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

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 CEC09 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 CEC09 problem sets.

☆This research was supported by Lloyd Register Foundation

*Corresponding author

Email addresses: pag1c15@soton.ac.uk (Przemyslaw A. Grudniewski), ajs502@soton.ac.uk (Adam J. Sobey)

Keywords: Evolutionary computation, genetic algorithms, hybrid GA, multi-level selection, multi-objective optimisation

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 [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.

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 specialization 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 (CEC09) [5]. Based on the previous results it is proposed that MLSGAs 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 [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 [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 CEC09 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 CEC09 [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 se-

Table 1: Fitness functions assignment for different MLS types.

Fitness definition type	MLS2	MLS2R	MLS1
Collective level	$f_2(x)$	$f_1(x)$	$f_1(x) + f_2(x)$
Individual level	*both* $f_1(x)$ *and* $f_2(x)$		

lection 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 CEC09 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 MOEA/D 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, $x_j,...,x_{Np}$.

Step 2) Classification:

Step 2.1) Classify the individuals in the initial population *P* into N_c collectives, $C_i,...,C_{Nc}$, so that each contains a separate population $P_i,...,P_{Nc}$. 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 λ_i to each individual in each corresponding collective.

Step 3) Individual level operations:

For i = 1,...,N_c do

Step 3.1) Individual level GAs 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 collectives entire population P_i, documented in the corresponding literature.,

Step 3.2) Update External Population:

For j = 1,...,—P_i— do

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

Step 4) Collective level operations:

Step 4.1) Calculate collective fitness:

For i = 1,...,N_c do

Calculate the fitness of the collective C_i as the average of the fitnesses of populationP_i based on the fitness definition assigned to that collective.

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.

In the case of the MOEA/D hybrid: Store the weight vectors of population $\{\lambda_j,..., \lambda_{size(Pz)}\}$ P_z as the matrix Λ_z.

Erase the collective C_z with population P_z.

Step 4.3) Collective reproduction:

For i = 1,...,N_c do

if (i != z)

Copy the best $s/(N_c - 1)$ individuals, according to the fitness definition of the eliminated collective C_z, from population P_i to P_z.

then

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 is met, stop and give EP as output. Otherwise, return to **Step 3)**.

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, $f_1(x)$, 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 $f_2(x)$ 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 [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 [9] defined in eq. 1, is recommended:

$$\bar{f}_i = \frac{f_i - z_i^*}{z_i^{nad} - z_i^*} \qquad (1)$$

where $z^* = (z_1^*, \ldots, z_m^*)$ is the reference point, i.e. $z_i^* = min f_i(x)|x \in \omega$, $z^{nad} = (z_1^{nad}, \ldots, z_m^{nad})^T$ is the nadir point, i.e. $z_i^{nad} = max f_i(x)|x \in \omega$, $\omega$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 f_1 and f_2 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 [22, 18]. 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 [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 CEC09 [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 CEC09 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 CEC09 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 CEC09 test set for the WFG test instances. Therefore, only the CEC09 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 CEC09 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;

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

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 $d(\nu, A)$ is the minimum Euclidean distance between point ν 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 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

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 f1 and f2		Both f1 and f2 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		

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 Figure 1, MOEA/D in Figure 2 and MTS in Figure 3, and for the CF5 illustrated in Figure 4, Figure 5 and Figure 6 for NSGA-II, MOEA/D and MTS

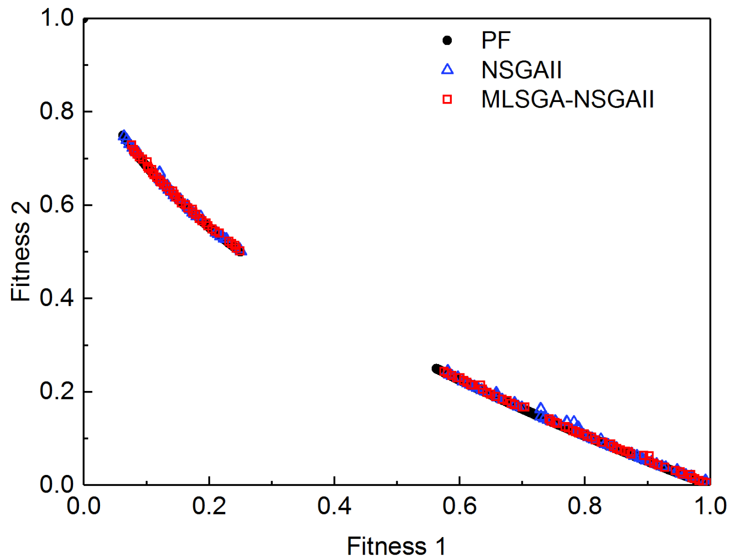

Figure 1: Pareto Front of NSGA-II and MLSGA-NSGA-II on the CF2.

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 Figure 7 for NSGA-II, Figure 8 for MOEA/D and Figure 9 for MTS, and UF5 is shown in Figure 10, Figure 11 and Figure 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 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 Wilcoxons 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 CEC09 benchmarking [17]. As MLSGA-NSGA-II is based on an updated version of NSGA-II, not on the version from the CEC09 ranking, the updated results for this algorithm are included in Table 3 and Table 4. As the results for MOEA/D on the constrained functions have not been presented in CEC09, these are simulated and included in Table 3.

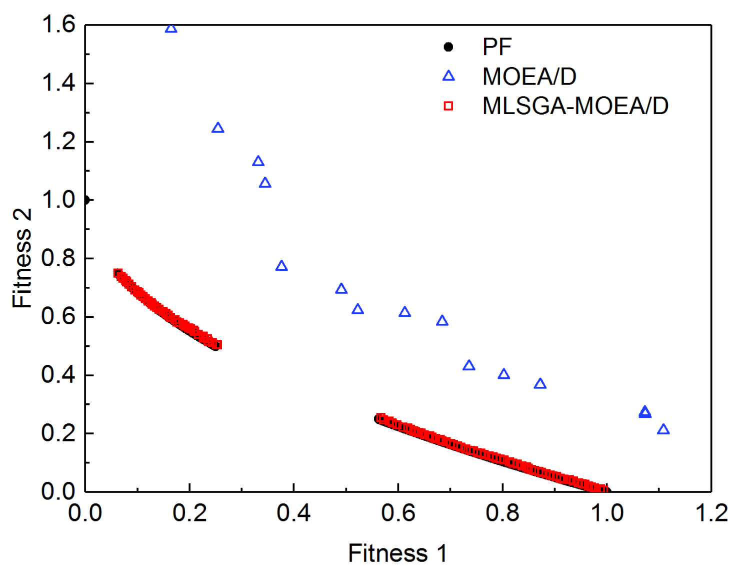

Figure 2: Pareto Front of MOEA/D and MLSGA-MOEA/D on the CF2.

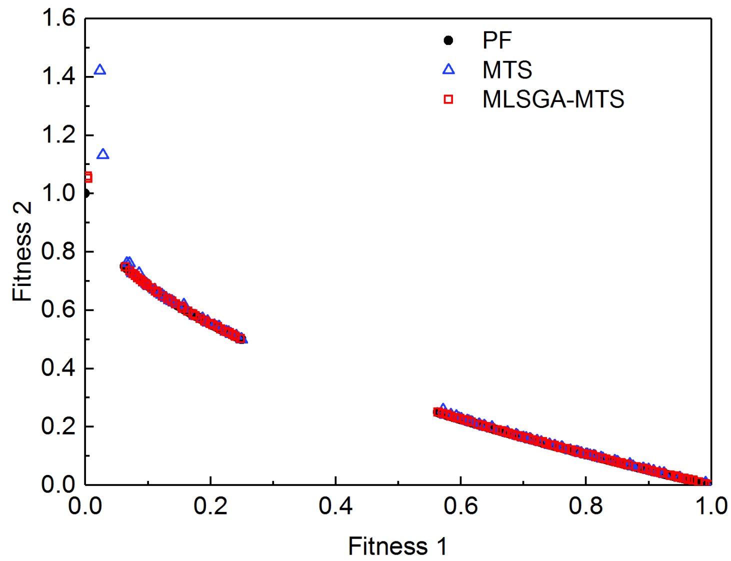

Figure 3: Pareto Front of MTS and MLSGA-MTS on the CF2.

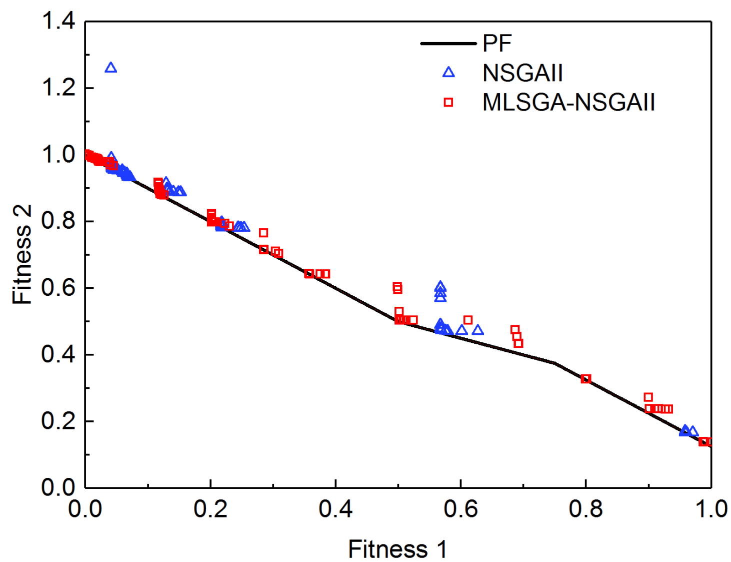

Figure 4: Pareto Front of NSGA-II and MLSGA-NSGA-II on the CF5.

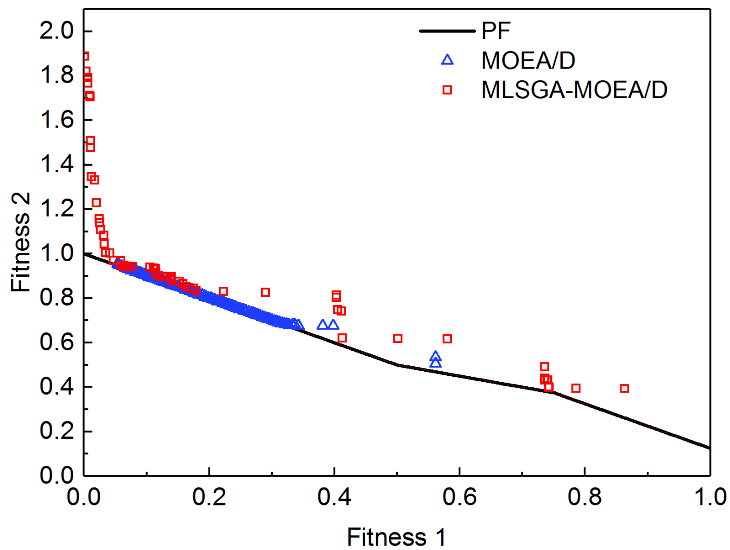

Figure 5: Pareto Front of MOEA/D and MLSGA-MOEA/D on the CF5.

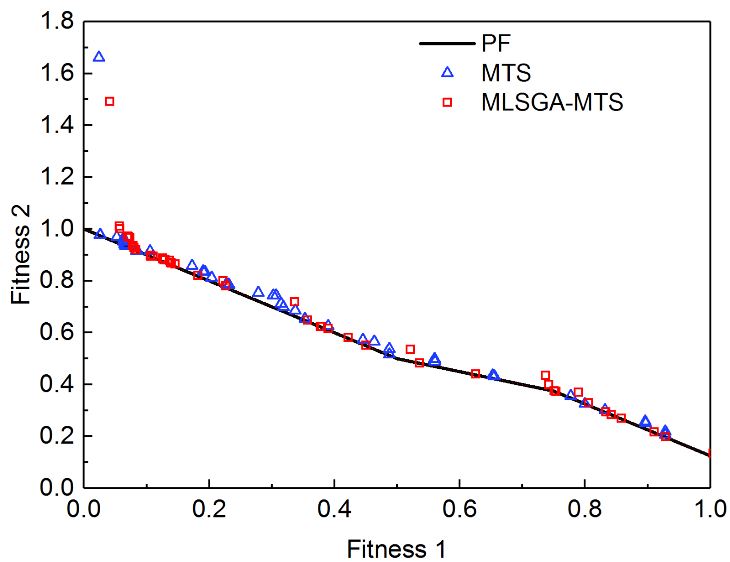

Figure 6: Pareto Front of MTS and MLSGA-MTS on the CF5.

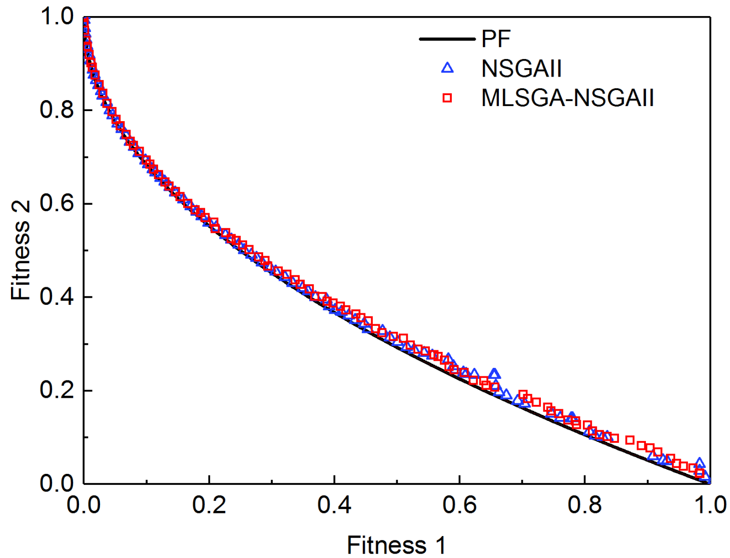

Figure 7: Pareto Front of NSGA-II and MLSGA-NSGA-II on the UF2.

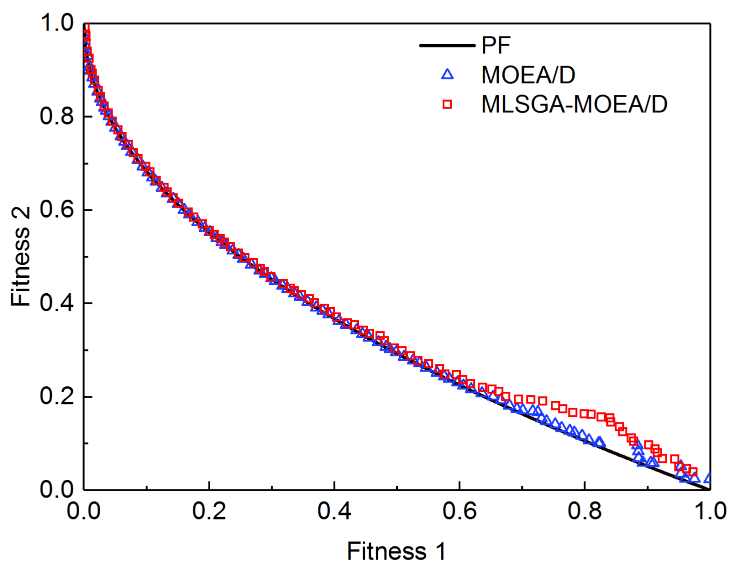

Figure 8: Pareto Front of MOEA/D and MLSGA-MOEA/D on the UF2.

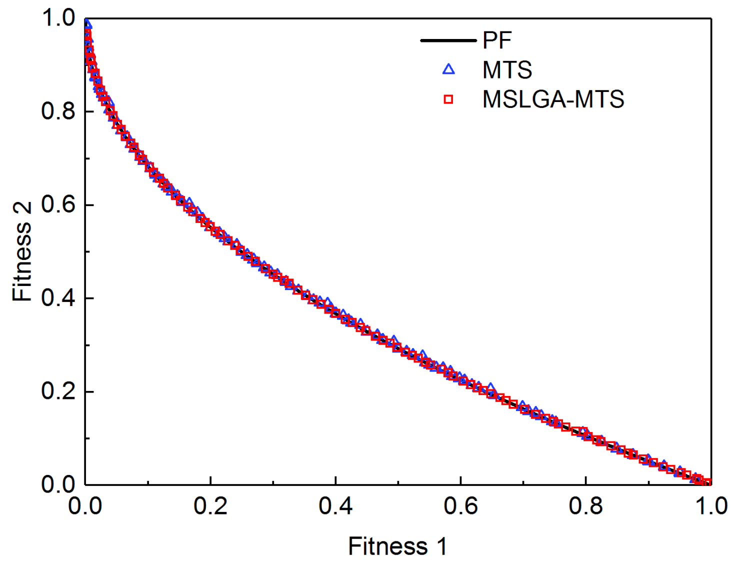

Figure 9: Pareto Front of MTS and MLSGA-MTS on the UF2.

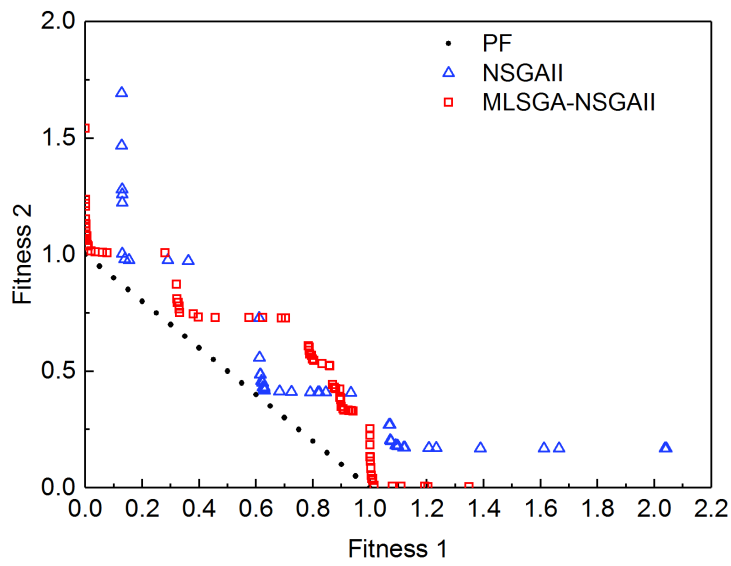

Figure 10: Pareto Front of NSGA-II and MLSGA-NSGA-II on the UF5.

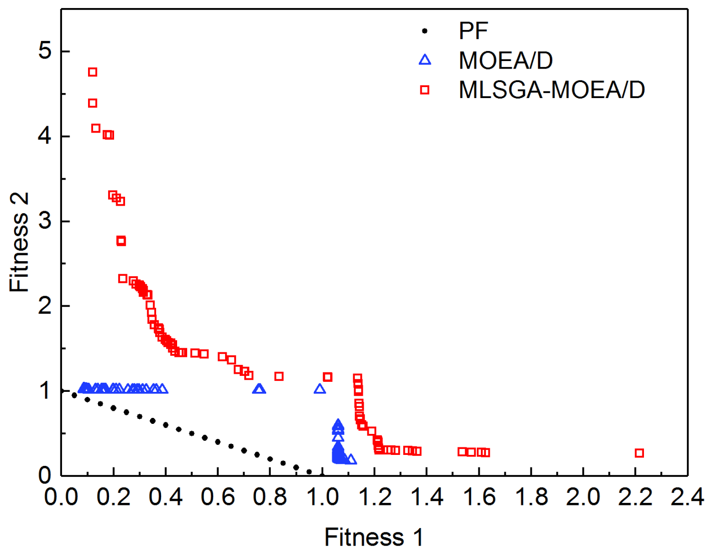

Figure 11: Pareto Front of MOEA/D and MLSGA-MOEA/D on the UF5.

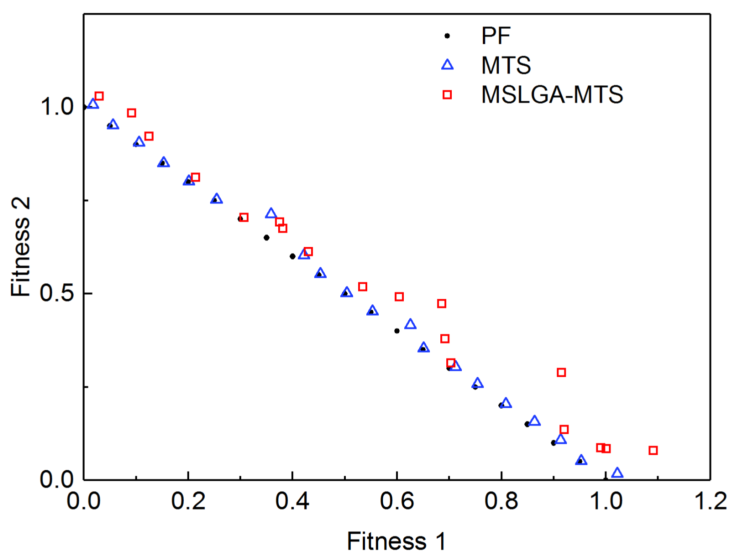

Figure 12: Pareto Front of MTS and MLSGA-MTS on the UF5.

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.

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 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 CEC09 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 CEC09, 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

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.

poor performance of the original MTS algorithm. The proposed methodology would have been placed 2nd 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 [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 to many-objectives problems as concerns

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

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

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.

The authors suggest that the increase in performance is due a combination of the use of novel reproduction mechanisms between sub-populations, 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 Figure 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 solvers 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 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 CEC09 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.

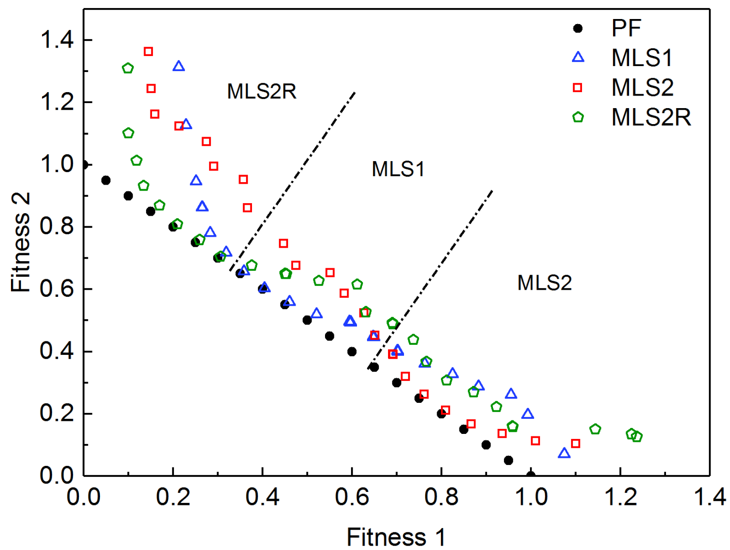

Figure 13: Pareto Front of MLS1, MLS2 and MLS2R fitness definition types from MLSGA-NSGA-II algorithm on the CF1 problem.

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 CEC09 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). doi: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, 2017 IEEE Congress on Evolutionary Computation, CEC 2017 - Proceedings (2017) 1613–1620doi: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 and Evolutionary Computation 1 (2011) 32–49. doi:10.1016/j.swevo.2011.03.001.
URL 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 Transactions on Evolutionary Computation 6 (2) (2002) 182–197. doi: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. doi:10.1109/CEC.2007.4424940.

[9] Q. Zhang, H. Li, MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731. doi: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: 8th International Conference, EMO 2015, Guimar{ã}es, Portugal, March 29 –April 1, 2015. Proceedings, Part II (2015) 34–49doi: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. doi: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, Journal of Computing and Information Technology 7 (1999) 33–47. doi:10.1.1.36.7225.

[13] H.-l. Liu, F. Gu, Q. Zhang (3). doi:10.1109/TEVC.2013.2281533.

[14] J. Branke, H. Schmeck, K. Deb, M. Reddy.S, Parallelizing multi-objective evolutionary algorithms: cone separation, Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753) 2 (2004) 1952–1957. doi: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 Congress on Evolutionary Computation (1) (2009) 2913–2918. doi:10.1109/CEC.2009.4983309.

[16] H. L. Liu, X. Li, The multiobjective evolutionary algorithm based on determined weight and sub-regional search, 2009 IEEE Congress on Evolutionary Computation, CEC 2009 (2009) 1928–1934doi:10.1109/CEC.2009.4983176.

[17] Q. Zhang, P. N. Suganthan, Final Report on CEC ' 09 MOEA Competition, Tech. rep. (2009).
URL http://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 Transactions on Evolutionary Computation 20 (5) (2016) 645–665. doi: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 Transactions on Evolutionary Computation 20 (5) (2016) 711–729. doi:10.1109/TEVC.2015.2512930.

[20] T. Lenaerts, A. Defaweux, P. V. Remortel, B. Manderick, Modeling Artificial Multilevel Selection, in: 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, Proceedings 2010 IEEE 5th International Conference on Bio-Inspired Computing: Theories and Applications, BIC-TA 2010 (2010) 271–277doi:10.1109/BICTA.2010.5645316.

[22] R. Akbari, K. Ziarati, A multilevel evolutionary algorithm for optimizing numerical functions, International Journal of Industrial Engineering Computations 2 (2) (2011) 419–430. doi: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. doi: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 Transactions on Evolutionary Computation 18 (4) (2014) 577–601. doi: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 Transactions on Intelligent Systems and Technology 2 (3) (2011) 1–27. arXiv:0-387-31073-8,doi:10.1145/1961189.1961199.
URL 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 Transactions on Evolutionary Computation 18 (4) (2014) 602–622. doi: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 Transactions on Evolutionary Computation 19 (5) (2015) 694–716. arXiv:arXiv:1011.1669v3, doi:10.1109/TEVC.2014.2373386.

[29] S. Jiang, S. Yang, Y. Wang, X. Liu, Scalarizing Functions in Decomposition-Based Multiobjective Evolutionary Algorithms, IEEE Transactions on Evolutionary Computation 22 (2) (2018) 296–313. doi: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 Transactions on Evolutionary Computation 22 (2) (2018) 226–244. doi:10.1109/TEVC.2017.2704118.

[31] S. Yang, M. Li, X. Liu, J. Zheng, A grid-based evolutionary algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 17 (5) (2013) 721–736. doi:10.1109/TEVC.2012.2227145.

[32] J. Bader, E. Zitzler, HypE : An algorithm for fast Hypervolume-Based Many-Objective Optimization, Evolutionary Computation 19 (1) (2011) 45–76. doi:10.1162/EVCO_a_00009.

[33] E. Zitzler, K. Deb, L. Thiele, Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, Evolutionary Computation 8 (2) (2000) 173–195. doi:10.1162/106365600568202.

[34] S. Huband, L. Barone, L. While, P. Hingston, A Scalable Multi-objective Test Problem Toolkit, in: Evolutionary Multi-Criterion Optimization, no. June, 2005, pp. 280–295. doi:10.1007/978-3-540-31880-4_20.