7412	0.7232	0.9394	0.9401	0.7164	0.6984	0.9400	0.9395	0.7674	0.7141
DAS-CMOP3(6)	0.7465	0.7139	0.9237	0.9245	0.7895	0.7333	0.9220	0.8341	0.7539	0.7091
DAS-CMOP4(6)	0.5613	0.6625	0.1303	0.0436	0	0	0.5132	0.2953	0.5966	0.2926
DAS-CMOP5(6)	0.6015	0.6935	0	0.1239	0	0	0.5273	0.4194	0.6258	0.2140
DAS-CMOP6(6)	0.5493	0.6782	0	0	0	0	0.4155	0.1750	0.4405	0.1912
DAS-CMOP1(7)	0.3197	0.3125	0.7802	0.7815	0.2988	0.2852	0.3293	0.3433	0.3678	0.3035
DAS-CMOP2(7)	0.4909	0.5418	0.9053	0.9065	0.6101	0.3079	0.4571	0.4793	0.7107	0.6508
DAS-CMOP3(7)	0.3693	0.3873	0.8808	0.8820	0.3696	0.3632	0.3879	0.3364	0.6840	0.6448
DAS-CMOP4(7)	0.0851	0.1465	0.0002	0.0028	0	0	0.0027	0.0027	0.3402	0.1343
DAS-CMOP5(7)	0.1141	0.2063	0	0.1572	0	0	0.0028	0.0029	0.3449	0.2125
DAS-CMOP6(7)	0.1154	0.1850	0	0	0	0	0.0029	0.0026	0.2647	0.0867

Better results between the hybrids and corresponding individual-level algorithm are highlighted in blue, if the results are statistically better, and in green otherwise.

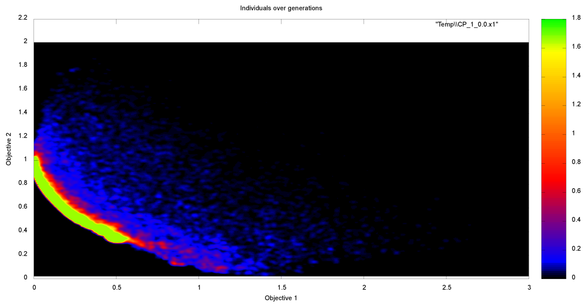
MTS utilises multiple local search methods in order to evolve each individual. Therefore, the fitness calculations per generations are significantly higher than in other GAs. This results in a lowered number of total generations. As an example, in the conducted benchmarks MTS lasted 7 generations in average with 300,000 iterations, while other algorithms reach at least 300 generations during that time. According to that, the collective-level mechanisms are not able to contribute to the search pattern in a reasonable part, thus explaining a low performance of the MLSGA_MTS . These issues could be solved by adjusting the collective-level mechanisms, to be bespoke to the algorithm utilised on the individual-level, so the characteristic of the implemented algorithms would be considered by them. However, it is not a principle of this thesis to develop a perfectly adjusted solution for each of tested search strategies.

4.3 Evaluating the impact of MLS

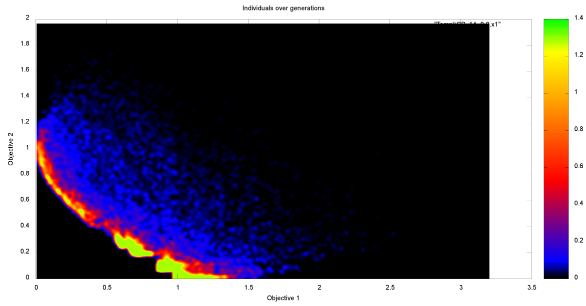
In previous sections it has been shown that implementation of stronger individual-level mechanisms leads to a significantly better performance in comparison to standard MLSGA. Furthermore, the MLSGA-hybrids are usually outperforming the implemented algorithms on most of tested cases, especially problems that require high diversity of the search. However, it is yet to be investigated if this behaviour is result of the region-based search, which is retained from the original MLSGA, or it is due to utilisation of sub-population approach or other new phenomena.

For this purpose, the combined search areas for each type of collective are visualised in the form of heat maps and compared with similar data obtained by simulating the implemented algorithm separately. These are presented in Fig. 4.1 for MLSGA_U-NSGA-III on UF2 problem, and in Fig. 4.2 for MLSGA_MOEA/D on CF2 case.

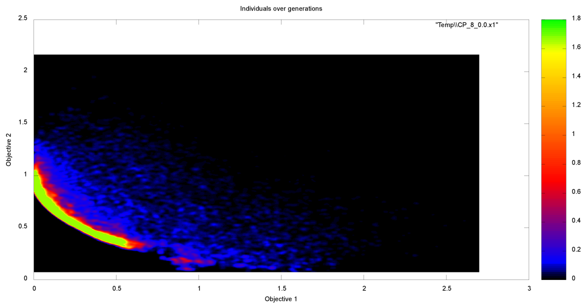
For both MLSGA-hybrids, for each type of collective a favoured region can be observed. For the U-NSGA-III based variant, Fig. 4.1, it is central-left region of the objective space for MLS1 and MLS2R types, range 0 to 0.5 of the Objective 1; and right-hand side of the Pareto optimal front for the MLS2-type, range 0.5-1.3. Interestingly, MLS1 and MLS2R preferred regions are similar, but it is caused by the bias towards the left-hand side of the Pareto optimal front introduced in UF2 problem. This is further supported by similar behaviour shown by U-NSGA-III, where the poor convergence can be observed in range 0.6-1. The differentiation between collectives is even more noticeable for MLSGA_MOEA/D on CF2 problem, Fig. 4.2. In this case MLS2 and MLS2R focuses on the right- and left-hand side of the Pareto optimal front, ranges 0.6-1 and 0-0.4 correspondingly; and the most uniform search can be observed for MLS1, but with a stronger focus towards the central region, range 0.1-0.4 of the Objective 1. According to that, it is shown that the region-based search strategy exhibited by standard MLSGA is maintained with stronger individual-level mechanisms. Interestingly in both presented cases, each type of collective is able to maintain a wide range of solutions instead



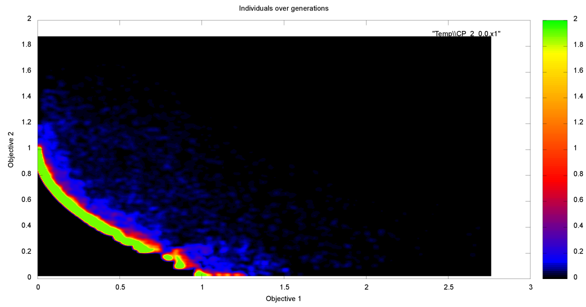
(A) MLS1-type



(B) MLS2-type

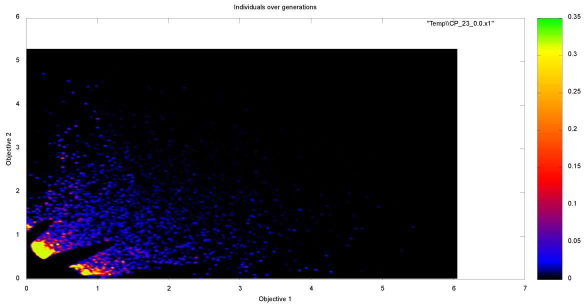


(C) MLS2R-type

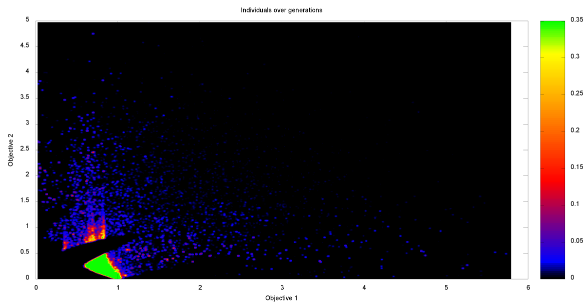


(D) U-NSGA-III

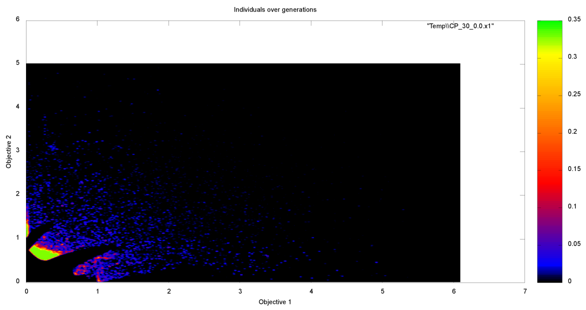
FIGURE 4.1: The heat maps of different collective types in the MLSGA_U-NSGA-III algorithm compared to the U-NSGA-III search pattern on UF2 problems.



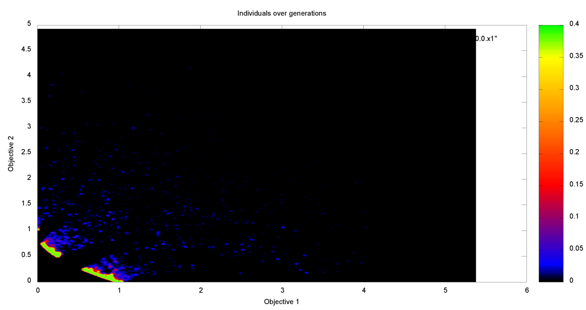
(A) MLS1-type



(B) MLS2-type

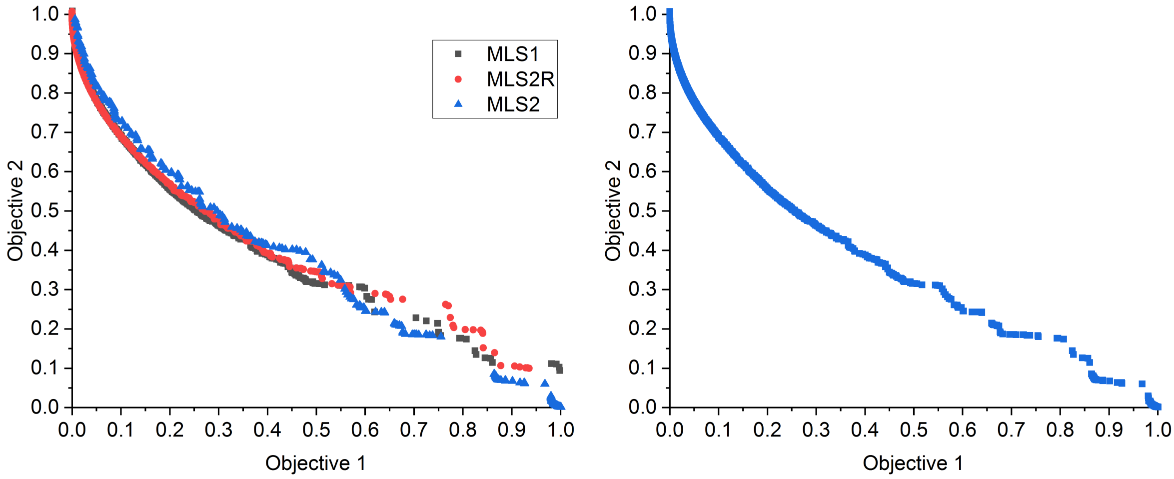


(C) MLS2R-type



(D) MOEA/D

FIGURE 4.2: The heat maps of different collective types in the MLSGA_MOEA/D algorithm compared to the MOEA/D search pattern on CF2 problems.



(A) Fronts obtained by collectives with different MLS-types

(B) Combined front

FIGURE 4.3: Combined Pareto optimal front of the MLSGA_U-NSGA-III on UF2 problem.

of concentrating only on the "preferred" areas. This indicates that the MLSGA's collective-level mechanisms are not harshly punishing the diversity inside each group. Therefore, each sub-group is simply a local instance of the implemented algorithm with a bias toward certain regions of the objective space, but without strong limitations in a free exploration. That explains a high performance of the MLSGA based algorithms.

Comparing to the original algorithms, U-NSGA-III in Fig. 4.1 and MOEA/D in Fig. 4.2, it can be seen that both MOEA/D and U-NSGA-III are developing a more uniform search than the MLSGA. This is due to utilisation of a search strategy that is based on the prevalence of a single front of non-dominated solutions. In the MLSGA, multiple fronts are maintained, and the final Pareto optimal front is a combination of them as illustrated in Fig. 4.3. In this case it can be observed that the final Pareto optimal front is combination of region 0-0.1 of the MLS2R-type collective, and regions 0.1-0.55 and 0.55-1 for the MLS1 and MLS2-types of collectives respectively, according to the first objective. High independence of collectives allows them to move around the infeasible "gaps" more easily in the objective space created by the constraints or the areas with discontinuities. Furthermore, due to utilisation of the collective elimination and reproduction mechanism, collectives are less likely to permanently get stuck on the boundaries of infeasible regions. This claim is supported by high gains in performance shown by the MOEA/D based hybrids on constrained and discontinuous problems. MOEA/D's search strategy is ineffective on non-continuous spaces as utilised weights are defined in straight lines. Therefore, these lines may pass through the infeasible regions, forcing the algorithm to look for valid solutions in these regions and thus leading to an inefficient search.

Table 4.8: Performance comparison of the sub-population and single-population variants of different algorithms taken from the current state-of-the-art, according to the IGD metric.

Case	U-NSGA-III		MOEA/D		HEIA		MTS	
	Sub-pop	Original	Sub-pop	Original	Sub-pop	Original	Sub-pop	Original
UF1	**2.074E-02**	6.162E-02	1.406E-02	2.156E-03	**1.698E-02**	1.617E-03	**2.420E-03**	1.829E-03
UF2	**1.580E-02**	1.755E-02	1.008E-02	8.272E-03	**1.175E-02**	3.147E-03	**2.348E-03**	1.592E-03
UF3	**1.084E-01**	7.572E-02	8.167E-02	2.398E-02	**6.937E-02**	2.367E-02	4.183E-02	4.294E-02
UF4	**4.564E-02**	3.881E-02	7.376E-02	6.432E-02	**4.441E-02**	3.515E-02	2.939E-02	2.931E-02
UF5	**2.088E-01**	2.299E-01	8.873E-01	3.898E-01	**2.754E-01**	1.900E-01	1.138E-01	1.137E-01
UF6	**6.988E-02**	8.959E-02	1.140E-01	1.661E-01	**7.429E-02**	1.321E-01	6.470E-02	6.332E-02
UF7	**1.182E-02**	2.819E-02	9.913E-03	3.220E-03	**9.234E-03**	1.811E-03	2.832E-02	3.213E-02
CF1	**2.615E-02**	1.040E-02	**8.530E-03**	2.140E-01	**2.946E-03**	3.510E-04	2.461E-02	2.209E-02
CF2	**1.090E-02**	2.567E-02	2.327E-03	9.460E-04	**4.621E-03**	7.980E-04	**1.308E-03**	1.051E-03
CF3	**1.563E-01**	2.633E-01	8.515E-02	1.755E-01	**9.872E-02**	1.718E-01	1.782E-01	1.169E-01
CF4	**3.067E-02**	7.288E-02	1.872E-02	1.495E-01	**1.496E-02**	8.233E-03	5.829E-03	5.704E-03
CF5	**1.021E-01**	1.812E-01	7.018E-02	2.190E-01	**5.635E-02**	9.337E-02	5.110E-02	2.340E-02
CF6	**3.442E-02**	5.332E-02	4.872E-02	1.332E-01	**4.923E-02**	3.615E-02	6.005E-03	5.445E-03
CF7	**1.197E-01**	2.099E-01	7.159E-02	1.275E+00	**8.983E-02**	1.248E-01	3.815E-02	2.462E-02
MOP1	**1.836E-01**	1.810E-01	1.822E-01	1.805E-01	**1.832E-01**	1.820E-01	1.347E-01	1.332E-01
MOP2	**3.635E-01**	3.550E-01	2.759E-01	3.411E-01	**2.667E-01**	3.411E-01	1.430E-01	1.767E-01
MOP3	**4.278E-01**	4.301E-01	4.136E-01	4.858E-01	**4.295E-01**	4.717E-01	1.492E-01	1.415E-01
MOP4	**2.625E-01**	2.999E-01	2.436E-01	2.643E-01	**2.367E-01**	2.813E-01	7.737E-02	6.607E-02
MOP5	**2.004E-01**	1.901E-01	2.478E-01	2.559E-01	**1.975E-01**	2.550E-01	1.064E-01	8.611E-02

Better results between the sub-population and single-population variants are highlighted in blue, if the results are significantly better, and in green otherwise. A value is in bold if the sub-population variant is significantly worse than corresponding MLSGA-hybrid.

Finally, the effect of population separation on the final performance is investigated. It has been indicated by other researchers that population separation by itself may provide certain benefits on the discontinuous problems [46]. For that purpose, sub-population variants of the implemented algorithms are developed for the purpose of this thesis and evaluated. They are prepared following the MLSGA's methodology, but with exclusion of the collective-level mechanisms i.e. without steps 2.2 and 4 in the MLSGA-hybrid pseudo-code. Developed variants are simulated over the same test cases, and the results are compared to the original algorithms. Those are presented according to the IGD indicator in Table 4.8 for CF, UF and MOP problems. The results according to the HV indicator; with IMB and DAS-CMOP test sets; and for the MOEA/D-PSF, MOEA/D-MSF, BCE and MTS are not included in the main part of this thesis, but similar trends to the presented data can be observed.

From the presented data, some similar trends to the MLSGA-hybrids, regarding the impact on performance, can be observed. It can be seen that the sub-population approach leads to a worse performance on average on unconstrained UF cases; and a better effectiveness on constrained CF, discontinuous UF6 and biased MOP cases. Therefore, it supports the statement that the sub-population approach is beneficial on diversity demanding functions, but less preferred on convergence-hard problems. That partly explains the elevated performance of the MLSGA based algorithms on those problems. On constrained and discontinuous problems, each sub-population decreases the chance of premature convergence on the infeasible regions. On continuous and constrained cases those sub-populations are often evaluating the same regions, lowering the efficiency of the search. Comparing the results of the sub-population variants to corresponding MLSGA-hybrids it can be seen that the sub-population variants of the U-NSGA-III and HEIA algorithms are achieving a significantly worse performance than corresponding MLSGA-hybrids on each tested case. Therefore, it is shown that additional mechanisms, such as the collective-level reproduction and fitness separation, are required to obtain a higher diversity of the results. Similar principle has been observed previously with standard MLSGA. The low performance of MLSGA-hybrids against the corresponding sub-population variants of MOEA/D, MOEA/D-MSF, MOEA/D-PSF and MTS, further indicates the incompatibility of collective-level mechanisms with these methodologies, as discussed previously in this chapter. Furthermore, significantly lower performance of the MTS sub-population variant, in comparison to the single-population one, indicates that dividing population is highly disadvantageous for this algorithm, as multiple sub-populations are evaluating the same regions.

4.4 Investigating the sensitivity to hyper-parameters

It has been suggested in the literature review that a high sensitivity to the hyper-parameters may lead to a lowered usefulness to real-world applications, due to lack of practical possibility to tweak them in advance. Therefore, in order to fully evaluate the usability of developed algorithm, the sensitivity to hyper-parameters must be investigated. Importantly, it is very unlikely that algorithm will be insensitive to all of them, as even population sizes

Table 4.9: The min, max and step values for each tested hyper-parameter of the MLSGA algorithm

Parameter	Population size	Number of collectives	Collective elimination
Min.	400	4	2
Max.	1800	10	20
Step	100	1	2

or mutation/crossover rates have a significant impact on the obtained results. However, the parameter-sensitive mechanisms do not decrease the value of the methodology, as long as the best possible setting of those exists and if this setting is universal for all potential problem; or if the clear guide for their set-up is defined that does not need prior knowledge about the optimised case. Along the sensitivity testing, a minor hyper-parameter tuning is conducted to approximate the best set of parameters for the benchmarked problems. Simple generate-evaluate brute-force method based on the Design of Experiment (DOE) approach has been chosen, It is considered sufficient for the principles of this work and preferred over more complex automatic and iterative methods such as: IRace, ParamILS or SMAC [143]. It is not principle of this thesis to produce the best possible results for a narrow range of problems, but to evaluate generality on a wide range of problems and investigate the search strategy of the MLSGA. Extensive hyper-parameter tuning to a narrow selection of benchmarking problems, would limit that generality.

Following the chosen method, a series of benchmarks is conducted with different values of the population size, number of collective and frequency of collectives' elimination. Importantly the parameters are evaluated separately, and the rest of them remains constant, due to computational limitations. The setting of constant values is defined based on the performed pre-benchmarks. Due to computational limitation only U-NSGA-III, HEIA, BCE and MOEA/D are simulated, and the DAS-CMOP, IMB and ZDT test sets are not included. Furthermore, due to huge differences in principles of working of MTS in comparison to other utilised algorithms, this methodology is not included in the main part of hyper-tuning but is discussed separately at the end of this section. Maximum and minimum values, and the steps used for the hyper-parameter tuning are summarised in Table 4.9. Tuning process is following the same rules as in the previous sections. Due to high quantity of data, only the summary is presented according to the following steps:

1. The results for each function are normalised based on the best and worst value obtained from among all tested hyper-parameters. Every problem is normalised separately. According to that, 1 means the best possible value and 0 the worst on a given case.

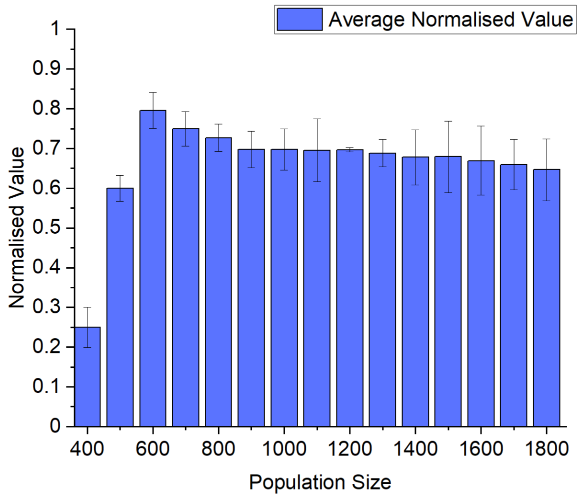
2. Then, the average value and the standard deviation is calculated for all algorithms and functions for a given value of hyper-parameter. Those values are presented on graphs.

Firstly, the impact of population-sizes on the MLSGA performance is evaluated and the results are illustrated with the IGD indicator in Fig. 4.4. The number of collectives is set to

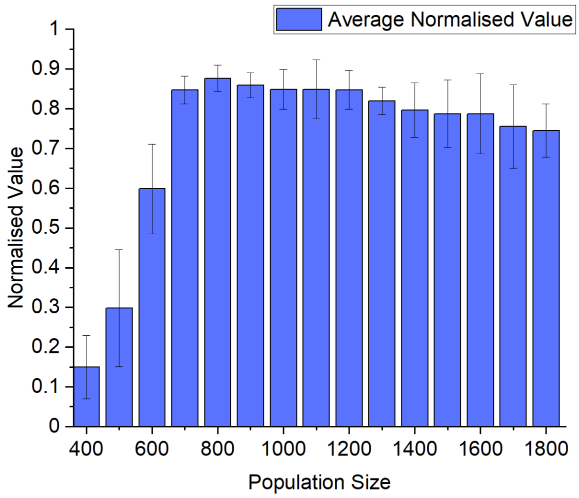
4, 6 and 8, whereas the collective elimination occurs once every ten generations. From the presented graphs it can be seen that the population sizes do not have significant impact on the performance within a wide range, but the larger population sizes are preferred. For 4 collectives, the performance is approximately stable within 600-1800 range, but a visible peak can be observed for 600 individuals. For 6 and 8 collectives, slightly bigger populations are preferred, and the stable zone is within 700-1800 range, with the peak for 1000 individuals for 6 collectives, and a range of 800-1800 for 8 collectives, with the peak in 1000. A significantly lower effectiveness can be observed with small sizes of the population for all presented cases. However, by comparing the data, it can be seen that for a lower number of collectives less individual are required in order to maintain effective search. This indicates that setting the population sizes too low will lead to tiny collectives, which are unable to properly develop the front and thus maintain a high quality search. In contrast, high population sizes result in enormous collectives that would require too many fitness evaluations in order to properly converge. Furthermore, higher population sizes limit the total number of generations that GA may achieve. Therefore, there is less time for the collective reproduction mechanisms. This principle is supported by the current state-of-the-art where population sizes are usually in range 100-600 in order to maximise the convergence rates. Bigger populations lead to lowered convergence, which is usually obtained on generation-to-generation basis [60]. Whereas, too small population sizes do not provide enough individuals to maintain a high diversity [74].

In the next step, the impact of the number of collectives is evaluated, with the calculated results presented in Fig. 4.5 according to the IGD indicator. The population size is constant and equal to 1000, whereas the collective elimination occurs once every 10 generations. From the chart it can be seen that the number of collectives does not have significant impact on the final performance within the range 6-9, with a peak for 8 collectives. Indicating the insensitivity to this parameter when minor changes are applied. Significantly lower performance with 4,5 and 10 collectives is due to synergy between population sizes and the number of collectives as discussed before; thus 1000 individuals is too much for 4 and 5 collectives but is not sufficient for 10 groups. Therefore, to find the best setting of those values, the top results for different numbers of collectives, each with a preferred population size, are compared and presented in Fig. 4.6. From the data it can be seen that the best results are obtained for 8 collectives and 1000 individuals, with comparable performance is achieved for 6, 7 and 9 collectives. It is indicating that at least 2 collectives of each type, and preferably 3 are needed for more effective search. With fewer collectives, there is not enough sub-populations of the same type to provide high quality solutions during the re-population process; whereas with more collectives, too many sub-populations are evaluating the same regions. Therefore, leading to an inefficient search. Surprisingly 8 collectives are preferred, instead of 9, which would seem more natural choice, as there is equal number of collectives of each type. The potential reason for this is that with higher numbers of collectives more individuals are needed in order to perform effective search, as discussed before.

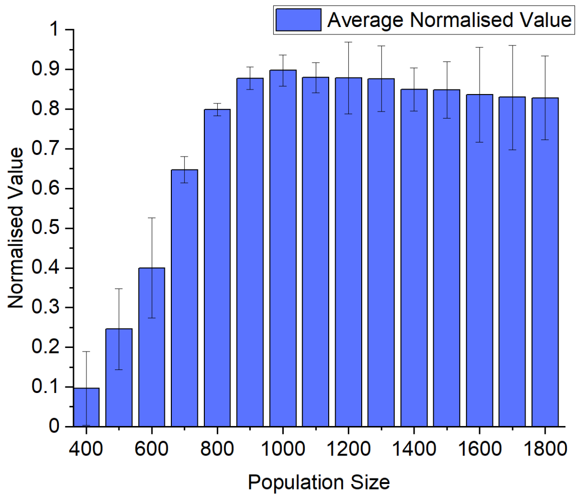
According to that, it is here suggested that 8 collectives and 1000 populations have to be utilised. However, if the optimisation has to be performed with less fitness evaluations e.g.



(A) 4 collectives



(B) 6 collectives



(C) 8 collectives

FIGURE 4.4: Comparison of the IGD values obtained by the 4, 6 and 8 collective versions of the MLSGA with different population sizes.

due to computational limitation, it is suggested that 6 collectives with 800 individuals should be utilised, or even 4 collectives with 600 individuals in more extreme cases. However, it has to be kept in mind that the overall performance will be lowered. According to that it is concluded that one of the MLSGA disadvantages is requirement for larger populations, as most of the state-of-the-art algorithms, needs 100-600 individuals to operate effectively. However, it was expected from an algorithm that focuses on a good spread of obtained solutions rather than the fast convergence.

Finally, the impact of the collective elimination frequency is evaluated for 8 collectives and 1000 individuals. The results for the IGD indicator are presented in Fig. 4.7. It can be observed that elimination should occur after 10 generations, the best result, but a similar performance is maintained within 8-16 range. The reason for this behaviour is that most of the implemented GAs rely on a constant front propagation, as discussed before. Therefore, if

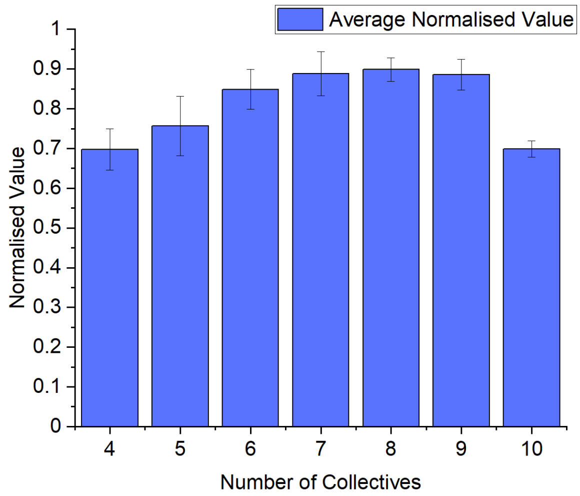


FIGURE 4.5: Comparison of the IGD values obtained by the MLSGA algorithm with different number of collectives.

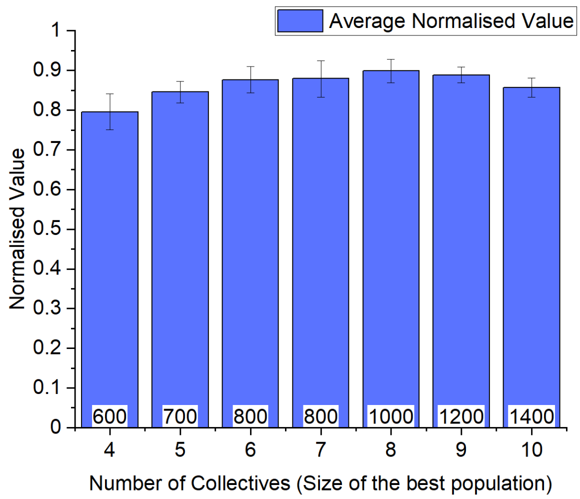


FIGURE 4.6: Comparison of the IGD values obtained by the MLSGA algorithm with different number of collectives, using the best population sizes.

the collective elimination occurs too often, the front cannot be properly generated leading to elimination of potentially good solutions and thus a random search. Whereas if the frequency is too low the collective-level mechanisms do not have a significant impact on the search. According to the presented data it is concluded that the MLSGA is not sensitive to introduced hyper-parameters. Furthermore, the best values for each of them have been provided, which with a high confidence can be used for all problems.

4.5 Diversity-first, convergence-second

In this chapter it has been investigated if the region-based search shown by the standard MLSGA is maintained with stronger individual-level mechanisms. For that purpose, eight

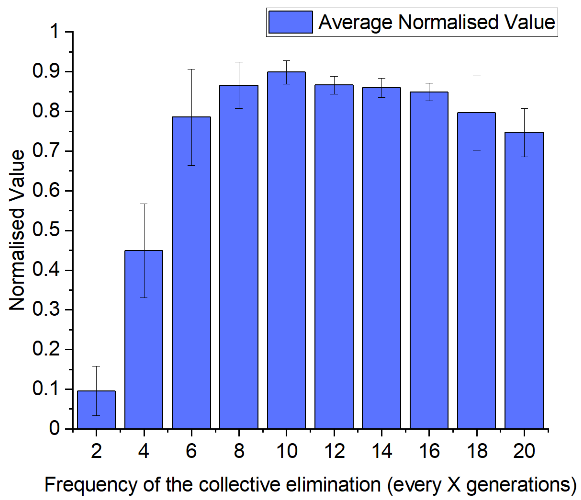


FIGURE 4.7: Comparison of the IGD values obtained by the MLSGA algorithm with different frequencies of the collective elimination.

different algorithms taken from the current state-of-the-art have been used to replace the individual reproduction mechanisms within the MLSGA, resulting in the hybrid approach.

The presented results have shown that the MLSGA-hybrids are strongly outperforming the original MLSGA on almost all of the tested cases. Furthermore, the improvement over most of the implemented algorithms can be observed, especially on cases where "convergence-first" based approaches are ineffective, such as constrained, biased and discontinuous problems. In proposed hybrid approach, the stronger individual-level reproduction leads to a better exploitation of both objective and variable spaces; and to a higher diversity preservation inside each group, as the collectives are local versions of the original algorithms. Furthermore, similarly to the standard MLSGA, the collective-level mechanisms strongly penalize certain regions of the objective space for each group, while the fitness separation leads to each group having a different favoured region. That leads to better spread of the sub-populations and thus a higher global diversity. According to that it is concluded that the unique region-based search of the MLSGA is retained in the hybrid approach.

It is here suggested that the high performance on cases where a high diversity is needed is the result of three factors: sub-population approach that decreases the chances of premature convergence on a local optima or infeasible regions, as typical behaviour for the algorithms with distinct sub-populations [46]; collective reproduction that introduces the additional selection pressure and removes the groups that are stuck in "bad" regions; and the fitness separation that diversify the collectives and thus enhance the ability to explore different parts of the objective space. Therefore, MLSGA is able to move around "gaps" more easily in the objective space and thus maintain a higher diversity. This claim is supported by high gains in performance shown by the MLSGA-MOEA/D hybrid on constrained problems. Original MOEA/D struggled to operate in non-continuous search spaces due to defining the weights in straight lines, which potentially may pass through infeasible regions, leading to an inefficient search.

However, the performance of MLSGA-hybrids is often lower than implemented algorithms on continuous and unconstrained problems, especially for the MOEA/D and MTS based variants. It is due to the fact that these problems require a high convergence of the search rather than diversity. Therefore, implementation of additional diversity-preservation mechanisms is disadvantageous and leads to lowered convergence, as there is no free lunch after all. Furthermore, the low performance of MOEA/D and MTS based hybrids is due to incompatibility of the utilised mechanisms, as discussed before. In the MTS, due to high demand for the fitness evaluation of the individual-level, the collective elimination do not occur a sufficiently number of times in order to provide any benefits; whereas utilisation of multiple sub-populations leads to exploration of the same regions repeatedly, leading to an inefficient search. In the MOEA/D based variant, the weigh-vectors used for the individual-level search are forcing exploitation of different regions than the collective-level mechanisms. Due to that, the individuals are not malleable to the evolutionary pressure propagated by collectives, and thus potentially “good” solutions are often eliminated. This indicates that MLSGA is moving from the “convergence-first” approach to the “diversity-first, convergence-second”, as even the more converged individuals are eliminated unless they follow the additional evolutionary pressure developed by the collective. This is further supported by a high performance on the diversity-demanding problems and a low effectiveness on the others.

Importantly, as the MLSGA-hybrid is utilising individual-level mechanisms taken from another algorithms, it is still highly bounded by the effectiveness of implemented methodology. As shown here by low performance of the MLSGA-hybrids on cases where the implemented algorithm is also achieving poor results. Therefore, the collective-level mechanism may provide a better global diversity, but if the individuals are not able to properly converge on the Pareto optimal front, or to maintain in-group diversity, the overall performance will remain low. According to that, it is concluded that, despite not showing a sensitivity to the hyper-parameters, the suggested approach do not provide any benefits regarding the generality of methodology, and thus its usefulness will remain low on the cases with limited knowledge about the problem's characteristics.

Chapter 5 Co-evolutionary approach for the MLSGA

In this chapter it is investigated if the generality of the MLSGA can be improved, by implementation of multiple distinct search strategies on the individual-level of the MLSGA, in parallel using different sub-populations, in a comparable manner to the co-evolutionary approaches. Therefore, improving its capability as the general-solver type algorithm. This will potentially lead to an increase in the overall robustness of the methodology also, as there is a lower chance that multiple search strategies will be ineffective on a particular problem. Simultaneously, MLSGA's group-level mechanisms will maintain the information exchange, regarding potential candidate solutions, between groups and will lead to a better overall diversity, as shown in the previous chapters.

5.1 Methodology

5.1.1 cMLSGA

The co-evolutionary based variant of MLSGA, denoted as cMLSGA, works as detailed below, with the changes regarding to the MLSGA-hybrid given in red text colour. Furthermore, simplified flowchart of the proposed methodology is presented for the illustrative purposes in Fig. 5.1, where darker circles indicate fitter individuals whereas darker yellow rectangles indicate a higher fitness of the collective. Only two collectives are shown for the illustrative purposes of using two distinct evolutionary algorithms. In the cMLSGA multiple collectives of each type are used.

Inputs:

- **Multi-objective problem;**
- **Np: Population size;**
- **Nc: Number of collectives;**
- **Fc: Frequency of collective elimination;**

- **Fitness function definitions for each level;**
- **Evolutionary algorithm 1 (EA1) e.g. MTS;**
- **Evolutionary algorithm 2 (EA2) e.g. MOEA/D;**
- **Individual-level specific parameters for EA1 and EA2;**
- **Stopping criterion;**

Output: External non-dominated Population (EP)

Step 1) Initialisation:

Step 1.1) Set $EP = NULL$.

Step 1.2) Randomly generate the initial population P of N_p individuals $\{x_j, \ldots, x_{Np}\}$.

Step 2) Classification:

Step 2.1) Classify the individuals from the initial population P into N_c collectives, $\{C_i, \ldots, C_{Nc}\}$, so that each contains a separate population $\{P_i, \ldots, P_{Nc}\}$. Classification is based on the decision variable space.

Step 2.2) Assign the EA1 to first $N_c/2$ collectives and EA2 to the rest.

Step 2.3) Assign the fitness definitions from MLS types {MLS1, MLS2, MLS2R} to each collective in the following order $MLS1 \rightarrow MLS2 \rightarrow MLS2R \rightarrow MLS1 \ldots etc.$.

Step 2.4) Perform the individual-level assignment specific to the implemented algorithm, separately for each collective, according to assigned evolutionary algorithm.

Step 3) Individual-level operations:

$For \quad i = 1, \ldots, N_c \quad do$

Step 3.1) Individual-level GA's operations: Perform the reproduction, improvement and update steps over the sub-population P_i according to assigned strategy {either EA1 or EA2}

Step 3.2) Update the external population:

$For \quad j = 1, \ldots, |P_i| \quad do$

Remove from the EP all solutions dominated by x_{ij} (the individual j, from population i). Add the individual x_{ij} to the EP if no solutions from EP dominate the x_{ij}.

Step 4) Collective-level operations:

Step 4.1) Calculate collective's fitness:

$For \quad i = 1, \ldots, N_c \quad do$

Calculate fitness of the collective C_i as an average fitness of the population P_i based on the collective-level fitness definition.

Step 4.2) Collective elimination:

Find the collective C_i with the worst fitness value, and store the index of that collective, z.

Store the size of the eliminated collective $|P_z|$ as the variable s.

Erase the sub-population P_z from the eliminated collective C_z.

Step 4.3) Collective reproduction:

$For \quad i = 1, \ldots, N_c \quad do$

$if \quad (i \quad != \quad z)$

Copy the best $\frac{s}{(N_c-1)}$ individuals, according to the collective-level fitness definition, from population P_i to P_z.

$then$

Copy all individual-level parameters, specific to the assigned evolutionary algorithm, from the eliminated collective C_z to new collective, e.g. In the case of MOEA/D strategy: assign the weight vector λ_z randomly to population P_z.

Step 5) Termination: If the stopping criteria are met, stop and give EP as an output. Otherwise, return to **Step 3)**.

It can be seen that general methodology of the cMLSGA is not significantly different to the MLSGA or MLSGA-hybrid. Only practical differences are in the individual-level reproduction mechanisms, where different strategies are applied to each distinct collective in parallel, Steps 2.2-2.4. In this case, first half of the collectives utilise the selected "Evolutionary Algorithm 1" (EA1) subroutine, and the others "Evolutionary Algorithm 2" (EA2). The type of evolution is assigned randomly to collectives during the classification step and does not change over the run. At any given time, half of the collectives are using EA1 and the other half use EA2. Therefore, it is recommended that the number of collectives should be even, in order to maintain even spread of search and to avoid the bias towards a single strategy. The individuals in each sub-population evolve separately using the assigned strategy. After a predefined number of generations, the collective with the worst fitness value is eliminated, with all of the individuals inside of it, and is repopulated by copying the best individuals from among rest of the collectives. After the collective reproduction step offspring collective inherits the individual reproduction methodology from the eliminated collective. It is demonstrated further in Fig. 5.1, where the left collective retains MTS strategy, when the right ones continues to use MOEA/D. Therefore, cMLSGA utilises the multi-level approach to generate a distinct competitive co-evolutionary approach, but this occurs via elimination and repopulation of one group by other sub-populations, rather than migration of the individuals [103]; competition in each group [99]; or recreation of all groups [36]. In addition, unlike in other co-evolutionary GAs, the sub-populations are not allowed to exchange information every generation but are doing so only during the collective reproduction steps. This will potentially lead to a high diversity of the search and a lack of bias towards specific problem types. Furthermore, the same rules as in MLSGA-hybrid are applied regarding

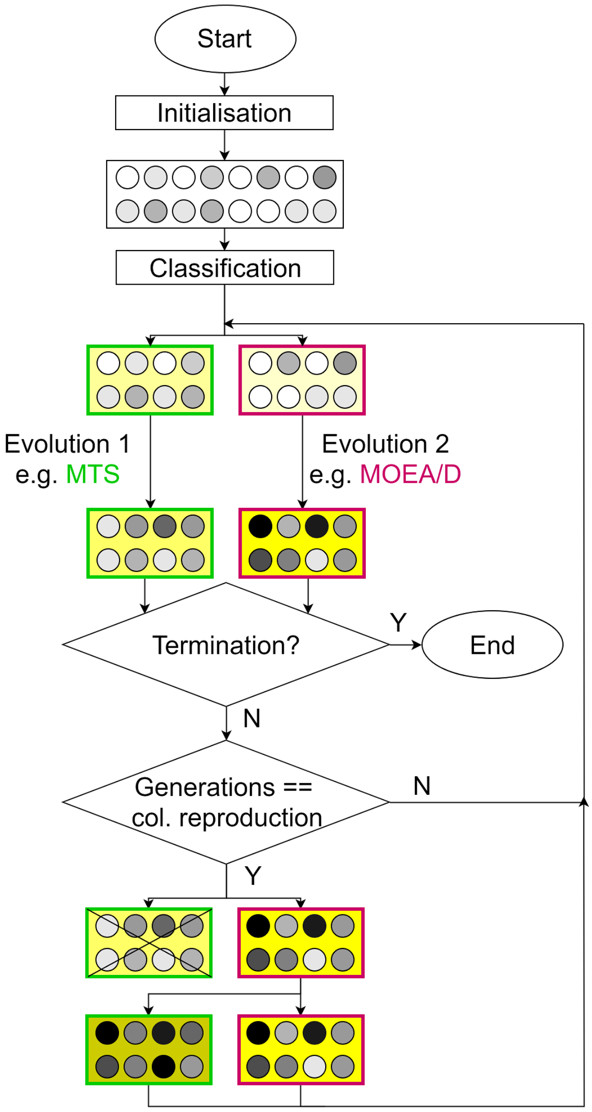


FIGURE 5.1: Simplified flowchart of the cMLSGA methodology.

the hard-coded individual-assigned parameters such as weight vectors in MOEA/D. In this chapter, a variety of reproduction mechanisms are replicated from the original papers and combined within the MLSGA methodology. A total of eight algorithms taken from the current state-of-the-art are utilised: U-NSGA-III [34], MOEA/D [35], MOEA/D-MSF and MOEA/D-PSF [83], MTS [93], IBEA [121], BCE [103] and HEIA [36]. These GAs are the same as in Chapter 4 and are selected due to the same reasons. Furthermore, all combinations of utilised mechanisms are evaluated. However, the combinations between U-NSGA-III, MOEA/D-TCH, MOEA/D-PSF and MOEA/D-MSF have been discarded. This is because all of the MOEA/D variants are based on the same framework while U-NSGA-III and MOEA/D are both based on the constant front development. The pre-benchmarks demonstrate that utilisation of similar mechanisms result in no performance gains within the co-evolutionary approach, The results for those variants are not reported in this work. Therefore, 22 variants in total are evaluated, but only the most significant findings are discussed in this thesis.

5.1.2 Hyper-parameters setting and benchmarking

Similarly, to the MLSGA-hybrid approach, the individual-level reproduction mechanisms are directly copied from the original algorithms and the same setting of corresponding hyper-parameters as in previous tests is used. Exact values of those parameters can be found in Appendix E.

The MLSGA-specific hyper-parameters remains the same as in previous benchmarks, as detailed in Table 5.1. Furthermore, the testing follows the same number of runs, total fitness evaluations and utilised performance indicators or statistical analysis, in order to maintain fair comparison.

TABLE 5.1: Hyper-parameters utilised in the cMLSGA for benchmarking.

Parameter	Value
Population Size	1000
Crossover rate	1
Mutation rate	0.08
Number of collectives	8
Collective reproduction delay	10

5.1.3 Computational complexity and constraint handling

Computational cost of the cMLSGA remains the same as for the MLSGA-hybrid. However, in this case two types of individual reproduction are introduced, which are sharing the same complexity as the original algorithms, denoted here as C_1 and C_2. Therefore, the overall computational complexity of one generation is bounded by C_1, C_2 or $O(mN^2)$ whichever is higher.

The constraint handling strategy remains unchanged.

5.1.4 Extending the test set

In order to rigorously evaluate the generality of developed methodology, the utilised test set has to be as diverse, in regard to the problems' characteristic, as possible. According to that, the test set is further extended with nine two-objectives WFG functions [141]. Resulting in 60 two-objective problems in total, with 28 constrained and 32 unconstrained cases. All benchmarking problems are divided here into twelve categories, according to the dominative characteristic and the presence of constraints, as summarised in Table 5.2.

Table 5.2: Summary of the utilised test set for benchmarking of the cMLSGA, with separation into distinct categories according to the main characteristic of a given problem.

Category	**Problem**	**d**	**Additional properties**
Unconstrained			
I. Simple	ZDT1	30	Convex
	ZDT2	30	Concave
	ZDT3	30	Discontinuous
	ZDT4	10	Multimodal, Convex
	ZDT6	10	Multimodal, Biased, Concave
II. Convex	UF1	30	Complex PS
	UF2	30	Complex PS
	UF3	30	Complex PS
III. Concave	UF4	30	Complex PS
	WFG4	22	Multimodal
	WFG5	22	Deceptive
	WFG6	22	Non-separable
	WFG7	22	Biased
	WFG8	22	Biased, Non-separable
	WFG9	22	Biased, Non-separable, Deceptive
IV. Linear/Mixed	UF7	30	Complex PS, Linear
	WFG1	22	Biased, Mixed
	WFG3	22	Non-separable, Degenerated, Linear
V. Discontinuous	UF5	30	Linear, Distinct points, Complex PS
	UF6	30	Complex PS
	WFG2	22	Convex, Non-Separable
	MOP4	10	Discontinuous
VI. Imbalanced	MOP1	10	Convex
	MOP2	10	Convex
	MOP3	10	Concave
	MOP5	10	Convex
	IMB1	10	Convex
	IMB2	10	Linear
	IMB3	10	Concave
	IMB7	10	Convex, Non-separable
	IMB8	10	Linear, Non-separable
	IMB9	10	Concave, Non-separable
Constrained			
VII. Discontinuous	CF1	10	Linear, Complex PS, Distinct points
	CF2	10	Convex, Complex PS
	CF3	10	Concave, Complex PS
VIII. Continuous	CF4	10	Linear, Complex PS
	CF5	10	Linear, Complex PS
	CF6	10	Mixed, Complex PS
	CF7	10	Mixed, Complex PS

TABLE 5.2: Summary of the utilised test set for benchmarking of the cMLSGA (continued)

Category	Problem	d	Additional properties
IX. Imbalanced	IMB11	10	Convex
	IMB12	10	Linear
	IMB13	10	Concave
X. Diversity-hard	DAS-CMOP1(5)	30	Concave, Discontinuous
	DAS-CMOP2(5)	30	Mixed, Continuous
	DAS-CMOP3(5)	30	Linear, Discontinuous, Multimodal
	DAS-CMOP4(5)	30	Concave, Discontinuous
	DAS-CMOP5(5)	30	Mixed, Discontinuous
	DAS-CMOP6(5)	30	Distinct points, Degenerated
XI. Feasibility-hard	DAS-CMOP1(6)	30	Concave, Discontinuous
	DAS-CMOP2(6)	30	Mixed, Continuous
	DAS-CMOP3(6)	30	Linear, Discontinuous, Multimodal
	DAS-CMOP4(6)	30	Concave, Discontinuous
	DAS-CMOP5(6))	30	Mixed, Discontinuous
	DAS-CMOP6(6)	30	Distinct points, Degenerated
XII. Convergence-hard	DAS-CMOP1(7)	30	Concave, Discontinuous
	DAS-CMOP2(7)	30	Mixed, Continuous
	DAS-CMOP3(7))	30	Linear, Discontinuous, Multimodal
	DAS-CMOP4(7)	30	Concave, Discontinuous
	DAS-CMOP5(7)	30	Mixed, Discontinuous
	DAS-CMOP6(7)	30	Distinct points, Degenerated

d denotes the number of decision variables.

5.2 Comparison to the hybrid variant and implemented algorithms

In order to verify if the cMLSGA is able to successfully combine the advantages of both implemented algorithms, leading to a more general approach, as well as the high diversity shown by previous versions, the performance of cMLSGA variants is compared to the original algorithms. Out of 28 developed variants, 3 cases are selected as representatives due to following reasons: cMLSGA$_{\text{IBEA_BCE}}$ as the case where the highest improvement in average over the implemented algorithms is shown; cMLSGA$_{\text{MOEA/D-MSF_HEIA}}$, as the combination of the best specialist-solver, MOEA/D-MSF, with the best currently existing general methodology, HEIA, resulting in the highest generality from among all tested variants; and cMLSGA$_{\text{MTS_MOEA/D}}$, which is the combination of two convergence-based algorithms, specialist-solver, MOEA/D, with a local search based method, MTS, as the model case for showing the risk of incompatibility between implemented mechanisms. The results according to both performance indicators are presented in Table 5.3 for cMLSGA$_{\text{IBEA_BCE}}$, in Table 5.4 for cMLSGA$_{\text{MOEA/D-MSF_HEIA}}$ and in Table 5.5 for cMLSGA$_{\text{MTS_MOEA/D}}$. The best variant in each group is in **bold** and results of the cMLSGA are highlighted: in green if cMLSGA is better than both of the implemented algorithms; in yellow if cMLSGA is better than one of them; and in red if it is worse than both of the methodologies.

TABLE 5.3: The average IGD and HV scores of the cMLSGA_IBEA_BCE compared to the implemented algorithms

Cat.	Case	IGD			HV		
		cMLSGA	IBEA	BCE	cMLSGA	IBEA	BCE
I	ZDT1	**7.45E-04**	1.48E-03	3.27E-02	**0.91638**	0.91570	0.89683
	ZDT2	**7.35E-04**	2.45E-03	1.09E-01	**0.83305**	0.83265	0.75229
	ZDT3	**8.85E-04**	6.84E-03	1.58E-02	**1.18970**	1.18880	1.17395
	ZDT4	**1.60E-03**	6.26E-01	2.35E+0	**0.91572**	0.60542	0.00000
	ZDT6	**2.15E-03**	3.05E-03	2.49E-03	0.75610	**0.75639**	0.75620
II	UF1	**2.14E-02**	9.70E-02	7.26E-02	**0.90235**	0.85355	0.85855
	UF2	**2.42E-02**	4.25E-02	3.91E-02	**0.89605**	0.87601	0.89058
	UF3	**1.01E-01**	2.25E-01	1.48E-01	0.85258	0.70082	**0.85755**
III	UF4	**4.94E-02**	5.40E-02	5.91E-02	**0.80240**	0.79986	0.79210
	WFG4	3.00E-02	**2.45E-02**	8.50E-02	0.56737	**0.57061**	0.54136
	WFG5	**6.52E-02**	6.84E-02	6.76E-02	**0.54747**	0.54321	0.54446
	WFG6	5.34E-02	6.15E-02	**4.80E-02**	0.55725	0.55247	**0.55889**
	WFG7	1.79E-02	**1.46E-02**	2.00E-02	0.57361	**0.57441**	0.57188
	WFG8	1.11E-01	1.18E-01	**1.04E-01**	0.53088	0.53072	**0.53196**
	WFG9	**3.17E-02**	8.56E-02	6.75E-02	**0.55521**	0.53294	0.54001
IV	UF7	**1.37E-02**	2.75E-01	3.18E-02	**0.86194**	0.67920	0.84339
	WFG1	1.03E+0	**5.78E-01**	1.25E+0	0.41956	**0.57345**	0.36617
	WFG3	2.70E-02	**1.80E-02**	3.63E-02	0.72093	**0.72393**	0.71137
V	UF5	**3.17E-01**	3.75E-01	1.47E+0	**0.59333**	0.57950	0.07095
	UF6	**1.12E-01**	2.18E-01	5.09E-01	**0.71427**	0.61143	0.49510
	WFG2	**1.16E-02**	2.07E-02	4.61E-02	0.75807	**0.76118**	0.74217
	MOP4	3.03E-01	2.99E-01	**2.62E-01**	0.76362	0.74233	**0.79537**
VI	MOP1	**1.83E-01**	1.86E-01	1.84E-01	**0.84482**	0.84260	0.84404
	MOP2	3.71E-01	3.55E-01	**3.33E-01**	0.73532	0.74724	**0.75213**
	MOP3	**4.09E-01**	4.08E-01	4.84E-01	**0.74999**	0.74919	0.69886
	MOP5	2.74E-01	**2.36E-01**	2.96E-01	0.78560	**0.80898**	0.73506
	IMB1	**2.11E-01**	3.37E-01	2.23E-01	**0.84770**	0.76956	0.84315
	IMB2	**1.84E-01**	2.24E-01	1.97E-01	**0.73357**	0.68093	0.71625
	IMB3	**2.54E-01**	2.97E-01	2.66E-01	**0.66016**	0.53266	0.62487
	IMB7	4.50E-03	**3.15E-03**	2.31E-02	0.91487	**0.91529**	0.90752
	IMB8	4.33E-03	**3.45E-03**	4.39E-02	0.87317	**0.87349**	0.85675
	IMB9	**3.81E-03**	5.53E-03	1.41E-01	0.80203	**0.80238**	0.73702
VII	CF1	**4.58E-03**	1.52E-01	1.20E-02	**0.86708**	0.73259	0.86423
	CF2	**7.52E-03**	6.08E-02	2.01E-02	**0.89599**	0.83738	0.89007
	CF3	**1.98E-01**	6.02E-01	5.63E-01	**0.68204**	0.45756	0.40027
VIII	CF4	**4.38E-02**	3.09E-01	9.65E-02	**0.79853**	0.62764	0.77658
	CF5	**1.06E-01**	4.71E-01	5.59E-01	**0.72568**	0.50870	0.43336
	CF6	**5.30E-02**	1.41E-01	9.38E-02	**0.86700**	0.81273	0.83949
	CF7	**1.55E-01**	2.55E+0	4.43E-01	**0.78466**	0.37303	0.57368
IX	IMB11	**9.59E-02**	1.69E-01	9.83E-02	**0.84317**	0.80606	0.84090
	IMB12	6.01E-02	2.52E-01	**5.88E-02**	0.76028	0.65527	**0.76208**
	IMB13	6.62E-02	2.96E-01	**6.48E-02**	0.67945	0.55772	**0.68069**
X	DAS_CMOP1(5)	**1.04E-01**	4.17E-01	4.03E-01	**0.72380**	0.45716	0.43082
	DAS_CMOP2(5)	**5.42E-02**	3.46E-01	3.54E-01	**0.87928**	0.63960	0.62808
	DAS_CMOP3(5)	**1.67E-01**	4.84E-01	5.36E-01	**0.75555**	0.61519	0.48258
	DAS_CMOP4(5)	**8.17E-01**	9.29E-01	1.88E+01	0.30318	**0.30629**	0.00000
	DAS_CMOP5(5)	1.13E+00	**9.02E-01**	2.05E+01	0.24288	**0.38613**	0.00000
	DAS_CMOP6(5)	**1.09E+00**	1.19E+00	2.05E+01	**0.19313**	0.16450	0.00000

TABLE 5.3: The average IGD and HV scores of the cMLSGA_IBEA_BCE compared to the implemented algorithms (continued)

Cat.	Case	IGD			HV		
		cMLSGA	IBEA	BCE	cMLSGA	IBEA	BCE
XI	DAS_CMOP1(6)	**6.35E-01**	7.96E-01	7.49E-01	**0.88482**	0.55666	0.52396
	DAS_CMOP2(6)	**6.50E-01**	8.33E-01	8.13E-01	**0.93940**	0.71409	0.69836
	DAS_CMOP3(6)	**6.33E-01**	7.00E-01	7.00E-01	**0.89873**	0.70911	0.73329
	DAS_CMOP4(6)	**1.33E+13**	5.33E+13	1.00E+14	**0.35899**	0.29262	0.00000
	DAS_CMOP5(6)	**4.00E+13**	7.33E+13	1.00E+14	**0.34421**	0.21398	0.00000
	DAS_CMOP6(6)	**2.67E+13**	6.67E+13	1.00E+14	**0.30961**	0.19123	0.00000
XII	DAS_CMOP1(7)	**2.33E-01**	6.13E-01	6.07E-01	**0.71547**	0.30351	0.28520
	DAS_CMOP2(7)	**3.48E-02**	3.56E-01	8.08E-01	**0.90529**	0.65082	0.30789
	DAS_CMOP3(7)	**1.90E-01**	2.93E-01	7.05E-01	**0.70016**	0.64477	0.36318
	DAS_CMOP4(7)	**1.17E+00**	1.19E+00	2.00E+01	0.12651	**0.13428**	0.00000
	DAS_CMOP5(7)	1.31E+00	**1.26E+00**	1.93E+01	0.18589	**0.21248**	0.00000
	DAS_CMOP6(7)	**1.24E+00**	1.60E+00	1.82E+01	**0.13717**	0.08671	0.00000

The best variant in each group is in ***bold****. Results of the cMLSGA are highlighted: in green, if cMLSGA is better than both of the implemented algorithms; in yellow, if cMLSGA is better than one of them; and in red, if it is worse than both of the algorithms.*

TABLE 5.4: The average IGD and HV scores of cMLSGA_MOEA/D-MSF_HEIA compared to the implemented algorithms

Cat.	Case	IGD			HV		
		cMLSGA	MOEA/D-MSF	HEIA	cMLSGA	MOEA/D-MSF	HEIA
I	ZDT1	**4.81E-04**	7.26E-04	6.74E-04	**0.91648**	0.91621	0.91643
	ZDT2	**4.34E-04**	7.13E-04	6.76E-04	**0.83314**	0.83296	0.83301
	ZDT3	**5.28E-04**	1.84E-03	8.05E-04	**1.18979**	1.18906	1.18976
	ZDT4	**4.65E-04**	7.28E-04	6.57E-04	**0.91649**	0.91612	0.91644
	ZDT6	**4.33E-04**	7.99E-04	5.28E-04	**0.75711**	0.75696	0.75710
II	UF1	1.89E-02	4.32E-03	**1.62E-03**	0.90785	0.91301	**0.91597**
	UF2	1.40E-02	1.05E-02	**3.15E-03**	0.90855	0.90987	**0.91544**
	UF3	7.23E-02	2.66E-02	**2.37E-02**	0.88339	0.89955	**0.90493**
III	UF4	4.17E-02	6.63E-02	**3.52E-02**	0.80741	0.78548	**0.81081**
	WFG4	8.52E-03	8.62E-02	**3.21E-03**	0.57795	0.55025	**0.57981**
	WFG5	**6.34E-02**	6.57E-02	6.35E-02	**0.55025**	0.54748	0.54985
	WFG6	**3.01E-02**	1.02E-01	3.92E-02	**0.56786**	0.53709	0.56398
	WFG7	3.98E-03	1.47E-02	**2.50E-03**	0.57970	0.57482	**0.58046**
	WFG8	8.02E-02	8.99E-02	**7.06E-02**	0.54610	0.53890	**0.54762**
	WFG9	**1.77E-02**	1.05E-01	7.83E-02	**0.56122**	0.52576	0.53674
IV	UF7	1.41E-02	8.55E-03	**1.81E-03**	0.86814	0.86754	**0.87403**
	WFG1	7.27E-01	1.14E+0	**3.16E-01**	0.52646	0.38100	**0.66579**
	WFG3	7.56E-03	1.27E-02	**2.99E-03**	0.73018	0.72817	**0.73210**
V	UF5	2.93E-01	4.61E-01	**1.90E-01**	0.63600	0.54374	**0.70573**
	UF6	**8.56E-02**	1.67E-01	1.32E-01	**0.75881**	0.71955	0.70494
	WFG2	4.45E-03	1.52E-02	**1.80E-03**	0.76362	0.75897	**0.76452**
	MOP4	2.26E-01	**2.32E-02**	2.81E-01	0.80538	**0.87123**	0.78028

TABLE 5.4: The average IGD and HV scores of cMLSGA_MOEA/D-MSF_HEIA compared to the implemented algorithms (continued)

Cat.	**Case**	**IGD**			**HV**		
		cMLSGA	**MOEA/D-MSF**	**HEIA**	**cMLSGA**	**MOEA/D-MSF**	**HEIA**
VI	MOP1	1.79E-01	**1.72E-02**	1.82E-01	0.84711	**0.91069**	0.84523
	MOP2	2.40E-01	**1.42E-02**	3.41E-01	0.77050	**0.82866**	0.75270
	MOP3	4.27E-01	**2.01E-02**	4.72E-01	0.73437	**0.79751**	0.70537
	MOP5	2.04E-01	**1.48E-02**	2.55E-01	0.83087	**0.91067**	0.78768
	IMB1	2.21E-02	**6.22E-03**	9.67E-03	0.90905	**0.91453**	0.91336
	IMB2	1.19E-01	**3.86E-02**	1.35E-01	0.78364	**0.85944**	0.76757
	IMB3	2.39E-01	**1.08E-02**	2.84E-01	0.65354	**0.79939**	0.56865
	IMB7	9.72E-04	1.52E-03	**7.26E-04**	0.91625	0.91589	**0.91637**
	IMB8	9.01E-04	9.42E-04	**7.12E-04**	0.87458	0.87440	**0.87468**
	IMB9	**5.30E-04**	2.67E-03	7.20E-04	**0.80332**	0.80241	0.80316
VII	CF1	1.62E-03	1.73E-01	**3.51E-04**	0.86815	0.77201	**0.86862**
	CF2	**4.13E-04**	1.24E-03	7.98E-04	**0.90261**	0.89100	0.88885
	CF3	**1.11E-01**	2.20E-01	1.72E-01	**0.70583**	0.66571	0.68887
VIII	CF4	1.49E-02	1.41E-01	**8.23E-03**	**0.82634**	0.71796	0.82509
	CF5	**7.33E-02**	2.12E-01	9.34E-02	**0.75611**	0.64077	0.72727
	CF6	3.92E-02	1.38E-01	**3.61E-02**	**0.87193**	0.83261	0.86652
	CF7	**8.97E-02**	3.25E+0	1.25E-01	**0.81267**	0.48507	0.77747
IX	IMB11	7.84E-02	**4.79E-02**	9.31E-02	0.85975	**0.90106**	0.84804
	IMB12	1.81E-02	**9.20E-04**	1.33E-03	0.86214	**0.87456**	0.87433
	IMB13	**1.02E-02**	2.99E-02	1.23E-02	**0.84119**	0.82414	0.80209
X	DAS_CMOP1(5)	3.52E-02	**3.29E-02**	3.40E-01	0.79153	**0.79417**	0.48040
	DAS_CMOP2(5)	3.66E-02	**3.30E-02**	3.31E-01	0.89464	**0.89653**	0.65248
	DAS_CMOP3(5)	9.03E-02	**6.12E-02**	2.58E-01	0.81793	**0.83660**	0.72256
	DAS_CMOP4(5)	**7.17E-01**	4.62E+00	1.16E+00	**0.35384**	0.00000	0.19335
	DAS_CMOP5(5)	1.01E+00	4.88E+00	**9.79E-01**	0.32504	0.00000	**0.35876**
	DAS_CMOP6(5)	**7.59E-01**	4.79E+00	1.25E+00	**0.39999**	0.00000	0.21723
XI	DAS_CMOP1(6)	**6.35E-01**	6.39E-01	6.39E-01	0.88485	0.88486	**0.88545**
	DAS_CMOP2(6)	**6.51E-01**	6.56E-01	6.54E-01	0.93947	0.93955	**0.94001**
	DAS_CMOP3(6)	**6.13E-01**	6.14E-01	7.06E-01	**0.92374**	0.92365	0.83408
	DAS_CMOP4(6)	**4.00E+13**	1.00E+14	4.67E+13	**0.32820**	0.00000	0.29532
	DAS_CMOP5(6)	**2.00E+13**	1.00E+14	3.33E+13	**0.53235**	0.00000	0.41944
	DAS_CMOP6(6)	**2.67E+13**	1.00E+14	7.33E+13	**0.35028**	0.00000	0.17495
XII	DAS_CMOP1(7)	2.38E-01	**1.46E-01**	5.62E-01	0.76396	**0.78035**	0.34326
	DAS_CMOP2(7)	3.54E-02	**3.28E-02**	5.94E-01	0.90513	**0.90588**	0.47925
	DAS_CMOP3(7)	1.90E-01	**2.34E-02**	7.90E-01	0.71247	**0.88098**	0.33638
	DAS_CMOP4(7)	**1.87E+00**	4.52E+00	1.93E+00	0.00266	0.00000	**0.00267**
	DAS_CMOP5(7)	2.13E+00	4.47E+00	**2.13E+00**	0.00280	0.00000	**0.00286**
	DAS_CMOP6(7)	**1.99E+00**	4.58E+00	2.07E+00	**0.01013**	0.00000	0.00262

The best variant in each group is in ***bold****. Results of the cMLSGA are highlighted: in green, if cMLSGA is better than both of the implemented algorithms; in yellow, if cMLSGA is better than one of them; and in red, if it is worse than both of the algorithms.*

TABLE 5.5: The average IGD and HV scores of cMLSGA_MTS_MOEA/D compared to the implemented algorithms

Cat.	Case	IGD			HV		
		cMLSGA	**MTS**	**MOEA/D**	**cMLSGA**	**MTS**	**MOEA/D**
I	ZDT1	1.49E-02	1.50E-02	**6.95E-04**	0.90642	0.90648	**0.91629**
	ZDT2	2.70E-02	2.66E-02	**6.39E-04**	0.80511	0.80495	**0.83302**
	ZDT3	**8.45E-03**	8.63E-03	1.72E-03	1.17946	1.17933	**1.18955**
	ZDT4	1.74E-02	6.45E-02	**6.43E-04**	0.90160	0.86694	**0.91640**
	ZDT6	2.02E-02	5.11E-02	**5.75E-04**	0.74401	0.71960	**0.75700**
II	UF1	**1.76E-03**	1.83E-03	2.16E-03	**0.91457**	0.91454	0.91420
	UF2	1.60E-03	**1.59E-03**	8.27E-03	0.91472	**0.91493**	0.91020
	UF3	4.17E-02	4.29E-02	**2.40E-02**	0.85481	0.85081	**0.89004**
III	UF4	2.93E-02	**2.93E-02**	6.43E-02	0.80657	**0.80700**	0.78802
	WFG4	2.11E-02	**1.99E-02**	2.54E-02	0.56038	0.56016	**0.56962**
	WFG5	7.18E-02	7.09E-02	**6.75E-02**	0.54069	0.54193	**0.54441**
	WFG6	**4.01E-02**	4.15E-02	1.04E-01	**0.55727**	0.55495	0.53650
	WFG7	2.22E-02	2.49E-02	**3.14E-03**	0.56348	0.56375	**0.57989**
	WFG8	1.02E-01	1.03E-01	**6.90E-02**	0.52721	0.52668	**0.54799**
	WFG9	6.86E-02	7.31E-02	**1.70E-02**	0.53540	0.53391	**0.56120**
IV	UF7	3.05E-02	3.21E-02	**3.22E-03**	0.84069	0.83744	**0.87110**
	WFG1	**1.06E+0**	1.08E+0	1.07E+0	0.33444	0.33176	**0.41342**
	WFG3	1.39E-01	1.39E-01	**4.02E-03**	0.66027	0.66103	**0.73139**
V	UF5	**1.13E-01**	1.14E-01	3.90E-01	**0.78097**	0.77529	0.58409
	UF6	6.42E-02	**6.33E-02**	1.66E-01	**0.75534**	0.75062	0.72652
	WFG2	1.28E-01	1.37E-01	**5.86E-03**	0.68146	0.67696	**0.76335**
	MOP4	7.78E-02	**6.61E-02**	2.64E-01	**0.86325**	0.86189	0.79243
VI	MOP1	1.34E-01	**1.33E-01**	1.80E-01	**0.86360**	0.86414	0.84632
	MOP2	1.98E-01	**1.77E-01**	3.41E-01	0.77042	**0.77422**	0.75083
	MOP3	1.57E-01	**1.42E-01**	4.86E-01	0.73701	**0.74107**	0.69644
	MOP5	9.03E-02	**8.61E-02**	2.56E-01	0.84970	**0.85370**	0.78177
	IMB1	**3.77E-02**	3.88E-02	9.51E-02	**0.89906**	0.89865	0.88632
	IMB2	**5.06E-02**	5.54E-02	1.16E-01	**0.84835**	0.84624	0.83062
	IMB3	9.00E-02	**8.17E-02**	2.23E-01	0.77281	**0.77416**	0.75966
	IMB7	9.15E-03	1.52E-02	**7.93E-04**	0.91268	0.91014	**0.91629**
	IMB8	1.31E-02	2.16E-02	**6.48E-04**	0.86912	0.86549	**0.87469**
	IMB9	1.96E-02	2.83E-02	**7.70E-04**	0.79528	0.79129	**0.80334**
VII	CF1	**2.00E-02**	2.21E-02	2.14E-01	**0.85298**	0.85014	0.69322
	CF2	1.18E-03	1.05E-03	**9.46E-04**	0.90085	**0.90133**	0.89050
	CF3	**1.07E-01**	1.17E-01	1.76E-01	**0.72171**	0.70842	0.69100
VIII	CF4	6.10E-03	**5.70E-03**	1.50E-01	0.82673	**0.82711**	0.71699
	CF5	**2.29E-02**	2.34E-02	2.19E-01	0.80637	**0.80805**	0.66101
	CF6	5.47E-03	**5.45E-03**	1.33E-01	0.89715	**0.89969**	0.83604
	CF7	**2.10E-02**	2.46E-02	1.28E+0	**0.87248**	0.86858	0.62259
IX	IMB11	1.06E-02	1.04E-01	**9.99E-04**	0.83442	0.83154	**0.91626**
	IMB12	7.52E-02	7.89E-02	**6.28E-04**	0.74653	0.74401	**0.87473**
	IMB13	9.40E-02	9.71E-02	**9.54E-04**	0.66510	0.65753	**0.83291**
X	DAS_CMOP1(5)	8.21E-02	7.82E-02	**3.15E-02**	0.68022	0.69055	**0.79310**
	DAS_CMOP2(5)	7.92E-02	8.12E-02	**3.22E-02**	0.82154	0.82072	**0.89663**
	DAS_CMOP3(5)	1.21E-01	**1.18E-01**	1.30E-01	0.79135	**0.79662**	0.77847
	DAS_CMOP4(5)	**7.48E-01**	7.61E-01	4.50E+00	**0.30650**	0.30629	0.11013
	DAS_CMOP5(5)	**6.77E-01**	7.44E-01	5.53E+00	**0.42840**	0.40439	0.00062
	DAS_CMOP6(5)	**7.43E-01**	7.54E-01	5.57E+00	0.36687	**0.37311**	0.00018

TABLE 5.5: The average IGD and HV scores of cMLSGA_MTS_MOEA/D compared to the implemented algorithms (continued)

Cat.	Case	IGD			HV		
		cMLSGA	**MTS**	**MOEA/D**	**cMLSGA**	**MTS**	**MOEA/D**
XI	DAS_CMOP1(6)	**5.99E-01**	6.00E-01	6.40E-01	0.79520	0.79697	**0.88555**
	DAS_CMOP2(6)	6.63E-01	6.67E-01	**6.56E-01**	0.87287	0.87585	**0.94014**
	DAS_CMOP3(6)	**6.10E-01**	6.15E-01	6.14E-01	0.84304	0.84654	**0.92450**
	DAS_CMOP4(6)	**1.00E-01**	1.17E-01	9.33E+13	0.46347	**0.48105**	0.04359
	DAS_CMOP5(6)	1.16E-01	**1.10E-01**	8.67E+13	0.53289	**0.54365**	0.12385
	DAS_CMOP6(6)	1.54E-01	**1.53E-01**	1.00E+14	**0.49608**	0.48583	0.00000
XII	DAS_CMOP1(7)	5.41E-01	5.37E-01	**1.43E-01**	0.37856	0.38254	**0.78147**
	DAS_CMOP2(7)	6.94E-01	6.53E-01	**3.22E-02**	0.44382	0.46265	**0.90650**
	DAS_CMOP3(7)	6.65E-01	5.90E-01	**2.22E-02**	0.42600	0.46383	**0.88203**
	DAS_CMOP4(7)	1.64E+00	**1.47E+00**	6.65E+00	0.03964	**0.06115**	0.00279
	DAS_CMOP5(7)	**1.45E+00**	1.45E+00	4.95E+00	0.12796	0.12944	**0.15718**
	DAS_CMOP6(7)	1.40E+00	**1.28E+00**	6.94E+00	0.10746	**0.14639**	0.00000

The best variant in each group is in ***bold****. Results of the cMLSGA are highlighted: in green, if cMLSGA is better than both of the implemented algorithms; in yellow, if cMLSGA is better than one of them; and in red, if it is worse than both of the algorithms.*

Comparing the presented cMLSGA variants against the original algorithms, it can be seen that the highest improvement is exhibited by the cMLSGAIBEA_BCE, Table 5.3. In this case the performance is improved for 45 out of 60 problems according to the IGD indicator, and on 39 cases for HV, whereas it is only worse on 2 cases for both indicators. For the most general variant, cMLSGA_MOEA/D-MSF_HEIA, Table 5.4, the better performance is achieved on 25 and 24 cases for the IGD and HV respectively, and it is worse than both methodologies on 6 and 7 cases correspondingly. For the last variant, cMLSGAMTS_MOEA/D , Table 5.5, better results are obtained only on 18 and 14 cases according to the IGD and HV respectively, but worse on 8 and 12 cases. Therefore, it can be seen that the cMLSGA is able to successfully combine distinct search strategies of two methodologies, leading to a better final performance than implementing algorithms, or at least not worse than them on average. However, the final performance is strongly dependent on the types of implemented algorithms. In the case where two methodologies with strong diversity mechanisms are implemented, such as indicator-based IBEA and dominance-based BCE, then cMLSGA is able to successfully combine both methodologies leading to high gains due to better convergence and diversity. Reduced performance of other two presented variants, which combine strong convergence-based mechanisms, may be caused by the inability of MLSGA to properly maintain their search pattern during the collective elimination and reproduction steps, as discussed in the previous chapter. In the case of MTS and MOEA/D based variants, the performance of combined algorithms strongly resembles the results of MTS, especially for the IGD indicator. This is due to the fact that the local searches utilised within MTS “dominate” the search. The local searches require a substantial number of iterations per generation, and thus not much computational effort is left for the secondary mechanisms, such as MLSGA and MOEA/D. According to that, if a method is combined with a significantly stronger convergence-first methodology, such as MTS, the weaker strategy will have negligible impact on the search. Therefore, the potential gains are due to dispersed search, which leads to more uniform results, rather than utilisation of multiple individual-reproduction strategies. Finally, in the case where the specialist convergence-based methodology is combined with a general diversity-based one, the cMLSGA is able to maintain distinct searches for both of them and to maintain their advantages. This indicates, that the cMLSGA provides better results if two methodologies with contrasting mechanisms are combined, rather than when one method is outperforming the other. Therefore, the success in combining methodologies is subject to how similar the methods are. Unsurprisingly, it is very unlikely that the resulting strategy will outperform specialist-solver on its preferred cases or improve the effectiveness on problems where at least one of the implemented algorithms achieve a near-perfect results, e.g. IGD lower than $1 * 10^{-2}$ on UF1-3 cases for MOEA/D-MSF and HEIA. This indicate that these mechanisms are well adjusted to those problems and any changes within them will reduce their effectiveness. After all, making the search less focused, by implementation of additional mechanisms that are created to enhance the overall diversity, will lead to worse overall performance, especially if the algorithm is already able to obtain excellent results. For those cases, even more specialist mechanisms are required for higher gains rather than a general search methodology.

Examining the results for distinct categories of the problems it can be seen that the cMLSGA exhibits high performance on functions requiring a high diversity. It operates significantly better on cases with discontinuous search spaces, categories V, VII and VIII, rather than continuous problems for both sets of objectives; and on diversity- and feasibility-hard functions, categories X and XI. On these problems the cMLSGA IBEA_BCE is better on 21 and 19 out of 23 cases for IGD and HV indicators respectively and worse only in 1 for IGD only; for cMLSGA_MOEA/D-MSF_HEIA it is better on 13 cases for both indicators and worse in 2 cases only for the HV indicator. Similar principle can be observed on imbalanced problems categories VI and IX, but in this case it does not applies to cMLSGA_MOEA/D-MSF_HEIA , as MOEA/D-MSF is specialist-solver for this kind of problems, which is confirmed by the low IGD values of MOEA/D based algorithms, especially in comparison to the other methods. The lower impact on the performance of the continuous functions is caused by the additional region searches provided by cMLSGA which are not necessary as the convergence is more important for these problems.

However, in order to fully investigate the impact of co-evolutionary approach on the generality, the developed cMLSGA variants have to be compared with corresponding MLSGA-hybrids. Similarly, to the previous case the cMLSGAIBEA_BCE and cMLSGA_MOEA/D-MSF_HEIA are selected for that purpose. cMLSGAMTS_MOEA/D is discarded, due to strong bias towards the MTS strategy, thus providing no insight in the matter. The results are presented in Table 5.7 for cMLSGAIBEA_BCE and in Table 5.6 for cMLSGA_MOEA/D-MSF_HEIA . Similarly, to previously presented results, the best variant in each group is in **bold** and Results of the cMLSGA are highlighted according to the same rules: green if cMLSGA is better than both of the corresponding hybrid variants; in yellow if cMLSGA is better than one of them; and in red if it is worse than both of the algorithms.

Table 5.6: The average IGD and HV scores of cMLSGA_MOEA/D-MSF_HEIA compared to the corresponding MLSGA variants

Cat.	Case	IGD			HV		
		cMLSGA	MLSGA		cMLSGA	MLSGA	
			MOEA/D-MSF	HEIA		MOEA/D-MSF	HEIA
I	ZDT1	4.81E-04	2.10E-03	**4.40E-04**	0.91648	0.91547	**0.91650**
	ZDT2	4.34E-04	3.86E-03	**4.31E-04**	0.83314	0.83166	**0.83317**
	ZDT3	5.28E-04	6.62E-02	**5.17E-04**	1.18979	1.09604	**1.18980**
	ZDT4	**4.65E-04**	1.21E+00	4.94E-04	**0.91649**	0.30717	0.91647
	ZDT6	**4.33E-04**	6.51E-03	4.52E-04	0.75711	0.75475	**0.75711**
II	UF1	1.89E-02	3.73E-02	**1.52E-02**	0.90785	0.89149	**0.91050**
	UF2	1.40E-02	2.60E-02	**1.14E-02**	0.90855	0.90037	**0.91132**
	UF3	7.23E-02	1.31E-01	**5.25E-02**	0.88339	0.86490	**0.88641**
III	UF4	**4.17E-02**	7.16E-02	4.34E-02	**0.80741**	0.78430	0.80703
	WFG4	**8.52E-03**	8.80E-02	1.04E-02	**0.57795**	0.54128	0.57701
	WFG5	**6.34E-02**	6.72E-02	6.36E-02	**0.55025**	0.54553	0.54972
	WFG6	3.01E-02	1.04E-01	**2.63E-02**	0.56786	0.53534	**0.56957**
	WFG7	**3.98E-03**	4.97E-02	5.04E-03	**0.57970**	0.55894	0.57917
	WFG8	**8.02E-02**	1.70E-01	9.41E-02	**0.54610**	0.50707	0.53811
	WFG9	**1.77E-02**	4.26E-02	1.79E-02	**0.56122**	0.54995	0.56110
IV	UF7	1.41E-02	2.42E-02	**8.42E-03**	0.86814	0.85822	**0.87127**
	WFG1	**7.27E-01**	1.26E+00	8.88E-01	**0.52646**	0.36203	0.47343
	WFG3	**7.56E-03**	4.62E-02	9.83E-03	**0.73018**	0.71108	0.72920
V	UF5	2.93E-01	6.45E-01	**2.33E-01**	0.63600	0.44456	**0.69318**
	UF6	8.56E-02	4.04E-01	**7.25E-02**	0.75881	0.60583	**0.76839**
	WFG2	**4.45E-03**	7.87E-02	5.71E-03	**0.76362**	0.73169	0.76293
	MOP4	2.26E-01	6.01E-02	**2.24E-01**	0.80538	**0.85250**	0.80629
VI	MOP1	1.79E-01	**8.90E-02**	1.78E-01	0.84711	**0.88199**	0.84786
	MOP2	2.40E-01	**1.92E-01**	2.03E-01	0.77050	0.76884	**0.77853**
	MOP3	4.27E-01	**2.68E-01**	4.16E-01	0.73437	0.71240	**0.74328**
	MOP5	2.04E-01	**6.67E-02**	1.88E-01	0.83087	**0.88720**	0.83233
	IMB1	**2.21E-02**	3.76E-02	2.28E-02	**0.90905**	0.90356	0.90877
	IMB2	1.19E-01	**7.21E-02**	1.65E-01	0.78364	**0.84472**	0.73065
	IMB3	2.39E-01	**6.01E-02**	2.94E-01	0.65354	**0.78041**	0.53764
	IMB7	**9.72E-04**	1.22E-02	1.29E-03	**0.91625**	0.91164	0.91612
	IMB8	**9.01E-04**	1.30E-02	1.16E-03	**0.87458**	0.86946	0.87448
	IMB9	**5.30E-04**	2.60E-02	7.58E-04	**0.80332**	0.79246	0.80318
VII	CF1	**1.62E-03**	4.98E-03	2.76E-03	**0.86815**	0.86589	0.86773
	CF2	**4.13E-04**	5.62E-03	3.76E-03	0.90261	0.89953	**0.90269**
	CF3	1.11E-01	3.84E-01	**9.27E-02**	0.70583	0.60089	**0.74682**
VIII	CF4	1.49E-02	3.54E-02	**1.45E-02**	0.82634	0.80858	**0.82656**
	CF5	7.33E-02	1.67E-01	**5.27E-02**	0.75611	0.65903	**0.78437**
	CF6	3.92E-02	**3.27E-02**	4.80E-02	0.87193	**0.88549**	0.86365
	CF7	8.97E-02	1.85E-01	**8.08E-02**	0.81267	0.72185	**0.83173**
IX	IMB11	7.84E-02	**7.56E-02**	9.40E-02	0.85975	**0.87231**	0.84612
	IMB12	**1.81E-02**	4.40E-02	5.86E-02	**0.86214**	0.82644	0.76173
	IMB13	**1.02E-02**	7.38E-02	6.52E-02	**0.84119**	0.71385	0.68021
X	DAS_CMOP1(5)	**3.52E-02**	3.92E-02	4.75E-02	0.79153	**0.79733**	0.77627
	DAS_CMOP2(5)	**3.66E-02**	3.72E-02	7.94E-02	0.89464	**0.89490**	0.85825
	DAS_CMOP3(5)	9.03E-02	**5.02E-02**	7.94E-02	0.81793	**0.84389**	0.81990
	DAS_CMOP4(5)	**7.17E-01**	4.10E+00	8.44E-01	**0.35384**	0.00041	0.29489
	DAS_CMOP5(5)	1.01E+00	3.72E+00	**9.64E-01**	0.32504	0.00434	**0.33122**
	DAS_CMOP6(5)	**7.59E-01**	4.09E+00	8.48E-01	**0.39999**	0.01427	0.35096

Table 5.6: The average IGD and HV scores of cMLSGA_MOEA/D-MSF_HEIA compared to the corresponding MLSGA variants (continued)

Cat.	Case	IGD			HV		
		cMLSGA	MLSGA		cMLSGA	MLSGA	
			MOEA/D-MSF	HEIA		MOEA/D-MSF	HEIA
XI	DAS_CMOP1(6)	6.35E-01	**6.29E-01**	6.35E-01	**0.88485**	0.88401	0.88484
	DAS_CMOP2(6)	6.51E-01	**6.48E-01**	6.51E-01	0.93947	0.93879	**0.93948**
	DAS_CMOP3(6)	6.13E-01	**6.11E-01**	6.28E-01	**0.92374**	0.92205	0.90834
	DAS_CMOP4(6)	4.00E+13	1.00E+14	**1.33E+13**	0.32820	0.00000	**0.48247**
	DAS_CMOP5(6)	**2.00E+13**	1.00E+14	2.67E+13	**0.53235**	0.00000	0.50688
	DAS_CMOP6(6)	2.67E+13	9.33E+13	**2.67E+13**	0.35028	**0.05943**	0.35880
XII	DAS_CMOP1(7)	2.38E-01	**1.53E-01**	5.72E-01	0.76396	**0.77841**	0.32827
	DAS_CMOP2(7)	**3.54E-02**	3.69E-02	5.22E-01	**0.90513**	0.90439	0.53218
	DAS_CMOP3(7)	1.90E-01	**3.43E-02**	6.73E-01	0.71247	**0.87744**	0.38772
	DAS_CMOP4(7)	**1.87E+00**	3.65E+00	1.88E+00	0.00266	0.00405	**0.00562**
	DAS_CMOP5(7)	2.13E+00	3.50E+00	**2.08E+00**	0.00280	**0.01722**	0.01349
	DAS_CMOP6(7)	**1.99E+00**	3.43E+00	2.04E+00	**0.01013**	0.00020	0.00575

*The best variant in each group is in **bold**. Results of the cMLSGA are highlighted: in green, if cMLSGA is better than both of the implemented algorithms; in yellow, if cMLSGA is better than one of them; and in red, if it is worse than both of the algorithms.*

Table 5.7: The average IGD and HV scores of cMLSGA_IBEA_BCE compared to the corresponding MLSGA variants

Cat.	Case	IGD			HV		
		cMLSGA	MLSGA		cMLSGA	MLSGA	
			IBEA	BCE		IBEA	BCE
I	ZDT1	7.45E-04	**4.23E-04**	1.85E-02	0.91638	**0.91650**	0.90596
	ZDT2	7.35E-04	**4.20E-04**	1.61E-01	0.83305	**0.83316**	0.71085
	ZDT3	8.85E-04	**5.31E-04**	6.20E-02	1.18970	**1.18978**	1.10333
	ZDT4	1.60E-03	**1.29E-03**	2.18E+01	**0.91572**	0.91490	0.00000
	ZDT6	2.15E-03	**1.76E-03**	2.34E-02	**0.75610**	0.75582	0.74902
II	UF1	**2.14E-02**	2.68E-02	8.35E-02	**0.90235**	0.89354	0.86041
	UF2	2.42E-02	**1.89E-02**	4.57E-02	0.89605	**0.89856**	0.88702
	UF3	**1.01E-01**	1.38E-01	2.35E-01	**0.85258**	0.73605	0.82537
III	UF4	**4.94E-02**	5.58E-02	6.36E-02	**0.80240**	0.80153	0.78584
	WFG4	3.00E-02	**2.58E-02**	9.04E-02	0.56737	**0.56963**	0.53838
	WFG5	6.52E-02	**6.40E-02**	6.84E-02	0.54747	**0.54877**	0.54408
	WFG6	5.34E-02	**5.26E-02**	6.47E-02	0.55725	**0.55821**	0.54981
	WFG7	1.79E-02	**1.48E-02**	4.16E-02	0.57361	**0.57429**	0.56170
	WFG8	1.11E-01	**1.06E-01**	1.57E-01	0.53088	**0.53281**	0.50893
	WFG9	3.17E-02	**2.71E-02**	3.93E-02	0.55521	**0.55717**	0.55048
IV	UF7	**1.37E-02**	2.36E-02	9.22E-02	**0.86194**	0.85879	0.81115
	WFG1	1.03E+00	**9.77E-01**	1.26E+00	0.41956	**0.43486**	0.36311
	WFG3	2.70E-02	**2.33E-02**	5.10E-02	0.72093	**0.72229**	0.70894
V	UF5	3.17E-01	**2.38E-01**	1.68E+00	0.59333	**0.65637**	0.03826
	UF6	1.12E-01	**6.09E-02**	4.96E-01	0.71427	**0.77382**	0.49649
	WFG2	1.16E-02	**1.10E-02**	7.68E-02	**0.75807**	0.75665	0.72572
	MOP4	3.03E-01	2.62E-01	**2.30E-01**	0.76362	0.79130	**0.80036**

TABLE 5.7: The average IGD and HV scores of cMLSGA_IBEA_BCE compared to the corresponding MLSGA variants (continued)

Cat.	Case	IGD			HV		
		cMLSGA	MLSGA		cMLSGA	MLSGA	
			IBEA	BCE		IBEA	BCE
VI	MOP1	1.83E-01	**1.82E-01**	1.84E-01	0.84482	**0.84618**	0.84415
	MOP2	3.71E-01	3.55E-01	**3.26E-01**	0.73532	0.74987	**0.75299**
	MOP3	4.09E-01	**4.09E-01**	4.66E-01	0.74999	**0.74999**	0.70705
	MOP5	2.74E-01	**2.20E-01**	2.91E-01	0.78560	**0.82433**	0.76377
	IMB1	**2.11E-01**	2.17E-01	2.21E-01	**0.84770**	0.84553	0.84378
	IMB2	1.84E-01	**1.64E-01**	1.96E-01	0.73357	**0.79904**	0.72184
	IMB3	2.54E-01	2.60E-01	**2.46E-01**	0.66016	0.65853	**0.68427**
	IMB7	4.50E-03	**1.22E-03**	1.42E-01	0.91487	**0.91615**	0.85287
	IMB8	4.33E-03	**1.62E-03**	2.19E-01	0.87317	**0.87429**	0.76823
	IMB9	3.81E-03	**5.76E-04**	4.11E-01	0.80203	**0.80340**	0.57700
VII	CF1	**4.58E-03**	3.36E-02	1.66E-02	**0.86708**	0.83133	0.86263
	CF2	**7.52E-03**	1.41E-02	1.15E-02	**0.89599**	0.88803	0.89572
	CF3	1.98E-01	**1.35E-01**	5.90E-01	0.68204	**0.71412**	0.39782
VIII	CF4	**4.38E-02**	4.64E-02	6.21E-02	**0.79853**	0.79157	0.79578
	CF5	**1.06E-01**	1.10E-01	3.06E-01	0.72568	**0.73214**	0.55742
	CF6	**5.30E-02**	8.50E-02	6.21E-02	**0.86700**	0.84685	0.84730
	CF7	1.55E-01	**1.30E-01**	3.93E-01	0.78466	**0.80366**	0.59798
IX	IMB11	**9.59E-02**	1.26E-01	1.24E-01	**0.84317**	0.82073	0.81618
	IMB12	**6.01E-02**	1.15E-01	7.07E-02	**0.76028**	0.72108	0.74990
	IMB13	**6.62E-02**	1.35E-01	9.51E-02	**0.67945**	0.62224	0.66042
X	DAS_CMOP1(5)	**1.04E-01**	2.93E-01	3.69E-01	**0.72380**	0.49878	0.44898
	DAS_CMOP2(5)	**5.42E-02**	2.69E-01	3.19E-01	**0.87928**	0.69356	0.64737
	DAS_CMOP3(5)	**1.67E-01**	3.73E-01	3.56E-01	**0.75555**	0.69200	0.60317
	DAS_CMOP4(5)	8.17E-01	**5.26E-01**	1.39E+01	0.30318	**0.44712**	0.00000
	DAS_CMOP5(5)	1.13E+00	**6.05E-01**	1.39E+01	0.24288	**0.49548**	0.00000
	DAS_CMOP6(5)	1.09E+00	**6.37E-01**	1.29E+01	0.19313	**0.42770**	0.00000
XI	DAS_CMOP1(6)	**6.35E-01**	7.19E-01	7.43E-01	**0.88482**	0.63547	0.55531
	DAS_CMOP2(6)	**6.50E-01**	7.62E-01	7.85E-01	**0.93940**	0.76739	0.71643
	DAS_CMOP3(6)	**6.33E-01**	7.46E-01	7.18E-01	**0.89873**	0.75389	0.78947
	DAS_CMOP4(6)	1.33E+13	**2.61E-01**	1.00E+14	0.35899	**0.59663**	0.00000
	DAS_CMOP5(6)	4.00E+13	**2.21E-01**	1.00E+14	0.34421	**0.62578**	0.00000
	DAS_CMOP6(6)	2.67E+13	**1.33E+13**	1.00E+14	0.30961	**0.44050**	0.00000
XII	DAS_CMOP1(7)	**2.33E-01**	5.30E-01	5.94E-01	**0.71547**	0.36785	0.29884
	DAS_CMOP2(7)	**3.48E-02**	2.55E-01	3.64E-01	**0.90529**	0.71071	0.61015
	DAS_CMOP3(7)	**1.90E-01**	2.31E-01	6.99E-01	**0.70016**	0.68399	0.36955
	DAS_CMOP4(7)	1.17E+00	**6.68E-01**	1.15E+01	0.12651	**0.34021**	0.00000
	DAS_CMOP5(7)	1.31E+00	**9.24E-01**	1.10E+01	0.18589	**0.34486**	0.00000
	DAS_CMOP6(7)	1.24E+00	**9.72E-01**	1.22E+01	0.13717	**0.26472**	0.00000

*The best variant in each group is in **bold**. Results of the cMLSGA are highlighted: in green, if cMLSGA is better than both of the implemented algorithms; in yellow, if cMLSGA is better than one of them; and in red, if it is worse than both of the algorithms.*

From the presented results it can be seen that cMLSGA_MOEA/D-MSF_HEIA is better than the corresponding MLSGA variants on 27 cases for the IGD and on 24 cases for the HV, whereas it only exhibits worse performance on only 7 and 6 cases, respectively. The cMLSGA_IBEA_BCE outperforms both MLSGA hybrids on the 22 test problems for the IGD indicator and 24 problems for the HV indicator. For both indicators it has a lower performance on 2 cases only. This demonstrates that the cMLSGA approach increases the generality of the search over the original MLSGA, similarly when comparing to the implemented algorithms. While the cMLSGA generally outperforms the MLSGA, this is usually not the case for some of the test problems where a higher diversity of the search is preferred, such as the imbalanced and constrained cases, categories VI-XI. MLSGA is developed to improve the diversity of the search and therefore these problems are highly suited to its mechanisms. Furthermore, the results indicate that the cMLSGA is often hindered by worse of the implemented algorithms, e.g. cMLSGA_IBEA_BCE on DAS_CMOP4-6, where IBEA is having good results and BCE is unable to find any valid solutions. Therefore, co-evolutionary approach leads to a higher generality of the performance across all problems, but it is unlikely to outperform a specialist-solver, diversity-based in that case, on its preferred problem, as there is no-free-lunch after.

5.3 General-solver approach

In this chapter the co-evolutionary-based variant of the MLSGA, denoted as cMLSGA, was proposed in order to enhance generality across different type of problems. Presented results show that the cMLSGA is able to successfully combine the advantages of both implemented methodologies, while maintaining the "diversity-first, convergence-second" behaviour seen in the previous iterations of MLSGA. Therefore, leading to a more general approach with a strong focus on the diversity. In most of tested cases, the performance of cMLSGA is better, or at least in between the two utilised methodologies, especially on the diversity-demanding problems such as discontinuous, constrained and bias functions. This indicates that the proposed methodology while providing a more general performance is usually limited by the worst of the implemented strategies. Therefore, the cMLSGA is highly unlikely to outperform the specialist-solvers on their favoured cases, or the problems where a strong convergence is needed. However, this behaviour was expected from the general-solver type of algorithm.

As the performance of the cMLSGA is bounded by the implemented algorithms a question could be raised, how to select the best combination of algorithms in practice. Unfortunately, currently there is no practical mean to assign the algorithms to sub-populations in the way that would be most suited for a given problem in the cases with limited knowledge about the search space. Here, the combination of HEIA and MOEA/D-MSF is suggested, as these methodologies show the most "general" behaviour, due to utilisation of two complimentary strategies: strong convergence of MOEA/D-MSF; and a high diversity/generality of HEIA. Obviously, it is still possible, that the performance of resulting cMLSGA will be low on a particular problem. Nevertheless, this applies to any existing methodology, but it is here suggested that by combination of two distinct search patters and utilisation of the diversity-first approach, cMLSGA's chances of success are significantly higher than other current state-of-the-art algorithms on cases where the solver has to be chosen in the uninformative process. However, in order to further validate these claims and finally validate the cMLSGA's performance as the diversity-oriented general-solver, it has to be comprehensively compared with the top-performing algorithms taken from the current state-of-the-art, including the simulations on practical problems.

Chapter 6 Applying the MLSGA to the practical problems and comparison with the current state-of-the-art

In this chapter, the "diversity-first" general-solver approach of the MLSGA is validated and it is evaluated if that approach is beneficial for the optimisation of practical problems. Firstly, the MLSGA's performance is compared with the state-of-the-art algorithms on the benchmarking set utilised in previous chapter, to find the best general-solver genetic algorithm. In the next step, the test set is expanded by functions with 3+ objective in order to evaluate if the previous findings apply to the cases with more than two objectives. Finally, the MLSGA and top performing algorithms from the current state-of-the-art are applied to three selected practical problems.

6.1 Rules and parameters

Similarly, to the previous chapters, the simulations are performed over 30 separate runs, with 300,000 function evaluations for each run, and both IGD and HV are used as the performance indicators. Furthermore, due to the fact that simulated algorithms utilise different population sizes, all Pareto optimal fronts are limited to 600 individuals, for the purposes of IGD and HV calculation, in order to assure the fairness of comparison.

Two MLSGA approaches, MLSGA-hybrid and cMLSGA, are compared with eight current state-of-the-art algorithms: U-NSGA-III [34], MOEA/D [35], MOEA/D-MSF and PSF [83], MTS [93], IBEA [121], BCE [103] and HEIA [36], which are chosen due to reasons detailed in Chapter 4. Those are utilised as is, if the code was provided or recreated based on the original publications and their principles of working can be found in the corresponding literature. However, for all algorithms, where the constraint handling was not originally implemented, such as MOEA/D, the same strategy as in the MLSGA is applied, which is described in Chapter 3. For the comparison, the MLSGA_HEIA and cMLSGA_MOEA/D-MSF_HEIA variants are selected as hybrid and co-evolutionary variant of the MLSGA respectively, due to highest focus on generality and diversity of the search as discussed in Chapters 4 and 5

The MLSGA-hybrid and cMLSGA are using the same parameters as in Chapters 4 and 5 respectively, whereas other algorithms utilises the same setting of hyper-parameters as in the original publications. Corresponding values of these parameters for each tested algorithm can be found in Appendix E.

6.2 Rankings on the multi-objective test sets

6.2.1 Two-objective problems

Performances of selected MLSGA-hybrid and cMLSGA variants are compared with the current state-of-the-art on 60 two-objective cases. The same test set as for evaluation of the cMLSGA in Chapter 5 is utilised. It is summarised in Table 5.2. The comparison is presented in Table 6.1 in the form of average ranking, according to the IGD and HV indicators for each category of problem. In addition, the average ranks across all tested problems and the standard deviations of ranks for each algorithm are provided as well to further evaluate the generality of each algorithm.

From presented ranking it can be seen that the best solver on average is cMLSGA, 3.60/3.82 for the IGD/HV metrics; the MLSGA-hybrid comes second with 4.33/4.15 for the IGD/HV indicators; whereas HEIA is third with 4.40/4.58 respectively. Higher differences in the HV scores indicates that the cMLSGA is more likely to promote a high diversity than HEIA but is less likely to do so than MLSGA-hybrid. In addition cMLSGA is shown to be more general than both of them, as it has a lower deviation in performance with smaller standard deviations of its positions in the rankings, 1.57/1.58 for the IGD/HV compared to the MLSGA's and HEIA's, 1.93/1.98 and 2.59/2.59 respectively. The differences in robustness are illustrated in Fig. 6.1, where the occurrence of each rank is presented for the MLSGA-hybrid, cMLSGA and HEIA algorithms. All three algorithms are often the top performers, rank 1 and 2, with HEIA and MLSGA-hybrid being the first more regularly, 11 times for the IGD and 10 for the HV for HEIA algorithm, and 8/5 for the MLSGA-hybrid, compared to 4/7 for the cMLSGA. However, when HEIA and MLSGA-hybrids show low performance on a problem they performs very poorly with rank 8 eight times for both indicators for HEIA, and 1/2 for MLSGA-hybrid; rank 9 one time for both algorithms according to HV and once according to IGD for MLSGA-hybrid; and rank 10 two times for the IGD and one time for the HV for HEIA only. Whereas the lowest rank for cMLSGA is 7 and it has a high occurrence of the 3^{rd}, 4^{th} and 5^{th} positions. This demonstrates that the cMLSGA is the best general-solver and it further proves a higher specialisation of the MLSGA-hybrid in comparison to the cMLSGA.

TABLE 6.1: Ranking of the 10 state-of-the-art genetic algorithms according to the average performance on distinct categories of two-objective problems according to the IGD/HV indicators.

Category	U-NSGA-III	MOEA/D	MOEA/D-PSF	MOEA/D-MSF	BCE	IBEA	MTS	HEIA	cMLSGA	MLSGA
I	5.60/3.60	4.00/4.80	5.60/6.00	7.60/8.20	9.40/9.60	8.00/7.80	8.40/8.40	3.40/3.60	1.60/1.80	1.40/1.20
II	8.00/8.33	3.00/3.67	4.67/4.33	3.33/4.00	9.00/8.33	10.0/10.0	2.67/4.00	1.33/1.00	7.00/6.67	6.00/4.67
III	3.43/3.71	5.43/5.57	8.43/8.14	6.29/6.14	7.43/7.14	7.86/7.71	6.43/8.00	2.86/2.43	2.71/2.29	4.14/3.86
IV	3.67/3.67	3.67/4.00	6.67/6.67	5.67/5.67	9.00/8.67	7.00/7.00	8.67/9.67	1.00/1.00	4.67/4.33	5.00/4.33
V	4.75/6.25	6.50/5.25	6.00/6.00	5.75/5.00	8.75/8.75	8.00/7.75	3.50/4.25	4.25/5.00	4.00/3.75	3.50/3.00
VI	5.90/4.90	5.60/5.50	3.10/3.20	2.40/2.60	8.90/9.00	8.50/8.40	5.10/5.40	5.40/5.80	4.60/4.80	5.50/5.40
VII	7.00/4.33	6.33/7.33	7.00/6.67	5.67/7.00	7.33/7.33	9.00/9.00	4.67/3.67	2.67/5.33	1.67/2.67	3.67/1.67
VIII	5.25/4.50	7.75/7.00	8.25/8.00	7.00/7.75	7.25/7.50	9.50/9.75	1.00/1.00	3.00/4.00	3.25/2.75	2.75/2.75
IX	8.33/8.00	1.00/1.33	3.67/3.00	2.67/2.67	6.67/6.33	10.0/10.0	8.67/9.00	4.00/4.67	3.67/3.33	6.33/6.67
X	4.33/4.67	5.67/5.50	5.50/5.92	4.33/4.33	9.83/9.75	7.00/6.33	3.83/4.00	6.67/6.17	3.67/4.00	4.17/4.33
XI	5.00/4.67	6.58/4.25	6.58/6.25	5.92/5.75	8.58/9.08	7.67/7.50	2.67/4.67	6.00/4.50	2.50/4.17	3.50/4.17
XII	4.17/4.33	5.00/3.67	5.33/6.08	4.67/3.83	9.67/9.42	4.33/4.17	4.67/4.83	7.00/7.67	4.33/5.00	5.83/6.00
Overall	5.22/4.85	5.28/4.94	5.76/5.78	4.98/5.04	8.61/8.58	7.87/7.72	4.95/5.55	4.42/4.58	3.60/3.82	4.33/4.15
std	2.59/2.68	2.95/2.75	2.68/2.78	2.67/2.86	1.47/1.55	2.35/2.44	3.14/3.04	2.59/2.59	1.57/1.58	1.93/1.98

The best algorithm in each category is highlighted in green, the second and third best solvers are highlighted in orange and red, respectively.

Comparing distinct categories of problems, it can be observed that cMLSGA and MLSGA-hybrids are achieving a significantly better performance on simple, concave and discontinuous cases without constraints, and constrained problems, categories I, III, V, VII, X and XI, respectively. Furthermore, a high performance can be observed on constrained functions with continuous characteristics, category VIII, but with a clear domination of the MTS algorithm, rank 1 on every problem of this type, indicating that it is specialist-solver for those cases. This further supports previous findings by showing a high focus on diversity of the MLSGA-based methodologies. For imbalanced problems, categories VI and IX, the highest performance is achieved by MOEA/D based solvers, further indicating that decomposition is preferred on those type of problems.

Comparing cMLSGA and MLSGA-hybrid, it can be seen that the cMLSGA is usually exhibiting a better performance on the problems where HEIA is showing a low effectiveness, categories VI, VII, X, XI and XII. Here it is suggested that it is due to implementation of a second search strategy, which helps to maintain a high performance on more kind of cases, whereas MLSGA-hybrid is strongly bounded by a single evolutionary algorithm, and thus its performance on a particular problem.

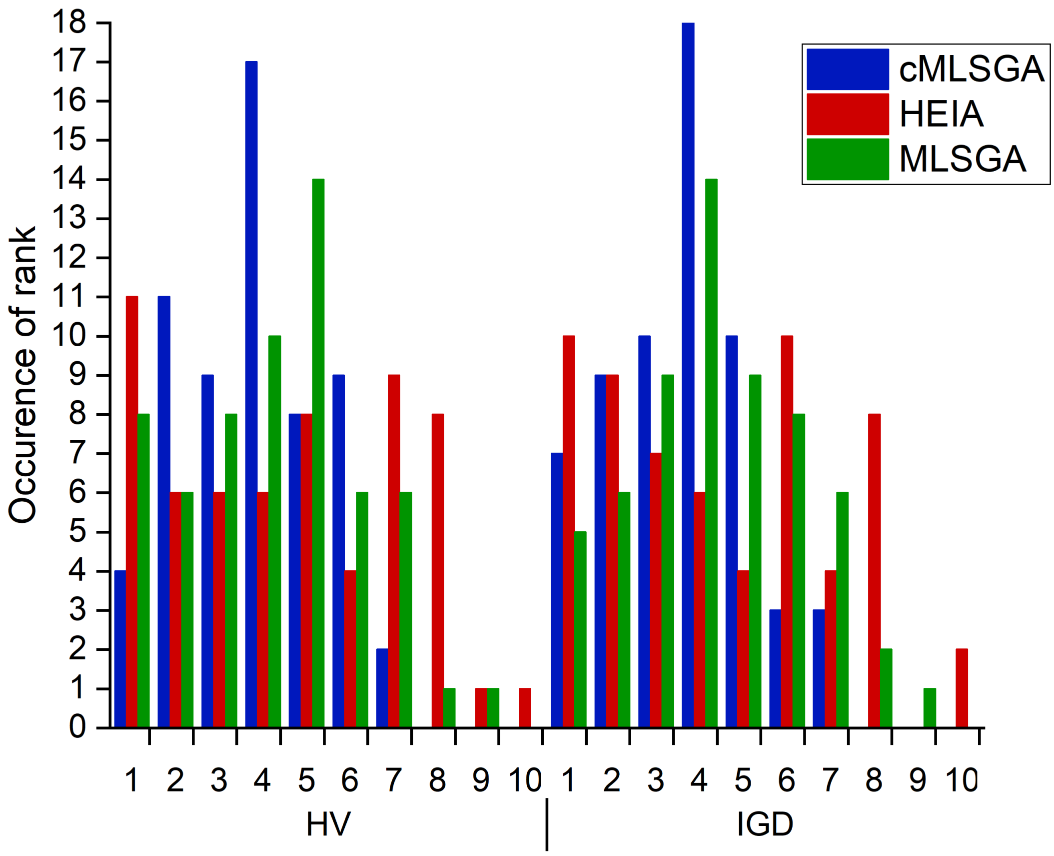


FIGURE 6.1: Occurrence of ranks for HEIA, cMLSGA and MLSGA-hybrid algorithms on two-objectives benchmarking problems.

6.2.2 Problems with 3+ objectives

In order to operate on problems with more than 2 objectives, the original MLS-U fitness separation strategy has to be modified. Here a simple extension of the MLS-U strategy is proposed, where $M + 1$ collective types are utilised for M-objective problem:

1. In Type-1 assign $\{f_1, \ldots, f_M\}$ to collective and individual-level
2. In Type-i, where $\{i = 2, \ldots, M + 1\}$, assign $\{f_1, \ldots, f_M\}$ to individual-level and $\{f_{i-1}\}$ to collective-level

Therefore, the first type of collectives contains all objectives on both levels of selection, and the rest of types contains a distinct objective on the collective-level, and all objectives on the individual-level. Exemplar fitness assignments to each level of selection for 3 and 5 objectives cases are presented in Table 6.2. For both number of objectives, the MLSGA_U-NSGA-III and MLSGA_MOEA/D-MSF_U-NSGA-III are used as hybrid and co-evolutionary variant respectively, due to the low performance of HEIA on problems with more than two-objective. The hyper-parameters for MLSGA-hybrid and cMLSGA remains as presented in Chapter 4 and 5 respectively. Interestingly, 8 collectives are preferred for 5 objectives, despite of having less than 2 groups of each type, according to conducted pre-benchmarks. It is most likely caused by the fact that higher numbers of collectives require more individuals in population in order to remain effective, as discussed previously in Chapter 4. For the state-of-the-art algorithms, 1000 population size is utilised as it is providing a better performance. The rest of parameters remains the same as in previous sections.

In order to operate on more than two-objectives, the test set is extended by 40 three- and 18 five-objective functions. These problems, are divided into categories, following the same rules as for the two-objective cases, as summarised in Table 6.3. All presented problems have three objectives, with distinction that the scalable functions that are used in both 3- and 5-objective benchmarks are in **bold**.

TABLE 6.2: Fitness functions used on each level of selection, depending on the collective type, for optimisation of 3- and 5-objective problems.

Type of collective	Collective-level	Individual-level
3 objectives		
1	$f_1(x)$	$f_1(x), f_2(x)$ *and* $f_3(x)$
2	$f_2(x)$	
3	$f_3(x)$	
5 objectives		
1	$f_1(x)$	$f_1(x), f_2(x), f_3(x), f_4(x)$ *and* $f_5(x)$
2	$f_2(x)$	
3	$f_3(x)$	
4	$f_4(x)$	
5	$f_5(x)$	

TABLE 6.3: Summary of the utilised test set with 3+ objectives

Category	Problem	d	Additional properties
	Unconstrained		
I. Concave	**DTLZ2**	M+9	
	DTLZ3	M+9	Multimodal
	DTLZ4	M+9	Biased
	DTLZ5	M+9	Degenerated
	DTLZ6	M+9	Degenerated, Biased
	UF8	30	Complex PS
	UF10	30	Complex PS
	WFG4	2M+18	Multimodal
	WFG5	2M+18	Deceptive
	WFG6	2M+18	Non-separable
	WFG7	2M+18	Biased
	WFG8	2M+18	Biased, Non-separable
	WFG9	2M+18	Biased, Non-separable, Deceptive
IV. Linear/Mixed	**DTLZ1**	M+4	Linear, Multimodal
	WFG1	2M+18	Biased, Mixed
	WFG3	2M+18	Non-separable, Degenerated, Linear
V. Discontinuous	**DTLZ7**	M+19	Mixed, Multimodal
	UF9	30	Complex PS
	WFG2	2M+18	Convex, Non-Separable
VI. Imbalanced	MOP6	10	Linear
	MOP7	10	Concave
	IMB4	10	Linear
	IMB5	10	Concave
	IMB6	10	Linear
	IMB10	10	Linear
	Constrained		
VII. Discontinuous	**DTLZ8**	10M	Mixed, Degenerated, Biased
	DTLZ9	10M	Concave, Degenerated
	CF8	10	Concave, Degenerated, Complex PS
	CF9	10	Concave, Complex PS
	CF10	10	Concave, Complex PS
IX. Imbalanced	IMB14	10	Linear
X. Diversity-hard	DAS-CMOP7(5)	30	Linear, Degenerated, Discontinuous
	DAS-CMOP8(5)	30	Concave, Discontinuous
	DAS-CMOP9(5)	30	Concave, Discontinuous, Biased
XI. Feasibility-hard	DAS-CMOP7(6)	30	Linear, Degenerated, Discontinuous
	DAS-CMOP8(6)	30	Concave, Discontinuous
	DAS-CMOP9(6)	30	Concave, Discontinuous, Biased
XII. Convergence-hard	DAS-CMOP7(7)	30	Linear, Degenerated, Discontinuous
	DAS-CMOP8(7)	30	Concave, Discontinuous
	DAS-CMOP9(7)	30	Concave, Discontinuous, Biased

d denotes the number of decision variables. The scalable problems are in bold. M denotes the number of objectives in the scalable problems.

The results are presented in the form of an average ranging on each category of the problem, according to the IGD and HV indicators, in Table 6.4 for 3 objectives functions and in Table 6.5 for 5 objectives.

Table 6.4: Ranking of the 10 state-of-the-art genetic algorithms according to the average performance on distinct categories of three-objective problems according to the IGD/HV indicators.

Category	U-NSGA-III	MOEA/D	MOEA/D-MSF	MOEA/D-PSF	BCE	IBEA	MTS	HEIA	cMLSGA	MLSGA
I	3.69/4.15	4.23/3.54	5.54/6.46	7.00/6.92	7.77/8.23	6.69/4.92	8.00/8.54	4.38/3.69	2.92/3.15	4.77/5.38
IV	2.67/2.33	4.00/5.00	5.33/5.00	6.33/7.33	10.00/9.67	4.00/5.33	8.33/9.33	3.33/1.33	4.67/4.00	6.33/5.67
V	3.00/2.33	8.00/5.33	5.33/5.33	7.33/6.33	4.33/7.67	7.67/8.33	5.67/9.00	5.00/4.33	3.67/2.33	5.00/4.00
VI	6.50/7.17	4.17/4.83	1.17/1.33	2.17/2.33	7.33/6.67	9.83/9.83	6.83/7.33	5.50/5.00	5.00/4.50	6.50/6.00
VII	5.00/4.20	5.80/5.70	5.60/6.10	4.80/6.00	6.20/6.90	7.20/7.00	3.80/3.70	7.80/6.40	3.60/3.40	5.20/5.60
IX	5.00/5.00	1.00/3.00	2.00/1.00	4.00/2.00	6.00/8.00	9.00/9.00	10.00/10.00	3.00/4.00	7.00/6.00	8.00/7.00
X	5.33/6.67	9.00/7.00	6.33/7.83	6.00/6.67	10.00/9.50	4.33/4.33	3.33/5.00	5.67/3.67	2.00/1.67	3.00/2.67
XI	1.00/4.33	6.67/6.33	8.33/8.17	8.33/8.17	8.33/8.17	5.83/5.67	7.50/7.50	4.00/3.00	3.00/2.33	2.00/1.33
XII	3.67/4.00	9.33/8.17	5.67/7.33	5.33/7.17	9.33/8.67	5.33/5.33	6.00/6.33	4.00/3.33	2.67/1.67	3.67/3.00
Overall	4.100/4.550	5.525/5.050	5.050/5.613	5.800/6.075	7.700/8.013	6.813/6.350	6.663/7.375	4.950/4.025	3.525/3.175	4.875/4.775
std	2.311/2.291	3.060/2.576	3.211/3.038	2.697/2.619	2.339/1.892	2.669/2.886	3.127/2.602	2.397/2.454	1.565/1.579	2.064/2.454

The best algorithm in each category is highlighted in green, the second and third best solvers are highlighted in orange and red, respectively.

Table 6.5: Ranking of the 10 state-of-the-art genetic algorithms according to the average performance on distinct categories of five-objective problem according to the IGD/HV indicators.

Category	U-NSGA-III	MOEA/D	MOEA/D-MSF	MOEA/D-PSF	BCE	IBEA	MTS	HEIA	cMLSGA	MLSGA
I	2.64/4.50	7.36/4.36	5.27/6.00	7.18/6.64	6.09/7.59	4.00/2.91	8.73/7.95	4.18/4.23	4.45/4.95	5.09/5.86
II	5.00/3.33	3.67/4.00	3.67/5.33	6.33/7.67	7.00/9.33	7.00/4.33	5.33/6.00	3.67/3.33	6.33/5.33	7.00/6.33
III	2.50/3.00	10.00/3.00	5.00/8.00	9.00/9.50	4.00/5.00	5.50/5.00	7.00/9.00	3.50/4.00	4.50/3.50	4.00/5.00
V	5.50/4.75	5.50/6.00	7.50/7.75	5.50/7.25	7.00/4.25	4.00/4.00	7.00/6.25	6.50/5.75	3.00/3.75	3.50/5.25
Overall	3.333/4.167	6.833/4.333	5.222/6.306	7.056/7.194	6.111/7.222	4.667/3.500	7.778/7.556	4.278/4.222	4.611/4.722	5.111/5.778
std	2.494/2.068	2.986/2.309	2.954/2.883	2.297/2.902	2.580/2.155	2.848/2.432	1.959/2.437	2.578/2.663	2.189/2.083	2.378/2.155

The best algorithm in each category is highlighted in green, the second and third best solvers are highlighted in orange and red, respectively.

From the presented data it can be seen that a similar behaviour can be observed for three-objective problems, Table 6.4, as for the cases with only two-objectives. The best general-solver is the cMLSGA, whereas U-NSGA-III comes second and the HEIA is third. The IGD rankings are 3.525, 4.100, 4.950 for these algorithms correspondingly, while according to the HV average ranks are: 3.175, 4.550 and 4.025. Similarly, to the two-objective cases, the cMLSGA also shows the lowest standard deviations, further proving its high generality. Comparing the occurrence of each rank for U-NSGA-III, cMLSGA and MLSGA-hybrid, illustrated in Fig. 6.2, it can be seen that cMLSGA is again more likely to avoid a poor performance. In this case, the cMLSGA is more likely to be top performer according to the HV, but it less likely to exhibit the highest performance according to the IGD metric, as it is often outperformed by U-NSGA-III. Furthermore, the cMLSGA is significantly more likely to be third, 13 times for both IGD and HV compared to 8 and 4 times for U-NSGA-III for the HV and IGD, respectively. In the contrast, U-NSGA-III is significantly more likely to perform poorly, as in the worst case scenario it is 10^{th} for the IGD and 9^{th} for the HV, whereas for the cMLSGA the lowest position is 7^{th} with both indicators. Furthermore, U-NSGA-III is more likely than cMLSGA to achieve ranks 6 and 7. Interestingly, both algorithms are more likely to be in the middle positions 2^{nd} to 5^{th}, showing the distribution that is reflecting a general performance. This indicates the high generality and universality of both solutions as they are often outperformed by the specialist-solvers on their preferred cases. For the MLSGA-hybrid

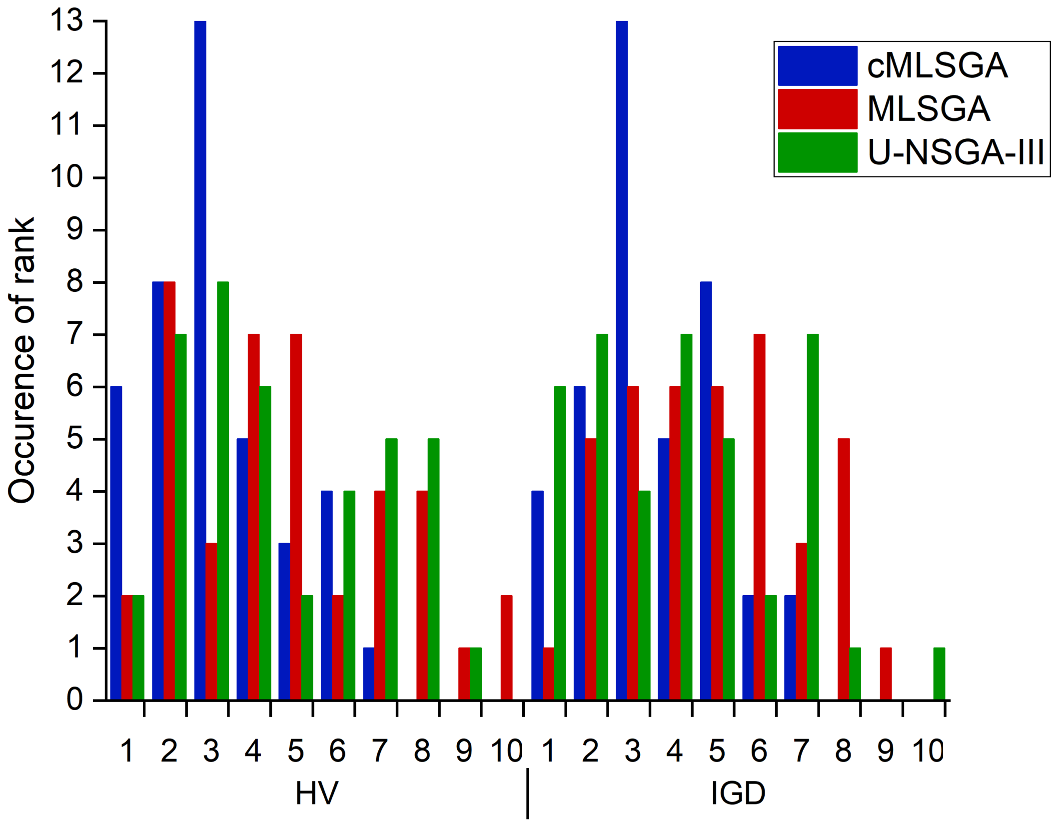


FIGURE 6.2: Occurrence of ranks for U-NSGA-III, cMLSGA and MLSGA-hybrid algorithms on three-objective benchmarking problems

it can be seen that it is more likely to perform poorly than both cMLSGA and U-NSGA-III, positions 7^{th} to 10^{th}. This indicates that the MLS-based strategy utilised in MLSGA is less effective with more than two-objectives.

Similarly, to the previous tests, a high performance of the MLS-based algorithms, especially the cMLSGA, can be observed for the diversity-hard problems, such as discontinuous and constrained problems, categories III-V, VII, X and XI. Therefore, showing a high focus of that methodology on the diversity. Lower performance on categories VI and IX, is explained by the imbalanced structure of those problems, which is favouring the forced problem decomposition, such as mechanisms used in the MOEA/D-based algorithms, as discussed previously. Interestingly, the cMLSGA is outperforming other solutions on unconstrained cases with 3 objectives and a concave geometry, category I. Similar principle was observed for the two-objective problems. This is indicating a higher demand for the diversity on biased problems with concave Pareto optimal front geometries and a lower adjustment of the current state-of-the-art for this kind of cases. Similarly, a high performance of the cMLSGA is shown on the convergence-hard constrained problems, category XII, which indicates that the three-objective functions of that kind, require more diversity in comparison to their two-objective counterparts.

cMLSGA is outperforming the hybrid variant on each category of problems on average. It is here suggested that this is caused by a higher complexity of the problems with more than two objectives. On those cases, most of the algorithms are not able to accurately approximate the whole Pareto optimal front and the MLS-based strategy is less effective on them, as indicated before. Therefore, utilising multiple search strategies is more beneficial than enhancing the diversity alone, in order to be able to converge on more regions of the objective space.

Lowered performance of the MLSGA methodology on the cases with increasing number of objectives is further shown with five-objective problems, as presented in Table 6.5. The cMLSGA is 4^{th} and MLSGA is 5^{th}, while U-NSGA-III and IBEA are showing the most general-approach. However, in this case a limited set of problems is utilised, with 18 functions only, where majority of them, 16 out of 18, does not contain any constraints, and only 4 cases are discontinuous. Furthermore, all simulated problems came from two test sets only, DTLZ and WFG, and thus the diversity of problem's characteristics is limited. According to that, it is possible that the utilised 5-objective test set is biased towards certain characteristics and algorithms. According to that, it is not possible to draw any overarching conclusions about the best general type solver for the five-objective cases or the lowered performance of the MLS-based algorithms on them. This is supported by a high performance of the MLSGA-hybrid and cMLSGA on all from the available five-objective constrained problems, category V, where cMLSGA comes first in average and MLSGA-hybrid is third.

The lowered performance on the five-objective cases can potentially indicate the limitations in scalability of the MLSGA approach. With increasing number of objectives, due to utilised split in fitness function, either: more collectives have to be implemented, resulting in lowered effectiveness due to small sub-populations; or the fitness separation have to be modified, so more than one objective is used during the collective-level fitness calculation, but in this case the region based search will be strongly hindered. However, the problem with scalability to a higher number of objectives is shared by all GAs not only MLSGA, as indicated in [144].

6.3 Optimisation of the engineering problems

Two engineering cases, taken from the literature, are used in this section: optimisation of the spatial arrangement of leisure boats (3DSA) [145] as the two-objective highly constrained case with a large number of variables, 3 constraints and 68 variables respectively; and the structural design of a composite stiffened plate [146], two cases with 2 (CLPT2) and 3 objectives (CLPT3) respectively, in order to verify the performance on unconstrained cases with different number of objectives. The test set is summarised in Table 6.6, with full description provided in Appendix F.

Similarly, to the previous benchmarks, 600 population size is used for all algorithms except of the cMLSGA and MLSGA-hybrid, where 1000 individuals are included. Those values have been selected as the best for corresponding algorithms after the series of pre-benchmarks. Rest of the operational parameters are kept constant from previous tests and can be found in Appendix E. As for the practical problems the true Pareto optimal front is not known the reference front, for the purposes of IGD and HV calculations, is obtained by non-dominated selection of all the Pareto optimal fronts achieved by every algorithm across all runs. However, this strategy can result in inaccurate IGD values, as the final front can be biased toward certain regions and it is not uniformly spread, therefore a stronger emphasis should be put on the HV metric in order to provide more accurate comparison.

Table 6.6: Summary of the utilised test set.

Case	Objectives	Variables	Constraints
3DSA	2	68	3
CLPT2	2	6	0
CLPT3	3	6	0

6.3.1 Verification of performance

Performance comparison of selected genetic algorithms is presented in Table 6.7 in the form of ranking according to the IGD and HV metrics. For each algorithm, the average values are presented with min, max and standard deviation given in brackets in *italics*. Furthermore, the results for cMLSGA and MLSGA-hybrid are highlighted in green and blue, respectively. However, as certain runs on the 3DSA problem do not provide any feasible solutions, only the runs where at least one valid solution is achieved are included in both tables. This issue is addressed separately later in this section.

Table 6.7: Ranking of genetic algorithms on the practical problems according to the average HV/IGD metrics.

Rank	Problem					
	HV			IGD		
	3DSA	CLPT2	CLPT3	3DSA	CLPT2	CLPT3
1	cMLSGA 0.75577[+] {0.68430; 0.78542; 0.02600}	cMLSGA 0.84595[+] {0.84580; 0.84626; 0.00019}	cMLSGA 0.86162 {0.86126; 0.86371; 0.00078}	cMLSGA 9.66E-2 {5.44E-2; 1.81E-1; 3.73E-2}	cMLSGA 8.66E-1[+] {6.54E-1; 9.93E-1; 1.47E-1}	cMLSGA 3.72E+0 {3.43E+0; 3.96E+0; 1.33E-1}
2	MLSGA 0.75336[+] {0.68411; 0.81625; 0.02500}	MLSGA 0.84584 {0.84581; 0.84617; 0.00006}	MLSGA 0.86155[+] {0.86126; 0.86372; 0.00070}	MLSGA 9.90E-2 {1.83E-2; 2.31E-1; 4.41E-2}	MLSGA 9.58E-1[+] {6.59E-1; 9.86E-1; 5.62E-2}	MLSGA 3.81E+0[+] {3.41E+0; 4.20E+0; 1.86E-1}
3	U-NSGA-III 0.75119 {0.51047; 0.84152; 0.07536}	HEIA 0.84583[+] {0.84582; 0.84584; 0.00000}	HEIA 0.86104[+] {0.86085; 0.86343; 0.00045}	HEIA 1.01E-1 {1.10E-2; 2.57E-1; 5.71E-2}	HEIA 1.15E+0[+] {1.00E+0; 1.18E+0; 2.89E-2}	HEIA 4.10E+0[+] {3.79E+0; 4.48E+0; 1.62E-1}
4	HEIA 0.75082[+] {0.61978; 0.82944; 0.04618}	BCE 0.84574[+] {0.84573; 0.84577; 0.00001}	U-NSGA-III 0.86059[+] {0.86047; 0.86066; 0.00005}	U-NSGA-III 1.06E-1[+] {8.28E-3; 3.26E-1; 8.06E-2}	BCE 1.29E+0[+] {1.13E+0; 1.33E+0; 4.75E-2}	U-NSGA-III 5.29E+0[+] {5.04E+0; 5.72E+0; 1.56E-1}
5	MOEA/D-MSF 0.67597 {0.65040; 0.70689; 0.01461}	MOEA/D-MSF 0.84569[+] {0.84566; 0.84571; 0.00002}	MOEA/D 0.85876[+] {0.85858; 0.85909; 0.00011}	MOEA/D-PSF 2.36E-1 {2.17E-1; 2.48E-1; 8.70E-3}	U-NSGA-III 1.57E+0[+] {1.52E+0; 1.63E+0; 2.57E-2}	BCE 6.51E+0 {5.90E+0; 7.20E+0; 2.77E-1}
6	MOEA/D-PSF 0.66761[+] {0.64449; 0.70895; 0.01481}	U-NSGA-III 0.84566[+] {0.84564; 0.84568; 0.00001}	MOEA/D-MSF 0.85819 {0.85799; 0.85836; 0.00009}	MOEA/D-MSF 2.37E-1[+] {2.21E-1; 2.50E-1; 7.71E-3}	MOEA/D-MSF 2.55E+0[+] {2.50E+0; 2.67E+0; 6.65E-2}	MOEA/D 1.84E+1 {1.72E+1; 1.90E+1; 4.10E-1}
7	BCE 0.60314 {0.39843; 0.68525; 0.08016}	MOEA/D 0.84549 {0.84547; 0.84549; 0.00000}	BCE 0.85835 {0.85752; 0.85879; 0.00028}	BCE 2.57E-1 {1.81E-1; 4.74E-1; 7.15E-2}	MOEA/D 5.51E+0[+] {5.49E+0; 5.57E+0; 1.75E-2}	MOEA/D-MSF 2.09E+1[+] {2.04E+1; 2.12E+1; 1.68E-1}
8	MOEA/D 0.55676[+] {0.38878; 0.65666; 0.08673}	MOEA/D-PSF 0.84545[+] {0.84540; 0.84548; 0.00003}	MOEA/D-PSF 0.85580[+] {0.85555; 0.85630; 0.00019}	MOEA/D 2.99E-1[+] {2.16E-1; 4.74E-1; 8.07E-2}	MOEA/D-PSF 5.83E+0[+] {5.66E+0; 6.03E+0; 9.93E-2}	MOEA/D-PSF 7.80E+1[+] {4.62E+1; 8.57E+1; 6.22E+0}
9	MTS 0 {0.00000; 0.00000; 0.00000}	MTS 0.83886[+] {0.83640; 0.84104; 0.00114}	MTS 0.84833[+] {0.84044; 0.85179; 0.00248}	MTS inf {inf; inf; 0.00E+0}	IBEA 3.53E+1[+] {1.15E+1; 3.15E+2; 6.36E+1}	IBEA 1.33E+2 {1.68E+1; 4.25E+2; 1.22E+2}
10	IBEA 0 {0.00000; 0.00000; 0.00000}	IBEA 0.82854 {0.81107; 0.83347; 0.00436}	IBEA 0.84218 {0.83710; 0.84864; 0.00316}	IBEA inf {inf; inf; 0.00E+0}	MTS 5.25E+1 {2.22E+1; 9.87E+1; 2.02E+1}	MTS 1.97E+2 {7.90E+1; 3.02E+2; 5.77E+1}

For each algorithm the average values are presented with min, max and standard deviation given in brackets in italics. Results from the cMLSGA and MLSGA-hybrid are highlighted in green and blue, respectively. [+] indicates if the results are significantly different to the next lower rank.

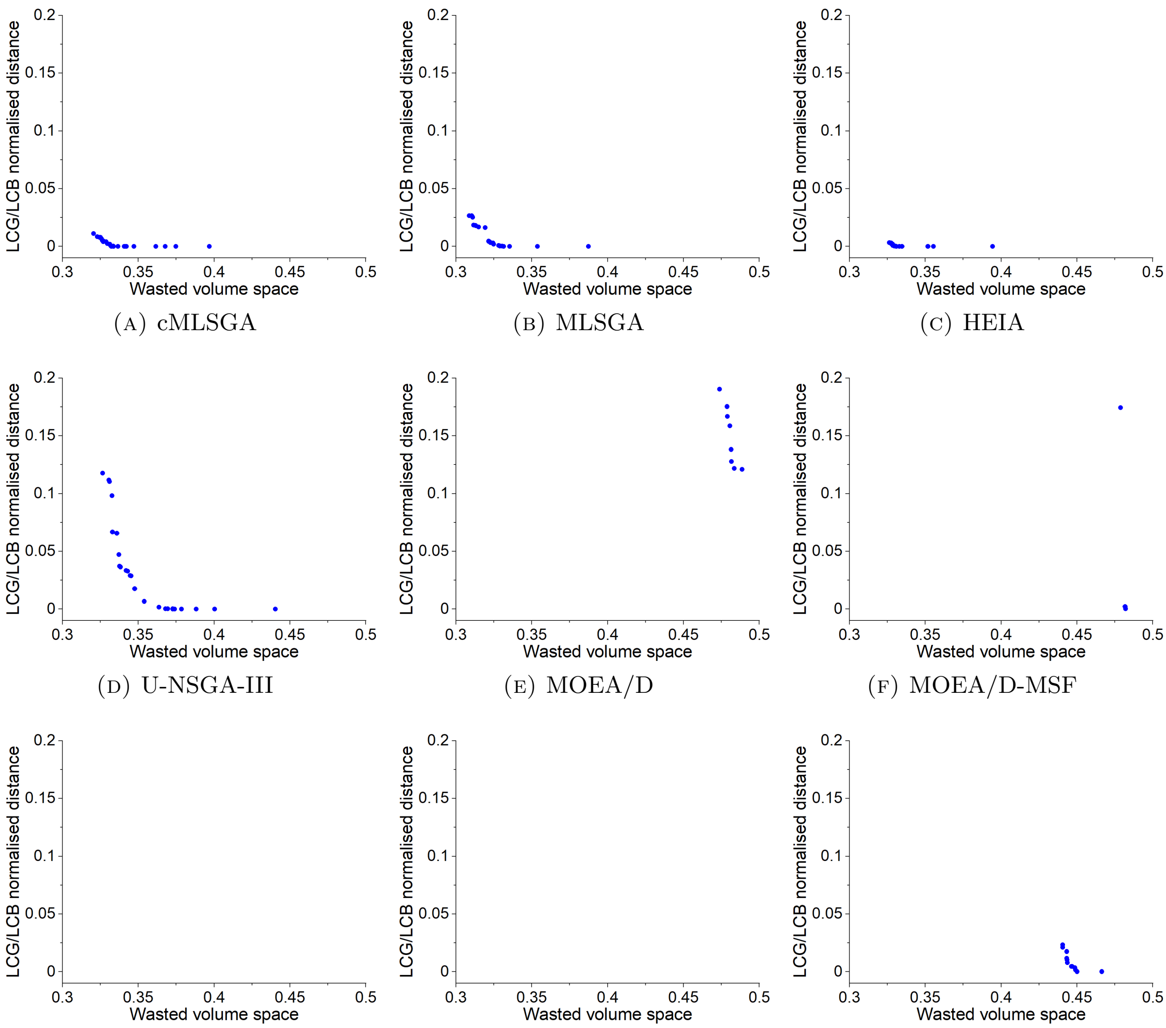
Table 6.8: Robustness of finding feasible solutions on the 3DSA engineering problem.

Algorithm	Robustness	
	Amount (/30)	Percentage (%)
cMLSGA	30	100
MLSGA	29	96.67
U-NSGA-III	29	96.67
HEIA	29	96.67
BCE	29	96.67
MOEA/D-MSF	27	90.00
MOEA/D-PSF	27	90.00
MOEA/D	26	86.67
MTS	0	0
IBEA	0	0

Results from the cMLSGA and MLSGA-hybrid are highlighted in green and blue, respectively.

From the table it can be seen that the best algorithm on all cases is cMLSGA; the second best is MLSGA-hybrid; the third in overall is HEIA, with rank 3 on the CLPT2 and CLPT3 and rank 4 on the 3DSA for the HV indicator, and rank 3 on all cases for the IGD; and the U-NSGA-III is fourth on average, as it is 3^{rd} on 3DSA, 4^{th} on CLPT3 and 6^{th} on CLPT2 according to the HV, and 4^{th} on 3DSA and CLPT3, and 5^{th} on CLPT2 according to the IGD. Bad performance is shown by the MOEA/D based algorithms, mostly ranks 5-8, and the worst solvers for these problems are MTS and IBEA. Therefore, it can be observed that three genetic algorithms with a higher emphasis on the generality of the search, MLSGA, HEIA and U-NSGA-III respectively, are outperforming the specialised-solvers, such as MOEA/D and MTS, on all cases, except for the CLPT2 problem where MOEA/D-MSF, MOEA/D-PSF and BCE have better diversity than U-NSGA-III. CLPT2 is the simplest of all problems, which is potentially explaining why the convergence in this scenario is becoming more important. Interestingly, all of the general-solvers exhibit similar performance on the highly constrained 3DSA problem, which has a large quantity of variables, while being significantly better than the specialist-solvers. This indicates that the specialist-solvers are unable to effectively solve more complex and constrained problems.

No feasible solutions are found within some runs of the highly constrained 3DSA problem. The robustness of each algorithm in finding feasible solutions on this case is presented in Table 6.8 as the number of times that an algorithm has found at least one feasible solution with the overall percentage out of 30 runs. Results for the cMLSGA and MLSGA are highlighted in green and blue, respectively. It can be seen that no algorithm except of the cMLSGA is able to achieve 100% robustness. Furthermore, robustness of the specialist-solvers is lower than of the more general ones. MOEA/D has found feasible solutions in 26 out of 30 cases, MOEA/D-PSF and MOEA/D-MSF in 27/30 cases, while HEIA, U-NSGA-III and MLSGA-hybrid have found feasible solutions in 29 out of 30 cases. Therefore, it can be seen that general-solvers are not only able to provide more accurate and diverse results, but also are having a higher

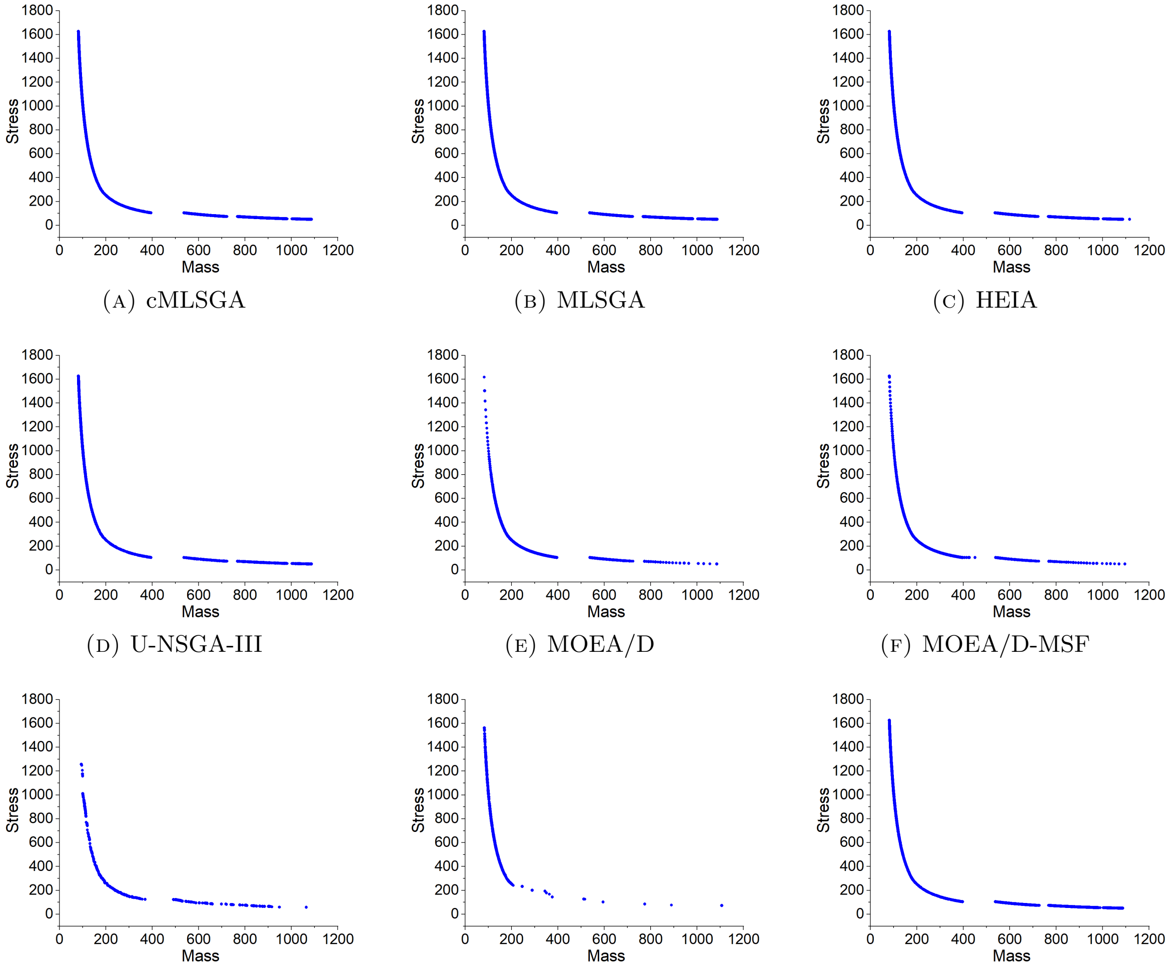


(A) cMLSGA (B) MLSGA (C) HEIA (D) U-NSGA-III (E) MOEA/D (F) MOEA/D-MSF (G) MTS (H) IBEA (I) BCE

FIGURE 6.3: Achieved Pareto optimal fronts on the 3DSA case

robustness on highly constrained cases. This is essential for the real-world applications, as often only a single optimisation run is conducted, due to cost and time limitations. That issue could be reduced by predefining the feasible solutions as the starting population instead of utilising fully random methods. However, in this case it has been decided not to follow this approach, as it might promote certain algorithms that are not focused on the search space exploration, thus reducing the potential for revolutionary designs as leading to final solutions being close to predefined points. Furthermore, in many real-world cases this adjustment is not possible, due to the lack of knowledge about where the feasible solutions are located.

The effectiveness of different GAs is further investigated by comparison of the achieved Pareto optimal fronts. Fronts achieved by each of the benchmarked algorithms are illustrated for the 3DSA problem in Fig. 6.3, CLPT2 case in Fig 6.4 and CLPT3 in Fig. 6.5. The representative Pareto optimal front is selected using a HV value closest to the average HV. The results for MOEA/D-PSF are not included due to a high resemblance to the Pareto optimal fronts achieved by MOEA/D-MSF. From the graphs, Figs 6.4 and 6.5, it can be observed that for

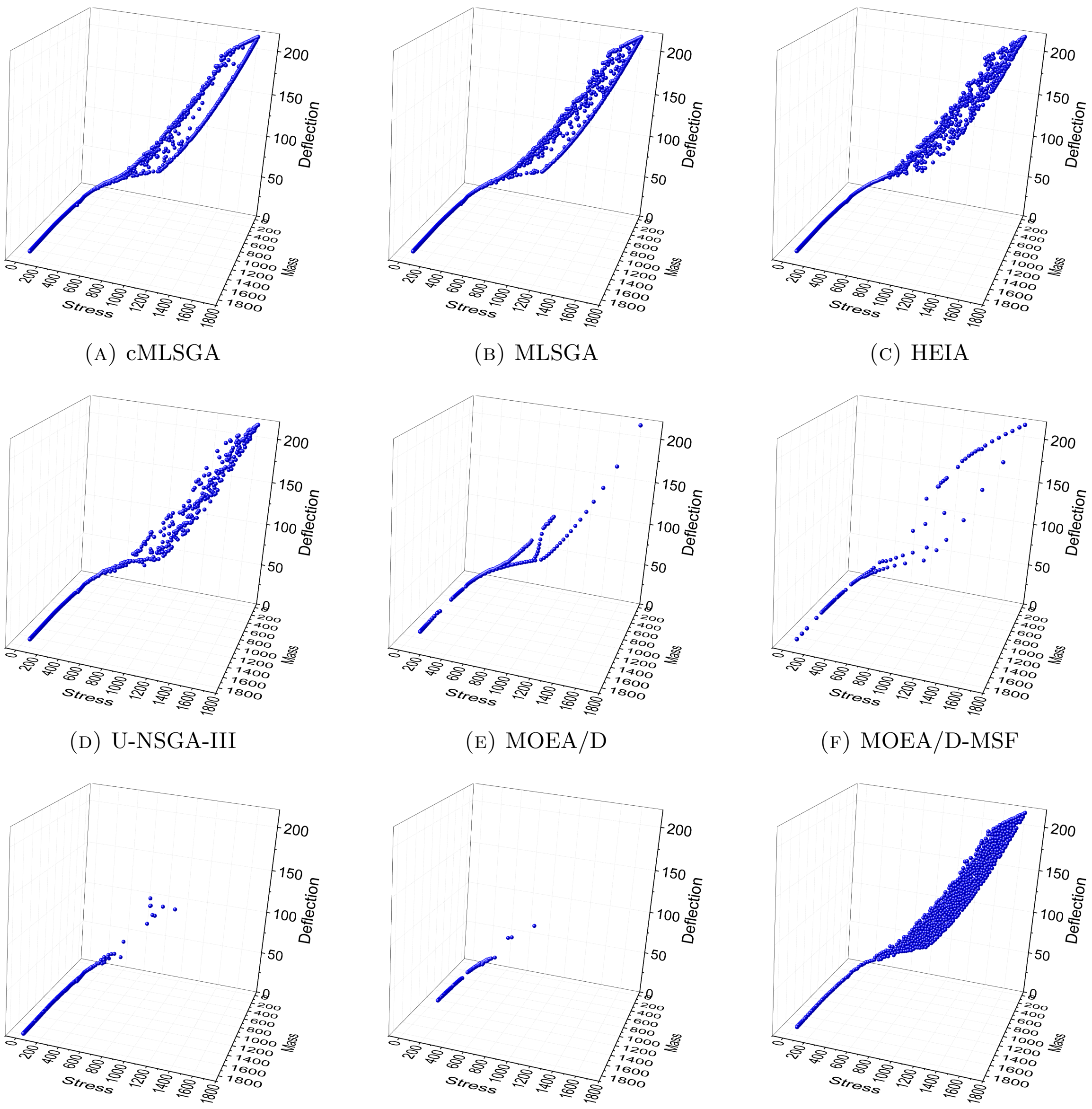


(A) cMLSGA (B) MLSGA (C) HEIA
(D) U-NSGA-III (E) MOEA/D (F) MOEA/D-MSF
(G) MTS (H) IBEA (I) BCE

FIGURE 6.4: Achieved Pareto optimal fronts on the CLPT2 case

the CLPT2 and CLPT3 all algorithms, except of MTS and IBEA, are able to find a wide spread set of points, including the boundary regions. However, it can be seen that in the cases of MOEA/D and MOEA/D-MSF, only a few solutions have been found in the boundary regions, whereas most points are located in the central regions of the objective space. All of the presented general genetic algorithms have a far greater uniformity due to a stronger emphasis on the diversity. For the 3DSA case, Fig. 6.3, it can be seen that cMLSGA, MLSGA, U-NSGA-III and HEIA are able to achieve a diverse set of points, while MOEA/D-MSF, MOEA/D and BCE are finding only a few random solutions. This further supports the previous findings that the specialist-solvers are unable to solve complex, constrained real-world cases, as are failing to provide a sufficient diversity.

Investigating the Pareto optimal fronts obtained by the tested algorithms on practical cases, some resemblance to the true fronts of the artificial benchmarking functions can be observed. This is illustrated in Fig. 6.6 for CLPT2 and in Fig. 6.7 for CLPT3. Pareto optimal front of the CLPT2, Fig. 6.6, is discontinuous and convex, showing similarities to ZDT1, ZDT4 [30], UF1-3 [31] and MOP1-2, 5 [32]. However, with exception that these benchmarking



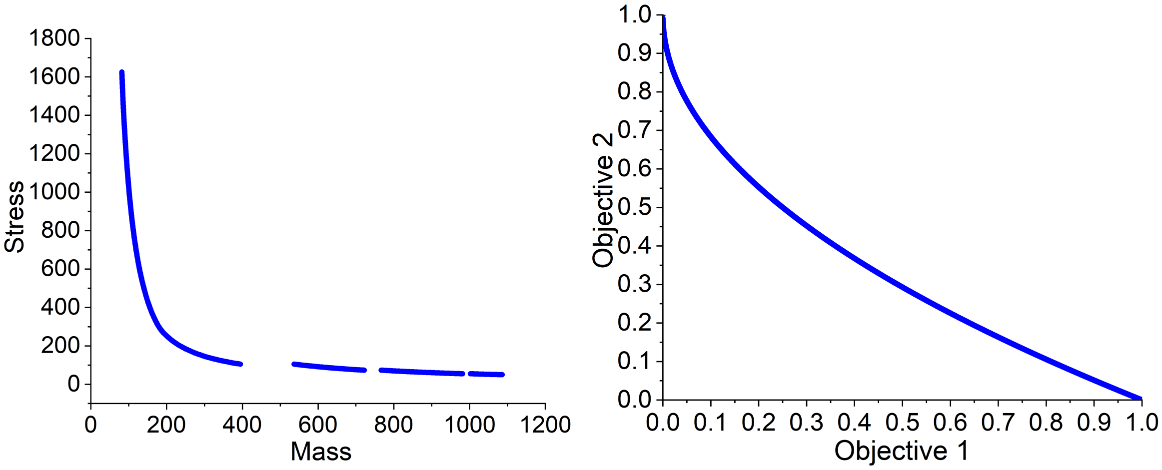
(A) cMLSGA (B) MLSGA (C) HEIA

(D) U-NSGA-III (E) MOEA/D (F) MOEA/D-MSF

(G) MTS (H) IBEA (I) BCE

FIGURE 6.5: Achieved Pareto optimal fronts on the CLPT3 case

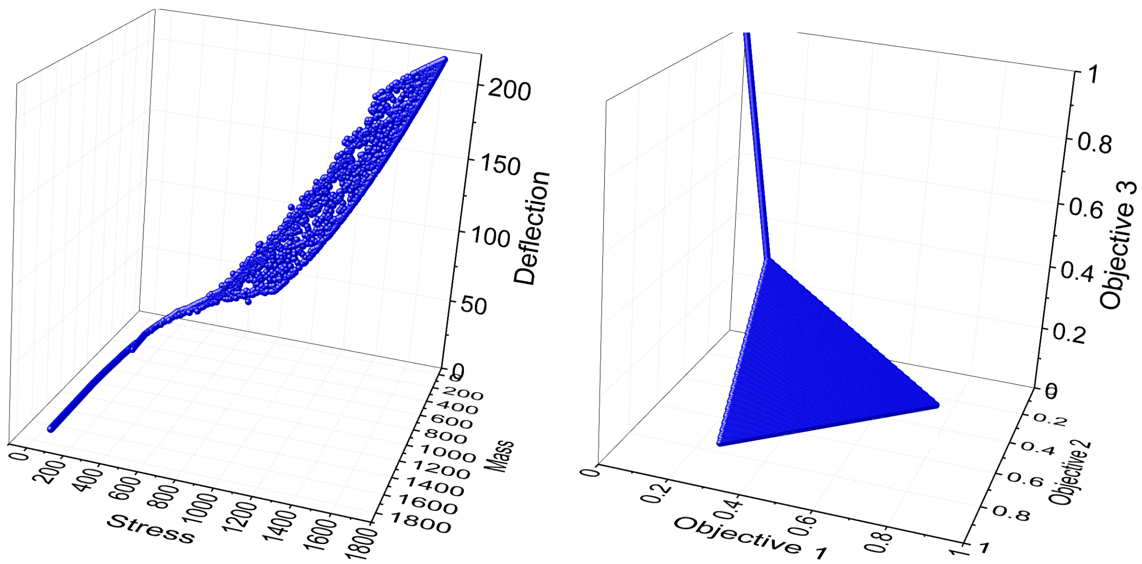
problems are continuous and with a lower bias towards the central regions of the Pareto optimal front. Furthermore, CLPT2 introduces an unique form of multimodality, which is not seen in the artificial problems, due to utilisation of step function. In this case the number of utilised stiffeners can be either 2,3,4,5,6,7 or 8, leading to seven separate local optima as illustrated in Fig. 6.8. For CLPT3, Fig. 6.7, the closest problem is DTLZ8 [40], which also contains two parts of the front: linear, degenerated front; and a planar one. However, according to presented results a strong bias towards the linear part of the front can be observed in CLPT3, but this characteristic does not occur in DTLZ8. Furthermore, in the DTLZ8 the degenerated front is due to inclusion of constraints, while in the CLPT3 there is no constraints. Regarding the 3DSA problem, 3 constraints are used, while the benchmarking functions are using maximum 2 at max; and more variables are utilised, 68



(A) CLPT2

(B) ZDT1, ZDT4, UF1-3, MOP1-2 and MOP5

FIGURE 6.6: Resemblance of the Pareto optimal fronts obtained on the CLPT2 problem with selected benchmarking functions.



(A) CLPT3

(B) DTLZ8

FIGURE 6.7: Resemblance of the Pareto optimal fronts obtained on the CLPT3 problem with DLZ8 function.

comparing to typical 10-30. Furthermore, obtained Pareto optimal fronts on the 3DSA case, shows that neither algorithm was able to reach the true Pareto optimal front. Therefore, it is indicating a significant complexity of this problem's search space in comparison to the artificial constrained problems, where most algorithms with a strong focus on the diversity have been able to properly approximate the final front.

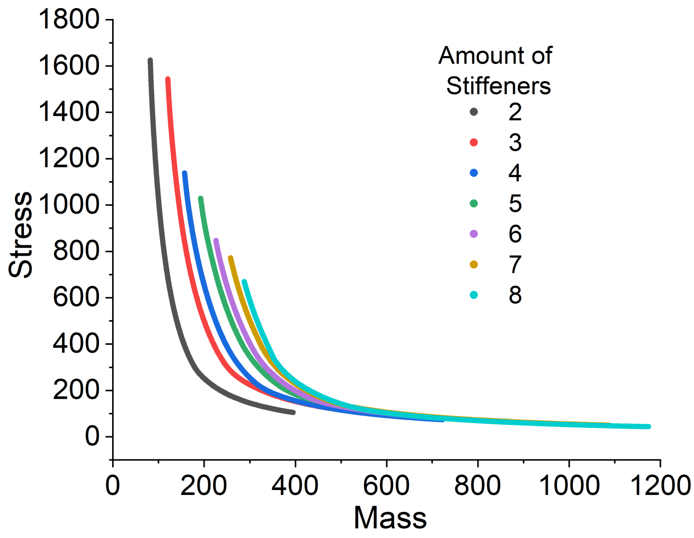


FIGURE 6.8: The Local optima of the CLPT2 problem

6.4 Diversity and generality in the current state-of-the-art

In this chapter the current state-of-the-art is evaluated on over 100 benchmarking problems and 3 practical cases. This is used to define: preferred problem-types for each methodology; the best general-solver genetic algorithm; and if the diversity and generality of the methodology has any impact on the practical applications. It has been found that the best general-solver methodologies are cMLSGA, HEIA and U-NSGA-III, whereas MTS and MOEA/D-based methodologies have more specialist approach. For MTS, the constrained and discontinuous problems are preferred; for MOEA/D the highest performance is observed for the continuous and imbalanced problems. For the general-solvers the preferred areas of applications also can be observed: biased, discontinuous and diversity-hard constrained problems for the cMLSGA and U-NSGA-III; continuous and unconstrained problems for HEIA; whereas BCE and IBEA algorithms performs poorly on most of the tested problems.

Testing on practical problems have shown that general-solver genetic algorithms are preferred for all tested cases, whereas specialist-solvers struggles to find a wide range of solutions, even on theoretically "preferred" problems. Therefore, it is likely that those problems possess more than one dominant characteristic, or they may be defined differently to the benchmarking sets. These findings are further supported by similar results obtained by other researchers while evaluating multiple GAs on the engineering problems [62, 146, 147].

Chapter 7 Conclusions

7.1 Discussion

The results show that the high diversity of MLSGA is achieved via a unique sub-regional search. It is developed by using different fitness definitions on each level of selection and an additional selection pressure resulting from the collective reproduction. Both mechanisms are necessary to maintain this unique behaviour. The collective-level creates virtual boundaries, by strongly penalizing the solutions in certain regions and pushing individuals into the preferred ones, where each collective has a different favoured region. As the collectives are eliminated unless they maintain the diversity designated by the fitness definitions, even if those are more converged on the Pareto optimal front, this leads to the promotion of diversity over convergence. Furthermore, due to high independence of collectives they are able to more easily move around the "gaps" in the objective space created by the constraints or the imbalanced regions on them. Therefore, the search is less likely to get "stuck" in unfavourable regions, similarly to other sub-population-based approaches with high independence of groups [46,47], leading to a better overall diversity of the results, and thus a higher performance on those problems. According to this, it is suggested that MLSGA is moving from the "convergence-first" to the "diversity-first" approach. In other methodologies with structured populations, such as island-model [46], social-model [148–150] and spatial-distance [151, 152] based algorithms, the sub-populations have a persistent character and the genetic information exchange is encouraged in the form of co-operation or competition between groups, showing certain similarities with the competitive-based approach utilised within the MLSGA. However, in those algorithms, the between-group mechanisms are based on the same principles as the individual reproduction, as no fitness separation is used, thereby no additional evolutionary pressure into different regions is developed that would induce a higher diversity. Furthermore, as indicated in [109] those methodologies usually require a hyper-parameter tuning to operate effectively on diverse types of problems, unlike the MLSGA.

In decomposition-based methodologies, such as MOEA/D [35], MOEA/D-MSF [83] and MOEA/D-PSF [83], the diversity is obtained by breaking down the optimised problem into a set of separate search spaces, each assigned with a distinct sub-population, via predefined weight vector. Therefore, diversity is dictated by that weight vector, and thus is highly dependent on its quality, as indicated by other researchers [47, 79–82]. Ideally, it should resemble the true Pareto optimal front as closely as possible and be uniformly spread. This is demonstrated in Section 6.2 by a high performance of the MOEA/D based algorithms on continuous and imbalanced cases. It is due to those algorithms being initially tested only on those cases, as further shown by their low performance on other kinds of problems. Therefore, the results contribute towards considering them as specialist-solvers and it is suggested that the continuous problems are preferred cases for the MOEA/D based algorithms, while the imbalanced problems favour decomposition of the problem. In the MLSGA all individuals operate on the same space, and no hyper-parameter tuning is required for the algorithm to operate effectively, as the diversity is obtained via competition between collectives rather than forced by the predefined weight vector. Furthermore, the MLSGA is more capable of avoiding the infeasible regions in constrained and discontinuous problems in comparison to decomposition-based methods. This claim is supported by high gains in performance of the MOEA/D-based variants of MLSGA over the implemented algorithms on those cases. Whereas, the MOEA/D based algorithms struggle to operate in non-continuous search spaces. In MOEA/D the weights are defined in straight lines which may pass through the regions without feasible solutions, leading to an inefficient search.

According to the conducted tests on over 100 test problems and 3 practical cases, detailed in Chapter 6, there is a correlation between the average performance on the artificial problems and the effectiveness on engineering cases, as detailed in Table 7.1. It can be seen that cMLSGA is the best on average on two- and three-objective artificial problems and it is the best algorithm for the engineering cases as well, while HEIA [36] and U-NSGA-III [43] usually come second or third on both types of problems. This shows a high usefulness of genetic algorithms with a more general performance. Therefore, by testing across 100 test problems, it was possible to gain a good performance on the real cases. Yet, only 2 of the current state-of-the-art algorithms, NSGA-II and MOEA/D, were tested on that scale, as shown in Section 2.3. However, most of those benchmarks are coming from other authors whom are benchmarking against those algorithms, 94 for NSGA-II and 52 problems for MOEA/D. Therefore, the results from those studies were not impacting the development of those algorithms. The high generality of the cMLSGA is due to utilisation of multiple search strategies in the co-evolutionary approach as discussed in Chapter 5. Similar principle applies to HEIA, where Simulated Binary Crossover (SBX) is combined with the Differential Evolution based crossover (DE) with Immune Algorithm based re-population of different subgroups [36], leading to a higher generality. Interestingly, BCE [103] shows poor performance on artificial benchmarking, but has a good performance on real problems, further supporting the importance of more general approach. Lowered performance on engineering problems in comparison to other general-solvers is due to the choice of evolutionary algorithms utilised in BCE. It is combination of the indicator-based strategy, similar to IBEA [121], and the decomposition-based strategy, similar to MOEA/D [35], and both are performing badly on the engineering problems. The high generality of U-NSGA-III is due to the fact that the search is not highly constrained, unlike in the decomposition-based methods. In MOEA/D each individual is hard bounded to the particular weight-vector, which dictates its convergence

Table 7.1: The average rankings on two- and three-objective cases compared to the average performance on engineering cases.

Position	**Benchmarking problems**		**Engineering problems**
	Two-objectives	**Three-objective**	
1	cMLSGA	cMLSGA	cMLSGA
2	HEIA	U-NSGA-III	HEIA
3	U-NSGA-III	HEIA	U-NSGA-III
4	MOEA/D-MSF	MOEA/D	BCE
5	MOEA/D	MOEA/D-MSF	MOEA/D-MSF
6	MTS	MOEA/D-PSF	MOEA/D
7	MOEA/D-PSF	IBEA	MOEA/D-PSF
8	IBEA	MTS	MTS
9	BCE	BCE	IBEA

and diversity, thus the region of exploration, while in U-NSGA-III the individuals have more freedom of exploration.

The results presented in Chapter 6 show that high diversity is also needed for successful engineering applications. In that case, MLSGA_HEIA is outperforming the standard HEIA, despite using the same reproduction mechanisms. This is especially noticeable on the 3D spatial arrangement of leisure boats (3DSA), where MLSGA-hybrid is obtaining significantly better results, whereas the difference in performance is not as distinct while solving the structural design of composite stiffened plates (CLPT2). It is because the 3DSA case contains multiple constraints, resulting in high quantities of infeasible regions and discontinuities of the search space, and thus a higher complexity. Furthermore, it is indicated in Chapter 6 that CLPT3 case is biased towards the linear part of the front. This is explaining the low performance of convergence-based solvers on CLPT3 and 3DSA cases, as it is known that these characteristics require a higher diversity of the search [29, 44].

Interestingly the results indicate that there may be no correlation between performance on distinct categories of artificial problems and engineering cases of similar type. According to the presented data, in Chapter 6: MTS should obtain high performance on 3D spatial arrangement of leisure boats case (3DSA), as it is highly effective on constrained benchmarking cases; MOEA/D-based algorithms should provide good results on unconstrained and imbalanced problems such as structural design of composite stiffened plates (CLPT2 and CLPT3); while HEIA should struggle on highly constrained problems, but it is not valid according to the presented results. It is suggested that the typical distinction used in artificial benchmarks, such as constrained/unconstrained, continuous/discontinuous or imbalanced/biased; do not reflect the complexity of the real-world problems. Therefore, it is likely that these cases possess more than one dominant characteristic or those may be defined differently to the benchmarking sets, leading to a higher complexity of the real-world cases and thus requirement for a better diversity. This may partly explain a low performance of the specialist-solver GAs on the engineering problems. Development of specialist-solvers is based on assumption that benchmarking functions are representative of the practical problems; or at least a similar real-problems to them exists, and thus an improvement of the performance on these cases will lead to better results on the latter. The lack of correlation between the engineering and similar artificial problems indicates that this assumption is not valid. However, the limited data provided in this thesis, one constrained and two unconstrained cases with no details regarding the characteristics of optimised engineering cases, does not allow to draw any overarching conclusions in this regard.

Therefore, the discussed data supports the hypothesis that the general-solver, diversity oriented genetic algorithms are preferred for the optimisation of real-world problems. It is suggested that it is due to: their higher chances of success on problems with different characteristics than the benchmarking problems; need for a high diversity of the search on more complex cases, especially with discontinuous spaces and multiple constraints; and the lack of needing a prior adjustment to the characteristics of optimised problems, which are usually not known in advance.

7.2 Limitations

Despite the discussed findings, the presented work has several limitations due to chosen scope of work, time constraints and limitations of the currently available literature and the state-of-the-art.

Firstly, this thesis shows that the high diversity and generality is preferred for the engineering applications, but it is not known if one of them is more important or if both are of the same significance. Therefore, here it is suggested that both should be considered while developing new genetic algorithms. However, if one of them is more essential, this knowledge would allow to further focus a future development, leading to even better methodologies. This understanding is restrained by two aspects: quality of the benchmarking functions and quality of developed performance indicators. As discussed in Section 2.3, most of the diversity-hard test problems, such as DAS-CMOP [44] or CF [31], are developed by the same groups of researchers, or have similar fitness functions definitions, as their non-diversity-hard counterparts. Therefore, those may promote certain solutions rather than a higher diversity of the search by itself. In this thesis, two quality indicators are selected: IGD, with a stronger focus on uniformity and convergence of solutions and HV, with a stronger focus on spread and diversity of points; from among over 100 indicators developed [139]. However, all indicators focus on the quality of the final solutions, rather than the search in overall. Therefore, those indicators may answer if the final solutions are converged/well-spread/uniform, but do not provide much information if the methodology has a higher focus on diversity of the search rather than convergence.

Secondly, it is not known how the findings of this thesis apply to problems outside of engineering. On one hand, cases such as dynamic problems, are shown to require a high convergence over diversity [104]. This applies even for constrained functions, which usually require a higher diversity, as presented and discussed in Appendix C. On the other hand, it is shown that many scientific problems from biology and chemistry, have discontinuous search spaces [153–156], and thus these may require even higher diversity. Furthermore, it is indicated that for some of them, due to associated measurement error, convergence on the global optimum is not as important [155].

Thirdly, it is indicated that tested practical problems may have multiple main characteristics or that these may be different from the benchmarking problems. However, it is not known what those characteristics are; if those are different than for the artificial cases; and if the engineering problems have any of them in common.

7.3 Future work

Due to discussed findings and limitations of this thesis the following approaches are suggested for consideration in the future:

- Investigating the characteristics of engineering problems. This should be done by mathematical analysis and full landscape analysis of the problems, including ruggedness and shape of the fitness landscape; number of local optima; time to reach a local optimum; the distance between different optima and to the global optimum; the number and localisation of the infeasible regions; and mapping between search and objective spaces; in a similar manner to [44,157,158]. This will allow to group the engineering problems in an analogous manner to the classification used in Chapter 5, and thus to design new benchmarking problems more accurately, which would resemble the practical cases to more extent. It is suggested that, if the benchmarking problems would represent the practical cases more accurately, there would be no need to test algorithms on 100 problems to develop novel search mechanisms, which would be effective for solving the practical problems.
- Testing the genetic algorithms on a wider range of practical problems, not only from engineering. This will allow defining, which course of development is preferred for a particular branch of science or industry, similarly to the diversity/generality for engineering cases and convergence for dynamic problems, as discussed before. It should be followed by an investigation of the characteristics of those problems, leading to a higher applicability of new genetic algorithms to the real-world problems.
- Development of a new metric and a new set of test problems created specifically to evaluate the diversity or modifying the currently existing ones to serve that purpose. This would allow to better define the "diversity-first" algorithms, and to evaluate if the diversity or convergence is more preferred in practical applications. The metric should be a measurement of the overall diversity of the search, not only of the final solutions. It is suggested that the coverage of the fitness landscape or the search space can be used for that purpose. Potential solution could be a combinatorial metric based on the sparseness indicator from the Novelty Search algorithms [113,114], to evaluate the diversity across search space; and the NSGA-II's crowding-distance [34], for evaluation

of the objective space. For new test sets, it is suggested that the findings from previous two points can be used to better tailor "diversity-hard" problems.

7.4 Concluding Remarks

The current state-of-the-art is dominated by specialist-solver genetic algorithms with a strong emphasis on high convergence. However, those approaches are rarely applied to the real-world problems, but a more general solutions are used. This thesis investigates if the methodology with high diversity and generality of the search is preferred for the practical optimisation, due to a high complexity of those cases.

To investigate this hypothesis Multi-Level Selection Genetic Algorithm is extended to develop the "diversity-first" general-solver algorithm. This is performed in three main steps: first it is investigated if the collective-level mechanisms and the fitness separation of MLSGA can be used to promote a higher diversity; then the hybrid-approach is proposed, where reproduction mechanisms from the current state-of-the-art are implemented into MLSGA in order to improve the overall performance while retaining the diversity in each collective; and finally, multiple reproduction mechanisms are combined in the co-evolutionary approach for a higher generality of the search, leading to the cMLSGA variant.

It is found that the proposed cMLSGA algorithm is outperforming the current state-of-the-art on all tested practical problems. Furthermore it is shown that the currently promoted "convergence-first" specialist-solvers such as MTS [93], MOEA/D [35], MOEA/D-MSF [83] and MOEA/D-PSF [83] are less suitable for the practical problems than other methodologies with a higher diversity and generality, such as HEIA [36], BCE [103] and U-NSGA-III [43]. It is concluded that proposed algorithm was successful in developing the "diversity-first" general-solver approach by utilisation of the additional level of selection and split in the fitness function, and that the high generality and diversity of the search strategy is needed for successful real-world applications. Therefore, a higher emphasis should be given on the development of such methodologies. Furthermore, the development of new algorithms should be followed by testing on a wide range of artificial and practical problems in order to maximise the chance of success on the practical problems.

Appendix A Re-inspiring the genetic algorithm with multi-level selection theory: multi-level selection genetic algorithm

Bioinspiration & Biomimetics

PAPER

Re-inspiring the genetic algorithm with multi-level selection theory: multi-level selection genetic algorithm

To cite this article: A J Sobey and P A Grudniewski 2018 *Bioinspir. Biomim.* **13** 056007

IOP Publishing Bioinspir. Biomim. 13 (2018) 056007 https://doi.org/10.1088/1748-3190/aad2e8

Bioinspiration & Biomimetics

RECEIVED
3 July 2018

ACCEPTED FOR PUBLICATION
12 July 2018

PUBLISHED
31 July 2018

PAPER

Re-inspiring the genetic algorithm with multi-level selection theory: multi-level selection genetic algorithm

A J Sobey[1] and P A Grudniewski

Fluid Structure Interactions, University of Southampton, University Rd, Southampton SO17 1BJ, England, United Kingdom

[1] Author to whom any correspondence should be addressed.

E-mail: ajs502@soton.ac.uk (A J Sobey)

Keywords: multi-level selection, multi-objective optimisation, single objective optimisation, evolutionary theory, algorithm development

Abstract

Genetic algorithms are integral to a range of applications. They utilise Darwin's theory of evolution to find optimal solutions in large complex spaces such as engineering, to visualise the design space, artificial intelligence, for pattern classification, and financial modelling, improving predictions. Since the original genetic algorithm was developed, new theories have been proposed which are believed to be integral to the evolution of biological systems. However, genetic algorithm development has focused on mathematical or computational methods as the basis for improvements to the mechanisms, moving it away from its original evolutionary inspiration. There is a possibility that the new evolutionary mechanisms are vital to explain how biological systems developed but they are not being incorporated into the genetic algorithm; it is proposed that their inclusion may provide improved performance or interesting feedback to evolutionary theory. Multi-level selection is one example of an evolutionary theory that has not been successfully implemented into the genetic algorithm and these mechanisms are explored in this paper. The resulting multi-level selection genetic algorithm (MLSGA) is unique in that it has different reproduction mechanisms at each level and splits the fitness function between these mechanisms. There are two variants of this theory and these are compared with each other alongside a unified approach. This paper documents the behaviour of the two variants, which show a difference in behaviour especially in terms of the diversity of the population found between each generation. The multi-level selection 1 variant moves rapidly towards the optimal front but with a low diversity amongst its children. The multi-level selection 2 variant shows a slightly slower evolution speed but with a greater diversity of children. The unified selection exhibits a mixed behaviour between the original variants. The different performance of these variants can be utilised to provide specific solvers for different problem types when using the MLSGA methodology.

1. Applications using genetic algorithms

There is a large literature pertaining to applications and developments of evolutionary algorithms. Coello Coello [1] has collected 10 714 references relating to genetic algorithms and a google scholar search of the term 'genetic algorithm' for 2017 supplied 37 600 examples using or developing these algorithms. These references cover a wide range of areas related to finding optimal solutions to complex problems across a wide range of different types of search space. A recent review by Zhou *et al* [2] lists 50 major applications for evolutionary algorithms. This is augmented by recent high profile examples that utilise these algorithms from a number of different fields of study, including architecture, bioinformatics, computational science, evolutionary theory, environmental science and materials engineering ([3–14]). The problems being solved are becoming more complex and therefore promising methods for the improvement of genetic algorithms should be further explored and documented. This research investigates the adoption of multi-level selection, a popular explanation for the mechanisms of evolution, as a potential inspiration to improve evolutionary computation.

1.1. Evolutionary inspiration

Evolutionary algorithms were originally proposed by Turing [15] and were first developed by Holland ([16] and [17]) to study evolution through simulations. The usages of genetic algorithms quickly expanded due to their ability to search large and complex spaces, with a corresponding increase in the number of available algorithms. Since these initial attempts, there has been limited success in improving genetic algorithm performance using evolutionary inspired methods. A recent benchmarking of top algorithms in the CEC'09, [18], show no biologically or evolutionary inspired mechanics and a recent review of the state-of-the-art in biomimetics does not list evolution in the top 100 topics [19]. However, other algorithms have benefited from developments that more closely mimic the original concepts that inspired them, such as the ant colony optimisation developed by Zhang [20].

In parallel with the improved performance in genetic algorithms, there have been developments in evolution theory, which have led to a larger range of mechanisms available to explain how organisms evolve; many researchers now believe that some of these mechanisms are necessary for biological evolution. These proposed mechanisms are missing from the genetic algorithm and the authors believe they should be used to re-inspire it; if they are critical to evolution then they should provide benefits to algorithm development. Amongst the newer evolutionary theories is the multi-level selection theory, originally proposed by Sober and Wilson [21], with the idea that evolution does not occur at only one level but is actually occurring at different levels of a hierarchical structure. An example of a hierarchical structure is shown in figure 1 where evolution at multiple levels is shown using collectives, larger units, which contain a number of individuals, smaller units; darker shading of the individuals or the collectives is used to illustrate higher fitness. As the generations progress the collectives may continue, and possibly reproduce, or be eliminated, depending on their ability to survive. Within the surviving collectives, the individuals also reproduce or are eliminated, based on their fitness. The strongest individuals generally have more offspring and this leads to an increase in average fitness. Whilst the authors believe the process makes intuitive sense, and there is a growing body of evidence to support the idea that it increases evolutionary speed, the theory of multi-level selection is not without controversy. There are many arguments for other mechanisms through which evolution might occur, such as Dawkin's popular selfish gene theory [22] which emphasises a gene-centric level of evolution, or Sterelny and Kitcher [23] which questions whether there must be a uniquely correct identification of the level at which selection is occurring. The authors believe multi-level selection provides the most promising inspiration for improvements to the genetic algorithm, allowing a collective mechanism to be put in parallel with current individual mechanisms and allowing more flexibility in the manner in which the fitness function can be interpreted. Initial research by the authors investigating one variant of multi-level selection shows promise [24] and can be used to solve practical problems [25] but further investigation into the inspiration behind these mechanisms needs exploring. This paper therefore documents the behaviour of different variants of this algorithm to provide specific solvers for different problem types when using the multi-level selection genetic algorithm (MLSGA) methodology. These will exploit the unique reproduction mechanisms at each level of MLSGA and the split in the fitness function.

1.2. Multi-level selection theory

Natural selection encourages individuals to adapt to the environment developing traits that increase their chance of survival; it is therefore common to consider adaptions at the individual level, for example, giraffes with longer necks can reach leaves on trees that others cannot. Multi-level selection proposes that it is rare for a population to compete as individuals without some kind of beneficial interaction; often a number of individuals will group together with the aim of helping each other. This can be in the form of a number of organisms grouping into a tribe or pack but it can also be at a lower level where a number of organelles create a hybrid form. An argument is often made to support the statement that there are characteristics that have developed which cannot be explained satisfactorily at the individual level, for instance altruism; which is a trait that does not benefit the individual and, by definition, actually results in a cost to itself. Sober and Wilson [21] suggested that evolution of altruists can be observed at a level higher than the individual by watching the development of colonies.

Multi-level selection is an outcome of three determinants based around those proposed by Lewontin [26], which holds that evolution occurs when:

1. Different individuals in a population have physical variations.
2. Different individuals have different rates of survival due to different fitness.
3. There is a correlation between parents and offspring so that fitness is heritable.

Natural selection occurs if there are variations in characteristics of individuals and some offspring are considered to be fitter than others. Individuals with these different characteristics can be defined as a 'unit of selection'. Individuals are encouraged to adapt over time to the environment developing traits that increase their chance of survival. This causes evolution as weaker members of the population are lost and the stronger members, on average, survive. Gradually the make-up of the population will change over successive generations to favour the stronger characteristics. A species can be described based on a hierarchical organisation, which is normally viewed as a nested hierarchy, with one level being enclosed within another. A nested hierarchical organisation is presented in figure 1 which forms the basis of the MLSGA. McShea [27, 28] pro-

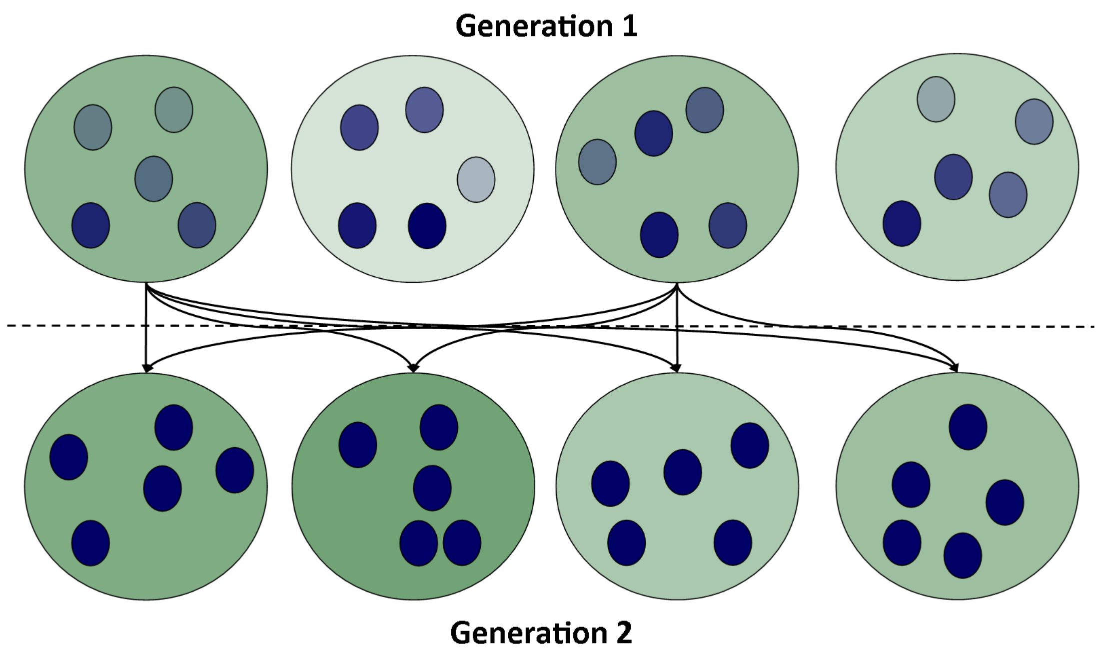


Figure 1. Nested hierarchical collective reproduction.

Table 1. Important concepts of multi-level selection.

	Individual level	Collective level
Hierarchical unit	Number of individuals in each collective	Number of collectives composed of individuals
Character	The qualities shown by the individual which can affect its fitness e.g. speed from body type	Collective characters are the qualities of the collective that can affect its fitness and can be the average or total character of individuals (aggregate, MLS1), e.g. the aggregate speed of each individual, or different from individual's character (emergent, MLS2) with independent qualities, e.g. the ability to communicate abstract concepts
Fitness	Individual fitness measured directly on individuals	Collective fitness measured the same as the individual (aggregate, MLS1) or differently to individual's (emergent, MLS2).
Heritability	Individuals with higher fitness produce more offspring with characters correlated to the parent individuals	Collectives with higher fitness leave more collectives with characters correlated to the parent collectives.

poses an interactionist approach indicating that selection at a higher level will occur when individuals at a lower level perform fitness-affecting interactions with each other, a physical connection is not required, and this is supported by Sober and Wilson [21] in their theory of multi-level selection. The new algorithm is developed by creating a hierarchical organisation, including collective level reproduction. However, since classical genetic algorithms already utilise selection of individuals based on their adaption to their environment it is therefore possible to utilise a standard genetic algorithms as the reproduction method at the individual level.

In addition to the collective level reproduction mechanism the new algorithm relies on a change to the way fitness is defined. Amongst the theories for multi-level selection, there are two main strands, multi-level selection 1 and 2 (MLS1 and MLS2) which are dependent on how fitness is defined on the individual and collective levels. For multi-level selection 1 the fitness is defined the same way at each level, the aggregate of the individuals. For multi-level selection 2 the fitness is defined differently at each level. For multi-level selection to occur there are a number of important concepts defined in table 1 including the unit evolution is working on at each hierarchical level, the character or abilities of a unit used to determine the fitness, the fitness used to judge survival for each unit and finally the implications of these factors on the heritability of the unit. Both of these theories have proponents and a large background of literature with which to support these views. For a larger and more in-depth review the authors recommend the excellent text by Okasha [29] outlining the history, theory and gaps in multi-level selection research.

1.3. Comparison to the state-of-the-art

In reviewing algorithms specifically pertaining to multi-level selection there are already a few attempts to use the process as inspiration to improve evolutionary algorithms, but these fail to replicate the key aspects of this theory and the resulting performance is poor.

Lenaerts *et al* [30] were the first to use multi-level selection and they study a biological model based on multi-level selection. It shows the importance of variation between groups as selection at group level occurs only when there is enough variation. However, the mechanics to generate selection at different levels are prescribed based on interactions between individuals, as opposed to fitness. Akbari *et al* [31] and [32] looked at multilevel selection including selection at different levels. However, they do not present group, collective, mechanisms such as characters, fitness and heritability, necessary for multilevel selection. The algorithm relies on a number of complex mechanisms to replicate the sub-problem generation and individual replacement. Finally, Wu and Banzhaf [33] focused on selections using any hierarchical model. However, in multilevel selection, selection at one level influences the selection at the adjacent levels and the concepts of multi-level selection, such as units of selection at all levels and the products induced by selections between levels, are not obviously applied. The authors feel that these efforts miss the key aspects of multi-level selection in that they are orientated around complex prescriptive mechanisms, forcing the selection to occur rather than letting it emerge as part of the process. As these mechanisms do not replicate the key aspects of multi-level selection the authors provide a novel evolutionary algorithm, supported by the fact that the results provide an improved performance compared to those previously documented.

Furthermore, multiple different methodologies in the current state of the art show similarities to the idea of a collective and fitness split, such as sub-population approach and problem decomposition. In Niching algorithms, such as NSGA-II [34] and U-NSGA-III [35], Co-evolutionary algorithms, for example BCE [36] and HEIA [37] and Island model algorithms proposed by [38] the population is divided into groups with often different operations performed on each sub-population. However, between these groups, only one level of selection is used with no competition between sub-groups and split in the fitness function, unlike MLSGA. The decomposition-based methods as MOEA/D [39], MOEA/D-M2M [40], CS-NSGA-II [41] DMOEA-DD [42], LiuLi [43], RVEA [44] and K-RVEA [45] operate by partitioning the entire objective space into subspaces, dividing the problem into a number of separate subproblems each with its own subpopulation. This is done by using a set of uniformly spread reference points, or weight vectors, and scalarization functions. However, the effectiveness of these methods decreases for discontinuous problems as subpopulations can be assigned to regions where feasible solutions do not exist. *A priori* knowledge about the objective and search spaces are therefore required to adjust the mechanisms. In MLSGA, the sub-regional search is created using different fitness definitions, instead of forced decomposition, and therefore all individuals operate on the same region.

Introduction of selection at multiple levels provides the inspiration for the development of a new algorithm for optimisation, multi-level selection genetic algorithm (MLSGA), adding multi-level concepts to the classic genetic algorithm. Current genetic algorithms consider the evolution of its individuals at a single level similar to the manner in which Holland [16] first developed the algorithm. It is proposed here that if multi-level selection speeds the process of evolution then its addition will elicit an increase in performance in the genetic algorithm, speeding up the rate at which higher fitness solutions will be found. This evolutionary inspiration has been used by the authors to develop an algorithm, multi-level selection genetic algorithm but there are a number of different variants of multi-level selection theory and these are explored on single- and multi-objective problems to categorize any differences in behaviour.

2. Methodology

Outlined are the mechanisms that are used, inspired by the concept of multi-level selection, and the methodology used, showing the adaptations from the classic genetic algorithm. This is documented for the two variants of multi-level selection theory to show the differences in performance, the unified theory and also to the commonly used NSGA-II algorithm and also to the commonly used NSGA-II algorithm and MOEA/D developed by Zhang and Li [39]. NSGA-II was created by Deb *et al* [34] and is the most commonly used genetic algorithm but is also selected to represent a niching algorithm. The specific method selected is the 2011 updated version which shows similar results on 2 objective functions to the more mature U-NSGA-III. MOEA/D developed by Zhang and Li [39], represents sub-region search algorithms, and is the highest performing genetic algorithm for unconstrained test functions, for multi-objective problems. In these cases the hyperparameters are taken from the CEC'09 benchmarking with no additional tuning [18].

2.1. Multi-level selection genetic algorithm (MLSGA)

The genetic algorithm starts with a randomly generated population of individuals each of which represents a set of variables representing the search space for an optimisation problem. This population is then evaluated against a fitness function. Once the variables for each individual have been assessed, and the fitness determined, a new generation can be created from the current parent generation. Whatever the specific selection methodology chosen the process involves finding the fittest individuals within a population to mate and produce a child generation. These children are produced by the process of crossover where the chromosome of the parents is split. The chromosomes are then combined forming a new offspring different to the parent generation. Further diversity is found

Table 2. Multi-level selection algorithm methodologies.

Step	Parameter	Value
1. Initialisation	Type	Random
	Encoding	Real values
	Pop. Size	600
2. Classification	Method	SVM
	Number of collectives	6
3. Individual reproduction		
Fitness evaluation	Type	MLS1 or MLS2 individual fitness
Selection	Type	Roulette wheel
Mating	Crossover type	Real variable SBX
	Crossover rate	0.7
	Mutation type	Polynomial mutation
	Mutation Rate	0.08
Elitism	Rate	0.1
4. Collective reproduction		
Fitness evaluation	Type	MLS1 or MLS2 collective fitness
Elimination	Number of eliminated collectives	2
Replacement	Number of new collectives	2
5. Termination	Criterion	Reaching 500 generations

through mutation of the chromosome, where random changes are made normally based on a probability value, which is chosen to be low. Once this process has been repeated for the whole population a new population is then ready for the process to be repeated. Over successive generations, the average fitness of the population will become lower until either the optimal value or the limit of the number of generations is reached. This replicates a simplified process of evolution first developed to simulate the process but since then many adaptions and improvements have been made.

The multi-level selection genetic algorithm is similar to previous genetic algorithms except that it utilises the key concepts from table 1, defining fitness and heritability, through reproduction mechanisms, at different hierarchical units, individuals and collectives. The main addition is therefore the collective, which contains a number of individuals and has a fitness dependent on the character of those individuals.

1. Initialisation—a population of individuals is created at random.
2. Classification—sorted into collectives of the most similar individuals, in this case using a support vector machine (SVM). The quality of the result is dependent on a good spread of results between the collectives where SVM is used over clustering algorithms to allow the user control over the size of the collectives.
3. Individual reproduction—As the algorithm progresses the individuals inside each of the collectives are treated in the same manner as for the standard genetic algorithm so that the process of selection and mating continues as normal.
4. Collective reproduction—the difference is that the collectives themselves have a fitness and elitism process where the least fit are eliminated and the fittest reproduce. Hyperparameter tuning is performed which shows that the results are relatively insensitive to the new parameters. Large or small numbers of collectives are shown to provide poor performance with 6 giving the best performance for these problems and 4 or 8 collectives giving similar but worse results. In this case 2 collectives are eliminated and 2 are created from the old surviving collectives with the individuals being populated from the fittest 4 collectives, where the number of new individuals in the offspring collective is equal to the number of individuals in the eliminated one. Individuals are generated by replicating the best individuals, according to the collective fitness function, equally from the remaining collectives. As an example in the case where there are 6 collectives and 2 are eliminated then if there are 40 individuals in the first eliminated collective and 60 in the second then the best 10 individuals from the remaining 4 collectives are replicated and added to the first new collective and the next 15 best are then taken from the remaining 4 collectives and are added to the second new collective.
5. Termination—The algorithm terminates, in these cases after a given number of generations.

Table 3. ZDT test Functions.

Problem	Objective functions	Comments
ZDT1 [47]	$f_1(x) = x_1; f_2(x) = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}}\right]$	Convex,
	$g(x) = 1 + 9\frac{\sum_{i=2}^{n} x_i}{n-1}$	continuous
ZDT2 [47]	$f_1(x) = x_1; f_2(x) = g(x)\left[1 - \left[\frac{x_1}{g(x)}\right]^2\right]$	Nonconvex,
	$g(x) = 1 + 9\frac{\sum_{i=2}^{n} x_i}{n-1}$	continuous
ZDT3 [47]	$f_1(x) = x_1$	Convex
	$f_2(x) = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)}\sin(10x_1)\right]$	Disconnected
	$g(x) = 1 + 9\frac{\sum_{i=2}^{n} x_i}{n-1}$	
ZDT4 [47]	$f_1(x) = x_1, f_2(x) = g(x)\left[1 - \sqrt{\frac{x_1}{g(x)}}\right]$	Convex,
	$g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}[x_i^2 - 10\cos(4x_i)]$	large search space
ZDT6 [47]	$f_1(x) = x_1 - \exp(-4x_1)\sin^6(6x_1)$	Nonconvex
	$f_2(x) = g(x)\left[1 - \left[\frac{x_1}{g(x)}\right]^2\right]$	Non-uniformly
	$g(x) = 1 + 9\left[\frac{\sum_{i=2}^{n} x_i}{n-1}\right]^{0.25}$	spaced

This process, with the specific details used in this paper for each stage, are documented in table 2.

2.1.1. MLS1

The two types of multi-level selection differ in the manner in which the fitness function is represented. In MLS1 the fitness of the collectives is the aggregate of the sum of the individuals and is illustrated using the ZDT1 function where $f_1(x)$ is,

$$f_1(x) = x_1, \tag{1}$$

and $f_2(x)$, is,

$$f_2(x) = 1 + \frac{9}{n-1}\sum_{i=1}^{n} x_i\left(1 - \sqrt{\frac{x_1}{1 + \frac{9}{n-1}\sum_{i=1}^{n} x_i}}\right), \tag{2}$$

where variables are in the range $0 \leqslant x_i \leqslant 1$, n is the number of variables. The first objective function is arbitrarily taken as $f_1(x)$ and the second, remaining, objective function is $f_2(x)$. This means that the same fitness function can be used to determine the selection in the collectives as well as the individuals. The fitness function for the MLS1 individuals and collectives is,

$$\text{Fitness} = \frac{f_1(x) + f_2(x)}{2}. \tag{3}$$

2.1.2. MLS2

The MLS2 theory states that the fitness for the collectives is different to the fitness for the individuals. Therefore, this variant only allows the use of multi-objective optimisation problems or the generation of an abstract objective for single objective problems. For the cases chosen here, the fitness function has been split into two with one of the objectives forming the fitness for the individuals,

$$\text{Individual Fitness} = f_1(x), \tag{4}$$

and the other forming the fitness function for the collective,

$$\text{Collective Fitness} = f_2(x). \tag{5}$$

The effect of reversing this order is also investigated to determine whether it affects the results of the optimisation and is designated MLS2R, i.e. in equations (4) and (5) individual fitness is assigned to be f_2 and collective fitness f_1.

2.1.3. MLS-U

Contradicting the need to define MLS1 or MLS2, Sterelny and Kitcher [23] indicate that a single mechanism for selection does not need to be defined at each level. United Multi-Level Selection (MLS-U) is introduced, inspired by this definition, exhibiting characteristics of both MLS types. In MLS-U all MLS variants are used in parallel where some collectives only utilise MLS1, MLS2 or MLS2R, maintaining an equal number of collectives of each type.

2.1.4. Computational complexity of one generation

The computational cost of the MLSGA is determined by three operations: the PF selection, individual reproduction and collective reproduction. The PF selection identifies nondominated individuals from 1.5N members at most, aggregate of external population of maximum size equal to the overall

Table 4. Comparison of multi-level selection algorithms for weighted average optimisation.

Test function	Fitness Max.	Min.	Mean	Standard deviation	Convergence (Generations)
MLS1					
ZDT1	**0.375**	**0.375**	**0.375**	0	**49**
ZDT2	**0.500**	**0.500**	**0.500**	0	**34**
ZDT3	**0.039**	**0.039**	**0.039**	0	**100**
ZDT4	**0.375**	**0.375**	**0.375**	1.90×10^{-06}	**500**
ZDT6	**0.500**	**0.500**	**0.500**	0	**145**
MLS2					
ZDT1	0.500	0.584	0.508	0.017	500
ZDT2	**0.500**	0.558	0.505	0.013	412
ZDT3	0.500	0.583	0.506	0.016	500
ZDT4	3.38	32.8	16.7	7.58	500
ZDT6	0.516	0.601	0.567	0.025	500
MLS2R					
ZDT1	**0.375**	**0.375**	**0.375**	0	169
ZDT2	**0.500**	**0.500**	**0.500**	0	57
ZDT3	**0.039**	**0.039**	**0.039**	1.00×10^{-06}	368
ZDT4	**0.375**	**0.375**	**0.375**	1.19×10^{-04}	**500**
ZDT6	**0.500**	0.501	**0.500**	4.11×10^{-04}	377
MLS-U					
ZDT1	0.378	0.393	0.388	2.84×10^{-03}	500
ZDT2	**0.500**	**0.500**	**0.500**	0	299
ZDT3	**0.039**	**0.039**	**0.039**	0	465
ZDT4	**0.375**	**0.375**	**0.375**	4.70×10^{-05}	**500**
ZDT6	**0.500**	0.501	**0.500**	4.33×10^{-04}	479

population size N and current collective of maximum size of 0.5 N, requires $O(m(1.5\ \mathrm{N})^2)$ comparisons using fast nondominated sorting approach, where m is the number of objectives [34]. In individual reproduction the full replacement of old generation with elitims take place which require $O(N^2)$ at most, as only one of the objectives is considered. For collective reproduction, the fitness evaluation step requires $O(N)$ computations, as is calculated as the average of single fitnesses of the individuals and collective replacement of $O((0.5N)^2)$ complexity at most, due to maximum size of the collective.

To summarize, the overall computational compexlity of one generation of MLSGA is bounded by $O(mN^2)$.

3. Characterisation of multi-level selection variants

The results of genetic algorithms developed using the two strands of multi-level selection, MLS1 and MLS2, are compared to each other for the different test functions from Ziztler *et al* [46], shown in table 3, on 30 separate runs. These are popular in testing genetic algorithms due to the wide range of near optimal points which are difficult to find, and which the classical genetic algorithms cannot solve. Whilst there are more complex functions available, these are not selected at this stage, as it is difficult to identify differences between variants on these problems. Furthermore, these problems have clear differences in complexity for both fitness functions, where f_2 is harder, making it easier to investigate the differences between MLS2, MLS2R and MLS1. For each of these functions the number of variables used is 30, except in the case of ZDT4 where 10 are used. The variable bounds are $[0, 1]$ for all cases except in ZDT4 where these are: $x_1 = [0, 1], x_{2..10} = [-5, 5]$. The stopping criterion is when a value of 300 000 function evaluations, generations multiplied by population size, is reached. If the optimal result is not found at this stage then the best fitness to this point is used as the minimum fitness value. To demonstrate the accuracy of each algorithm results show the minimum, the maximum and the average optimal results found over the all the runs. The robustness of the algorithm is investigated to determine the percentage of runs over which the algorithm found the optimal result.

3.1. Weighted average optimisation—evolutionary speed

A weighted average optimization is used to demonstrate the algorithm working for single objective problems. MLS2 and MLS1 become the same for a true single objective problem, $f_1 = f_2$, however, in the weighted average case two objectives still exist and so MLS2 can still be examined, even though it still splits the function into two components but is trying to find the solution illustrated in equation (3), conversely MLS1 considers all problems as a weighted average. The results are shown in table 4 with the best performing algorithm shaded for each category other than the standard deviation.

The results show that the MLS1 function rapidly finds the solution to all of these problems, except for ZDT4; in the case of the ZDT4 function a close to best value is found. A similar performance is seen for the MLS2R function but with a slightly greater variation in the solutions for the more complex functions, ZDT3-6, finding the best solution 43% of the time for the ZDT3 function and 77% for the ZDT6 function. The MLS2 variant performs extremely poorly for most of the functions only finding the best solution for the ZDT2 function and in this case only 30% of the time. The unified variant, MLS-U, exhibits a similar performance to the MLS1 and MLS2R variants and outperforms MLS2, in terms of the final accuracy of the Pareto front. However, the results take considerably more generations to form than for the other variants, especially MLS1. The results demonstrate the manner in which the MLS1 algorithm dives towards the Pareto front reaching it quickly, in single objective optimisation or scenarios where the optimal value is not of so much interest and the near optimal is required the MLS1, with a low generation cut off, would provide

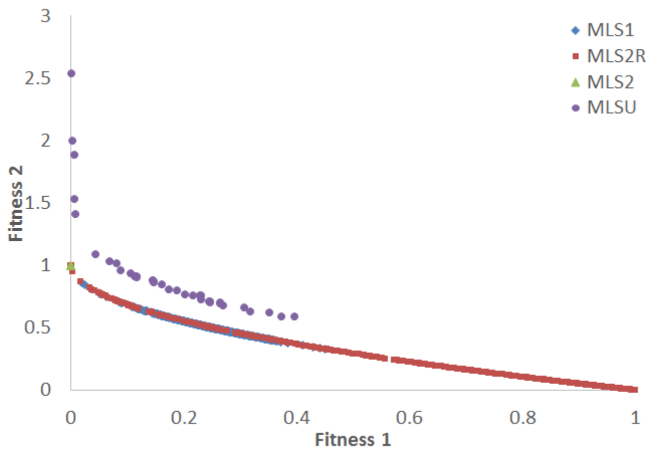


Figure 2. Pareto front comparison at 50 generations ZDT1 function.

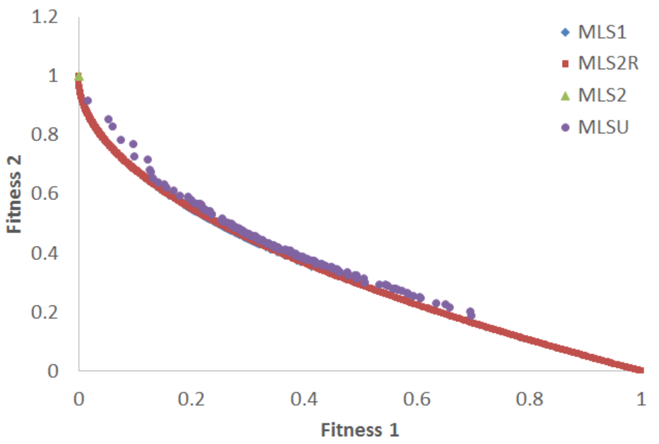


Figure 3. Pareto front comparison at 100 generations ZDT1 function.

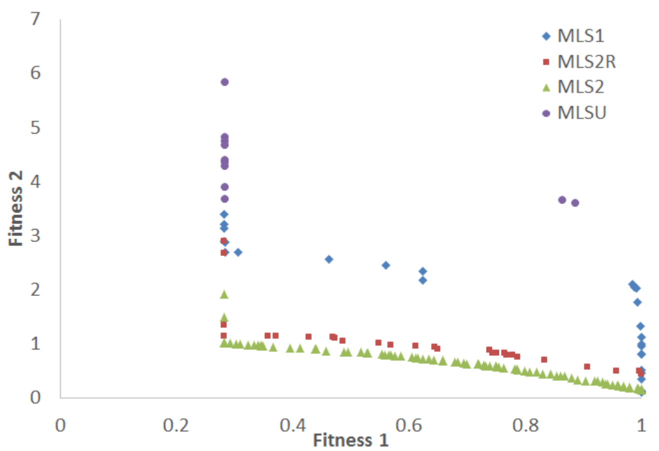


Figure 4. Pareto front comparison at 50 generations ZDT6 function.

an excellent solution in few generations. For the MLS2 the choice of function for the individual and collective is important in determining the performance with one variant finding the Pareto front rapidly and the other struggling to find any optimal solutions. Computationally the new algorithm is rapid and the only extra computational effort over a standard genetic algorithm is the generation and elimination of the collectives.

3.2. Multi-objective optimisation—evolutionary diversity

The speed and accuracy with which the optimal point is found is interesting and useful for a subset of problems.

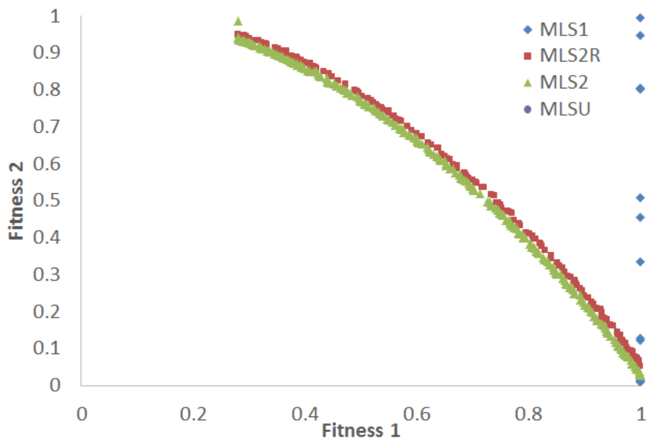


Figure 5. Pareto front comparison at 100 generations ZDT6 function, where the MLS-U results are not visible due to poor convergence.

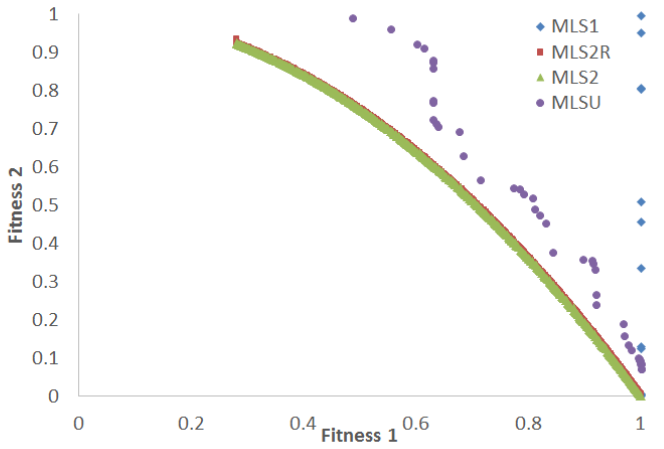


Figure 6. Pareto front comparison at 250 generations ZDT6 function.

However, the benefits of evolutionary algorithms lie in their ability to assess multi-objective cases where a spread of optimal points is required. Whilst these fronts are useful they can be difficult to generate, as the algorithm must be accurate while ensuring an even spread of results to ensure that sharp variations in the front are found. The algorithm is adapted from the weighted-average results by storing the 400 best values in the Pareto front the objective functions are evaluated in the same way and only the objectives against which they are judged are changed, i.e. the spread of results is now evaluated unlike in the previous section. In this case, the 3 algorithms, including the 4 variants of MLSGA, are used to generate a Pareto front for the previously defined functions. The results for ZDT1-4 and 6 are found and are illustrated for the MLSGA variants at 50 and 100 generations for the ZDT1 case at 50 generations, figure 2, and 100 generations, figure 3, to demonstrate the difference in performance. This is shown again for the more complex ZDT6 function in figure 4, for 50 generations, figure 5, 100 generations, and figure 6, 250 generations. Blue diamonds represent MLS1, red squares represent MLS2, green triangles represent MLS2R and purple circles represent MLS-U.

The Pareto front evolution shows a large difference between the algorithms at the early generations. However, by the time 100 generations have been completed then the algorithms have all found, or are close to finding the Pareto front. Figure 3 demonstrates the speed with which the Pareto front is found. In this case, the MLS2R algorithm finds the front rapidly with a good spread of results. The MLS2 algorithm finds a few points on the top left hand part of the front, but is unable to provide a greater diversity of points. The MLS1 algorithm finds some points along the front rapidly and is the first to reach it but struggles to diversify these with extra points along this front. The MLS-U shows the worst speed for reaching the front, and never reaches it in over the 30 runs. This is because the MLS-U variant uses 3 different methods simultaneously where each has approximately only a third of the overall population. In this case, the evolutionary pressure is not as concentrated as when only a single variant is used and results in slower progression.

The process is repeated for the more complex ZDT6 functions where all of the algorithms are slower to find the front, shown in figures 4–6. The MLSGA variants find the front with the MLS2 and MLS2R

Table 5. Comparison of multi-level selection algorithms for multi-objective optimisation.

Test function	IGD Max.	Min.	Mean	Std. deviation
MLS1				
ZDT1	0.115 00	0.180 00	0.160 00	1.6×10^{-02}
ZDT2	0.118 00	0.251 00	0.158 00	3.1×10^{-02}
ZDT3	0.029 00	0.081 00	0.051 00	1.3×10^{-02}
ZDT4	0.029 00	0.153 00	0.098 00	3.6×10^{-02}
ZDT6	0.137 00	0.468 00	0.294 00	1.0×10^{-01}
MLS2				
ZDT1	0.840 00	0.881 00	0.846 00	1.2×10^{-02}
ZDT2	0.609 00	0.725 00	0.619 00	2.4×10^{-02}
ZDT3	0.820 00	0.962 00	0.835 00	3.4×10^{-02}
ZDT4	9.8	62.3	34.4	13.4
ZDT6	**0.003 13**	**0.003 34**	**0.003 20**	5.10×10^{-05}
MLS2R				
ZDT1	**0.003 90**	**0.004 03**	**0.003 95**	3.80×10^{-05}
ZDT2	**0.003 89**	**0.004 03**	**0.003 96**	3.00×10^{-05}
ZDT3	**0.004 77**	**0.005 61**	**0.004 99**	1.40×10^{-04}
ZDT4	**0.003 88**	**0.004 14**	**0.004 01**	6.70×10^{-05}
ZDT6	0.003 16	0.003 97	0.003 40	2.60×10^{-04}
MLS-U				
ZDT1	0.005 67	0.027 70	0.015 85	5.28×10^{-03}
ZDT2	0.004 05	0.394 42	0.146 10	1.08×10^{-01}
ZDT3	0.029 03	0.174 26	0.078 54	4.92×10^{-02}
ZDT4	0.004 43	0.061 26	0.007 76	1.00×10^{-02}
ZDT6	0.003 97	0.101 47	0.055 35	3.98×10^{-02}

finding a good spread of results. The MLS2 in this case is able to find the front more rapidly than the MLS2R. The MLS1 algorithm finds some on the front but many of the points are a considerable distance from the front. The MLS-U results show a slower progression to form the Pareto front to the extent that the results for figure 5 are not visible as they are out of range, resulting from a change in axes from figure 4 to figure 5. Figure 6 shows the results after 250 generations where MLS-U has not found the front due to the lower evolutionary pressure exhibited by this variant.

Table 5 illustrates the IGD values for the different algorithms with the best results in bold, MLS-U is ignored because of its poor performance. The MLS2R algorithm provides slightly better results in almost all cases except for the ZDT6 function, where MLS2 provides better results. This raises a question about how the collective and individual functions should be defined. It appears that the collective level mechanism is much weaker than that of the individual. Improvements to the collective level mechanism should result in a considerable increase in performance.

Table 6 presents the best results from MLS2, the best performing variant, using the best result from MLS2 or MLS2R, making the assumption that the correct algorithm is selected. These results show an increase in performance for MLSGA over the most popularly used GA, NSGA-II, and the top algorithm on unconstrained functions, MOEA/D. This is despite the fact that the algorithm uses a simple mechanism at the individual level and shows the potential for this method. In one case the MOEA/D algorithm has a better mean performance that MLSGA, but NSGA-II never shows better performance.

Table 6. Comparison of MLSGA with the state-of-the-art.

Test function	IGD Max.	Min.	Mean	Std. deviation
MLSGA				
ZDT1	**0.003 90**	**0.004 03**	**0.003 95**	3.80×10^{-05}
ZDT2	**0.003 89**	**0.004 03**	**0.003 96**	3.00×10^{-05}
ZDT3	**0.004 77**	0.005 61	**0.004 99**	1.40×10^{-04}
ZDT4	**0.003 88**	**0.004 14**	**0.004 01**	6.70×10^{-05}
ZDT6	**0.003 13**	**0.003 34**	**0.003 20**	5.10×10^{-05}
NSGA-II				
ZDT1	0.004 07	0.004 25	0.004 16	5.30×10^{-05}
ZDT2	0.004 01	0.004 20	0.004 09	4.80×10^{-05}
ZDT3	0.004 88	0.005 21	0.005 10	8.60×10^{-05}
ZDT4	0.004 09	0.255 00	0.057 30	7.70×10^{-02}
ZDT6	0.003 40	0.003 61	0.003 51	5.00×10^{-05}
MOEA/D				
ZDT1	0.004 07	0.004 22	0.004 14	3.20×10^{-05}
ZDT2	0.004 06	0.004 17	0.004 10	2.90×10^{-05}
ZDT3	0.004 85	**0.005 03**	0.005 00	4.10×10^{-05}
ZDT4	0.004 09	0.004 15	0.004 13	1.50×10^{-05}
ZDT6	0.003 35	0.003 45	0.003 37	2.20×10^{-05}

4. Evolutionary performance

Amongst the theories for multi-level selection, there are two main strands, multi-level selection 1 and 2 (MLS1 and MLS2) which are dependent on how fitness is defined on the individual and collective levels. For multi-level selection 1 the fitness is defined the same way at each level, the aggregate of the individuals, which results in deep specialisation of individuals, and greater evolutionary pressure thus increasing the rate of evolution. For multi-level selection 2 each level has different 'goals', which results in competition between individuals and collectives, and leads to increased diversity of population as a compromise between the two goals is found. Importantly separate groups, across the whole environment, can have different fitness definitions at each level, similar to species where each becomes specialised, developing different traits. According to Okasha [29], multi-level selection is proposed to enhance the specialization of different sub-groups, and this correlates with the results found here.

The results show that the introduction of MLS1 or MLS2 provides an increase in performance over the classic genetic algorithm with respect to both optimisation and generation of the Pareto front. This provides some promising behaviour on some simple test functions when judged against that of its contemporaries, as an example NSGA-II and MOEA/D. The MLS2R algorithm outperforms the other algorithms in the generation of Pareto fronts on the simple test functions proposed. The Pareto front itself is found rapidly but, more importantly, the diversity is high whereby the Pareto front has a good spread of results. This is in contrast to the MLS1 code, which searches one specific zone to a high degree but does not develop any front, it finds a close-to-optimal location rapidly but there is a lack of spread of results. MLS1 shows the best performance on single objective problems. Whilst on the multi-objective problems the MLS1 code can outperform the MLS2 in rapidly finding near optimal results the performance is not that much faster than the MLS2R and the absolute optimum is never found. The MLS2R performs strongly in both tasks with a wide spread of different optimal areas investigated showing a wide diversity. Interestingly the MLS2 algorithm outperforms the MLS2R algorithm for the ZDT6 function, in contrast to the weighted fitness results. The difference between MLS2 and MLS2R results are caused by the fact that individual selection is more efficient than collective selection, and the f_2 functions are more complex than the f_1 functions in most of the test problems, this is reversed in the ZDT6 function. The authors suggest that the individual level should have the more complex function assigned to it. MLS-U is outperformed only by the best variants, MLS2R for the ZDT1-4 functions and MLS2 for ZDT6, and shows better results than the remaining variants. This sensitivity can be removed by utilisation of a unified MLS-U approach. MLS-U, despite a decrease in performance in comparison to the best variants, does not require extensive *a priori* knowledge about the optimised problem and eliminates the necessity of performing the optimisation process repeatedly in order to find the best possible solutions, in the case when close-to-the-best result is sufficient. However, in many real world cases the complexity of the problem will be well understood, through domain knowledge or simple computational tests, meaning that the most powerful variant can be selected. Improvements to the novel collective level reproduction mechanism will further reduce this sensitivity and reduce the hyperparameters associated with MLSGA.

Whilst the authors propose the MLSGA method they are surprised at the increase in robustness, diversity and evolutionary speed generated from the simple process of considering the fitness at multiple levels. The optimal fitness is found rapidly and the average fitness of the populations is high, making it suitable for finding Pareto fronts. The authors hesitate to draw overarching conclusions into evolutionary theory based on a biologically inspired algorithm however, it is interesting to note the increase in speed with which the generations find higher fitness solutions due to the increased evolutionary pressure. Furthermore, there are considerable differences in behaviour between the MLS1, MLS2 and MLS2R algorithms and the resultant differences in fitness. The MLS1 seems to find the optimal value correctly but lacks the diversity of children seen in the MLS2 algorithm. In this case, the MLS1 has a concentrated search in which there are two layers to remove individuals that are unfit. In MLS2 this phenomenon is weaker but still provides a concentrated search towards the individual's objective function, many of the poor solutions are removed at each generation pushing the search quickly towards the Pareto front but using the collectives to retain some diversity; on finding the Pareto front the search then rapidly spreads out increasing the range of solutions. This also differs from other currently high performing, computationally inspired, algorithms that first aim to create a diverse set of solutions that push towards the optimal front, with the potential to perform well on discontinuous fronts. While it may be shown that multi-level selection is not the process by which evolution should be considered, the emergent change in the optimisation process is interesting and beneficial.

The test functions that have been selected are simple and the early results on the ZDT6 function already show that the MLSGA algorithm is starting to struggle to find the optimal solution with the more complex functions. This set is used to allow an easy comparison in performance but the ZDT1-4 functions have a special structure, with f_1 being only determined by a single decision variable, while all other decision variables are zero for all Pareto optimal solutions. Further iterations of the algorithm will move away from evolutionary inspiration and utilise more complex individual and collective reproduction methods to improve performance based on the findings here and this will include determining the sensitivity of performance to the various hyper-parameters. Algorithms such as NSGA-II and MOEA/D are shown to outperform the original GA, currently used at the individual level, by a considerable margin, it is proposed that their inclusion at the individual level will increase performance. The effectiveness of the new collective reproduction will also be explored using simple search functions, such as hill climb, and evolutionary algorithm mechanisms, such as MTS, to boost the effectiveness of the collective search.

5. Conclusions

There are a number of evolutionary mechanisms proposed in recent years, which have not been explored in genetic algorithms. One of these, multi-level selection, has been used to inspire a new genetic algorithm but the different variants of this theory have not been compared. Interestingly the two variants of the algorithm, MLS1 and MLS2 show considerable differences in behaviour instigated by the definitions of fitness. The algorithm performs well on the simple ZDT functions with the MLS2 finding a wide diversity of results along the front and the MLS1 algorithm rapidly finding the weighted fitness optimum, demonstrating excellent single-objective performance. These different variants allow the MLSGA methodology to have different behaviour for different sets of problems, allowing strong performance on multi-objective and single objective problems.

Acknowledgments

The authors would like to thank the Lloyd's Register Foundation for their continued support of the LRF UTC in Ship Design for Enhanced Environmental Performance at the University of Southampton and without whom this research would not have been possible. We would also like to thank Dr. Yikun Wang for her kind support and effort in helping us prepare this manuscript, LCdr Suttipong Paksuttipol for his early work developing this algorithm and Dr. Richard Watson for his useful interrogation of the underlying concepts.

ORCID iDs

A J Sobey https://orcid.org/0000-0001-6880-8338
P A Grudniewski https://orcid.org/0000-0003-0635-3125

References

[1] Coello Coello, C.A., personal website http://delta.cs.cinvestav.mx/~ccoello-/EMOO/EMOOstatistics.html accessed:2016-05-11

[2] Zhou A, Qu B, Li H, Zhao S-Z, Suganthan P and Zhang Q 2011 Multiobjective evolutionary algorithms: a survey of the state of the art *Swarm Evolutionary Comput.* **1** 32–49

[3] Menges A 2012 Biomimetic design processes in architecture: morphogenetic and evolutionary computational design *Bioinspiration Biomimetics* **7** 1–10

[4] Bounds D 1987 New optimization methods from physics and biology *Nature* **329** 215–9

[5] Barhen J, Protopopescu V and Reister D 1997 TRUST: a deterministic algorithm for global optimization *Science* **276** 1094–7

[6] Wales D and Scheraga H 1999 Global optimization of clusters, crystals and biomolecules *Science* **285** 1368–72

[7] Glick M, Rayan A and Goldblum A A 2002 A stochastic algorithm for global optimization and for best populations: a test case of side chains in proteins *Proc. Natl Acad. Sci.* **99** 703–8

[8] Nowak M and Sigmund K 2004 Evolutionary dynamics of biological games *Science* **303** 793–9

[9] Schoups G, Hopmans J, Young C, Vrugt J, Wallender W, Tanji K and Panday S 2005 Sustainability of irrigated agriculture in the san joaquin valley, california *Proc. Natl Acad. Sci.* **102** 15352–6

[10] Menon V, Spruston N and Katha W 2009 A state-mutating genetic algorithm to design ion-channel models *Proc. Natl Acad. Sci.* **106** 16829–34

[11] Carter W 2010 Structure predictions: the genetics of grain boundaries *Nat. Water* **9** 383–5

[12] Tran D and Johnston R 2011 Study of 40-atom pt-au clusters using a combined empirical potential-density functional approach *Proc. R. Soc.* A **467** 2131

[13] McNally L, Brown S and Jackson A 2012 Cooperation and the evolution of intelligence *Proc. R. Soc.* B **279** 3027–34

[14] Srinivasan B, Vo T, Zhang Y, Gang O, Kumar S and Venkatasubramanian V 2013 Designing DNA-grafted particles that self-assemble into desired crystalline structures using the genetic algorithm *Proc. Natl Acad. Sci.* **110** 18431–5

[15] Turing A 1950 Computing machinery and intelligence *Mind* **49** 433–60

[16] Holland J 1975 *Adaption in Natural Artificial Systems* (Ann Arbor: University of Michigan Press)

[17] Holland J 2013 Genetic algorithms *Sci. Am.* **267** 66–72

[18] Zhang Q and Suganthan P 2009 Final report on CEC'09 MOEA competition *IEEE Congress on Evolutionary Computation CEC'09 Proc., Trondheim, Norway*

[19] Lepora N, Verschure P and Prescott T 2013 The state of the art in biomimetics *Bioinspiration Biomimetics* **8** 1–11

[20] Zhang Z, Gao C, Liu Y and Qian T 2014 A universal optimization strategy for ant colony optimization algorithms based on the physarum-inspired mathematical model *Bioinspiration Biomimetics* **9** 1–14

[21] Sober E and Wilson D 1999 *Unto Others: Evolution and Psychology of Unselfish Behaviour* (Cambridge, MA: Havard University Press)

[22] Dawkin C 1976 *The Selfish Gene* (Oxford: Oxford University Press)

[23] Sterelny K and Kitcher P 1988 The return of the gene *J. Phil.* **85** 339–61

[24] Grudniewski P and Sobey A 2017 Multi-level selection genetic algorithm applied to CEC'09 test instances *IEEE Congress on Evolutionary Computation (San Sebastian)* pp 1613–20

[25] Wang Z, Bai J, Sobey A, Shenoi R and Xiong J 2018 Optimal design of triaxial weave fabric composites under tension *Compos. Struct.* **201** 616–24

[26] Lewontin R 1970 The units of selection *Ann. Rev. Ecol. Syst.* **1** 1–18

[27] McShea D 1996 Metazoan complexity and evolution: is there a trend? *Evolution* **50** 477–92

[28] McShea D 1998 Possible largest-scale trends in organismal evolution: eight 'live hypotheses' *Ann. Rev. Ecol. Syst.* **29** 293–318

[29] Okasha S 2006 *Evolution and the Levels of Selection* (Oxford: Clarendon)

[30] Lenaerts T, Defaweux A, Remortel P and Manderick B 2003 Modelling artificial multi-level selection *AAAI Spring Symp. on Computational Synthesis (AAAI Spring Symposium Series) AAAI Technical Report SS-03-02*

[31] Akbari R, Zeighami V and Ziarati K 2010 MLGA: A multilevel cooperative genetic algorithm *IEEE 5th Int. Conf. on Bio-Inspired Computing: Theories and Applications* pp 271–7

[32] Ziarati K and Akbari R 2011 A multilevel evolutionary algorithm for optimizing numerical functions *Int. J. Ind. Eng. Comput.* **2** 419–30

[33] Wu S X and Banzhaf W 2010 A hierarchical cooperative evolutionary algorithm *Proc. of the 12th Annual Conf. on Genetic and Evolutionary Computation*

[34] Deb D, Pratap A, Agarwal S and Meyarivan T 2002 A fast and elitist multiobjective genetic algorithm: NSGA-II *IEEE Trans. Evolutionary Comput.* **6** 182–97

[35] Seada H and Deb K 2015 U-NSGA-III: a unified evolutionary optimization procedure for single, multiple and many objectives: proof-of-principle results *Evolutionary Multi-Criterion Optimization: 8th Int. Conf. Proc., Part II (Guimarães, Portugal, 29 March–1 April 2015)* pp 34–49

[36] Li M, Yang S and Liu X 2016 Pareto or Non-Pareto: Bi-criterion evolution in multiobjective optimization *IEEE Trans. Evolutionary Comput.* **20** 645–65

[37] Lin Q, Chen J, Zhan Z-H, Chen W-N, Coello Coello C A, Yin Y, Lin C-M and Zhang J 2016 A hybrid evolutionary immune algorithm for multiobjective optimization problems *IEEE Trans. Evolutionary Comput.* **20** 711–29

[38] Whitley D, Rana S and Heckendorn R 1998 The island model genetic algorithm: on separability, population size and convergence *J. Comput. Inf. Technol.* **1** 33–47

[39] Li H and Zhang Q 2007 MOEA/D: a multiobjective evolutionary algorithm based on decompostion *IEEE Trans. Evolutionary Comput.* **11** 712–31
[40] Liu H-L, Gu F and Zhang Q 2014 Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems *IEEE Trans. Evolutionary Comput.* **18** 450–5
[41] Branke J, Schmeck H, Deb K and Reddy S M 2004 Parallelizing multi-objective evolutionary algorithms: cone separation *Proc. of the 2004 Congress on Evolutionary Computation* (Piscataway, NJ: IEEE) vol 2 pp 1952–7
[42] Liu M, Zou X, Yu C and Wu Z 2009 Performance assessment of DMOEA-DD with CEC 2009 MOEA competition test instances *IEEE Congress Evolutionary Computation CEC'09 Proc., Trondheim, Norway* pp 2913–8
[43] Liu H L and Li X 2009 The multiobjective evolutionary algorithm based on determined weight and sub-regional search *IEEE Congress on Evolutionary Computation* pp 1928–34
[44] Cheng R, Jin Y, Olhofer M, Sendhoff B and Member S 2016 A reference vector guided evolutionary algorithm for many-objective optimization *IEEE Trans. Evolutionary Comput.* **20** 773–91
[45] Chugh T, Chakraborti N, Sindhya K and Jin Y 2017 A data-driven surrogate-assisted evolutionary algorithm applied to a many-objective blast furnace optimization problem *Mater. Manuf. Process.* **32** 1172–8
[46] Zitzler E, Laumanns M and Thiele L 2001 SPEA2: improving the strength pareto evolutionary algorithm *TIK-Report* 103
[47] Zitzler E, Deb K and Thiele L 2000 Comparison of multiobjective evolutionary algorihms: empirical results *Evolutionary Comput.* **8** 173–95

Appendix B Behaviour of Multi-Level Selection Genetic Algorithm (MLSGA) using different individual-level selection mechanisms

Swarm and Evolutionary Computation 44 (2019) 852–862

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Behaviour of Multi-Level Selection Genetic Algorithm (MLSGA) using different individual-level selection mechanisms☆

Przemyslaw A. Grudniewski *, Adam J. Sobey

Fluid Structure Interactions, University of Southampton, Southampton, SO17 1BJ England, UK

ARTICLE INFO

Keywords:
Evolutionary computation
Genetic algorithms
Hybrid GA
Multi-level selection
Multi-objective optimisation

ABSTRACT

The Multi-Level Selection Genetic Algorithm (MLSGA) is shown to increase the performance of a simple Genetic Algorithm. It is unique among evolutionary algorithms as its sub-populations use separate selection and reproduction mechanisms to generate offspring sub-populations, called collectives in this approach, to increase the selection pressure, and uses a split in the fitness function to maintain the diversity of the search. Currently how these novel mechanisms interact with different reproduction mechanisms, except for the one originally tested at the individual level is not known. This paper therefore creates three different variants of MLSGA and explores their behaviour, to see if the diversity and selection pressure benefits are retained with more complex individual selection mechanisms. These hybrid methods are tested using the CEC′09 competition, as it is the widest current benchmark of bi-objective problems, which is updated to reflect the current state-of-the-art. Guidance is given on the new mechanisms that are required to link MLSGA with the different individual level mechanisms and the hyperparameter tuning which results in optimal performance. The results show that the hybrid approach increases the performance of the proposed algorithms across all the problems except for MOEA/D on unconstrained problems. This shows the generality of the mechanisms across a range of Genetic Algorithms, which leads to a performance increase from the MLSGA collective level mechanism and split in the fitness function. It is shown that the collective level mechanism changes the behaviour from the methods selected at the individual level, promoting diversity first instead of convergence, and focuses the search on different regions, making it a particularly strong choice for problems with discontinuous Pareto fronts. This results in the best general solver for the updated bi-objective CEC′09 problem sets.

1. Multi-level Selection Genetic Algorithm

Multi-level Selection Genetic Algorithm (MLSGA) is a genetic algorithm incorporating advanced evolutionary concepts. It is inspired by Multi-Level Selection Theory which is used to describe evolution, with these concepts originally being based on selection in ant colonies [1]. Multi-Level Selection Theory states that evolutionary selection can be considered at more than one level, for example the probability of survival for a wolf is determined as dependent on its own abilities but also that of its pack. Based on these ideas, MLSGA introduces a collective level that groups individuals, which has similarities to sub-populations in other algorithms. However, separate reproduction and elimination mechanisms are introduced at the collective level, which provide additional selection pressure, in contrast to typical sub-population approaches that use reproduction mechanisms only at the individual level. Multi-level selection theory defines two ways in which the fitness can be defined at the collective level: MLS1, where the fitness of the collectives is simply the aggregate of the individuals, and MLS2, where the collective fitness is based on a separate, emergent, property rather than the summation of the fitnesses of the individuals inside of it. MLS2 therefore has different reproduction mechanisms operating on different parts of the fitness function; the behaviours of these variants are explored in Ref. [2]. Inside each collective individuals are treated in the same manner as the populations in the standard Genetic Algorithm and there is no ability to share the information between the sub-groups. The algorithm provides additional evolutionary pressure through the increase in the number of reproduction mechanisms, one at collective level and one at individual level, but unlike in other algorithms where multiple mechanisms are utilised [3], diversity is not lost due to the separation of the fitness functions.

☆ This research was supported by Lloyd's Register Foundation.
* Corresponding author.
E-mail addresses: pag1c15@soton.ac.uk (P.A. Grudniewski), ajs502@soton.ac.uk (A.J. Sobey).

https://doi.org/10.1016/j.swevo.2018.09.005
Received 13 May 2018; Received in revised form 17 September 2018; Accepted 20 September 2018
Available online 21 September 2018
2210-6502/

MLSGA has previously been tested by incorporating a simple GA mechanism at the individual level which is based on the original Holland GA but includes elitism [4]. The previous studies show interesting behaviour including an increase in overall evolutionary pressure and in diversity through the dispersion of the search, achieved through specialisation of the collectives each to different regions of the objective space. This emerges "naturally" rather than being forced by decomposition or other mechanisms that require heavy tuning for each new problem. However, the previous implementation lacks leading performance on the test functions which are taken from the 2009 Congress of Evolutionary Computation (CEC′09) [5]. Based on the previous results it is proposed that MLSGA's performance may be improved by implementation of stronger mechanisms at the individual level. Combinations of different methodologies and hybrid approaches have already been shown to be effective when dealing with complicated MOPs at least in part, because they can maintain the advantages of both approaches [6]. However, it is also possible that the stronger individual level mechanisms will dominate, making the collective level reproduction mechanisms redundant, or that the introduced mechanisms will not be compatible and decrease the performance. This is considered to be likely as more recent Genetic Algorithm mechanisms are further from the evolutionary roots of the original algorithm and more complex than the one used previously in Ref. [4]. In addition, some modifications to the original method are necessary in order to incorporate stronger mechanisms at the individual level and it is important to see how these modifications affect the performance of the algorithm. In addition, MLSGA utilises a "diversity first, convergence second" approach, which is uncommon for GAs, and it is proposed that it can be used with other algorithms to retain the diversity of their searches, but this needs investigating.

In this paper the individual selection mechanism in MLSGA is replaced by current state-of-the-art algorithms: NSGA-II [7], MTS [8] or MOEA/D [9], selected as the leading Genetic Algorithm mechanisms and representing three different types of mechanism: niching, distributed search and hierarchical. Three hybrid variants are developed and benchmarked on an updated version of the CEC' 09 comparison which provides complex bi-objective test instances on a range of different evolutionary algorithms. Bi-objective problems are selected to reduce the complexity allowing easier visualisation of the behaviour of the algorithms. Additional mechanisms are developed in order to make the selected individual mechanisms compatible with MLSGA and a comprehensive hyper-parameter tuning is performed in order to adjust the existing collective-level mechanisms to the state-of-the-art algorithms, with guidance provided for the optimal parameters.

This paper is organised as follows: section 2 presents a literature review of GAs utilising sub-populations; section 3 introduces MLSGA mechanisms in detail and the principles of the conducted benchmark; section 4 benchmarks the MLSGA hybrids against the updated CEC '09 competition; section 5 presents a discussion of experimental results and observed behaviours, followed by conclusions in section 6.

2. Current sub-population mechanisms

Reviewing the current literature, a number of genetic algorithm methodologies show similarities to the idea of a collective; for examples those that utilise sub-population mechanisms including niching, hierarchical, co-evolution and island algorithms [6], in addition to some previous approaches inspired by multi-level selection.

Niching algorithms such as NSGA-II [7] and U-NSGA-III [10] utilise sorting mechanisms, which rank the whole population, depending on the non-dominance level or other indicators such as IBEA [11]. However, different sub-groups are not allowed to cooperate or compete as in MLSGA, and no separate mechanisms are applied to these groups. The Island model mechanism proposed by Ref. [12] separates the population into sub-groups and cooperation is introduced by allowing migration of the individuals between neighbouring groups but only in a single direction; only one level of selection is used between sub-groups with no competition between them. In hierarchical algorithms such as MOEA/D [9], MOEA/D-M2M [13], CS-NSGA-II [14] DMOEA-DD [15] and LiuLi [16] sub-groups operate separately on different sub-regions of the search space, but with no additional selection mechanisms between sub-groups. These decomposition mechanisms implement a number of additional parameters, which are far from trivial to determine, but the effects on the performance of the algorithm are substantial. Therefore, these methods usually require a priori knowledge about the objective space and an additional issue is that they cannot usually find points in negative regions. Additionally, in discontinuous problems, large regions can exist without feasible solutions and the use of standard decomposition methods leads to a waste of computational power resulting in poorer final solutions. In MLSGA, the sub-regional search is created using different fitness definitions, instead of decomposition, therefore all individuals operate on the same region. Furthermore, fewer parameters are used, making it simpler to use and requiring less tuning to a specific problem.

Despite many of the hierarchical algorithms showing poor performance on constrained problems DMOEA-DD [15] and LiuLi [16] stand out as they show top performance on these problems in the CEC′09 benchmarking. It was hoped to incorporate these algorithms in this study but the authors find that the literature does not give sufficient information to replicate these codes to a standard that provides the results documented in CEC′09 [17] and the codes provided online do not compile, making their integration with MLSGA impossible.

In the co-evolution mechanisms the population is divided into groups with different operations performed on each sub-population, which is inspired by the idea that two or more species in nature can have a reciprocal evolutionary relationship which increases the rate of evolution; two of the more successful examples are BCE [18] and HEIA [19]. In these algorithms the groups are allowed to exchange individuals and cooperation between groups is introduced. However, the selection process occurs only at the individual level without additional mechanisms or competition at the group level, unlike in MLSGA. In addition, in MLSGA the same mechanisms are applied to each group and each individual, and only the fitness function definitions change in the process.

In addition to the sub-population based mechanisms there are already approaches inspired by Multi-Level Selection Theory but they ignore key aspects of the theory and do not demonstrate an improvement over current methods [20] were the first to use multi-level selection by studying a biological model and altruism. It shows the importance of variation between groups as selection at group level occurs only when there is enough variation. However, in this approach, the groups are destroyed and recreated after each generation where the best individuals are strongly preferred in this process, resulting in limited diversity. MLSGA removes collectives much less regularly, retaining a wider diversity of search. Furthermore, no fitness split is introduced and the selection of individuals is based on interactions between individuals and groups as opposed to their fitness [21,22] looked at multi-level selection including selection at different levels and three additional cooperative mechanisms are introduced. Importantly the collective fitness definition and additional selection levels are implemented in a manner that strongly favours the best solutions and therefore leads to the loss of many good solutions and a reduction in the overall gene pool. Finally [23], focus on selection using a hierarchical model. In the proposed mechanisms selection at one level influences the selection at the adjacent levels and a single selection process is used at different levels without the distinct separation of the fitness values. Therefore, no separate units of selection are introduced at all levels, which is a basic concept of the multi-level selection theory.

In summary, the authors feel that these efforts miss the key aspects of multi-level selection in that they are orientated around complex prescriptive mechanisms, forcing the selection to occur rather than letting it emerge as part of the process. None of the multi-level selection algorithms demonstrates an increase in performance through the incorporation of the additional level and fail to mimic key concepts in the multi-level selection theory. The traditional GAs utilising sub-populations have only one level of selection and no split in the fitness functions, lacking separate reproduction mechanisms at each level that makes MLSGA unique.

3. Methodology

Outlined is a brief review of the MLSGA mechanisms inspired by the concept of multi-level selection, the methodology to integrate different GAs for the so-called hybrid forms and an outline of the benchmarking procedure.

3.1. MLSGA-hybrid

In this work three different genetic algorithms are chosen for testing at the individual level resulting in three distinct hybrids: MLSGA-NSGA-II, MLSGA-MOEA/D and MLSGA-MTS, selected for the following reasons:

1. NSGA-II [7] as a general solver and most commonly utilised algorithm with better performance on bi-objective problems than NSGA-III [24];
2. MOEA/D [9] as the best GA for unconstrained problems according to the CEC′09 comparison [17];
3. MTS [8] as the best available algorithm for constrained problems and representing distributed search algorithms.

The resulting algorithm works as follows with a more detailed description of the MLSGA mechanisms specific for hybridisation below:

Inputs:

- Multi-objective problem;
- Np: Population size;
- Nc: Number of collectives;
- MLSGA specific parameters;
- NSGA-II, MTS or MOEA002FD specific parameters;
- Stopping criterion;

Output: External Population (EP)

Step 1) Initialisation:
Step 1.1) Set *EP = NULL*.
Step 1.2) Randomly generate an initial population *P* of *Np* individuals, *xj,…,xNp.*
Step 2) Classification:
Step 2.1) Classify the individuals in the initial population *P* into *Nc* collectives, *Ci,…,CNc*, so that each contains a separate population *Pi,…,PNc.* Classification is based on the decision variable space using a Support Vector Machine (SVM).
Step 2.2) Assign the fitness definitions from types {MLS1, MLS2, MLS2R - explained in detail below} to each collective so that there is a uniform spread of collectives using each type.
Step 2.3) In the case of the MOEA/D hybrid: Assign the nearest weight vectors *λi* to each individual in each corresponding collective.
Step 3) Individual level operations:
For i = 1, …, Nc do
Step 3.1) Individual level GA's operations: Perform the reproduction, improvement and update steps from NSGA-II [7], MTS [8] or MOEA/D [9], subject to the hybrid variant, over the collective's entire population Pi, documented in the corresponding literature.
Step 3.2) Update External Population:
For j = 1, …, *|Pi| do*
Remove from the EP all solutions dominated by *xij* (the individual *j*, from population *i*). Add individual *xij* to EP if no solutions from EP dominate *xij.*
Step 4) Collective level operations:
Step 4.1) Calculate collective fitness:
For i = *1, …, Nc do*
Calculate the fitness of the collective *Ci* as the average of the fitnesses of population *Pi* based on the fitness definition assigned to that collective.
Step 4.2) Collective elimination:
Find the collective *Ci* with the worst fitness value, and store the index of that collective, *z.*
Store the size of the eliminated collective *|Pz|* as the variable *s.*
In the case of the MOEA/D hybrid: Store the weight vectors of population {*λ* j, …, *λ* size(Pz)} *Pz* as the matrix *Λ z.*
Erase the collective *Cz* with population *Pz.*
Step 4.3) Collective reproduction:
For i = *1, …, Nc do*
if (*i !* = *z*)
Copy the best *s*/(*Nc - 1*) individuals, according to the fitness definition of the eliminated collective *Cz*, from population *Pi* to *Pz.*
then
In the case of the MOEA/D hybrid: assign the weight vector *λz* randomly to population *Pz.*
Step 5) Termination: If the stopping criteria is met, stop and give EP as output. Otherwise, return to **Step 3).**

Table 1
Fitness functions assignment for different MLS types.

Fitness definition type	MLS2	MLS2R	MLS1
Collective level	f2(x)	f1(x)	f1(x) + f2(x)
Individual level	both f1(x) and f2(x)		

A key element to retaining diversity is the split in the fitness function introduced in Step 4, where separate fitness values for the collectives are calculated and utilised. Three types of collectives are introduced, replicating the MLS1 and MLS2 definitions from Multi Level Selection Theory [25] as shown in Table 1, where MLS2R is the reverse of MLS2. In MLS2, the collective fitness function is based on the first objective of the function, f1(x), and is calculated as the average of the first objective function of the individuals inside; MLS2R is based on the average of the second objective f2(x) and in MLS1 it is the sum of both objectives. A strategy where each collective has a different fitness function definition, called MLS-U, is also introduced in Ref. [2] and is shown to greatly increase the diversity of the search and is the mechanism used within this research as it shows the best performance. As MLS1 is an aggregation of both objectives normalization has to be utilised in cases where the objectives are disparately scaled, to increase the solution uniformness. Due to the nature of the selected benchmarking functions, no objective normalization is necessary and is therefore not implemented. However, for cases where it is required the objective normalization strategy taken from Ref. [9] defined in Eq. (1), is recommended:

$$\overline{f_i} = \frac{f_i - z_i^*}{z_i^{nad} - z_i^*} \tag{1}$$

where $z^* = (z_1^*, \ldots, z_m^*)$ is the reference point, i.e. $z_i^* = \min f_i(x) \mid x \in \omega$, $z^{nad} = (z_1^{nad}, \ldots, z_m^{nad})^T$ is the nadir point, i.e. $z_i^{nad} = \max f_i(x) \mid x \in \omega$, $\omega$is the search space, and m is the number of objectives, assuming a minimisation problem.

In the Classification Step a supervised learning classification method, SVM, is used to assign collective labels to each individual in the initial population, based on the distances between them in the decision variable space. In this paper the multi-class classification SVM with C parameter, called C-SVC, and linear function is used. The utilised code has been taken from LIBSVM open library [26] and the original SVM-train parameters have been used. The user predefines the number of label types, and thus the number of collectives. However, the number of individuals in each collective depends only on the classification method and the distribution of the initial population, and therefore is different in each collective. Only the minimum of 10 individuals and maximum half of the overall population size, is predefined in order to avoid empty, small or big collectives as it has been shown in pre-benchmarks to decrease the overall performance. Organising the collectives with the most similar individuals has been shown to be beneficial over random initiation after testing using three different classification variants: SVM, k-means clustering and random assignment. Clustering and SVM exhibit similar performances but the SVM is chosen due to its lower calculation time and higher robustness.

In the case of the MLSGA-MOEA/D hybrid a set of weight vectors of the size of the population is randomly generated. These weight vectors are randomly assigned to the individuals in increasing order, starting from the vector with the smallest value for the first objective. The individuals in the first collective are assigned first, followed by the next collectives until every individual in all the collectives have values assigned to them. In development of the hybrid algorithms it is shown that maintaining the closest neighbourhoods of weight vectors inside of each collective has been shown to be beneficial over a completely random assignment. Calculation and pre-assignment of the best weight vectors to each individual, despite demonstrating the lowest starting fitness, has been shown to have no statistically significant impact on the final performance, while increasing the calculation cost.

Inside each collective NSGA-II, MOEA/D or MTS specific operations are applied to the individuals at each generation, in the same manner as in the original documentation with no further modifications. NSGA-II, MOEA/D and MTS are multi-objective GAs, and require both objective functions to work. In the hybrid approach both f1 and f2 objective functions are utilised at the individual level. Therefore, no fitness function separation is introduced at this level, unlike in the original MLSGA [4].

For the collective elimination, in Step 4.2, the collective with the worst collective fitness value is eliminated and all of the individuals inside are erased. This collective is repopulated in Step 4.3 by copying the best individuals, according to the eliminated collective fitness definition, from all of the remaining collectives. This is done in order to maximise the fitness of the offspring collective. Importantly some information is inherited from the eliminated collective: the size of the population in the collective, in order to maintain constant population size; the collective type; and in the case of MOEA/D the weight vector of the eliminated population. Therefore, no randomness or variation is introduced in these steps.

MLSGA is not a cooperative based GA, rather competitive based one, as there is no direct information transfer between the levels of selection or between sub-groups, such as migration, colonization or regrouping [18,22]. There is only one step in which different sub-groups are able to "communicate" with each other, Step 4.3, where the best individuals are selected in order to recreate the eliminated collectives, however there is no effect on the parent collectives. In between the different levels of selection the only information passed is the fitness of the individuals, necessary to calculate the collective fitness in Step 4.1.

3.2. Computational complexity and constraint handling

The computational cost of the MLSGA-hybrids is determined by two operations: individual reproduction, taken from the embedded algorithms, and the MLSGA collective operations. In this case the individual reproduction has the same complexity as the embedded algorithm, denoted as C, and the collective operations requires $O(mN^2)$ comparisons at most as detailed in Ref. [2]. Therefore, the overall computational complexity of one generation of the MLSGA-hybrids is bounded by $O(mN^2)$ or C whichever is larger.

In this work the complexity of the utilised algorithms is $O(mN^2)$ in the case of NSGA-II, $O(mNT)$ in case of MOEA/D, where T is number of solutions in the neighbourhood and is typically 0.2N, or $O(mN^2)$ in case of MTS. Therefore, the complexity of all proposed hybrids is $O(mN2)$, which is the same or not significantly higher when compared to the original algorithms and the similar computational times exhibited by the algorithms support this.

When constraints are present the constraint-domination principle is adopted for all the MLSGA-hybrid algorithms, taken from NSGA-II [7] and NSGA-III [27], and defined as:

An individual x_1 is said to dominate another individual x_2, if: 1) x_1 is feasible and x_2 is infeasible or, 2) both x_1 and x_2 are infeasible, and x_1 has a smaller constraint violation (CV) value or, 3) both x_1 and x_2 are feasible, and x_1 dominates x_2 with the standard fitness domination principle.

This applies whenever two individuals are compared. However, no direct constraint handling is introduced on the collective-level and therefore, the fitness of the collective is not affected by the infeasibility of individuals inside of it. By implementing the constraint violation penalty at only one level, the individual-level, the diversity of the search can to be maintained avoiding premature convergence of the collectives.

3.3. Benchmarking

Bi-objective problems are selected as a first step as they allow a problem where the behaviour is simple to interpret and study while providing enough complexity to replicate a number of real world problems. The CEC′09 [5] benchmarking test set is selected to illustrate the results due to the number of algorithms compared and range of different problem types, including a substantial set of constrained problems. The unconstrained results are updated to include HEIA and BCE to ensure the results reflect more recent developments and to compare the proposed methodology with similar approaches. NSGA-II has also been improved since the CEC′09 competition, therefore the new results have been run and the tables have been updated to reflect this. NSGA-II is preferred over NSGA-III [24] or U-NSGA-III [10] in this comparison as it has been shown to exhibit better performance on bi-objective problems. Results for MOEA/D on the constrained problems are not included in the original CEC′09 benchmark and so it is benchmarked on these problems and the tables are updated. The newer variants of MOEA/D, such as MOEA/D-M2M [13], MOEA/D-DD [28], MOEA/D-PSF or MSF [29], MOEA/D-2TCHMFI [30], MOEA/D-MTCH [24] are not included as on average these algorithms do not show a higher performance than MOEA/D for the selected bi-objective problems. Similarly, the results of other algorithms from the current state of the art, such as GrEA [31] and HypE [32], are not added to the comparison as these algorithms have been shown to be outperformed by MOEA/D and NSGA-II on two-objective functions. Tests on the ZDT test set [33], which is highly unimodal, and WFG test set [34], which shows a bias towards certain regions of the objective space, non-separability of the input variables and different modality, have also been conducted. There are no statistically significant changes in comparative performance between MTS, NSGAII, MOEA/D and the hybrid algorithms, for the ZDT cases, and high similarity of behaviour in comparison to the CEC′09 test set for the WFG test instances. Therefore, only the CEC′09 functions have been included, as they provide a clearer illustration of the comparative performance. The CEC ′09 competition [5] used 14 different constrained and unconstrained functions. The unconstrained functions, UF1-UF7, have 30 variables each and the constrained functions, CF1-CF7, have 10 variables each with CF1-5 having 1 constraint and CF6-7 having 2 constraints. The tests are performed following the CEC′09 comparison rules [5] where each function is evaluated over 30 separate runs and the average of these results is compared; the stopping criterion is 300,000 function evaluations for each run; and the performance is evaluated based on the Inverted Generational Distance (IGD) values calculated using only the 100 best, evenly-spread, individuals taken from each run. IGD is the performance measure function of the Pareto Front, which shows the average distance between all points in the true Pareto Front and the closest solution from the achieved set and is calculated in Eq. (2);

$$IGD(A, P^*) = \frac{\sum_{v \in P^*} d(v, A)}{|P^*|} \quad (2)$$

where P^* is a set of uniformly distributed points along the true Pareto Front, in the objective space, A is the approximate set to the Pareto Front being evaluated and $d(v, A)$ is the minimum Euclidean distance between point v and the points in A. The IGD metric is the preferred method to calculate the diversity and accuracy of the Pareto front, as it allows comparison to the results in CEC′09. The Hyper Volume (HV) metric would provide a more accurate assessment of diversity but there is less available data for a comparison and so the results are not included.

Different MLSGA parameters: population size, number of collectives, steps between collective reproduction and number of eliminated collectives have been parametrically evaluated. The parameters that give the best performance across all of the problems are presented in this work. During the development of the hybrids the optimal number of collectives is shown to be dependent on the overall population size; with a higher population count more collectives should be introduced. In the case of 800 or more individuals then 8 collectives are preferred, and for smaller sizes 6 collectives are implemented. Using too few collectives limits the spread of the search and therefore the final diversity of the solutions. When too many collectives are used the same areas of the search space are re-evaluated by different groups, decreasing the efficiency of the search. However, minor changes away from the optimal number of collectives do not have a significant effect on the final performance. This results in six collectives being used with the MTS hybrid, using a lower population of 225, and two collectives of each type, MLS1, MSL2 and MLS2R. In the case of eight, used in NSGA-II and MOEA/D which use a high population size of 1800, there are 3 MLS2 collectives, 3 MLS2R collectives and 2 MLS1 collectives.

The overall population sizes are bigger than those commonly used in the literature as the collectives must maintain a reasonable population size for the individual level mechanisms to be effective. As MTS utilises multiple local searches for each individual, and thus requires a significantly higher number of iterations per generation, a lower population size is used compared to the MOEA/D and NSGA-II hybrids, and the collective reproduction steps have to occur more frequently. The number of eliminated collectives, 1, is the optimal value for all mechanisms and all problems. The number of steps between collective reproduction, 1 every 10 generations for NSGA-II and MOEA/D and 1 every generation for MTS, are shown to be problem independent and these values are used are the optimal values for all problems. However, the number of steps between collective reproduction should be balanced in order to maximise the added evolutionary pressure by the collective-level and should be adjusted for different types of mechanism. The NSGA-II and MOEA/D mechanisms are dependent on developing and maintaining a uniform Pareto front so the frequency of the

Table 2
MLSGA hybrid parameters utilised for benchmarking.

Step	Parameter	Value		
		MLSGA-MTS	MLSGA-NSGA-II	MLSGA-MEOAD
1. Initialisation	Type	Random		
	Encoding	Real values		
	Pop. Size	225	1800	
2. Classification	Method	SVM		
	No. Collectives	6	8	
	Collective size limits	min 10 individuals and max 1/6 of overall population size		
3. Individual level operations				
Fitness Evaluation	Type	Both f 1 and f 2		Both f 1 and f 2 based on Chebycheff Scalarizing Function [9]
Selection	Type	n/a	Binary tournament with crowding distance and non-dominated ranking	Tournament
Mating	Crossover type	n/a	Real variable SBX	Differential evolution crossover
	Crossover rate		1	
	Mutation type	3 local search methods	Polynomial	
	Mutation rate		0.08	
4. Collective level operations				
Fitness evaluation	Type	MLS1, MLS2, MLS2R, depending on the collective		
Elimination	Number of elim. collectives	1 every generation	1 every 10 generations	
5. Termination	Criterion	300000 function evaluations		

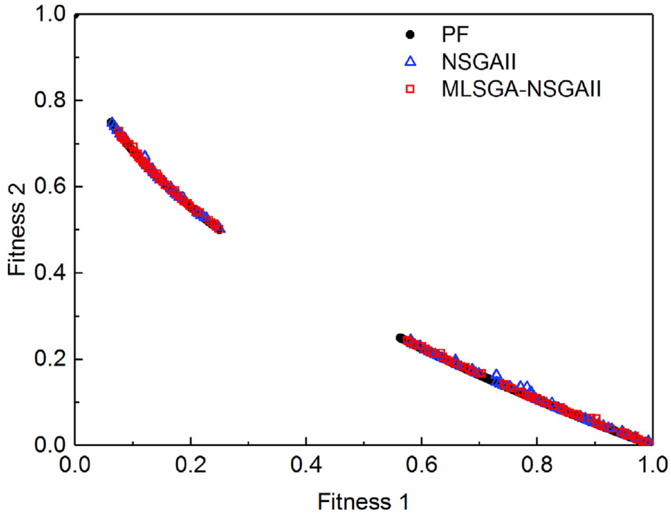


Fig. 1. Pareto front of NSGA-II and MLSGA-NSGA-II on the CF2.

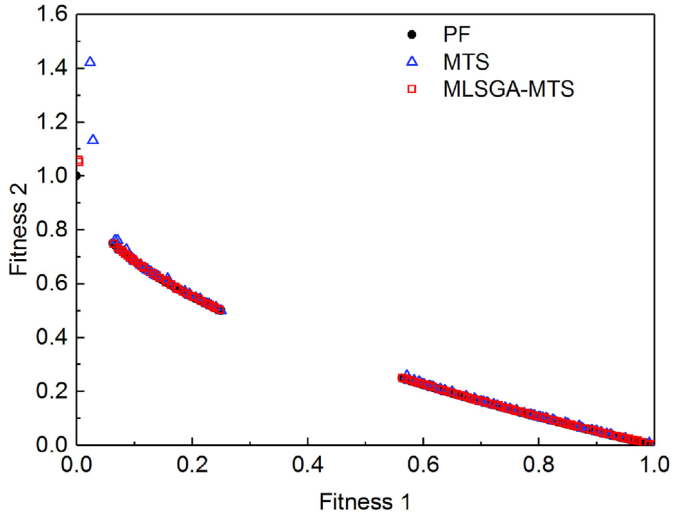


Fig. 3. Pareto front of MTS and MLSGA-MTS on the CF2.

elimination and reproduction step has to be reduced compared to MTS. Otherwise, the individual-level mechanisms do not have enough iterations to properly develop the front. This leads to premature elimination of potentially good solutions which significantly reduces the diversity of the final solutions and the final performance. Parameters used by each hybrid are detailed in Table 2 and remain constant over all the runs for different functions. The parameters at the individual level are retained from the original sources but some small performance gains might be possible by adjusting these values.

4. Performance benchmarking

The constrained function results are simulated for the hybrid algorithms and the Pareto Fronts are compared to those generated by running the individual level algorithms separately. These are illustrated for the CF2 with NSGA-II in Fig. 1, MOEA/D in Fig. 2 and MTS in Fig. 3, and for the CF5 illustrated in Figs. 4–6 for NSGA-II, MOEA/D and MTS respectively. CF2 and CF5 are chosen as they provide results representative of the worst and the best cases for hybrid algorithms on the constrained problem set. The Pareto Fronts for the figures have been randomly chosen from the 5 runs with the lowest IGD value.

For the CF2 the resulting Pareto Fronts are close to the best possible for all of the hybrids, with an even spread of points. The MLSGA-MOEA/D hybrid shows higher diversity and accuracy of points compared to MOEA/D which is not able to reach the front. In the case of NSGA-II and MTS, the MLSGA hybrids show similar performance to the individual level algorithms where the resulting points cover the entire length of the Pareto Front and it is difficult to visually determine which has the better performance.

In the worst-case scenario, CF5, both MLSGA-MTS and MLSGA-NSGA-II hybrids have a similar performance to the original algorithms in terms of accuracy but with a more even spread of points. For MOEA/D the points are concentrated mostly on one region of the Pareto Front and the hybrid shows a wider diversity, covering the entire length of the Pareto Front. However, the accuracy of these points is poor with fewer of the hybrid points lying on the true Pareto Front.

The process is repeated for the unconstrained problems, UF1-7. UF2 is shown in Fig. 7 for NSGA-II, Fig. 8 for MOEA/D and Fig. 9 for MTS, and UF5 is shown in Figs. 10–12 for NSGA-II, MOEA/D and MTS respectively. These figures are chosen as they illustrate the worst and the best results for the unconstrained test set, similarly to the previous cases.

For the unconstrained problems, the results vary more between the different hybrids and the individual level algorithms. The MLSGA-NSGA-II hybrid has similar performance to NSGA-II, and it is hard

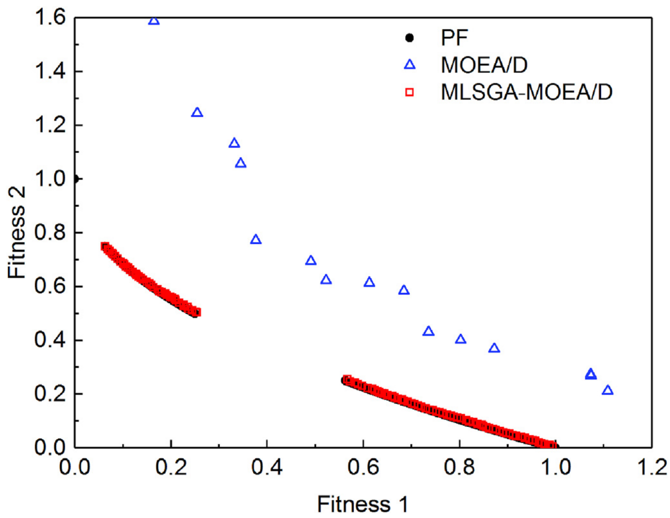


Fig. 2. Pareto front of MOEA/D and MLSGA-MOEA/D on the CF2.

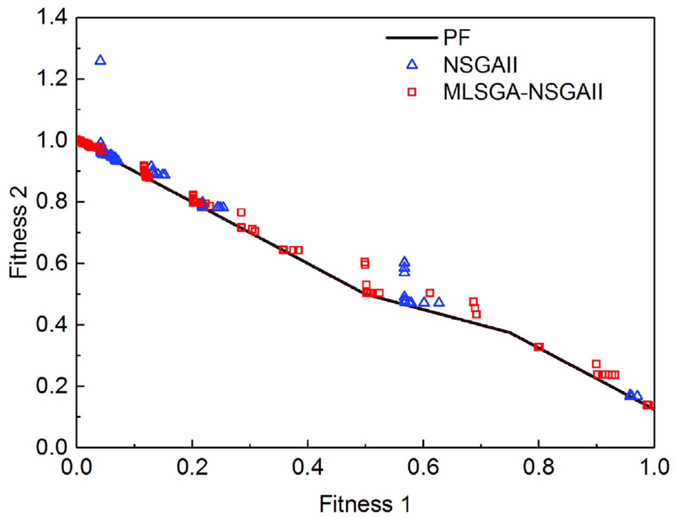


Fig. 4. Pareto front of NSGA-II and MLSGA-NSGA-II on the CF5.

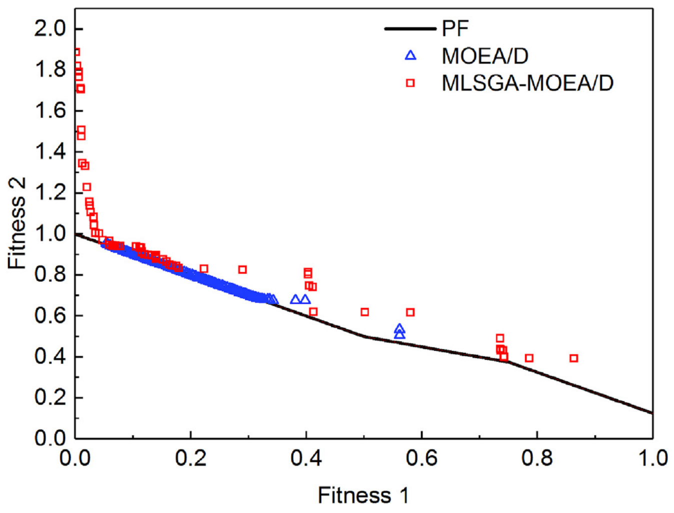


Fig. 5. Pareto front of MOEA/D and MLSGA-MOEA/D on the CF5.

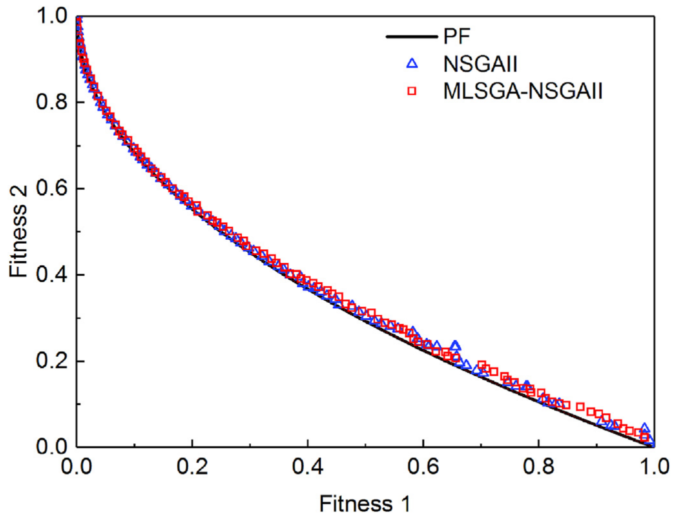


Fig. 7. Pareto front of NSGA-II and MLSGA-NSGA-II on the UF2.

to distinguish visually which variant is better for both presented functions. For MLSGA-MTS and MTS algorithms, similar results are obtained on UF2 function but MTS exhibits better performance on UF5 in terms of diversity and accuracy. The MLSGA-MOEA/D is outperformed by MOEA/D in both presented cases.

The results of the three hybrids, MLSGA-NSGA-II, MLSGA-MOEA/D and MLSGA-MTS are also compared to the original algorithms based on the average IGD values and presented for constrained test cases in Table 3 for CF1-7, and for UF1-7, in Table 4. In both tables the better results between the hybrid and the individual level algorithm are highlighted in blue and are in bold font. The minimum, maximum and standard deviation over 30 runs are given in the brackets for each MLSGA variant. Additionally, the Wilcoxon's rank sum test was conducted to assess the statistical significance of the differences between the results obtained by MLSGA hybrids and the original algorithms with a significance level of $\alpha = 0.05$. Results for the original algorithms are taken from the CEC′09 benchmarking [17]. As MLSGA-NSGA-II is based on an updated version of NSGA-II, not on the version from the CEC′09 ranking, the updated results for this algorithm are included in Tables 3 and 4. As the results for MOEA/D on the constrained functions have not been presented in CEC′09, these are simulated and included in Table 3.

The constrained results show that implementation of collective level mechanisms improves the performance of GAs in general, for NSGA-II improvements are shown on 6 out of 7 cases. The MLSGA-MTS also shows improvements over the original algorithm on 6 out of 7 cases but with statistical significance on 4 of the functions. For MOEA/D the MLSGA mechanisms leads to improvements in the performance in all cases. For the unconstrained cases better results can be observed for MLSGA-NSGA-II in comparison to NSGA-II for all the presented functions, with statistical significance for 5 out of 7. In the case of MLSGA-MTS hybrid the improvement has been shown on all the functions except for UF4 and UF5, though this is not statistically significant for the UF6. Implementation of the MLSGA mechanisms decreases the performance of MOEA/D on all problems of this type. This is most likely caused by the collective-level operations which are not adjusted for the MOEA/D specific mechanisms, such as the weight vectors and neighbourhoods of solutions. In this case the weight vectors of the eliminated solutions are not taken into account during the collective reproduction step when the offspring collective is created. Therefore, the weights are randomly assigned to the new solutions, which results in lower overall fitness at the individual-level. The mechanisms were not fully adjusted as the assignment of the best

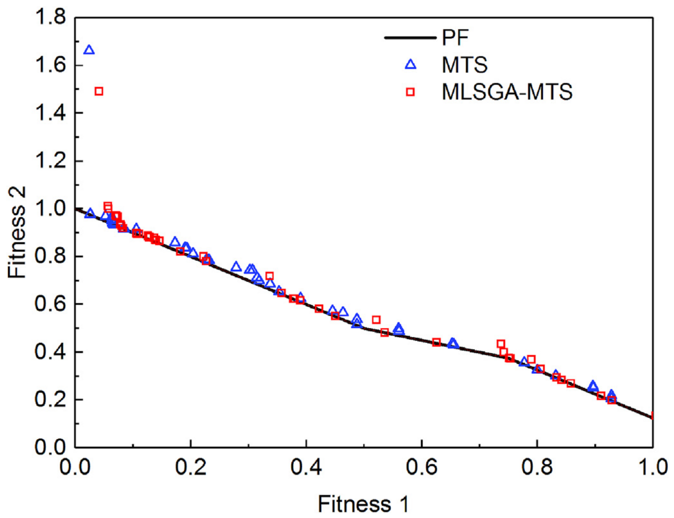


Fig. 6. Pareto front of MTS and MLSGA-MTS on the CF5.

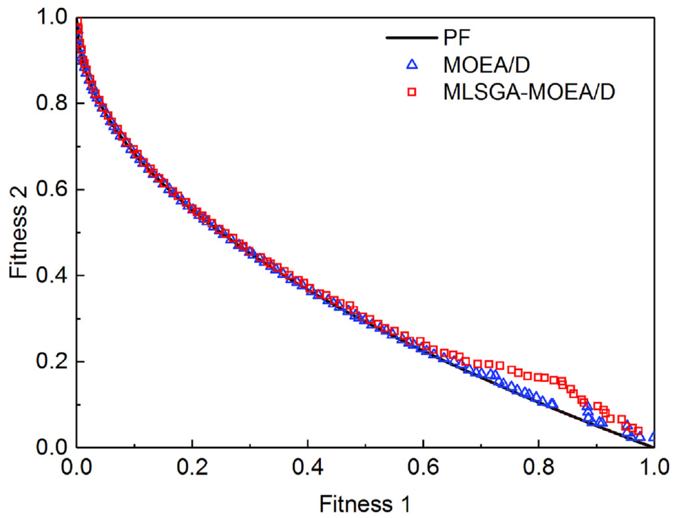


Fig. 8. Pareto front of MOEA/D and MLSGA-MOEA/D on the UF2.

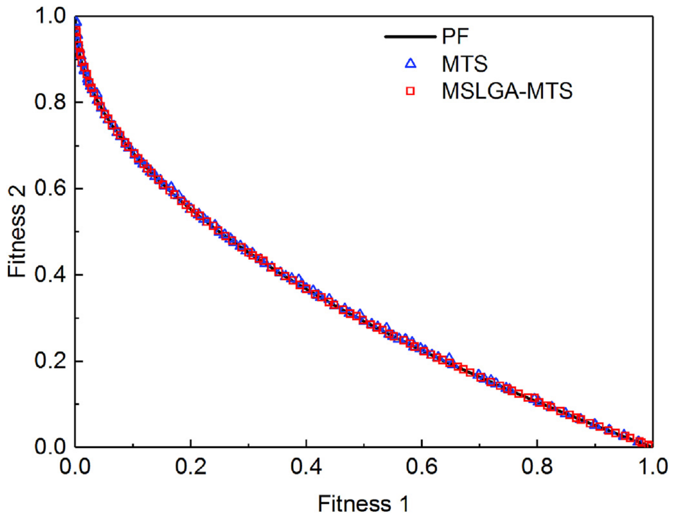


Fig. 9. Pareto front of MTS and MLSGA-MTS on the UF2.

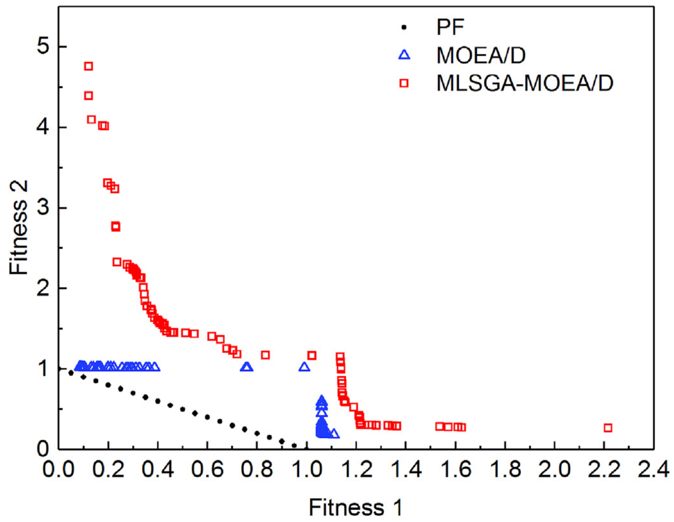


Fig. 11. Pareto front of MOEA/D and MLSGA-MOEA/D on the UF5.

individuals to each weight vector during every collective reproduction would require a significant number of comparisons and therefore would result in drastically higher computational costs over the original algorithm.

To highlight the performance of the MLSGA approach the MLSGA-MTS variant is compared to the updated CEC′09 competition rankings. This is despite the fact that it is not the strongest performing hybrid on every problem and a priori knowledge of the problem could lead to stronger performance by matching the MLSGA hybrid to a given problem. These are presented for constrained test cases in Table 5 for CF1-7, and for unconstrained functions, UF1-7, in Table 6. In the tables, the results of the hybrid are in bold and highlighted in light blue colour and the results for the original MTS are highlighted in dark blue. Additionally, the results for the two other hybrid methods the MOEA/D + TCH variant of BCE [18] and HEIA [19] are included in bold, as they are hybrid methodologies that show leading performance.

In the updated CEC'09, rankings the MLSGA-MTS hybrid would be placed in the top 3 algorithms for constrained functions for 6 out of 7 cases, showing the best performance for CF6. For the unconstrained problems, the presented hybrid would be in the top 3 algorithm for 5 out of 7 cases, and the best performing for two problems, UF1 and UF2. Interestingly, for the UF7 function, the MLSGA-MTS hybrid is outperformed by most of the updated CEC ′09 competitors. In this case, the performance of the hybrid is strongly affected by poor performance of the original MTS algorithm. The proposed methodology would have been placed 2^{nd}over the constrained test sets, and 2nd place on the unconstrained functions, leading to the best general performance.

5. Understanding MLSGA mechanisms and limitations

It is shown that the MLSGA hybrids perform better than the individual level algorithms; NSGA-II and MTS in all cases and MOEA/D for the constrained problems. The results demonstrate a capability to improve the performance of a wider range of GAs than that shown in Ref. [4] and that the split in the fitness function is robust to the selection of a number of GAs at the individual level. However, it is also shown for the first time that the performance of all algorithms is not improved across all problem types.

The results show that as long as the levels are different there can be more than one objective per level allowing for some overlap between elements at each level. For example, in these cases objective 1 can be used as part of the individual level and the collective level without impairing performance. These results indicate promise for extension

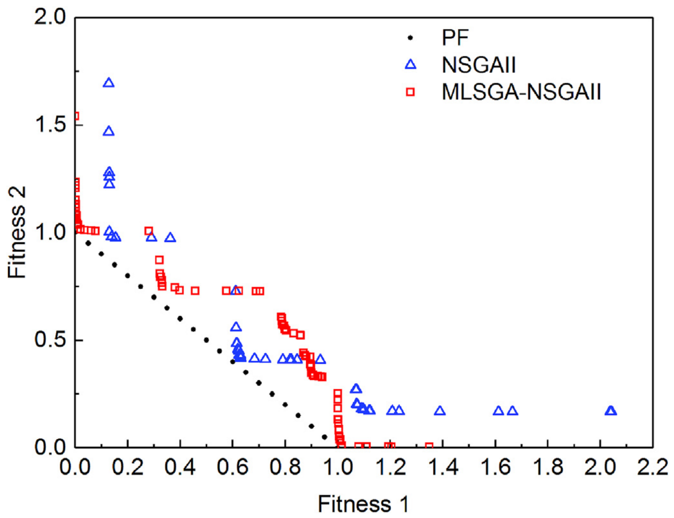


Fig. 10. Pareto front of NSGA-II and MLSGA-NSGA-II on the UF5.

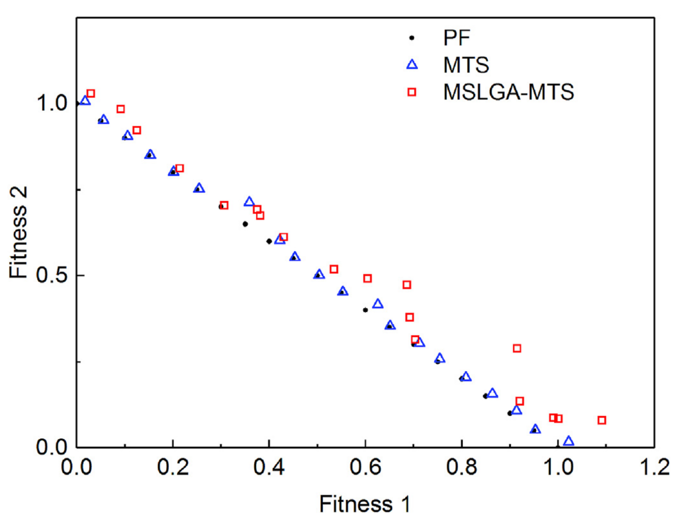


Fig. 12. Pareto front of MTS and MLSGA-MTS on the UF5.

Table 3
Comparison of MLSGA hybrids and original algorithms on CEC'09 two-objective constrained problems CF1-7.

Algorithm	Average IGD (*min; max; std*)						
	CF1	CF2	CF3	CF4	CF5	CF6	CF7
NSGA-II	**0.01480**	0.01249	0.24428	0.04946	0.13595	0.03309	0.13749
MLSGA-NSGA-II	0.02429 *(0.02074; 0.02842; 0.001883)* [-]	**0.00692** ***(0.00402; 0.01252; 0.002491)*** **[+]**	**0.11619** ***(0.07265; 0.15936; 0.027561)*** **[+]**	**0.01924** ***(0.01289; 0.03344; 0.004994)*** **[+]**	**0.05513** ***(0.03812; 0.11002; 0.015264)*** **[+]**	**0.02231** ***(0.01191; 0.04561; 0.007163)*** **[+]**	**0.06110** ***(0.03247; 0.17206; 0.031170)*** **[+]**
MOEA/D	0.0204	0.06324	0.543	0.1503	0.223	0.1392	1.192
MLSGA-MOEA/D	**0.00401** ***(0.00255; 0.00700; 0.001151)*** **[+]**	**0.00550** ***(0.00366; 0.01266; 0.001814)*** **[+]**	**0.27730** ***(0.14522; 0.37486; 0.051974)*** **[+]**	**0.03404** ***(0.02203; 0.05054; 0.005912)*** **[+]**	**0.20834** ***(0.10708; 0.37707; 0.068027)*** **[+]**	**0.03710** ***(0.02475; 0.05598; 0.008044)*** **[+]**	**0.15495** ***(0.08564; 0.31225; 0.056544)*** **[+]**
MTS	0.01918	0.02677	0.10446	0.01109	**0.02077**	0.01616	0.02469
MLSGA-MTS	**0.01825** ***(0.01267; 0.02396; 0.002441)*** **[+]**	**0.00285** ***(0.0026; 0.00306; 0.000111)*** **[+]**	**0.10195** ***(0.07297; 0.13768; 0.014918)*** **[≈]**	**0.00822** ***(0.00729; 0.00932; 0.000524)*** **[+]**	0.02876 *(0.02183; 0.03499; 0.003313)* [-]	**0.00747** ***(0.00662; 0.0094; 0.000707)*** **[+]**	**0.02405** ***(0.01969; 0.02927; 0.002662)*** **[≈]**

[+], [-], and [≈] indicate that the results of MLSGA hybrid are significantly better, worse or similar to the original algorithm using Wilcoxon's rank sum test.

Table 4
Comparison of MLSGA hybrids and original algorithms on CEC'09 two-objective unconstrained problems UF1-7.

Algorithm	Average IGD (*min; max; std*)						
	UF1	UF2	UF3	UF4	UF5	UF6	UF7
NSGA-II	0.048182	0.01671	0.15966	0.04798	0.22088	0.302	0.02751
MLSGA-NSGA-II	**0.01407** ***(0.01079; 0.02067; 0.00239)*** **[+]**	**0.01532** ***(0.01404; 0.01814; 0.000945)*** **[+]**	**0.15700** ***(0.13171; 0.19965 0.016248)*** **[≈]**	**0.04663** ***(0.04549; 0.04775; 0.000640)*** **[≈]**	**0.18373** ***(0.11668; 0.26350; 0.032948)*** **[+]**	**0.25145** ***(0.22234; 0.29777; 0.016909)*** **[+]**	**0.01175** ***(0.00962; 0.01688; 0.001641)*** **[+]**
MOEA/D	**0.00435**	**0.00679**	**0.00742**	**0.06385**	**0.18071**	**0.00587**	**0.00587**
MLSGA-MOEA/D	0.04281 *(0.02609; 0.07431; 0.010244)* [-]	0.02321 *(0.01804; 0.02995; 0.002626)* [-]	0.09862 *(0.05775; 0.14859; 0.023678)* [-]	0.07288 *(0.06169; 0.08331; 0.004818)* [-]	0.82728 *(0.55343; 1.21254; 0.161068)* [-]	0.36551 *(0.32494; 0.39631; 0.020352)* [-]	0.02279 *(0.01613; 0.03927; 0.005208)* [-]
MTS	0.00646	0.00615	0.0531	**0.02356**	**0.01489**	0.05917	0.04079
MLSGA-MTS	**0.0041** ***(0.00398; 0.00427; 0.000071)*** **[+]**	**0.00411** ***(0.00400; 0.00427; 0.000068)*** **[+]**	**0.03632** ***(0.0279; 0.04397; 0.003525)*** **[+]**	0.02724 *(0.02628; 0.02889; 0.000602)* [-]	0.06154 *(0.04357; 0.07368; 0.006264)* [-]	**0.0579** ***(0.04937; 0.06716; 0.00482)*** **[≈]**	**0.03714** ***(0.01664; 0.07475; 0.01714)*** **[+]**

[+], [-], and [≈] indicate that the results of MLSGA hybrid are significantly better, worse or similar to the original algorithm using Wilcoxon's rank sum test.

to many-objectives problems as concerns that each objective would require its own level is dismissed. This concern would have led to a large number of levels, potentially one per objective, and an extensive tree of corresponding collectives of collectives. Expansion in the number of collectives required would necessitate an exponential increase in individual population sizes to allow for a large enough number of higher level collectives. However, additional investigations will be required into how best to adjust the fitness function at each level for these problems.

Table 5
Updated CEC '09 ranking on the two-objective constrained CF1-7 problems including MLSGA-MTS hybrid.

Rank	Name/Average IGD						
	CF1	**CF2**	**CF3**	**CF4**	**CF5**	**CF6**	**CF7**
1	LiuLi 0.00085	DMOEA-DD 0.0021	DMOEA-DD 0.056305	DMOEA-DD 0.00699	DMOEA-DD 0.01577	**MLSGA-MTS 0.00747**	DMOEA-DD 0.01905
2	NSGA-IILS 0.00692	**MLSGA-MTS 0.00285**	**MLSGA-MTS 0.10195**	GDE3 0.00799	MTS 0.02077	LiuLi 0.013948	**MLSGA-MTS 0.02405**
3	NSGA-II 0.01480	LiuLi 0.0042	MTS 0.10446	**MLSGA-MTS 0.00822**	**MLSGA-MTS 0.02876**	DMOEA-DD 0.01502	MTS 0.02469
4	MEOAD-GM 0.0108	MEOAD-GM 0.008	GDE3 0.127506	MTS 0.01109	GDE3 0.06799	MTS 0.01616	GDE3 0.04169
5	DMOEA-DD 0.01131	NSGA-IILS 0.01183	LiuLi 0.182905	LiuLi 0.01423	LiuLi 0.10973	NSGA-IILS 0.02013	LiuLi 0.10446
6	**MLSGA-MTS 0.01825**	NSGA-II 0.01249	NSGA-IILS 0.23994	NSGA-IILS 0.01576	NSGA-II 0.13595	NSGA-II 0.03309	NSGA-II 0.13749
7	MTS 0.01918	GDE3 0.01597	NSGA-II 0.24428	NSGA-II 0.04946	NSGA-IILS 0.1842	GDE3 0.06199	NSGA-IILS 0.23345
8	GDE3 0.0294	MTS 0.02677	MEOAD-GM 0.5134	MEOAD-GM 0.0707	MEOAD-GM 0.5446	DECMO-SA 0.14782	DECMO-SA 0.26049
9	DECMO-SA 0.10773	DECMO-SA 0.0946	DECMO-SA 1000000	DECMO-SA 0.15265	DECMO-SA 0.41275	MEOAD-GM 0.2071	MEOAD-GM 0.5356

The authors suggest that the increase in performance is due a combination of the use of novel reproduction mechanisms between subpopulations, collectives, and the split in the fitness function. In MLSGA, the individual-level operations lead to exploration and exploitation of both objective and variable spaces, in a similar manner to the original algorithms. The collective-level creates artificial boundaries, by strongly penalizing solutions in certain regions and pushing the individuals in to the preferred ones. Splitting the population, either in the form of separate fronts [7], decomposition [9] or direct subpopulation approaches [15], has been shown to be beneficial in other algorithms. This is because it decreases the chances of premature convergence for the whole population at a local optima and leads to an increase in the overall diversity of the final solutions. The MLSGA mechanisms enhance the ability to explore different parts of the objective space even further, by introducing an additional selection pressure. The hybrid algorithms demonstrate an increase in performance over the original variants on the constrained problems as the diversity is more crucial on this type of problem. The split in fitness function and independence of collectives allows them to more easily move around "gaps" in the objective space created by the constraints; for the unconstrained problems this diversity is less essential. This claim is supported by the high gains in performance of the MLSGA-MOEA/D hybrid on constrained problems, as the original MOEA/D struggled to operate in non-continuous search spaces as the weights are defined in straight lines which may pass through these regions, leading to an inefficient search.

In the hybrid approach, the addition of MLS-U, defining different fitnesses from MLS1, MLS2 and MLS2R in each collective, leads to a "specialisation" of collectives, where each collective type creates a selection pressure into a different direction, and thus leads to exploration of different regions of the objective space. This is illustrated in Fig. 13, where the different collective types are shown to be exploring different parts of the Pareto Front. MLS1 focuses on the "middle" regions of the Pareto Front, as individuals with the lowest value of the average of both fitness functions are promoted, due to its fitness definition. MLS2 and MLS2R exploit the peripheral regions of the Pareto Front and in these cases the collective reproduction mechanisms promote extreme solutions, as only one fitness function is considered. Furthermore, the proposed methodology is able to search negative regions of the objective space, unlike decomposition methods, as it can avoid the regions without feasible solutions where the decomposition methods has been proven ineffective, such as in discontinuous problems. This is exhibited by high performance on the constrained problems, and non-continuous functions such as UF5 and UF6.

The MSLGA mechanisms do not increase the performance for all of the algorithms on all cases. In the unconstrained functions, the MOEA/D hybrid performs worse than the original. The authors suggest that this reduction in performance is caused by the collective reproduction mechanisms, which do not properly maintain a number of the parameters that the original MOEA/D utilises, such as the weight vectors and neighbourhood of solutions. It might also be that such a specialist solver's performance is also degraded by the addition of the MLSGA mechanism, which improves the generality of the algorithm; there is no free lunch after all. In the MLSGA-MOEA/D hybrid the newly created collective inherits the weight vector of the eliminated collective during the collective reproduction step. However, the solutions copied from the remaining collectives are not subject to this weight vector as the specific weights are assigned to the individuals randomly. This means that the best weight vector is not assigned to each new solution. The neighbourhood of solutions in the hybrid approach is recreated in the offspring collective, based only on the weight values. Therefore, the new neighbourhood does not reflect the relationship between the two neighbour solutions in both the objective and decision variable spaces. This lowers the overall fitness of the individuals in the new collective and thus decreases the performance compared

Table 6
Updated CEC'09 ranking on the two-objective unconstrained UF1-7 problems including MLSGA-MTS hybrid.

Rank	Name/Average IGD						
	UF1	**UF2**	**UF3**	**UF4**	**UF5**	**UF6**	**UF7**
1	**BCE 0.00164**	**MLSGA-MTS 0.00411**	MOEA/D 0.00742	MTS 0.02356	MTS 0.01489	MOEA/D 0.00587	**HEIA 0.00309**
2	**HEIA 0.0027**	**HEIA 0.00581**	**BCE 0.00957**	GDE3 0.0265	GDE3 0.03928	**MLSGA-MTS 0.0579**	MOEA/D 0.00587
3	**MLSGA-MTS 0.0041**	MTS 0.00615	**HEIA 0.00128**	**MLSGA-MTS 0.02724**	**MLSGA-MTS 0.06154**	MTS 0.05917	LiuLi 0.0073
4	MOEA/D 0.00435	MOEAD-GM 0.0064	LiuLi 0.01497	DECMO-SA-SQP 0.03392	AMGA 0.09405	DMOEA-DD 0.06673	MOEAD-GM 0.0076
5	GDE3 0.00534	**BCE 0.00656**	DMOEA-DD 0.03337	**HEIA 0.0377**	LiuLi 0.16186	OMOEA-II 0.07338	DMOEA-DD 0.01032
6	MOEAD-GM 0.0062	DMOEA-DD 0.00679	**MLSGA-MTS 0.03632**	AMGA 0.04062	DECMO-SA-SQP 0.16713	Clustering MOEA 0.0871	**BCE 0.01212**
7	MTS 0.00646	MOEA/D 0.00679	MOEAD-GM 0.049	DMOEA-DD 0.04268	OMOEA-II 0.1692	MOEP 0.1031	MOEP 0.0197
8	LiuLi 0.00785	OWMOS-aDE 0.0081	MTS 0.0531	MOEP 0.0427	MOEA/D 0.18071	DECMO-SA-SQP 0.12604	NSGA-II-LS 0.02132
9	DMOEA-DD 0.01038	GDE3 0.01195	Clustering MOEA 0.0549	LiuLi 0.0435	**HEIA 0.205**	AMGA 0.12942	Clustering MOEA 0.0223
10	NSGA-II-LS 0.01153	LiuLi 0.0123	AMGA 0.06998	OMOEA-II 0.04624	NSGA-II 0.22088	**HEIA 0.152**	DECMO-SA-SQP 0.02416
11	OWMOS-aDE 0.0122	NSGA-II-LS 0.01237	DECMO-SA-SQP 0.0935	MOEAD-GM 0.0476	MOEP 0.2245	LiuLi 0.17555	GDE3 0.02522
12	Clustering MOEA 0.0299	AMGA 0.01623	MOEP 0.099	NSGA-II 0.04798	Clustering MOEA 0.2473	OWMOS-aDE 0.1918	NSGA-II 0.02751
13	AMGA 0.03588	NSGA-II 0.01671	OWMOS-aDE 0.103	OWMOS-aDE 0.0513	DMOEA-DD 0.31454	GDE3 0.25091	OMOEA-II 0.03354
14	NSGA-II 0.048182	MOEP 0.0189	NSGA-II-LS 0.10603	NSGA-II-LS 0.0584	**BCE 0.40341**	NSGA-II 0.302	**MLSGA-MTS 0.03714**
15	MOEP 0.0596	Clustering MOEA 0.0228	GDE3 0.10639	Clustering MOEA 0.0585	OWMOS-aDE 0.4303	NSGA-II-LS 0.31032	MTS 0.04079
16	DECMO-SA-SQP 0.07702	DECMO-SA-SQP 0.02834	NSGA-II 0.15966	**BCE 0.06063**	NSGA-II-LS 0.5657	**BCE 0.425**	AMGA 0.05707

to the original MOEA/D. In addition, the NSGA-II and MOEA/D algorithms rely on a constant front to generate the optimal results, which the current simple collective reproduction mechanisms do not account for. Development of improvements to the collective level mechanisms, bespoke to the algorithm used at the individual level, may provide a further increase in performance. For NSGA-II and MOEA/D, this will promote the constant front generation and will consider the original algorithms specific mechanisms, such as weight and neighbourhoods in MOEA/D.

MLSGA-MTS provides strong performance across all of the problem sets, providing the best general performance. Its only poor performance, 12th in the updated CEC′09 rankings, is on UF7. It is difficult to determine the reasons for the poor performance but UF7 is a continuous linear non-uniform problem with a strong bias towards the right side of the Pareto Front. MTS and MLSGA do not have as strong diversity preservation metrics as other leading algorithms, and this potentially causes problems in regions where there is such a strong bias towards points in one area, a limitation to the MLSGA-MTS approach. In comparison to the other hybrid approaches, BCE [18] and HEIA [19], the MLSGA-MTS exhibits better performance in 4 out of 7 functions. Unfortunately, the algorithms cannot be compared on the constrained functions due to lack of benchmarks of BCE and HEIA on these problems.

The authors believe that MLSGA mechanisms can be successfully utilised on various GAs to improve the performance, not only those presented in this work. Guidance is provided on how to tune the algorithm for these different hybrids, and the result is that many of the new parameters require little adjusting.

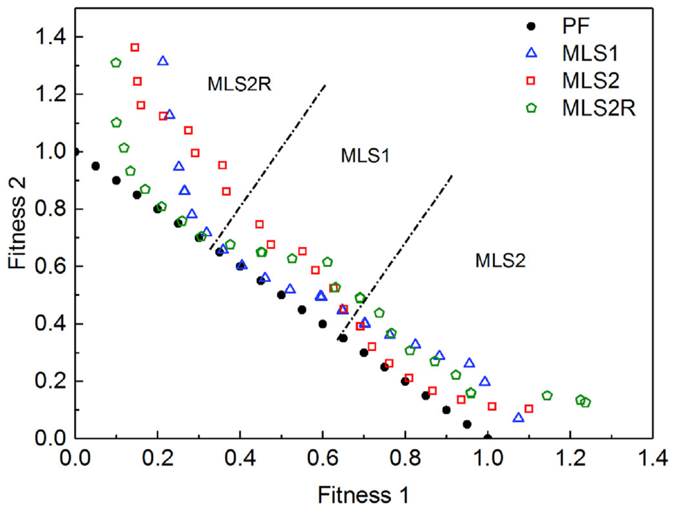


Fig. 13. Pareto Front of MLS1, MLS2 and MLS2R fitness definition types from MLSGA-NSGA-II algorithm on the CF1 problem.

6. Conclusions

This paper investigates the performance of implementing different mechanisms at the individual level of Multi-Level Selection Genetic Algorithm (MLSGA). This leads to a better understanding of how the split in the fitness function and collective level reproduction mechanisms interact with a range of individual level mechanisms. Utilisation of MLSGA to create hybrid algorithms is shown to be beneficial in a number of cases. Hyperparameter tuning is performed which shows that the results are relatively insensitive to the new parameters, or that these parameters are easy to select. The MLSGA mechanisms improve the overall performance and provide a different behaviour to the typical "convergence first, diversity second" approach, which leads to the hybrids demonstrating particularly strong performance on discontinuous problems. The hybrid genetic algorithms, MLSGA-MTS and MLSGA-NSGA-II, show increased performance on the CEC′09 constrained and unconstrained problems over their original implementations but MLSGA-MOEA/D only exhibits this improvement on the constrained problems. The best hybrid is MLSGA-MTS, which performs the best on a number of the CEC′09 benchmarking test problems and on the updated rankings places 2nd on the constrained test sets and 2nd on the unconstrained functions, leading to the best general performance across all the problems.

References

[1] E. Sober, D.S. Wilson, Unto Others: the Evolution and Psychology of Unselfish Behavior, Harvard University Press, 1999.
[2] A. J. Sobey, P. A. Grudniewski, Re-inspiring the genetic algorithm with multi-level selection theory: multi-level selection genetic algorithm, Bioinspiration Biomimetics 13 (5). https://doi.org/10.1088/1748-3190/aad2e8.
[3] K.A. De Jong, Evolutionary Computation: a Unified Approach, MIT Press, Cambridge, Mass., London, 2006.
[4] P.A. Grudniewski, A.J. Sobey, Multi-level selection genetic algorithm applied to CEC '09 test instances, in: 2017 IEEE Congress on Evolutionary Computation, CEC 2017 - Proceedings, 2017, pp. 1613–1620, https://doi.org/10.1109/CEC.2017.7969495.
[5] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition, Tech. rep., 2009.
[6] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: a survey of the state of the art, Swarm Evol. Comput. 1 (2011) 32–49, https://doi.org/10.1016/j.swevo.2011.03.001 http://dx.doi.org/10.1016/j.swevo.2011.03.001.
[7] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, https://doi.org/10.1109/4235.996017.
[8] L.-Y. Tseng, C. Chen, Multiple trajectory search for multiobjective optimization, in: 2007 IEEE Congress on Evolutionary Computation (CEC 2007), 2007, pp. 3609–3616, https://doi.org/10.1109/CEC.2007.4424940.
[9] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731, https://doi.org/10.1109/TEVC.2007.892759.
[10] H. Seada, K. Deb, U-NSGA-III: a unified evolutionary optimization procedure for single, multiple, and many objectives: proof-of-principle results, evolutionary multi-criterion optimization, in: 8th International Conference, EMO 2015, Guimar{ã}es, Portugal, March 29 –April 1, 2015. Proceedings, Part II, 2015, pp. 34–49, https://doi.org/10.1007/978-3-319-15892-1_3.
[11] E. Zitzler, K. Simon, Indicator-based selection in multiobjective search, in: Parallel Problem Solving from Nature - PPSN VIII, 2004, pp. 832–842, https://doi.org/10.1007/978-3-540-30217-9_84.
[12] D. Whitley, S. Rana, R.B. Heckendorn, The island model genetic algorithm: on separability, population size and convergence, J. Comput. Inf. Technol. 7 (1999) 33–47 10.1.1.36.7225.
[13] H.-l. Liu, F. Gu, Q. Zhang (3). https://doi.org/10.1109/TEVC.2013.2281533.
[14] J. Branke, H. Schmeck, K. Deb, M. Reddy, Parallelizing multi-objective evolutionary algorithms: cone separation, Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753) 2, 2004, pp. 1952–1957, https://doi.org/10.1109/CEC.2004.1331135.
[15] M. Liu, X. Zou, C. Yu, Z. Wu, Performance assessment of DMOEA-DD with CEC 2009 MOEA competition test instances, IEEE Congr. Evol. Comput. 1 (2009) 2913–2918, https://doi.org/10.1109/CEC.2009.4983309.
[16] H.L. Liu, X. Li, The multiobjective evolutionary algorithm based on determined weight and sub-regional search, in: 2009 IEEE Congress on Evolutionary Computation, CEC 2009, 2009, pp. 1928–1934, https://doi.org/10.1109/CEC.2009.4983176.
[17] Q. Zhang, P.N. Suganthan, Final Report on CEC ' 09 MOEA Competition, Tech. rep. 2009http://dces.essex.ac.uk/staff/zhang/moeacompetition09.htm.
[18] M. Li, S. Yang, X. Liu, Pareto or Non-Pareto: Bi-criterion evolution in multiobjective optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 645–665, https://doi.org/10.1109/TEVC.2015.2504730.
[19] Q. Lin, J. Chen, Z.-H. Zhan, W.-N. Chen, C.A. Coello Coello, Y. Yin, C.-M. Lin, J. Zhang, A hybrid evolutionary immune algorithm for multiobjective optimization problems, IEEE Trans. Evol. Comput. 20 (5) (2016) 711–729, https://doi.org/10.1109/TEVC.2015.2512930.
[20] T. Lenaerts, A. Defaweux, P.V. Remortel, B. Manderick, Modeling artificial multi-level selection, in: AAAI Spring Symposium on Computational Synthesis. AAAI Spring Symposium Series, 2003.
[21] R. Akbari, V. Zeighami, K. Ziarati, MLGA: a multilevel cooperative genetic algorithm, in: Proceedings 2010 IEEE 5th International Conference on Bio-inspired Computing: Theories and Applications, BIC-TA 2010, 2010, pp. 271–277, https://doi.org/10.1109/BICTA.2010.5645316.
[22] R. Akbari, K. Ziarati, A multilevel evolutionary algorithm for optimizing numerical functions, Int. J. Ind. Eng. Comput. 2 (2) (2011) 419–430, https://doi.org/10.5267/j.ijiec.2010.03.002.
[23] S.X. Wu, W. Banzhaf, A hierarchical cooperative evolutionary algorithm, in: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, 2010, pp. 233–240, https://doi.org/10.1145/1830483.1830527.
[24] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (4) (2014) 577–601, https://doi.org/10.1109/TEVC.2013.2281534.
[25] S. Okasha, Evolution and the Levels of Selection, Oxford University Press, 2006.
[26] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 1–27, https://doi.org/10.1145/1961189.1961199. arXiv:0-387-31073-8 http://dl.acm.org/citation.cfm?doid=1961189.1961199.
[27] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: handling constraints and extending to an adaptive approach, IEEE Trans. Evol. Comput. 18 (4) (2014) 602–622, https://doi.org/10.1109/TEVC.2013.2281534.
[28] K. Li, K. Deb, Q. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Trans. Evol. Comput. 19 (5) (2015) 694–716, https://doi.org/10.1109/TEVC.2014.2373386 arXiv:arXiv:1011.1669v3.
[29] S. Jiang, S. Yang, Y. Wang, X. Liu, Scalarizing functions in decomposition-based multiobjective evolutionary algorithms, IEEE Trans. Evol. Comput. 22 (2) (2018) 296–313, https://doi.org/10.1109/TEVC.2017.2707980.
[30] X. Ma, Q. Zhang, G. Tian, J. Yang, Z. Zhu, On Tchebycheff decomposition approaches for multiobjective evolutionary optimization, IEEE Trans. Evol. Comput. 22 (2) (2018) 226–244, https://doi.org/10.1109/TEVC.2017.2704118.
[31] S. Yang, M. Li, X. Liu, J. Zheng, A grid-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 17 (5) (2013) 721–736, https://doi.org/10.1109/TEVC.2012.2227145.
[32] J. Bader, E. Zitzler, HypE: an algorithm for fast hypervolume-based many-objective optimization, Evol. Comput. 19 (1) (2011) 45–76, https://doi.org/10.1162/EVCO_a_00009.
[33] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: empirical results, Evol. Comput. 8 (2) (2000) 173–195, https://doi.org/10.1162/106365600568202.
[34] S. Huband, L. Barone, L. While, P. Hingston, A scalable multi-objective test problem toolkit, in: Evolutionary Multi-criterion Optimization, June, 2005, pp. 280–295, https://doi.org/10.1007/978-3-540-31880-4_20.

Appendix C Benchmarking the performance of Genetic Algorithms on Constrained Dynamic Problems

Noname manuscript No.
(will be inserted by the editor)

Benchmarking the performance of Genetic Algorithms on Constrained Dynamic Problems

P. A. Grudniewski · A. J. Sobey

Received: date / Accepted: date

Abstract The growing interest in dynamic optimisation has accelerated the development of genetic algorithms with specific mechanisms for these problems. To ensure that these developed mechanisms are capable of solving a wide range of practical problems it is important to have a diverse set of benchmarking functions to ensure the selection of the most appropriate Genetic Algorithm. However, the currently available benchmarking sets are limited to unconstrained problems with predominantly continuous characteristics. In this paper, the existing range of dynamic problems is extended with 15 novel constrained multi-objective functions. To determine how genetic algorithms perform on these constrained problems, and how this behaviour relates to unconstrained dynamic optimisation, 6 top-performing dynamic genetic algorithms are compared alongside 4 re-initialization strategies on the proposed test set, as well as the currently existing unconstrained cases. The results show that there are no differences between constrained/unconstrained optimisation, in contrast to the static problems. Therefore, dynamicity is the prevalent characteristic of these problems, which is shown to be more important than the discontinuous nature of the search and objective spaces. The best performing algorithm overall is MOEA/D, and VP is the best re-initialisation strategy. It is demonstrated that there is a need for more dynamic specific methodologies with high convergence, as it is more important to performance on dynamic problems than diversity.

P. A. Grudniewski
Fluid Structure Interactions Group, University of Southampton, Southampton, SO16 7QF, England, UK
E-mail: pag1c18@soton.ac.uk

A. J. Sobey
Fluid Structure Interactions Group, University of Southampton, Southampton, SO16 7QF, England, UK
Marine and Maritime Group, Data-centric Engineering, The Alan Turing Institute, The British Library, London, NW1 2DB, England, UK
E-mail: ajs502@soton.ac.uk

Keywords Dynamic multi-objective optimisation · constrained problems · genetic algorithms · performance benchmark · re-initialisation.

1 Dynamic optimisation problems

Many real-world problems have characteristics which change over time. Examples include forecasting of financial markets and stocks to increase profits in automated stock market investors [Bagheri et al., 2014]; forecasting, control systems and pattern recognition in traffic management to determine the scale of the traffic on the road and reduce the waiting time [Wismans et al., 2014]; climate controlling systems for crop growing greenhouses [Zhang, 2008] and LNG storage tank terminal design to reduce the cost of transporting LNG [Effendy et al., 2017]. This is increasing the interest in optimisation of Dynamic Single-Objective Problems [Branke, 2002] (DSOP) and Dynamic Multi-Objective Problems [Helbig and Engelbrecht, 2015, Li et al., 2011, Farina et al., 2004] (DMOP), via Genetic Algorithms (GA).

In order to identify the mechanisms with the strongest performance on these types of problems there are a growing number of dynamic test sets such as: FDA [Farina et al., 2004], JY [Jiang and Yang, 2017], UDF [Biswas et al., 2014], DSW [Mehnen et al., 2006], HE [Helbig and Engelbrecht, 2014] and ZJZ [Zhou et al., 2007]. These are used to replicate the most significant characteristics from different types of real-world problems. Current Dynamic Multi-Objective Problems exhibit various dynamic characteristics: search and objective space geometry changes, Pareto Optimal Front (POF) and Set (POS) curvature changes and shifts, discontinuities, modalities, periodicities, randomness of changes or predictabilities and variable-linkages [Helbig and Engelbrecht, 2014] but all of the presented benchmarking sets are limited to unconstrained problems and are focused on solutions with continuous Pareto Optimal Fronts. Conducting benchmarking on a limited range of problem types can lead to overdevelopment of mechanisms with a strong bias towards these problems, which reduces their efficiency across the entire set of problems. Taking dynamic optimisation as an example, then since most of the current benchmarks utilise continuous characteristics there are difficulties when selecting the best solver for a constrained problem. It is also possible that the mechanism development is biased towards these problems and it is likely that these problems will require different mechanisms in the same way that static problems tend to have solvers specialised either for constrained or unconstrained problems [Zhang and Suganthan, 2009]. Testing on a wider set of problems is more likely to highlight potential improvements to the mechanisms and understanding the limitations of each mechanism is essential if they are to be utilised effectively in real-life applications.

Therefore, this paper introduces a novel constrained dynamic test set to improve the knowledge related to Genetic Algorithm performance on dynamic problems with the aim of broadening the benchmarking possibilities for the current state-of-the-art. A set of 15 constrained dynamic multi-objective problems are developed and 6 Genetic Algorithms are selected for their top performance on dynamic or constrained/discontinuous problems: NSGA-II [Deb et al., 2002], MOEA/D [Zhang and Li, 2007], MLSGA-MTS [Grudniewski and Sobey, 2018], MTS [Tseng and Chen, 2009], HEIA [Lin et al., 2016] and BCE [Li et al., 2016], and their performance is compared on these problems. This performance is also tested in combination with different re-initialisation mechanisms, as these mechanisms make a considerable improvement to Genetic Algorithm performance on dynamic problems.

This paper is organised as follows: section 2 presents a literature review of different Genetic Algorithms and dynamic optimisation specific mechanisms; section 3 introduces a new set of constrained dynamic (CDF) multi-objective problems along with a review of currently existing sets; section 4 presents a brief overview of the Genetic Algorithms and the re-initialisation mechanisms used in the benchmarking; section 5 shows the results followed by conclusions in section 6.

2 Increasing the diversity of the population for dynamic optimisation

In dynamic multi-objective problems, the diversity of solutions has to be maintained over the generations in order to achieve a complete set of results, the same as for static problems. However, the diversity of the gene pool must also be maintained through different time steps, meaning that diversity must be retained after convergence has occurred, unlike in static problems. To solve this problem a number of novel re-initialisation methodologies are proposed as additions to the current Genetic Algorithms, to improve performance on these dynamic problems [Nguyen et al., 2012]. These methods generally enhance the diversity of the population whenever an environmental change occurs, without significant impact on the computational cost of the whole algorithm. Three major classes are found in the current state-of-the-art: niching and other diversity preservation schemes [Goh and Tan, 2009], partial replacement of the population or hypermutation [Deb et al., 2007] and guided re-initialisation [Zhou et al., 2007].

In niching and other diversity preservation schemes the internal algorithm mechanisms are used in order to regain the diversity of the population after the environment changes. An example is dCOEA [Goh and Tan, 2009] where a separate competitive process occurs when a dynamic change is detected, without the need to re-evaluate the whole population. Despite the positive impact on the final performance, the mechanisms from this methodology group often cannot be utilised outside of specific algorithms in which they were introduced and therefore the number of applications is limited. In addition, they show poor performance on problems with discontinuous search or objective spaces and cases where the environmental change is significant.

A second type of preservation scheme is hypermutation where a fraction of the set of potential solutions is modified by mutating the current individuals, or by replacing them with randomly generated new ones when an environmental change is detected. This method is implemented in the most commonly used Genetic Algorithm, NSGA-II [Deb et al., 2002], where the resulting algorithm is designated DNSGA-II [Deb et al., 2007], and is successfully tested on the FDA2 test function. However, the algorithm with the updated dynamic element is not compared to the original variant and therefore it is hard to determine how much benefit is gained from the hypermutation mechanism [Deb et al., 2007]. Various benchmarks show that DNSGA-II struggles on more complex cases, and is outperformed by other algorithms, such as MOEA/D and dCOEA [Jiang and Yang, 2017, Biswas et al., 2014], indicating that hypermutation does not sufficiently maintain diversity over the time changes.

Re-initialisation methods are the final class of preservation scheme. Four main variants have been described by Zhou et. al. [Zhou et al., 2007]:

1. random, where the new population is initiated arbitrarily;
2. prediction-based, where the movement of the Pareto Optimal Set or Front is predicted, based on historical information, and the new population is sampled around the predicted location using Gaussian noise;
3. variation-based, where new points are created by varying the solution in the last time window with a predicted Gaussian noise, therefore only the information from the last time window is used, without the need to store the historical data;
4. the mixed method, variance and prediction (VP), where the variation method is applied to half of the population, and the prediction method to the rest.

In an initial benchmarking exercise the random variant is shown to be highly inefficient, as the algorithm needs to restart at every time change, the prediction-based method is shown to outperform the variation-based approach but the variance and prediction method shows the top performance as it can maintain benefits from both of these methods. However, all of the variants are benchmarked on a limited test set, ZJZ [Zhou et al., 2007] and FDA [Farina et al., 2004], and are only combined with a single algorithm, DMEA/PRI [Zhou et al., 2007], and therefore further testing is required to test the pervasiveness of these conclusions. The prediction-based methods are also investigated by Biswas et. al. [Biswas et al., 2014] and a new strategy is introduced that relies on Controlled Extrapolation with a Pareto Optimal Front based on a distance approach (CER-POF). In this method, the positions of the parent solutions are evaluated based on the Pareto Optimal Front distances, instead of the Pareto Optimal Set distances as in Zhou et. al. [Zhou et al., 2007], and the offspring points are sampled using Gaussian noise or a hyperbolic function, with equal probability. The proposed CER-POF method, combined with MOEA/D, is shown to outperform the prediction-based re-initialisation strategy presented by Zhou [Zhou et al., 2007], and the hypermutation-based DNSGA-II algorithm [Deb et al., 2007]. However, it is not compared to MOEA/D without re-initialisation, in combination with the VP method or to the Pareto Optimal Set based prediction introduced by Zhou [Zhou et al., 2007], which limits the generality of the conclusions. In addition, the prediction-based re-initialisation methods assume that the change of the Pareto Optimal Set or Pareto Optimal Front follow specific patterns, which can be found and exploited to predict the next time step. The performance of these methods on problems where a pattern does not exist or cannot be found easily, has not been properly evaluated despite it being the case for many real-world applications.

The re-initialisation methods have shown to increase the effectiveness of Genetic Algorithms on Dynamic Multi-Objective Problems without significant impact on the computational cost. These mechanisms outperform the hypermutation-based algorithms and exhibit comparable performance to niching-based approaches. Three re-initialisation schemes: CER-POF, VP and CER-POS, are selected as the most promising mechanisms for dynamic problems due to their combination of high performance and flexibility, where the Controlled Extrapolation with a Pareto Optimal Set based distance approach (CER-POS) [Biswas et al., 2014] is introduced in order to determine the importance of different nearest-distance definitions on dynamic problems.

3 A test set for constrained dynamic problems

To increase the range of dynamic problems this paper develops a set of 15 novel two-objective Constrained Dynamic Functions (CDF)[1], which reflect the constrained and discontinuous problem types found in many applications. These modify the Constrained Functions (CF) developed for the CEC 09 multiobjective competition [Zhang et al., 2009] according to the dynamic function framework proposed by Biswas et.al [Biswas et al., 2014], as it allows straightforward modification of static functions into dynamic problems. Only two-objective problems are considered as these allow a simpler interpretation of the results, without issues of scaling, while providing enough complexity to resemble a number of real-world problems, such as [Bagheri et al., 2014, Wismans et al., 2014, Zhang, 2008, Effendy et al., 2017].

3.1 General summary of dynamic problems

According to Farina et.al [Farina et al., 2004] 4 types of dynamics can be distinguished: type I, where changes are made to the Pareto Optimal Set only; type III, where the Pareto Optimal Front is dynamic; type II, where both the Pareto Optimal Set and the Front are subject to environmental changes and type IV, where only the objective function or constraints are altered, but both the Pareto Optimal Front and the Set remain static. Part of developing a new set of constrained problems is determining to what extent the constraint should be time dependant in relation to the Pareto Optimal Front and the Set, three cases are considered:

[1] These functions are available in C++ at: https://www.bitbucket.org/Pag1c18

a) A static constraint where no time dependency is introduced. However, despite the constraint itself being static the effect of the constraint, and the resulting regions where there are no feasible solutions, are not constant due to the time dependency of the search and objective spaces. Therefore, the Pareto Optimal Front and Set change with respect to the resulting gaps.
b) A dynamic constraint where the time dependency is bonded to the dynamic change of the optimisation problems Pareto Optimal Front and Set. In this case, the applied constraints follow the dynamic change in the pattern of the optimisation problem as the same time-dependency is applied to both of them. Therefore, the resulting gaps in both the Pareto Optimal Set and Pareto Optimal Front do not change.
c) A dynamic constraint where the dynamic change is independent from the change in the optimisation problem. In this case, the constraint is subject to different dynamic changes than those applied to both the Pareto Optimal Set and Pareto Optimal Front, or they remain static. Therefore, the resulting gaps on both the search and objective spaces shift independently every time step, making the problem harder to solve.

Table 1: Summary of the Constrained Dynamic Test set

Name	Type of dynamic problem	Type of constraint	Continuity	Type of POF/POS change	Constraints
CDF1	I	Dynamic	Continuous	Curvature change	2 variable constriants
CDF2	I	Static	Continuous	Curvature change; horizontal shift	1 variable constraint
CDF3	I	Dynamic	Discontinuous	Curvature change	1 variable constraint
CDF4	I	Static	Discontinuous	Vertical shift	1 objective constraint
CDF5	II	Dynamic	Continuous	Diagonal shift	1 variable constraint
CDF6	II	Static	Continuous	Diagonal shift	2 variable constriants
CDF7	II	Dynamic	Discontinuous	Curvature change; diagonal shift	1 objective constraint
CDF8	II	Static	Discontinuous	Curvature change	1 objective constraint
CDF9	III	Dynamic	Continuous	Curvature change; diagonal shift	2 variable constriants
CDF10	III	Static	Continuous	Curvature change	1 variable constraint
CDF11	III	Dynamic	Discontinuous	Horizontal shift	1 variable constraint
CDF12	III	Static	Discontinuous	Curvature change; angular shift	1 objective constraint
CDF13	II	Static	Discontinuous	Random curvature change or shift	1 objective constraint
CDF14	IV	Dynamic	Discontinuous	Continuity change	1 objective constraint
CDF15	IV	Dynamic	Discontinuous	Discontinuity shift	1 objective constraint

3.2 Test instances

In this paper, the CDF test set is developed to ensure a range of different problem types by considering all 3 cases, with changes to the location of the Pareto Optimal Set or Front, changes to the geometries of the search and objective space and dynamic or static constraints introduced to the variables and objectives. Due to the quantity of potential combinations for these parameters, orthogonal design is used to develop 12 representative functions as described in Table 1; this is performed according to the types of dynamic problems and constraints, changes in Pareto Optimal Front and Pareto Optimal Set, and the continuity of both Pareto Optimal Front and Pareto Optimal Set. This generates a set of problems including at least one of each of the type I-III problems with different Pareto Optimal Front and Pareto Optimal Set geometries. Additionally, one problem with a random dynamic change is introduced, as the random element greatly increases the complexity of a problem as the change cannot be predicted; and two problems of type IV are added in order to investigate the different types of time-dependency in constraints with static variable and objective spaces. All of the functions are to be minimised and have 10 decision variables.

CDF1: is generated by combining the UDF2 problem with two variable constraints; the resulting function is a Type I problem. This combination allows investigation of the behaviour of algorithms on continuous objective spaces in dynamic Pareto Optimal Set environments. However, as the constraint is also dynamic relative to the change in $sin(t)$ the infeasible regions are not observed over time, which reduces the overall complexity of the problem. The problem is described on eq. 1 with the Pareto Optimal Front illustrated in Fig. 1.

$$
\begin{gathered}
f_1 = x_1 + \tfrac{2}{|J1|} \sum_{j \in J_1} (x_i x_1^{y_j})^2 \\
f_2 = (1 - x_1)^2 + \tfrac{2}{|J2|} \sum_{j \in J_2} (x_i x_1^{y_j})^2 \\
y_j = 0.5(1 + \tfrac{3(j-2)}{n-2}) + |G(t)|, \quad j = 2, \dots, n \\
x \in [0,1] \times [-1,2]^{n-1} \\
g_1(x,t) = \begin{cases} x_2 - x_1^{(0.5*2+|G(t)|)-sign(k_1)\sqrt{|k_1|}} \\ k_1 = 0.5(1-x_1) - (1-x_1)^2 \end{cases} \\
g_2(x,t) = \begin{cases} x_4 - x_1^{(0.5*(2+\frac{3*(4-2)}{n-2})+|G(t)|)-sign(k_2)\sqrt{|k_2|}} \\ k_2 = 0.25\sqrt{1-x_1} - 0.5(1-x_1) \end{cases} \\
POF : \begin{cases} f_2 = (1-f_1)^2 \\ 0 \leq f_1 \leq 1 \end{cases}
\end{gathered} \tag{1}
$$

CDF2: introduces a vertical shift of the Pareto Optimal Set into the CF4; the resulting problem is Type I. In comparison to CDF1, the introduced constraint is static and limits the obtainable Pareto Optimal Front values on the right-hand side of objective space. The complexity is due to the high multimodality with an uneven distribution of points across the Pareto Optimal Front. The problem is described in eq. 2 with the Pareto Optimal Front illustrated in Fig. 2.

$$\begin{aligned}
f_1 &= x_1 + \sum_{j \in J_1} w_j(y_j) \\
f_2 &= 1x_1 + \sum_{j \in J_2} w_j(y_j) \\
y_j &= x_j sin(6\pi x_1 + \tfrac{(j\pi}{n}), \quad j = 2, \dots, n \\
w_2(z) &= \begin{cases} |z| \quad \text{if} \quad z < 1.5(1-0.5\sqrt{2}) \\ 0.125 + (z-1)^2 \quad \text{otherwise} \end{cases} \\
w_j(z) &= (z - G(t))^2, \quad j = 3, \dots, n \\
x &\in [0,1] \times [-2,2]^{n-1} \\
g(x,t) &= \begin{cases} \frac{a}{1+e^{4|a|}} \\ a = x_2 - sin(6\pi x_1 + \frac{2\pi}{n})0.5x_1 + 0.25 \end{cases} \\
POF &: \begin{cases} f_2 = \begin{cases} 1 - f_1 \quad \text{if} \quad 0 \le f_1 \le 0.5 \\ -0.5f_1 + 0.75 \quad \text{else if} \quad f_1 \le 0.75 \\ 1 - f_1 + 0.125 \quad \text{else if} \quad f_1 \le 1 \end{cases} \\ 0 \le f_1 \le 1 \end{cases}
\end{aligned} \tag{2}$$

CDF3: is a combination of the UDF3 with a simple dynamic, variable constraint. The resulting function is a type I problem, but unlike CDF1 and CDF2, a high discontinuity of both objective and search spaces can be observed and CDF3 requires a higher diversity of the search. The problem is described in eq. 3 with the Pareto Optimal Front illustrated in Fig. 3.

$$\begin{aligned}
f_1 &= x_1 + h + \frac{2}{|J1|} \sum_{j \in J_1} (x_i x_1^{y_j})^2 \\
f_2 &= 1 - x_1 + h + \frac{2}{|J2|} \sum_{j \in J_2} (x_i x_1^{y_j})^2 \\
h &= (\frac{0.5}{N} + \epsilon)|sin(2N\pi x_1)| \\
y_j &= 0.5(2 + \frac{3(j-2)}{n-2}) + |G(t)|, \quad j = 2, \dots, n \\
x &\in [0,1] \times [-1,1]^{n-1} \\
N &= 10; \epsilon = 0.1 \\
g(x,t) &= \begin{cases} x_2 - x_1^{(1+|G(t)|)} \end{cases} \\
POF &: \begin{cases} f_1 = \frac{i}{2N} \\ f_2 = 1 - \frac{i}{2N} \\ for \quad i = 0, 1, \dots, 2N \end{cases}
\end{aligned} \tag{3}$$

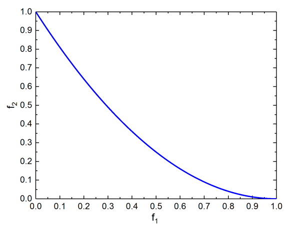


Fig. 1: Pareto Optimal Front for CDF1 over the first 11 times steps

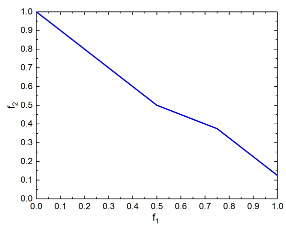


Fig. 2: Pareto Optimal Front for CDF2 over the first 11 times steps

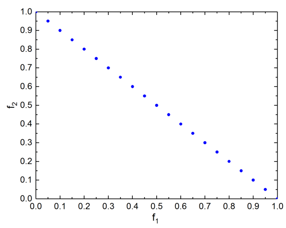


Fig. 3: Pareto Optimal Front for CDF3 over the first 11 times steps

CDF4: is generated by combining the objective constraint from CF1 with a UDF2 dynamic shift; the resulting function is a Type I problem. The complexity of the problem is higher than of CDF3, due to the utilisation of the static constraint where the relative position of the infeasible region changes every time step. The problem is described in eq. 4 with the Pareto Optimal Front illustrated in Fig. 4.

$$
\begin{array}{c}
f_1 = x_1 + \frac{2}{|J1|} \sum_{j \in J_1} (x_i x_1^{y_j})^2 \\
f_2 = 1 - x_1^2 + \frac{2}{|J2|} \sum_{j \in J_2} (x_i x_1^{y_j})^2 \\
y_j = 0.5(1 + \frac{3(j-2)}{n-2}) + |G(t)|, \quad j = 2, \dots, n \\
x \in [0,1] \times [-2,2]^{n-1} \\
g(x,t) = f_1 + f_2 - |sin(N\pi(f_1 - f_2 + 1))| - 1 \\
N = 10 \\
POF^* : \begin{cases} f_2 = 1 - f_1^2 \\ 0 \leq f_1 \leq 1 \end{cases}
\end{array}
\tag{4}
$$

CDF5: the CF5 is combined with the UDF1 constraint resulting in CDF5, which is a type II problem with a dynamic constraint. A diagonal shift in both the Pareto Optimal Set and Front is introduced, and infeasibility is not dependant on time. The high complexity of this problem is the result of the high multimodality and an uneven distribution of points across the Pareto Optimal Front, which is similar to CDF2. The problem is described in eq. 5 with the Pareto Optimal Front illustrated in Fig. 5.

$$
\begin{array}{c}
f_1 = x_1 + |G(t)| + \sum_{j \in J_1} w_j(y_j) \\
f_2 = 1x_1 + |G(t)| + \sum_{j \in J_2} w_j(y_j) \\
y_j = \begin{cases} x_j 0.8 x_1 cos(6\pi x_1 + \frac{j\pi}{n}) - G(t), & j \in J_1 \\ x_j 0.8 x_1 sin(6\pi x_1 + \frac{j\pi}{n}) - G(t), & j \in J_2 \end{cases} \\
w_2(z) = \begin{cases} |z| \quad \text{if} \quad z < 1.5(1 - 0.5\sqrt{2}) \\ 0.125 + (z-1)^2 \quad \text{otherwise} \end{cases} \\
w_j(z) = 2z^2 - cos(4\pi z) + 1, \quad j = 3, \dots, n \\
x \in [0,1] \times [-2,2]^{n-1} \\
g(x,t) = x_2 - 0.8x_1 sin(6\pi x_1 + \frac{2\pi}{n}) 0.5x_1 + 0.25 - G(t) \\
POF : \begin{cases} f_1 = y_1 + |G(t)|; \quad f_2 = y_2 + |G(t)| \\ y_2 = \begin{cases} 1 - y_1 \quad \text{if} \quad 0 \leq y_1 \leq 0.5 \\ -0.5y_1 + 0.75 \quad \text{else if} \quad y_1 \leq 0.75 \\ 1 - y_1 + 0.125 \quad \text{else if} \quad y_1 \leq 1 \end{cases} \\ 0 \leq y_1 \leq 1 \end{cases}
\end{array}
\tag{5}
$$

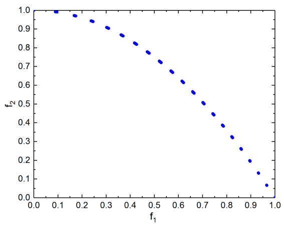


Fig. 4: Pareto Optimal Front for CDF4 over the first 11 times steps

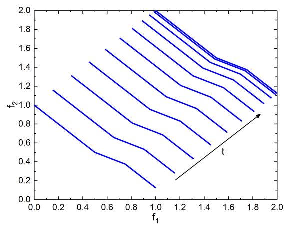


Fig. 5: Pareto Optimal Front for CDF5 over the first 11 times steps

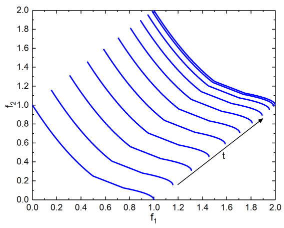


Fig. 6: Pareto Optimal Front for CDF6 over the first 11 times steps

CDF6: the UDF1 is combined with the CF6 constraint resulting in a type II problem. This problem is similar to the CDF5 in terms of the changes in both Pareto Optimal Set and Front. Two constraints are incorporated, compared to one in the CDF5 and the utilised objective function is easier to solve due to the higher uniformity of the Pareto Optimal Front. The problem is described in eq. 6 with the Pareto Optimal Front illustrated in Fig. 6.

$$
\begin{gathered}
f_1 = x_1 + |G(t)| + \sum_{j \in J_1} (y_j)^2 \\
f_2 = (1x_1)^2 + |G(t)| + \sum_{j \in J_2} (y_j)^2 \\
y_j = \begin{cases} x_j 0.8 x_1 sin(6\pi x_1 + \frac{j\pi}{n}), & j = 2,4 \\ x_j 0.8 x_1 cos(6\pi x_1 + \frac{j\pi}{n}) - |G(t)|, & j \in J_1 \\ x_j 0.8 x_1 sin(6\pi x_1 + \frac{j\pi}{n}) - |G(t)|, & j \in J_2 - \{2,4\} \end{cases} \\
x \in [0,1] \times [-2,2]^{n-1} \\
g_1(x,t) = \begin{cases} x_2 - |G(t)| - 0.8 x_1 sin(6\pi x_1 + \frac{2\pi}{n}) sign(k_1)\sqrt{|k_1|} \\ k_1 = 0.5(1-x_1) - (1-x_1)^2 \end{cases} \\
g_2(x,t) = \begin{cases} x_4 - |G(t)| - 0.8 x_1 sin(6\pi x_1 + \frac{4\pi}{n}) sign(k_2)\sqrt{|k_2|} \\ k_2 = 0.25\sqrt{1-x_1} - 0.5(1-x_1) \end{cases} \\
POF: \begin{cases} f_1 = y_1 + |G(t)|; \quad f_2 = y_2 + |G(t)| \\ y_2 = \begin{cases} (1-y_1)^2 & \text{if} \quad 0 \le y_1 \le 0.5 \\ 0.5(1-y_1) & \text{else if} \quad y_1 \le 0.75 \\ 0.25\sqrt{1-y_1} & \text{else if} \quad y_1 \le 1 \end{cases} \\ 0 \le y_1 \le 1 \end{cases}
\end{gathered} \tag{6}
$$

CDF7: is generated by combining the CF1 with a UDF2 dynamic shift; the resulting function is a Type II problem. This combination allows investigation of the behaviour of algorithms on highly discontinuous spaces in dynamic Pareto Optimal Set and Front environments. However, as the constraint is also dynamic then changes in the infeasible regions are not observed over time, which reduces the overall complexity of the problem. The problem is described in eq. 7 with the Pareto Optimal Front illustrated in Fig. 7. This function allows an investigation into the performance of algorithms based on a diagonal shift in the Pareto Optimal Front and vertical shift with curvature change of the Pareto Optimal Set on a highly discontinuous space, without relative changes to the positions of the infeasible regions over time.

$$
\begin{array}{c}
f_1 = x_1 + |G(t)| + \frac{2}{|J_1|} \sum_{j \in J_1} (x_j - x_1^{y_j} - G(t))^2 \\
f_2 = 1x_1 + |G(t)| + \frac{2}{|J_2|} \sum_{j \in J_2} (x_j - x_1^{y_j} - G(t))^2 \\
y_j = 0.5(1 + \frac{3(j-2)}{n-2}), \quad j = 2, \dots, n \\
x \in [0,1] \times [-2,2]^{n-1} \\
g(x,t) = f_1 + f_2 2|G(t)| - |sin[N\pi(f_1 - f_2 + 1)]| - 1 \\
N = 10 \\
POF: \text{ points: } (\frac{i}{2N} + |G(t)|, 1 - \frac{i}{2N} + |G(t)|), \quad i = 0, 1, \dots, 2N
\end{array} \tag{7}
$$

CDF8: the UDF5 is combined with the CF3 constraint to form CDF4. This problem has a similar Pareto Optimal Front pattern to CDF4. However, an angular shift of the Pareto Optimal Set with a curvature change is included and this function introduces a constraint in a more continuous objective space. In addition, the objective function is less complex but a larger search space is utilised. The problem is described in eq. 8 with the Pareto Optimal Front illustrated in Fig. 8.

$$
\begin{array}{c}
f_1 = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} (x_j - x_1^{y_j})^2 \\
f_2 = 1M(t)(x_1)^{H(t)} + \frac{2}{|J_2|} \sum_{j \in J_2} (x_j - x_1^{y_j})^2 \\
y_j = 0.5(2 + \frac{3(j-2)}{n-2}), \quad j = 2, \dots, n \\
x \in [0,1] \times [-1,2]^{n-1} \\
g(x,t) = \sqrt{f_1} + f_2 - sin[2\pi(\sqrt{f_1} - f_2 + 1)] - 1 \\
POF^*: f_2 = 1M(t)(f_1)^{H(t)}, \quad 0 \le f_1 \le 1
\end{array} \tag{8}
$$

CDF9: is a modified CDF6 function, with the dynamic change removed from the Pareto Optimal set and instead added to the constraints. This results in a Type III problem, with a significant curvature change in the Pareto Op-

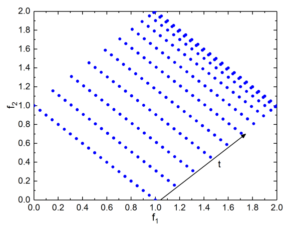


Fig. 7: Pareto Optimal Front for CDF7 over the first 11 times steps

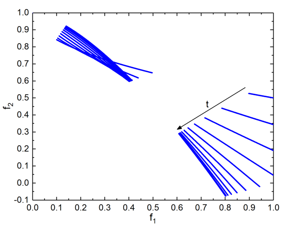


Fig. 8: Pareto Optimal Front for CDF8 over the first 11 times steps

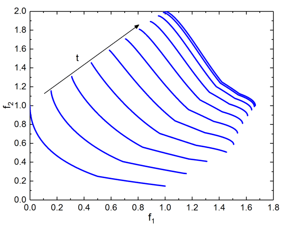


Fig. 9: Pareto Optimal Front for CDF9 over the first 11 times steps

timal front. The problem is described in eq. 9 with the Pareto Optimal Front illustrated in Fig. 9.

$$
\begin{gathered}
f_1 = x_1 + |G(t)| + \sum_{j \in J_1} (y_j)^2 \\
f_2 = [1(M(t)x_1)^{H(t)}]^2 + |G(t)| + \sum_{j \in J_2} (y_j)^2 \\
y_j = \begin{cases} x_j 0.8 x_1 cos(6\pi x_1 + \frac{j\pi}{n}), & j \in J_1 \\ x_j 0.8 x_1 sin(6\pi x_1 + \frac{j\pi}{n}), & j \in J_2 - \{2,4\} \end{cases} \\
x \in [0,1] \times [-2,2]^{n-1} \\
g_1(x,t) = \begin{cases} x_2 - |G(t)| - 0.8 x_1 sin(6\pi x_1 + \frac{2\pi}{n}) sign(k_1)\sqrt{|k_1|} \\ k_1 = 0.5[1(M(t)x_1)^{H(t)}] - [1(M(t)x_1)^{H(t)}]^2 \end{cases} \\
g_2(x,t) = \begin{cases} x_4 - |G(t)| - 0.8 x_1 sin(6\pi x_1 + \frac{4\pi}{n}) sign(k_2)\sqrt{|k_2|} \\ k_2 = 0.25\sqrt{1(M(t)x_1)^{H(t)}} - 0.5[1(M(t)x_1)^{H(t)}] \end{cases} \\
POF: \begin{cases} f_1 = y_1 + |G(t)|; \quad f_2 = y_2 + |G(t)| \\ y_2 = \begin{cases} [1z]^2 & \text{if} \quad 0 \le z \le 0.5 \\ 0.5[1-z] & \text{else if} \quad z \le 0.75 \\ 0.25\sqrt{1z} & \text{else if} \quad z \le 1 \end{cases} \\ z = (M(t)y_1)^{H(t)} \\ 0 \le y_1 \le 1 \end{cases}
\end{gathered}
\tag{9}
$$

CDF10: is generated by adding curvature variation of the Pareto Optimal Front to the CF7; the resulting function is a Type III problem. Due to the utilisation of the static constraint the time complexity of the problem is higher than for CDF9, but the complexity overall is lower. The problem is described in eq. 10 with the Pareto Optimal Front illustrated in Fig. 10;

$$
\begin{gathered}
f_1 = x_1 + \sum_{j \in J_1} (w_j(y_j))^2 \\
f_2 = (1 - x_1)^{H(t)} + \sum_{j \in J_2} (w_j(y_j))^2 \\
y_j = \begin{cases} x_j cos(6\pi x_1 + \frac{j\pi}{n}), & j \in J_1 \\ x_j sin(6\pi x_1 + \frac{j\pi}{n}), & j \in J_2 \end{cases} \\
w_2(z) = w_4(z) = z^2 \\
w_j(z) = 2z^2 - cos(4\pi z) + 1, \quad j = 3, \ldots, n \\
x \in [0,1] \times [-2,2]^{n-1} \\
g(x,t) = x_2 - 0.8 x_1 sin(6\pi x_1 + \frac{2\pi}{n}) 0.5 x_1 + 0.25 - G(t) \\
POF: \begin{cases} f_2 = \begin{cases} (1-f_1)^{H(t)} & \text{if} \quad 0 \le f_1 \le 0.5 \\ (1-f_1)^{H(t)} - (1-f_1)^2 + 0.5 * (1-f_1) & \text{else if} \quad f_1 \le 0.75 \\ (1-f_1)^{H(t)}(1-f_1)^2 + 0.25 * \sqrt{1-f_1} & \text{else if} \quad f_1 \le 1 \end{cases} \\ 0 \le f_1 \le 1 \end{cases}
\end{gathered}
\tag{10}
$$

CDF11: is a modified CDF3 function, with the dynamic change removed from the Pareto Optimal set, but added to the constraints. This results in a Type III problem, with a vertical change of the Pareto Optimal front rather than changes in the curvature. The problem is described in eq. 11 with the Pareto Optimal Front illustrated in Fig. 11.

$$
\begin{aligned}
f_1 &= x_1 + |G(t)| + h + \sum_{j\in J_1} w_j(y_j)\\
f_2 &= 1x_1 + |G(t)| + h + \sum_{j\in J_2} w_j(y_j)\\
h &= (\tfrac{0.5}{N} + \epsilon)|sin(2N\pi x_1 + G(t)\pi)|\\
y_j &= \begin{cases} x_j 0.8x_1 cos(6\pi x_1 + \frac{j\pi}{n}), & j \in J_1\\ x_j 0.8x_1 sin(6\pi x_1 + \frac{j\pi}{n}), & j \in J_2 \end{cases}\\
w_2(z) &= \begin{cases} |z| \quad \text{if} \quad z < 1.5(1-0.5\sqrt{2})\\ 0.125 + (z-1)^2 \quad \text{otherwise} \end{cases}\\
w_j(z) &= 2z^2 - cos(4\pi z) + 1, \quad j = 3,\dots,n\\
& x \in [0,1] \times [-1,1]^{n-1}\\
& N = 10; \epsilon = 0.1\\
g(x,t) &= x_2 - 0.8x_1 sin(6\pi x_1 + \tfrac{2\pi}{n})0.5x_1 + 0.25\\
POF: & \begin{cases} f_1 = \frac{i-G(t)}{2N}\\ f_2 = \begin{cases} 1 - f_1 \quad \text{if} \quad 0 \le f_1 \le 0.5\\ -0.5f_1 + 0.75 \quad \text{else if} \quad f_1 \le 0.75\\ 1 - f_1 + 0.125 \quad \text{else if} \quad f_1 \le 1 \end{cases}\\ for \quad i = 0,1,\dots,2N \end{cases}
\end{aligned}
\tag{11}
$$

CDF12: introduces angular shift and curvature variations of the Pareto Optimal Front into the CF2. A static objective constraint is introduced and the resulting function is a Type III problem, as no environmental change is applied to the Pareto Optimal Set. This function allows the behaviour in the constrained and dynamic Pareto Optimal Front environments to be evalu-

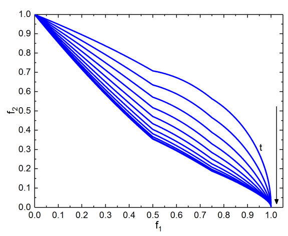


Fig. 10: Pareto Optimal Front for CDF10 over the first 11 times steps

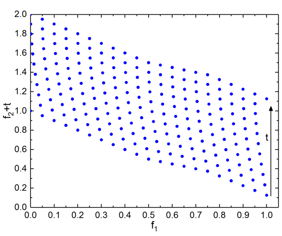


Fig. 11: Pareto Optimal Front for CDF11 over the first 11 times steps

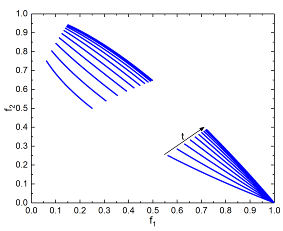


Fig. 12: Pareto Optimal Front for CDF12 over the first 11 times steps

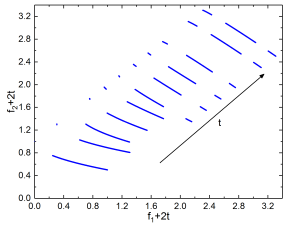


Fig. 13: Pareto Optimal Front for CDF13 over the first 11 times steps

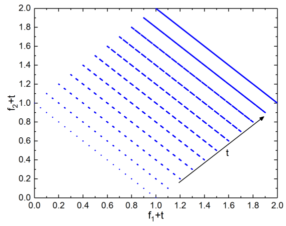


Fig. 14: Pareto Optimal Front for CDF14 over the first 11 times steps

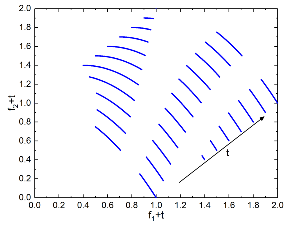


Fig. 15: Pareto Optimal Front for CDF15 over the first 11 times steps

ated. The problem is described in eq. 12 with the Pareto Optimal Front illustrated in Fig. 12.

$$
\begin{gathered}
f_1 = x_1 + \tfrac{2}{|J_1|} \sum_{j \in J_1} y_j^2 \\
f_2 = 1(x_1)^{H(t)} + \tfrac{2}{|J_2|} \sum_{j \in J_2} y_j^2 \\
y_j = \begin{cases} x_j 0.8 x_1 cos(6\pi x_1 + \frac{j\pi}{n}), & j \in J_1 \\ x_j 0.8 x_1 sin(6\pi x_1 + \frac{j\pi}{n}), & j \in J_2 \end{cases} \\
x \in [0,1] \times [-1,1]^{n-1} \\
g(x,t) = \begin{cases} \frac{a}{1+e^{4|a|}} \\ a = \sqrt{f_1} + f_2 - sin[2\pi(\sqrt{f_1} - f_2 + 1)] - 1 \end{cases} \\
POF^* : f_2 = 1(f_1)^{H(t)}, \quad 0 \le f_1 \le 1
\end{gathered} \tag{12}
$$

CDF13: the UDF8 is combined with the CF4 constraint resulting in the CDF13. This problem is type II, and uses the same objective function as CDF9. However, the constraint is dynamic as a random change is introduced. The random element greatly increases the complexity of the problem as the change cannot be predicted. The problem is described in eq. 13 with the Pareto Optimal Front illustrated in Fig. 13.

In CDF13, several types of dynamic changes are incorporated, in a similar manner to UDF8, but only one of them is employed at a time; during the environmental change one of t1 to t5 is selected randomly and is increased by 1.

$$
\begin{gathered}
f_1 = x_1 + |G(t_3)| + \tfrac{2}{|J_1|} \sum_{j \in J_1} y_j^2 \\
f_2 = 1M(t_4)(x_1)^{H(t_5)} + |G(t_3)| + \tfrac{2}{|J_2|} \sum_{j \in J_2} y_j^2 \\
y_j = x_j sin(6\pi x_1 + \tfrac{(j+K(t_1))\pi}{n}) G(t_2), \quad j = 2, \ldots, n \\
x \in [0,1] \times [-2,2]^{n-1} \\
g(x,t) = \begin{cases} w + f_2 - sin[2\pi(w - f_2 + 1)] - 1 \\ w = M(t_4)(f_1)^{H(t_5)} \end{cases} \\
POF^* : \begin{cases} f_2 = 1M(t_4)(f_1 - |G(t_3)|)^{H(t_5)} + |G(t_3)|, \\ 0 + |G(t_3)| \le f_1 \le 1 + |G(t_3)| \end{cases}
\end{gathered} \tag{13}
$$

CDF14: introduces a time-dependent constraint into the static CF1. The additional constraint is independent of the change in objective function and the result is a Type IV problem. The time-dependency of the constraint changes the geometry of the Pareto Optimal Front and the Set over time from distinct points, through to disconnected fronts and finally to a continuous set. This means that retaining the diversity of solutions over time steps is essential, especially when the sizes of the infeasible regions are decreasing. Due to this the boundary ranges are lowered, in order to decrease the complexity. The problem is described in eq. 14 with the Pareto Optimal Front illustrated in Fig. 14.

$$\begin{gathered} f_1 = x_1 + \tfrac{2}{|J_1|} \sum_{j \in J_1} (x_j - x_1^{y_j})^2 \\ f_2 = 1x_1 + \tfrac{2}{|J_2|} \sum_{j \in J_2} (x_j - x_1^{y_j})^2 \\ y_j = 0.5(1 + \tfrac{3(j-2)}{n-2}), \quad j = 2, \dots, n \\ x \in [0,1]^n \\ g(x,t) = f_1 + f_2 - |sin[N\pi(f_1 - f_2 + 1)]| - 1 + |G(t)| \\ N = 10 \\ POF^* : f_2 = 1 - f_1, \quad 0 \le f_1 \le 1 \end{gathered} \tag{14}$$

CDF15: is Type IV, similar to CDF14, as the time-dependent constraint is introduced into the CF3, without any direct environmental changes in objective or variable spaces. However, in this case the position of the infeasible regions is shifting in both the search and objective spaces, instead of changing the size of them. The problem is described in eq. 15 with the Pareto Optimal Front illustrated in Fig. 15.

$$\begin{gathered} f_1 = x_1 + \tfrac{2}{|J_1|} \sum_{j \in J_1} y_j^2 \\ f_2 = 1x_1^2 + \tfrac{2}{|J_2|} \sum_{j \in J_2} y_j^2 \\ y_j = x_j - sin(6\pi x_1 + \tfrac{j\pi}{n}), \quad j = 2, \dots, n \\ x \in [0,1] \times [-2,2]^{n-1} \\ g(x,t) = f_1^2 + f_2 sin[2\pi(f_1^2 - f_2 + 1) + G(t)] - 1 \\ POF^* : f_2 = 1 - f_1^2, \quad 0 \le f_1 \le 1 \end{gathered} \tag{15}$$

4 Methodology

4.1 Selection of mechanisms and unconstrained test problems

The performances of different Genetic Algorithms are evaluated in combination with different re-initialization methods over the developed test instances, CDF1-15. Furthermore, eighteen bi-objective unconstrained functions are used to evaluate the differences in behaviour of the utilised solvers between the constrained and unconstrained problems. The unconstrained functions are UDF1-9 [Biswas et al., 2014], except for UDF7 as it is a three-objective problem, JY1-8 [Jiang and Yang, 2017], except JY4 as the parameters given in [Jiang and Yang, 2017] do not reflect the problem description and the provided Pareto Optimal Front, and FDA1-3 [Farina et al., 2004]. The selected set provides a range of problems including all types of dynamics, I-IV, different kinds of Pareto Optimal Front and Set changes, varying Pareto Optimal Front and Set geometries, and problem characteristics: modalities, biases and variable-linkages.

Six different Genetic Algorithms are selected for the benchmarking due to their high performance: MOEA/D, as the best algorithm for static and dynamic unconstrained problem types [Zhang and Suganthan, 2009, **?**]; NSGA-II [Deb et al., 2002] as the most commonly utilised Genetic Algorithm with high performance on static constrained and unconstrained problems; MTS [Tseng and Chen, 2009] due to the high performance on constrained and unconstrained static problems while also representing a Genetic Algorithm with no crossover, it has a high potential for dynamic problems as it demonstrates rapid convergence but is currently untested on these problems; BCE [Li et al., 2016] and HEIA [Lin et al., 2016] as examples of the current state-of-the-art of static Genetic Algorithms, with top performance on static problems and a higher emphasis on diversity; and the MLSGA-MTS hybrid [Grudniewski and Sobey, 2018], in MLS1 configuration, which shows the best general performance on static problems where the addition of the MLS1 mechanisms have been shown to increase the convergence speed [Sobey and Grudniewski, 2018], essential in dynamic optimisation. Each of the six selected algorithms, MOEA/D, NSGA-II, MTS, BCE, HEIA and MLSGA-MTS, are combined with the 3 different re-initialisation methodologies: VP, CER-POS and CER-POF, and compared to results without any re-initialisation mechanisms. DEMO [Farina et al., 2004], DMEA/PRI [Zhou et al., 2007] and DQ-MOO/AR [Hatzakis and Wallace, 2006], are not selected as they are shown to be outperformed by MOEA/D [Biswas et al., 2014]. DNSGA-II [Deb et al., 2007] is also not included as it is outperformed by other methods and not compatible with the selected re-initialisation mechanisms. dCOEA [Goh and Tan, 2009] is not selected as it performs poorly on discontinuous search and objective spaces, while showing similar performance on other problems to MOEA/D [Biswas et al., 2014]. Only Genetic Algorithms are considered, instead of the other Evolutionary Algorithms, as the principle of this paper is not to find the best solver for each kind of problem, but rather to evaluate the impact of implementing constraints into dynamic problems on Genetic Algorithms.

In addition, two constraint-handling approaches are tested against each other in order to minimise the potential bias of a single technique towards a certain Genetic Algorithm or re-initialisation methodology: the dominance-based [Deb et al., 2002] and the adaptive penalty-based [Azzouz et al., 2018] constraint-handling methods are selected as the most commonly utilised techniques.

4.2 Benchmarking parameters

The benchmarks are conducted using two different combinations of dynamic settings, the time change, T, and n_s. T denotes the time-window when the problem remains static, as the number of generations; n_s denotes the severity of change. The first set of benchmarks represents a harsh dynamic environment with both values taken to be 5. In the second set of benchmarks, developed to demonstrate optimisation in a less harsh dynamic condition, both of the parameters are set equal to 10. These two tests are conducted in order to eliminate the selection of certain parameters which promote the performance of certain algorithms or mechanisms, especially the importance of diversity and convergence rate. However, the results show similar trends in both cases, and therefore only the results for T and n_s equal to 5 are documented with the values set to 10 documented in the supplementary material. The performance is evaluated based on the Inverted Generational Distance (IGD) [Bosman and Thierens, 2003] and Hyper Volume (HV) [While et al., 2012].

In most benchmarks a relatively small population is used, around 200-500. However, as the presented constrained functions are more challenging, larger population sizes may be required to prevent the algorithm from getting stuck in an infeasible region of the search space and premature convergence. Therefore, two population sizes of 300 and 1000 are evaluated to avoid the situation where the population size is the limiting factor for the algorithm, i.e. it is not enough to develop the necessary diversity of solutions. However, the population size does not affect the trends in the results, equally reducing the overall performance of each algorithm meaning no mechanism is unfairly penalised. Therefore, the results for the 300 population sizes are attached as a data supplement but are not documented in the paper.

In most benchmarks of Genetic Algorithms on dynamic problems, the termination criterion utilised is the number of generations and the time change occurs after T generations [Jiang and Yang, 2017, Biswas et al., 2014]. However, the number of objective function evaluations conducted at each generation is not uniform across different methodologies. For NSGA-II it is the population size and for MOEA/D it is the number of new offspring, which is significantly lower than the parent population size, usually 0.1 of the overall population size. For the MTS and MLSGA-MTS algorithms the number of evaluations is a multiple of the population size, ranging from 5x to 50x, as multiple local searches are used. Comparing the performance of these algorithms using the number of generations gives a wide range of different function calls and does not provide a fair benchmark. Furthermore, the total number of generations is irrelevant in most cases, as the calculation time per generation is normally relatively small in comparison to the objective function evaluation. Therefore, the selected stopping criterion is the number of fitness function evaluations and the time change occurs after a certain number of evaluations. The total number of fitness function evaluations for T and n_s equal to 5 and 10 are therefore 300,000 and 600,000 respectively, which allows 6 cycles of the UDF and CDF [Biswas et al., 2014], and 3 cycles of the JY [Jiang and Yang, 2017] and FDA [Farina et al., 2004]. The time change occurs after population size $\times T$ iterations and is equal to 5000 iterations in the case of T equal to 5 and 10,000 iterations in the second case. All cases are evaluated over 30 independent runs. In order to show the importance of a proper termination condition, a separate test is conducted with the commonly utilised generations termination condition, with 300 generations as the limit. The results show that MTS outperforms all of the other algorithms on almost all of the selected functions and MOEA/D performs poorly due to the disproportionate number of fitness evaluations per generation. Therefore, the data for the generation termination condition is not included but attached as supplementary material to allow comparison to the previous literature. Even though this issue is widely known in static MOPs, the previous dynamic benchmarks have not addressed it [Jiang and Yang, 2017, Biswas et al., 2014] and fitness evaluations limits have not been mentioned or obviously implemented. The obtained results show that future dynamic benchmarking exercises should use fitness evaluation limits, to allow a fair comparison of algorithms across different exercises.

5 Performance Benchmarking

A summary of the top performing algorithms and re-initialization mechanisms is presented in this section for T and n_s equal to 5, using 1000 individuals in the population. Pre-benchmarking simulations show that the dynamic parameters only have a minor impact on the comparative performance of the utilised algorithms and re-initialisation strategies. In the conducted benchmarks the adaptive penalty-based constraint-handling approach leads to significantly better results in comparison to the second technique. However, there is no significant differences in comparative performance of the selected Genetic Algorithms and re-initialisation mechanisms between those constraint-handling techniques. The only exception is that HEIA shows stronger performance with the dominance-based strategy and therefore all of the results shown are from the domination-based approach. The detailed data, with minimum, maximum and average values of the IGD and HV, along with standard deviations and calculation times are added as supplementary material.

The most dominant factor in determining the performance on the dynamic problems is the selection of the Genetic Algorithm. Despite this, the results indicate that in most cases the selection of the re-initialization method can have a significant impact on the final performance. Only in a few, seemingly random, cases is the performance of the Genetic Algorithm independent of these re-initialisation methods. Interestingly, the impact of the re-initialisation strategy is not uniform across different Genetic Algorithms and problems, i.e. on the same problem the VP variant can exhibit the best performance for MTS but for BCE with VP shows the worst performance. Therefore, selection of the best strategy is difficult and further work needs to be performed to determine how this selection can be made in advance. These results are summarised in section 5.1 for the re-initialisation methods and in section 5.2 for the different algorithms to understand the final performance.

5.1 Re-initialisation

In order to compare the different re-initialization methods, the average rank, according to the IGD and HV metrics, of each mechanism is calculated on problems with the same characteristics and presented in Table 2 for T and n_s equal to 5. In most cases, the VP re-initialisation outperforms the other methodologies tested and has the lowest rank in total across all of the problem types, despite the lack of uptake for this method outside of the original paper [Zhou et al., 2007]. The high performance of the VP method is due to the inclusion of the variance module, which is not based on pattern prediction but on random Gaussian noise. The prediction-based strategies utilize historical information from the Pareto Optimal Set at previous generations to approximate the position of the final Pareto Optimal Set in the next time step. Therefore, these methods are inefficient on problems where the Pareto Optimal Set remains static, such as type III problems: FDA2, UDF3, UDF6 and CDF9-12, and the problems where the pattern for how the objectives change cannot be predicted, such as the random change problems: UDF8, UDF9 and CDF13. This is supported by the higher performance of the variant with no re-initialisation over other methods. Industrial applications are often unpredictable, or extremely complex, and so the separation in performance of these mechanisms is likely to be even greater. Furthermore, in the vast majority of cases there is a significant difference between the ranked results according to Wilcoxons rank sum test with a confidence level of $\alpha = 0.05$, especially comparing VP to other methods, showing its superiority.

The results show the importance of the selection of an appropriate re-initialisation mechanism. Importantly, the effect on performance of different strategies is not uniform across the Multi-Objective Evolutionary Algorithms. For example, on the two top performing Genetic Algorithm methodologies, MTS works best with the VP method, but usually there is no statistical significance between the 4 used re-initialisation mechanisms for this methodology, whereas MOEA/D is more likely to show higher performance without any re-initialisation method. In the case of MOEA/D the authors suggest that this is caused by the MOEA/D specific mechanisms such as weight-vectors and neighbourhoods of solutions. In the re-initialisation step these features are not taken into consideration when the new population is created and the parent solutions are selected. Therefore, the overall efficiency is limited and re-initialisation mechanisms specific to MOEA/D might be necessary. This is similar to the performance of MOEA/D with the MLSGA mechanisms [Grudniewski and Sobey, 2018], and it seems the MOEA/D mechanisms are more difficult to blend with additional elements.

Despite the importance of the re-initialisation method, the choice of appropriate method is often neglected in the dynamic Genetic Algorithm lit-

Table 2: The average ranks of each re-initialisation method for the 1000 population size with iterations termination criterion and T and n_s equal to 5

	None	CER-POS	CER-POF	VP
Overall	2.83/ 3.07	2.47/ 2.29	2.74/ 2.49	1.95/ 2.15
Type I	3.02/ 3.43	2.38/ 2.07	2.62/ 2.52	1.98/ 1.98
TypeII	3.09/ 3.39	2.31/ 2.18	2.81/ 2.43	1.79/ 2.00
Type III	2.13/ 2.28	2.81/ 2.69	2.74/ 2.63	2.31/ 2.41
Type IV	3.17/ 3.08	2.33/ 2.33	2.58/ 2.25	1.92/ 2.33
UDF	2.79/ 3.21	2.46/ 2.31	2.81/ 2.40	1.94/ 2.08
CDF	2.78/ 3.06	2.40/ 2.07	2.72/ 2.51	2.10/ 2.37

The best mechanism in each category is highlighted in green.
The second best is highlighted in yellow.

erature and in most cases the re-initialization method is chosen arbitrarily and not compared to the other variants [Biswas et al., 2014] or not utilized at all [Jiang and Yang, 2017]. Therefore, benchmarking of novel re-initialization mechanisms with other variants, on the same algorithm methodologies, and with the algorithm without any diversity preservation mechanisms is necessary, to ensure that new re-initialisation methods provide benefits over the current methods.

5.2 Genetic Algorithm methodologies

In order to investigate the performance of the different algorithms, the average ranks according to the IGD and HV indicators are calculated across all of the dynamic problem types and presented in Table 3 for T and n_s equal to 5. There are significant differences between the ranked results according to Wilcoxons rank sum test with a confidence level of $\alpha = 0.05$ in most of the presented cases. The performances of the Genetic Algorithm mechanisms are less consistent across the utilised problems than for the re-initialisation methods, and it is not easy to choose a single best method. According to the average ranks, the best overall performer is MOEA/D, while MTS and HEIA come second and third respectively. However, MOEA/D is less effective when the Pareto Optimal Set remains static, demonstrated by comparatively poor performance on Type III problems. This is caused by the static weight vectors, where the previous best point according to one weight is no longer the most suitable one after the time change. Therefore, it is likely that the points with the best values may be eliminated quickly, as their vectors are not properly assigned. This problem could be solved by reassignment of each weight after the time changes occur. MTS does not have these disadvantages and is less susceptible to the time changes due to the utilisation of multiple local searches that maintains both the convergence and diversity of the search; this results in a lack of significant differences between the implemented re-initialisation methods. High performance of these high convergence-based

Table 3: The average ranks of each Genetic Algorithm for the 1000 population size with iterations termination criterion and T and n_s equal to 5

	MOEA/D	MTS	NSGA-II	MLSGA	BCE	HEIA
Overall	2.45/ 2.31	2.71/ 4.97	5.10/ 2.70	3.16/ 4.70	4.58/ 2.92	2.99/ 3.39
Type I	2.25/ 2.43	2.89/ 4.89	4.82/ 2.50	3.11/ 4.82	4.64/ 3.04	3.29/ 3.32
TypeII	1.97/ 1.75	2.88/ 5.15	5.15/ 2.87	3.45/ 4.48	4.37/ 3.15	3.18/ 3.60
Type III	3.22/ 3.08	2.33/ 4.89	5.28/ 2.28	2.86/ 5.03	4.72/ 2.61	2.58/ 3.11
Type IV	2.63/ 2.13	2.88/ 4.25	4.88/ 4.63	2.88/ 4.50	5.38/ 1.38	2.38/ 4.13
UDF	2.59/ 2.13	2.56/ 4.91	5.16/ 2.63	3.88/ 4.44	3.91/ 2.91	2.91/ 4.00
CDF	2.37/ 2.32	3.05/ 4.63	4.77/ 3.05	2.93/ 5.55	5.57/ 2.28	2.32/ 3.17

The best algorithm in each category is highlighted: in yellow according to IGD; in blue for HV; and green with both metrics.

solvers, such as MOEA/D, further supports the findings that Dynamic Multi-Objective Problems requires high convergence of the search over diversity, unlike many static problems, which was originally suggested in [Biswas et al., 2014].

The HEIA and MOEA/D exhibit the best overall performance for the constrained results. The high performance of MOEA/D on these problems is interesting as previous research shows poor results on the static constrained problems [Zhang and Suganthan, 2009], and MOEA/D is outperformed by MLSGA-MTS, MTS and NSGA-II [Grudniewski and Sobey, 2018]. Furthermore, for most of the tested algorithms there is no significant difference in comparative performance on the CDFs than on the UDFs, on which the developed set is based. This unexpected behaviour indicates that, for dynamic problems, the fast convergence of MOEA/D is more important than the ability to discover discontinuous search and objective spaces, characteristics where the MOEA/D mechanisms perform poorly. Therefore, the constrained dynamic problems act similarly to the unconstrained cases. These findings further indicate that high convergence rate is essential when dealing with Dynamic Multi-Objective Problems.

5.3 Performance of the different algorithms on the Constrained Functions over time

In order to investigate the behaviour on the developed constrained set, the IGD values obtained by the benchmarked algorithms are tracked during a single run, period $t = 0$ to $t = 11.8$. This is illustrated in Fig. 17 for CDF4 and in Fig. 16 for CDF10, as problems where the behaviour is representative of these rest of the set. In all cases the VP re-initialisation is used, and only the results for MTS, MOEA/D and NSGA-II are shown as the two best and the worst performing algorithm respectively. The figures for the other developed problems, as well as for the HV metric, are attached as supplementary data.

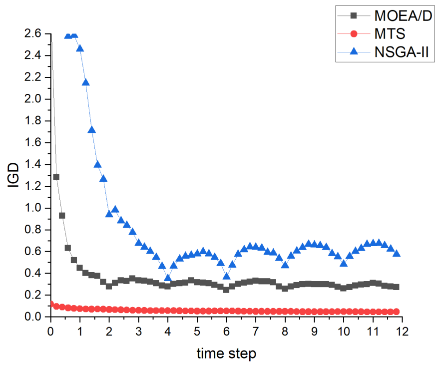


Fig. 16: IGD(t) values obtained by the MOEA/D, MTS and NSGA-II on the CDF10 case.

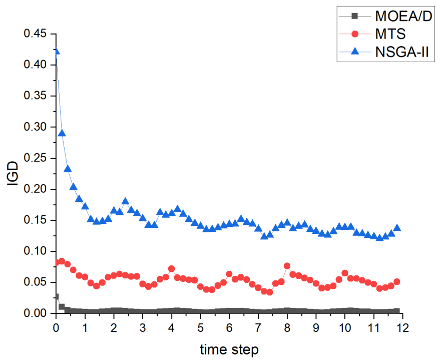


Fig. 17: IGD(t) values obtained by the MOEA/D, MTS and NSGA-II on the CDF4 case.

There is a considerable difference in performance between the top algorithms for a problem and the next best performer. Fig. 16 shows that MTS is able to accurately approximate the true Pareto Optimal Front at every time step for the CDF10 problem, resulting in stable IGD values. NSGA-II is not able to properly converge on the Pareto Optimal Front whereas MOEA/D is unable to find all of the regions of the Pareto Optimal Front on the highly discontinuous search space; this is especially clear after the shift of the constraints. This behaviour of MOEA/D is expected as a similar principle is shown on constrained static problems and is due to the low performance on the Type I problems, as discussed previously. However, a different behaviour is exhibited on the CDF4 function, Fig. 17. In this case MOEA/D outperforms MTS, showing lower IGD values that remain stable at every time step, indicating that the true Pareto Optimal Front has been found, whereas the performance of NSGA-II remains the worst. This is most likely caused by a higher uniformity and continuity of the CDF4 problem in comparison to the CDF10. According to that CDF10 case requires a higher diversity of the search, which MOEA/D is unable to provide, unlike the MTS. The low performance of NSGA-II on both problems, and the effectiveness of MOEA/D, even over MTS on some cases, is contrasting to the trends shown on static problems. This indicates that the dynamicity is the dominant characteristic of the Dynamic Multi-Objective Problems, even over the level of discontinuities in the search spaces.

6 Conclusions

Dynamic optimisation has many applications. However, the current benchmarking problems are limited to unconstrained problems, which may benefit certain types of Genetic Algorithms. In this paper the range of dynamic test problems is expanded by proposing a set of constrained test functions. A number of combinations of current state-of-the-art algorithms and re-initialization methods are then benchmarked to determine the most effective methods for solving constrained dynamic optimisation problems. Overall MOEA/D shows the best performance on both the new constrained and traditional unconstrained test sets. The Variation and Prediction re-initialisation strategy performs the best on most problems but in some cases no significant improvement is shown over no re-initialization. Importantly, the performance of the algorithms on the constrained and unconstrained problems is similar, showing that the dynamic characteristic is dominating these problems. The following recommendations are made:

- The importance of dynamic behaviour dominates the non-continuous characteristic of the constrained problems, as there is no significant difference in the comparative performance of the tested algorithms between the constrained and unconstrained cases. The results show that Genetic Algorithms with strong convergence profiles are best at solving these problems.
- In most cases the obtained solutions are far from the true Pareto Optimal Front. As the highly dynamic problems are dominated by convergence, this puts the Genetic Algorithms at a disadvantage over other methods. Therefore, there is a need for dynamic specific methodologies with high convergence and low diversity.
- The prevalence of the predictable dynamic test problems leads to mechanisms that do well on these problems. However, in most applications the dynamic changes are unpredictable, which makes them more difficult to solve and more random elements should be included in future test sets to develop more realistic mechanisms.

References

Azzouz et al., 2018. Azzouz, R., Bechikh, S., Said, L. B., and Trabelsi, W. (2018). Handling time-varying constraints and objectives in dynamic evolutionary multi-objective optimization. *Swarm and Evolutionary Computation*, 39(April 2017):222–248.

Bagheri et al., 2014. Bagheri, A., Mohammadi Peyhani, H., and Akbari, M. (2014). Financial forecasting using ANFIS networks with Quantum-behaved Particle Swarm Optimization. *Expert Systems with Applications*, 41(14):6235–6250.

Biswas et al., 2014. Biswas, S., Das, S., Suganthan, P. N., and Coello Coello, C. A. (2014). Evolutionary Multiobjective Optimization in Dynamic Environments: A Set of Novel Benchmark Functions. In *2014 IEEE Congress on Evolutionary Computation (CEC 2014)*, volume 1, pages 3192–3199.

Bosman and Thierens, 2003. Bosman, P. A. and Thierens, D. (2003). The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188.

Branke, 2002. Branke, J. (2002). *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic Publishers, Norwell, MA, USA.

Deb et al., 2002. Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deb et al., 2007. Deb, K., Rao N., U. B., and Karthik, S. (2007). Dynamic Multi-Objective Optimization and Decision-Making Using Modified NSGA-II: A Case Study on Hydro-Thermal Power Scheduling. In *Evolutionary Multi-Criterion Optimization*, pages 803–817.

Effendy et al., 2017. Effendy, S., Khan, M. S., Farooq, S., and Karimi, I. A. (2017). Dynamic modelling and optimization of an LNG storage tank in a regasification terminal with semi-analytical solutions for N2-free LNG. *Computers and Chemical Engineering*, 99:40–50.

Farina et al., 2004. Farina, M., Deb, K., and Amato, P. (2004). Dynamic multiobjective optimization problems: Test cases, approximations, and applications. *IEEE Transactions on Evolutionary Computation*, 8(5):425–442.

Goh and Tan, 2009. Goh, C.-K. and Tan, K. C. (2009). A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 13(1):103–127.

Grudniewski and Sobey, 2018. Grudniewski, P. A. and Sobey, A. J. (2018). Behaviour of Multi-Level Selection Genetic Algorithm (MLSGA) using different individual-level selection mechanisms. *Swarm and Evolutionary Computation*, 44(September 2018):852–862.

Hatzakis and Wallace, 2006. Hatzakis, I. and Wallace, D. (2006). Dynamic Multi-ObjectiveOptimization with Evolutionary Algorithms: A Forward-Looking Approach. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pages 1201–1208.

Helbig and Engelbrecht, 2014. Helbig, M. and Engelbrecht, A. P. (2014). Benchmarks for Dynamic Multi-Objective Optimisation Algorithms. *ACM Computational Surveys*, 46(3):37:1–37:39.

Helbig and Engelbrecht, 2015. Helbig, M. and Engelbrecht, A. P. (2015). Benchmark Functions for CEC 2015 Special Session and Competition on Dynamic Multi-objective Optimization. Technical report.

Jiang and Yang, 2017. Jiang, S. and Yang, S. (2017). Evolutionary Dynamic Multiobjective Optimization: Benchmarks and Algorithm Comparisons. *IEEE Transactions on Cybernetics*, 47(1):198–211.

Li et al., 2011. Li, C., Yang, S., and Pelta, D. A. (2011). Benchmark Generator for the IEEE WCCI-2012 Competition on Evolutionary Computation for Dynamic Optimization Problems. Technical report.

Li et al., 2016. Li, M., Yang, S., and Liu, X. (2016). Pareto or Non-Pareto: Bi-criterion evolution in multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 20(5):645–665.

Lin et al., 2016. Lin, Q., Chen, J., Zhan, Z.-H., Chen, W.-N., Coello Coello, C. A., Yin, Y., Lin, C.-M., and Zhang, J. (2016). A Hybrid Evolutionary Immune Algorithm for Multiobjective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 20(5):711–729.

Liu et al., 2017. Liu, H.-L., Chen, L., Deb, K., and Goodman, E. D. (2017). Investigating the effect of imbalance between convergence and diversity in evolutionary multiobjective algorithms. *IEEE Transactions on Evolutionary Computation*, 21(3):408–425.

Mehnen et al., 2006. Mehnen, J., Wagner, T., and Rudolph, G. (2006). Evolutionary optimization of dynamic multi-objective test functions. Technical Report May.

Nguyen et al., 2012. Nguyen, T. T., Yang, S., and Branke, J. (2012). Evolutionary dynamic optimization: A survey of the state of the art. *Swarm and Evolutionary Computation*, 6:1–24.

Sobey and Grudniewski, 2018. Sobey, A. J. and Grudniewski, P. A. (2018). Re-inspiring the genetic algorithm with multi-level selection theory: Multi-level selection genetic algorithm. *Bioinspiration and Biomimetics*, 13(5):1–13.

Sobey and Grudniewski, 2019. Sobey, A. J. and Grudniewski, P. A. (2019). There's no Free Lunch: A Study of Genetic Algorithm Use in Maritime Applications. In *18th Conference on Computer Applications and Information Technology in the Maritime Industries (COMPIT)*.

Tseng and Chen, 2009. Tseng, L.-Y. and Chen, C. (2009). Multiple trajectory search for unconstrained/constrained multi-objective optimization. In *2009 IEEE Congress on Evolutionary Computation (CEC 2009)*, pages 1951–1958.

Wang et al., 2018. Wang, Z., Bai, J., Sobey, A. J., Xiong, J., and Shenoi, A. (2018). Optimal design of triaxial weave fabric composites under tension. *Composite Structures*, 201(June):616–624.

While et al., 2012. While, L., Bradstreet, L., and Barone, L. (2012). A fast way of calculating exact hypervolumes. *IEEE Transactions on Evolutionary Computation*, 16(1):86–95.

Wismans et al., 2014. Wismans, L., Van Berkum, E., and Bliemer, M. (2014). Handling multiple objectives in optimization of externalities as objectives for dynamic traffic management. *European Journal of Transport and Infrastructure Research*, 14(2):159–177.

Zhang and Li, 2007. Zhang, Q. and Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731.

Zhang and Suganthan, 2009. Zhang, Q. and Suganthan, P. N. (2009). Final Report on CEC ' 09 MOEA Competition. Technical report.

Zhang et al., 2009. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., and Liu, W. (2009). Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition. Technical report.

Zhang, 2008. Zhang, Z. (2008). Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control. *Applied Soft Computing Journal*, 8(2):959–971.

Zhou et al., 2007. Zhou, A., Jin, Y., Zhang, Q., Sendhoff, B., and Tsang, E. (2007). Prediction-based Population Re-initialization for Evolutionary Dynamic Multi-objective Optimization. In *Evolutionary Multi-Criterion Optimization*, pages 832–846. Springer Berlin Heidelberg.

Appendix D Predictive prosthetic socket design: part 2 generating person-specific candidate designs using multi-objective genetic algorithms

Biomechanics and Modeling in Mechanobiology
https://doi.org/10.1007/s10237-019-01258-7

ORIGINAL PAPER

Predictive prosthetic socket design: part 2—generating person-specific candidate designs using multi-objective genetic algorithms

J. W. Steer[1] · P. A. Grudniewski[1] · M. Browne[1] · P. R. Worsley[2] · A. J. Sobey[1] · A. S. Dickinson[1]

Received: 28 May 2019 / Accepted: 8 November 2019

Abstract

In post-amputation rehabilitation, a common goal is to return to ambulation using a prosthetic limb, suspended by a customised socket. Prosthetic socket design aims to optimise load transfer between the residual limb and mechanical limb, by customisation to the user. This is a time-consuming process, and with the increase in people requiring these prosthetics, it is vital that these personalised devices can be produced rapidly while maintaining excellent fit, to maximise function and comfort. Prosthetic sockets are designed by capturing the residual limb's shape and applying a series of geometrical modifications, called rectifications. Expert knowledge is required to achieve a comfortable fit in this iterative process. A variety of rectifications can be made, grouped into established strategies [e.g. in transtibial sockets: patellar tendon bearing (PTB) and total surface bearing (TSB)], creating a complex design space. To date, adoption of advanced engineering solutions to support fitting has been limited. One method is numerical optimisation, which allows the designer a number of likely candidate solutions to start the design process. Numerical optimisation is commonly used in many industries but not prevalent in the design of prosthetic sockets. This paper therefore presents candidate shape optimisation methods which might benefit the prosthetist and the limb user, by blending the state of the art from prosthetic mechanical design, surrogate modelling and evolutionary computation. The result of the analysis is a series of prosthetic socket designs that preferentially load and unload the pressure tolerant and intolerant regions of the residual limb. This spectrum is bounded by the general forms of the PTB and TSB designs, with a series of variations in between that represent a compromise between these accepted approaches. This results in a difference in pressure of up to 31 kPa over the fibula head and 14 kPa over the residuum tip. The presented methods would allow a trained prosthetist to rapidly assess these likely candidates and then to make final detailed modifications and fine-tuning. Importantly, insights gained about the design should be seen as a compliment, not a replacement, for the prosthetist's skill and experience. We propose instead that this method might reduce the time spent on the early stages of socket design and allow prosthetists to focus on the most skilled and creative tasks of fine-tuning the design, in face-to-face consultation with their client.

Keywords FEA · Amputation · Residual limb · Optimisation

✉ A. S. Dickinson
alex.dickinson@soton.ac.uk

J. W. Steer
joshua.steer@soton.ac.uk

P. A. Grudniewski
pag1c15@soton.ac.uk

M. Browne
doctor@soton.ac.uk

P. R. Worsley
p.r.worsley@soton.ac.uk

A. J. Sobey
ajs502@soton.ac.uk

[1] Faculty of Engineering and Physical Sciences, University of Southampton, Southampton, UK

[2] Faculty of Environmental and Life Sciences, University of Southampton, Southampton, UK

1 Requirement of automation in design of prosthetics

Approximately 40 million people globally require access to prosthetic or orthotic services (Eklund and Sexton 2017). Prosthesis–human interface design aims to maximise comfort and functionality for people with amputations, towards ambulatory rehabilitation. This is commonly provided through a prosthetic socket, which is designed through geometric modifications to the captured shape of the residual limb, known as rectifications, to create a desired pattern of load transfer. This is currently an iterative process performed by a highly skilled prosthetist, who manages the residuum's changing size, shape, soft tissue healing and biomechanical adaptation. Indeed, due to these factors, the development of a definitive socket takes a considerable period of time. Prosthetic limb users require lifelong access to prosthetics services, and in the UK the annual cost of prosthesis provision and care is over £2800 per patient (Kerr et al. 2014). This includes the replacement of prosthetic limb components typically every 2 to 5 years. Skilled prosthetists take many years to train to a high standard, and often prosthetic users develop relationships with their preferred clinician to maintain socket comfort. However, there are limited numbers of these highly skilled individuals and practice efficiencies are required in the face of growing clinical demand. Researchers have considered mechanisms for employing quantitative prediction in the socket design process (Goh et al. 2005; Colombo et al. 2013), although at present these work to a single design target for a single individual and have not entered conventional clinical use.

In Part 1 of this study (Steer et al. 2019), a kriging-based surrogate model was generated for a parametric FE model of a population-based transtibial residual limb and accompanying total surface bearing (TSB) socket design. This enabled the prediction of biomechanical relationships between the residual limb morphology and prosthetic socket design, while reducing the computational cost of each new prediction by six orders of magnitudes (1.6 ms vs. 30 min). The simplified total surface bearing socket design was defined parametrically from the limb's neutral shape, by reducing the cross-sectional area along its length with three points at the proximal, mid and distal regions of the socket. However, within a clinical setting, the socket design process is substantially more nuanced. There are several different design philosophies, all with different intended residual limb load transfer mechanisms. The classic patella tendon bearing (PTB) socket design was developed in 1957 and is still commonly used in-clinic today (Radcliffe 1962). This socket design aims to apply pressure over load-tolerant areas of the limb such as the patella tendon, and off-load pressure-sensitive regions such as the anterior tibia, fibula head and residuum tip. Other sockets include the Kondylen-Bettung Münster (KBM) which provides supracondylar suspension in addition to features consistent with the patella tendon bearing design (Kuhn 1966), and hydrostatic sockets (Murdoch 1964) such as the PCAST system (Lee et al. 2000; Goh et al. 2003; Goh et al. 2004; Laing et al. 2017; Laing et al. 2018) which uses a pressurised fluid as a medium to form the shape of the socket with the aim of achieving minimal residuum surface pressure gradients with less manual intervention. More recently, total surface bearing sockets, which were proposed in 1987, are used to generate near-total contact in between the residual limb and the socket (Staats and Lundt 1987). In theory, this should maximise the contact area between the residual limb and prosthetic socket and the uniformity of pressure across the surface of the residual limb, thereby minimising potentially harmful pressure gradients (Hachisuka et al. 1998).

Despite the fundamental differences in the load distribution between these socket designs, they can potentially all deliver satisfactory outcomes for prosthesis users. There is substantial research into quantifying the biomechanical differences between these socket designs, which is comprehensively reviewed by Safari and Meier in (2015). Their systematic review concluded that 'the included studies only had low to moderate methodological rigour', thus demonstrating the difficulties in defining biomechanical guidelines for the highly dynamic environment of the residual limb—prosthetic socket system, or selection of the preferred socket type for a particular individual or situation. One possible reason for the difficulty in establishing the definitive guidelines of these different socket types is that they are defined primarily by design intent, rather than quantitative rules. This effect has been illustrated for a simple total surface bearing socket using parametric FEA (Steer et al. 2019), and it is almost certain the within-type variability would be increased for more complex designs. We propose that there is a large potential to enhance the evidence base behind this clinical challenge, allowing prosthetists to develop, critique and share their own expertise and decision-making, making more effective use of their valuable design and consultation time. A key and relatively unexplored possibility is to apply automated search algorithms to explore designs prior to optimisation for the individual.

Optimisation algorithms are common in many areas of engineering. They are used as concept design methods, providing an initial product which engineers can use as a starting point and to increase the proportion of their time spent on creatively solving complex problems. In addition, they provide a visualisation for how these changes will affect the final product's performance, allowing a greater understanding of the design space which can be put to use in the

Table 1 Parameters of the four cases extracted from the parametric residual limb model

Virtual person	Residuum length, v_1	Residuum profile, v_2	Tibia length, v_3	Soft tissue initial modulus, v_4
A	-0.8σ (Short)	-0.8σ (Bulbous)	+ 20% (Long)	40 kPa (Soft)
B	-0.8σ (Short)	$+0.8\sigma$ (Conical)	− 5% (Short)	50 kPa (Stiff)
C	$+0.8\sigma$ (Long)	-0.8σ (Bulbous)	+ 20% (Long)	40 kPa (Soft)
D	$\mp 0.8\sigma$ (Long)	$+0.8\sigma$ (Conical)	− 5% (Short)	50 kPa (Stiff)

Soft tissue initial modulus corresponds to the initial stiffness of the applied neo-Hookean hyperelastic material model

more detailed stages of the process. A choice of potential candidate designs can be provided to the decision maker, which weight the objectives differently, for example putting more load on one region of a structure and removing it from another, and therefore give a range of performances. This requires algorithms capable of multi-objective optimisation that provide a rapid convergence on the global optimum while retaining a high diversity of the search, to ensure that the entire search space is investigated and that the focus is not upon local optima. Many methodologies have been developed, and state-of-the-art research focuses on improvements in diversity or convergence.

This paper employs optimisation algorithms to generate personalised 'candidate' prosthetic socket designs for the first time. This is applied to the transtibial case, which is the most common major lower limb amputation and where most clinical success has been achieved with associated CAD/CAM socket design and fabrication tools. Different design problems require different optimisation processes. The aim is therefore to determine appropriate methods for the automated application of candidate socket rectifications, collating the state of the art in biomechanical analysis of prosthesis–limb interfaces, surrogate modelling and optimisation. Genetic Algorithms are chosen due to their ability to effectively search large and complex design spaces, which is the problem presented by the continually variable distribution of possible limb–socket shape rectifications. These methods rely on thousands of function calls, and using FE models would not be feasible beyond single cases due to the time required for each simulation. However, by leveraging the speed increases of the surrogate model (Steer et al. 2019), it becomes technically feasible to perform automated socket optimisation based upon structural analysis of the limb–prosthesis system. This provides the motivation for the current study, to perform a first-of-kind, subject-specific, multi-objective design optimisation of the prosthetic socket using the previously reported surrogate model. The result will be a series of personalised 'candidate' transtibial prosthetic socket designs, to which the prosthetist would add local modifications based upon their knowledge and conventional patient consultation. Finally, equipped with these results, a prosthetist would then further refine these concepts to achieve a desired pattern of prosthesis–limb load transfer, by using these designs to augment their experience-based decision-making.

2 Optimisation of transtibial prosthetic sockets

2.1 Population-based surrogate model

A detailed description of the population-based surrogate model is found in Part 1 of this paper (Steer et al. 2019). In short, a generic residual limb was generated by producing a volume mesh from an MRI scan and imposing radial basis function mesh morphing to apply parametric variation in residuum length and profile (conical to bulbous) obtained from principal modes of variation from a population of 3D surface scans. These were varied by $\pm 1\ \sigma$ (standard deviation) about the mean length and profile in the statistical shape model (SSM). Furthermore, internal parametric variation of the relative tibia length (i.e. distal soft tissue coverage) from − 15% to + 30% of the tibia length from the MRI scan, and soft tissue material properties between stiff, flaccid muscle and contracted muscle (Palevski et al. 2006; Portnoy et al. 2009; Hoyt et al. 2008) were applied. The soft tissue was assigned a neo-Hookean material to capture the nonlinear behaviour of the soft tissue. The present surrogate model implementation investigates the effects of socket design variation for four synthetic 'virtual' people sequentially by selecting exemplar values for the model's residuum variability parameters (Table 1, Fig. 1). These cases were chosen as being close to the models' population extremes while remaining within the bounding box of the sampling plan, to avoid extrapolating beyond the surrogate. These meshes were imported into the finite element solver (ABAQUS 6.14, Dassault Systèmes, Vèlizy-Villacoublay, France). The socket was donned under displacement control and loaded uniaxially to 400 N to simulate a two-leg stance. The resultant pressure and soft tissue strain outputs from 75 simulations were used to construct a kriging surrogate model for each virtual person, enabling a function call to be made in ~ 2 ms.

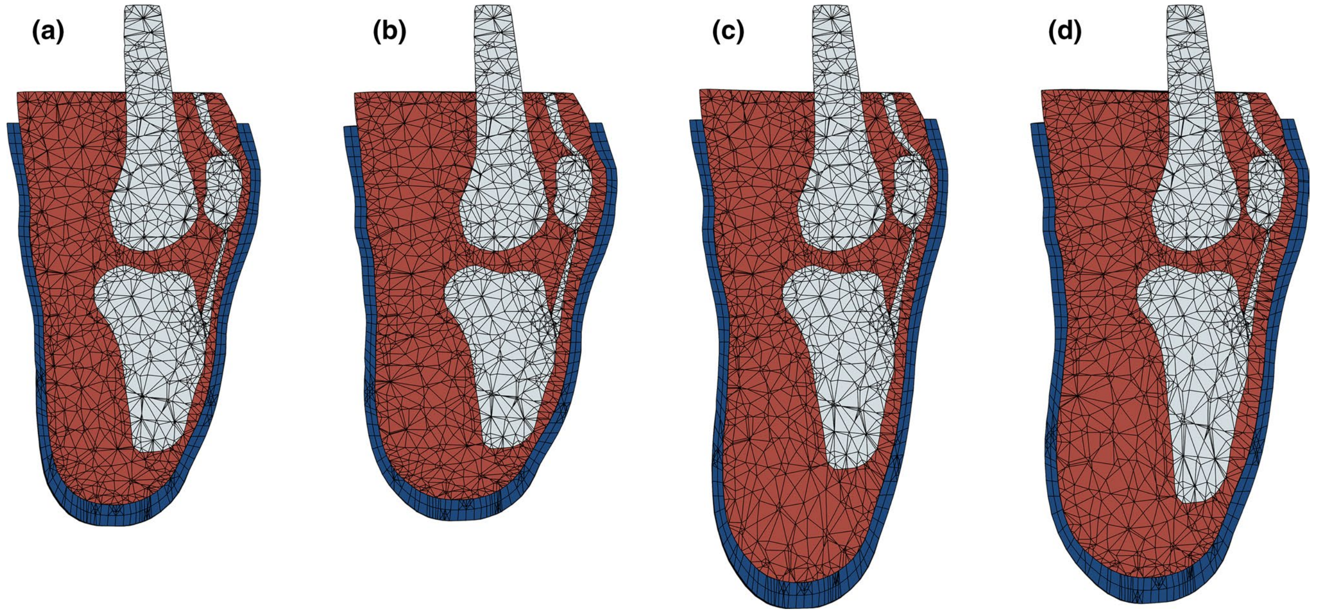


Fig. 1 Sagittal sections through equivalent residuum FE models for the four virtual people. Blue indicates the liner, red the soft tissue, and grey the bones. The prosthetic socket is not shown

2.2 Parametric socket design

In the preceding work (Steer et al. 2019), a simplified, three-parameter total surface bearing socket design was used. This model enabled control of the socket press fit by reducing its cross-sectional area through a B-spline function with proximal, mid and distal control points. The three variables were constrained between − 1% and 3% by cross-sectional area reduction. The present study's socket design was extended to include the localised rectifications observed in patella tendon bearing sockets. Control points were generated over the fibula head, patella tendon and either side of the tibial crest (Fig. 2). These localised rectifications were applied using the same radial basis function mesh morphing algorithm detailed above, by radially displacing the control points between 0 and 6 mm.

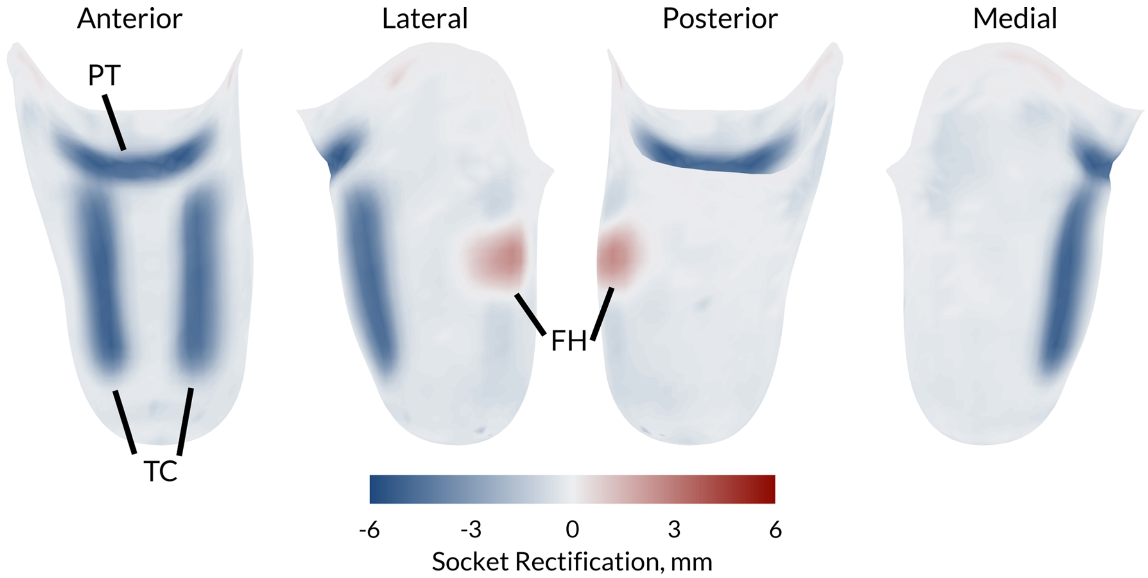


Fig. 2 Rectification maps of the patella tendon bearing socket design at the maximum values of patella tendon bar (PTB), fibula head (FH) relief and tibial crest (TC) rectifications. The figure demonstrates the resulting socket shape change once the control nodes have been displaced and explains the convention directions of each rectification type (FH vs. PT and TC)

2.3 Optimisation via genetic algorithms

Genetic algorithms (GA) are population-based multi-objective solvers inspired by the principles of Darwinian evolution (Goldberg and Holland 1988). In a simple Genetic Algorithm, a set of potential solutions, called individuals, reproduce via an evolutionary-like process. Each individual contains set of decision variables, called chromosomes, with an initial population with variables that are usually assigned randomly. The fitness of each individual can be evaluated according to some predefined objectives. After this step individuals are then chosen for reproduction, and according to the principles of natural selection, the fitter individuals have significantly higher chances of reproducing than those with a low fitness. Offspring are generated from the selected parents using crossover and mutation processes. During crossover, the chromosomes of the offspring are produced by mixing the genes of the parents, providing convergence and diversity. In the mutation step, the offspring's genes have a small chance to be randomly modified, improving the population diversity. Finally, the old population becomes extinct and is replaced by the new generation, with the new generation being fitter, on average, than the parent generation. This process continues until the predefined termination condition is met, often specified as a maximum number of objective functions calls or total calculation time.

Many competing genetic algorithms have been developed, each introducing novel mechanisms to increase the convergence rates and diversity of the search. In the current state of the art of Gas, there is particular emphasis on specialist solvers. According to the 'no free lunch' theorem (Wolpert and Macready 1997), a specialist solver exhibits high performance on a narrow set of problems but its performance will rapidly decline when outside of this set. Therefore, a suitable methodology has to be selected with respect to the particular problem's characteristics in order to avoid poor performance. The optimisation problem characteristics and their difficulty are defined by the topology of the search and objective spaces, number of local optima and the applied constraints. If the problem characteristics are not known, then more than one GA methodology should be applied as their performance can differ drastically. This will provide more reliable results and allow an evaluation of the problem's difficulty and its dominant characteristics (Wang et al. 2018). In the case presented in this paper, no knowledge about the characteristic of the problem is available a priori, except that no constraints are used. However, this is not sufficient to choose a single properly adjusted optimiser. Therefore, five different Genetic Algorithms are compared: NSGA-II as the most commonly utilised Genetic Algorithm which retains a high diversity of search and has had much success in the applied literature (Deb et al. 2002); MOEA/D as the most proficient algorithm for unconstrained problems (Zhang and Li 2007); MTS as an aggregation of a Genetic Algorithm and a local-search method which provides improved convergence (Tseng and Chen 2007); cMLSGA and HEIA as the general-type GAs that exhibit high performance across wide range of problem types and therefore higher robustness (Lin et al. 2016; Grudniewski and Sobey 2019). HEIA is more dominant in scenarios where convergence is more important and cMLSGA provides a higher diversity of search. The detailed principles of working and parameter settings of each methodology can be found in their respective publications.[1] All the tests were performed over 30 separate runs, with 50,000 fitness function evaluations as a termination criterion. Multiple runs must be performed in order to assure the robustness of the method and the best likelihood of identifying the true Pareto Front. Different population sizes were tested and 600 individuals were utilised as the best for NSGA-II, MTS, MOEA/D and HEIA, while cMLSGA utilised 1800 as it requires significantly higher population sizes (Grudniewski and Sobey 2018).

The socket design process presented in our prior work (Steer et al. 2019) can be framed as a formal engineering design optimisation problem. In this case the individual socket rectifications function as design parameters across a multi-dimensional input space, and the resultant pressure and soft tissue strain fields are formulated as the objective functions. The locations across the limb for the objective functions were selected because they are known to be load-intolerant (Radcliffe 1962). It was predicted that the introduction of a peripheral press fit around the main body of the residuum will allow load transfer through the longitudinal shear forces and thus reduce the residuum tip pressure, at the expense of pressure concentrations over the bony prominences of the tibial tuberosity and fibula head. Four state variables were defined: the pressure over the residuum tip (f_1), the tibial tuberosity (f_2), the fibula head (f_3) and the soft tissue strain around the distal tibia (f_4). These model outputs can be described as competing fitness functions, indicating proximal and distal loading, defined as $FF1 = f_1 + f_4$ and $FF2 = f_2 + f_3$. These were evaluated using the surrogate model developed previously (Steer et al. 2019) for the four synthetic people defined in Table 1.

One of the issues with multi-objective optimisation is the comparison of the results obtained by different methods. The visual comparison is limited, only providing useful information when the performance of two solvers differs drastically. Otherwise, the points will overlap making objective comparison near impossible. Therefore, multiple quality indicators have been developed (Li and Yao 2018). Most of them are able to indicate the performance in both convergence and

[1] Source codes for all methodologies can be found at: https://bitbucket.org/Pag1c18/cmlsga.

diversity of the solutions. However, each of them has certain drawbacks or biases and it is common practice to utilise more than one indicator (Li and Yao 2018). In this paper the inverted generational distance (IGD) and hypervolume (HV) were chosen as indicators. IGD measures the average Euclidean distance between each point in a real Pareto Optimal Front (POF) and the closest solution in the obtained set. Lower values indicate better convergence and uniformity of the points and are calculated according to Eq. 1:

$$\mathrm{IGD}(A, P^*) = \frac{\sum_{\nu \in P^*} \mathrm{d}(\nu, A)}{|P^*|}, \tag{1}$$

where P^* is a set of uniformly distributed points along the true PF, A is the approximate set to the POF, which is being evaluated and $\mathrm{d}(\nu, A)$ is the minimum Euclidean distance between the point ν and points in A.

However, this IGD shows poor performance in determining the diversity of a population when the Pareto Front population is small. HV is calculated as the volume of an objective space between a predefined reference point and the obtained solutions where higher values are preferred (Li and Yao 2018). This indicator has a stronger focus on the diversity and boundary points. Most indicators require a predefined reference Pareto Optimal Front that illustrates the ideal set of solutions. However, in cases where the optimal answer is not known the utilisation of these indicators can be problematic. A solution is to calculate a reference Pareto Optimal Front using the non-dominated selection of Pareto Optimal Fronts achieved by every algorithm when performing multiple runs, or performing a few test runs with significantly higher numbers of iterations than that utilised for comparison (Wang et al. 2018). In this paper both are applied, and a combined non-dominated front obtained by brute force from all 6 Genetic Algorithms after 300,000 fitness function evaluations was used to determine the success of the algorithm.

3 Results

A single Genetic Algorithm run with a maximum of 50,000 function calls was computed in approximately 30 min, where Fig. 3a shows the individuals evaluated over this lifetime and the final Pareto Front. Comparing the different genetic algorithms, it was observed that the shape of the Pareto Optimal Fronts remains consistent. Therefore, visual comparison only shows that all of the methodologies exhibit similar performance and it is not possible to unanimously choose the best methodology (Fig. 3b). The bias between Fitness Functions FF1 and FF2 along the normalised Pareto Optimal Front is visualised in Fig. 3c. The reason the no-bias point is not in the middle of the front is due to the longer 'tail' when minimisation is biased towards FF2 (minimising proximal bony prominence loading), compared with bias towards FF1 (minimising distal tip loading). It was also observed that while the minima of FF1 for all individuals plateaued at just below 20 (unitless), the minima of FF2 were different for all of the virtual people (Fig. 3d). The minima of the short, conical limb of person B and the long, bulbous limb of person C plateaued at FF2 = 55 kPa and 40 kPa, respectively.

From visualisation of the sockets at either end of the Pareto Front, as well as the neutral case, consistencies in design emerged across the four people (Fig. 4). When the optimiser was biased towards FF1 (minimising tip loading), designs of higher press fit which off-loaded the residuum tip emerged from the Genetic Algorithm. For person B (Fig. 4b) and person D (Fig. 4d), pressure hotspots were generated where there was little soft tissue coverage over the proximal bony prominences. When the model was biased towards FF2, sockets with higher fibula head relief evolved in order to off-load over this region.

Trends in the designs can be observed between the competing fitness functions by visualising how the optimal socket designs change along the Pareto Optimal Front (Fig. 5). Across all virtual people, the patella tendon bar variable converged at the constraint maximum of 6 mm for almost all of the points along the Pareto Optimal Front. The exception was in Person D, with the longest and thinnest residual limb. When the optimiser was biased towards FF1, a few designs evolved with the patella tendon bar rectification at the 0 mm lower limit. This was offset by removal of the fibula head relief to ensure that the pressure over the residuum tip is still minimised. A clear trend for all virtual people was in the mid reduction in the socket, where the press fit decreases along the Pareto Optimal Front from FF1 (with the aim of minimising the distal loading) to FF2 (with the aim of minimising the proximal loading). By reducing the press fit, the pressure over the bony prominences and the peripheral shear both decreased, resulting in an increase in distal tip pressure and soft tissue strain.

The performance of different methodologies was evaluated using the proposed indicators, IGD and HV and presented in the form of rankings with average values and standard deviations (Table 3). In this case the algorithms all performed in the same manner for IGD and HV. HEIA and cMLSGA were the best performing algorithms and MOEA/D and MTS performed the worst. However, the relatively similar performances of all five algorithms indicate that the complexity of the presented cases is low. The final Pareto Front was continuous and there were no constraints, which led to convergence-dominated HEIA having the best performance, over cMLSGA and NSGA-II. MOEA/D and MTS perform less well. However, this may be due to a lack of hyperparameter tuning to the particular problem. These

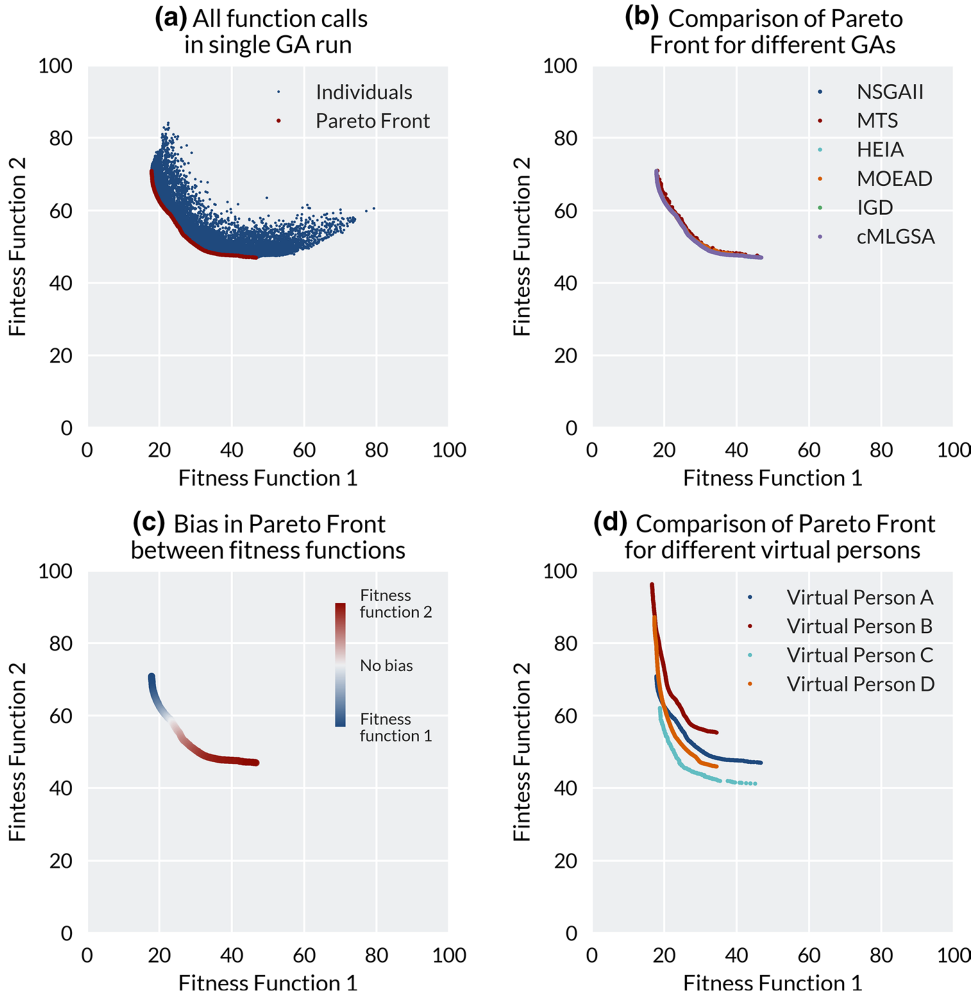


Fig. 3 Analysis of the Pareto fronts from the multi-objective optimisation. **a** Individuals from a single run of the HEIA optimisation for Person A, with all individuals plotted in blue and Pareto front in red. **b** Comparison of the generated PFs for the six different GAs tested on Person A. **c** Bias along the Pareto front between the two fitness functions, with 'no bias' defined as the minimum distance from the origin to the normalised Pareto Optimal Front, with blue indicating bias towards FF1 and red towards FF2. **d** Comparison of the Pareto Fronts for the four different People when using HEIA

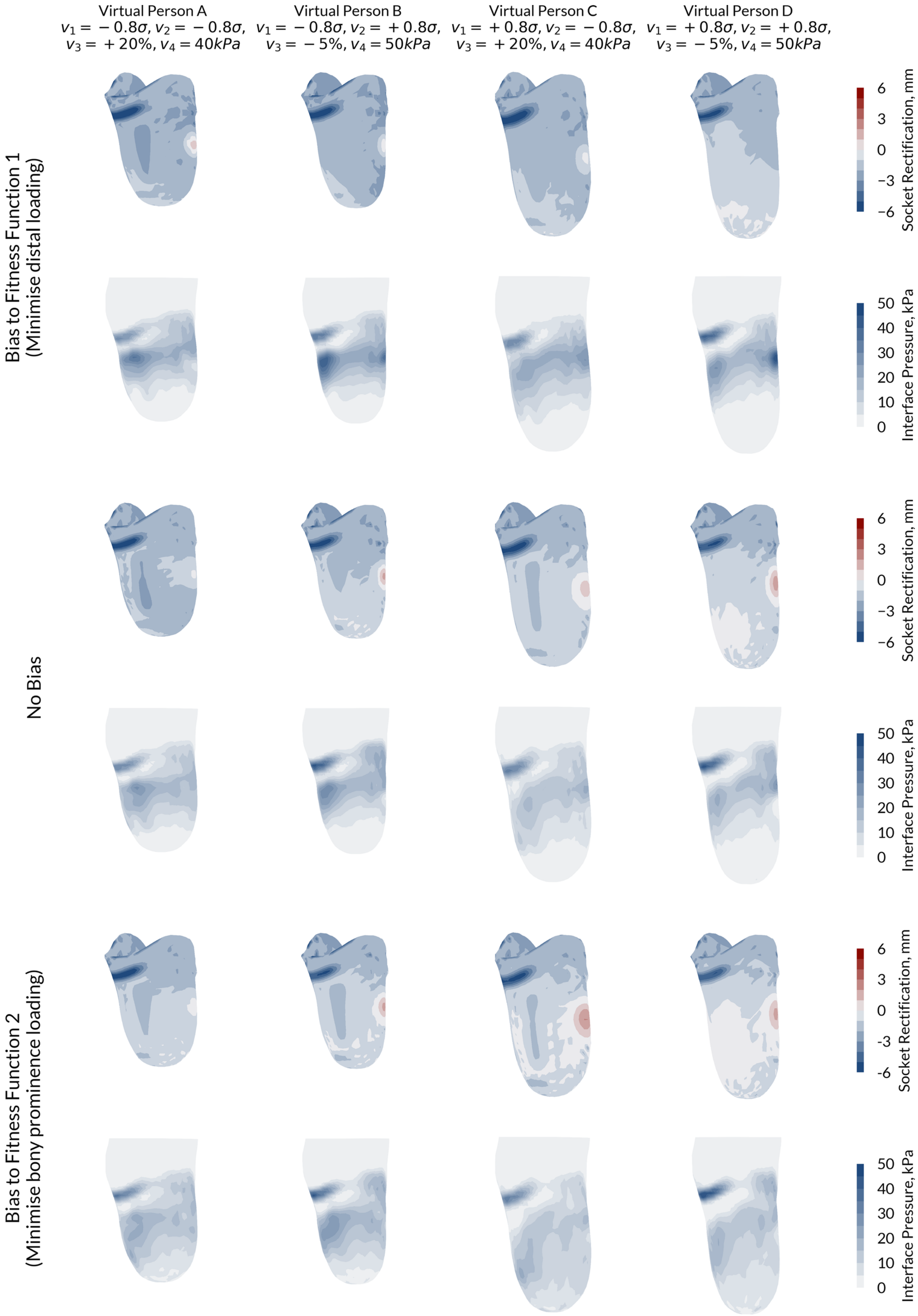
two algorithms are dependent on a number of parameters which must be optimised for each problem, and in the present work the authors used default values described in the algorithms' original papers. The MOEA/D and MTS algorithms may perform better once tuned, now that a priori knowledge has been developed, but the present results indicate the caution with which these algorithms should be used.

In order to better understand the complexity of the problem and to check whether the best possible set of solutions has been found, a set of 5 runs with 300,000 total iterations was conducted on each virtual person, utilising HEIA. In this case hardly any difference was observed between 50,000 and 300,000 iterations. Figure 6a shows some very slightly higher uniformity and diversity of the points in the high FF1 bias region with 300,000 iterations. When comparing the performance over the number of generations in Fig. 6b, virtually no improvement in performance can be seen after 50,000 generations and the highest performance gain occurred before 25,000 iterations. The low possible performance increase beyond 50,000 iterations in this case would not justify conducting optimisation of this problem with higher values, unless the virtual person is suspected to benefit from an extreme reduction in pressure over the residuum tip and the soft tissue strain around the distal tibia (FF1 bias).

4 Discussion

This study aimed to explore a range of potential concepts for transtibial socket design using FE modelling, surrogate modelling and GA-based optimisation techniques, to provide a quantitatively informed starting point for the prosthetist when designing a bespoke prosthetic socket.

Exploring the parametric socket design space demonstrates that biomechanical objective functions are in competition and illustrates the challenges associated with defining the 'best' socket design solution. As explored in our previous work (Steer et al. 2019), by increasing the socket press fit, particularly in the mid-section, an increase in longitudinal shear around the main body of the residual



Virtual Person A
$v_1 = -0.8\sigma, v_2 = -0.8\sigma,$
$v_3 = +20\%, v_4 = 40kPa$
Virtual Person B
$v_1 = -0.8\sigma, v_2 = +0.8\sigma,$
$v_3 = -5\%, v_4 = 50kPa$
Virtual Person C
$v_1 = +0.8\sigma, v_2 = -0.8\sigma,$
$v_3 = +20\%, v_4 = 40kPa$
Virtual Person D
$v_1 = +0.8\sigma, v_2 = +0.8\sigma,$
$v_3 = -5\%, v_4 = 50kPa$
Bias to Fitness Function 1
(Minimise distal loading)
No Bias
Bias to Fitness Function 2
(Minimise bony prominence loading)
Socket Rectification, mm
6
3
0
-3
-6
Interface Pressure, kPa
50
40
30
20
10
0

◂**Fig. 4** Optimal socket designs and corresponding predicted pressure maps for the four different virtual people at the two ends of the POF, i.e. biased towards minimising distal tip loading (top) and minimising proximal bony prominence loading (bottom), and the design with no bias (centre)

limb was predicted. This resulted in a pressure reduction at the residuum tip coupled to a reduction in the internal strain around the distal tibia. By oversizing the socket (i.e. negative press fit), these peripheral shear forces were not

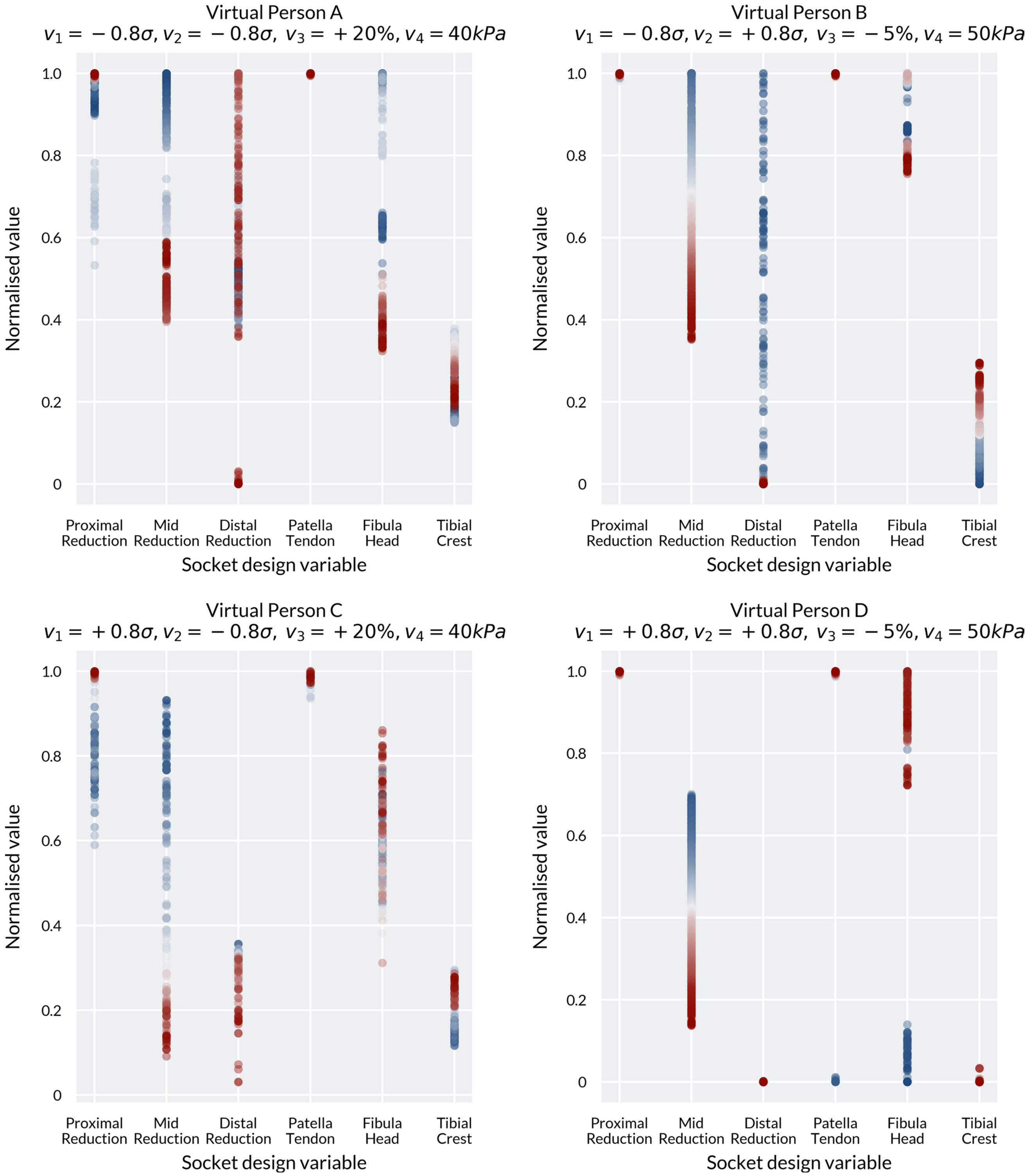


Fig. 5 Comparison of how the socket design variables (see Table 2) changed between the four cases along the Pareto Optimal Front. Blue denotes a bias towards FF1 (distal loading), while red denotes bias towards FF2 (proximal loading)

Table 2 Parameters and limits of the parametric socket design

Socket rectification variable name	Lower bound	Upper bound
Proximal press fit	−2%	+ 6%
Mid press fit	−2%	+6%
Distal press fit	−2%	+6%
Patella tendon bar	0 mm	6 mm
Fibula head relief	0 mm	6 mm
Tibial crest	0 mm	6 mm

generated, thereby increasing the distal pressure and soft tissue strain. These represent the competing fitness functions inherent in prosthetic socket design.

Introduction of the patella tendon bar and tibial crest rectifications provided an alternative method of off-loading the residuum tip beyond a uniform press fit. Fibula head relief is predicted to be effective in reducing the high pressure that was observed over this bony prominence for the total surface bearing socket designs, thus reiterating the importance of localised shape change beyond applying gross scaling to the limb shape (Goh et al. 2003).

The sockets that emerged from the Genetic Algorithm exhibited features of both total surface bearing (TSB) and patella tendon bearing (PTB) manual socket design philosophies. One consistent feature along the Pareto Front for all virtual people was the patella tendon bar rectification variable, which saturated at its maximum limit. This is because no optimisation cost was associated with applying pressure over this region, which the Genetic Algorithm exploited to off-load the high-cost residuum tip region. This effect is observed clinically for the patella tendon bearing socket where prosthetists produce a marked rectification over the patella tendon to leverage its load bearing capacity. Although load tolerant, there clearly would be a load threshold for injury at the patella tendon, so with enhanced spatial data of load tolerance across these key

Table 3 Ranking of different genetic algorithms using HV and IGD as the performance indicators

Rank	1	2	3	4	5
IGD					
Algorithm	HEIA*	cMLSGA*	NSGA-II*	MOEA/D*	MTS
Average	0.029349	0.057565	0.1384	0.281601	0.590279
(S.D.)	0.001741	0.001553	0.139218	0.158822	0.049102
HV					
Algorithm	HEIA*	cMLSGA	NSGA-II*	MOEA/D	MTS
Average	0.174846	0.174475	0.174461	0.174094	0.168158
(S.D.)	0.000027	0.000039	0.000288	0.000214	0.000464

*indicates if the results are significantly different to the next lowest rank, using the Wilcoxon's rank sum with a 0.05 confidence

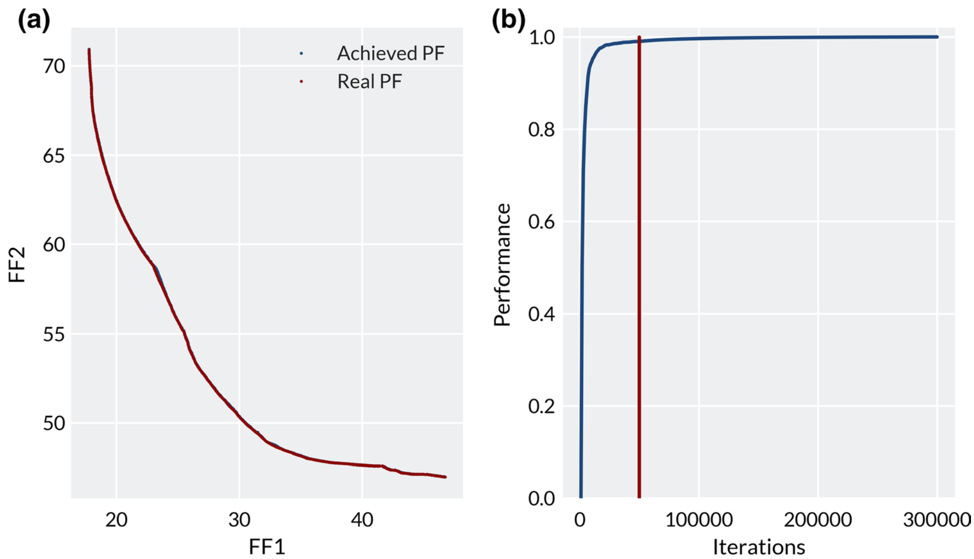


Fig. 6 **a** The comparison of Pareto Fronts from Virtual Person 1, achieved using HEIA over 50,000 iterations ('achieved') and 300,000 iterations ('real'). **b** The performance of HEIA over 300,000 iterations on Person 1. 0 is the starting population, and 1 is the best attainable set of solutions, based on the IGD values, and the red line indicates the number of function calls utilised in this study

residuum locations (Bramley et al. 2019) an additional constraint of maximal patellar tendon pressure could be included in the optimisation problem.

Along the Pareto Front of the best solutions, trends were predicted as the bias of the optimised socket varied between the two fitness functions. When fitness function 1 was dominant and the Genetic Algorithm aimed to minimise pressure and soft tissue strain at the residuum tip, sockets with high levels of mid-height press fit emerged from the model. Conversely, when fitness function 2 was dominant and the Genetic Algorithm aimed to minimise pressure over the proximal bony prominences, the global press fit was reduced and local relief over the fibula head was increased. The sockets which minimised residuum tip pressure (FF1-biased) exhibited characteristics associated with a total surface bearing socket design, while the patella tendon bearing rectifications were dominant when minimising pressure over bony prominences (FF2-biased). While it is difficult to validate these findings from the current literature, a systematic review of transtibial prosthetic socket designs by Safari and Meier concluded that TSB sockets exhibited improved weight-bearing, greater suspension and reduced pistoning, which may be, in part, due to the increased peripheral shear from the TSB socket (Safari and Meier 2015). However, extensive experimental data collection is required to validate such a hypothesis.

Differences in the Pareto Front were observed between the virtual people. While the minimum value of fitness function 1 was consistent across the cases at just below 20, the minima of fitness function 2 varied substantially. This result was to be expected based upon the results of the population model where residuum morphology, in particular the residuum profile, had a substantial effect on the pressure over the bony prominences.

A range of genetic algorithms proved effective in performing multi-objective design optimisation of the socket by handling the complex task of simulating the interplay between rectifications on the competing objective functions of the residual limb across the presented design space. In this case the performance of all methodologies was comparable and it could be concluded that the utilisation of several GAs was unnecessary. However, in this case the problem is rather simple to optimise, as 50,000 fitness function calls are sufficient to provide good approximation of the best Pareto Front, and in some cases 20,000 was adequate. The problem has continuous search and objective spaces which further indicates its simplicity (Woldesenbet et al. 2009). However, as the importance of utilising multiple methodologies has been shown by previous researchers (Wang et al. 2018), it is strongly advised here to follow this procedure as good practice until the design space for transtibial prosthetic sockets is better understood. In the future, as more variables and objectives are added to the search space, it is expected that the topology of the design space will change and therefore provide an increasing challenge to resolve the optimal points and require review of the required GA parameters and convergence limits.

The presented multi-objective design optimisation provides an early demonstration of how the speed increases achieved by surrogate techniques enable the socket design process to be framed as an engineering design problem. There are several potential improvements that could be implemented within this process. One such approach may be a dual-level solver where the solver starts with no data, runs a full simulation on a limited population of designs, creates a surrogate from these designs and evaluates the fitness of a substantially larger group across the surrogate. The elite individuals, the fittest individuals in the population which are often defined as the top 10%, would be retained for the next generation and the process repeated. This approach would enable the GA to ignore regions which are clearly sub-optimal, and instead prioritise expensive FE analyses in regions where the minima of the fitness function is more likely to be found. As an alternative approach, to prevent overfitting, the surrogate might be used to generate initial generations, and more expensive FE analyses used at the end to select a preferred design from the options along the Pareto Front.

5 Limitations

User satisfaction with the socket is ultimately a subjective measure dependent upon a range of human factors such as comfort, pain thresholds and proprioception arising from a firm, functional prosthesis-skeletal coupling. This means that the predictions of pressure, shear stress and soft tissue strain are not directly related to comfort (Mak et al. 2001). Furthermore, the model would not account for local tissue sensitivities associated with neuromata and soft tissue injuries which could only be identified in limb assessment by the prosthetist. This process might therefore be enhanced by surveying functional and user-reported outcome measures across a population of socket designs.

No direct experimental validation of the underlying relationships between socket design and load transfer predicted by the model in this study has been performed, and such validation evidence must be obtained prior to any clinical evaluation. Pressure and shear sensors (Laszczak et al. 2015) and laboratory-based residuum-socket simulators (McGrath et al. 2017) measure the interaction between the residual limb and socket and could be used to reinforce the findings of this study. As the model uses invented residual limb shapes with thousands of socket designs, it is clearly infeasible to perform experimental validation upon any more than a limited subset of data points in this model. However, in future, a limited number of key socket designs should be tested to validate its conclusions. Furthermore, the population-based surrogate model only characterises a simplified representation of the variability which exists across the population. As discussed previously (Steer et al. 2019), a practical application of these tools requires further data to construct the surrogate model, for example variation in femoral or patella geometry, bone and liner material properties, as well as dynamic load cases. Some confidence is provided by corroboration with literature reports of pressure predictions across the limb between 30 and 100 kPa during gait for TSB sockets (Al-Fakih et al. 2013; Beil et al. 2002; Beil and Street 2004; Dumbleton et al. 2009) and 25–320 kPa for PTB (Dumbleton et al. 2009; Zhang et al. 1998; Dou et al. 2006), which is consistent with the range of predicted pressures for the FF1-biased sockets in this study.

As the study is an initial investigation into the methods and potential it forms the basis for further investigations that provide a more complex design. In increasing this complexity a number of elements will change. First, the Kriging model itself will become more complex providing some challenges in the use of this model which must be investigated. Second, the design space will change and this will provide a different set of optimisation challenges. In both instances the methods used will need to be evaluated carefully. In the case of more complex design spaces, other surrogates might become more appropriate, such as Deep Reinforcement learners. These are subject to a disadvantage of requiring more input data. For the optimisation, it is likely that the space will become more discontinuous (Sobey et al. 2019), similar to other more complex applications, and this will require algorithms with stronger diversity (Wang et al. 2018): NSGA-II and cMLSGA. There is also likely to be a greater separation between specialist, which will have even further reduced performance compared to the general solvers: NSGA-II, cMLSGA and HEIA.

6 Clinical applications

Attempting to use simulations to inform clinical decision-making requires extreme caution, especially when applied to devices which depend upon personalised design to ensure comfort and functional efficacy, as comfort and proprioception are difficult to quantify. Crucially, in prosthetic limb design we would argue that these techniques should not be used in isolation, or substituted for human-facing clinical practice. The expert prosthetist must retain control over socket design, and the presented optimisation approach could be used to provide a 'first-guess' rectification map. The prosthetist would then modify this candidate design according to their own clinical reasoning which combines palpation, user feedback and re-evaluation. Other technologies such as real-time pressure measurements and predictions from the previously reported PCA-kriging model (Steer et al. 2019) incorporated with their skill and experience could provide a technology-enhanced limb assessment. This approach will help the community to test the key translational research question in this field: can the clinical application of FEA support the prosthetist's evidence base and enable delivery of comfortable, highly functional prosthetic limbs to users in a more timely and efficient manner?

7 Conclusion

This paper provides a first assessment of the use of multi-objective optimisation in the design of prosthetic sockets. The experiential judgement and skill-based process of prosthetic socket design is framed as a multi-objective engineering design problem. This is achieved by developing parametric models of the residual limb informed by statistical shape modelling techniques and the prosthetic socket incorporating both total surface bearing and patella tendon bearing rectifications, which allow the underlying biomechanical relationships between the residual limb and prosthetic socket to be predicted. In line with experimental data to allow detailed biomechanical validation, the developed methods show substantial potential to be used as part of a more informed socket design process, and provide clinicians with support for selecting from the range of candidate design approaches. The resulting designs correspond with the general forms of the two most popular designs: patella tendon bearing and total surface bearing sockets, at the extremes with a series of variations that result in designs that are a compromise between both in the centre. This results in a difference in pressure of up to 31 kPa over the fibula head and 14 kPa over the residuum tip.

Acknowledgements We thank Herbert Ganter and the other members of the ImpAmp consortium under EUROSTARS project (ref 9396) for technical support and data collection.

Funding The authors would like to thank the following for their financial support: JWS: the University of Southampton's EPSRC Doctoral Training Program (ref EP/M508147/1); PRW: the EPSRC-NIHR 'Medical Device and Vulnerable Skin Network' (ref EP/M000303/1 and EP/N02723X/1); ASD: the Royal Academy of Engineering, UK, (ref RF/130).

Data availability Supporting data are openly available from the University of Southampton repository at https://doi.org/10.5258/SOTON/D0980

References

Al-Fakih EA, Osman NAA, Eshraghi A, Adikan FRM (2013) The capability of fiber Bragg grating sensors to measure amputees' trans-tibial stump/socket interface pressures. Sensors (Switzerland) 13(8):10348–10357

Beil TL, Street GM (2004) Comparison of interface pressures with pin and suction suspension systems. J Rehabil Res Dev 41(6A):821–828

Beil TL, Street GM, Covey SJ (2002) Interface pressures during ambulation using suction and vacuum-assisted prosthetic sockets. J Rehabil Res Dev 39(6):693–700

Bramley JL, Worsley PR, Bostan L, Bader DL, Dickinson AS (2019) Investigating the effects of simulated prosthetic loading on intact and trans-tibial residual limb dermal tissues. Prosthet Orthot Intl 43(Suppl):1

Colombo G, Facoetti G, Regazzoni D, Rizzi C (2013) A full virtual approach to design and test lower limb prosthesis. Virtual Phys Prototyp 8(2):97–111

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Dou P, Jia X, Suo S, Wang R, Zhang M (2006) Pressure distribution at the stump/socket interface in transtibial amputees during walking on stairs, slope and non-flat road. Clin Biomech 21(10):1067–1073

Dumbleton T et al (2009) Dynamic interface pressure distributions of two transtibial prosthetic socket concepts. J Rehabil Res Dev 46(3):405–415

Eklund A, Sexton S (2017) WHO standards for prosthetics and orthotics

Goh JCH, Lee PVS, Chong SY (2003a) Stump/socket pressure profiles of the pressure cast prosthetic socket. Clin Biomech 18(3):237–243

Goh JCH, Lee PVS, Chong SY (2003b) Static and dynamic pressure profiles of a patellar-tendon-bearing (PTB) socket. Proc Inst Mech Eng H 217(2):121–126

Goh JCH, Lee PVS, Chong SY (2004) Comparative study between patellar-tendon-bearing and pressure cast prosthetic sockets. J Rehabil Res Dev 41(3):491–501

Goh JCH, Lee PVS, Toh SL, Ooi CK (2005) Development of an integrated CAD-FEA process for below-knee prosthetic sockets. Clin Biomech 20(6):623–629

Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. Mach Learn 3(2):95–99

Grudniewski PA, Sobey AJ (2018) Behaviour of multi-level selection genetic algorithm (MLSGA) using different individual-level selection mechanisms. Swarm Evol Comput 44:1–29

Grudniewski PA, Sobey AJ (2019) cMLSGA: co-evolutionary multi-level selection genetic algorithm. Swarm Evol Comput Under Rev

Hachisuka K, Dozono K, Ogata H, Ohmine S, Shitama H, Shinkoda K (1998) Total surface bearing below-knee prosthesis: advantages, disadvantages, and clinical implications. Arch Phys Med Rehabil 79(7):783–789

Hoyt K, Kneezel T, Castaneda B, Parker KJ (2008) Quantitative sonoelastography for the in vivo assessment of skeletal muscle viscoelasticity. Phys Med Biol 53(15):4063–4080

Kerr M, Rayman G, Jeffcoate WJ (2014) Cost of diabetic foot disease to the National Health Service in England. Diabet Med 31(12):1498–1504

Kuhn GG (1966) Kondylen Bettung Münster am Unterschenkel-Stumpf 'KBM-Prothese. Atlas d'Appareillage Proth{é}tique et Orthop{é}dique 14

Laing S, Lythgo N, Lavranos J, Lee PVS (2017) Transtibial prosthetic socket shape in a developing country: a study to compare initial outcomes in pressure cast hydrostatic and patella tendon bearing designs. Gait Posture 58(July):363–368

Laing S, Lee PVS, Lavranos J, Lythgo N (2018) The functional, spatio-temporal and satisfaction outcomes of transtibial amputees with a hydrocast socket following an extended usage period in an under-resourced environment. Gait Posture 66(July):88–93

Laszczak P, Jiang L, Bader DL, Moser D, Zahedi S (2015) Development and validation of a 3D-printed interfacial stress sensor for prosthetic applications. Med Eng Phys 37(1):132–137

Lee P, Goh J, Cheung S (2000) Biomechanical evaluation of the pressure cast (PCast) prosthetic socket for transtibial amputee. In: Proceedings of the World congress on medical physics & biomedical engineering

Li M, Yao XIN (2018) Quality evaluation of solution sets in multi-objective optimisation: a survey. ACM Comput Surv 1(1):1–43

Lin Q et al (2016) A hybrid evolutionary immune algorithm for multiobjective optimization problems. IEEE Trans Evol Comput 20(5):711–729

Mak AF, Zhang M, Boone DA (2001) State-of-the-art research in lower-limb prosthetic biomechanics-socket interface: a review. J Rehabil Res Dev 38(2):161–174

McGrath MP et al (2017) Development of a residuum/socket interface simulator for lower limb prosthetics. Proc Inst Mech Eng Part H J Eng Med 231(3):095441191769076

Murdoch G (1964) The 'Dundee' socket—a total contact socket for the below-knee amputation. Orthop Prosthet Appl J 19:231–234

Palevski A, Glaich I, Portnoy S, Linder-Ganz E, Gefen A (2006) Stress relaxation of porcine gluteus muscle subjected to sudden transverse deformation as related to pressure sore modeling. J Biomech Eng 128(5):782

Portnoy S, Siev-Ner I, Yizhar Z, Kristal A, Shabshin N, Gefen A (2009) Surgical and morphological factors that affect internal mechanical loads in soft tissues of the transtibial residuum. Ann Biomed Eng 37(12):2583–2605

Radcliffe CW (1962) The biomechanics of below-knee prostheses in normal, level, bipedal walking. Artif Limbs 6:16–24

Safari MR, Meier MR (2015a) Systematic review of effects of current transtibial prosthetic socket designs—Part 2: quantitative outcomes. J Rehabil Res Dev 52(5):509–526

Safari MR, Meier MR (2015b) Systematic review of effects of current transtibial prosthetic socket designs—Part 1: qualitative outcomes. J Rehabil Res Dev 52(5):491–508

Sobey A, Blanchard J, Grudniewski P, Savasta T (2019) There's no free lunch: a study of Genetic Algorithm use in Maritime Applications. In: Conference on computer applications and information technology in the maritime industries

Staats T, Lundt J (1987) The UCLA total surface bearing suction below-knee prosthesis. Clin Prosthet Orthot 11(3):118–130

Steer JW, Browne M, Worsley PR, Dickinson AS (2019) Predictive prosthetic socket design. Part 1: population-based evaluation of transtibial prosthetic sockets by FEA-driven surrogate modelling. Biomech Model Mechanobiol

Tseng LY, Chen C (2007) Multiple trajectory search for multiobjective optimization. In: 2007 IEEE congress on evolutionary computation, CEC 2007, vol 758, pp 3609–3616

Wang Z, Bai J, Sobey A, Xiong J, Shenoi A (2018) Optimal design of triaxial weave fabric composites under tension. Compos Struct 201(June):616–624

Woldesenbet YG, Yen GG, Tessema BG (2009) Constraint handling in multiobjective evolutionary optimization. IEEE Trans Evol Comput 13(3):3077–3084

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization 1 introduction. IEEE Trans Evol Comput 1(1):67–82

Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput 11(6):712–731

Zhang M, Turner-Smith AR, Tanner A, Roberts VC (1998) Clinical investigation of the pressure and shear stress on the trans-tibial stump with a prosthesis. Med Eng Phys 20(3):188–198

Appendix E List of utilised parameters for genetic algorithms

For the ease of reproducibility, the parameters are presented as in the C++ code used for simulations[1].

/*GA parameters - MOEA/D*/

const float MOEAD_niche_multi = 0.1; //multiplier of the neighbour size (* pop size)

const float MOEAD_limit_multi = 0.01; //multiplier of the number of solutions replaced(* pop size)

const float MOEAD_mating_chance = 0.9; //probability chance for mating

const int MOEAD_new_sol_multi = 5; //How many new individuals are generated each generation = pop_size/MOEAD_new_sol_multi

const int MOEAD_evol_operation_type = 1; //Define type of evol. operations; 1 - DE cross. and poly. mut.; 2 - LL cross. and mut.

const std::string MOEAD_weight_generator "Uniform" ; //Define how the MOEA/D generate weight vectors; "File" - get from file in /Input/MOEADWeight; "Sobol" - generated by Sobol - not implemented properly; "Uniform" - uniform method from MSF/PSF and mine

const std::string MOEAD_weight_assign "Pop" ; //Define how the weight vectors are assigned to the population - in case of MLS; "Pop" -Assigned to population before the collective split; "Col" - Assigned to separately to each collective; "Sep" - Each collective have separate weight vector from 0. to 1. (i.e. each covers the whole search space)

/*GA parameters - MTS*/

const int MTS_LocalSearch_test_amount = 5; //How many times each local search is tested each generation

const int MTS_LocalSearch_amount = 45; //How many local searches are made each generation

[1]The code of MLSGA is made available online at https://bitbucket.org/Pag1c18/cmlsga

const float MTS_foreground_multp = 0.125; //Size of the foreground (* Population size)

const int MTS_MPL1 = 9; //Bonus 1

const int MTS_MPL2 = 2; //Bound 2

/*GA parameters - BCE*/

const int BCE_PC_capacity = 600; //maximum size of the PC population

const short BCE_mode = 2; //setting variation in the NPC and PC evolution; 1: SBX crossover, 2: DE **/*GA parameters - HEIA*/**

const float HEIA_crossover_prob = 0.5; //Crossover probability for HEIA (between SBX and DE)

/*Additional GA parameters*/

const unsigned short Di_c = 20; //Distribution index for SBX crossover

const unsigned short Di_m = 20; //Distribution index for polynomial mutation

const float mut_prob_min = 0.08; //Mutation probability

const float cross_prob_min = 1; //Crossover probability

const int Pop_size = 600; //Population size for all algorithms except of MLSGA and MTS

const int Pop_size_MTS = 100; //Population size

/*MLSGA parameters*/

const std::vector<std::vector<std::string>> GA_mode "HEIA", "MEOADMSF" ; //Used evolutionary algorithms for cMLSGA

const std::vector<std::vector<std::string>> GA_mode "HEIA"; //Used evolutionary algorithms for MLSGA-hybrid

const unsigned short MLSGA_n_col = 8; //Numbers of collectives

const unsigned short MLSGA_n_col_elim = 1; //Number of collectives eliminated

const unsigned short MLSGA_col_elim_limit = 10; //Define how many generations have to pass before collective elimination occurs

const unsigned short MLSGA_n_MLS = 7; //Index of the MLS type used: 7 - MLS-U; 1 - MLS1; 2 - MLS2R; 3 - MLS2

Appendix F Description of utilised engineering problems

3D spatial arrangement of leisure boats

The leisure boatbuilding industry is a competitive market and therefore it is essential to minimise the cost of overall production by utilising as much space as possible. Due to the curved shape of the boats and smaller sizes of leisure boats, this is a complex task with additional problems such as requiring as much space as possible for customer used rooms and as much space as necessary for practical rooms, such as the engine. The case shows a nice "toy problem" to determine methods with the best chance of success on larger vessels.

The first objective of the optimisation is therefore to minimise the wasted volume space in the hull defined as:

$$f_1 = (V_h - V_c)/V_h, \tag{1}$$

where, V_h, is the volume of the hull calculated based on the lines plan of the motor yacht which has the parameters given in the Table F.1 representing typical values for a motor yacht and V_c is the sum of all of the volumes of all of the cabins.

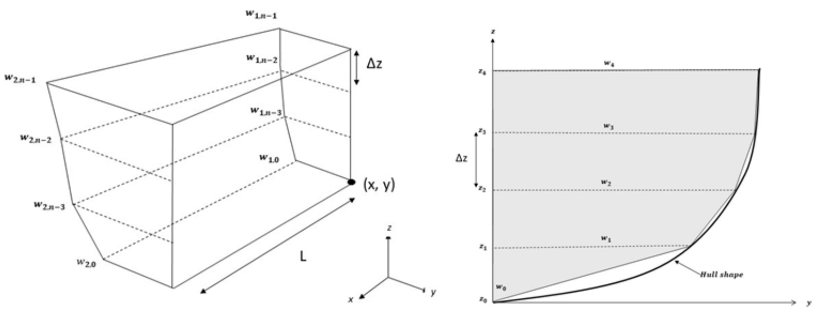


FIGURE F.1: The 3D cabin geometry and the cabin face

Table F.1: Motor yacht particulars.

Parameter	Value
Length	24.8m
Beam	6.9m
Draught	1.955m
Displacement	152.8t
LCB	8.673m

The overall design is split into a number of building blocks, each representing a different cabin. Therefore, each block is optimised semi-separately and then fit into overall design. Each cabin can be defined as two location parameters (x, y), the length L and a number of the offset w points on each side of the cabin, as presented in fig. F.1. Therefore, the total number of variables is defined as:

$$d = n_c * (3 + n_w * 2), \tag{2}$$

Where, n_c, is the number of cabins and n_w, is the number of offsets on each side. There is a balance between the number of offsets, the accuracy and the computational time. Too many offsets increase the complexity with no increase in accuracy but too few will give a poor utilisation volume. In this case the number of offsets is equal to 7 and 4 cabins are optimised, which results in 68 decision variables.

Due to the different masses of each cabin, which change during the optimisation process as they change size, the trimming of the boat becomes an issue and needs to be optimised. Therefore, the second objective function is defined as the normalised distance between the longitudinal centre of buoyancy (LCB) and longitudinal centre of gravity (LCG) as given in eq. 3, which should be minimised,

$$f_2 = (LCB - LCG)/LCB. \tag{3}$$

The LCG is calculated by summing the mass moments of each cabin and dividing by the total mass of all the cabins, while LCB is derived from the boat lines plan and the design water line. As the cabins are optimised separately, three constraints are added to ensure that the cabins form a valid formation. Firstly, the cabins are not allowed to overlap, and the first constraint is defined as:

$$c_1 = min \begin{cases} x_i + l_i \leq x_j \\ x_i - l_j \geq x_j \\ y_i + max(w_{i0}, w_{i1} \dots w_{n_w - 1}) \leq y_j \\ y_i - max(w_{i0}, w_{i1} \dots w_{n_w - 1}) \geq y_j \end{cases} \tag{4}$$

Secondly, all cabins must be within the yacht hull's boundaries. As the hull shape is defined by a lines plan, the calculation of the maximum allowable breadth is composed of three steps corresponding to interpolations in the (x, z), (y, z) and (x, y) planes, respectively.

Thirdly, all of the cabins volume has to be within a 20% margin of the predefined standard sizes. The given limit is a simplifying assumption and the standard sizes are derived from linear regression of 39 similar vessels. This ensures that the cabins are realistically sized.

Structural design of composite stiffened plates

Composite materials are often selected due to their high strength/weight ratio and ability to design the properties to a given parameters. They are the main materials used for composite leisure boat hulls under 130ft with all of the internal structure also normally made from these materials. These structures can be simplified to a flat plated grillage, with a simple analytical model developed to be accurate to FEA, an accurate modelling method common in structural design, within 5% [146]. The representation of the stiffened plates is given in fig. F.2. In order to minimise the mass of the plate, and thus the cost, while maintaining sufficient structural strength it is essential to choose the number of the stiffeners and their geometrical parameters correctly.

The typical design parameters are the number of transverse and longitudinal stiffeners; stiffener heights; stiffener widths; thickness of the stiffener crown; thickness of the stiffener web; and overall thickness of the plate. These parameters are taken from [159] and are detailed with respectable lower and upper boundaries in Table F.2 and represented in fig. F.3. All parameters are given in millimetres.

In order to simplify the problem only rectangular stiffeners are considered ($a = e$), the amount of longitudinal and transverse stiffeners is assumed to be the same to simplify the problem and the stiffeners are evenly spread across the whole plate. The size of the plate is predefined and equal to 3810mm in both length and breadth and the uniform pressure of 137 kPa is

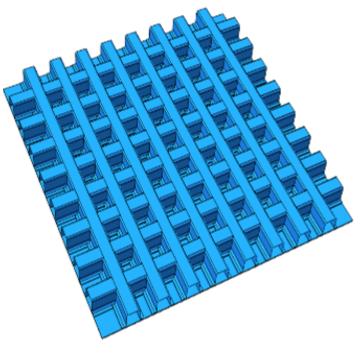


Figure F.2: 8by8 stiffened plate.

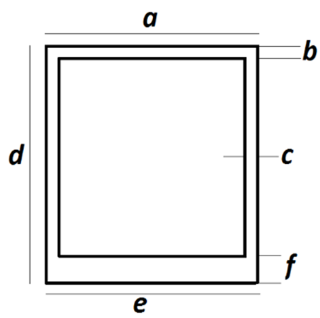


Figure F.3: Cross section of a stiffener.

Table F.2: Design parameters of the stiffener

Parameter	**Variable**	**Lower boundary**	**Upper boundary**
Number of stiffeners	N_s	2	8
Width of crown	a	150	300
Crown thickness	b	5	20
Web thickness	c	5	20
Height	d	150	300
Base width	e	150	300
Plate thickness	f	5	20

Table F.3: Fitness function assignment for the stiffener optimisation

	CLPT2	**CLPT3**
f_1	mass	
f_2	stress	
f_3	w/o	deflection

applied to the structure. The material is carbon/epoxy composite for which the properties can be found in [146].

In the presented problem, three objectives are considered: overall mass of the structure, total stress in the outer layer of the stiffener and deflection. All objectives are to be minimised and are calculated using a modified Navier grillage model with Classical Laminate Plate Theory (CLPT2) implemented as described in detail in [146]. In the second case, the problem is simplified and only the mass and stress are set as objectives and the deflection is neglected. The respective objectives for each case are presented in Table F.3

Appendix G Additional figures

Enlarged heat maps from Chapter 3

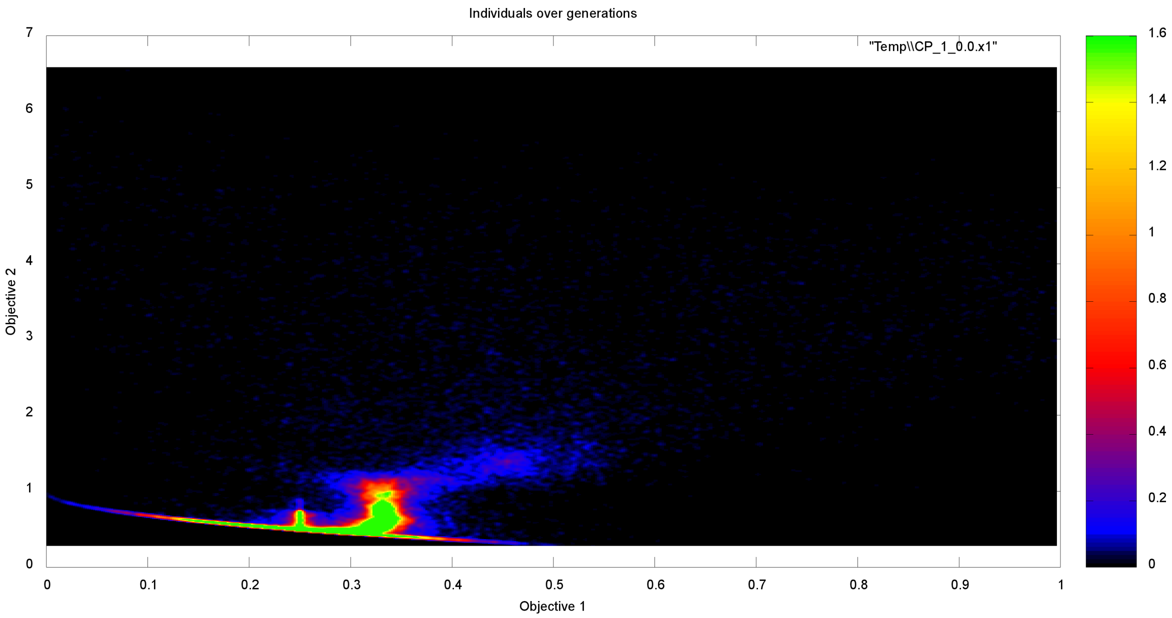


FIGURE G.1: The heat map of the MLS1 variant of MLSGA on ZDT1 problem.

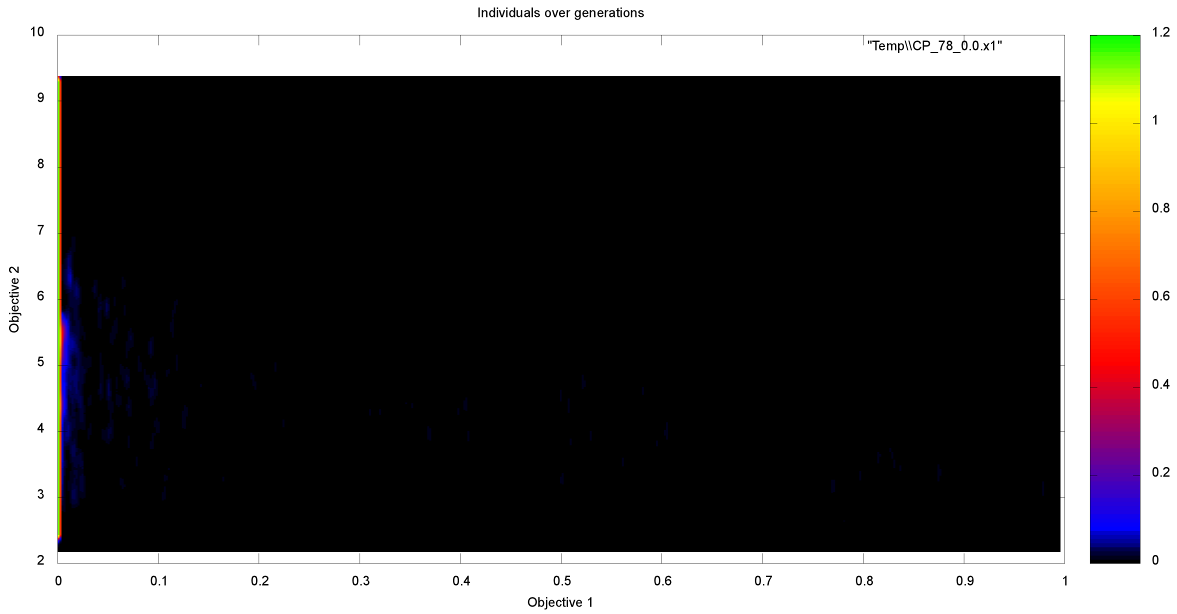


FIGURE G.2: The heat map of the MLS2 variant of MLSGA on ZDT1 problem.

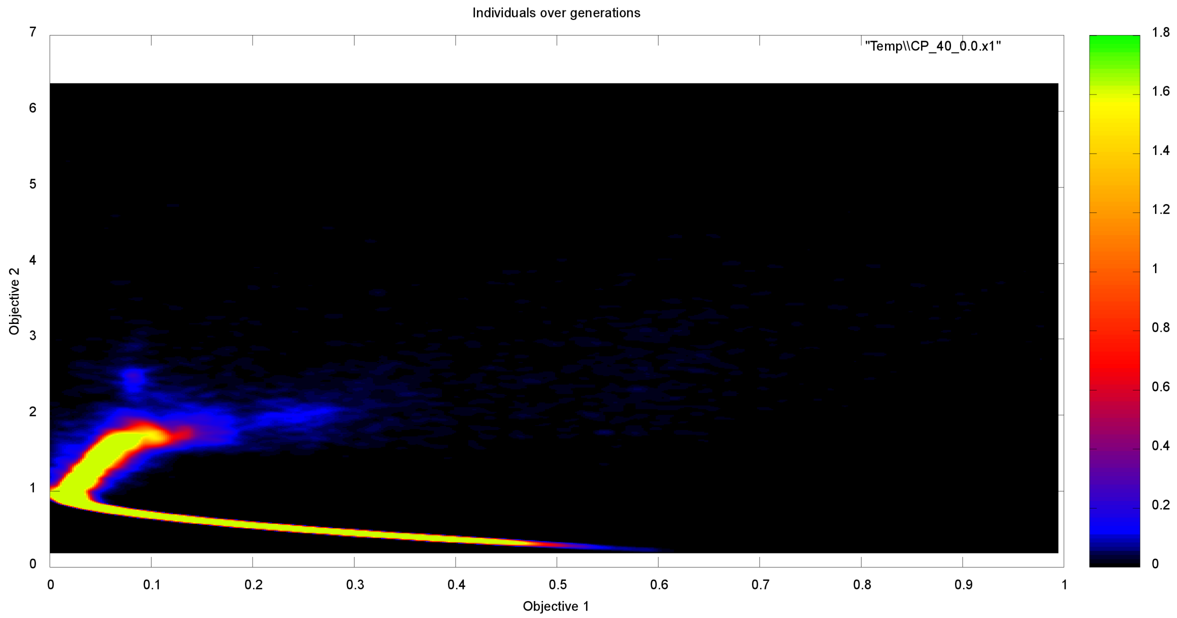


FIGURE G.3: The heat map of the MLS2R variant of MLSGA on ZDT1 problem.

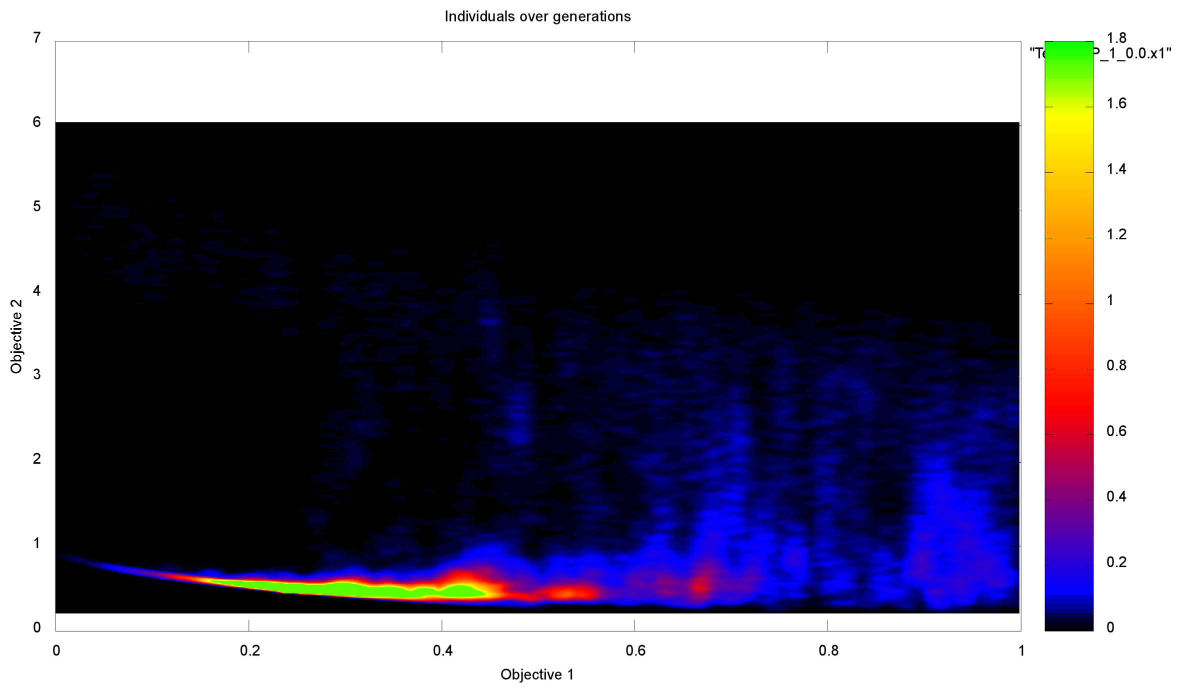


FIGURE G.4: The heat map of the single population GA on ZDT1 problem.

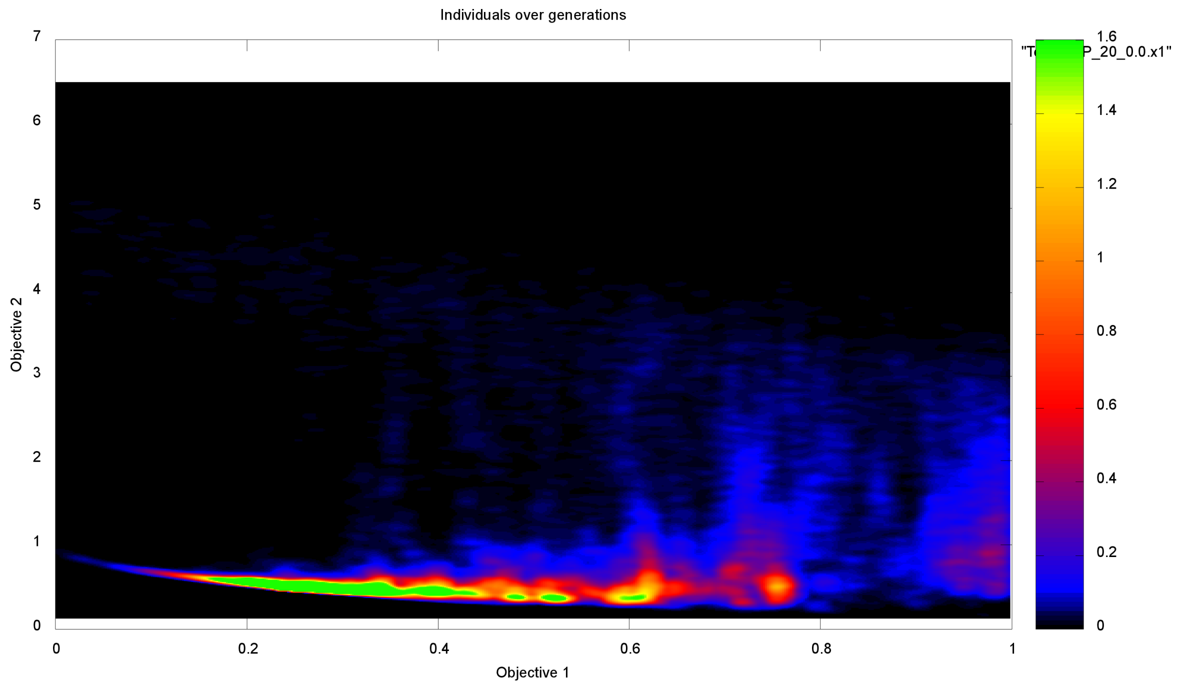


FIGURE G.5: The heat map of the sub-population GA on ZDT1 problem.

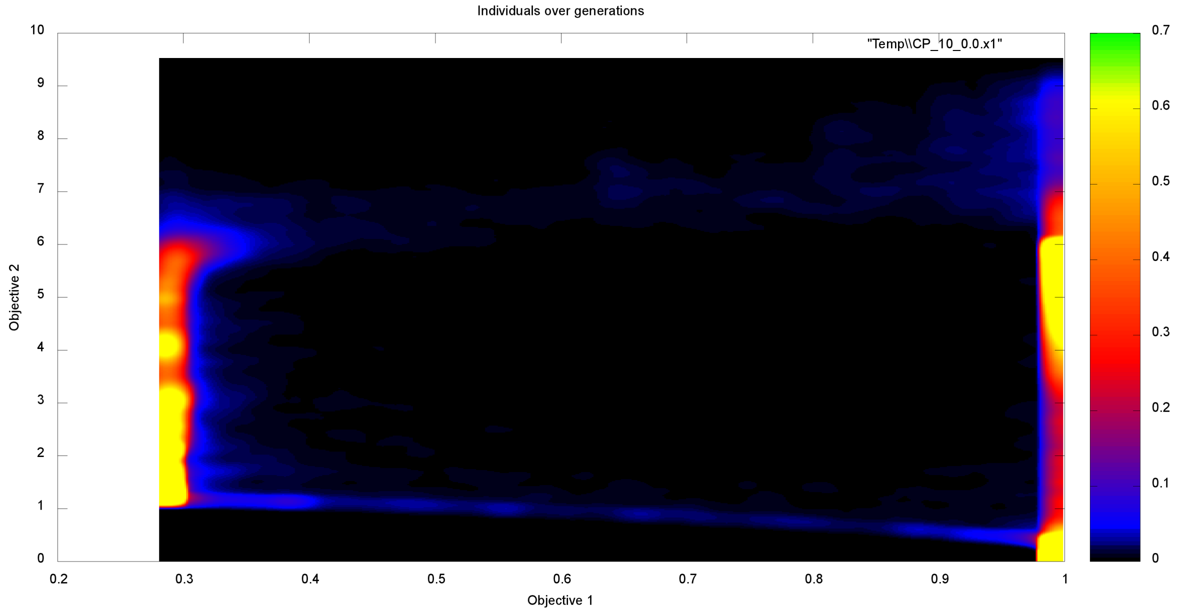


FIGURE G.6: The heat map of the MLS1 variant of MLSGA on ZDT6 problem.

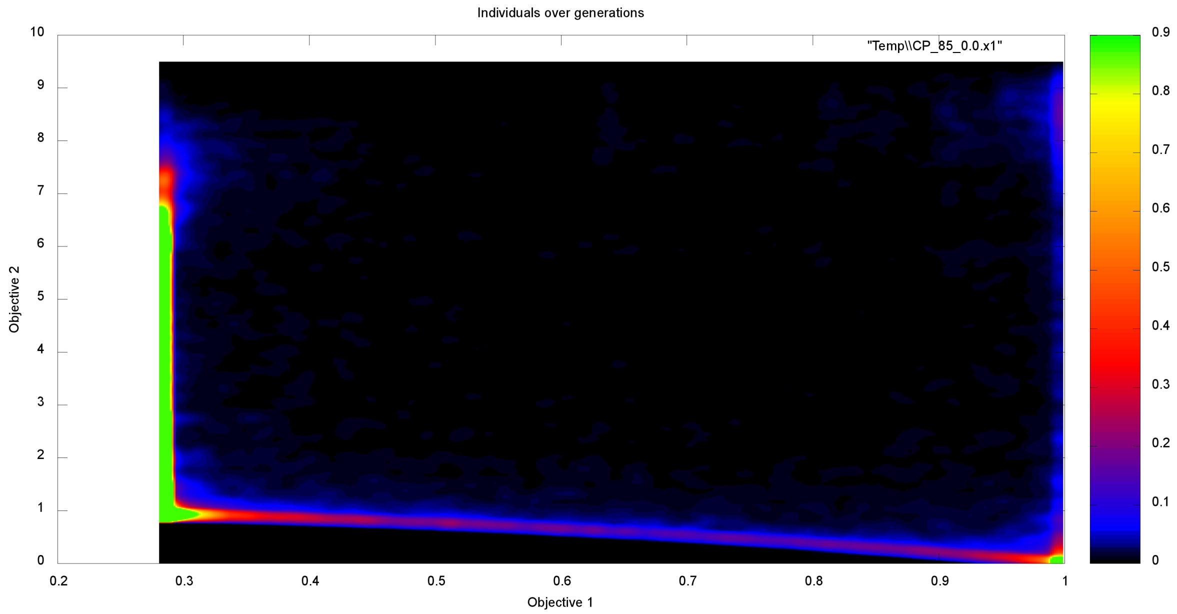


FIGURE G.7: The heat map of the MLS2 variant of MLSGA on ZDT6 problem.

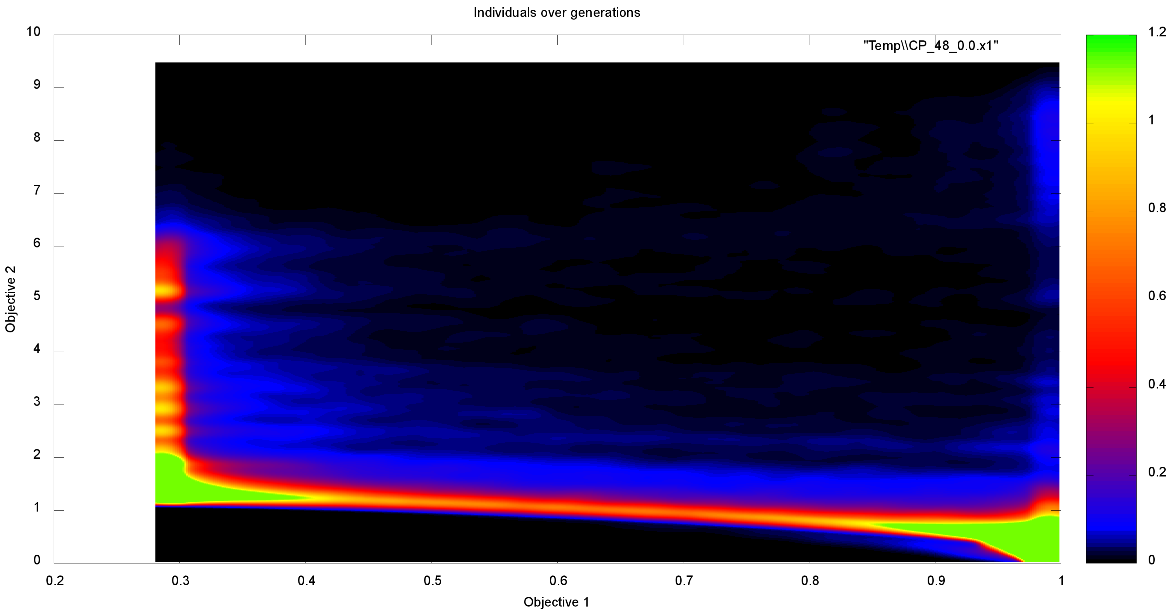


FIGURE G.8: The heat map of the MLS2R variant of MLSGA on ZDT6 problem.

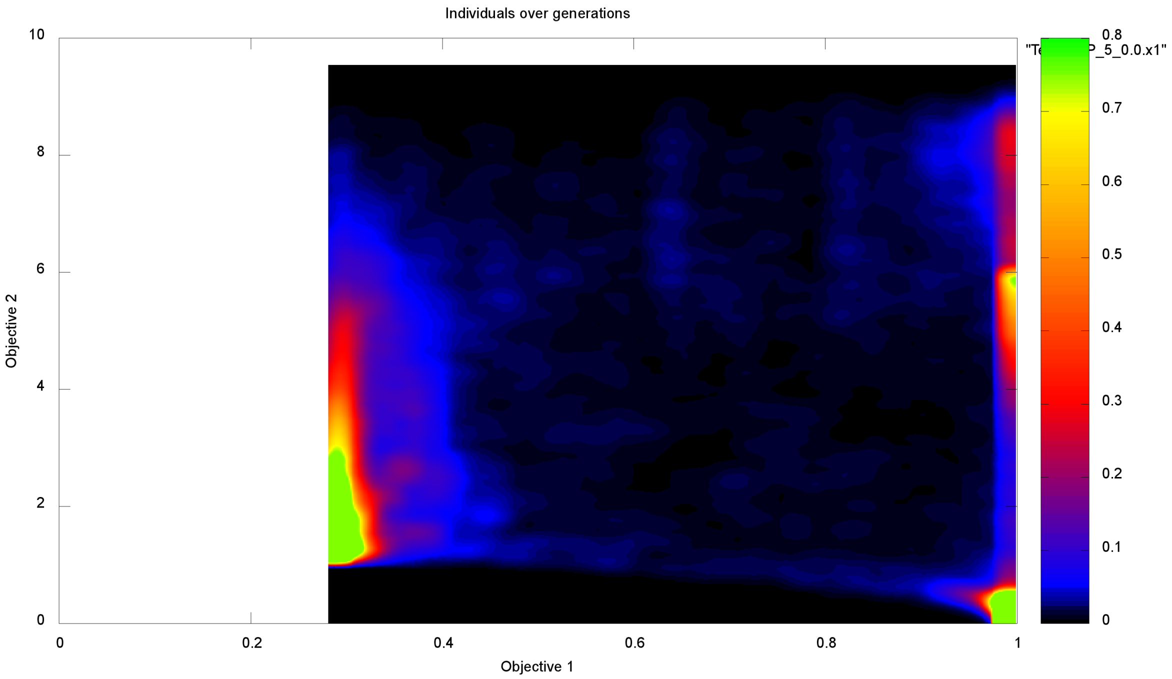


FIGURE G.9: The heat map of the single population GA on ZDT6 problem.

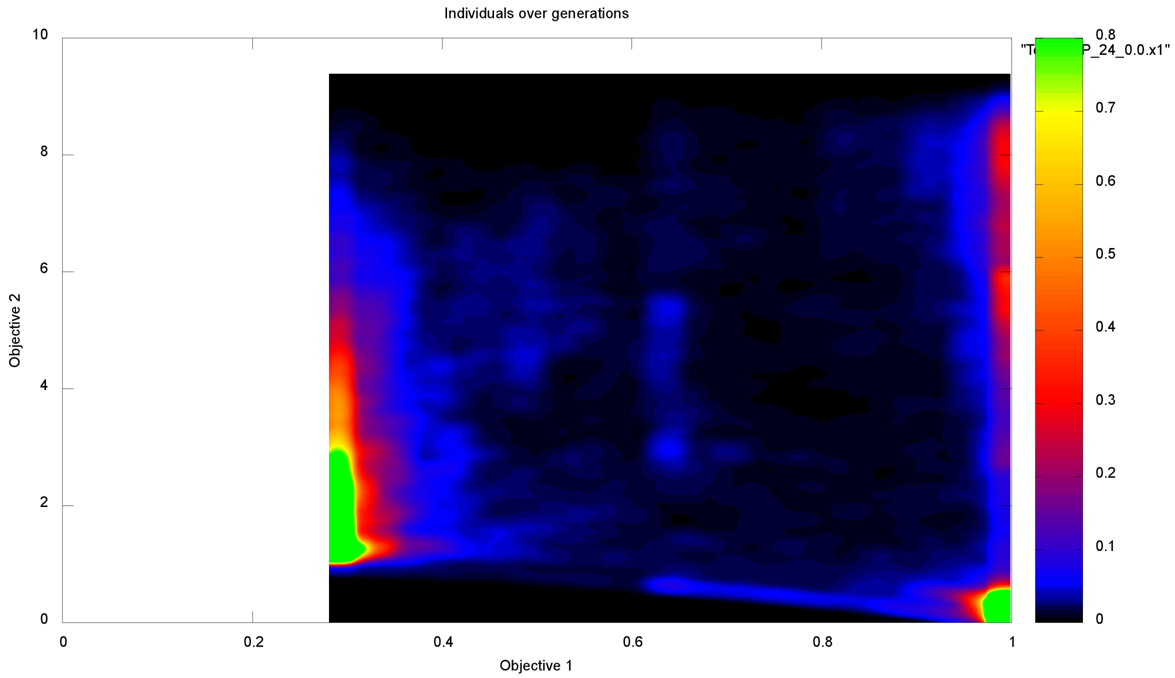


FIGURE G.10: The heat map of the sub-population GA on ZDT6 problem.

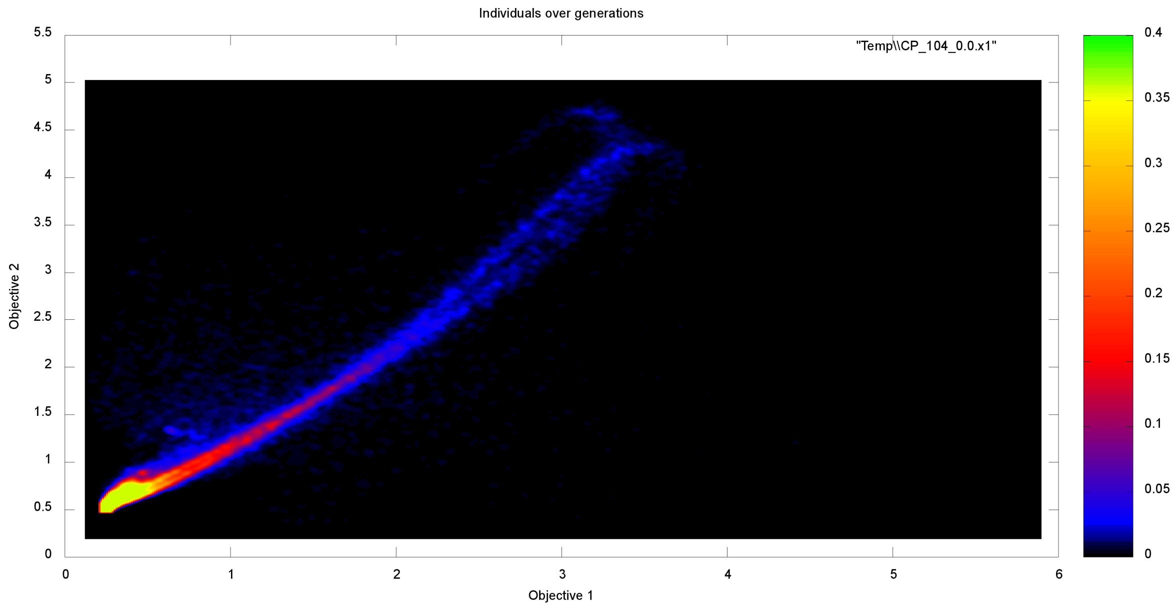


FIGURE G.11: The heat map of the MLS1 variant of MLSGA on CF2 problem.

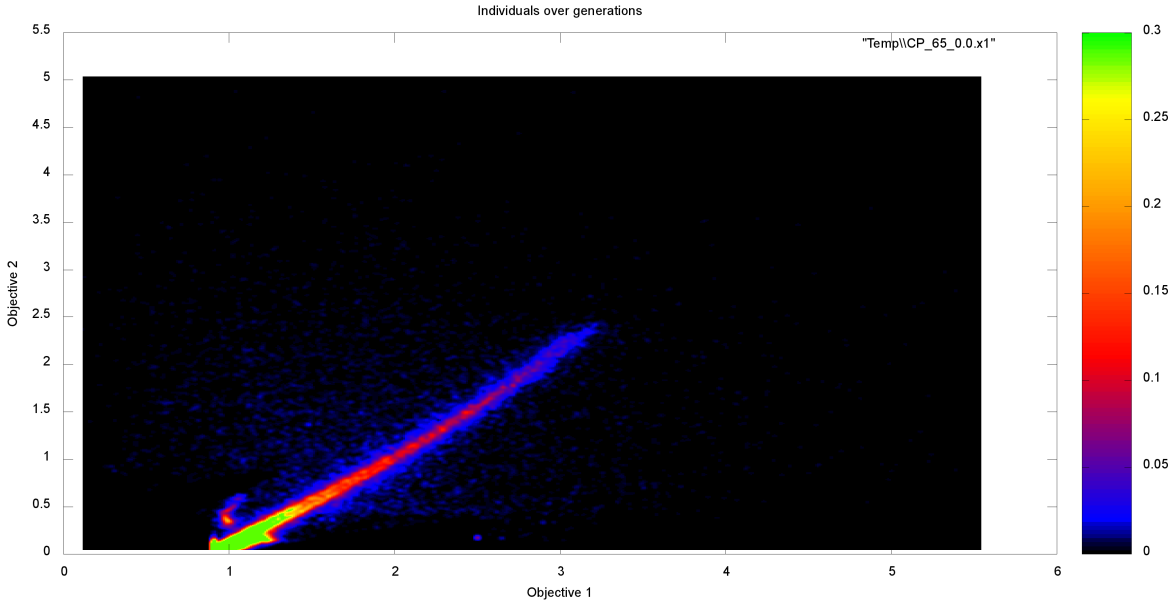


Figure G.12: The heat map of the MLS2 variant of MLSGA on CF2 problem.

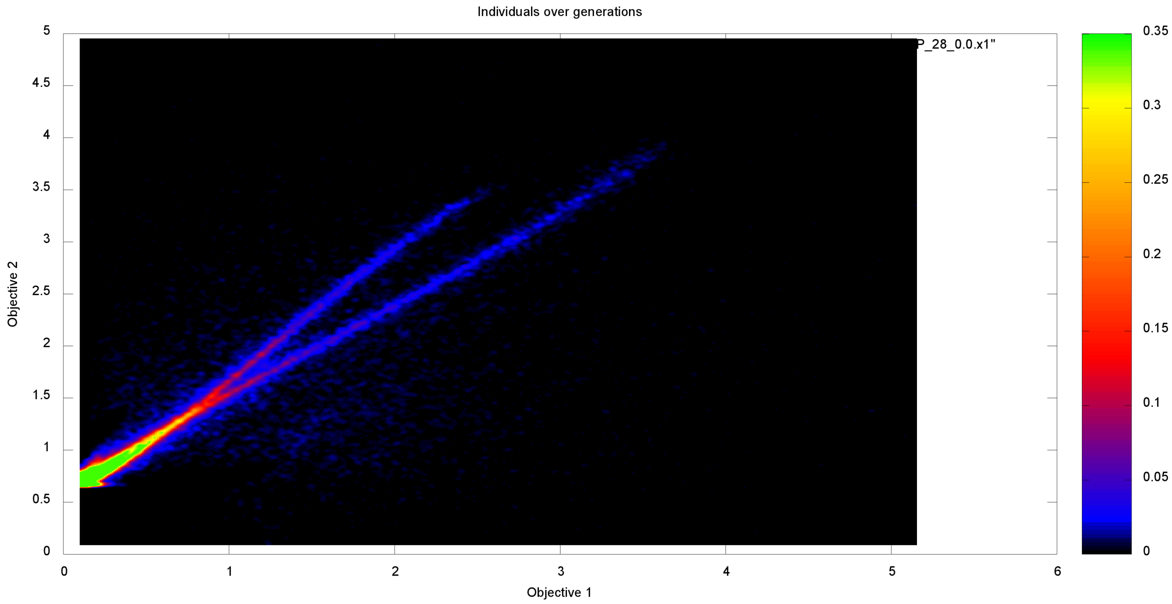


Figure G.13: The heat map of the MLS2R variant of MLSGA on CF2 problem.

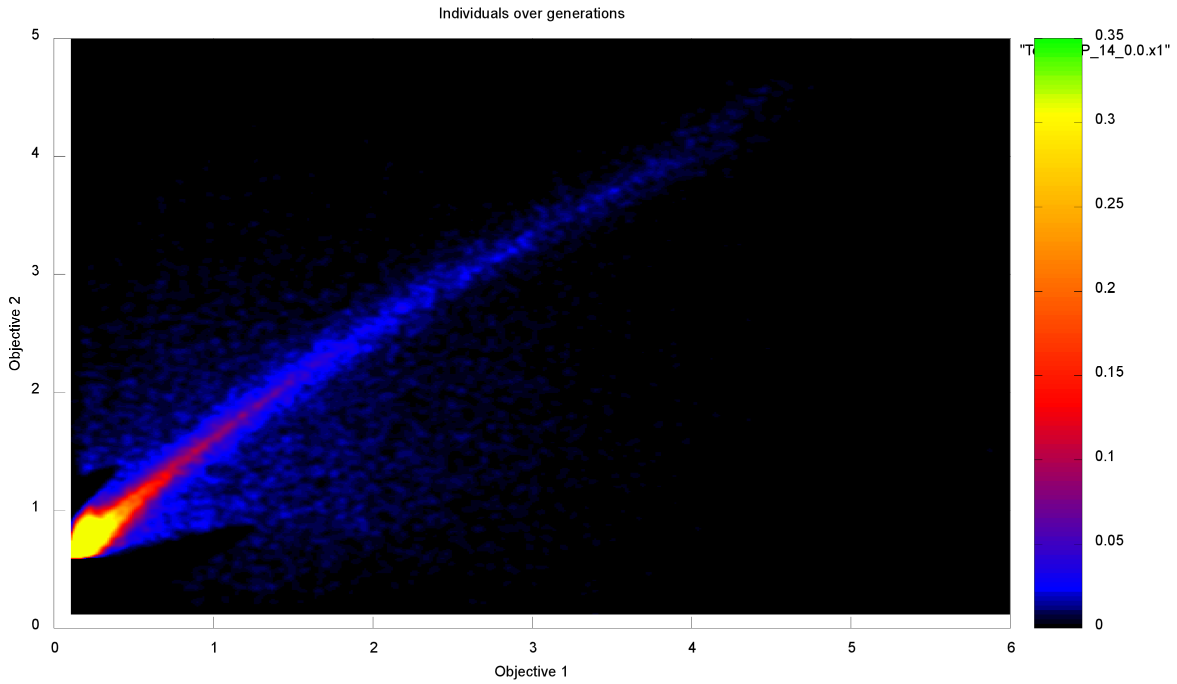


FIGURE G.14: The heat map of the single population GA on CF2 problem.

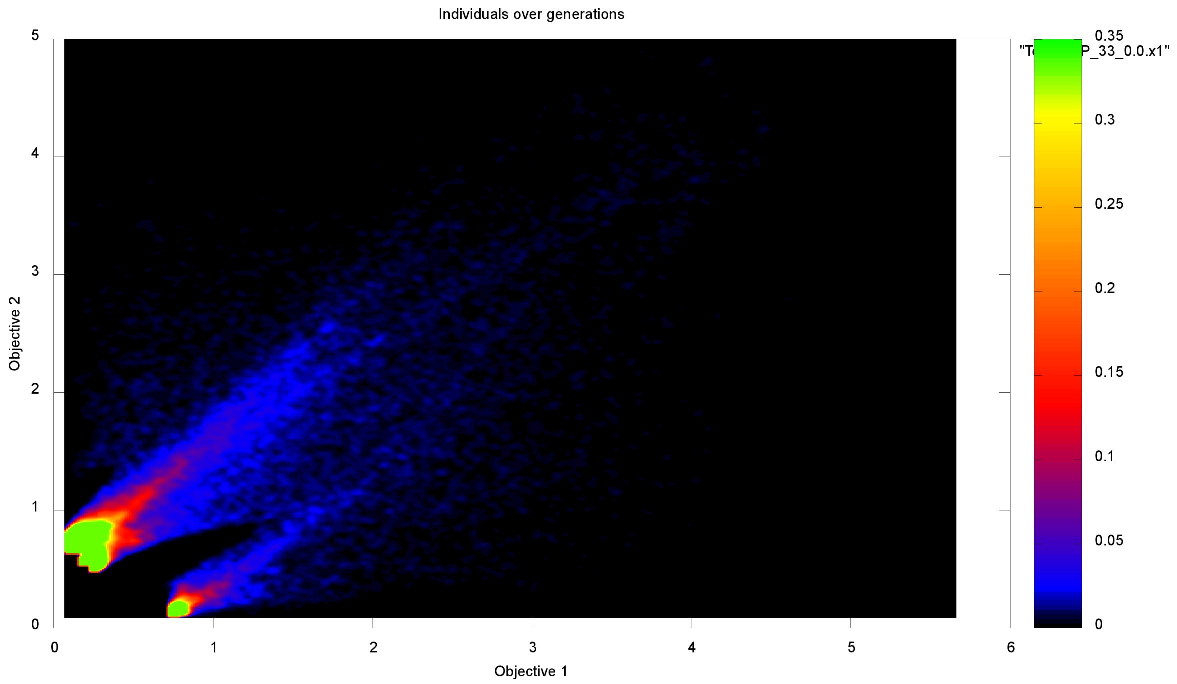


FIGURE G.15: The heat map of the sub-population GA on CF2 problem.

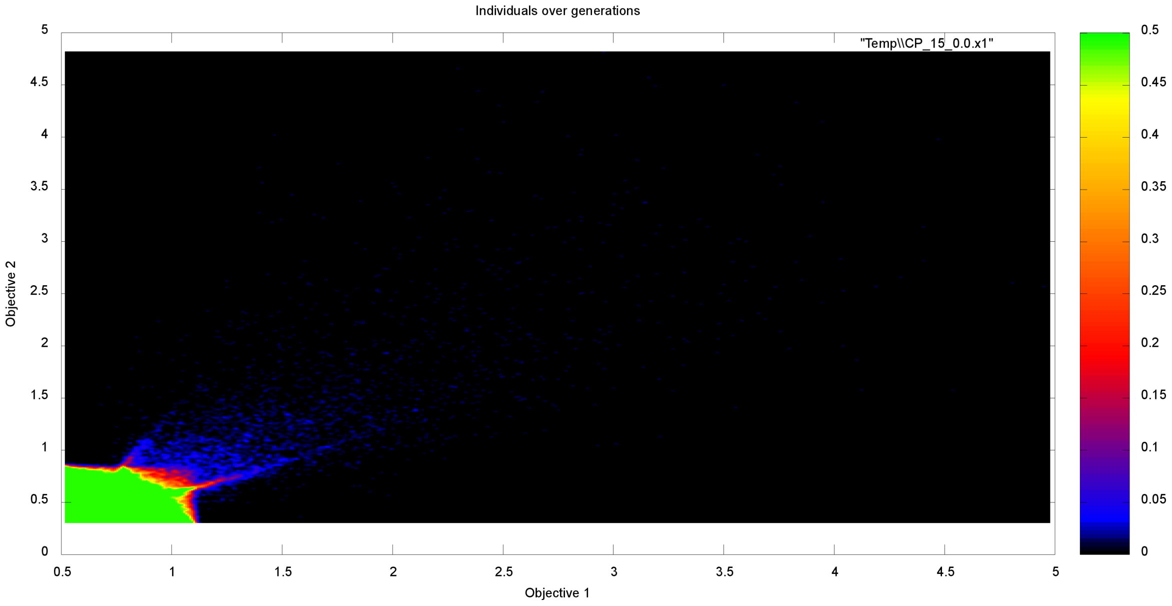


FIGURE G.16: The heat map of the MLS1 variant of MLSGA on UF3 problem.

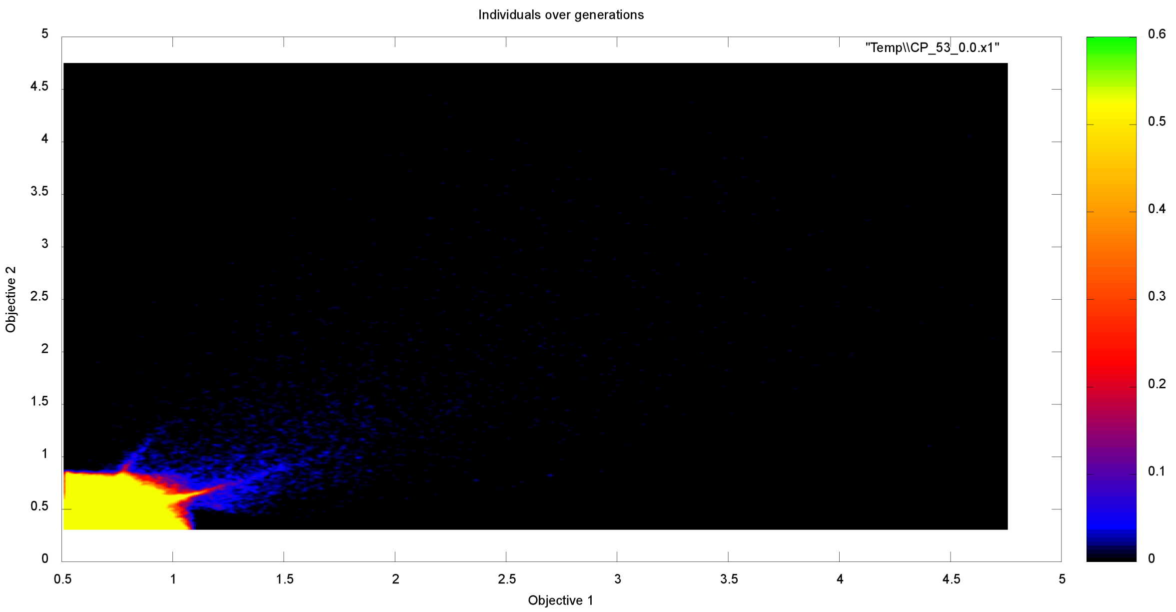


FIGURE G.17: The heat map of the MLS2 variant of MLSGA on UF3 problem.

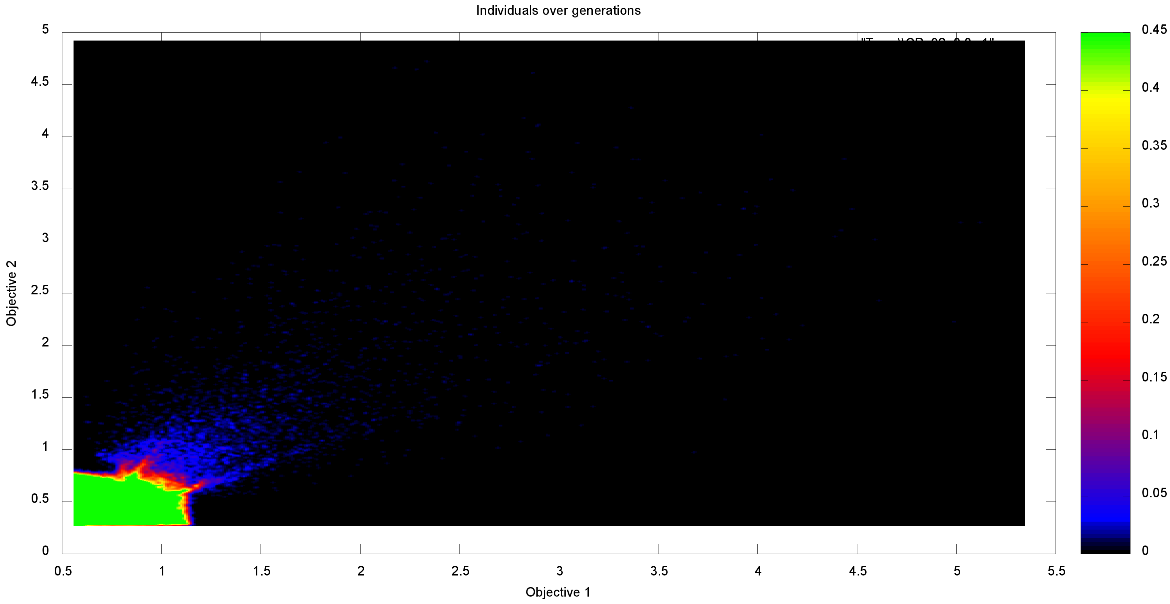


FIGURE G.18: The heat map of the MLS2R variant of MLSGA on UF3 problem.

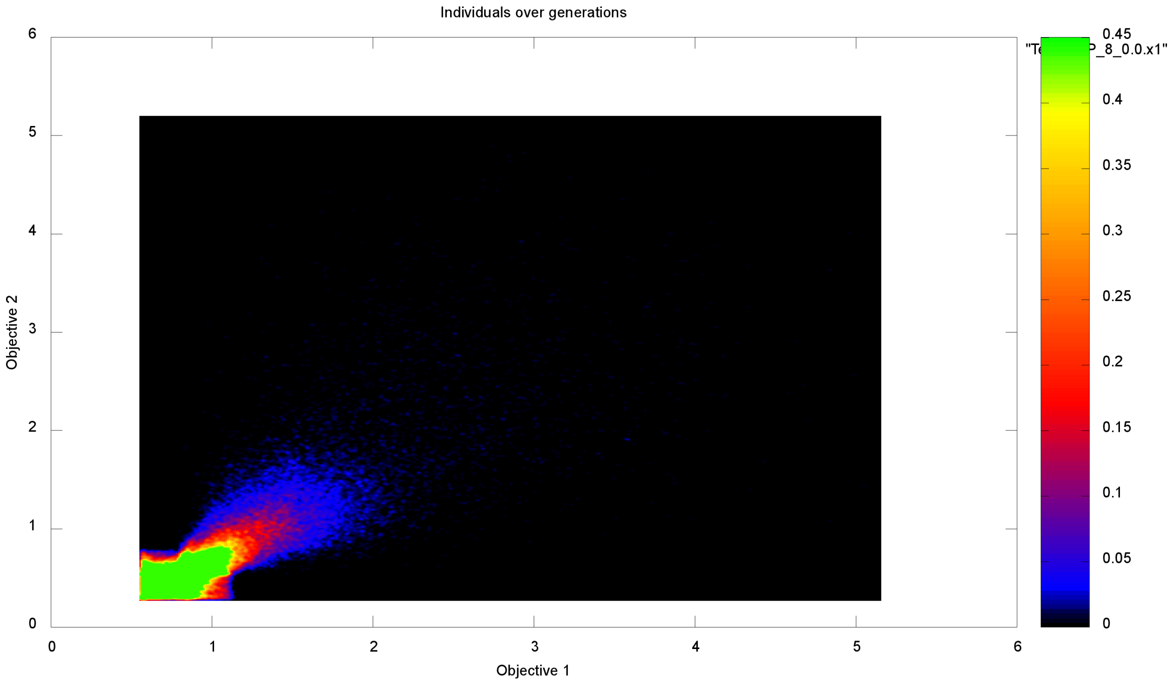


FIGURE G.19: The heat map of the single population GA on UF3 problem.

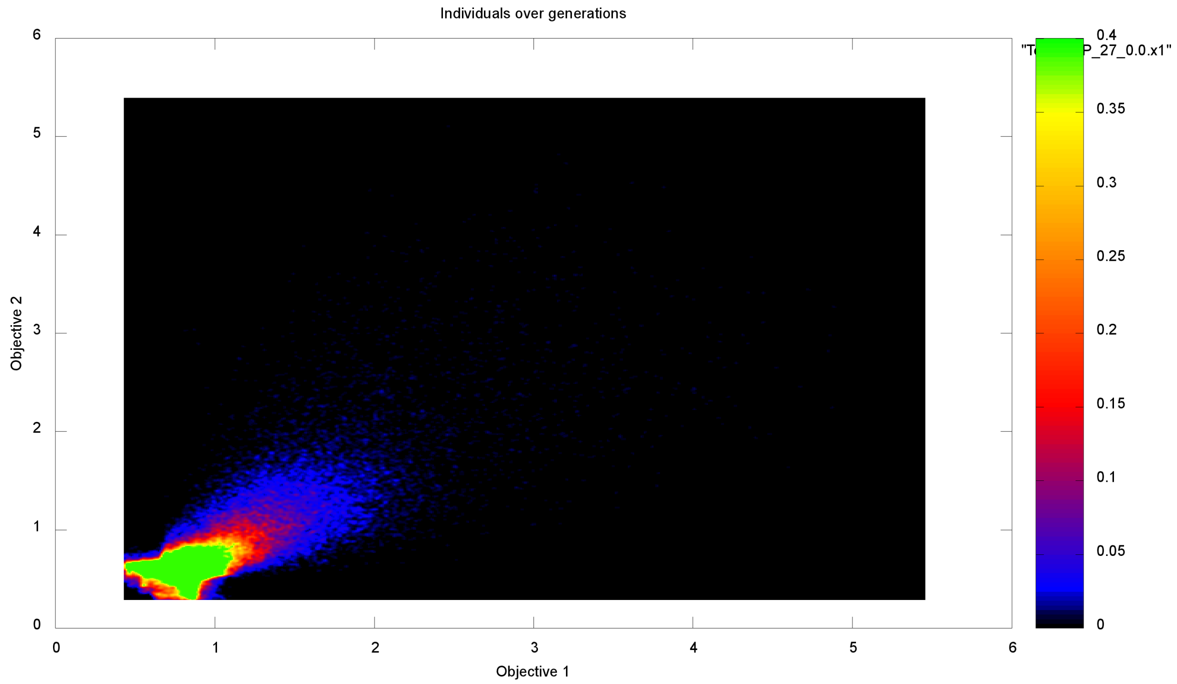


FIGURE G.20: The heat map of the sub-population GA on UF3 problem.

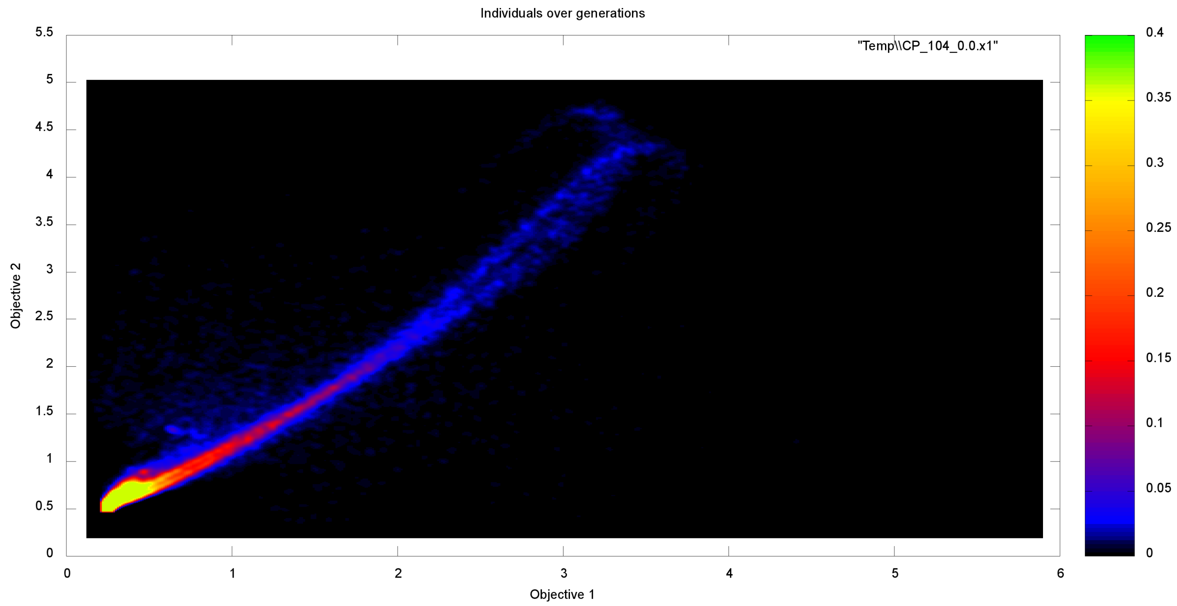


FIGURE G.21: The heat map of the MLS1 variant of MLSGA on CF2 problem.

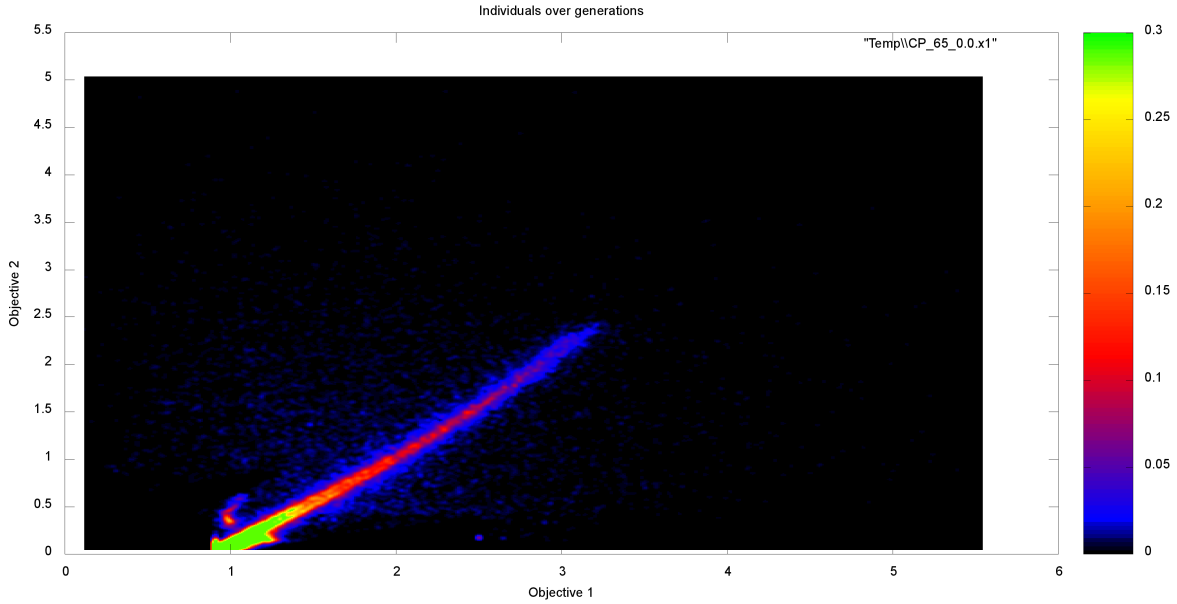


FIGURE G.22: The heat map of the MLS2 variant of MLSGA on CF2 problem.

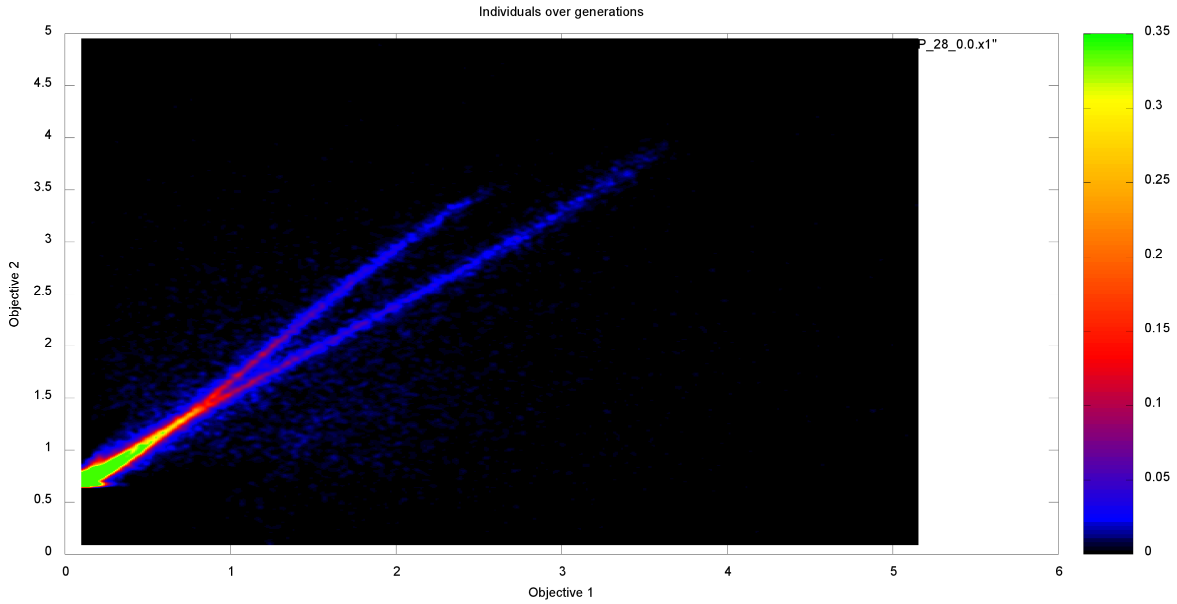


FIGURE G.23: The heat map of the MLS2R variant of MLSGA on CF2 problem.

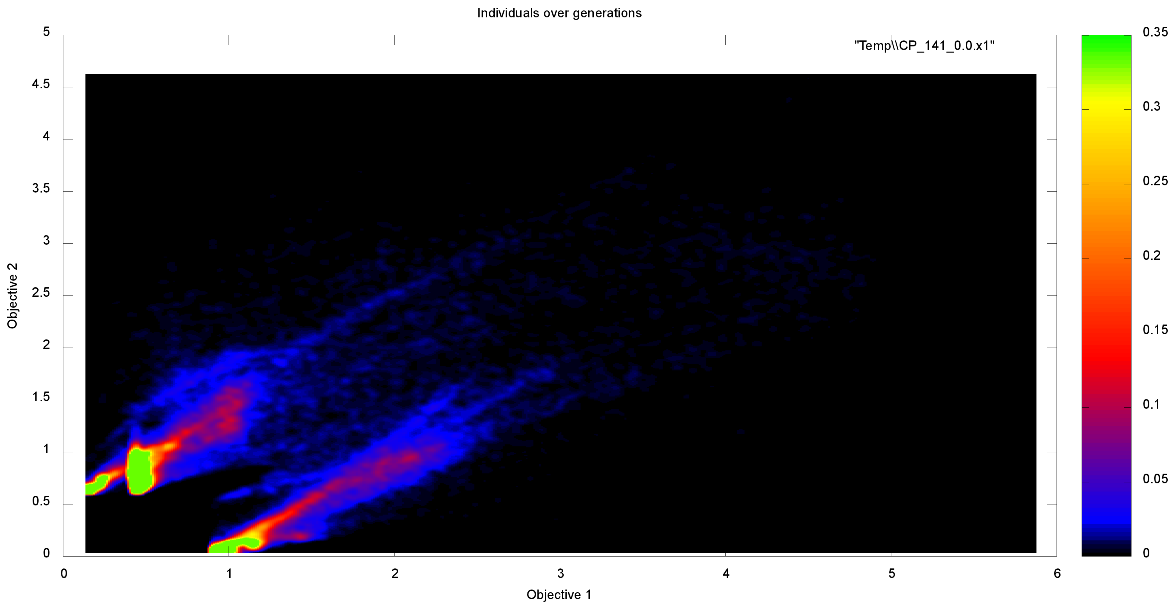


FIGURE G.24: The heat map of the MLSU variant of MLSGA on CF2 problem.

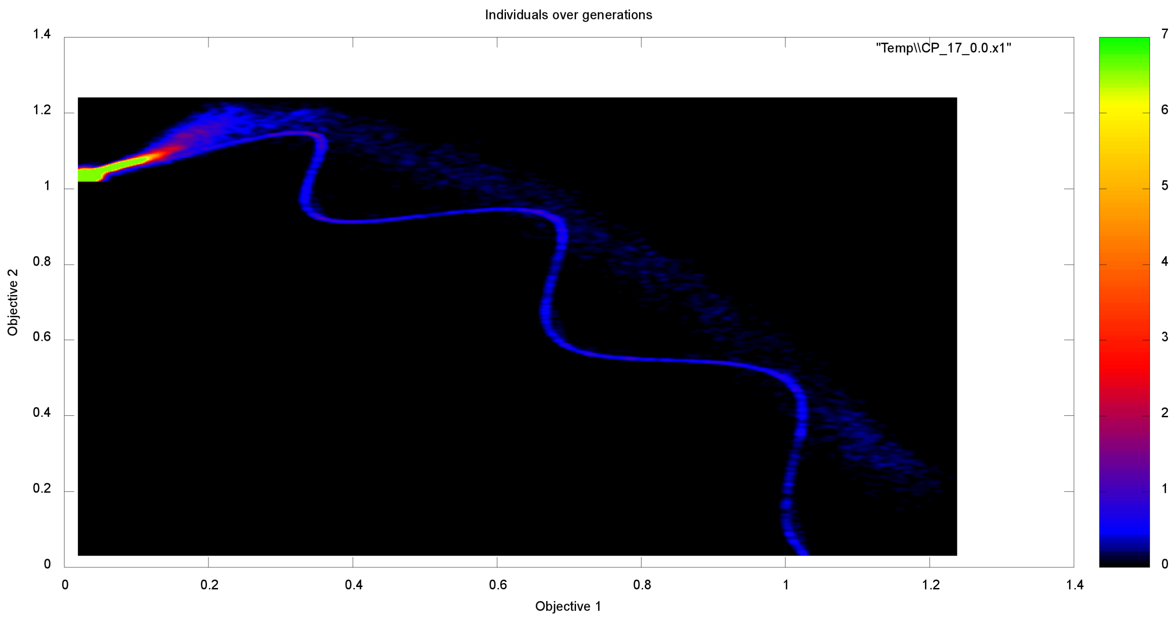


FIGURE G.25: The heat map of the MLS1 variant of MLSGA on UF4 problem.

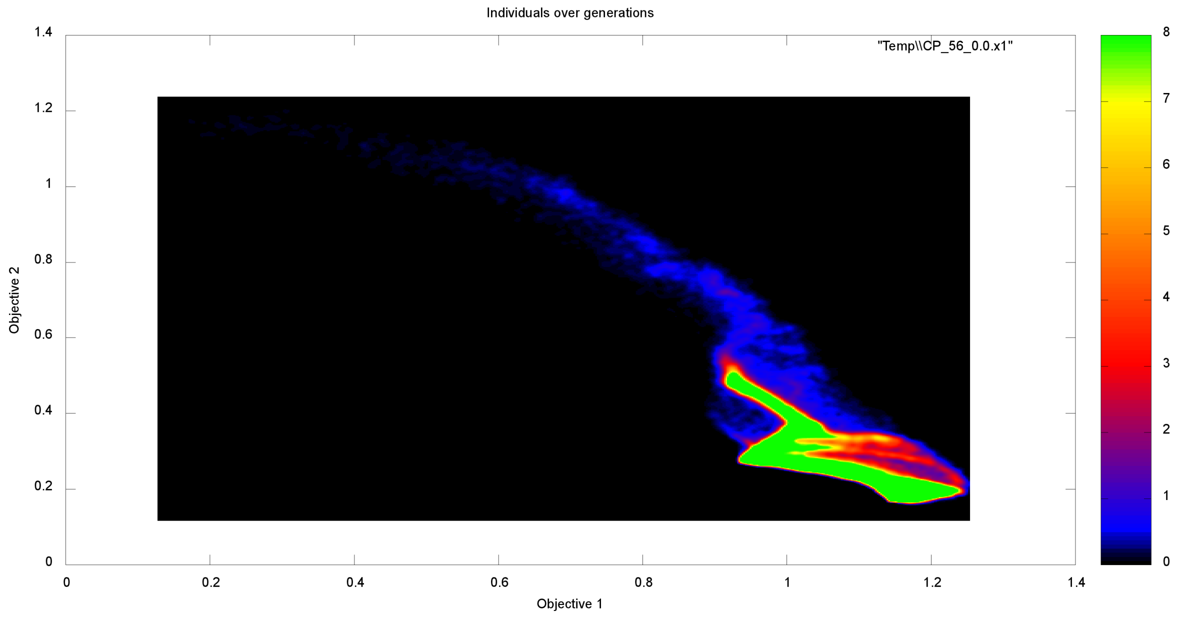


FIGURE G.26: The heat map of the MLS2 variant of MLSGA on UF4 problem.

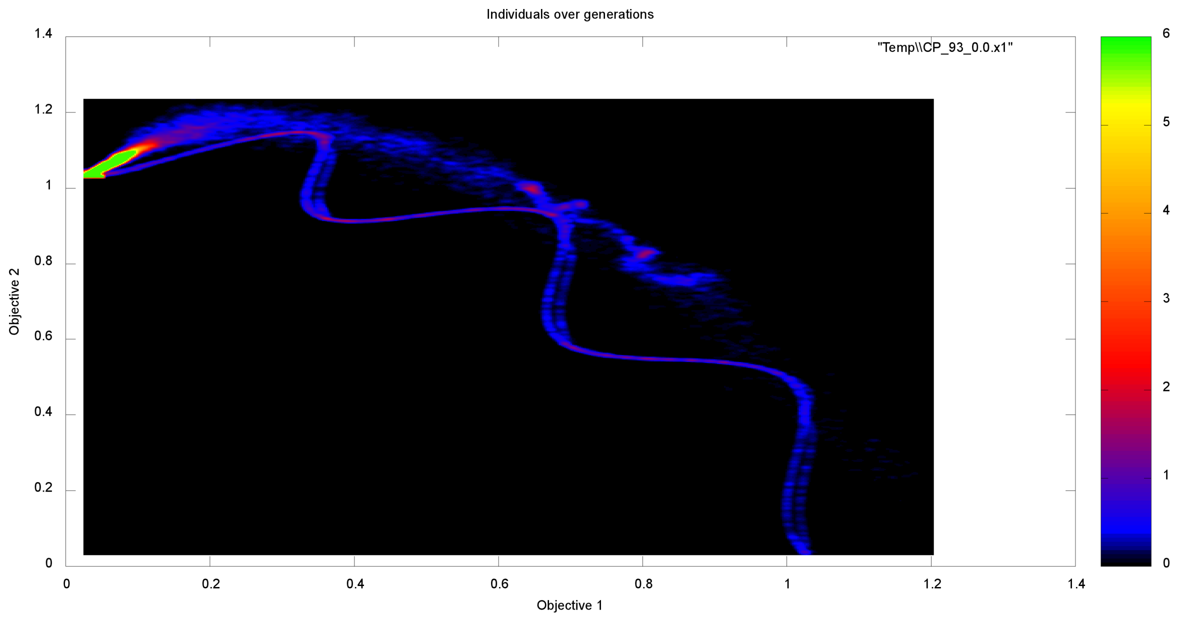


FIGURE G.27: The heat map of the MLS2R variant of MLSGA on UF4 problem.

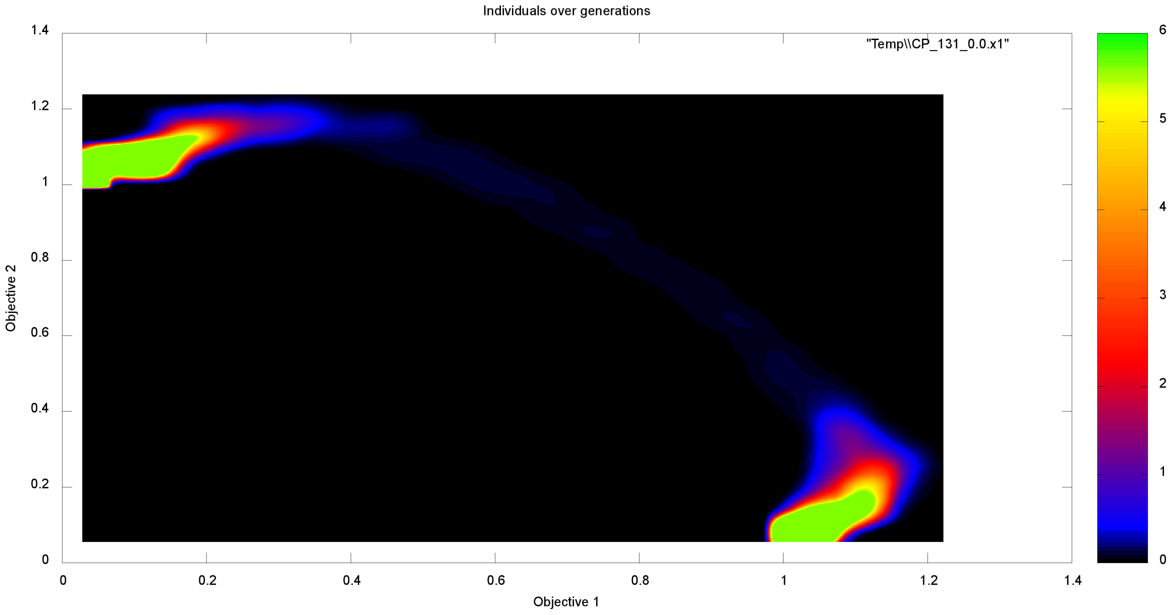


Figure G.28: The heat map of the MLSU variant of MLSGA on UF4 problem.

Enlarged heat maps from Chapter 4

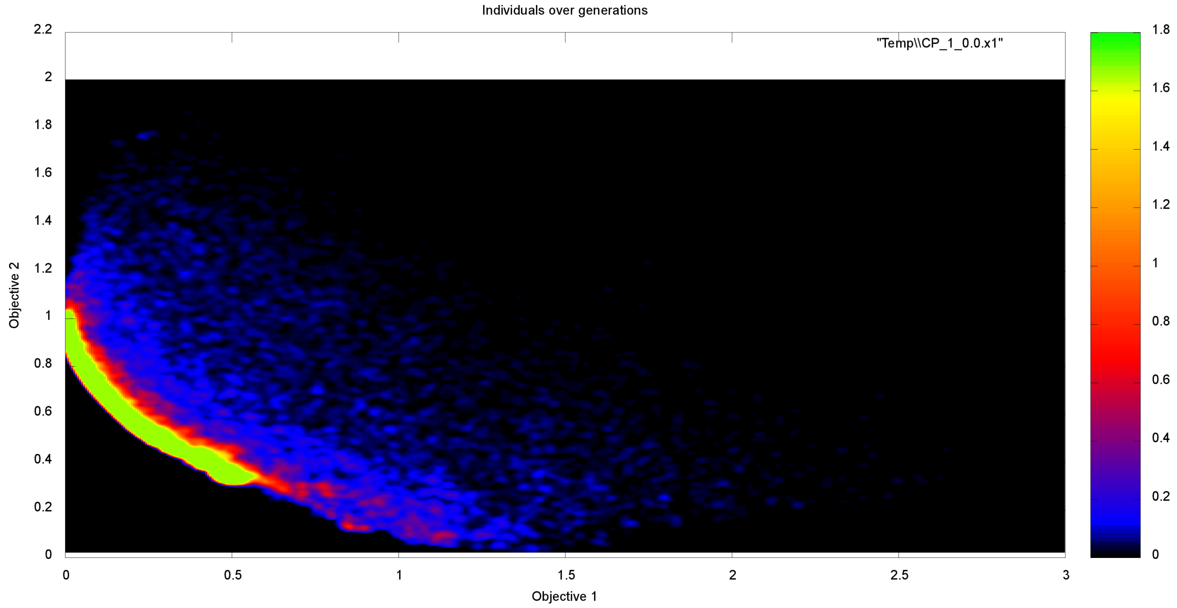


FIGURE G.29: The heat map of the MLS1 type of collective in MLSGA_U-NSGA-III algorithm on UF2 problem.

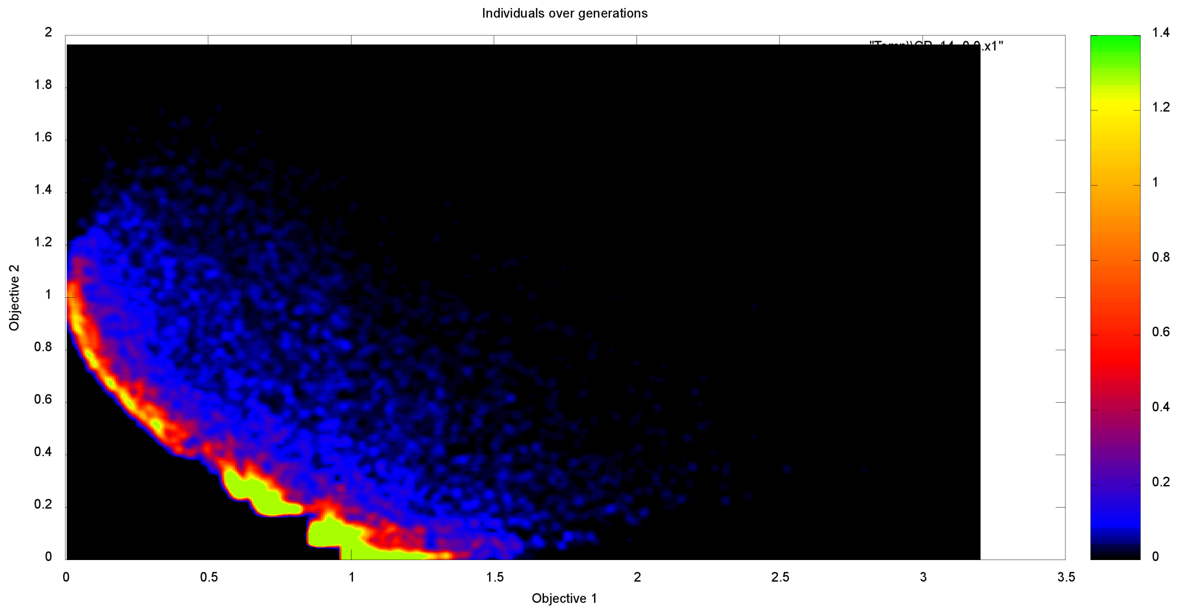


FIGURE G.30: The heat map of the MLS2 type of collective in MLSGA_U-NSGA-III algorithm on UF2 problem.

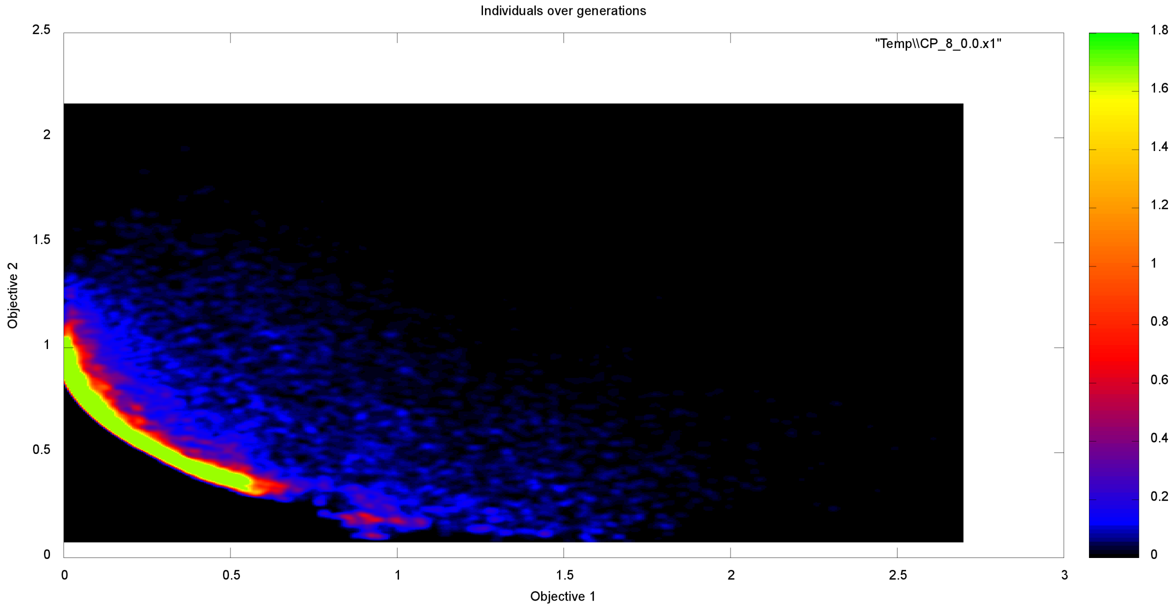


FIGURE G.31: The heat map of the MLS2R type of collective in MLSGA_U-NSGA-III algorithm on UF2 problem.

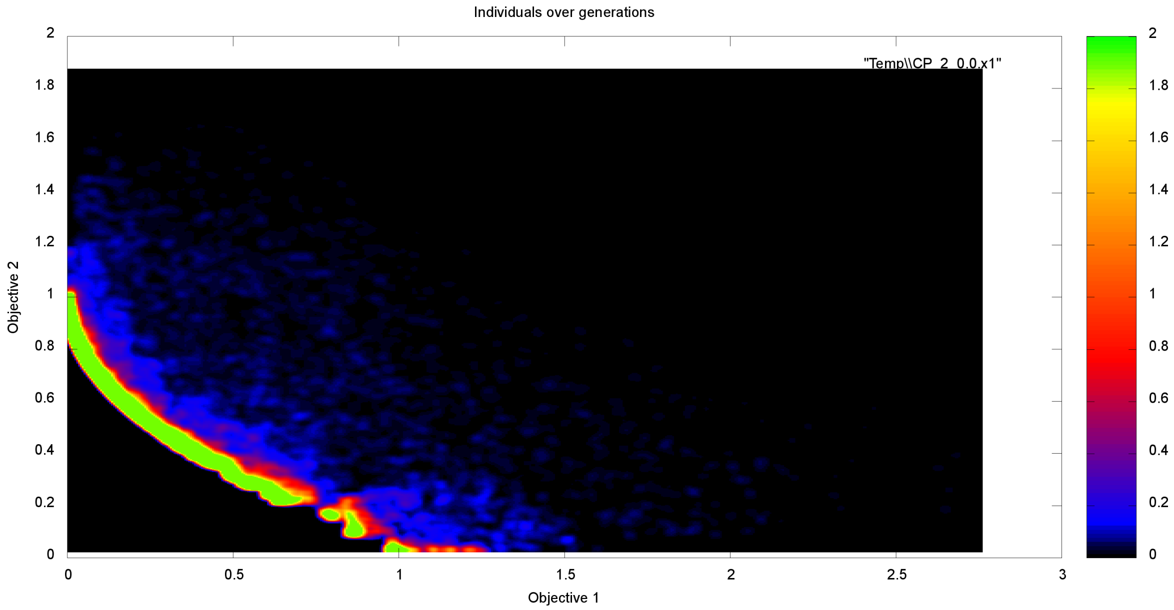


FIGURE G.32: The heat map of the original U-NSGA-III algorithm on UF2 problem.

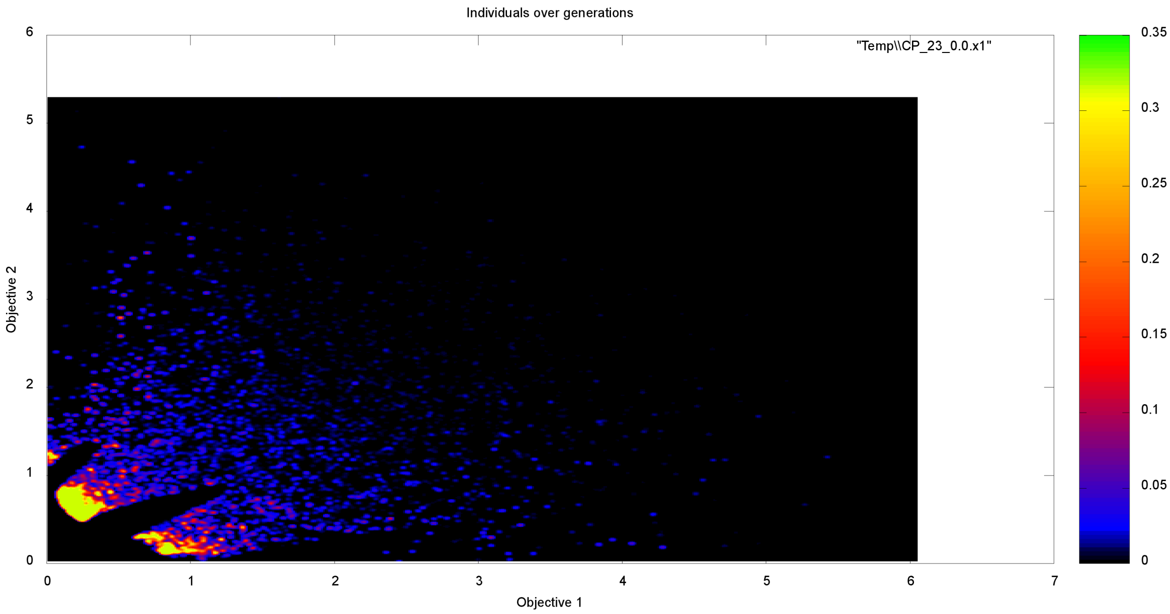


FIGURE G.33: The heat map of the MLS1 type of collective in MLSGA_MOEA/D algorithm on CF2 problem.

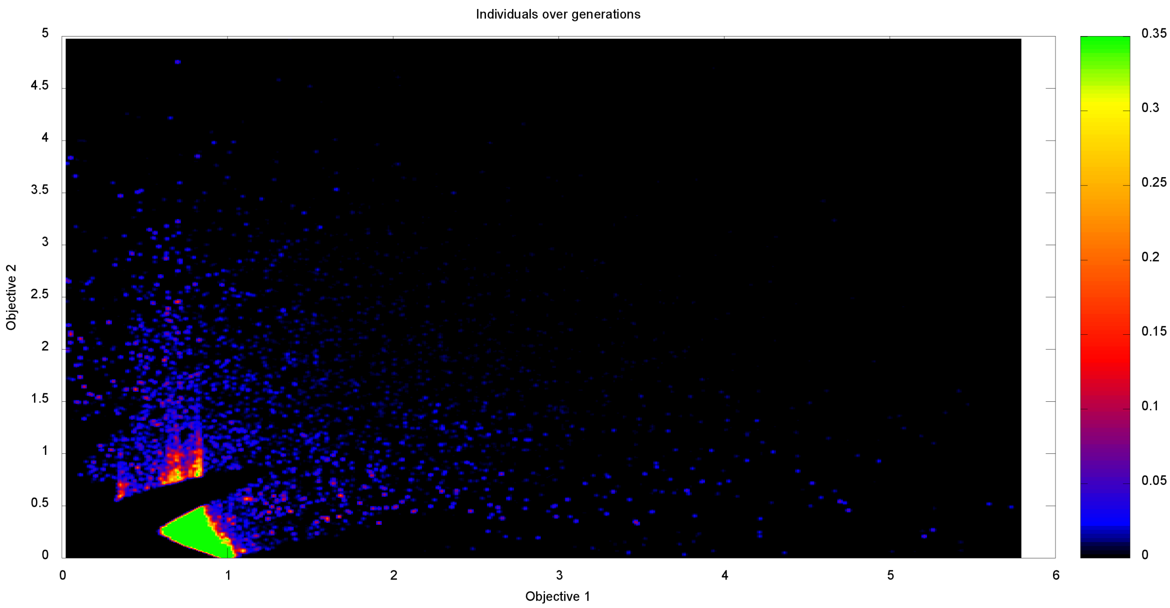


FIGURE G.34: The heat map of the MLS2 type of collective in MLSGA_MOEA/D algorithm on CF2 problem.

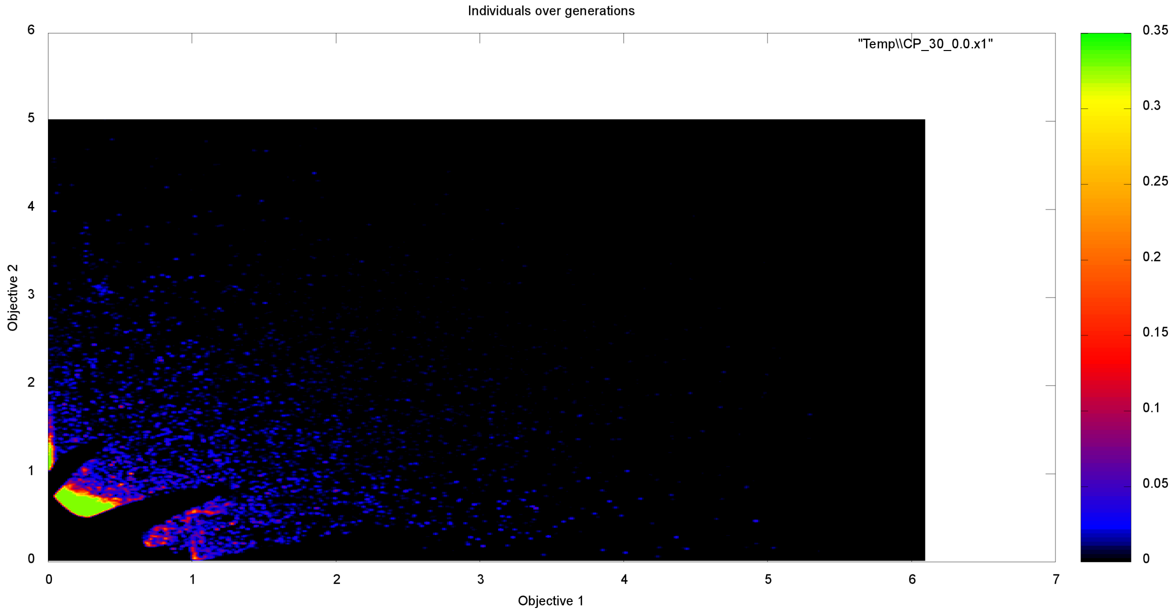


FIGURE G.35: The heat map of the MLS2R type of collective in MLSGA_MOEA/D algorithm on CF2 problem.

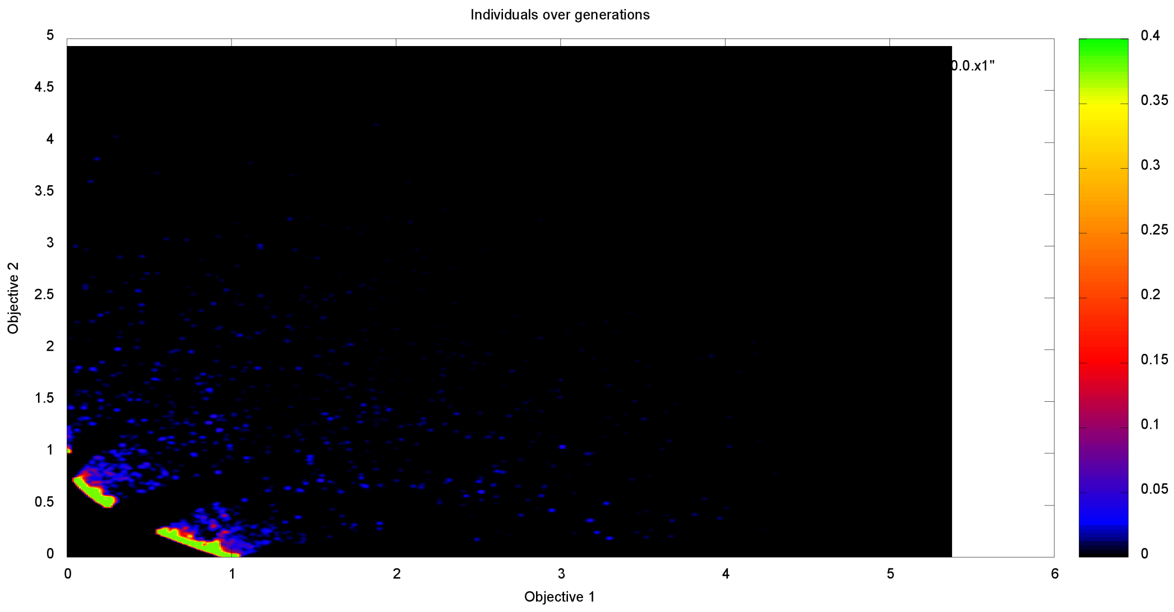


FIGURE G.36: The heat map of the original MOEA/D algorithm on CF2 problem.

Bibliography

[1] G. Hornby, A. Globus, D. Linden, and J. Lohn, "Automated antenna design with evolutionary algorithms," *Space 2006*, vol. 5, no. September, pp. 1–8, 2006. [Online]. Available: http://hdl.handle.net/2060/20060024675{%}5Cnhttp://arc.aiaa.org/doi/10.2514/6.2006-7242

[2] M. Szczepański, "Economic impacts of artificial intelligence (AI)," *European Parliamentary Research Service*, no. July, 2019.

[3] C. Darwin, *On the origin of species by means of natural selection, or, The preservation of favoured races in the struggle for life*. London: John Murray, 1859.

[4] C. A. Coello Coello, "Statistics of the EMOO repository," 2017. [Online]. Available: http://delta.cs.cinvestav.mx/{~}ccoello/EMOO/EMOOstatistics.html

[5] L. Shen, H. Chen, Z. Yu, W. Kang, B. Zhang, H. Li, B. Yang, and D. Liu, "Evolving support vector machines using fruit fly optimization for medical data classification," *Knowledge-Based Systems*, vol. 96, pp. 61–75, 2016.

[6] S. R. Dash, S. Dehuri, and S. Rayaguru, "Discovering interesting rules from biological data using parallel genetic algorithm," *Proceedings of the 2013 3rd IEEE International Advance Computing Conference, IACC 2013*, pp. 631–636, 2013.

[7] Y. Li and J. Lei, "A feasible solution to the beam-angle-optimization problem in radiotherapy planning with a DNA-based genetic algorithm," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 3, pp. 499–508, 2010.

[8] U. Kamath, J. Compton, R. Islamaj-Doğan, K. A. De Jong, and A. Shehu, "An evolutionary algorithm approach for feature generation from sequence data and its application to DNA splice site prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 5, pp. 1387–1398, 2012.

[9] L. Y. Wei, "A GA-weighted ANFIS model based on multiple stock market volatility causality for TAIEX forecasting," *Applied Soft Computing Journal*, vol. 13, no. 2, pp. 911–920, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2012.08.048

[10] A.-h. Kakaee, P. Rahnama, A. Paykani, and B. Mashadi, "Combining artificial neural network and multi-objective optimization to reduce a heavy-duty diesel engine emissions and fuel consumption," *J. Cent. South Univ.*, vol. 22, pp. 4235–4245, 2015.

[11] D. Golmohammadi, R. C. Creese, H. Valian, and J. Kolassa, "Supplier selection based on a neural network model using genetic algorithm," *IEEE TRANSACTIONS ON NEURAL NETWORKS*, vol. 20, no. 2, pp. 1504–1519, 2009.

[12] S. Nguyen, M. Zhang, M. Johnston, and K. C. Tan, "Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 193–208, 2014.

[13] A. Turing, "Computing machinery and intelligence," *MIND*, vol. LIX, no. 236, pp. 433–460, 1950.

[14] J. H. Holland, "Outline for a logical theory of adaptive systems," *Journal of the ACM*, vol. 9, no. 3, pp. 297–314, 1962.

[15] N. A. Barricelli, "Numerical testing of evolution theories," *Acta Biotheoretica*, vol. 16, no. 3-4, pp. 99–126, 1963.

[16] J. H. Holland, "Adaptive plans optimal for payoff-only environments," in *Proceedings of the Second Hawaii Conference on Systems Sciences*, 1969.

[17] ——, "A new kind of turnpike theorem," The University of Michigan, Tech. Rep., 1969.

[18] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pp. 93–100, 1985. [Online]. Available: http://dl.acm.org/citation.cfm?id=657079

[19] D. E. Goldberg, "Genetic algorithm in search, optimization, and machine learning," *Addison-Wesley, Reading, Massachusetts*, vol. XIII, no. 1, 1989.

[20] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: formulation, discussion and generalization," in *The fifth International conference on Genetic Algorithms*, vol. 93, no. July, 1993, pp. 416–423.

[21] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.

[22] K. Deb and D. E. Goldberg, "An investigation of niche and species formation in genetic function optimization," pp. 42–50, 1991.

[23] K. Deb and R. Bhushan Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, pp. 115–148, 1995. [Online]. Available: https://pdfs.semanticscholar.org/b8ee/6b68520ae0291075cb1408046a7dff9dd9ad.pdf

[24] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, vol. 1, pp. 82–87, 1994.

[25] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[26] W. Stadler, "A survey of multicriteria optimization or the vector maximum problem, part I: 1776-1960," *Journal of Optimization Theory and Applications*, vol. 29, no. 1, pp. 1–52, 1979.

[27] J. Branke, H. Schmeck, K. Deb, and M. Reddy.S, "Parallelizing multi-objective evolutionary algorithms: cone separation," *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, vol. 2, pp. 1952–1957, 2004. [Online]. Available: http://ieeexplore.ieee.org/document/1331135/http://repository.ias.ac.in/81667/

[28] M. A. Potter and K. A. De Jong, "A cooperative coevolutionary approach to function optimization," *The Third Parallel Problem Solving From Nature*, pp. 249–257, 1994. [Online]. Available: http://link.springer.com/10.1007/3-540-58484-6{_}269

[29] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, pp. 32–49, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.swevo.2011.03.001

[30] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[31] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, and W. Liu, "Multiobjective optimization test instances for the CEC 2009 special session and competition," Tech. Rep., 2009.

[32] H.-l. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.

[33] H.-L. Liu, L. Chen, K. Deb, and E. Goodman, "Investigating the effect of imbalance between convergence and diversity in evolutionary multiobjective algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 408–425, 2017.

[34] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[35] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[36] Q. Lin, J. Chen, Z.-H. Zhan, W.-N. Chen, C. A. Coello Coello, Y. Yin, C.-M. Lin, and J. Zhang, "A hybrid evolutionary immune algorithm for multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 711–729, 2016.

[37] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6600851{%}5Cnhttp://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6600851

[38] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using gaussian process based inverse modeling," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 838–856, 2015.

[39] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Adaptive replacement strategies for MOEA/D," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 474–486, 2016.

[40] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Tech. Rep. 1990, 2001. [Online]. Available: http://www.springerlink.com/content/q404757t4q25m64l/{%}5Cnhttp://www.springerlink.com/index/q404757t4q25m64l.pdf

[41] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Evolutionary Multi-Criterion Optimization*, no. June, 2005, pp. 280–295. [Online]. Available: http://link.springer.com/10.1007/978-3-540-31880-4{_}20

[42] Q. Zhang and P. N. Suganthan, "Final report on CEC ' 09 MOEA competition," Tech. Rep., 2009. [Online]. Available: http://dces.essex.ac.uk/staff/zhang/moeacompetition09.htm

[43] H. Seada and K. Deb, "U-NSGA-III: A unified evolutionary algorithm for single, multiple, and many-objective optimisation," in *Evolutionary Multi-Criterion Optimization: EMO 2015.*, A. Gaspar-Cunha, C. Henggeler Antunes, and C. C. Coello, Eds. Cham: Springer International Publishing, 2015, pp. 1–30. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-15892-1{_}3

[44] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Difficulty adjustable and scalable constrained multiobjective test problem toolkit," *Evolutionary Computation*, no. x, pp. 1–40, 2019.

[45] A. J. Sobey and P. A. Grudniewski, "Re-inspiring the genetic algorithm with multi-level selection theory: Multi-level selection genetic algorithm," *Bioinspiration and Biomimetics*, vol. 13, no. 5, pp. 1–13, 2018.

[46] D. Whitley, S. Rana, and R. B. Heckendorn, "The island model genetic algorithm: on separability, population size and convergence," *Journal of Computing and Information Technology*, vol. 7, pp. 33–47, 1998. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.36.7225{&}rep=rep1{&}type=pdf

[47] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 440–462, 2017.

[48] J. H. Holland, "Genetic algorithms and the optimal allocation of trials," *SIAM Journal on Computing*, vol. 2, no. 2, pp. 88–105, 1973. [Online]. Available: http://dx.doi.org/10.1137/0202009

[49] I. Rechenberg, "Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution," Ph.D. dissertation, 1971.

[50] H.-P. Schwefel, "Numerische Optimierung von Computer-Modellen," Ph.D. dissertation, 1974.

[51] J. L. Bosworth, N. Y. Foo, and B. P. Zeigler, "Comparison of genetic algorithms with conjugate gradient methods," *NASA Contractor Reports*, no. August, p. 44, 1972.

[52] K. A. De Jong, "Analysis of the behaviour of a class of genetic adaptive systems," The University of Michigan, Tech. Rep. 185, 1975.

[53] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.

[54] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," Tech. Rep., 2001.

[55] D. W. Corne, J. D. Knowles, and M. J. Oates, "The Pareto envelope-based selection algorithm for multiobjective optimization," *Parallel Problem Solving from Nature*, no. Mcdm, pp. 839–848, 2000. [Online]. Available: http://www.springerlink.com/index/q576765808l68p34.pdf

[56] D. W. Corne, N. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2001, pp. 283–290. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.2194

[57] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.

[58] R. Hinterding, Z. Michalewicz, and T. C. Peachey, "Self-adaptive genetic algorithm for numeric functions," *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, vol. 1141, pp. 420–429, 1996. [Online]. Available: http://www.springerlink.com/content/f221963607h2554p/

[59] M. Kurdi, "An effective new island model genetic algorithm for job shop scheduling problem," *Computers and Operations Research*, vol. 67, pp. 132–142, 2016. [Online]. Available: http://dx.doi.org/10.1016/j.cor.2015.10.005

[60] C. A. Coello Coello and G. Toscano Pulido, "A micro-genetic algorithm for multiobjective optimization," in *Evolutionary Multi-Criterion Optimization*, 2001, pp. 126–140.

[61] S. Tiwari, G. M. Fadel, P. Koch, and K. Deb, "AMGA: An archive-based micro genetic algorithm for multi-objective optimisation," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2008, pp. 729–736.

[62] Z. Wang, J. Bai, A. J. Sobey, J. Xiong, and A. Shenoi, "Optimal design of triaxial weave fabric composites under tension," *Composite Structures*, vol. 201, no. June, pp. 616–624, 2018. [Online]. Available: https://doi.org/10.1016/j.compstruct.2018.06.090

[63] X. N. Shen and X. Yao, "Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems," *Information Sciences*, vol. 298, no. 219, pp. 198–224, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2014.11.036

[64] G. N. Demir, A. S. Uyar, and S. Gunduz-Oguducu, "Multiobjective evolutionary clustering of Web user sessions: A case study in Web page recommendation," *Soft Computing*, vol. 14, no. 6, pp. 579–597, 2010.

[65] J. G. Iniestra and J. G. Gutiérrez, "Multicriteria decisions on interdependent infrastructure transportation projects using an evolutionary-based framework," *Applied Soft Computing Journal*, vol. 9, no. 2, pp. 512–526, 2009.

[66] R. Saravanan, S. Ramabalan, N. G. R. Ebenezer, and C. Dharmaraja, "Evolutionary multi criteria design optimization of robot grippers," *Applied Soft Computing Journal*, vol. 9, no. 1, pp. 159–172, 2009.

[67] S.-y. Shin, "Multi-objective evolutionary optimization of DNA sequences for molecular computing," *Design*, vol. 9, no. August, pp. 143–158, 2005.

[68] M. A. Panduro, C. A. Brizuela, D. Covarrubias, and C. Lopez, "A trade-off curve computation for linear antenna arrays using an evolutionary multi-objective approach," *Soft Computing*, vol. 10, no. 2, pp. 125–131, 2006.

[69] C.-K. Ting, C.-N. Lee, H.-C. Chang, and J.-S. Wu, "Using multiobjective variable-length genetic algorithm," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 39, no. 4, pp. 945–958, 2009.

[70] S. Huband, P. Hingston, L. Barone, and L. While, “A review of multi-objective test problems and a scalable test problem toolkit,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006. [Online]. Available: http://ro.ecu.edu.au/ecuworks/2022

[71] K. Deb, U. B. Rao N., and S. Karthik, “Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling,” in *Evolutionary Multi-Criterion Optimization*, 2007, pp. 803–817.

[72] A. Lancinskas and J. Zilinskas, “Solution of multi-objective competitive facility location problems using parallel NSGA-II on large scale computing systems,” *Applied Parallel and Scientific Computing. PARA 2012. Lecture Notes in Computer Science*, vol. 7782, pp. 422–433, 2013.

[73] K. Deb and A. Sinha, “Solving bilevel multi-objective optimization problems using evolutionary algorithms,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5467 LNCS, pp. 110–124, 2010.

[74] H. Jain and K. Deb, “An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part II: Handling constraints and extending to an adaptive approach,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[75] H. Li, K. Deb, Q. Zhang, P. N. Suganthan, and L. Chen, “Comparison between MOEA/D and NSGA-III on a set of many and multi-objective benchmark problems with challenging difficulties,” *Swarm and Evolutionary Computation*, 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2210650218307016

[76] W. Abdou, C. Bloch, D. Charlet, and F. Spies, “Multi-pareto-ranking evolutionary algorithm,” *EvoCOP*, vol. 7245, pp. 194–205, 2012.

[77] W. Abdou, D. Charlet, C. Bloch, and F. Spies, “Adaptive multi-objective genetic algorithm using multi-pareto-ranking categories and subject descriptors,” in *GECCO’12, Philadelphia, USA*, 2012, pp. 449–456.

[78] K. Miettinen, *Nonlinear multiobjective optimization.* Kluwer Academic Publishers, 1998, vol. 12. [Online]. Available: http://link.springer.com/10.1007/978-1-4615-5563-6

[79] Y.-Y. Tan, Y.-C. Jiao, H. Li, and X.-K. Wang, “MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives,” *Computers and Operations Research*, vol. 40, no. 6, pp. 1648–1660, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.cor.2012.01.001

[80] X. Ma, Y. Qi, L. Li, F. Liu, L. Jiao, and J. Wu, “MOEA/D with uniform decomposition measurement for many-objective problems,” *Soft Computing*, vol. 18, no. 12, pp. 2541–2564, 2014.

[81] L. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 475–480, 2016.

[82] H.-L. Liu, L. Chen, Q. Zhang, and K. Deb, "An evolutionary many-objective optimisation algorithm with adaptive region decomposition," *2016 IEEE Congress on Evolutionary Computation (CEC 2016)*, no. 2016014, pp. 4763–4769, 2016.

[83] S. Jiang, S. Yang, Y. Wang, and X. Liu, "Scalarizing functions in decomposition-based multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 296–313, 2018.

[84] X. Ma, Q. Zhang, G. Tian, J. Yang, and Z. Zhu, "On Tchebycheff decomposition approaches for multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 226–244, 2018.

[85] Z. Wang, Q. Zhang, M. Gong, and A. Zhou, "A replacement strategy for balancing convergence and diversity in MOEA/D," *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, pp. 2132–2139, 2014.

[86] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[87] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and ant colony," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1845–1859, 2013.

[88] C. M. Chen, Y. P. Chen, and Q. Zhang, "Enhancing MOEA/D with guided mutation and priority update for multi-objective optimization," *2009 IEEE Congress on Evolutionary Computation (CEC 2009)*, pp. 209–216, 2009.

[89] H.-L. Liu and X. Li, "The multiobjective evolutionary algorithm based on determined weight and sub-regional search," *2009 IEEE Congress on Evolutionary Computation (CEC 2009)*, pp. 1928–1934, 2009.

[90] M. Liu, X. Zou, C. Yu, and Z. Wu, "Performance assessment of DMOEA-DD with CEC 2009 MOEA competition test instances," *2009 IEEE Congress on Evolutionary Computation (CEC 2009)*, no. 1, pp. 2913–2918, 2009.

[91] X. Zou, M. Liu, L. Kang, and J. He, "A high performance multi-objective evolutionary," *Parallel Problem Solving from Nature*, pp. 922–931, 2004.

[92] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[93] L.-Y. Tseng and C. Chen, "Multiple trajectory search for multiobjective optimization," in *2007 IEEE Congress on Evolutionary Computation (CEC 2007)*, 2007, pp. 3609–3616.

[94] A. Elhossini, S. Areibi, and R. Dony, "Strength pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization," *Evolutionary Computation*, vol. 18, no. 1, pp. 127–156, 2010.

[95] B. B. Li and L. Wang, "A hybrid quantum-inspired genetic algorithm for multiobjective flow shop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 3, pp. 576–591, 2007.

[96] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 28, no. 3, pp. 392–403, 1998.

[97] K. Sindhya, A. Sinha, K. Deb, and K. Miettinen, "Local search based evolutionary multi-objective optimization algorithm for constrained and unconstrained problems," in *2009 IEEE Congress on Evolutionary Computation (CEC 2009)*, 2009, pp. 2919–2926.

[98] J. Hill, "The three C's - competition, coexistence and coevolution - and their impact on the breeding of forage crop mixtures," *Theoretical and Applied Genetics*, vol. 79, no. 2, pp. 168–176, 1990.

[99] C.-K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, 2009.

[100] C. D. Rosin and R. K. Belew, "Methods for competitive co-evolution," *Evolutionary Computation*, vol. 5, no. 1, pp. 1–29, 1997. [Online]. Available: http://www.mitpressjournals.org/doi/abs/10.1162/evco.1997.5.1.1

[101] K. A. De Jong, *Evolutionary computation : a unified approach.* Cambridge, Mass., London: MIT, 2006.

[102] Y.-H. Jia, W.-N. Chen, T. Gu, H. Zhang, H. Yuan, S. Kwong, and J. Zhang, "Distributed cooperative co-evolution with adaptive computing resource allocation for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. In press, pp. 1–15, 2018.

[103] M. Li, S. Yang, and X. Liu, "Pareto or non-Pareto: Bi-criterion evolution in multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 645–665, 2016.

[104] S. Jiang and S. Yang, "Evolutionary dynamic multiobjective optimization: Benchmarks and algorithm comparisons," *IEEE Transactions on Cybernetics*, vol. 47, no. 1, pp. 198–211, jan 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7381632/

[105] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point," in *Foundations of Genetic Algorithms (FOGA 2009)*, 2009, pp. 1–17.

[106] D. S. Wilson and E. Sober, "Reintroducing group selection to the human behavioral sciences," *Behavioral and Brain Sciences*, vol. 136, no. 4, pp. 337–356, 1989.

[107] S. Okasha, *Evolution and the levels of selection.* Oxford University Press, 2006.

[108] T. Lenaerts, A. Defaweux, P. V. Remortel, and B. Manderick, "Modeling artificial multi-level selection," Tech. Rep., 2003.

[109] T. Y. Lim, "Structured population genetic algorithms: A literature survey," *Artificial Intelligence Review*, vol. 41, no. 3, pp. 385–399, 2014.

[110] R. Akbari, V. Zeighami, and K. Ziarati, "MLGA: A multilevel cooperative genetic algorithm," *Proceedings 2010 IEEE 5th International Conference on Bio-Inspired Computing*, pp. 271–277, 2010.

[111] R. Akbari and K. Ziarati, "A multilevel evolutionary algorithm for optimizing numerical functions," *International Journal of Industrial Engineering Computations*, vol. 2, no. 2, pp. 419–430, 2011.

[112] S. X. Wu and W. Banzhaf, "A hierarchical cooperative evolutionary algorithm," in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, 2010, pp. 233–240.

[113] J. Lehman and K. O. Stanley, "Exploiting open-endedness to solve problems through the search for novelty," *Artificial Life XI: Proceedings of the 11th International Conference on the Simulation and Synthesis of Living Systems, ALIFE 2008*, pp. 329–336, 2008.

[114] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002. [Online]. Available: https://www.cs.ucf.edu/{~}kstanley/neat.html{%}0Ahttp://eplex.cs.ucf.edu/hyperNEATpage/HyperNEAT.html{%}0Ahttps://github.com/crisbodnar/TensorFlow-NEAT{%}0Ahttps://github.com/uber-research/PyTorch-NEAT{%}0Ahttps://github.com/CodeReclaimers/neat-python

[115] J. Lehman and K. O. Stanley, "Efficiently evolving programs through the search for novelty," *Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference, GECCO '10*, no. Gecco, pp. 837–844, 2010.

[116] ——, "Evolving a diversity of virtual creatures through novelty search and local comp," *Proc. of 13th Genetic & Evol. Comp. Conf*, no. Gecco, pp. 211–218, 2011. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.365.4807{&}rep=rep1{&}type=pdf

[117] ——, "Abandoning objectives: Evolution through the search for novelty alone," *Evolutionary Computation*, vol. 19, no. 2, pp. 189–222, 2011.

[118] S. Risi, S. D. Vanderbleek, C. E. Hughes, and K. O. Stanley, "How novelty search escapes the deceptive trap of learning to learn," *Proceedings of the 11th Annual Genetic and Evolutionary Computation Conference, GECCO-2009*, pp. 153–160, 2009.

[119] J.-b. Mouret, "Novelty-Based Multiobjectivization," in *New Horizons in Evolutionary Robotics*, 2011, pp. 193–154.

[120] H. Shahrzad, D. Fink, and R. Miikkulainen, "Enhanced optimization with composite objectives and novelty selection," *ALIFE 2018 - 2018 Conference on Artificial Life: Beyond AI*, pp. 616–622, 2020.

[121] E. Zitzler and K. Simon, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature*, 2004, pp. 832–842.

[122] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering and System Safety*, vol. 91, no. 9, pp. 992–1007, 2006.

[123] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary algorithms for solving multi-objective problems.* Kluwer Academic Publishers, 2002.

[124] P. A. Bosman and D. Thierens, "The balance between proximity and diversity in multi-objective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.

[125] M. Helbig and A. P. Engelbrecht, "Benchmarks for dynamic multi-objective optimisation algorithms," *ACM Computational Surveys*, vol. 46, no. 3, pp. 37:1–37:39, 2014.

[126] J. Mehnen, T. Wagner, and G. Rudolph, "Evolutionary optimization of dynamic multi-objective test functions," Tech. Rep. May, 2006. [Online]. Available: http://laral.istc.cnr.it/wiva3/atti/gsice2/papers/GSICE06{_}09{_}Wagner.pdf

[127] X. Li, K. Tang, M. N. Omidvar, Z. Yang, and K. Qin, "Benchmark functions for the CEC'2013 special session and competition on large-scale global optimization," in *2013 IEEE Congress on Evolutionary Computation (CEC 2013)*, 2013, pp. 1–23.

[128] A. Sinha, P. Malo, and K. Deb, "Test problem construction for single-objective bilevel optimization," *Evolutionary Computation*, vol. 22, no. 3, pp. 439–477, 2014.

[129] P. Fattahi, M. Saidi Mehrabad, and F. Jolai, "Mathematical modeling and heuristic approaches to flexible job shop scheduling problems," *Journal of Intelligent Manufacturing*, vol. 18, no. 3, pp. 331–342, 2007.

[130] S. Biswas, S. Das, P. N. Suganthan, and C. A. Coello Coello, "Evolutionary multiobjective optimization in dynamic environments: A set of novel benchmark functions," in *2014 IEEE Congress on Evolutionary Computation (CEC 2014)*, vol. 1, 2014, pp. 3192–3199.

[131] M. Farina, K. Deb, and P. Amato, “Dynamic multiobjective optimization problems: Test cases, approximations, and applications,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 5, pp. 425–442, 2004.

[132] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, “Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization,” in *Evolutionary Multi-Criterion Optimization*. Springer Berlin Heidelberg, 2007, pp. 832–846.

[133] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, “Constraint handling in multi-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 514–525, 2009.

[134] Q. Liu, X. Liu, J. Wu, and Y. Li, “An improved NSGA-III algorithm using genetic k-means clustering algorithm,” *IEEE Access*, vol. 7, pp. 185 239–185 249, 2019.

[135] A. J. Sobey and S. Paksuttipol. private communication, November 2015.

[136] R. Dawkins, *The selfish gene*. Oxford: OUP, 1989.

[137] K. Sterelny and P. Kitcher, “The return of the gene,” *Journal of Philosophy*, vol. 85, no. 7, pp. 339–361, 1988.

[138] C.-C. Chang and C.-J. Lin, “Libsvm: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?doid=1961189.1961199

[139] M. Li and X. I. N. Yao, “Quality evaluation of solution sets in multiobjective optimisation : A survey,” *ACM Computational Surveys*, vol. 26, pp. 1–43, 2019.

[140] E. Zitzler, L. Thiele, and M. Laumanns, “Performance assessment of multiobjective optimizers: An analysis and review,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2002. [Online]. Available: http://www.tik.ee.ethz.ch/education/lectures/DSE/ZLTF03.pdf

[141] L. While, L. Bradstreet, and L. Barone, “A fast way of calculating exact hypervolumes,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 86–95, 2012.

[142] R. Azzouz, S. Bechikh, L. B. Said, and W. Trabelsi, “Handling time-varying constraints and objectives in dynamic evolutionary multi-objective optimization,” *Swarm and Evolutionary Computation*, vol. 39, no. April 2017, pp. 222–248, 2018. [Online]. Available: http://dx.doi.org/10.1016/j.swevo.2017.10.005

[143] C. Huang, Y. Li, and X. Yao, “A Survey of Automatic Parameter Tuning Methods for Metaheuristics,” *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 2, pp. 201–216, 2020.

[144] S. Yang, M. Li, X. Liu, and J. Zheng, “A grid-based evolutionary algorithm for many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[145] A. J. Sobey and P. A. Grudniewski, “There’s no free lunch: A study of genetic algorithm use in maritime applications,” in *18th Conference on Computer Applications and Information Technology in the Maritime Industries (COMPIT)*, 2019.

[146] J. M. Blanchard, U. Mutlu, A. J. Sobey, and J. I. Blake, “Modelling the different mechanical response and increased stresses exhibited by structures made from natural fibre composites,” *Composite Structures*, vol. 215, no. December 2018, pp. 402–410, 2019. [Online]. Available: https://doi.org/10.1016/j.compstruct.2019.02.042

[147] J. W. Steer, P. A. Grudniewski, M. Browne, P. R. Worsley, A. J. Sobey, and A. S. Dickinson, “Predictive prosthetic socket design: part 2—generating person-specific candidate designs using multi-objective genetic algorithms,” *Biomechanics and Modeling in Mechanobiology*, pp. 1–14, 2019. [Online]. Available: http://link.springer.com/10.1007/s10237-019-01258-7

[148] N. A. Al-Madi and A. T. Khader, “De Jong ’s sphere model test for a social-based genetic algorithm (SBGA),” *International Journal of Computer Science and Network Security*, vol. 8, no. 3, pp. 179–185, 2008.

[149] R. Lahoz-Beltra, G. Ochoa, and U. Aickelin, “Cheating for problem solving: A genetic algorithm with social interactions,” in *Proceedings of the 11th annual conference on genetic and evolutionary computation (GECCO ’09)*, 2009, pp. 811–817.

[150] N. A. Al-Madi, “De Jong’s sphere model test for a human community based genetic algorithm model (HCBGA),” *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 1, pp. 166–172, 2014.

[151] E. Alba, H. Alfonso, and B. Dorronsoro, “Advanced models of cellular genetic algorithms evaluated on SAT,” in *GECCO’05*, 2005, pp. 1123–1130.

[152] R. K. Ursem, “Multinational evolutionary algorithms,” in *Proceedings of the 1999 Congress on Evolutionary Computation (CEC’99)*, no. July 1999, 2013, pp. 1633–1640. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=785470

[153] R. Leardi, “Genetic algorithms in chemometrics and chemistry: A review,” *Journal of Chemometrics*, vol. 15, no. 7, pp. 559–569, 2001.

[154] R. L. Johnston, “Evolving better nanoparticles: Genetic algorithms for optimising cluster geometries,” *Journal of the Chemical Society. Dalton Transactions*, vol. 3, no. 22, pp. 4193–4207, 2003.

[155] R. Leardi, “Genetic algorithms in chemistry,” *Journal of Chromatography A*, vol. 1158, no. 1-2, pp. 226–233, 2007.

[156] E. Ramadan, A. Naef, and M. Ahmed, “Protein complexes predictions within protein interaction networks using genetic algorithms,” *BMC Bioinformatics*, vol. 17, no. Suppl 7, 2016.

[157] M. H. Tayarani-N. and A. Prügel-Bennett, "Anatomy of the fitness landscape for dense graph-colouring problem," *Swarm and Evolutionary Computation*, vol. 22, pp. 47–65, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.swevo.2015.01.005

[158] M. H. Tayarani-N. and A. Prugel-Bennett, "An analysis of the fitness landscape of travelling salesman problem," *Evolutionary Computation*, vol. 24, no. 2, pp. 347–384, 2016.

[159] U. Mutlu, P. A. Grudniewski, A. J. Sobey, and J. I. R. Blake, "Selecting an optimisation methodology in the context of structural design for leisure boats," *Design & Construction of Super & Mega Yachts*, no. May, pp. 10–11, 2017.