University of Southampton Research Repository

Copyright © and Moral Rights for this thesis and, where applicable, any accompanying data are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis and the accompanying data cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content of the thesis and accompanying research data (where applicable) must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holder/s.

When referring to this thesis and any accompanying data, full bibliographic details must be given, e.g.

Thesis: Sizhe Yuen (2025) ”Epigenetic Crossover for Multi-objective Evolutionary Algorithms”, University of Southampton, Faculty of Engineering and Physical Sciences, School of Engineering, PhD Thesis, 1 – 173.

Data: Sizhe Yuen (2025) Dataset in support of the Southampton doctoral thesis ’Evolutionary algorithms multi-objective benchmarking’. URI [https://doi.org/10.5258/SOTON/D2855]

University of Southampton

Faculty of Engineering and Physical Sciences
School of Engineering

Epigenetic Crossover for Multi-objective Evolutionary Algorithms

by

Sizhe Yuen
ORCiD: 0000-0001-5552-2074

A thesis for the degree of
Doctor of Philosophy

September, 2025

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
School of Engineering

Doctor of Philosophy

Epigenetic Crossover for Multi-objective Evolutionary Algorithms

by Sizhe Yuen

There are hundreds of different Evolutionary Computation approaches developed to solve multi-objective optimisation problems. Among these approaches there are only two fundamental evolutionary concepts, split between the two main areas of the field: Evolutionary Algorithms with genetic inheritance, and Swarm Intelligence with cultural inheritance. Modern evolutionary biology has since continued to study further evolutionary and non-genetic mechanisms that is relatively unexplored in Evolutionary Computation.

In this thesis, a framework to analyse existing Evolutionary Computation algorithms is developed, based on evolutionary concepts from the Extended Evolutionary Synthesis. The gap in epigenetic inheritance is identified through this analysis as an approach with high potential due to its fast partially-genetic adaptability to dynamic changes in the environment. Furthermore, a detailed benchmarking suite is used to test and compare existing Genetic Algorithms and Particle Swarm Optimisation algorithms to determine their differences and suitability to incorporate a new epigenetic mechanism.

Genetic Algorithms are therefore chosen for epigenetics because the increased diversity balances the convergence properties of epigenetic fast adaptations. Next, a novel epigenetic blocking mechanism based on gene silencing is developed and tested. Performance on static and dynamic multi-objective problems show the improvement the epigenetic mechanism can make, with improved performance across the duration of the optimisation process. Further study and comparison of the hyperparameters and a gradient-based approach indicate the mechanism can be both problem and algorithm specific. The choice of blocking variables with all positive or negative gradients achieve the best general results, and utilising different hyperparameters specifically tuned for problems with changing Pareto sets achieve the best performance.

Finally, the epigenetic mechanism is applied to a real-world voyage optimisation system. Faster convergence is demonstrated for voyages in calm weather conditions, and savings on fuel consumption are found for more complex voyages in severe weather conditions.

Contents

List of Figures

List of Tables

Nomenclature

Multi-objective optimisation

CR	The crossover rate, or probability of crossover.
$\mathbf{F}(\mathbf{x})$	The objective functions to be optimised.
F	The mutation or scale factor for Differential Evolution.
L	The learning strategy for Swarm Intelligence algorithms.
MR	The mutation rate, or probability of mutation.
P	A population of candidate solutions.
$\mathbf{x}$	The decision variable vector representing a candidate solution.
β	The spread factor for Simulated Binary Crossover.
d	The number of decision variables.
η_m	Distribution index of polynomial mutation.
o	The number of objectives.
x_i	The ith decision variable.

Particle Swarm Optimisation

c_1	Constant to influence the importance of **pbest**.
c_2	Constant to influence the importance of **gbest**.
r_1, r_2	Random values in range $[0, 1]$.
gbest	The globally best solution found by the entire population so far.
ω	Inertia of particles to balance convergence or diversity.
pbest	The best solution found by a particle so far.

Ant Colony Optimisation

$P(v_i)$	The probability of selecting the candidate value v_i for the decision variable x_i.
V	The set of candidate values available for a given decision variable.
α	Parameter controlling the influence of the pheromone value $\tau(v_i)$ on the selection probability.

β	Parameter controlling the influence of the heuristic value $\eta(v_i)$ on the selection probability.
$\Delta\tau(v_i)$	The amount of pheromone deposited by an ant on the trail corresponding to v_i.
$\eta(v_i)$	The heuristic value associated with the candidate value v_i.
p	The pheromone evaporation rate.
$\tau(v_i)$	The pheromone value associated with the candidate value v_i.
v_i	A candidate value for the decision variable x_i.

Artificial Bee Colony

P_e, P_o, P_s	Sub-populations representing employed, onlooker, and scout bees, respectively.
$P(i)$	The probability of an onlooker bee selecting the ith employed bee.
$\mathbf{r}$	A vector of random numbers with values in the range $[0, 1]$.
w_1	Control parameter determining the influence of the archive solution.

Grey Wolf Optimiser

$\mathbf{A}$	A coefficient vector used to influence diversity.
$\mathbf{C}$	A coefficient vector used to influence convergence.
$\mathbf{x}_\alpha$	Alpha wolf: The best solution found so far.
$\mathbf{x}_\beta$	Beta wolf: The second best solution found so far.
$\mathbf{D}_\alpha$	Distance vector between an omega wolf and the alpha wolf.
$\mathbf{D}_\beta$	Distance vector between an omega wolf and the beta wolf.
$\mathbf{D}_\delta$	Distance vector between an omega wolf and the delta wolf.
$\mathbf{x}_\delta$	Delta wolf: The third best solution found so far.
$\mathbf{r}_1$, $\mathbf{r}_2$	Random vectors with values in the range $[0, 1]$.
$\mathbf{a}$	A constant vector with values decreasing from 2 to 0.

Firefly Algorithm

α	A randomisation coefficient.
β_0	The attractiveness at distance zero.
ϵ	A vector of random numbers.
γ	Light absorption coefficient.
$\mathbf{r}_{ij}$	Distance between two solutions $\mathbf{x}_i$ and $\mathbf{x}_j$.

Cuckoo Search

α	Step size scaling factor.
Lévy(λ)	A random step drawn from a Lévy distribution.

Dynamic multi-objective optimisation problems

n_t	The severity of dynamic changes.
τ_t	The frequency of dynamic changes.
t	The time step of the dynamic problem.

Abbreviations

EA	Evolutionary Algorithm.
EC	Evolutionary Computation.
GA	Genetic Algorithm.
PSO	Particle Swarm Optimisation.
SI	Swarm Intelligence.
ABC	Artificial Bee Colony.
ACO	Ant Colony Optimisation.
APS	Average Performance Score.
cMLSGA	Co-evolutionary Multi-Level Selection Genetic Algorithm.
CMPSO	Co-evolutionary Multi-Swarm Particle Swarm Optimisation.
DE	Differential Evolution.
EIB	Epigenetic variant with increasing mask sizes.
EIP	Epigenetic variant with increasing probabilities.
GN1	Epigenetic variant where all variables with negative gradients are blocked.
GP1	Epigenetic variant where all variables with positive gradients are blocked.
GWO	Grey Wolf Optimisation.
HEIA	Hybrid Evolutionary Immune Algorithm.
HPO	Hyperparameter optimisation.
HV	HyperVolume.
IBEA	Indicator-based Evolutionary Algorithm.
IGD	Inverted Generational Distance.
MOEA/D	Multi-objective Evolutionary Algorithm based on Decomposition.
NSGA-II	Nondominated Sorting Genetic Algorithm II.
PN1	Epigenetic variant where variables with positive gradients have higher probability to be blocked.
PP1	Epigenetic variant where variables with negative gradients have higher probability to be blocked.
SBX	Simulated Binary Crossover.
SMAC	Sequential Model-Based Optimisation for General Algorithm Configuration.
SMPSO	Speed-constrained Multi-objective Particle Swarm Optimisation.

U-NSGA-III Unified Nondominated Sorting Genetic Algorithm III.

Declaration of Authorship

I, *Sizhe Yuen*, declare that this thesis and the work presented in it is 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 listed in publications.

Signed:.. Date: 22nd September 2025

Acknowledgements

I would first like to thank and acknowledge my supervisors Professor Adam Sobey and Professor Tom Ezard, for their advice, guidance, and insight, particularly during the turbulent times of the coronavirus lockdowns. Thank you to Adam for pushing me to present and publish my work, providing me with invaluable experiences that helped me learn and grow. And thank you to Tom for bringing your fresh biological points of view that helped guide the direction of my research.

I am grateful for the financial support of the Southampton Marine and Maritime Institute, and acknowledge the use of the IRIDIS 5 High Performance Computing Facility, both of which made this research possible.

Finally I want to thank my parents and my close friends for their support and encouragement through the years.

Chapter 1

Introduction

1.1 Evolutionary Computation

The idea of applying evolutionary mechanics to algorithms was first proposed by Turing in the 1950s [188] with a learning machine, drawing parallels between how a machine could learn with Darwinian evolution and the survival of the fittest. He proposed that the machine could incrementally improve itself through mutations, receiving punishments and generating rewards as part of a teaching process. Holland successfully implemented a Genetic Algorithm (GA) in 1969 [84, 85] which popularised the use of GAs by the 1980s.

Today, the Evolutionary Computation literature is divided between two general fields:

- **Evolutionary Algorithms** [11] (EA) - Algorithms with a foundation in genetics, these date back to Turing's learning machine [188], and include the branches of Evolutionary Programming (EP), Evolution Strategies (ES), and Genetic Algorithms (GA). These algorithms generally involve analogies to the genetic mechanisms of selection, recombination (crossover) and mutation.
- **Swarm Intelligence** [22] (SI) - Algorithms based on collective intelligence with patterns of communication and interaction in a population. Swarm Intelligence algorithms cover a wide range of biological inspirations, from animal behaviour algorithms such as Particle Swarm Optimisation [100] (PSO) and Ant Colony Optimisation [51] (ACO) as well as more esoteric inspirations such as political anarchy (Anarchic Society Optimisation [1]).

In Evolutionary Algorithms, candidate individual solutions are represented in the search space as genotypes, evolving with genetic operators. The algorithms use genetic operators such as mutation and crossover to evolve the solutions through genetic inheritance. Some EAs use a genotype-phenotype mapping, where the underlying genotype is decoded into a phenotype candidate solution for evaluation, separating the search space from objective space. Throughout this thesis, the Evolutionary Algorithms discussed will not utilise this mapping unless explicitly stated, and instead there is no encoding process, i.e. the genotype representation is the same as the candidate solution. In Swarm Intelligence algorithms such as PSO, there is typically no separate encoding or genotype-phenotype mapping. Behavioural metaphors such as hunting, foraging, and movement behaviours are used to iteratively improve the candidate solutions. As the EAs discussed in this thesis do not use a genotype-phenotype mapping, the two types of algorithms discussed use the same representation, but different methods of traversing the search space.

As the field developed, Evolutionary Computation methods such as Genetic Algorithms were found to perform well in more complex optimisation problems with multiple objectives. This led to new methods of solving multi-objective problems with 2-3 objectives that must be optimised at the same time, from the Pareto-based niching method of NSGA-II [45] to the mathematical decomposition-based method of MOEA/D [217].

Evolutionary Computation algorithms are typically used to solve optimisation problems, as they can continue to search and develop solutions iteratively. Multi-objective optimisation problems involve optimisation on multiple objectives at the same time. Multi-objective optimisation problems have a wide range of real-world applications, from engineering design and financial modelling, to detailed scheduling [52, 152, 177]. For Evolutionary Computation algorithms, multi-objective problems present a challenge to determine how to assign a fitness value to a solution. Unlike single objective problems, objectives in a multi-objective problem can contradict one another. Finding a better solution for one objective may deteriorate the other objective, and it is difficult to define how one solution may be better or worse than another for all objectives.

1.1.1 Biological analogies

In evolutionary biology, the Modern Synthesis [89] was developed throughout the first half of the 1900s particularly during the 1930s, combining the ideas of Darwin's theory of evolution by natural selection [38] and Mendel's principles of inheritance [15]. However, new research into non-genetic concepts suggest that the idea of the Modern Synthesis should be extended [148, 149] to include the effects such as epigenetics, cultural inheritance, parental effects, and multi-level selection. These effects suggest that both genetic and non-genetic inheritance have a strong effect on the process of evolution. While the development of Evolutionary Algorithms can be compared directly to the concepts of the Modern Synthesis, there have only been small steps to including these new modes of inheritance to the field of Evolutionary Algorithms, despite their increasing centrality to modern evolutionary theory [35].

1.2 Aim and objectives

The aims of this project are to: identify suitable new concepts in the Extended Evolutionary Synthesis for Evolutionary Computation, then to develop and test a new mechanism based on the identified Extended Synthesis concept of epigenetics.

Due to the faster-than-generational adaptation capabilities of epigenetics, it is hypothesised that the incorporation of epigenetic processes within Evolutionary Algorithms will enable the algorithms to adapt more effectively to multi-objective optimisation problems, particularly in dynamic scenarios where the problem or objectives change during the optimisation. The new mechanism could demonstrate improved performance in terms of finding optimal solutions and adapting to dynamic environments.

This will be achieved with the following objectives:

1. Summary of opportunities for Evolutionary Computation from the Extended Evolutionary Synthesis

   While the development of Evolutionary Computation algorithms can be compared directly to the concepts of the Modern Synthesis, there have only been small efforts to including new concepts from the Extended Evolutionary Synthesis to the field of Evolutionary Computation. A number of popular Evolutionary Algorithms and Swarm Intelligence algorithms must be analysed to discover which concepts and mechanisms of current evolutionary biology have been applied, and which have been overlooked.

2. Suitability of Evolutionary Algorithms and Swarm Intelligence algorithms

   There are a large number of possible algorithms a new epigenetic mechanism could be applied to. Notably there are two main classes of Evolutionary Computation that must be considered: Evolutionary Algorithms, and Swarm Intelligence algorithms. The algorithms are put through a rigorous benchmarking suite to determine their strengths and weaknesses in performance. The methodology and benchmarking framework used for testing and comparing the existing algorithms is then followed in all future performance testing.

3. An epigenetic blocking mechanism

   A new epigenetic mechanism is developed based on the research into the Extended Evolutionary Synthesis. Two base algorithms identified during the benchmarking work are used to test and evaluate the new mechanism. A further study into hyperparameters and dynamic multi-objective problems is performed to show the strengths and weaknesses of the mechanism, and how it aligns to the expected behaviour and circumstances of epigenetics in the natural world.

1.3 Novelty

There are a large number of Evolutionary Computation algorithms in the current literature, from a multitude of variants of a single Genetic Algorithm, to a host of Swarm Intelligence algorithms aiming to mimic different animal behaviours. Reviews in the literature focus on subsections of algorithms such as Genetic Algorithms or Particle Swarm Optimisation algorithms, detailing their mechanics and approaches, but without linking these mechanics back to their biological roots. It is difficult to distinguish between the different algorithms on a fundamental level, especially in the vast number of Swarm Intelligence algorithms, where each algorithms uses different terms and descriptions for the same underlying processes. Criticism of this issue has been discussed in the literature [9, 174], but new algorithms continue to be developed. The literature review in this project describes the Evolutionary Computation algorithms and their mechanics from an evolutionary biological perspective, grouping many similar algorithms together under a biological framework. The algorithms are analysed and key concepts relating to evolutionary biology are extracted.

The few instances of epigenetic mechanisms and algorithms in the literature have only focused on simple single-objective problems [30, 146, 178], or miss a key aspect of inheritance [29]. Furthermore they have not been thoroughly tested against multiple algorithms, and do not take advantage of epigenetics' adaptability for dynamic environments. The previous studies demonstrate performance improvements, suggesting the potential of an epigenetic mechanism despite lacking key epigenetic features. The epigenetic mechanism developed here can be added on to any existing Evolutionary Computation algorithm, and is benchmarked against a large number of state-of-the-art static and dynamic multi-objective test problems. Using many problems with a mixture of different categories of problems allows the performance to be more accurately measured. Additionally, dynamic elements to change the hyperparameters of the epigenetic mechanism is tested to show the best case scenarios for the new mechanism. The epigenetic mechanism is further applied onto a real-world voyage optimisation system to exhibit improved convergence properties and optimising shipping routes with reduced fuel consumption.

1.4 Structure of thesis

Chapter 2 frames current Evolutionary Computation approaches in a biological framework, demonstrating how few of the mechanisms that describe evolution have led to algorithms inspired by those mechanisms. An analysis on the mechanics used in Evolutionary Algorithms and Swarm Intelligence algorithms show a gap in epigenetic inheritance that is yet to be explored.

Next, Chapter 3 analyses the differences between current state-of-the-art Genetic Algorithms and Particle Swarm Optimisation algorithms, the two most popular group of algorithms within Evolutionary Computation. Comparisons and benchmarks between algorithms in Evolutionary Computation are common, but they are usually done in the context of using those algorithms to solve a specific problem [159, 175, 215], rather than comparing algorithms as a whole against many different problems. Their performance is compared across a range of 97 benchmark problems. The strengths and weaknesses of each type of algorithm on specific problem categories are shown.

Chapter 4 introduces a new epigenetic mechanism that can be applied to any Evolutionary Algorithm based on epigenetic gene silencing with inheritance. Benchmarks problems for static and dynamic multi-objective optimisation problems are tested on the two most popular multi-objective Genetic Algorithms to show the performance of the mechanism, its strengths and its weaknesses.

Chapter 5 delves deeper into testing the parameters of the epigenetic mechanism. The effects of dynamically changing parameters, gradient calculations to decide on which decision variables to block, and hyperparameter optimisation are investigated to demonstrate the best case performances of the epigenetic mechanism on dynamic multi-objective optimisation problems.

A real world application of the epigenetic mechanism is then presented in Chapter 6 with a commercial voyage optimisation problem, demonstrating the practical advantages of an Evolutionary Algorithm with the epigenetic mechanism to save fuel usage and carbon emissions.

Finally, Chapter 7 concludes the thesis, and points to directions that future work on epigenetic mechanisms can take to improve on a gradient-based approach, and build an adaptive and reversible epigenetic mechanism.

1.5 List of publications

The following chapters and sections have been submitted or published:

- Chapter 2:

  S. Yuen, T. H. G. Ezard, A. J. Sobey. “Epigenetic Opportunities for Evolutionary Computation” In: *The Journal of Royal Society Open Science* 10.5 (2023) 10:221256. http://doi.org/10.1098/rsos.221256

- Section 5.2:

  S. Yuen, T. H. G. Ezard, A. J. Sobey. “The effect of epigenetic blocking on dynamic multi-objective optimisation problems” In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion, GECCO ’22.* Pages 379 – 382. https://doi.org/10.1145/3520304.3529022

Chapter 2

Epigenetic opportunities for Evolutionary Computation

This Chapter reviews the literature in biological evolutionary theory, multi-objective optimisation, and evolutionary computation. First a review of the biological literature establishes the various concepts within the Modern Synthesis, then continues with the Extended Synthesis and non-genetic concepts of inheritance and information transfer. Next, evolutionary computation is introduced and related back to the biological literature to establish the connection between the two fields. Then, the multi-objective optimisation problem is defined, and different evolutionary computation algorithms are discussed in the context of multi-objective optimisation and the evolutionary or non-genetic operators they use. Finally, the possibilities within the Extended Synthesis that have not been well explored in evolutionary computation is discussed, and the potential of epigenetic inheritance is identified.

2.1 Review of current evolutionary theory

The current study of evolution has gone through significant iterations since Darwin's original theory of natural selection. Modern evolutionary theory has evolved to integrate new discoveries, resulting in the further understanding of information transfer and non-genetic inheritance. The Modern Synthesis provided a unifying framework that combined Mendelian genetics with Darwinian evolution, but continued refinements within the Extended Evolutionary Synthesis further incorporate non-genetic factors like epigenetics, niche construction, and developmental processes.

First, the major developments in evolutionary theory are reviewed, following its progression from the Modern Synthesis to the contemporary work on the Extended Evolutionary Synthesis.

2.1.1 The origins of evolution: The Modern Synthesis

The Modern Synthesis [89] was developed throughout the 1900s to combine the ideas of Darwin's evolution by natural selection [38] and Mendel's principles of inheritance [15]. Darwin's initial concepts in "*On the Origin of Species by Means of Natural Selection*" lay down the building blocks of evolutionary theory with common descent and natural selection. Common descent discusses how different organisms descend from the same common ancestors, and natural selection describes how some populations have traits which are more suitable to an environment and therefore survive compared to others. However Darwin's theory struggled with incorporating heredity and inheritance and therefore describing how important traits are passed down or developed in new generations. Meanwhile a new theory of Mendelian inheritance grew popular after Mendel's work on pea plant genetics was rediscovered. Mendel's work provided an important discovery on how genetic characteristics in offspring individuals are inherited from their parents. These discoveries seemed to oppose Darwin's view of continuous variation through blending inheritance, where an offspring inherits the average characteristics of its parents. These opposing theories were consolidated by of Fisher [62] and Wright [202] who showed a mathematical framework for how discrete Mendelian genes collectively demonstrated the effect of continuous variation under Darwin's theory.

The Modern Synthesis can be summarised with a number of key points [64]:

- Genetic mutation is random and there is no connection between the direction of mutations and the direction of higher fitness.
- Inheritance is genetic but characteristics, traits acquired by individuals in their lifetime, are not inherited.
- Mutations have small effects which lead to gradual evolutionary changes with time.
- The relative frequency of genes in the population evolve due to chance (natural selection, mutation, and genetic drift, where the frequency of genes change over time from chance).
- Natural selection is the only explanation as to why organisms have traits that are well adapted to their environment.

These ideas give a picture of evolution which revolves around random genetic variation, genetic inheritance, and natural selection. Organisms survive because their genetic characteristics happen to fit into the direction of natural selection.

According to the Modern Synthesis, processes such as mutation and natural selection affect components at different levels of an individual organism. An organism's genetic make-up consists of a number of components. Figure 2.1 shows the differences between alleles, genotypes, and phenotype components of an organism. Alleles of genes, the form of a given gene variant, combine to form the genetic variation among individuals. The set of genes are known as the genotype of an organism. Genetic mutation and recombination occur at the level of genes and alleles, where offspring inherit the genes of its parents and mutation can occur.

The phenotype of an individual refers to its observable traits such as physical appearance and behaviour. It is affected by the individual's inherited genotype, but can also be affected by external factors such as environmental conditions and learnt behaviours. The relationship between genotypes and phenotypes can be expressed as a mapping [4] to determine how a genotype influences an individual's phenotype.

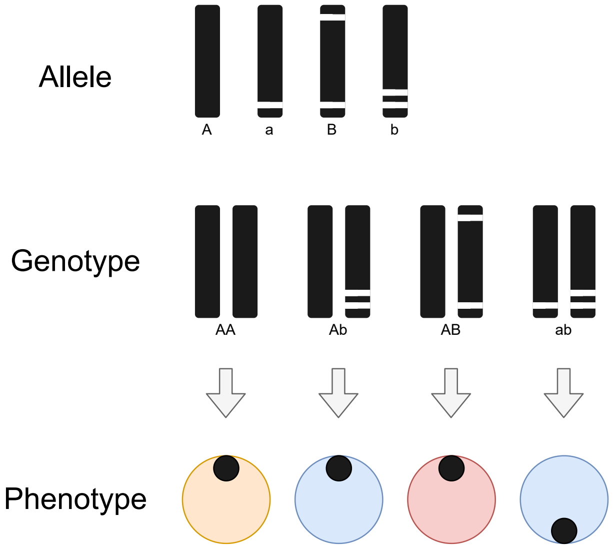


Figure 2.1: Simplified distinction between alleles, genotypes, and phenotypes.

2.1.2 Extending the Modern Synthesis

Starting from the 1990s, there have been a number of questions raised about the Modern Synthesis, and new evidence on modes of non-genetic mechanisms that also have a role to play in evolutionary theory [24, 134, 160]. As new discoveries were made, it became increasingly difficult for the Modern Synthesis to answer new questions about the role of development in evolution, different levels of selection, the relationship between evolutionary biology and ecology, and additional modes of inheritance. Pigliucci [148] and Müller [134] suggested that the ideas of the Modern Synthesis should be extended [149] to include the additional biological effects such as epigenetics, cultural inheritance, evolutionary developmental biology, and multilevel selection.

The elements of the Extended Synthesis expand on to the existing concepts of the Modern Synthesis to allow for a broader range of ideas and explanations. Table 2.1 shows how each concept of the Modern Synthesis is expanded upon in the Extended Synthesis. More nuance and flexibility regarding inheritance, variation, and the methods of evolutionary change is added, allowing for more explanations on how organisms can adapt and change to the environment, rather than a random, natural selection based explanation.

Table 2.1: Concepts of the Modern Synthesis are expanded upon in the Extended Evolutionary Synthesis to build more understanding of the evolutionary process.

Modern Synthesis concept	**Extended Synthesis concept**
Genetic inheritance	Inclusive inheritance - extending the concept of inheritance beyond genetics to include epigenetic, ecological, social and cultural modes of non-genetic inheritance.
Random genetic variation	Non-random variation - non-random mutation leading to developmental bias, so some characteristics become more common within a population
Natural selection	Reciprocal causation - Developmental processes and niche construction allow organisms to both affect and be affected by the environment
Gradual evolutionary change	Variable rates of change - Developmental processes and epigenetic inheritance can result in varying rates of evolutionary change as a response to environmental cues

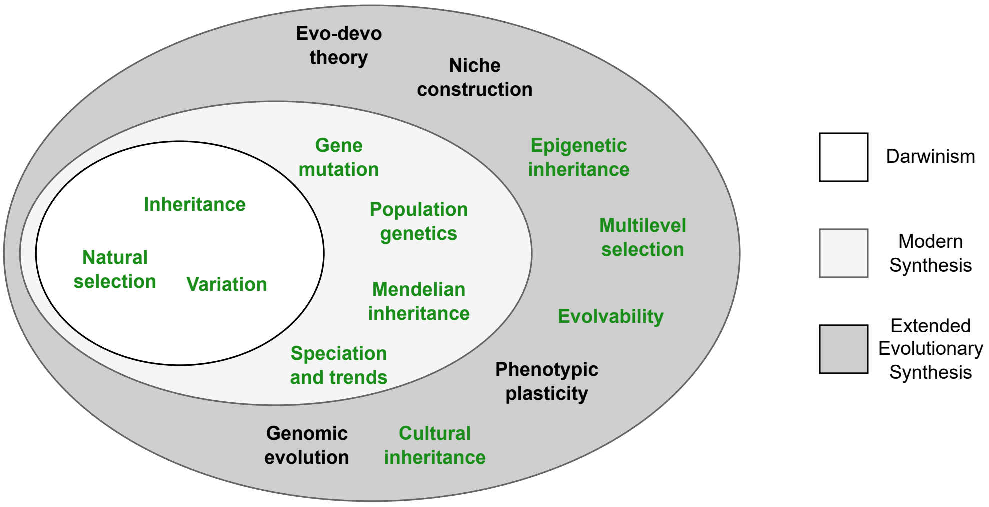


Figure 2.2: Key concepts of Darwinism, the Modern Synthesis, and the Extended Synthesis, based on Pigliucci and Müller [149]. The green highlights concepts which have been implemented in existing evolutionary algorithms.

Adding on to the list of concepts in the Modern Synthesis, Figure 2.2 lists the concepts of the Extended Synthesis and shows how Darwin's original theory of evolution and the Modern Synthesis are contained within the Extended Synthesis.

Non-genetic inheritance

The different forms of non-genetic inheritance can be divided based on their mode of transmission [36]:

- **Epigenetic inheritance** - In genetic inheritance, the evolution of species comes from changes in the DNA sequence due to natural selection. Epigenetic mechanisms alter DNA expression without altering the DNA sequence [50]. This allows for a rapid change in phenotypes without a change in the inherited genotype, leading to faster adaptation to selection pressures. Epigenetic marks trigger changes in an individual's phenotype based on external factors such as the environment, and are passed on from parent to offspring. While epigenetic marks can cause changes to a phenotype, they can also remain dormant awaiting the triggering of the appropriate environmental cue. This is a crucial difference between epigenetic and genetic inheritance. Traits transmitted through genetics are considered much harder to change or revert compared to epigenetics.

- **Cultural inheritance** - Apart from inherited genotypes, information relevant to survival can also be exchanged and inherited socially. Cultures themselves can also "evolve" [127] and can be found in a large number of animal species. A trait is at least partially culturally inherited if it follows the following four criteria [37]:

1. The trait is expressed as a result of learning from other individuals, and not inherited in another way.
2. The trait must be passed on through multiple generations. This allows socially learned traits to become a part of evolution, as transmission across the same generation will not lead to inheritance.
3. The phenotype of an individual must be changed as a result of social learning for long enough to allow new individuals to observe and learn the same trait.
4. Changes to the individual's phenotype must be general and adaptable to other similar conditions.

- **Ecological inheritance** - Darwin [39], observed that individuals can alter their surrounding environment to improve their chances of survival. Examples of this are dams and nests made by animals that can sometimes be passed on to future generations. This can be both an in-generational and trans-generational effect, depending on whether the ecological changes made by one generation is able to last for the next generation's benefit. This can lead to significant changes to the environment if the ecological changes are accumulated over many generations [54].

Evolutionary developmental biology (Evo-devo)

Evolutionary developmental biology aimed to bridge the gap and explore the relationship between developmental biology and evolutionary biology [109]. Evo-devo provides an understanding for how individual development occurs and how the developmental process is shaped by evolution.

In the Modern Synthesis theory of evolution, variations in phenotypes are based solely on changes on genotype, through mutation and recombination of alleles and genes, and the mapping from genotype to phenotype. However this makes it difficult to explain how developmental processes, such as the development of embryos, are evolved.

Evidence of developmental bias also suggests that diversity among populations and species are only partly due to natural selection, and some bias occurs during development to increase the convergence of fit phenotypes. For example, cichlid fish from different lakes were found to have similar body shapes even though they were genetically more related to species of fish in their own lakes [5]. This evidence does not mean developmental bias goes against the concepts of the Modern Synthesis, rather they build upon the building blocks of genetic variation and natural selection. The theory of facilitated variation [66] illustrates how small changes in genotypes can lead to larger innovative changes in the phenotype through core processes and signalling pathways that are activated during developmental stages based on small genetic mutations or environmental cues [91].

Niche construction

Niche construction [107] is the concept of changing the environment to better suit the development of future generations. It follows from ecological inheritance and is the process in which individuals change the environmental state around them. In the traditional view of the Modern Synthesis, natural selection is based on the fitness or suitability of the phenotype of organisms to the environment. However, changes to the environment due to niche construction can become a part of evolution when they in turn affect natural selection by introducing selection bias from the changes made.

Niche construction becomes significant in evolution when [139]:

- The changes to the environment are non-random and create a selection bias in the environment. For example when bacteria decompose compounds and excrete matter that affects the chemical make up of the environment.
- The ecological inheritance from niche construction leads to stability in environmental conditions lasting for multiple generations.
- The characteristic of modifying environments is inherited.

Even if only one species engages in niche construction activities, other species in the same ecosystem may also be affected, leading to co-evolution which can occur in both shorter and longer timescales.

Phenotypic plasticity

Phenotypic plasticity is the idea that an organism's behaviour or physiology could change due to environmental factors [154]. The plasticity is the ability of a single genotype to produce multiple phenotypes in response to environmental cues, unlike the direct mapping shown in Figure 2.1. The response can occur at any level of organisation, from development to morphology [196]. Phenotypic plasticity specifically involves changing the phenotype without any modification to the genotype. With high plasticity, one genotype could produce many different phenotypes for different environments that are more suitable. The trigger for differing phenotypes can also come from developmental stages, linking back to evo-devo. For example, queens and workers in honey bee colonies develop from the same larva, but the diet of the larva determines whether it develops into a worker bee or a queen bee [141].

The decoupling between an organism's genotype and phenotype is a key aspect to allow individuals to quickly adapt to drastic changes in the environment. For static optimisation problems changes to the environment do not occur. It could be argued that the environment around a population of solutions changes depending on the depth of the search, but this change does not happen retrospectively, in other words requiring evolved solutions to be reversed to adapt to a different environment. When a natural environment changes, the fitness landscape of biological organisms change accordingly, as the phenotype and conditions for successful survival and reproduction become different. Organisms with greater plasticity therefore gain an advantage in adapting to the new environment and fitness landscape faster. The concept of including plasticity as a parameter or mechanic is therefore more important in the area of dynamic problems, where the fitness of solutions can change dynamically and may require backtracking of genetically evolved solutions.

Evolvability and Genomic evolution

Evolvability in evolution is the capacity for an organism to evolve [102]. In evolutionary literature, the definition ranges from a few different concepts:

- Heritability [86] - the existing pool of genetic variation available
- Variability [193] - the propensity for characteristics to vary depending on genetic variation
- Innovation [126] - the capacity to overcome existing genetic and developmental constraints leading to radically different designs.

The effect of evolvability is to allow long-term adaptation through changes in an organism's evolutionary processes and lead to major breakthroughs in behaviour and physiology [147]. This relates to genomic evolution, the evolution of the genome architecture itself, with evolvability being the capacity to allow this form of evolution to occur. Computationally, this is related to how variables and solutions to problems are structured, which is often decided by the definition of the optimisation problem and solution.

Evolvability has been modelled computationally in the past by Valiant [189], to show how it can be represented as a method of computational learning. However, his work is focused on modelling biological evolution, to provide a means to determine which biological mechanisms are evolvable and which are not, rather than applying the concept of evolvability to Evolutionary Computation.

Multilevel selection theory

Multilevel Selection is the idea that natural selection occurs at different levels such as the genetic level [41], the individual organism level [138] or the species level. The existence of altruism, a phenotype that contributes to group advantage at the cost of disadvantaging itself [79], suggests that a disadvantage at one hierarchical level may be a advantage at another level, justifying why this behaviour can be seen. The short term payoff is lower because the altruistic individual sacrifices personal benefits, but over the longer term, long-term benefits may be seen at the group level.

A key element to multilevel selection is group selection. That is the concept that natural selection acts on an entire group of organisms rather than solely on individual organism. There are three criteria used to determine what is a suitable unit of selection in evolution [112].

1. Phenotypic variation must exist for that unit of selection
2. There is a correlation between the phenotype and the fitness of the phenotype

3. The phenotype, and therefore the fitness, is heritable

For example in an individual organism, there exists different phenotypes, different fitnesses according to those phenotypes, and heritability of the phenotypes. The question is whether the same criteria could be applied to groups of individuals too. There have been numerous experiments over the years testing group selection [70], showing how groups of individuals can demonstrate traits and variation from other groups, and how such traits are inherited in future generations of the same group. For example Wade in 1979 [192] showed the effect of group selection on the population size of flour beetles. Muir in 1996 [133] demonstrated improvement on mortality and egg production in later generations of hens with a group selection procedure, showing the heritability and fitness of the selected groups.

2.2 Defining current bio-inspired algorithms within the Extended Evolutionary Synthesis

In Evolutionary Algorithms, individuals are presented as a genotype that evolves with genetic operators. The algorithms use genetic operators to evolve candidate solutions through genetic inheritance. For real-value encoded algorithms and problems used in this thesis, the genotype directly represents the candidate solutions, with no encoding and decoding step in between the algorithm and solution representations. The genetic operators manipulate the genotypes to create new variations, and the fittest individuals are selected for the next generation. While individuals in Swarm Intelligence algorithms could be seen as phenotypes in the biological sense of an organism's observable traits, computationally they are represented the same as real-encoded EAs, directly representing problem solutions. Update functions such as the velocity of particles are used to adjusts a solution's position in the search space to continually improve in subsequent iterations. Mathematically, the two types of algorithms have the same underlying mathematical representations in terms of the problem and solution formulations using numeric representations, with different search operators, the details of which will be discussed shortly in section 2.3. The focus of the Extended Evolutionary Synthesis is that there are many interacting routes that influence the final phenotype, not only genetic inheritance, as encoded in Evolutionary Algorithms, or within-generation cultural transmission, as encoded in Swarm Intelligence, but rather a mix of influences both within and across generations. This diversity of influences reduces reliance on a single mode of inheritance and generates phenotypes from a diversified portfolio of influences This allows hedging against maladaptations while also providing more rapid adaptation when genetic, indirect genetic, and phenotypic effects align [149].

A large focus in Evolutionary Computation has been put on improving the convergence of existing algorithms through heuristics relevant to existing benchmark problems, such as

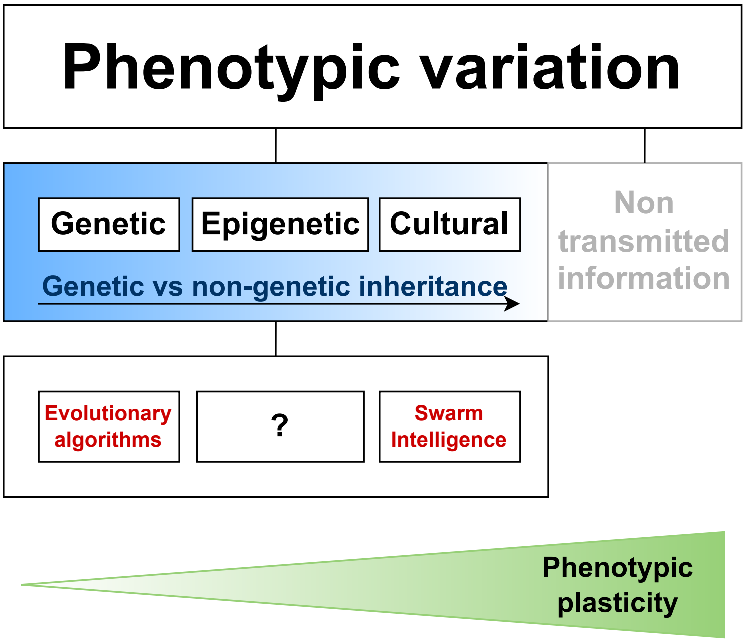


Figure 2.3: Sources of information transmission for phenotypic variation linked to inspiration for bio-inspired algorithms. Based off the framework and adapted from [35].

adjusting weight vectors for concave problems [81] or using predefined reference points to classify and select solutions in many-objective problems [43], leaving potentially beneficial concepts of the Extended Evolutionary Synthesis unexplored. Using a biological framework for information transfer [35], Figure 2.3 shows how the genetic and cultural sources of information are closely linked to Evolutionary Computation and Swarm Intelligence respectively. The figure shows the biological elements of information transmission that leads to variation between phenotypes. Phenotypic plasticity is the ability for the same genotype to produce different phenotypes in response to epigenetic or environmental conditions [97]. On the far left, genetic inheritance has low plasticity as genotypes take many generations to mutate and evolve. Genetic inheritance by itself cannot react to sudden changes to the environment and adapt the genotype immediately. Cultural and non-transmitted information leads to higher phenotypic plasticity (higher variation) in nature as adaptation can occur quickly within a few generations, for example from social learning. On the far right, non-transmitted information is information that is not inherited by future generations. The mechanisms further right leverage non-genetic transmission more than those on the left.

Non-transmitted information can include some algorithms that utilise a population of individuals acting with certain behaviour that does not require interaction with other individuals. Non-transmitted information has the highest plasticity as it can act immediately on environmental changes, but is not stable and does not carry forward to future generations.

The categories of parental and ecological inheritance are omitted for simplicity as they can be argued as a higher level of cultural inheritance from a computational perspective. For example, niche construction [139] is the concept of changing the environment to better suit the development of future generations. These changes become a part of evolution when they affect natural selection, such as burrows and nests defending family units or otherwise less fit organisms. However, niche construction can be seen as a form of cultural inheritance where the trait for creating ecological change in the environment is passed down through social learning. Similarly, parental effects can be classed as part of cultural inheritance. Parental behaviour, which affects the offspring phenotype, can be passed on through social learning, either horizontally (among the same generation) or vertically (to the next generation).

Using this framework Tables 2.2 and 2.3 splits up the categories of genetic, epigenetic and cultural inheritance shown in Figure 2.3 into mechanics from a computational perspective.

2.3 Multi-objective optimisation problems

Evolutionary Computation algorithms are often used to solve closed box optimisation problems. Multi-objective optimisation problems have more than a single objective function to be optimised. Multi-objective optimisation typically refers to problems with $2-3$ objectives. Problems with more than 4 objectives are referred to as many-objective optimisation problems [43]. The different objectives for optimisation can often conflict with each other. Therefore, rather than finding a single optimal solution to the problem, the concept of Pareto optimality is applied to approximate a set of solutions with the best possible trade-offs between all the objectives. Solutions are considered Pareto optimal when there are no other solutions that can improve in one objective without sacrificing a different objective.

A multi-objective optimisation problem can be defined as follows.

$$\begin{aligned} \text{Minimise or maximise} \quad & F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_o(\mathbf{x})) \\ \text{subject to} \quad & g_1(\mathbf{x}) \leq 0, g_2(\mathbf{x}) \leq 0, \dots, g_p(\mathbf{x}) \leq 0 \\ & h_1(\mathbf{x}) = 0, h_2(\mathbf{x}) = 0, \dots, h_q(\mathbf{x}) = 0 \end{aligned} \tag{2.1}$$

where:

- $F(\mathbf{x})$ are the objective functions, up to o objectives.
- $\mathbf{x}$ is a solution to the optimisation problem, defined as a vector of decision variables, $x_1, x_2, ..., x_d$, up to d number of variables

- $g_i(\mathbf{x})$ for $i = 1, 2, ..., p$ are p inequality constraints that define the boundaries of the feasible region in decision space.
- $h_j(\mathbf{x})$ for $j = 1, 2, ..., q$ are q equality constraints that restrict the feasible region to specific values.

The fitness of a solution $\mathbf{x}$ is determined by evaluating the objective functions $F(\mathbf{x}) = (f_1(\mathbf{x}), f_1(\mathbf{x}), ..., f_o(\mathbf{x},))$, which map the decision variables to the objective space. In multi-objective optimisation, fitness is not a single scalar value, but a vector of objective values, each reflecting an objective function to form the trade-offs between conflicting objectives. To compare solutions in a multi-objective problem, the concept of dominance is used to identify which solutions may be "better" than another accounting for multiple objectives simultaneously. The standard definition of dominance is Pareto dominance. It is defined as solution $\mathbf{x}_i$ dominates another solution $\mathbf{x}_j$ if:

- $\mathbf{x}_i$ is no worse than $\mathbf{x}_j$ on all objective functions.
- And $\mathbf{x}_i$ is strictly better than $\mathbf{x}_j$ on at least one objective function.

Solutions dominated under this definition are considered to be weakly dominated. For solutions to be strictly dominated, it must always be better in all objectives. Furthermore, the concept of ϵ-dominance extends Pareto dominance to allow for a small threshold parameter ϵ, so a solution $\mathbf{x}_i$ ϵ dominates another solution $\mathbf{x}_j$ if:

- $f(\mathbf{x}_i)$ is no worse than $f(\mathbf{x_j}) + \epsilon$ for all objectives.
- And $f(\mathbf{x}_i)$ is strictly better than $\mathbf{x}_j - \epsilon$ on at least one objective.

The ϵ threshold creates a tolerance in which solutions are allowed to be slightly worse in some objectives as long as it improves significantly, by at least ϵ, in at least one objective.

Solutions which are not dominated by another solution in the population are called non-dominated solutions. A non-dominated set is the set of solutions where none of the solutions in the set are dominated by any other solution within that set. It represents the best solutions found within the current population so far. The non-dominated set at the end of an algorithm's optimisation is the best approximation of the true Pareto set found by the algorithm. The true Pareto optimal set of solutions is the set of all possible solutions in the decision space that are non-dominated, representing the globally optimal solutions in the sense that no other solutions dominate them. The Pareto front is then the Pareto set within the objective space, representing the optimal trade-offs between objectives, where improvement in one objective would lead to worsening in at least another objective.

Decision variables $\mathbf{x} = (x_1, x_2, ..., x_d)$ can be represented in different forms depending on the problem and algorithm. The most common representations are real number and binary representations. In real number representations, each x_i is a real number, typically constrained to a specific range of values. This is most suitable for continuous optimisation problems such as engineering design problems. For binary representations, a binary value of 0 or 1 is used, which is suitable for discrete decision spaces or combinatorial problems. Other representations can be used for different types of problems, for example integer representations for discrete problems such as scheduling or resource allocation, tree-based representations where solutions are represented as a hierarchical tree structure for network design and optimisation, or permutation representations where decision variables represent a permutation of elements for ordering and sequencing problems. The choice of representation is often based on the problem to be optimised, and impacts the evolutionary operators used. In this thesis, real number representation is the default representation used unless stated otherwise.

2.3.1 Evolutionary operators

The canonical GA as formalised by Goldberg [69], serves as a foundational framework for Evolutionary Computation. It typically operates on a population of binary-encoded solutions, using selection, crossover, and mutation as the operators to evolve populations of solutions over multiple generations. There is a fitness-based selection mechanism, a single-point or multi-point crossover, and a bit-flip mutation. This canonical form assumes a single-objective optimisation problem, where the fitness value directly correlates to solution quality. In the context of multi-objective optimisation, as discussed here, these operators are adapted to handle multiple objectives.

The evolutionary operators for genetic, cultural, and epigenetic inheritance described in tables 2.2 and 2.3 can be further defined under the context of multi-objective optimisation problems.

Selection is the process of choosing a set of parent solutions from the current population to reproduce part the next generation based on their fitness, inspired by natural selection. For multi-objective problems, selection may be based on Pareto dominance. Solutions that are not dominated by any others in the population form the Pareto front, and selection prioritises these non-dominated solutions to approximate the optimal Pareto set. To maintain diversity and prevent premature convergence, additional criteria like crowding distance can be used to measure the density of solutions in the objective space, and favour those in less crowded regions. Selection is based on the objective space, as the objective value of each solution makes up the selection criteria. The selection process can be defined as:

- Let P be the current population of solutions, where the objective values of a solution $\mathbf{x}$ is $F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_o(\mathbf{x}))$ for o objectives.
- A selection operator S takes P as input to create a set of parent solutions P' with a size $\leq P$ based on the selection criteria. For example, in algorithms like NSGA-II,

solutions are ranked by non-domination levels. Then within the same rank, solutions with higher crowding distance are preferred to ensure diversity. The operator S selects the top-ranked solutions, often using tournament selection, until P' is filled.

Recombination, also called crossover, takes a pair of the parent solutions from P' as input, using their decision variables to create new child solutions. There are many different methods of crossover such as single-point crossover, where a single point from each parent's decision vector $\mathbf{x}$ is chosen and two child solutions $\mathbf{x}'_1$ and $\mathbf{x}'_2$ inherit the decision variables up to the single point from one parent, swapping the remaining variables with the other parent, or multi-point crossover where multiple crossover points are used. In real number representations, Simulated Binary Crossover (SBX) [42] is often used. The SBX operator mimics the average and spread properties of binary single-point crossover, ensuring the mean of the child variables equals the mean of the parent variables, and the spread factor β_i follows a polynomial probability distribution. SBX is applied to each decision variable independently with a crossover probability p_c. For each variable x_i, a random number $r \in [0, 1]$ is generated, if $r \leq p_c$, crossover is performed, otherwise the child inherits the parent's variable value unchanged. To follow the average property, the values of the two child solutions $\mathbf{x}'_1$ and $\mathbf{x}'_2$ created through recombination of the two parent solutions $\mathbf{x}_1$ and $\mathbf{x}_2$ are as follows:

$$\begin{aligned} \mathbf{x}'_1 &= 0.5(\mathbf{x}_1 + \mathbf{x}_2) - 0.5\beta(\mathbf{x}_2 - \mathbf{x}_1) \\ \mathbf{x}'_2 &= 0.5(\mathbf{x}_1 + \mathbf{x}_2) - 0.5\beta(\mathbf{x}_2 - \mathbf{x}_1) \end{aligned} \tag{2.2}$$

The **mutation** operator adds random changes to a child solution's decision variables. In binary representations, this could be done as a bit-flip. Polynomial mutation is an example of a mutation operator used in real number representations. Given a solution with a decision vector $\mathbf{x} = x_1, x_2, ..., x_d$, polynomial mutation works by applying a polynomial distribution to each decision variable in x. Mutation is applied to each decision variable independently with a mutation probability p_m, typically $p_m = 1/d$, where d is the number of variables to ensure one mutation per solution on average. For each variable x_i, a random number $r \in [0, 1]$ is generated and if $r \leq p_m$, the mutation is performed.

The mutated variable is calculated as:

$$x'_i = x_i + \delta_i \cdot (\text{bound}_{i,\max} - \text{bound}_{i,\min} \tag{2.3}$$

where $\text{bound}_{i,\min}$ and $\text{bound}_{i,\max}$ are the variable's bounds, and the perturbation δ_i is:

$$\delta_i = \begin{cases} (2r)^{1/(\eta_m+1)} - 1, & \text{if } r < 0.5, \\ 1 - [2(1-r)]^{1/(\eta_m+1)}, & \text{otherwise,} \end{cases} \tag{2.4}$$

with $r \in [0, 1]$ a random number and $\eta_m \geq 1$ (e.g., 20) the distribution index. A larger

η_m produces smaller perturbations.

2.3.2 Cultural operators

For the cultural operators of specialised roles, individual social learning, and population social learning, inheritance can be modelled as the process of knowledge transfer within the population. Decision variables or learning strategies can be altered based on the knowledge transfer, typically via a set of leaders or best solutions found so far.

In individual social learning, any individual $\mathbf{x}$ can update is strategy or variables based on information gathered and shared by other individuals. The best solutions $\mathbf{x}^*$ can be chosen based on Pareto dominance, creating a set of non-dominated solutions. In the case a single solution is used, the best can be chosen from this set based on a second desirable metric such as crowding distance to measure diversity. Other individual solutions $\mathbf{x}_i$ can learn from the best solutions $\mathbf{x}'_i = L(\mathbf{x}_i, \mathbf{x}^*)$ where L is the learning strategy to be updated. The different learning strategies and how they interact with function objectives and decision variables are shown later for the different Swarm Intelligence algorithms.

In the case of population-level social learning, information is transferred between populations rather than individuals. Take two populations (or sub-populations) P_1 and P_2, all individuals from one population can learn from some or all individuals in the other population $P'_1 = L(P_1, P_2)$. The population may be divided for specialised roles, where each role focuses on a specific aspect or subset of the multi-objective optimisation problem. Depending on the roles, each may have separate operators or objective functions to focus on.

2.3.3 Epigenetic operators

As epigenetic inheritance operates on top of genetic inheritance, epigenetic operators would be able operate on individuals on top of the genetic operators.

Mitotic epigenetic inheritance is the transmission of epigenetic marks from parents to children. The marks can be represented as additional information on top of an individual solution $\mathbf{x}$. The decision vector $\mathbf{x} = x_1, x_2, ..., x_d$ would have an associated vector of epigenetic marks $\mathbf{e} = e_1, e_2, ..., e_d$ to cause variation in the decision variables in the presence of environmental cues. Depending on the epigenetic process involved, the marks could cause, prevent, or reverse the effects of recombination and mutation.

In the context of multi-objective optimisation, germline epigenetic inheritance is the propagation of changes made to the decision variables in following generations, in other words the inheritance of the epigenetic marks $\mathbf{e}$.

2.4 Categorising algorithms

Using the definitions of multi-objective optimisation problems and their operators, Evolutionary Computation algorithms can be described and broken down into each category of inheritance. Table 2.5 breaks down each category of inheritance into the biological concepts defined in Table 2.2, to show the overlap between the selected algorithms. Due to the number of algorithms in the available literature, a representative set were chosen based on a combination of performance, popularity, and to cover a range of different mechanics in each algorithm family. While the use of citation count as a measure of popularity is not a rigorous robust metric, but it indicates an estimate of the the uptake of the algorithms in real world problems, and how often they are used in benchmarking test problems. The full list of algorithms, and why they were chosen, are shown in Table 2.4. Multi-objective versions of all algorithms are described in the following sections. While many of the original algorithms are single-objective, the theory behind them is described generally then related back to the multi-objective optimisation problem.

There is a split between Evolutionary Algorithms, which use the evolutionary operators of selection, crossover, and mutation operators, and Swarm Intelligence algorithms, which utilise cultural operators of specialised roles for individuals, different forms of social learning and communication. While some algorithms fit neatly into Evolutionary Algorithms or Swarm Intelligence based on the operators they use, contrasting with evolutionary theory where multiple mechanisms act together, there are a number algorithms that use a spread of multiple biological mechanisms together: cMLSGA uses specialised roles and social learning, while GB-ABC and Firefly Algorithm utilise elements of selection; HEIA uses co-evolution to spread social learning; and Cuckoo Search uses crossover and mutation genetic operators. Some other algorithms such as the Univariate Marginal Distribution Algorithm (UMDA) [145] and Compact Genetic Algorithm (cGA) [82] fall under the category of Estimation of Distribution Algorithms (EDAs), which do not fit into the same biological categorisations as the other Evolutionary Algorithms and Swarm Intelligence algorithms as they use explicit probability distribution models over biological mechanisms.

Table 2.2: Biological categories of transmission for the genetic and cultural levels of evolution.

Category	**Sub-category**	**Description**
Genetic	Selection	In each generation, a subset of the population is chosen to reproduce the next generation. In biological terms, this relates to the survival of the fittest. Individuals with fitter genes are typically able to find mates more easily. In computational terms, individuals are selected based on their fitness against the objective function.
	Recombination	Also called crossover, when two individuals produce offspring, genetic material from both parents are used. This results in offspring that share some attributes with both parents. Crossover affects both diversity (avoiding inbreeding depression) and convergence of the population as it is able to create new combinations while keeping the same values from its parents. However, without an addition form of diversity such as mutation, it becomes closely linked to the epistasis of the environment.
	Mutation	When offspring are formed, mutation can occur. Depending on the environment, these mutations could be neutral, beneficial or detrimental to the survival of that individual. Mutation helps improve diversity by producing values that have not been explored before by the population.
Cultural	Specialised roles	Individuals in a population have different roles to fulfil. For example, scout and guard bees in a colony, or mongooses that form foraging niches [169] due to competition within the population. In some cases, specialised roles are formed and last through multiple generations, as in the case of ants and bees. In others such as the mongooses, the specialised roles may only form in a particular generation due to external factors during the generation's lifespan.
	Social learning (individual level)	Individuals learning from other individuals in the population through direct information sharing, teaching, environmental stimulus or imitation and emulation of other individuals.
	Social learning (population level)	Information transfer between populations (cultures) where different populations may have different variations in behaviour as a result of social learning. Analogous to genetic drift in classical genetic theory.

Table 2.3: Biological categories of transmission for epigenetic inheritance.

Category	**Sub-category**	**Description**
Epigenetic	Mitotic	The transmission of epigenetic marks directly from the parent generation to the offspring generation. Epigenetic marks allow variation to occur in an individual when the marks are triggered as a response to environmental cues. Mitotic epigenetic inheritance only transfers the epigenetic mark across one generation.
	Germline	Epigenetic changes caused by environmental factors in the parent are passed down to the offspring and to further generations [190]. This affects across multiple generations even if the environmental factor that triggered the change only happened during the first generation.
	Experience-dependent	Epigenetic marks that influence parental behaviour causing the same epigenetic mark to appear in the offspring generation. The marks can persist across multiple generations, but the transmission can also be broken when environmental factors cause one generation to stop the same parental behaviour. For example maternal behaviour in rodents cause the offspring to exhibit the same behaviour to their offspring. But if there is a break in one generation where the maternal behaviour does not occur, the epigenetic transmission stops [25].

Table 2.4: Evolutionary algorithms are chosen based on a combination of popularity via citations, performance on benchmarking problems, application in real-world problems, and for a variety of different computational mechanics and algorithm families.

Algorithm	**Reason**
NSGA-II [45]	Most popular multi-objective Genetic Algorithm with good performance, over 50,000 citations, and uses a crowding niching approach
MOEA/D [217]	A popular decomposition-based family of multi-objective Genetic Algorithm.
IBEA [223]	An indicator-based Genetic Algorithm using an indicator for selection, which contrasts with the common Pareto dominance ranking used for selection.
DE [180]	Original differential evolution algorithm which many CEC single objective competition winners are based off of.
CMA-ES [80]	An Evolution Strategy approach that uses a unique mechanic of a covariance matrix to learn and adjust a mutation distribution.
HEIA [116]	A multi-objective algorithm that utilises mechanics from Immune algorithms and co-evolution.
cMLSGA [74]	A multi-objective algorithm that shows good performance on real world problems [75] using multilevel selection and co-evolution.
SMPSO [136]	Popular variant of the particle swarm optimisation algorithm.
ACO [51]	A popular Swarm Intelligence algorithm with over 18,000 citations used primarily for path finding and routing problems with a unique method of solution construction based on probabilities.
ABC [3]	Algorithm based on the foraging behaviour of bees with over 9,000 citations and a wide range of real world applications [99].
Firefly Algorithm [208]	Swarm Intelligence algorithm based on the flashing behaviour of fireflies with an emphasis on multimodal problems.
Cuckoo Search [206]	Based on the Cuckoo species laying eggs in the nests of other birds with over 9,000 citations and large number of applications due to its simplicity and few parameters, though there is a lack of mathematical and benchmarking analysis [168].
GWO [129]	Algorithm inspired by the hunting behaviour of grey wolves with over 17,000 citations and applications in power engineering, image processing, and bioinformatics [61]

Table 2.5: How bio-inspired algorithms fit into a biological framework for different forms of inheritance.

Algorithm category	**Genetic**			**Epigenetic**			**Cultural**		
	Selection	Recombination	Mutation	Mitotic	Germline	Experience-dependent	Specialised roles	Social learning (individual)	Social learning (population)
NSGA-II [45]	✓	✓	✓						
MOEA/D [217]	✓	✓	✓				✓		
HEIA [116]	✓	✓	✓				✓		
cMLSGA [74]	✓	✓	✓				✓		✓
CMA-ES [80]	✓	✓	✓						
SMPSO [214]								✓	
ACO [51]							✓	✓	
ABC [98]								✓	
GWO [129]							✓	✓	
Firefly Algorithm [208]	✓							✓	
Cuckoo Search [206]	✓							✓	

A justification of Table 2.5 is given in the following subsections. The algorithms abbreviations are listed as follows:

- NSGA-II: Non-dominated Sorting Genetic Algorithm II
- MOEA/D: Multiobjective Evolutionary Algorithm Based on Decomposition
- CMA-ES: Covariance Matrix Adaptation Evolution Strategy
- HEIA: Hybrid Evolutionary Immune Algorithm
- cMLSGA: Co-evolutionary Multilevel Selection Genetic Algorithm
- SMPSO: Speed-constrained Particle Swarm Optimisation
- ACO: Ant Colony Optimisation
- ABC: Artificial Bee Colony
- GWO: Grey Wolf Optimiser

2.4.1 Genetic Algorithms

Genetic Algorithms (GA) are a branch of Artificial Intelligence, which primarily solve incomplete search or optimisation problems. First suggested by Turing [188] as a means to apply evolution as part of a learning machine, GAs were popularised by Holland [85] and now used in a variety of real world applications such as finance, data analysis, medicine, manufacturing, multi-criteria decision-making, and decision-making under uncertainty [14, 63, 125, 140, 167].

GAs are inspired by natural selection and evolution, evolving a population of potential solutions towards an optimal solution. Solutions are coded and represented as a chromosome of genes, in some binary form or real number form. Figure 2.4 shows the basic flow of a Genetic Algorithm. The selection, crossover, and mutation evolutionary operators are applied to the population in each generation to create the next generation.

In addition to these operators, many GAs also incorporate other mechanics such as elitism. Elitism [92] is a selection operator that keeps the best individuals in a population, copying them directly into the next generation to improve convergence at the cost of diversity. In multi-objective optimisation, the population of elite individuals can be represented by the non-dominated solutions found so far.

Niching approaches

There are a number of niching methods used within genetic algorithms for the purpose of improving the population diversity [171] such as fitness sharing, crowding, clearing, and

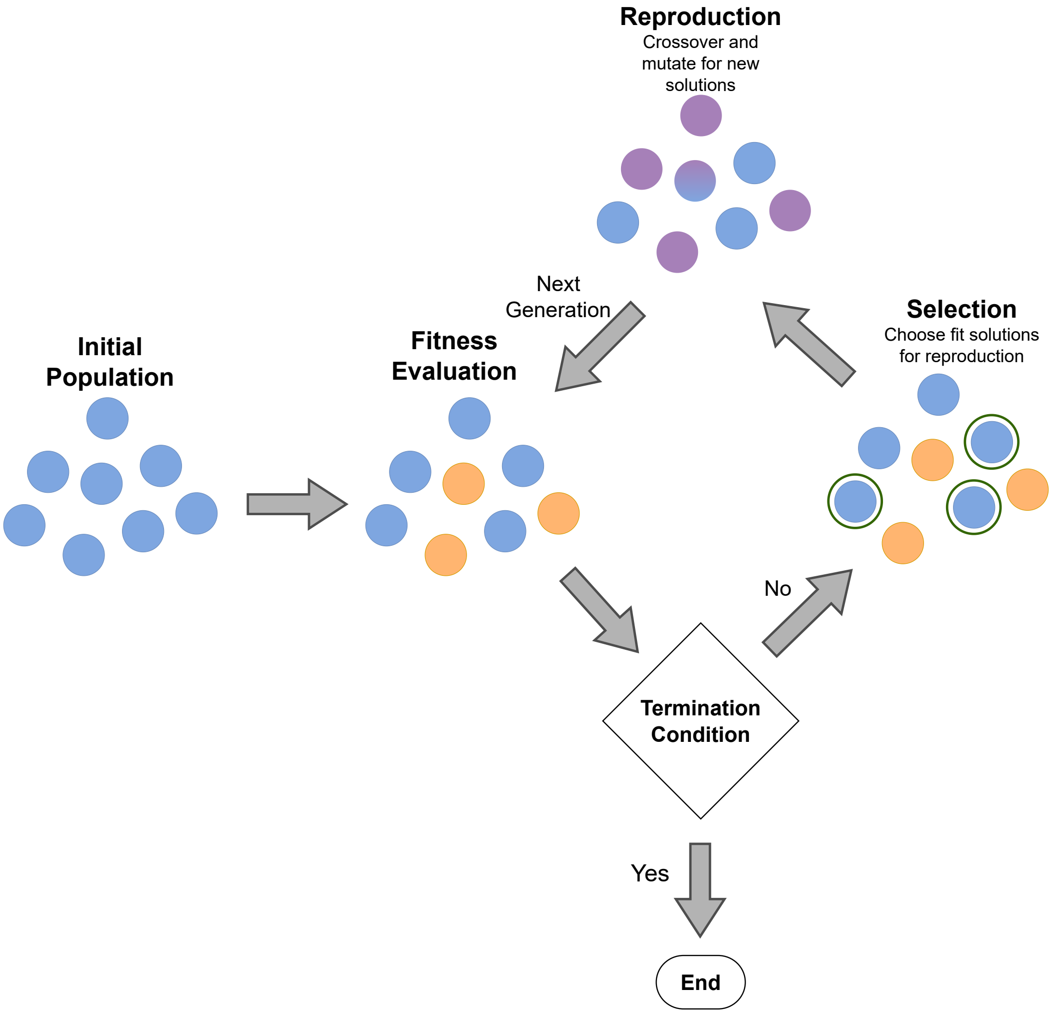


Figure 2.4: Mechanics of the classic GA. The blue circles are individuals in the population. During fitness evaluation, fit and less fit (yellow) individuals are identified. Next, individuals are chosen for selection in green outline. In algorithms with elitism, some of these individuals may be kept in the next generation. New individuals are created during reproduction (in purple) through crossover of the selected parents, with a small chance of mutation, represented by the purple and blue gradient.

clustering methods. The well-known NSGA-II [45] and NSGA-III [43] genetic algorithm uses a crowding method to quantify the density of surrounding solutions. The algorithm use two main mechanisms: an efficient form of non-dominated sorting, and a density estimator with crowding distance. The non-dominated sorting is used to rank the population into hierarchies of fronts based on Pareto dominance, where individuals with the highest ranks have a higher probability for selection. A crowding distance metric is then used to maintain diversity in the population by encouraging search in less dense regions of the search space.

NSGA-II improved upon the computational complexity of NSGA by calculating the dominance relation between pairs only once, and calculating the distance between close individuals. This reduced the time complexity of the dominance ranking to $O(MN^2)$ and the crowding distance to $O(MNlogN)$. Although NSGA-II has since been outperformed by newer algorithms, it is still the most popular Genetic Algorithm used in a large number of real world applications, and is consistently used as the baseline algorithm to benchmark against.

In 2014, Deb and Jain developed the next iteration of the algorithm, NSGA-III [43]. NSGA-III improved upon NSGA-II's weak performance on complex problems with more than two objectives. It uses predefined reference points instead of crowding distance as the density estimator to maintain diversity. The points are either provided by a user, or defined using a systematic approach to return a distributed set of points along the Pareto Front. Benchmarking showed the improved performance of NSGA-III on problems with many objectives [43, 166], but an issue in retaining strong performance for 1 or 2 objective problems. The reliance on these reference points meant NSGA-III was affected by low quality reference points.

The methods of NSGA-II and NSGA-III were combined into a unified algorithm U-NSGA-III in 2015 [166]. It used both the crowding distance calculation from NSGA-II and the reference points from NSGA-III during selection. This allowed for higher population sizes and improved performance on 1 and 2 objective problems compared to NSGA-III with a small decrease in performance on the higher objective problems. The main drawback of this branch of solvers is the restricted scalability in objective function space [49] which requires a large population size to overcome.

Decomposition based approaches

The MOEA/D framework converts a multi-objective problem with M objectives into N sub-problems and assigns a weight vector for each sub-problem. A neighbourhood is defined for each specific sub-problem as the set of T sub-problems closest to it based on Euclidean distance. During reproduction, individuals can only reproduce with individuals within the same sub-problem or the same neighbourhood. Compared to other methods, decomposition simplifies the difficulty of maintaining diversity and allocating fitness, as each sub-population explores different regions of the search space. MOEA/D showed strong performance in the CEC'09 unconstrained benchmark problems [218] compared to NSGA-II, the most popular Genetic Algorithm at the time. Though it did not compete on the constrained set of problems. The original MOEA/D framework also has a few limitations, which spawned a number of variants aimed at addressing and refining these limitations [205]:

- The use of fixed weight vectors is not suitable for problems with complex Pareto front shapes.
- It is difficult to choose a suitable neighbourhood size for a given problem, as large and small neighbourhoods have their advantages and disadvantages.

- Evolutionary operators that are not suitable for a range of different problems.

These limitations divide the MOEA/D variants into distinct categories for improvement: decomposition methods [88, 120, 123, 194], weight vector generation [81, 113, 219], and evolutionary operators [28, 114, 195]. New reproduction operators such as DE (Differential Evolution) have also been included into MOEA/D to improve its performance [114]. Jiang et al. [94] used two new decomposition methods that found strong performance improvements: multiplicative scalarizing function (MSF) and penalty-based scalarising function (PSF). Further variants aim to extend the algorithm to solve many-objective problems [27, 34, 183], and constrained multi-objective problems [57, 58, 93] more effectively. Compared to other approaches MOEA/D puts a large focus on mathematical methods and optimising for specific types of problems. It outperforms other methods in unconstrained and dynamic problems [120] by prioritising convergence over diversity, but is less effective on constrained and discontinuous problems.

Co-evolutionary approaches

Another strategy that is used is to combine multiple methods in a hybrid or co-evolutionary way with the goal to take advantage of the strengths of different approaches while mitigating the weaknesses. A co-evolutionary Genetic Algorithm was first introduced by Potter and De Jong in 1994 [153] as cooperative co-evolution, where different algorithms worked together, sharing information between their sub-populations. Competitive co-evolution was proposed in 1997 [162] where sub-populations would compete against each other to reproduce. Decomposition has also been used to improve convergence by working on a different sub-problem, then sharing the information to form solutions.

In 2009, Goh and Tan developed COEA (Competitive-Cooperative Coevolutionary Algorithm) [68]. The algorithm decomposes a multi-objective problem into single-objective problems, and solves each with distinct populations. Competition is used for individuals to fight for a place in each population, and cooperation is used to combine the separate populations together to form valid solutions. COEA exhibited strong performance for dynamic problems, but weaker performance on discontinuous problems due to low diversity.

In 2016, BCE [115] and HEIA [116] were introduced as high diversity hybrid methods. BCE used two different fitness calculation methods, a Pareto-based criterion (PC) for convergence, and a Non-Pareto-based criterion (NPC) for diversity. PC used the normal Pareto domination ranking while NPC used the Hypervolume [221] metric as its selection criterion. HEIA used two different evolutionary strategies on its sub-populations. Although SBX [42] and DE [179] were used in the original paper, HEIA was designed as a framework to allow any two evolutionary strategies to be used. Instead of decomposing the problem, the two strategies simply worked on different sub-populations. Then after each generation, the best individuals from each sub-populations were saved in an external archive and the sub-populations are cloned for the next generation with an Immune Algorithm. HEIA showed strong performance across a range of benchmarks due to its emphasis on diversity. However, it has not been tested with a range of different evolutionary strategies in place of SBX and DE to see what performance benefits or loses occur on different types of test problems.

Indicator-based approaches

Finally, another useful approach is to use a performance indicator when selecting individuals. This provides a concrete guide for algorithms to know when particular individuals are closer to the Pareto front and should be selected for reproduction. These indicator based algorithms draw parallels with non-genetic inheritance mechanisms from biology, where individuals are affected by signals in the environment.

Zitzler and Künzli introduced the IBEA framework [223] in 2004 that used this method. An indicator can be added to the fitness assignment of individuals to directly affect their fitness based on their performance using a given indicator. With an indicator that applies to the distribution of solutions, additional diversity preservation mechanisms like the crowding distance in NSGA-II are not required. Although, the convergence rate of IBEA does not hold up to other state-of-the-art algorithms by itself, it has been shown to improve performance when combined with other algorithms in a hybrid method like BCE [115].

Multi-level selection approach

Although the levels of selection is an important aspect in the evolutionary biology community, the first GAs to implement the concept did not find strong performance improvements over existing algorithms. A multi-level selection algorithm was developed by Lenaerts et al. [110] where selection occurs at the group level when there is a large enough variation between different groups. However, all groups were re-assigned in each generation, losing the advantages of group selection across multiple generations. Subsequent methods of integrating a multi-level theory into GAs [2, 157] also fell short in improving performance over the mainstream niching and decomposition based methods.

In 2017, Grudniewski and Sobey developed MLSGA (Multi-Level Selection Genetic Algorithm) [72] and included co-evolutionary mechanics in the next iteration cMLSGA (Co-evolutionary Multi-Level Selection Genetic Algorithm) [74]. This version of implementing a multi-level selection concept into GAs was successful, as only one group of individuals is removed in each generation, and separate mechanisms are used for its sub-populations. MLSGA groups individuals into sub-populations called collectives. Collectives are formed based on the similarities between the individuals. The individual and collective levels used different mechanisms, instead of different mechanisms on each collective. MLSGA further defines fitness separately for the individuals and collectives to maintain diversity. In cMLSGA, different collectives now used different Evolutionary Algorithms to introduce competitive co-evolutionary pressure on the collectives. After a number of generations evolving independently, the collective with the worst fitness is eliminated and repopulated with the best individuals from the other collectives. Similar to HEIA, cMLSGA provides a framework for different Evolutionary Algorithms to be used.

Evolution Strategies

Evolution Strategies [18] use the mechanics of selection, recombination and mutation to evolve the population. However, unlike traditional GAs, Evolution Strategies are self-adapting with a distinct set of endogenous and exogenous parameters. Endogenous parameters are evolved along with an individual solution and passed down to offspring individuals, independent of others in the population. This contrasts with GAs where all parameters such as crossover and mutation rate are set beforehand, making them all exogenous. Endogenous parameters control the strategy of ES algorithms by changing the statistical properties for mutation. There are two steps of recombination and mutation for an individual i in each generation:

- Recombination to form the endogenous strategy parameters $s(i)$.
- Recombination to form the solution variables $y(i)$.
- Mutation of the strategy parameters.
- Mutation of the solution variables using the strategy parameters.

CMA-ES [80] is an improvement to the traditional ES algorithm. It uses a covariance matrix to generate its mutation distribution which is adaptive and evolves with the search. This allows the strategy parameters to adapt more closely during local search. The parameters are adapted based on statistics gathered through previous generations. The self-adaptive endogenous parameters can be seen as a form of evolvability. The mechanism of passing down and evolving this extra set of parameters changes the capacity of evolution in the population, as the mutation rate is changed and adapted based on previous generations. As evolvability is similar to tuning hyperparameters, the use of a set of self-adapting parameters to control mutation automates tuning of the mutation rate. While CMA-ES shows strong performance, it is mostly used in dynamic and single-objective problems making it difficult to compare to state-of-the-art multi-objective algorithms. Despite its additional features with parameter control, it does not make use of any cultural or epigenetic forms of inheritance.

Differential Evolution

Differential Evolution [180] (DE) is another branch of Evolutionary Algorithms that is similar to GAs. Selection, recombination and mutation operators are all used in DE. The main difference between GAs and DE algorithms is the representation of the individual. DE uses real value vectors rather than binary encodings in traditional GAs, though modern multi-objective GAs such as NSGA-II and MOEA/D also use real values directly rather than a binary encoding. Hybrid combinations of GAs and DEs have shown to provide benefits in algorithm performance as shown in MOEA/D-DE [114].

Memetic algorithms

Memetic algorithms [118] are inspired by the human information concept of memes [41] as a simple unit of knowledge that can be transferred and evolved in human society and culture. Memetic algorithms take this concept and apply it onto an evolutionary framework. Genetic algorithm concepts of selection and recombination are used, but instead of mutations to make additional changes to offspring solutions, further local search techniques such as simulated annealing [181] are used to improve the fitness of offspring solutions as much as possible before the cycle is repeated for the next generations. The addition of local search heuristics to improve offspring solutions is more similar to the concepts of evo-devo and phenotypic plasticity than epigenetics or cultural information transfer. Multi-objective memetic algorithms such as M-PAES (memetic-Pareto Archived Evolution Strategy) [104] borrows the use of a nondominated archive of solutions, and combines them with local search optimisers. These algorithms use selection and recombination mechanisms with a variety of local search techniques that can be considered to be plastic or evolutionary developmental approaches, but would not be considered epigenetic, as the local search behaviour does not change according to environmental changes. Hybrid memetic algorithms that utilise PSO [118] or other Swarm Intelligence algorithms as their local search method could be considered to include cultural elements of social learning.

2.4.2 Swarm Intelligence

Swarm Intelligence algorithms are made up of optimisation techniques inspired by collective group behaviour of biological organisms and systems. Solutions are represented as individuals within the group that acts together to search for the optimal solutions, for example, simulating the behaviour and movement of ants leaving pheromone trails [51] or the hunting behaviour of grey wolf packs [129]. The abundance of Swarm Intelligence algorithms based on a wide assortment of biological systems has created criticism over their true novelty as the mathematical models are obscured by metaphoric language [174]. Here, the algorithms are presented with their mathematical formulations to distinguish their differences and emphasise their similarities.

Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) is an algorithm developed in 1995 by Eberhart and Kennedy [100] for optimising nonlinear functions. The algorithm is based on the concept of social behaviour and sharing information. In the classic PSO, a population of particles represent potential solutions in the search space. Then, in each iteration, each particle updates its velocity and position in the search space. The velocity update is based on a combination of the best position found so far by the particle, and the best position found so far by the entire swarm. Given the decision variables $\mathbf{x} = x_1, x_2, ..., x_d$ for d variables, each variable is updated by adding a velocity v to its value. The new value of variable x_i'

in each iteration is calculated as:

$$x'_i = x_i + v_i \tag{2.5}$$

where

$$v'_i = \omega v_i + c_1 r_1 (pbest_i - x_i) + c_2 r_2 (gbest_i - x_i), \tag{2.6}$$

pbest is the best solution found by the particle; **gbest** is the best solution found by the entire swarm; both are decision vectors where $pbest_i$ and $gbest_i$ represent the ith decision variable. v'_i is the updated velocity for the ith decision variable, replacing the previous velocity v_i; $\mathbf{v} = v_1, v_2, \ldots, v_d$ is the velocity vector for d variables; ω is the inertia weight, controlling the influence of the previous velocity and typically set around 0.5 or linearly decreased (e.g., from 0.9 to 0.4) over iterations to balance global exploration early and local exploitation later [170]; c_1 and c_2 are coefficients that balance the movement towards personal and global bests found so far; finally, $r_1, r_2 \in [0, 1]$ are random numbers drawn uniformly for each particle and iteration. The velocity update combines three components: the inertia term (ωv_i) maintains the particle's previous direction, the cognitive term ($c_1 r_1 (pbest_i - x_i)$) pulls the particle toward its personal best, and the social term ($c_2 r_2 (gbest_i - x_i)$) pulls it toward the global best. Higher c_1 emphasises exploration based on personal experience, while higher c_2 prioritises convergence toward the swarm's best solution. The best positions *pbest* and *gbest* are updated after every iteration based on the fitness function. For multi-objective optimisation, **pbest** and **gbest** may be selected using Pareto dominance or other criteria such as crowding distance. The learning strategy of social learning $\mathbf{x}'_i = L(\mathbf{x}_i, x^*)$ for PSO is the velocity update in equation 2.5, with x^* being the *pbest* and *gbest* solutions.

Particles in the swarm can be connected to each other in any graph topology to share the best global and local solutions found. Similar to Genetic Algorithms, there are multiple approaches to solving multi-objective problems using PSO. They can be grouped as follows:

- Aggregating approaches - Aggregating approaches aim to combine the objectives in a multi-objective problem in a single objective, in direct contrast to the decomposition-based approach of Genetic Algorithms. Parsopoulos and Vrahatis [144] developed a multi-objective PSO using different methods of weighted aggregation, with fixed, dynamic and random weights. However, the choice of weights is dependent on each problem and are difficult to choose. Furthermore, the approach only finds one solution per run, as the aggregated single-objective problem will only have one solution, making it computationally inefficient.
- Lexicographic ordering - The lexicographic ordering approach required user input to rank objectives in order of importance. Each objective function could then be solved separately in order of importance. Hu and Eberhart [203] introduced MOPSO that optimised one objective at a time and added an external archive [204] to improve the dynamic neighbourhood of their approach. The approach also requires user input and tuning to determine the order of importance, which is not effective for real

world applications with no a priori knowledge as the ordering significantly affects performance.

- Pareto-based approaches - A more common approach is the use of Pareto dominance to sort the best particles found and select leaders for the swarm. Reyes and Coello [172] also used the concepts of Pareto dominance and nearest neighbour density estimator to select leaders in OMOPSO. The algorithm uses two external archives: one to store leaders and the other to store the set of non-dominated solutions found. The algorithm outperformed a number of multi-objective algorithms including NSGA-II and MOPSO on the ZDT and DTLZ benchmark problems. Further improvement to OMOPSO was made in [53, 136] to develop SMPSO, where the velocity of particles were constrained to disallow extreme values.
- Multi-swarm approaches - Multi-swarm approaches use independent swarms for each objective, solving them as single-objective problems. The swarms may then exchange information or recombine after some generations. This is similar to co-evolutionary approaches in GAs like COEA, strictly separating each subpopulation for each objective. A multiple populations for multiple objectives (MPMO) approach was introduced in [143] where information is shared using an external archive. CMPSO was able to outperform algorithms such as NSGA-II, MOEA/D-DE and OMOPSO on ZDT problems with a low number of function evaluations and had competitive performance in DTLZ and WFG problems [216]. However, to gain the best performance from CMPSO, the external archive and elitist learning strategy used have to be tuned to balance convergence and diversity for different problems.

Ant Colony Optimisation

Ant Colony Optimisation (ACO) [51] simulates a population of ants that moves through the search space probabilistically based on pheromone trails left behind by previous generations. Pheromone values decrease with each iteration so that old trails fade and new ones form as the search space is explored. Some pheromone trails will be reinforced if the next generations continue to follow the same path, leading to higher pheromone values, which are associated with better solutions. Unlike most other Swarm Intelligence algorithms that use a form of update function as the learning strategy, based on a combination of a local or global best solutions found so far, Ant Colony Optimisation constructs each solution based on the probability of candidate values at each variable, which is determined by the pheromone trails. To construct a solution $\mathbf{x}$, the probability that a value v_i from a set of V candidate values will be selected for x_i is

$$P(v_i) = \frac{[\tau(v_i)]^\alpha \cdot [\eta(v_i)]^\beta}{\sum_{v_k \in V} [\tau(v_k)]^\alpha \cdot [\eta(v_k)]^\beta} \tag{2.7}$$

where $\tau(v_i)$ is the pheromone value and $\eta(v_i)$ is a heuristic value for the candidate value v_i. $P(v_i)$ is the probability of selecting v_i from the set of all candidates V. The parameters α

and β control the relative influence of the pheromone trail and the heuristic information, respectively.

The pheromone trails for a candidate value v_i is updated as follows:

$$\tau(v_i) = (1 - p) \cdot \tau(v_i) + \Delta\tau(v_i) \tag{2.8}$$

where p is the pheromone evaporation rate and $\Delta\tau(v_i)$ is the amount of pheromone deposited by an ant on the trail associated with the v_i value. The values of $\Delta\tau(v_i)$ and heuristic $\eta(v_i)$ may be defined differently depending on the problem and algorithm. The construction of solution variables are then based on a combination of the probability for each decision variable and a heuristic specific to the problem. To adapt ACO to multi-objective problems, there are a three key aspects that must be considered:

- The pheromone matrix – In a single objective problem, each decision variable corresponds to a pheromone value within a single pheromone matrix. In a multi-objective problem, multiple pheromone matrices could be implemented with each matrix corresponding to one of the objectives.
- Solution construction – With multiple objectives and potentially multiple pheromone and heuristic matrices, solutions can be construction through an evenly weighted combination of all these values. Weights can be applied to specific objectives and used for a subset of solutions in the population to effectively form a multi-ant colony system [65].
- Solution evaluation – Like other multi-objective algorithms, solutions are no longer evaluated by a single fitness value. Pareto-based approaches such as domination ranking can be used to satisfy multiple objectives, but other approaches such as decomposition-based [210] indicator-based methods [56] have been explored in the literature.

Although ACO uses multiple iterations/generations, there is no information transfer through genetic inheritance. The individuals in the previous generation leave behind ecological changes with pheromone trails that affect the behaviour of new generations, but there are no genetic operators in use. This can be seen as ecological inheritance, which has been defined here as part of social learning. The use of pheromone trails is core to the concept of ACO, but due to the probabilistic method of solution construction from candidate values, the algorithm is more suited to problems with discrete variable ranges.

The ACO is based on social learning at the individual level and specialised roles can be applied for multi-objective problems. The pheromones left by solutions of the previous generation would not be considered inheritance as pheromones from all ants of the previous generation are used rather than specific parents.

Artificial Bee Colony

In an Artificial Bee Colony [3] (ABC), bees are used as the biological inspiration and analogy for a multi-objective solution. The population is split into three different types of solutions: employed solutions, onlooker solutions and scout solutions. This can be viewed as three sub-populations P_e, P_o, and P_s.

- Employed solutions search for better food sources, points in the decision variable space, in their local neighbourhood and share that information to onlooker solutions in the region. They are able to remember new points if they dominate an existing point in its memory. To accomplish this, an archive of the non-dominated solutions found so far is kept as memory. For each employed solution $\mathbf{x}_i$, the learning strategy is to select a random solution $\mathbf{x}_j$ from the archive. Then the variables of $\mathbf{x}_i$ are updated by moving towards the selected archive solution:

  $$\mathbf{x}_i' = \mathbf{x}_i + w_1 \cdot \mathbf{r}(\mathbf{x}_i - \mathbf{x}_j) \tag{2.9}$$

  where r is a vector of random numbers with values ranges $[0, 1]$, and w_1 is a control parameter on the importance of the archive solution $\mathbf{x}_j$. The new decision vector $\mathbf{x}_i'$ is evaluated based on the objective functions, and if it dominates $\mathbf{x}_i$ it replaces $\mathbf{x}_i$. Otherwise a trial count is incremented.

- Onlooker solutions take the information given by employed solutions and move towards new points in decision space based on the information. For k employed solutions, each onlooker has a probability of selecting the employed solution $\mathbf{x}_i$

  $$P(i) = \frac{fit(\mathbf{x}_i)}{\sum\limits^{k} fit(\mathbf{x}_k)} \tag{2.10}$$

  with $fit(\mathbf{x}_k)$ representing the quality of the solution $\mathbf{x}_k$ as the number of other solutions that $\mathbf{x}_k$ dominates. After the selection process, the variables of the onlooker solution are updated following equation 2.9.

- Scout solutions search for new points randomly without taking into account any information, potentially replacing existing employed or onlooker solutions if their trial counts go past a certain threshold. A new random decision vector $\mathbf{x}_j$ is created. If $\mathbf{x}_j$ dominated the existing $\mathbf{x}$, it replaces $\mathbf{x}$. The trial count is reset to zero afterwards.

The scout solutions carry out the exploration phase of the search by moving randomly while employed solutions carry out an exploitative search by searching locally for non-dominated solutions. Social learning is observed as the solutions change behaviour based on information from other solutions and contextual clues from the environment. The role of an individual is not static, for example scout solutions become employed solutions if their random solution dominates an existing solution that has not improved after a number of trials. Some variants on the ABC algorithm include genetic operators (GB-ABC [142]) to improve global and local search for binary optimisation problems, but have not been adapted to multi-objective problems. The ABC algorithm uses both specialised roles and social learning between individuals.

Grey Wolf Optimiser

The Grey Wolf Optimizer (GWO) [129] is an algorithm inspired by grey wolf social structure and hunting techniques. It mimics a leadership hierarchy with four types of grey wolves: alpha, beta, delta, and omega. In a multi-objective optimisation problem, each grey wolf represents a solution, denoted as $\mathbf{x} = x_1, x_2, ..., x_d$ for d decision variables. The population is divided into the four groups:

- Alpha (α): The best solution found so far $\mathbf{x}_\alpha$
- Beta (β): The second best solution found so far $\mathbf{x}_\beta$
- Delta (δ): The third best solution found so far $\mathbf{x}_\delta$
- Omega (ω): The rest of the solutions in the population

Best solutions are chosen through a leader selection mechanism. To adapt the original algorithm for multi-objective optimisation, an archive of non-dominated solutions is maintained, and the alpha, beta, and delta solutions are chosen using roulette wheel with probabilities based on crowding distance in objective space [130]. The less crowded solutions are more likely to be chosen as the alpha, beta, and delta solutions.

The omega solutions, the rest of the population, are updated based on the positions of the alpha, beta, and delta solutions. For each omega solution $\mathbf{x}_i$, the distance vectors $\mathbf{D}_\alpha$, $\mathbf{D}_\beta$, $\mathbf{D}_\delta$ are calculated as

$$\begin{aligned} \mathbf{D}_\alpha &= |\mathbf{C}_1(\mathbf{x}_\alpha - \mathbf{x}_i)| \\ \mathbf{D}_\beta &= |\mathbf{C}_2(\mathbf{x}_\beta - \mathbf{x}_i)| \\ \mathbf{D}_\delta &= |\mathbf{C}_3(\mathbf{x}_\delta - \mathbf{x}_i)| \end{aligned} \tag{2.11}$$

where $\mathbf{C}_1$, $\mathbf{C}_2$, and $\mathbf{C}_3$ are random coefficient vectors used to control the influence of each leader solution on the position update. Each is calculated as $\mathbf{C} = 2 \cdot \mathbf{r}_1$, where $\mathbf{r}_1$ is a random vector with values in $[0, 1]^d$. Three direction vectors based on the distance vectors are then calculated and used to update $\mathbf{x}_i$:

$$\begin{aligned} \mathbf{x}'_\alpha &= \mathbf{x}_\alpha - \mathbf{A}_1 \cdot \mathbf{D}_\alpha \\ \mathbf{x}'_\beta &= \mathbf{x}_\beta - \mathbf{A}_2 \cdot \mathbf{D}_\beta \\ \mathbf{x}'_\delta &= \mathbf{x}_\delta - \mathbf{A}_3 \cdot \mathbf{D}_\delta \end{aligned} \tag{2.12}$$

$$\mathbf{x}'_i = \frac{\mathbf{x}_\alpha + \mathbf{x}_\beta + \mathbf{x}_\delta}{3} \tag{2.13}$$

$\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$ are coefficient vectors calculated as

$$\mathbf{C} = 2 \cdot \mathbf{r}_1 \tag{2.14}$$

$$\mathbf{A} = 2\mathbf{a} \cdot \mathbf{r}_2 - \mathbf{a} \tag{2.15}$$

$\mathbf{r}_1$ and $\mathbf{r}_2$ are random vectors with values [0,1] and $\mathbf{a}$ is a constant vector with values decreasing from 2 to 0.

The two coefficient vectors $\mathbf{A}$ and $\mathbf{C}$ are used to fluctuate between exploration and exploitation. When $|A| > 1$ the wolves diverge from non-dominated solutions instead of moving towards them. Similarly, $|C| > 1$ emphasises moving towards the non-domianted solutions so the population move faster towards it while $|C| < 1$ de-emphasises this movement.

There is a use of specialised roles among population to form the hierarchy and individual social learning occurs during the optimisation, when the omega solutions follow global best solutions represented by the alpha, beta, and delta solutions for direction. Information exchange between solutions is used as the dominated solutions move to position themselves based on the positions of the non-dominated solutions. The GWO algorithm uses both specialised roles and social learning between individuals.

Firefly Algorithm

The Firefly Algorithm [207, 208] (FA) is based on the flashing patterns and behaviour of fireflies. The fireflies represent the solutions to the optimisation problem as a vector of d decision variables $\mathbf{x} = (x_1, x_2, ..., x_d)$. The algorithm explores the search space by having each solution be "attracted" to other solutions in the population based on dominance and distance. Solutions that are dominated move towards non-dominating ones.

At each iteration, for every pair of solutions (i, j) where $\mathbf{x}_j$ dominates $\mathbf{x}_i$, the update of $\mathbf{x}_i$ is defined as:

$$\mathbf{x}'_i = \mathbf{x}_i + \beta_0 e^{-\gamma \mathbf{r}^2_{ij}} (\mathbf{x}_j - \mathbf{x}_i) + \alpha \cdot \epsilon_i \tag{2.16}$$

where $\beta_0 e^{-\gamma r^2_{ij}}$ is the brightness which scales with distance r_{ij} between $\mathbf{x}_i$ and $\mathbf{x}_j$ based on the inverse square law. β_0 is the attractiveness at distance 0, usually set to 1. γ is the light absorption coefficient, which determines the impact of distance on the attractiveness between individual solutions. r_{ij} is the distance between the two solutions $\mathbf{x}_i$ and $\mathbf{x_j}$, usually calculated as the Euclidean distance. The α term is a randomisation coefficient, and ϵ is a vector of random numbers drawn from a Gaussian or uniform distribution. Typically $\gamma = \frac{0.5}{(U-L)^2}$ and $\alpha = 0.01(U - L)$, where U and L are the upper and lower bounds values of $\mathbf{x}$. picked from a uniform distribution.

Solutions that are non-dominated move based on the best solution **gbest** found so far:

$$\mathbf{x}'_i = \mathbf{x}_i + \mathbf{gbest} + \alpha \cdot \epsilon_i \tag{2.17}$$

As the solutions move based on the fitness of other nearby solutions, the algorithm can be seen as a similar variation to a Particle Swarm Optimisation algorithm. The main difference is the lack of a single global best in the swarm, as each individual solution is attracted to all others in the swarm, moving different distances based on the attractiveness. This uses elements of selection, as individual solutions are more attracted to fitter non-dominated solutions. Social learning is also observed, as an individual moving towards non-dominated solution mayb become non-dominated themselves.

Cuckoo Search

Cuckoo Search [206] is an algorithm inspired by the aggressive egg laying behaviour of cuckoo birds, which lay eggs in the nests of other birds. It is a population-based algorithm that uses the concept of Levy flights [23] for exploration, and selection mechanisms for exploitation. The biological analogy of "eggs" is used as the solution vector $\mathbf{x} = (x_1, x_2, ..., x_d)$ for d variables. The algorithm follows three rules for multi-objective problems with o objectives [209]:

- New solutions are generated by performing Lévy flights from existing solutions. The new solution $\mathbf{x}_{i+1}$ is generated as follows:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha \cdot \text{Lévy}(\lambda) \tag{2.18}$$

  where α is a step-size scaling factor and Lévy(λ) is a step drawn from a Lévy distribution defined by:

$$\text{Lévy} \sim u = t^{-\lambda}, 1 < \lambda \leq 3 \tag{2.19}$$

- Non-dominated solutions are carried forward into the next generation.
- A fraction p_a of the most dominated (worst) solutions are discarded.

These rules balance exploration and exploitation towards the Pareto front, with poor solutions discarded as a selection mechanism. The degree of exploration and exploitation is controlled by the α and p_a parameters, making the algorithm sensitive to parameter tuning. Furthermore, Cuckoo Search requires additional hybrid mechanisms to handle many-objective problems due to premature convergence and computional scaling issues [32].

2.4.3 Summary of current algorithms in a biological framework

In general, most of the categories of algorithm use a single category of information transfer, the Evolutionary Algorithms focus on different forms of genetic transfer with mechanisms focused around changing the selection of the population to mate and Swarm Intelligence focuses on different forms of cultural information transfer. Many of the popular Evolutionary Algorithms such as NSGA-II [45], MOEA/D [217], and SHADE [184] have diverged from their biological roots into mathematical or statistical methods. In the Swarm Intelligence algorithms, many follow a learning strategy of updating the decision variables with different formulae. For example,

- PSO velocity update

$$\mathbf{x}_i' = \mathbf{x}_i + \mathbf{v}_i \tag{2.20}$$

- Grey Wolf hunting strategy

$$\mathbf{x}_i' = \frac{\mathbf{x}_\alpha + \mathbf{x}_\beta + \mathbf{x}_\delta}{3} \tag{2.21}$$

- Artificial bees

$$\mathbf{x}_i' = \mathbf{x}_i + w_1 * \mathbf{r}(\mathbf{x}_i - \mathbf{x}_j) \tag{2.22}$$

Each of these algorithms follows the mechanism of social learning in cultural inheritance. Each solution representation $\mathbf{x}$ is updated based on its previous values and additional information through other members of the population. Table 2.6 sums up the Swarm Intelligence algorithms based on their biological analogy, update function, and information transfer mechanics. While some are based on biological studies of animal behaviour, such as Ant Colony Optimisation with pheromone trails to probabilistically construct a solution, many are computational analogies for movement behaviour in variable space, with the difference between algorithms being how the reference directions are computed.

Table 2.6: Swarm Intelligence Algorithm properties

Algorithm	**Biological inspiration**	**Update Function**	**Information Transfer**
PSO	Particles	Velocity update	Global best and local best
Ant Colony Optimization (ACO)	Ants	Solution construction based on pheromones	Pheromone trails
Artificial Bee Colony (ABC)	Bees (employed, onlooker, scout)	Position update	Global best
Grey Wolf Optimizer (GWO)	Wolves (alpha, beta, delta)	Position update	3 global bests (alpha, beta, delta)
Firefly Algorithm (FA)	Fireflies	Position update	All global bests
Cuckoo Search (CS)	Cuckoo birds	Selection and position update	Lévy flight-based

Existing studies [30, 146, 178] which investigate the use of epigenetics in Evolutionary Computation will be further analysed in Chapter 4, presenting a discussion on the key concepts of epigenetics and how these existing studies follow or miss the key concepts.

2.5 Missing concepts

Despite the range of available algorithms, Evolutionary Computation can be closely linked to a relatively small part of the evolutionary literature, focused on genetic inheritance and the Modern Synthesis. Mechanisms from the Darwinian concepts and Modern Synthesis have all been included, but there is limited inspiration taken from the Extended Synthesis. A few elements from the Extended Evolutionary Synthesis have been explored: Swarm Intelligence algorithms fit closely to the concepts of cultural inheritance and information transfer; cMLSGA has been shown to be a successful implementation of multilevel selection; Evolvability is represented by hyper-parameter choices, which algorithms such as CMA-ES dynamically alter during a search.

While the Extended Synthesis provides a number of opportunities, not all of the elements provide easy inspiration for practical optimisation problems. Evolutionary and developmental biology (Evo-devo) includes the developmental stages of living organisms and the evolution of developmental processes; which in many ways most closely replicating the initial vision proposed by Turing [188] of generating a child and teaching it to learn. Adapting the concept for general Evolutionary Algorithms would require substantial expansion of the simple genotype model of evolution used in evolutionary algorithms, such as including properties relating to gene regulation, homeobox genes and allometry. Evo-devo provides a number of developmental steps for fine tuning an organism, but in the algorithmic world this is unnecessary when more generations can be run instead. It is possible to apply evo-devo to specific applications, such as digital architectures [135], where domain-specific concepts in architecture can be linked to evo-devo processes, and the time required to generate solutions makes fewer generations more suitable. The design of neural networks using Genetic Algorithms [103] mimics evo-devo concepts with limited success as neural networks already learn through their training process, and Evolutionary Algorithms are typically used to network parameters such as weights and architecture [211].

Genomic evolution involves the evolution of genome architecture itself. In computational terms, genomic evolution would involve the evolution of the number of variables or range of values in an optimisation problem. These values are typically set based on the problem and do not require evolutionary mechanics applied to them.

This leaves epigenetics, niche construction, and phenotypic plasticity as unused mechanisms with the most promise for providing excellent bio-inspiration for new algorithms. Particular focus is placed on the exploration of epigenetic mechanisms, alongside the possible benefits of niche construction and phenotypic plasticity.

2.5.1 Phenotypic plasticity

The concept of phenotypic plasticity is the idea that an organism's behaviour or physiology could change due to environmental factors [154]. Information from the environment is taken and used when creating the final phenotype from the genotype. For example, if a genotype for a neutral network is evolved first through training, a plastic neural network would be one that is able to learn and respond to the environment it is placed in, to alter the final weights of the network, leading to a plastic phenotype after genetic evolution. Further alterations such as change to the architecture of the network itself and number of hidden layers is possible too. Plastic neural networks have been explored in the past [128, 173, 187] with success, showing the benefits of including phenotypic plasticity in intelligent systems that need to run and adapt in the environments they are placed in. Learning can be further extended as a developmental process to include altering the topology of the network, although this has yet to be explored.

Algorithms such as Indicator-Based Evolutionary Algorithm (IBEA) [223] uses a performance indicator, such as Hypervolume, to guide the selection process. This indicator-based selection evaluates solutions based on their contribution to the quality of the approximated Pareto front. By prioritising solutions that improve the indicator, IBEA can mimic aspects of plasticity by focusing the search on regions of the objective space that adapt to changing problem characteristics, such as shifting objectives or constraints. The class of optimisation problems that focuses on dynamic problems, where the objectives and constraints may be unstable and change throughout the search, would also be suitable for algorithms with flexible, plastic responses. Algorithms in Swarm Intelligence can exhibit some of this behaviour, for example when candidate solutions take different roles in the Artificial Bee Colony depending on the number of existing solutions in other roles. However, there is no scale or range of plastic responses in reaction to a changing environment in the ABC implementation.

Phenotypic plasticity is a relevant concept particularly in intelligent agents that require adaptation in a working environment. While genotypes can be evolved during an optimisation search, phenotypic plasticity allows for learning and adaptation after the initial search.

2.5.2 Niche construction

A separate concept from niching in genetic algorithms, niche construction is the concept of organisms altering their environment to better suit their needs, and in doing so leaving behind these useful alterations for the next generations to benefit from [139]. Because this typically requires individuals in the population to change their own environment, it is difficult to accomplish computationally. Optimisation problems are usually predefined and the fitness landscape is dependent on the problem. If an algorithm is able to change the fitness landscape to suit the solutions it produces, it changes the problem definition, meaning the solutions found may no longer be applicable or useful to the original problem.

However, the concept of niche construction can be expanded to not only look at the fitness landscape as the environment that can be changed. The social environment in which the methods of interaction between individuals could be changed over time. Improved methods of communication and information flow can also be considered niche construction. In this case algorithms such as the Ant Colony Optimisation could be argued as using niche construction through the pheromone trails that change the information pathways between workers.

2.5.3 Epigenetics

Epigenetics plays an important role in adapting a population to new conditions and environments quickly. In computational terms, epigenetic mechanisms should help to improve convergence, potentially spreading changes through a population faster than genetic evolution, and improve stability around a solution in the face of environmental changes. There are a number of different epigenetics processes such as DNA methylation, bookmarking, gene silencing, gene repression, and genomic imprinting that are triggered by different factors such as environment, diet, or the presence of certain chemical compounds [77]. Because of the wide branch of epigenetics research, it is difficult to narrow down which mechanisms may be helpful to the performance of Evolutionary Algorithms.

In the Modern Synthesis view of evolution, genetic changes occur randomly and fitness is guided by natural selection. Similar to this, the development of modern genetic algorithms focus on improving the selection process while keeping genetic changes random to retain diversity. A number of generations is required for suitable traits with high fitness to spread throughout a population, even with mechanisms such as elitism. Epigenetic inheritance allows for faster changes based on environmental cues which can occur simultaneously among multiple individuals in the same generation. These adaptive adjustments do not affect the underlying genotype, allowing regular genetic processes to occur and epigenetic processes to be reversed. For example, in Genetic Algorithms, the final solution to an optimisation problem is generated from the phenotype consisting of the combination genes and phenotypic changes from the epigenetic tags. This allows solutions or values not found in the genotypes to be in the final solution, allowing rapid changes to not require slower genetic propagation. These rapid changes are ideal for scenarios such as dynamic optimisation problems or variable-length problems.

The inheritance of epigenetic tags in parallel with genetic inheritance results in continual rapid changes with a diverse set of tags among a population, without disrupting underlying genetic processes [201]. This is an important aspect in evolutionary biology to guide phenotypic variation in a direction suitable for the environment instead of relying solely on random mutation and natural selection.

Existing epigenetics studies in Evolutionary Computation

The key concept of epigenetics is to allow for fast variation when appropriate. Existing studies [30, 146, 178] inspired by epigenetics are currently missing the key feature of triggering mechanisms based on the fitness of the population to the environment. Mechanisms are triggered probabilistically without any distinction between individuals, parents, and how epigenetic marks are passed on. This probabilistic method is more akin to bet hedging [105] than epigenetics, where the mechanics do not improve individual fitness in stationary conditions with no drastic changes in the environment, but create advantages in extreme conditions such as being stuck at a local optima.

An epigenetic algorithm based on intra-generational epigenetic processes used by bio-molecules was developed by Periyasamy et al. [146]. This system mimics cellular organisation, with individuals of bio-molecules performing independent tasks in a swarm-like manner, and require specific conditions to be met. The focus on the epigenetic processes uses no genetic operators, which misses a genetic component to contribute to the final phenotype. The use of epigenetics was found to improve the convegence of this algorithm, but it potentially loses diversity from the lack of genetics and can get stuck at local optima. The intra-generational, self-creating system proposed is also not suitable for solving practical multi-objective optimisations, and no concrete benchmarks were used or tested in their work.

Chrominski et al. [30] used the epigenetic process of cytosine methylation in a Genetic Algorithm to solve the Knapsack problem. This approach integrated the method of cytosine methylation where a part of an individual's genotype can be blocked during the crossover operation. It aims to transfer a larger portion of the fitter parent's genotype while silencing the poorer parent individual. It was found that a 30-40% probability of the epigenetics operation occurring results in the lowest number of generations to find the best solution. Only the simple single objective Knapsack problem was tested, and performance was only compared to an unmodified GA, so it is not known if the concept extends well to complex multi-objective problems, or to more established evolutionary algorithms. Stolfi and Alba [178] also explored the concept of methylation and gene silencing in their version of an epiGenetic Algorithm (epiGA). Similar to Chrominski, a number of parent genes may be masked based on the probability of the epigenetics mechanism occurring. Although the algorithm was tested against the multi-dimensional Knapsack problem against multiple competing algorithms, it is still only a single objective problem.

Chikumbo, Goodman, and Deb explored the use of epigenetic silencing in a multi-objective land use management problem [29]. In their approach, decision variables are randomly assigned a value out of the 111 possible options based on a beta distribution at the start of the optimisation. Then selected variables are edited if the variable values violated adjacency constraints. The epigenetic mechanisms utilised in this study helped to ensure the constraints of the problem were satisfied, and thus was able to achieve a speed up in computation time. The results illustrate the potential use of epigenetics in the specific multi-objective optimisation problem of land use management. It is not yet known how the effects of the epigenetic mechanisms extend to other problems with fewer objectives and constraints, or how the mechanisms aid multi-objective algorithms in dynamic optimisation problems. Furthermore, the element of inheritance for epigenetic changes were not explored, leaving many aspects of applying epigenetics to multi-objective optimisation algorithms unknown.

So, while some algorithms inspired by epigenetics have been studied in the literature, they are currently missing the key features that lead to the benefits seen in evolution. These features are outlined in the following subsections.

Epigenetic tags - the epigenotype

A key aspect not captured by existing studies is the inheritance and transfer of epigenetic information to future generations. While the epigenetic mechanisms implemented were accurate, this transfer is important as it guides the direction of phenotypic change. Without this aspect, the epigenetic mechanisms simply act as another form of mutation, probabilistically switching genes on and off. To include the epigenetic information transfer, epigenetic tags can be used to form an epigenotype [17], to keep a history of inherited epigenetic changes and allow changes to an individual's genotype to be triggered based on these tags. Epigenetic tags can be added and removed based on signals from the environment, or based on inheritance and crossover operations when forming offspring individuals [19]. Epigenetic tags can also help control gene expression in response to environmental changes. In biology this helps to form "memory" based on changing environments [224]. The memory of the recent environment allows for fast adaptation and stability. By controlling how genes are expressed using an epigenotype, suitable traits are constantly adjusted to improve fitness before longer term genetic changes can be applied.

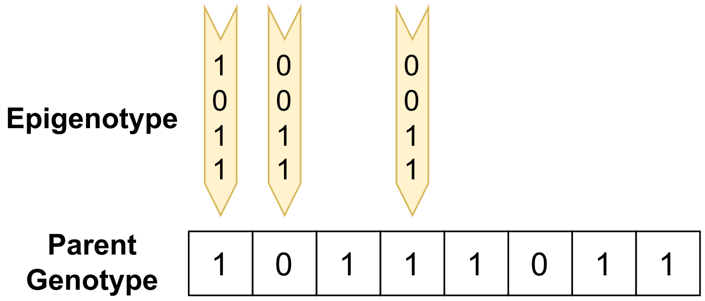


Figure 2.5: An epigenotype with epigenetic tags for some genes in the genotype.

Computationally, each variable in a solution can contain a set of tags that can be inherited and modified. Figure 2.5 shows the epigenetic tags on top of some variables in the genotype. The tag can be used to encode mechanisms to alter the variable after genetic operators are applied. These mechanisms can then help increase convergence by guiding phenotypic variation in a direction matching selection pressures. The underlying genetic mechanisms are further unaffected by epigenetic changes, and epigenetic changes can be easily reversed, reducing the cost of poor mutations compared to genetic mutations. The assumption of the one way flow from genotype to phenotype in biology is adopted, where the epigenotype sits over the underlying genotype without a backwards flow of information. It is possible that the underlying genome could benefit from learned changes of the epigenotype. However, this could also affect genes that are suitable for the long term being replaced by short term epigenetic adapations, leading to over convergence in a local optima. The advantage of fast adapations that can be quickly turned on and off could then be lost.

The epigenotype represents the three key aspects of epigenetics:

- the transfer of epigenetic information,
- self-adaptability to environmental changes,
- fast convergence from direction variation.

These aspects may have potential benefits especially for dynamic problems where the Pareto set is not static. In dynamic multi-objective problems, both the Pareto optimal set and its corresponding Pareto front may change over time due to shifting environmental conditions or objectives. The self-adaptability and fast convergence would allow algorithms with an epigenotype to adjust to a changing Pareto optimal set and its associated Pareto front quickly. With the use of an epigenotype [17], epigenetic tags can be added, removed, and inherited to future generations. An epigenotype alters the phenotype without changes to the underlying genotype. The tags can then be used to encode different epigenetic mechanisms to be triggered. The mechanisms have a range of effects on the genotype, such as switching genes on and off, or reducing gene expression based on the location and number of tags in the epigenotype.

Genomic imprinting

Genomic imprinting [26] restricts the expression of a gene to one parent. Imprinting is useful when the imprinted alleles lead to different phenotypes that affect an individual's fitness. This process does not directly change the genotype and can prevent segregation according to Mendel's laws at the phenotype level. Epigenetic tags are imprinted in the germline and cause the imprinted genes to be expressed from only one parent [200].

There are three hypothesised theories for the process of imprinting

- the kinship theory [78] – The theory suggests that an imbalance exists between parental genes due to conflicting fitness strategies from both parents. This is mostly apparent in sexual reproduction where the father and mother have differing interests to pass on their own genes.
- The sexual antagonism theory [182] – This theory uses sex-specific selection pressure. It predicts an uneven allele frequency between males and females when natural se-

lection favours one sex over the other so that offspring genes are enriched to benefit a particular sex.

- The maternal–offspring coadaptation theory [198] – Based on the correlation between the genes of the mother and the maternal genes of the offspring, the maternal-offspring coadaptation theory states that the offspring is more likely to inherit from its mother because it leads to a higher probability that the offspring has a positive interaction with its maternal phenotype, and the interaction leads to higher fitness.

Gene regulation

The presence of epigenetic tags enables gene regulation mechanics to occur on the tagged genes. There are multiple forms of gene regulation: gene silencing, gene activation, and gene repression. All forms of gene regulation affect the expression of the affected genes leading to variation in phenotypes from the same genotype [156]. Gene silencing is a mechanism for turning entire sections of the genotype on and off independent of mutation. In evolutionary biology gene silencing has the effect of protecting the host organism from viruses [151] by silencing genes that are used in viral reproduction. In terms of convergence and diversity, convergence should be increased and diversity decreased as silenced genes are not fully expressed compared to other genes.

Gene repression acts on individual genes rather than entire sections of the genotype. In evolutionary biology it switches off genes whose products are required to maintain cell functions [164]. To implement this computationally, each variable in a candidate solution can be switched on or off, based on the tags of the epigenotype. The modification of the tags can be based on the fitness of the individual, adapted based on the progress of the search or based on environmental cues.

2.6 Summary

The current development of multi-objective GAs trend towards the use of mathematical functions, moving away from the biological inspirations of the algorithms. The niching method of NSGA-II focuses on Pareto dominance to rank solutions and crowding distance to maintain diversity. The decomposition based approach decomposes the problems into well-distributed subproblems using defined weight vectors. None of these aspects have strong links to phenomena or concepts in biology. The approaches also make improvements to specific aspects of the algorithms to increase convergence with a priori knowledge, such as the addition of reference points in NSGA-III.

The less popular indicator and co-evolutionary approaches are closer to their biological inspirations. Co-evolutionary approaches have strong diversity mechanics making them more effective on some real-world problems such as Classical Laminate Plate Theory [75] and provide better generality. In particular, methods like HEIA and cMLSGA are able to combine the strong convergence aspects of niching or decomposition methods in a co-evolutionary environment to maintain diversity.

For Swarm Intelligence, a similar trend can be seen. For instance, the best performing PSOs on benchmarking problems are Pareto-based with a focus on convergence. Additional mechanisms to improve performance such as weights [144], objective ordering [203] and speed constraints [136] all take advantage of a priori knowledge of the test functions which would not necessarily be available for real-world problems. Furthermore, many Swarm Intelligence algorithms exhibit the same underlying mechanism of positional updates on the solution vectors, the novelty coming from the use of animal behavioural terms and slight alterations to the update function rightly drawing criticism [9, 174]. However there are some unique cases, such as Ant Colony Optimisation which uses a probabilistic method to solution construction, though it is more suitable to discrete problem domains.

The most successful additional of non-genetic processes is the use of multi-level selection in GAs. In the case of multi-level selection, previous iterations of the concept did not provide significant improvements as they missed key features of multi-level selection until the recent MLSGA. This shows that bio-inspired concepts only work when there is a bio-mimickry of the key features. The success of MLSGA suggests there are more non-genetic biological concepts that have only been briefly explored in the past, such as epigenetics, that may show performance gains when applied to more complex multi-objective problems. Epigenetics provides benefits in convergence rate, stability in volatile environments, and reversibility [131] compared to genetic mechanisms in evolution. This provides a self-adaptive means to quickly converge and stabilise in a changing environment. Few algorithms have explored epigenetics in depth and there are many underlying mechanisms that are both self-adaptive and convergence based.

To help make distinctions between different novel algorithms it is important to understand what the differences in their mechanisms are at the core. What are the biological operators used? What are the key similarities and differences between two bio-inspired algorithms apart from the terminology used? In some cases of the Swarm Intelligence algorithms such as Particle Swarm Optimisation and the Grey Wolf Optimiser, different biological inspirations are used but the underlying operators of learning strategy updates are the same. To ensure that future bio-inspired algorithms do not recycle the same concepts the key concepts and mechanisms are categorised under a biological framework. This shows how current algorithms relate to each other and demonstrate gaps in the evolutionary synthesis that computational algorithms have yet to explore. Between the genetic and cultural modes of information transfer, there is a gap in epigenetic mechanisms that are not fully explored. The next step is to take specific algorithms from Evolutionary Algorithms and Swarm Intelligence algorithms, test and compare them against a suite of problems to determine if their performances differences match the biological differences that are expected, and how suitable the algorithms are for epigenetic additions.

Chapter 3

Evolutionary Algorithms vs Swarm Intelligence

3.1 GA and PSO benchmarking

Genetic Algorithms (GAs) and Particle Swarm Optimisation (PSO) are often used interchangeably in multi-objective problems. However, there are currently limited comparisons of state-of-the-art methods to determine which types of problems these methods are most suited to. In many cases, new algorithms are developed for specific problems [150], although this leads to repeated effort and lack of general applicability, in addition to the general approaches which are used in a range of applications. Algorithm selection is therefore difficult, especially to someone unfamiliar with the field. Two of the most popular bio-inspired computation methods are GAs and PSOs. However, determining when each of these algorithms is most appropriate is difficult. PSO has traditionally been used for single objective problems and GAs for multi-objective problems, but increasingly the multi-objective literature features PSOs and in much of the applied literature, these two methods are often used interchangeably.

There are two major differences [6] between the PSO algorithm and Genetic Algorithms:

- GAs use parent selection and reproduction mechanisms to produce the next generation of individuals. While the reproduction mechanisms in GAs and velocity changes in PSO are similar, PSO does not use a selection function to choose or remove individuals from the population. Instead, the search is guided by the leaders and the best local and global solutions found so far.

- PSO's velocity update can be seen as a directional mutation that is guided by the best solutions found by the swarm and the individual so far. It will generally move in a similar direction if the two best solutions are close, or at an angle if the solutions are different. In GAs, mutations can set the direction of search in any direction.

Although the two algorithms were inspired by two distinctly different biological concepts, their general performance on current benchmarking problems are comparable, making it difficult to distinguish their differences.

3.1.1 Motivation for comparing GAs and PSOs

There are a number of studies comparing GAs, PSOs, and other Evolutionary Computation algorithms on specific applications [159, 175, 215] but these often use dated algorithms, such as the original GA or PSO, and are rarely conclusive. In the Evolutionary Computation literature, existing surveys of the state-of-the-art multi-objective optimisation algorithms tend to focus on the performance of each algorithm, but do not compare different classes [67]. The systematic differences between them are not compared to determine which classes of problems are each most beneficial to. Table 3.1 shows the benchmarking performed in the existing literature on the benchmark problems and algorithms that will be tested in this chapter. Not all of the Evolutionary Computation algorithms have been benchmarked, especially against the new multi-objective test sets, which introduce harder problems more akin to many real world multi-objective problems. In particular, PSOs have not been thoroughly benchmarked against these newer problems and are often excluded or poorly represented in benchmark comparisons, with new algorithms often tested against the older test problems. In order to test which form of Evolutionary Computation algorithms are best suited to further Extended Evolutionary Synthesis mechanisms, they must be compared, and more details on their biological behaviour and properties studied.

Table 3.1: Summary for the current status of benchmarking of Genetic Algorithms and Particle Swarm Optimisation algorithms in this chapter. This is not an exhaustive list of all benchmarking literature.

Algorithm	ZDT	DTLZ	WFG	UF	LZ_09	IMB	MOP	DAS-CMOP
U-NSGA-III	[166]	[121, 166]	[166]	[121, 166]				
MOEA/D	[116, 217]	[116, 217]	[88, 116]	[116]	[114]	[119]	[120]	[59]
IBEA	[223]	[12]	[12]					
HEIA	[116]	[116]	[116]	[116]				
cMLSGA	[75]	[75]	[75]	[75]		[75]	[75]	[75]
OMOPSO	[53, 172]	[53, 172]	[53]	[67]				
SMPSO	[53, 116, 136]	[53, 116, 136]	[53, 116]	[116]				
CMPSO	[216]	[216]	[216]					

3.1.2 Choice of benchmark problems

97 test problems from the multi-objective optimisation literature were selected for benchmarking. The test set is selected so that it contains a diverse range of properties, from simple concave shapes to larger multimodal problems and discontinuous Pareto Sets. The test problems selected were: ZDT [47], WFG [88], LZ09 [114], UF [218], DTLZ [48], MOP [120], IMB [119], and DAS-CMOP [59]. These cover the main test problems developed for multi-objective problems in the recent literature. Appendix A summarises the test problems chosen, grouping them into categories based on their properties.

3.1.3 Choice of algorithms to benchmark

5 GAs and 3 PSOs were chosen to be benchmarked and compared in this paper. HEIA, cMLSGA and SMPSO were selected due to their current excellent performance on multi-objective problems [75, 136], which are less commonly utilised. HEIA represents the co-evolutionary family of GAs, and cMLSGA represents the non-genetic multi-level selection with elements of co-evolution. U-NSGA-III, MOEA/D and IBEA were selected as the most commonly used algorithms in the other three families of multi-objective algorithms: niching, decomposition-based, and indicator-based respectively. The U-NSGA-III algorithm is selected in place of the original NSGA-II as it unifies the improvements made in NSGA-III to problems with more than 3 objectives, and a tournament selection approach to decide between individuals assigned to the same reference direction and improve performance on problems with fewer objectives. A large number of variants have been developed for MOEA/D since it was first introduced [205]. Most of these variants improve specific aspects of MOEA/D for better performance on specific types of problems. The MOEA/D-DE variant [114] is chosen for its strong general performance, as the differential evolution operator is able to find a widely and uniformly distributed set of solutions.

OMOPSO, the precursor to SMPSO was selected as it remains competitive with SMPSO on certain sets of test problems [53]. Finally, CMPSO was selected for its multiple swarm approach, which improves the diversity retention of the population compared to traditional PSO approaches, a property which has shown to be important in more complex multi-objective problems. The algorithms are summarised in Tables 3.2 and 3.3.

3.1.4 Experimental setup

Each algorithm was benchmarked with 100,000 function calls each, on 20 independent runs, for each test problem. The solutions found are placed into an external unbounded archive, storing non-dominated solutions found throughout the search rather than taking solutions only from the current or final population [185]. Performance metrics are calculated every

Table 3.2: Review of the selected Genetic Algorithm mechanisms

Algorithm	**Year**	**Reason**
Epsilon-IBEA [223]	2004	An indicator based approach using the additive epsilon indicator, IBEA uses the indicator during selection, along with a binary tournament mating selector, to choose the individuals for the next generation. This contrasts with the more common Pareto dominance ranking used in the selection procedure of algorithms such as NSGA-II.
MOEA/D [217]	2007	A decomposition based Evolutionary Algorithm, MOEA/D decomposes multi-objective problems into subproblems to be solved simultaneously. Information is shared between neighbouring subproblems to reduce computational complexity. This leads to excellent performance on continuous search spaces but large discontinuous regions may lead towards ineffective decomposition of the problem.
U-NSGA-III [166]	2014	Combines the approaches from NSGA-II for multi-objective problems (2-3 objectives) and NSGA-III designed for many-objective (4-10 objectives) problems. It uses references points calculated in the search space to guide the evolution of individuals. Comparisons between NSGA-II and U-NSGA-III [31, 90] show that U-NSGA-III performs better on a majority of problems, especially those with 3 objectives or fewer. It exhibits good performance on most problem types.
HEIA [116]	2015	The Hybrid Evolutionary Immune Algorithm uses cloned individuals in subpopulations, based on multi-objective immune algorithms. Rather than using the same evolutionary operator for every individual, each subpopulation then evolves separately using different evolutionary strategies, with a Pareto and non-Pareto based optimiser. Elites are also cloned back into different subpopulations to enable information sharing between generations. This provides high performance on most problem types.
cMLSGA [75]	2019	A diversity first, multi-level selection approach, cMLSGA uses the evolutionary concepts of multi-level selection and co-evolution at the collective level, rather than the individual level used in most popular algorithms, to combine different Evolutionary Algorithms.

Table 3.3: Review of the selected Particle Swarm Optimisation mechanisms.

Algorithm	**Year**	**Reason**
OMOPSO [172]	2005	Study from [53] showed that OMOPSO outperformed a number of state-of-the-art multi-objective PSOs, including CLMOPSO, the multi-objective variant of the single objective CLPSO with strong global search properties and mechanisms to handle premature convergence. This algorithm uses Pareto dominance and the crowding distance to filter leader solutions, and two mutation operators on subpopulations to balance between convergence and divergence.
SMPSO [136]	2009	SMPSO makes an enhancement to the OMOPSO algorithm by constraining the velocity of particles to disallow extreme values, giving the algorithm better performance on ZDT and DTLZ problems compared to OMOPSO, which contain many local optima, but worse on the WFG problems, which are mostly concave in nature.
CMPSO [216]	2013	CMPSO uses multiple populations to solve multi-objective problems, with each population focusing on a single objective. CMPSO shares information between swarms using an external archive. The archive is updated with an elitist learning strategy to improve diversity. The algorithm also changes the velocity update function of the traditional PSO to approximate the whole Pareto front with information from different populations.

1,000 function calls to show algorithm performance over time.

The U-NSGA-III, MOEA/D-DE, Epsilon-IBEA, OMOPSO, and SMPSO algorithm implementations are from the jMetalpy framework [16]. The HEIA, cMLSGA, and CMPSO implementations are implemented based on their original papers. Table 3.4 shows the hyperparameters used for each algorithm. The hyperparameters were chosen based on the original algorithm papers where possible, to avoid hyperparameter tuning and to show the performance in a scenario where a priori knowledge is lacking. Otherwise, generally accepted values such as a high crossover probability and low mutation probability [44] were used. MOEA/D used a differential evolution crossover and the other GAs used simulated binary crossover. All the GAs and SMPSO used polynomial mutation as the mutation method, with probabilities as shown in the table. cMLSGA was used with the MOEA/D-MSF, HEIA variant for the collective algorithms as it has been shown to be the top performing variant for bi-objective problems [74].

Table 3.4: The selected hyperparameter settings for each algorithm. N is the size of the population, D is the number of variables, MR and CR are the probabilities of mutation and crossover respectively, and η is the distribution index of the SBX crossover. OMOPSO uses the same mutation rate for its uniform and non-uniform mutation. H is the number of points. F is the differential weight for differential evolution. NS is the neighbourhood size and NP is the neighbourhood selection probability for MOEA/D-DE. c_1, c_2 and c_3 are the constants in the PSO velocity update equation.

Algorithm	**Hyperparameters**
U-NSGA-III	N=128, $MR = 1/d$, $CR = 0.9, \eta = 20, H = 16$
MOEA/D-DE	N=100, $MR = 1/d$, $CR = 0.9, F = 0.5, NS = 3, NP = 0.9$
Epsilon-IBEA	N=100, $MR = 1/d$, $CR = 0.9, \eta = 20$
cMLSGA	N=1000, $MR = 0.08$, $CR = 1, \eta = 20$, collectives $= 8$
OMOPSO	N=100, $MR = 1/d$, $c_1 \in [1.5, 2], c_2 \in [1.5, 2]$
SMPSO	N=100, $MR = 1/d$, $c_1 \in [1.5, 2.5], c_2 \in [1.5, 2.5]$
CMPSO	N=20, $c_1 = c_2 = c_3 = \frac{4}{3}$

3.1.5 Performance metrics

In multi-objective benchmarking and evaluation, the true Pareto front, that is the set of all non-dominated solutions, of a problem is often known and can be computed, to compare against the approximated Pareto front generated by an algorithm. In addition to the difficulties of evaluating solutions in multi-objective optimisation, determining the quality of solutions found on the approximated front is also important and difficult. Zitzler, Deb, and Thiele defined a good approximation of a true Pareto front as one which [222]:

- minimises the distance between the true Pareto front and its approximation,
- distributes of points across the approximated front,
- maximises the extent of the approximated front.

Solutions found by an Evolutionary Algorithm form the approximated Pareto front, which aims to be as close to the true Pareto front as possible, well distributed across the front, and find a wide range of values for each objective. For simplicity, the approximated Pareto front found by an algorithm will be called the Pareto front throughout the rest of the thesis, while the true Pareto front will be explicitly called as the true Pareto front.

Many performance indicators [220] have been developed to determine how ‘good’ a set of solutions are. These metrics can be categorised based on their properties [158] into four distinct categories:

- Cardinality - The quantity of non-dominated solutions found by an algorithm. Cardinality metrics are rarely used in state-of-the-art comparisons as they do not capture the distribution or convergence of the set of solutions, both of which are strong indicators of the quality of solutions founds.
- Convergence - The distance between the set of non-dominated solutions and the true Pareto front.
- Distribution and diversity - The distribution of solutions across the Pareto front and the extent of the solutions. Although these metrics are not very useful alone, they can be used to compare algorithms with similar convergence to determine the diversity of solutions found.
- Convergence and distribution - Capture both convergence and distribution metrics together as a single score.

Two popular metrics used in the multi-objective literature are Inverted Generational Distance (IGD) [165] and Hypervolume (HV) [221]. These metrics are popular as they measure multiple categories with a single score. Both metrics were used in the benchmarking.

The Inverted Generational Distance (IGD) [165] is a popular metric used to determine the distance between a set of solutions and the true Pareto front. Given a set of solutions S found by an algorithm, and a set of solutions P uniformly sampled along the true Pareto front, the $IGD(S, P)$ is calculated as

$$IGD(S,P) = \frac{\sum\limits_{\mathbf{x} \in P} d(\mathbf{x}, S)}{|P|} \tag{3.1}$$

where $d(\mathbf{x}, S)$ is the Euclidean distance between the solution $\mathbf{x}$ and the solution in S that is nearest to $\mathbf{x}$ in the objective space. The true Pareto front is required to determine the IGD metric and the more reference points used, the more accurate the IGD values can be. This can be an issue as a small number of reference points leads to an inaccurate generational distance. IGD values also cannot be calculated without knowledge of the true Pareto front, making it unsuitable for evaluating real world applications where the Pareto front is unknown. As the IGD computes the distance between solutions found by an algorithm and the reference points on the Pareto front, it favours algorithms that return few close solutions compared to distributed solutions further from the reference points. A short distance of the points to the true solution and a uniform spread of solutions along the true Pareto front leads to smaller IGD values. However, IGD is not Pareto-compliant. A performance indicator is Pareto-compliant if, for two solution sets S_1 and S_2, it assigns a better score to S_1 whenever every solution in S_2 is dominated by at least one solution in S_1. IGD is not Pareto-compliant because it focuses on minimising the average distance to the true Pareto front, which may favour solutions closer to reference points even if they are dominated, potentially leading to misleading quality assessments. The reference Pareto fronts used to calculated the IGD values for each benchmark problem are displayed in A.15 in the appendix.

The HyperVolume (HV) metric [221] measures the volume of space between a reference point and a solution. Given a set of solutions S found by an algorithm, the HV represents the volume of the objective space that is dominated by all the points in S. The reference point for calculating the HV is usually set as the worst point in the objective space, or the anti-optimal point, such that every solution dominates it. The larger the volume, the further away the solutions are from the anti-optimal. The HV captures both the closeness to the optimal Pareto front, and the diversity of solutions across the objective space, as both are required to achieve a larger volume. Calculating the HV can be expensive, and the method described by While et al. [197] is used due to its rapid encoding. HV is a Pareto-compliant metric because it measures the dominated volume, ensuring that sets with non-dominated solutions closer to the true Pareto front and with better spread across the front are favoured. Due to its properties in capturing both the convergence and diversity of solutions, and its Pareto-compliance, it is widely used to compare and benchmark state-of-the-art algorithms.

When HV and IGD rank solutions differently, it often indicates that IGD is prioritising proximity to reference points over dominance. Specifically, because IGD measures the average Euclidean distance from each reference point on the true Pareto front to the nearest solution in S, it rewards solutions sets with solutions that are geometrically closer to the front, even if they are dominated. For example, with two solution sets S_1 and S_2, if S_1 has solutions that are closer to the true Pareto front in terms of Euclidean distance but has solutions that are better in at least one objective and no worse in others, in other words dominated, S_1 may achieve a better IGD score due to the proximity to the true Pareto front, while S_2 will have a better HV because HV accounts for the dominated volume, and therefore the non-dominated solutions. This difference implies that solution sets with good IGD but worse HV may contain solutions that are geometrically close to the true Pareto front but fail to dominate others, indicating poorer quality in terms of Pareto optimality.

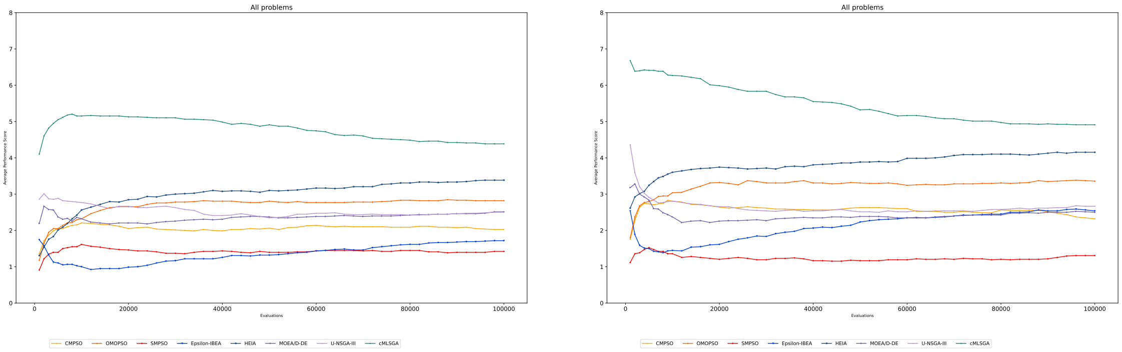
Furthermore, the **average performance score (APS)** [13] is calculated to compare the performances of each algorithm against each other on every test problem. Given n algorithms $\{A_1, ..., A_n\}$, the performance score of an algorithm $PS(A_i)$ on any given problem is defined as

$$PS(A_i) = \sum_{j \in \{1,...,n\} \setminus \{i\}}^{n} \delta_{i,j} \tag{3.2}$$

where $\delta_{i,j} = 1$ if A_j statistically outperforms A_i according to a Wilcoxon rank-sum test with $p < 0.05$. Otherwise, $\delta_{i,j} = 0$. Type I errors are adjusted for using the Bonferroni correction, where the threshold of 0.05 is divided by the number of pairwise tests. The performance score $PS(A_i)$ indicates the number of algorithms that outperforms A_i. The lower the score, the better the algorithm performs compared to other algorithms. The score can be averaged across all problems or a subset of problems to give the relative performance of A_i to the other algorithms on those problems, forming the average performance score. The APS is calculated every 1,000 function evaluations to provide the performance of every algorithm over time.

3.2 Comparison between Genetic Algorithms and Particle Swarm Optimisation

The algorithms are compared on the test problems at 100,000 function evaluations to demonstrate the capabilities of each algorithm. Figure 3.1 displays the score averaged on all problems. HV reference points are calculated using slightly worse (10%) nadir points. For example in ZDT1, ZDT2, and ZDT4, the nadir point is (1.0, 1.0) and so a reference point of (1.1, 1.1) is used for HV calculations. Uniform sampling is used to generate the reference Pareto fronts, with the number of solutions in the reference set specified in Appendix A. The performance scores calculated with both the IGD and HV metrics are shown. When averaged on all problems, SMPSO and Epsilon-IBEA consistently outperform the other algorithms for a majority of the 100,000 function evaluations. Using the IGD metric shows a much closer performance between Epsilon-IBEA and the two genetic algorithms MOEA/D and U-NSGA-III, suggesting better convergence properties of SMPSO compared to its diversity and spread across the Pareto front. As the IGD metric is not Pareto compliant, a better IGD score compared to HV for SMPSO could imply that solutions are closer in proximity to the true Pareto front. However, it does not guarantee that all solutions dominate in Pareto terms. Furthermore, depending on coverage of the reference Pareto set of the benchmark problem, gaps in the reference set would lead to IGD penalising solutions near the gaps due to the distance to a reference solution. cMLSGA performs the worst on both metrics, but is able to show the most continued improvement with more function evaluations.



(a) HV

(b) IGD

Figure 3.1: Average performance score for all problems tested.

Further, the average performance scores over time show little variation between algorithm performance at later stages of the optimisation. Most algorithms stay at the same performance compared to the other algorithms after 40,000 function evaluations, with the exception of Epsilon-IBEA. The first 10,000 – 20,000 function evaluations show the greatest differences between the algorithm performances. For example, the PSO algorithms have a strong start and a slight decrease in performance from 1,000 to 10,000 evaluations. HEIA also displays this behaviour. Epsilon-IBEA shows interesting performance as it improves at the beginning, but after 10,000 evaluations its performance slowly decreases, being overtaken by SMPSO, and MOEA/D-DE when using IGD values, at the end of the 100,000 evaluations.

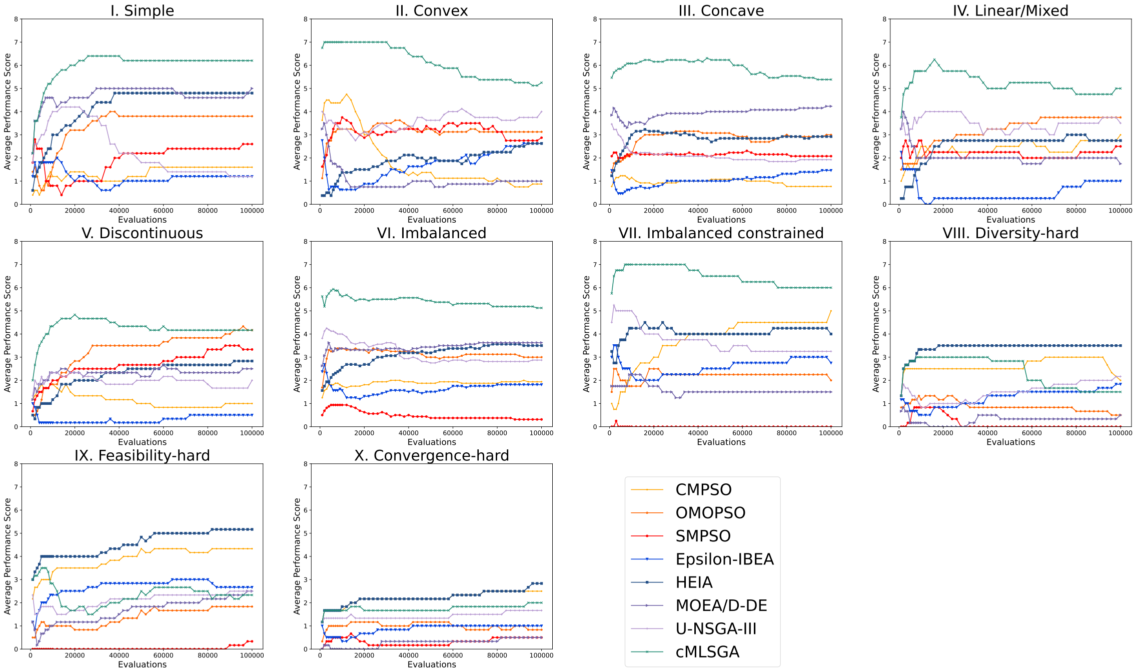


Figure 3.2: Average performance score calculated with HV, grouped by problem category.

A focused overview on each problem category is illustrated in Figure 3.2. For the best overall performing algorithm SMPSO, its performance category V discontinuous problems is relatively worse compared to other algorithms. OMOPSO, the predecessor algorithm to SMPSO finds similar performance patterns with increasingly worse performance with more function evaluations. For further examples, both algorithms find poor performance on discontinuous problems, and strong performance on diversity-hard and feasibility-hard problems. On discontinuous problems, SMPSO is the third worst performing algorithm from 20,000 function evaluations onwards, with OMOPSO as the second worst performing algorithm. Mixed performances are found for all the Genetic Algorithms based on each category of benchmark problem. Epsilon-IBEA generally performs the best, exhibiting low average performance score throughout the optimisation on categories I, III, IV, V, and VI. However, every Genetic Algorithm has problems where they show strong performance. For example, MOEA/D is the top performing algorithm for category II and X problems while

U-NSGA-III also strong performance on categories I, III, and V. Even though cMLSGA showed the generally the worst performance from categories I to VII, on the DASCMOP set of diversity-hard and feasibility-hard problem, cMLSGA exhibits an average competitive performance while previously strong performance algorithms such as CMPSO and Epsilon-IBEA show increased average performance scores. These mixed results suggest that the performance of each algorithm is problem-dependent, and while some general findings can indicate one or two algorithms showing an overall pattern of good performance, the properties of specific problems can have a large impact on each individual algorithm's performance.

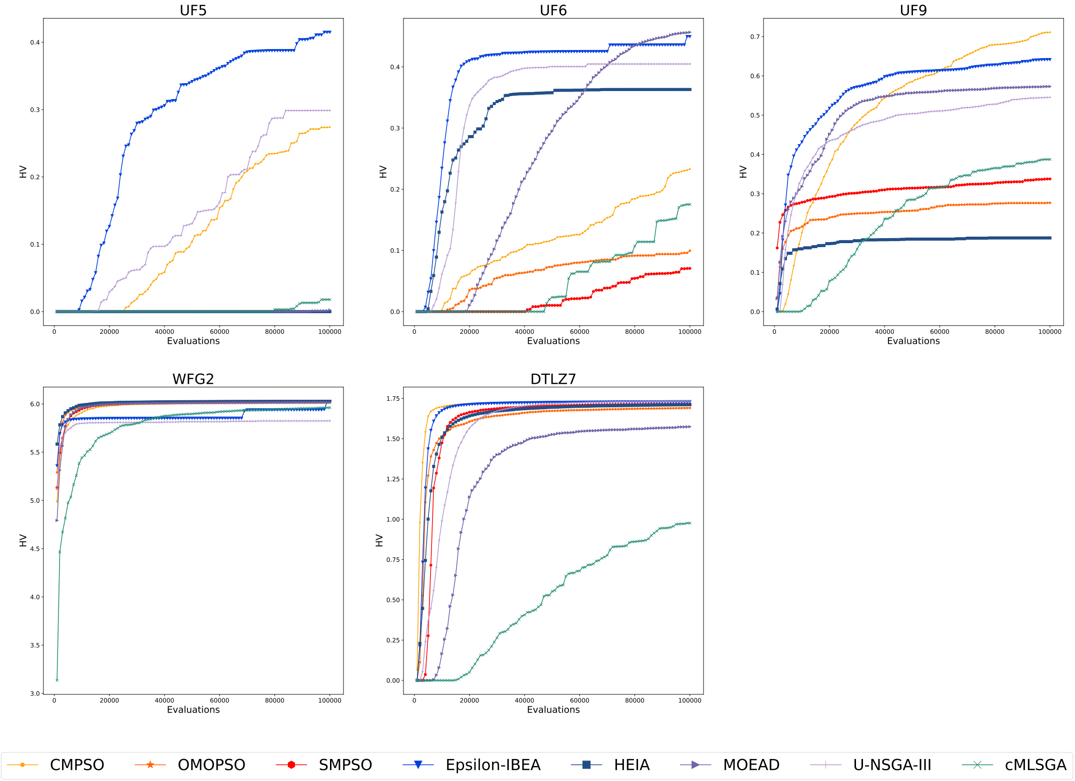


Figure 3.3: HV values of discontinuous problems

Diving deeper into some specific problems, The PSO performance on discontinuous problems are shown in Figure 3.3 with the HV values of each problem in this category. While the WFG2 and DTLZ7 HV values of all the algorithms are similar and converge quickly, the differences are more clear on the UF6 and UF9 problems. The fast convergence and plateau for WFG2 and DTLZ7 point to the lower difficulty of the two problems compared to the other UF problems where performance is mixed between the algorithms. Genetic algorithms such as Epsilon-IBEA, MOEA/D-DE and U-NSGA-III are able to find good solutions, specifically on UF5, UF6 and UF9, and start to converge compared to the two PSO algorithms SMPSO and OMOPSO. However, CMPSO performance contrasts the other two PSO algorithms on these discontinuous problems. Its performance surpasses the genetic algorithms on UF9 after 60,000 function evaluations and is one of the few algorithms to find good solutions on UF5 and MOP4, indicating the effectiveness of its multiple population approach. There is a trade-off with the CMPSO approach, as it performs worse on other categories that SMPSO and OMOPSO perform well in such as category VII imbalanced constrained problems or category IX feasibility-hard problems, supporting the theorem of no free lunch [199].

For GAs, Epsilon-IBEA generally shows the top performance among the genetic algorithms on the UF problems, however other algorithms such as MOEA/D-DE on UF6 or CMPSO on UF9 are able to find better HV values towards the end of the optimisation. Another example of improving performance over time is cMLSGA. In the UF9 problem, cMLSGA begins as the worst performance algorithm in the first 30,000 evaluations, but is able to find better solutions throughout, achieving higher HV values with more function evaluations and surpassing the performance of SMPSO, OMOPSO, and HEIA. Both CMPSO and cMLSGA show interesting and similar behaviour as algorithms utilising co-evolutionary mechanics, with cMLSGA utilising multilevel selection which may require additional evaluations to provide performance benefits as suggested by the longer term payoffs in multilevel altruistic behaviour [79].

3.2.1 Performance at an early stage in the optimisation.

As stated previously, much of the final performance at 100,000 function evaluations of the algorithms are determined at the early stages of the optimisation. Figure 3.4 shows the average performance scores from 1,000 to 30,000 evaluations. Here the initial convergence properties of each algorithm can be seen. The fast convergence of PSO algorithms can be seen across many categories such as category II the convex problems and category V the discontinuous problems. The PSO algorithms show the strong starts at the first few thousand evaluations, but the performance gets worse later on. For example in the convex problems, SMPSO and OMOPSO both start strong but their performance is quickly overtaken by the other algorithms after 7,000 evaluations. A similar pattern is seen for the discontinuous problems, strong initial performance followed up other algorithms outperforming as more function evaluations are done. The performance contrast of CMPSO to the other two PSO algorithms is seen in more detail. In the convex problems, SMPSO and OMOPSO performance scores start to increase, but the CMPSO performance score is able to decrease significantly after 20,000 function evaluations. The pattern repeats once again for the discontinuous problems, indicating the function of CMPSO's multiple populations.

Another interesting category is category VIII, the diversity-hard set of problems. SMPSO and OMOPSO's performance is strong initially, but is overtaken by genetic algorithms such as Epsilon-IBEA, MOEA/D-DE, and U-NSGA-III at about 10,000 evaluations. However, the performance of the two algorithms improve quickly after 20,000 evaluations, with SMPSO becoming the best performing algorithm at 30,000 evaluations. From the performance at the full 100,000 evaluations seen in Figure 3.1 both PSO algorithms continue to retain their strong performance. Additionally, CMPSO finds good

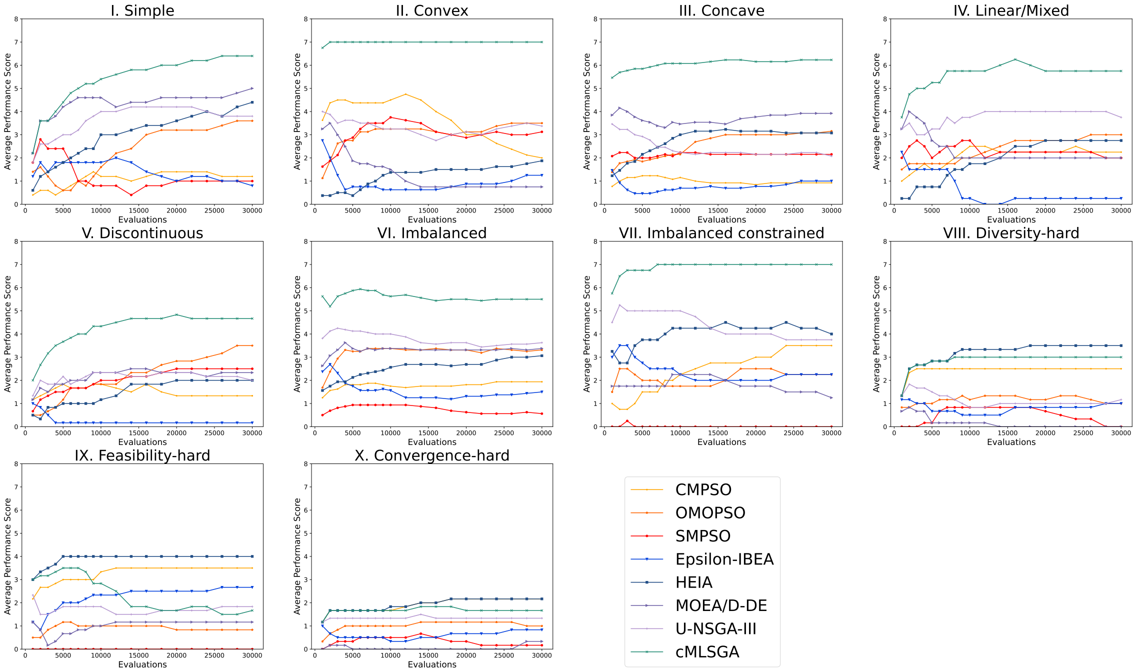


Figure 3.4: Average performance score up to 30,000 function evaluations.

solutions towards the end of the optimisation, with its performance score dropping from 4.5 to 2.8 in the last 5,000 function evaluations. The result is surprising as PSO algorithms are traditionally seen as convergence-based while genetic algorithms are more diversity-based. Strong performance combined with the HV metric indicates the PSO algorithms are able to find a diverse set of solutions along the Pareto front. The observed performance of PSO algorithms in the diversity-hard problems may come from the different techniques and mechanisms developed in modern optimisation algorithms, to combat their inherent limitations. For example, SMPSO's leader selection scheme using crowding distance in addition to domination, a mechanism similar to NSGA-II, or CMPSO's multiple population approach. In both cases it is no longer a simple diversity-convergence comparison between PSOs and GAs.

For the genetic algorithms, performance is equally mixed between well-performing algorithms such as Epsilon-IBEA, consistent performing algorithms such as MOEA/D-DE or U-NSGA-III. In only the first 30,000 evaluations, cMLSGA, which showed mostly poor results, is able to exhibit strong performance improvements in category IX feasibility-hard problems from 5,000 to 15,000 function evaluations, going from an average performance score of 5 to 2.17. In most cases, the performance at 30,000 evaluations is indicative of the final performance of an algorithm at 100,000 evaluations with two exceptions: U-NSGA-III for simple problems and CMPSO for convex problems. On simple problems at 30,000 evaluations, U-NSGA-III has an average performance score of 3.8, meaning an average of 3.8 algorithms outperform U-NSGA-III at this point of the optimisation. However, by 100,000 evaluations, U-NSGA-III has an average performance score of 1.2 and is tied with

Epsilon-IBEA as the best performing algorithm at the end of the optimisation. Similarly, but not to the same extent, MOEA/D-DE is able to show better early performance in the first 20,000 evaluations compared to later at 100,000 for the DASCMOP set of problems. For example on both diversity-hard and convergence-hard problems, MOEA/D-DE achieves an average performance score of 0 from 10,000 – 20,000 evaluations and remains at a low average performance score after 30,000 evaluations. Simiarly, MOEA/D-DE achieves an average performance score of 1.17 at 30,000 evaluations for feasibility-hard problems. However, performance over 100,000 evaluations as seen in Figure 3.1 shows MOEA/D-DE getting an average performance score of 0.5, 3, and 0.5 for diversity-hard, feasibility-hard, and convergence-hard problems respectively. The results show that most algorithm performance is determined early on in the optimisation search, and while some fluctuations and exceptions can occur, the general behaviour holds across multiple different algorithms and problem categores.

3.2.2 Comparing Genetic Algorithms and Particle Swarm Optimisation

Individual algorithms may perform better or worse on specific problems and categories of problem. The two different types of algorithms, genetic algorithms and particle swarm optimisation algorithms, are compared to each other next. The performance differences between the two types of algorithm are more clearly displayed in Figure 3.5. GAs are coloured in red while PSO algorithms are coloured in blue.

There is not a clear distinctive difference between the two types of algorithms. On each category of problems, there are categories where a PSO algorithm performs the best, and categories where a GA performs the best. For example, categories 6 and 7 show two PSO algorithms in the top three algorithms for most of the optimisation while categories 2 and 5 show a GA performing best throughout the 100,000 function evaluations.

A few performance patterns can be seen that stand out in some categories. The PSO algorithms perform best at the early stages of category I, then their performance slowly decreases as some of the GAs are able to find better solutions later in the optimisation. However, even in this category I case, the results are more algorithm-dependent as one GA performs consistently well, one GA's performance improves substantially throughout the optimisation, and the other three GAs all remain at poor performance worse the the PSOs. Another example is category IV. Here all three PSOs seem to show relatively poor performance. However, all other algorithms except one GA also show similarly poor performance, indicating that the specific algorithm has a larger impact on performance than the type of algorithm it is. This is further evidenced in category IX, where one PSO performs best, one PSO performs similarly to most of the GAs, and finally one PSO and one GA perform the worst. The range of performances in both GAs and PSOs on the variety of problem categories demonstrates that in general use cases, the performance different between the two classes of algorithms is more algorithm-dependent than problem-

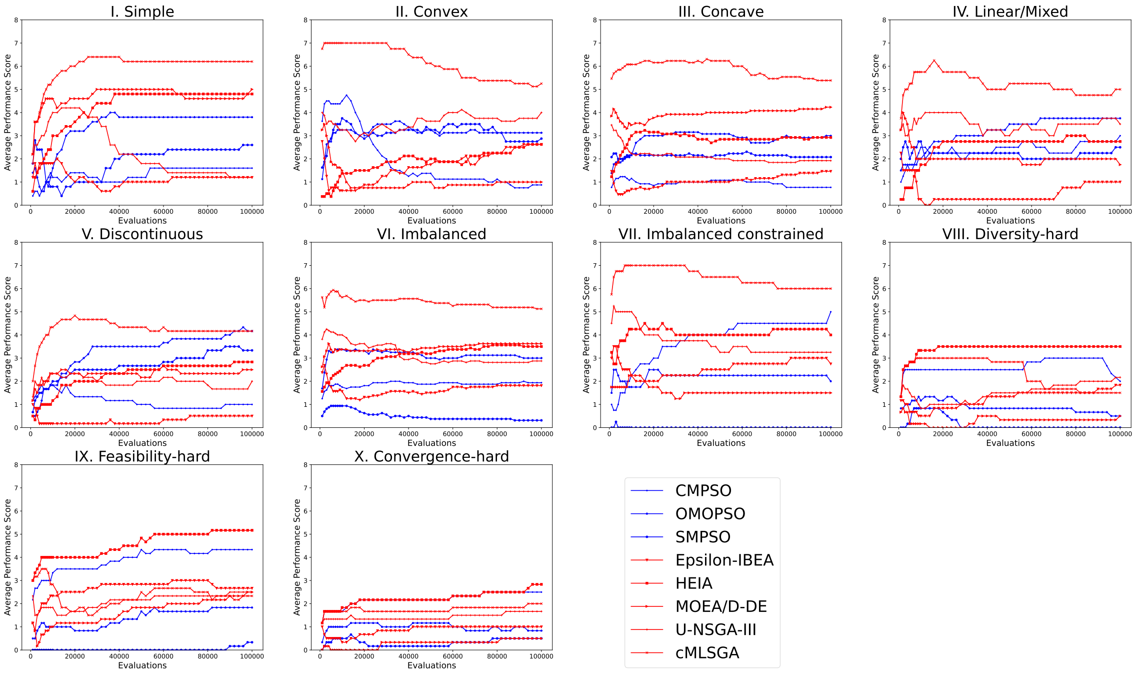


Figure 3.5: Average performance scores of GAs and PSOs on all ten categories, with GAs shown in red and PSOs shown in blue.

dependent.

3.3 GAs and PSOs analogies to evolutionary biology

The results from comparing the GAs and PSOs show there is not a fundamental difference in their performance, each algorithm displays strengths and weaknesses on different categories of problems, with only a few patterns occurring across the whole class of algorithms. Notably, the differences in performance between algorithms across each benchmark problem category is dependent on the algorithm and problem, more so than the class of algorithm it is, GA or PSO. Epsilon-IBEA and SMPSO are the two best performing algorithms from GAs and PSOs respectively, but there are problems categories where one or both of these algorithms are outperformed by another algorithm.

The multiple swarms with an elitist learning strategy employed within CMPSO is an example of a mechanism to improve the diversity of PSO solutions. Having multiple swarms should ensure the search does not get stuck at local optima. However, their inclusion also impacts on the convergence of the algorithm, making its performance contrast that of the other two PSO algorithms. While these mechanisms can help increase the diversity of the solutions where CMPSO outperforms the other PSOs and gives a comparable performance on the discontinuous problems such as UF9 to other good performing algorithms, it generally does not outperform SMPSO across all problems.

The performance of these algorithms relates to some extent with theoretical work in evolutionary biology [55, 106, 108, 155] and ecology [21, 71, 191] which is increasingly focussed on variation amongst individuals within populations because those individual responses aggregate to the population level. These individual differences are typically related to differential processing of environmental cues - if the optimum/target is a different shape (sharp vs flatter, wider peak) then the optimal combination of environmental cues to deliver the best adapted traits will change. Angeline [6] discussed the philosophical and performance differences of the classic GA and PSO, and found that while the PSO moved towards a peak faster, it was less adept at finding fine-scale optima around the peak than the GA due to being unable to scale the velocity steps dynamically to the problem. This is similar to transgenerational components in Hoyle and Ezard [87] where strong maternal effects can lead to overshooting the original phenotype, while negative effects that slow down these movements towards an optimum leads to maximising fitness in the long term. The success of CMPSO on problems where the other PSO algorithms struggle show the potential of including additional biological mechanism in existing algorithms to alter their behaviour.

Chapter 4

Epigenetic blocking

4.1 Borrowing from the Extended Evolutionary Synthesis

The Modern Synthesis [89], built from Darwin and Wallace's ideas of natural selection [38, 40], and Mendel's principles of inheritance [15], has been an important inspiration for the concepts used in Evolutionary Algorithms. Despite the number of approaches available, the core inspiration is often just genetic inheritance. However, modern evolutionary theory has since continued to explore the mechanisms of evolution, extending the Modern Synthesis to include concepts of non-genetic inheritance such as epigenetics, parental effects, multilevel selection, and cultural inheritance in a portfolio proposed as the Extended Evolutionary Synthesis.

In Chapter 2 the existing concepts from evolutionary theory were explored to determine which have been incorporated into Evolutionary Computation, and which have not. The concepts of cultural inheritance have been shown to be explored extensively in different mechanisms developed within Swarm Intelligence. Epigenetics in evolutionary biology has been demonstrated as a key component in rapid adaptation [10] of changing environments and central to genetic regulation and phenotypic plasticity, but Evolutionary Computation is not incorporating the same breadth of mechanisms. Chapters 3 explored the differences between Genetic Algorithms and Particle Swarm Optimisation algorithms. The diversity and generality of Genetic Algorithms and genetic inheritance is observed, compared to the convergence properties of Particle Swarm Optimisation and culture inheritance. The rapid adaptation of epigenetics can be seen as a strong convergence property, making Genetic Algorithms more suitable compared to Particle Swarm Optimisation algorithms.

Epigenetic mechanisms in evolutionary theory can alter DNA expression, leading to a change in phenotype without a change in the underlying genotype [50]. This "phenotypic plasticity" leads to a faster rate of change to quickly adapt to changes in the environment, and the ability to revert changes if the environmental conditions do not activate the epigenetic mechanism. This is to allow more rapid adaptation to a natural world that is changing, optimising an organism's fitness without altering its underlying genotype. In multi-objective dynamic problems, a similar set of challenges is reflected by changing the optimal Pareto set or Pareto front over time [60]. This chapter explores a novel epigenetic blocking mechanism, and how it might be used within Evolutionary Algorithms to improve performance on static and dynamic multi-objective benchmark problems.

4.2 Epigenetic blocking mechanism

There are three forms of epigenetic transfer possible: mitotic, germline, and experience-dependent [25, 190]. Germline transfer passes down epigenetic marks that direct the epigenetic process for future generations, while mitotic transfer only propagates changes in the same generation. Experience-dependent epigenetic changes are induced through exposure to specific environmental conditions that trigger the change. For an exploration of epigenetic mechanisms, genetic blocking is chosen for inspiration based on germline transfer. A probabilistic blocking mechanism is used to block some number of decision variables in each individual from being changed during crossover, where parent variables are normally crossed over to create child solutions. The chosen variables form an epigenetic mask which can be inherited by child solutions. In the case where both parents have an epigenetic mask, the masks are combined together. However, the parameters of the epigenetic process are not inherited across multiple generations, remaining the same throughout (Figure 4.1). A simple mechanism allows the properties to be analysed and to more clearly understand the effect of an epigenetic process on the performance of an algorithm. The simplicity also allows the mechanism to be adapted to any Evolutionary Algorithm by altering the crossover method.

Estimation of Distribution Algorithms (EDAs) are optimisation algorithms with a similar mechanism to epigenetic blocking. Unlike Evolutionary Algorithms, EDAs use a probabilistic model to learn from good solutions within the population and sample new solutions using the probabilistic model. In univariate EDAs, each decision variable is sampled with an independent distribution while multivariate EDAs use a joint probability distribution to reflect dependencies and relationships between variables [83]. Similarly in epigenetic blocking, impactful decision variables can be identified and propagated to future generations, allowing specific decision variables to be modified or retained. However, the distribution models used in EDAs are global, affecting the entire population as each generation is sampled from the global distribution. In contrast, the epigenetic blocking mask is inherited and passed down by individual solutions, utilising the selection pressure of Evolutionary Algorithms propagate epigenetic masks.

The mechanism has a probability to trigger during the reproduction stage for every parent without bias towards fitness, as blocking both the fitter and less-fit parents have merits. Blocking fitter parents reduces stagnation of the population, i.e. maintains diversity, should the variables or objectives of the dynamic problem change, while blocking less fit parents increases the convergence of the population through more rapid selection at the variable level.

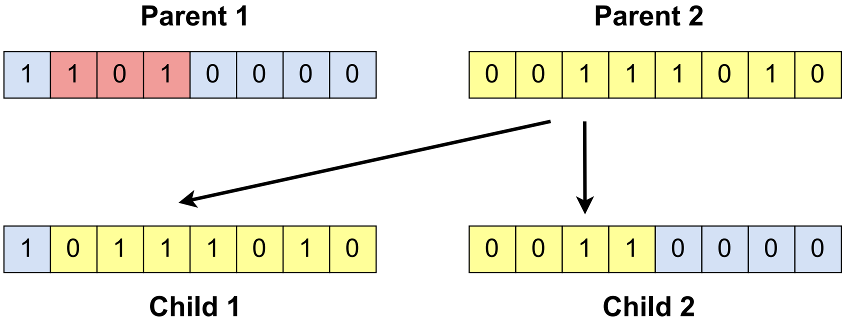


Figure 4.1: A simplified view of the blocking mechanism where some variables are blocked from carrying over to the next generation. The blue and yellow blocks are parent 1 and parent 2's variables. In a typical single point crossover, the two child solutions inherit half of the variables from each parent. The red represents decision variables blocked from being inherited, leading to child 1's variables to contain more variables from parent 2.

This probabilistic mechanism is not fully epigenetic, as changes in the problem are not directly reflected in how the mechanism reacts. However, by varying the probability and the number of genes that are blocked, and controlling the duration of each dynamic cycle in a problem, the impact of epigenetic blocking can be analysed and compared to a baseline algorithm and to a constant probability of triggering the mechanism. The key feature of inheritance is included, allowing the epigenetic mask of blocked variables to be passed on. Importantly, the variables are not always blocked despite the inheritance of the mask, effectively allowing the chosen variables to be blocked or unblocked within two generations, greatly speeding up the rate of adaption to environmental changes. The selection pressure of Evolutionary Algorithms further ensures inherited masks are blocking variables which impact the fitness of the solution.

4.2.1 Algorithm

The steps of the epigenetic blocking process are shown in Algorithm 2. After the crossover step is completed, the variables chosen to be blocked in the offspring are set to the unblocked parent variables. The step prevents the blocked parent variables from passing onto the offspring only at the selected positions. The chosen positions are saved as an epigenetic mask and passed onto the offspring. The mask is inherited even if the epigenetic mechanism does not take effect. Furthermore, if both parents contain an inherited epigenetic mask, the two masks are combined and randomly sampled to create a combined mask of the set mask length. In some cases, the selected variable positions may only contain variables from the unblocked parent after crossover. In these cases, the epigenetic mechanism will not take effect, as no blocked variables are present in the offspring solution.

Algorithm 1 Epigenetic step after crossover where ER is the probability for epigenetic blocking to occur, the $maskLength$ is pre-defined based on the number of decision variables in a given optimisation problem, and both are globally defined by the algorithm. If either parent have a mask, or the epigenetic mechanism is trigger and a new mask is created, the child inherits the epigenetic mask.

```
function REPRODUCTION($parent1$, $parent2$)
    $offspring \leftarrow$ CROSSOVER($parent1$, $parent2$)
    $epigeneticMask \leftarrow$
    EPIGENETICBLOCKING($offspring, parent1, parent2, maskLength, ER$)
    if length($epigeneticMask$) $> 0$ then
        $offspring.epigeneticMask \leftarrow epigeneticMask$
    end if
    MUTATION($offspring$)
    return $offspring$
end function
```

The epigenetic mask can be made up of a contiguous set of variables, such as $\{x_1, x_2, x_3, x_4\}$ for a decision vector of length 10 and a mask length of 4. Alternatively, the epigenetic mask can be non-contiguous and consist of randomly selected positions, for example, $\{x_4, x_6, x_9, x_{10}\}$. Note that mutation steps continue to occur after the epigenetic process.

4.3 Benchmarking the epigenetic mechanism

In the development of epigenetic mechanisms for Evolutionary Algorithms, rigorous benchmarking is performed to test the performance of the algorithms. This involves having a wide range of benchmark problems, statistical tests, performance metrics, and qualitative visualisations.

The work in the previous chapters comparing Evolutionary Computation methods provides a foundation for benchmark problems and metrics to measure and analyse the performance of the epigenetic mechanism. The same benchmark problems and performance metrics as before are used here, namely ZDT [47], WFG [88], LZ09 [114], CEC 09 (UF) [218], DTLZ [48], MOP [120], IMB [119], and DAS-CMOP [59]. Similarly, the Wilcoxon ranked sum test with a p-value of 0.05 is also used to determine whether the performance of an algorithm with the epigenetic mechanism is statistically significant compared to the performance without the mechanism. Both the IGD metric and the HV value is used to quantitatively determine performance. An unbounded external archive is used to store the best solutions found by each algorithm, which is re-evaluated in dynamic problems at each time step.

Two variations of the epigenetic mechanism are tested: $non - cont$ and $cont$. The $non - cont$ variant uses a mask of non-contiguous decision variables while the $cont$ vari-

Algorithm 2 Creating and applying the epigenetic mask to block variables specified in the mask. If either parent has an epigenetic mask, use the parent's mask instead.

```
function EPIGENETICBLOCKING($offspring$, $parent1$, $parent2$, $maskLength$, $ER$)
    Initialise $epigeneticMask$ as a zero length array
    if $parent1$ & $parent2$ has $epigeneticMask$ then
        $combinedMask \leftarrow parent.epigeneticMask \cup parent2.epigeneticMask$
        $epigeneticMask \leftarrow$ RANDOMSAMPLE($combinedMask, maskLength$)
        $epigeneticParent \leftarrow$ RANDOMPARENT($parent1$, $parent2$)
    else if $parent1$ has $epigeneticMask$ then
        $epigeneticMask \leftarrow parent1.epigeneticMask$
        $epigeneticParent \leftarrow parent1$
    else if $parent2$ has $epigeneticMask$ then
        $epigeneticMask \leftarrow parent2.epigeneticMask$
        $epigeneticParent \leftarrow parent2$
    end if
    if $ER$ then
        if length($epigeneticMask$) $< 0$ then
            $epigeneticMask \leftarrow$ RANDOMARRAY($maskLength$)
        end if
        for $varIndex$ in $epigeneticMask$ do
            $offspring.variables[varIndex] \leftarrow epigeneticParent.variables[varIndex]$
        end for
    end if
    return $epigeneticMask$
end function
```

ant uses a mask of contiguous decision variables. The same experimental methodology as Chapter 3 is used, with 100,000 function evaluations and 20 independent runs. Results are recorded every 1,000 function evaluations to provide the performance over time. The best solutions found throughout are saved into a non-dominated solutions archive. The crossover mechanism employed is not changed from the base algorithms, for example NSGA-II employs SBX and MOEA/D-DE employs a DE crossover. In addition to the same benchmarking problems and metrics, a new set of dynamic multi-objective problems is used.

4.3.1 Dynamic multi-objective optimisation problems

Dynamic multi-objective optimisation problems use an additional time step as an external variable to alter either the Pareto set or the Pareto front. The dynamic nature is captured by an explicit time variable t. Each increment of the time variable is referred to as a time step. There are two factors that influence a dynamic problem, severity and frequency. Severity, n_t, refers to the magnitude of each dynamic change. Smaller values of n_t correspond to larger changes in the time variable at each step. Frequency, τ_t, affects how often a dynamic change occurs. For example, A τ_t value of 10 means that dynamic changes occur every 10 generations. Generations refers to the evaluation of a full population of candidate solutions. As algorithms with different population sizes would cause out of sync comparisons in time step changes, both algorithms evaluated use the same population size of 100. The value of t is computed as:

$$t = \frac{1}{n_t} \cdot \frac{g}{\tau_t} \tag{4.1}$$

where g is the current generation counter. This time value is then used to alter the problem with, typically defined as a function $G(t)$ in benchmarking problems as:

$$G(t) = sin(\frac{1}{2}\pi \cdot t) \tag{4.2}$$

where g is the number of generations produced by an algorithm.

To benchmark against dynamic multi-objective problems, dynamic variants of the MOEA/D-DE and NSGA-II algorithms are used to give a fairer comparison. The MOEA/D-DE algorithm [114] with the re-initialisation strategy outlined in [20], and the dynamic NSGA-II (D-NSGA-II) as outlined in [46] are implemented as the baseline algorithms, rather than using a non-dynamic version of the algorithms. The D-NSGA-II implementation in the jMetalpy library is used, and the MOEA/D-DE implementation in jMetalpy is modified to include the re-initialisation strategy. In particular, the fast converging nature of MOEA/D has been studied in dynamic problems in the past, and shown to adapt quickly in changing environments [96]. An initial probability of 0.1 for the epigenetic blocking mechanism is used, and a constant block size of number of variables / 5 are used as arbitrary values to test the mechanism. Further exploration of these parameters will be performed in the next Chapter 5. For the problem parameters, the severity value is set at $n_t = 10$, and frequency value at $\tau_t = 5$ for all problems used, as is the most common recommended setting in the benchmarking suites [20, 60]; this gives 5 generations before the problem changes and 10 distinct steps. In total there are 100 generations (50,000 iterations) to complete a full cycle back to the original variables and objectives of the problem.

The FDA [60], JY [96], UDF [20], and CDF [73] benchmark functions are chosen for the dynamic multi-objective test of problems. The FDA and UDF functions are considered to be simpler unconstrained problems as they are based on the multi-objective ZDT [47] problems while the JY problems are more complex [95], including elements such as linkage between variables and multiple knee points. The CDF problems are based on the CF [218] and UDF problems with both dynamic and static constraints.

The dynamic problems can be categorised based on how the Pareto set and Pareto front changes over time, forming four types of dynamic problems.

1. The Pareto set changes dynamically over time;

2. the Pareto set and Pareto front change dynamically over time;
3. the Pareto front changes dynamically over time;
4. the Pareto set and Pareto front remain static, but the constraints or objective function can change dynamically over time.

Appendix A categorises the dynamic problem sets and describes the shifts that occur to the Pareto front and Pareto sets. Vertical and horizontal shifts are simpler, moving the Pareto front or Pareto set in one axis while angular shifts represent a change in multiple axes and are more complex to solve. To differentiate between the categories of static problems and the new categories of dynamic problems, Roman numerals are used for the static problem categories, and numbers are used for the dynamic problem categories.

4.3.2 Measuring performance

Performance metrics such as the IGD and HV values are obtained throughout an algorithm's optimisation process, not only the final value, to show how an algorithm performs over the course of each dynamic change.

A reference Pareto front to compute the IGD is created for every time step. This ensures the IGD calculation follows the dynamic changes of the problem. As the performance metrics are captured every 1,000 evaluations, the performance metrics before and after dynamic changes in the problem will be recorded. The same approach is applied to the HV metric, where the HV value is recorded at each step to calculate average performance over time.

4.4 Performance on static multi-objective optimisation problems

First, the static multi-objective benchmark problems are tested on MOEA/D and U-NSGA-III to compare the performance with and without the epigenetic mechanism. All the same benchmark problems from Chapter 3 are used. Figures 4.2 and 4.3 show the average performance score between the baseline algorithm and the algorithm with the epigenetic mechanism included for 100,000 function evaluations for MOEA/D-DE and U-NSGA-III respectively. The problems are categorised following the same categories as Chapter 3, the details of which are displayed in Tables A.1 to A.10 in Appendix A. The average performance score of one algorithm is a count of the number of times another algorithm outperforms it. A lower average performance score indicates better performance compared to the other algorithms.

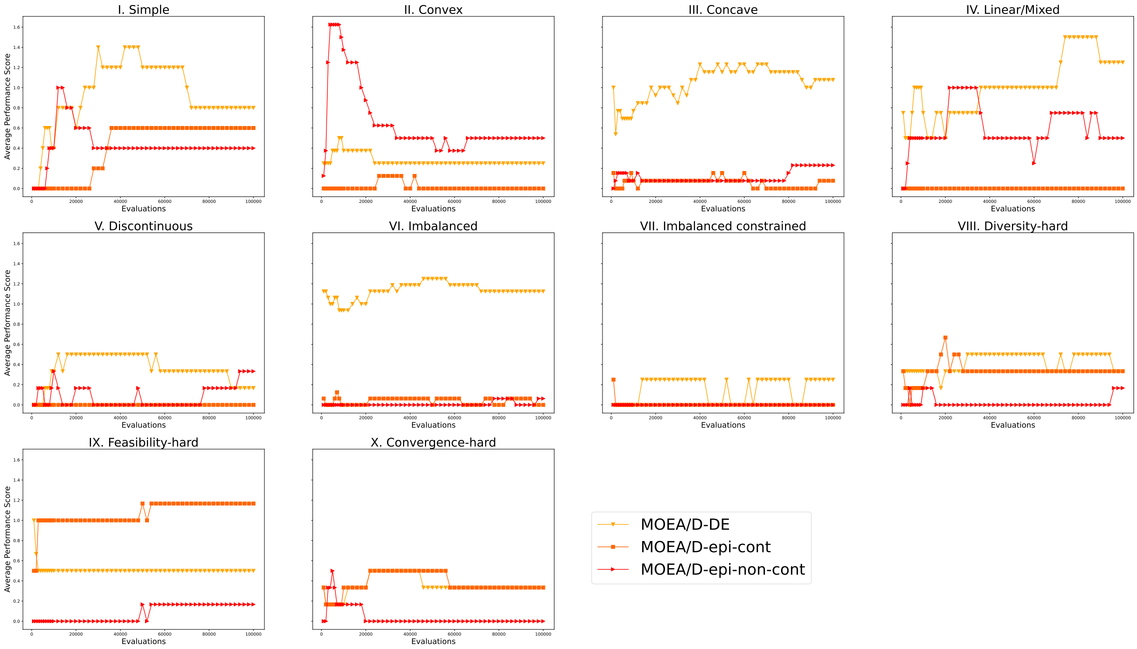


Figure 4.2: The average performance scores of MOEA/D with epigenetics on static benchmark problems, grouped by problem set, calculated by HV.

The performance of MOEA/D-DE with epigenetics shows improved performance over most of the benchmark problems, notably the simple, concave, and imbalanced categories of problems. On concave and imbalanced problems, there is a clear difference between the baseline MOEA/D-DE algorithm and the epigenetic variants, where the epigenetic mechanism displays consistently better performance throughout all 100,000 function evaluations. In other categories such as the simple and discontinuous problems, MOEA/D-DE with epigenetics provided better performance from the early to middle segments of the optimisation, around 10,000 – 60,000 function evaluations, after which the baseline MOEA/D-DE algorithm catches up, and for example in discontinuous problems, overtakes one of the epigenetic variants.

A notable difference can be seen between the use of a contiguous and non-contiguous epigenetic mask. For example, on all three types of DASCMOP problems, diversity-hard, feasibility-hard, and convergence-hard, a non-contiguous epigenetic mask displayed the best performance. This is likely because the non-contiguous mask allows the epigenetic mechanism to block non-adjacent decision variables to promote further diversity. Especially in the case of DASCMOP problems, where the Pareto front is fragmented, or there are tight feasibility constraints on the valid variable space, the use of non-contiguous masks allows broader exploration to solve these complex problems. In contrast, a contiguous mask showed comparable results to the baseline MOEA/D-DE on diversity-hard and convergence-hard problems, and performed worse in the feasibility-hard problems, indicating its disadvantage in complex and fragmented search spaces, where blocking masks on variables that are less correlated in sequence is disadvantages. However, on convex and linear problems, the contiguous epigenetic mask had better performance than the

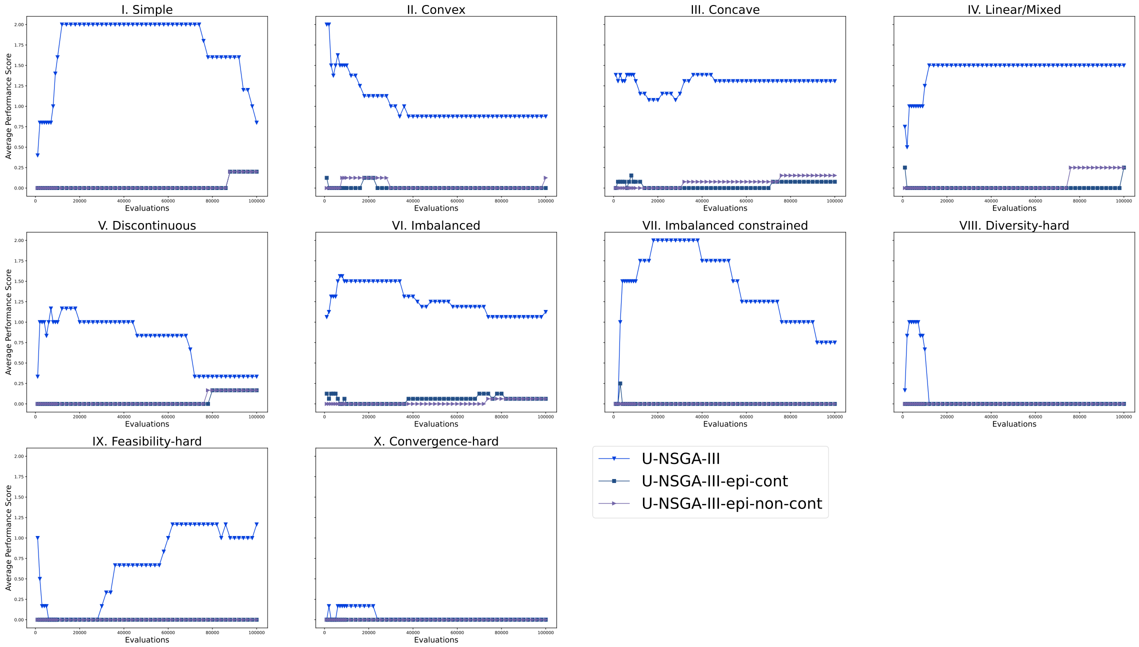


Figure 4.3: The average performance scores of U-NSGA-III with epigenetics on static benchmark problems, grouped by problem set, calculated by HV.

non-contiguous mask due to the way those benchmark problems are setup with decision variables that are correlated in sequence. For example, in DTLZ1 the decision vector is formed of two ‘blocks’, the first o variables where o is the number of objectives form the position variables, then the next $n - M + 1$ variables where n is the total number of variables form the distance variables which is a contiguous block that affects the distance function g. Similarly, the WFG problems use position and distance variables within the decision vector for form the Pareto front, creating distinct contiguous blocks of variables in which the contiguous-blocking mask of the epigenetic-mechanism has a better effect.

The performance improvements gained from the epigenetic mechanism can be seen for U-NSGA-III in Figure 4.3. Here, performance improvements are shown for every problem category. For some categories such as the concave and imbalanced problems, U-NSGA-III with the epigenetic mechanism provides consistent improvement throughout the 100,000 function evaluations. On other categories such as the discontinuous and simple problems, the baseline U-NSGA-III performance becomes more comparable near the end of the optimisation. These results are similar to the MOEA/D-DE results, indicating the epigenetic mechanism is able to apply the same effects to both algorithms. Interestingly, there is little difference between non-contiguous and contiguous epigenetic masks, both showing similar performance scores throughout, which differs from MOEA/D-DE. This can be explained by the algorithm’s use of reference vectors along with crowding distance selection to maintain diversity. The niching-based selection and elitism in U-NSGA-III may suppress the differences that mask configuration would otherwise introduce. However, in the decomposition-based MOEA/D-DE, there is no such structured mechanism to maintain diversity, which allows the epigenetic mask to have a more visible impact.

The performance of the epigenetic mechanism can be split into two types: better performance throughout, and better performance at the beginning with mixed performance towards the end. The specifics of each will be discussed and analysed in the following subsections.

4.4.1 Better performance throughout

On both algorithms, concave and imbalanced problems showed better performance of the epigenetic mechanism throughout. Figure 4.4 displays the HV values of select problems in the concave category, and Figure 4.5 displays the HV values of select problems in the imbalanced category. A few selected problems from each category are shown to highlight the performance differences. MOEA/D-DE and U-NSGA-III are separated to focus on the performance differences between the baseline algorithms and the epigenetic variants, and not on the performance differences of the two algorithms.

The same pattern of performance by the epigenetic mechanism is demonstrated across the selected concave problems for both MOEA/D-DE and U-NSGA-III. A performance gap between the epigenetic variants and the baseline algorithms is seen early in the optimisation, from 10,000 – 30,000 evaluations. This early performance difference is most notable in DTLZ4. For MOEA/D-DE, problems such as UF4 show the improved performance carrying on throughout the optimisation, with the baseline MOEA/D-DE HV value statistically significantly lower than the two epigenetic variants. Many of the other problems show only small differences between MOEA/D-DE and the epigenetic variants.

The strong performance throughout is more clear for U-NSGA-III. For example in WFG6 and WFG8, the baseline U-NSGA-III algorithm plateaus at a lower HV value than the epigenetic variants. In the UF4 and UF8 problems, the HV values are much closer but there is a distinct difference that can be seen, and there is still a statistical significance. Finally, for DTLZ4, the baseline U-NSGA-III and the epigenetic variants plateau at a similar HV value with no statistical significance, but the epigenetic mechanism reaches those values much earlier in the optimisation, where the large performance gap between 10,000 – 30,000 can be seen.

Figures 4.5 and 4.8 illustrate the same performance patterns in the imbalanced category of problems. A few anomaly problems are displayed too, such as IMB1 and MOP1 where the baseline MOEA/D-DE achieves high HV values, or IMB2 for U-NSGA-III.

For MOEA/D-DE, many problems show a drastic improvement in performance for the epigenetic variants, starting from early in the optimisation and throughout all 100,000 function evaluations. In some cases such as MOP6 and IMB3, the absolute differences are small, less than 0.01 different in HV values, but a clear pattern emerges nonetheless. Two major anomalies are noted where the baseline MOEA/D-DE finds much better performance: MOP1 and IMB1. Both are convex problems with 10 decision variables and

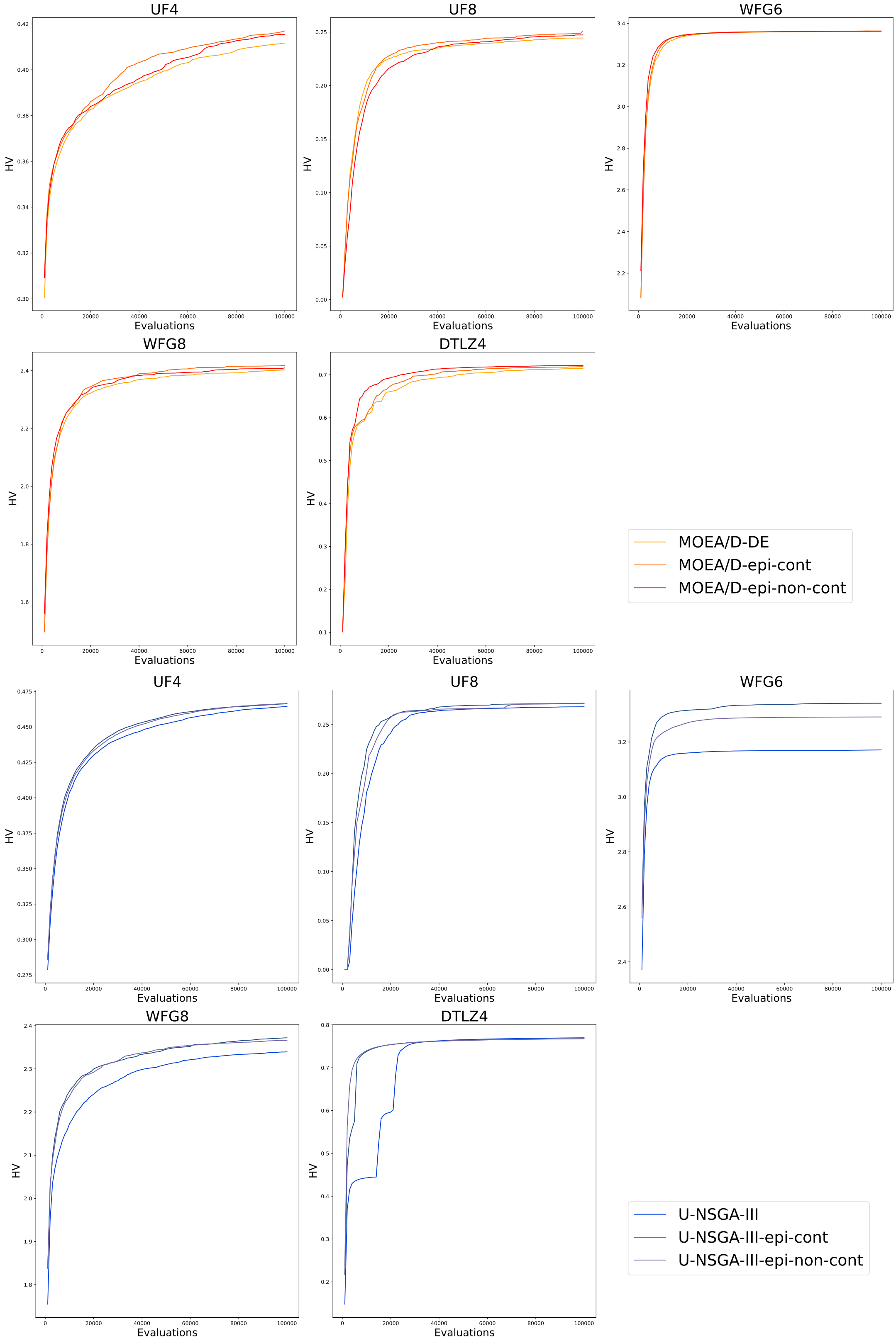


Figure 4.4: Median HV values over time of selected benchmark problems from Category III. Concave problems

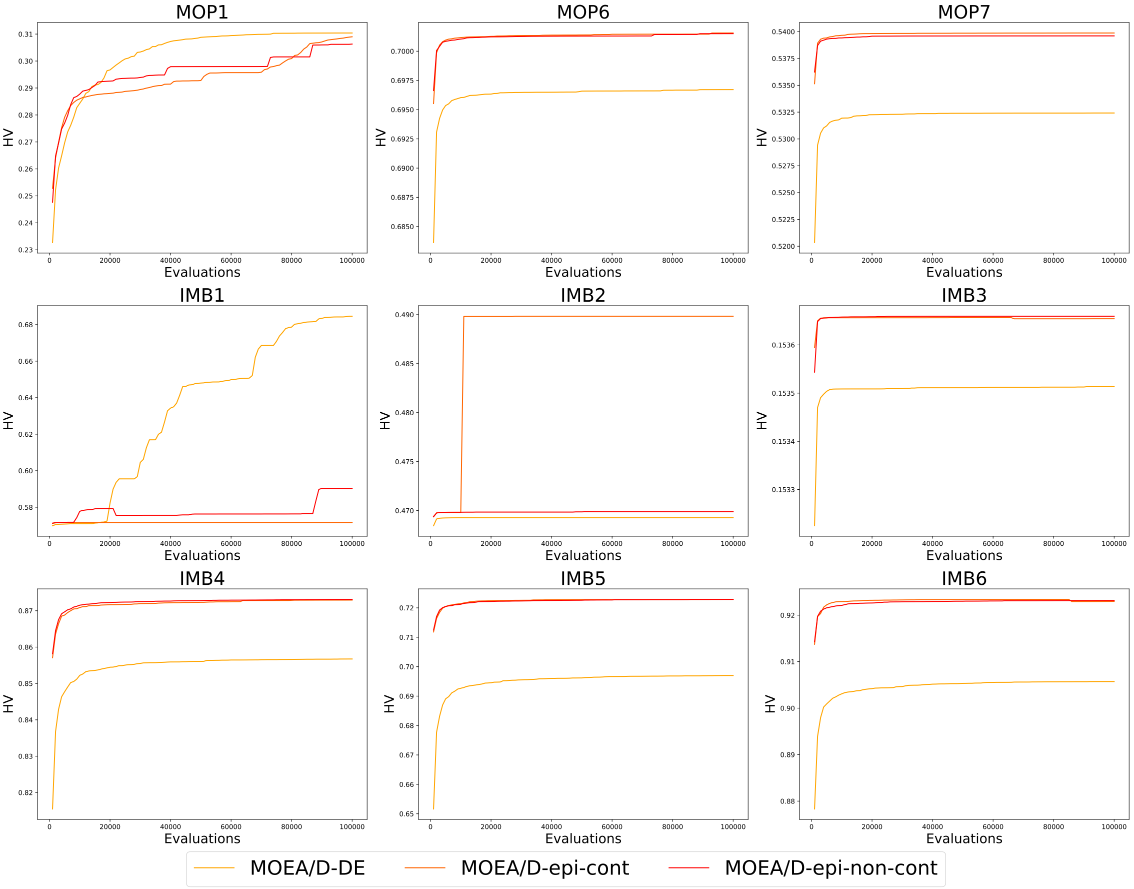


Figure 4.5: Median HV values over time of selected benchmark problems from Category VI. Imbalanced problems for MOEA/D-DE.

2 objectives. As Figure 4.2 showed, the epigenetic mechanism does not perform well for MOEA/D-DE on convex problems. The number of objectives may also play a role as many problems in the concave category are 3-objective problems, and the IMB4, IMB5, IMB6 problems are also 3-objective. Furthermore a majority of convex problems are 2-objective.

Figure 4.7 averages the performance scores from all 2- and 3-objective problems. The epigenetic variants finds some success towards the middle sections of the optimisation, but the use of non-contiguous masks performs similarly to the baseline algorithm. Further, the baseline MOEA/D-DE on 2-objective problems achieved from 0.4 – 0.6 average performance score, indicating it is not far behind the epigenetic variants. In contrast on the 3-objective problems, the baseline MOEA/D-DE had over 1.0 average performance score at many points of the optimisation, meaning it was out-performed by at least one epigenetic variant at almost all times.

The same effect of the epigenetic mechanism on 2- and 3-objective problems can be seen for U-NSGA-III to a lesser extent. There is a a clear improvement for U-NSGA-III on both 2- and 3-objective problems, but the performance difference is larger for 3-objective problems, achieving over 1.2 average performance score for the entire 100,000 function

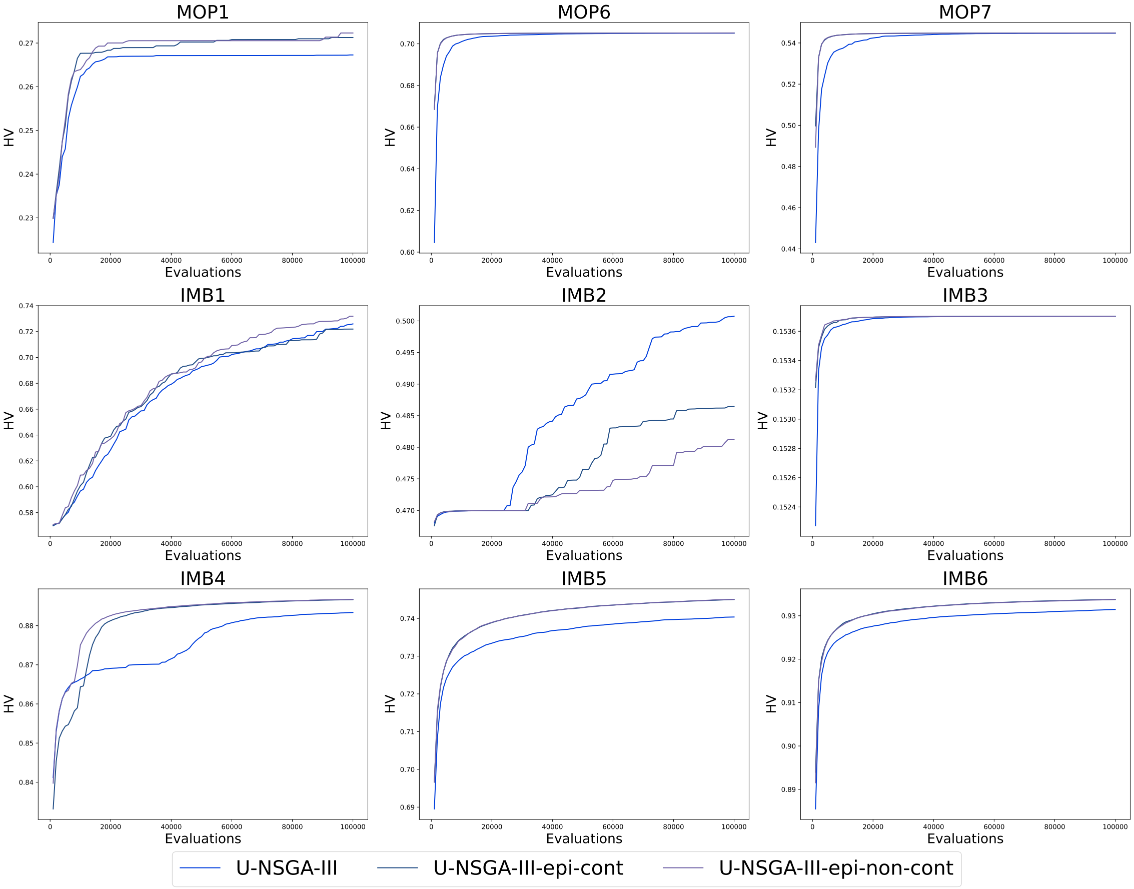


Figure 4.6: Median HV values over time of selected benchmark problems from Category VI. Imbalanced problems for U-NSGA-III.

evaluations.

The epigenetic mechanism is shown to be effective for U-NSGA-III on the imbalanced problems in Figure 4.8. Two forms of performance improvement can be seen, increased convergence, and improved final solutions. For example, MOP6 and MOP7 show increased convergence as both the baseline U-NSGA-III and epigenetic variants find the same final solutions, but the epigenetic variants found the final solutions in fewer function evaluations. In other examples such as MOP1, IMB4, IMB5, and IMB6, not only do the epigenetic variants have increased convergence, they also find the better solutions as noted in the final curves. The anomaly for U-NSGA-III on imbalanced problems is IMB2. Here, the algorithm performs similarly until 25,000 function evaluations, where the baseline U-NSGA-III achieves a 0.03 increase in HV value while the epigenetic variants only find a 0.02 increase. While IMB2 is a 2-objective problem, this is unlikely the reason for the epigenetic mechanism's poor performance, as its performance on other imbalanced 2-objective problems such as IMB7 – 13 all show increased convergence or improved final solutions, or in some cases both.

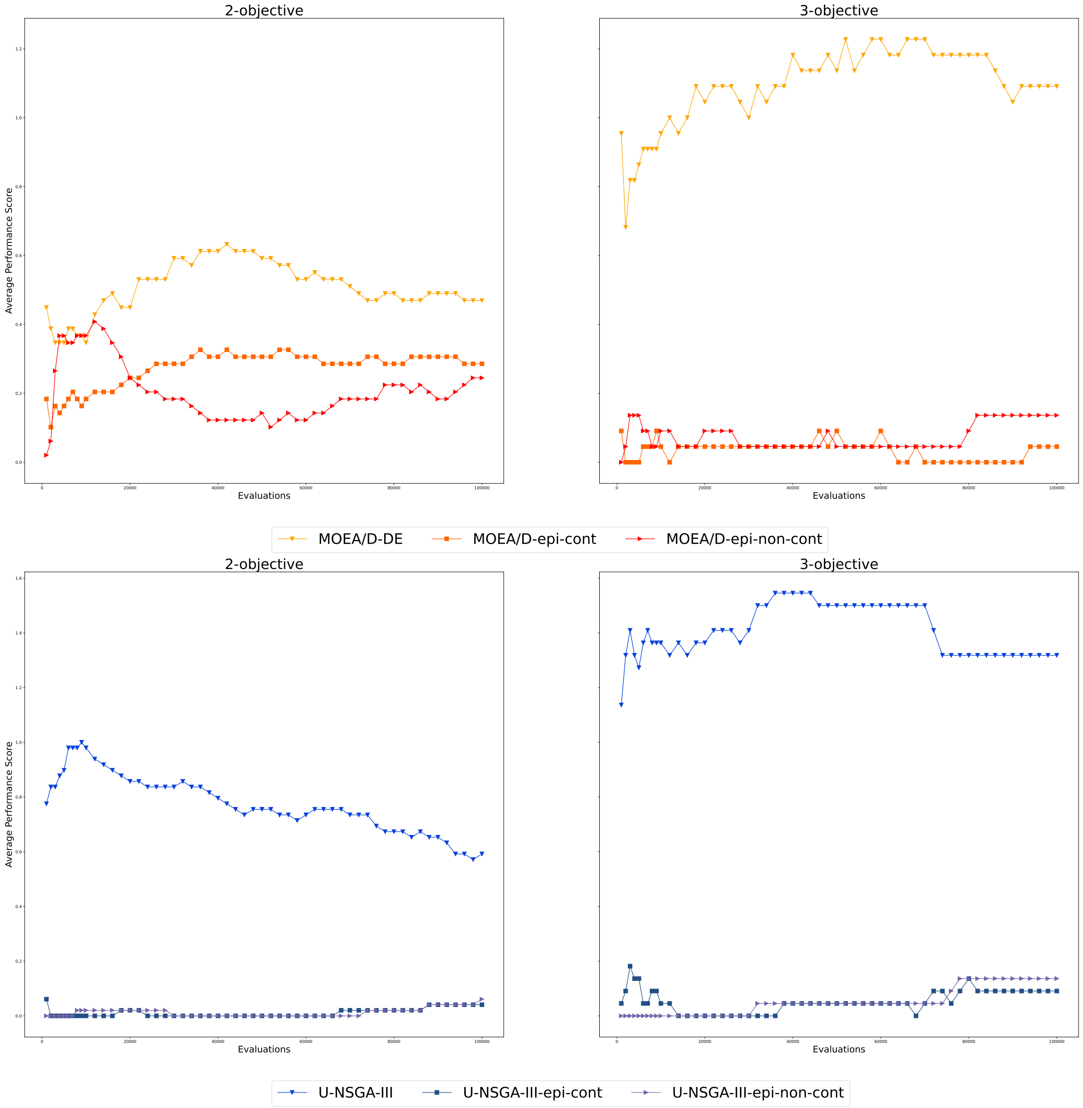


Figure 4.7: Average performance scores of MOEA/D-DE and U-NSGA-III with epigenetics on 2- and 3-objective problems, calculated by HV values.

4.4.2 Better performance at the beginning

The behaviour showing faster convergence at the start of the optimisation problem can be seen in a number of the previous examples, where there is a clear curve reaching higher HV values with fewer function evaluations. The average performance score Figures 4.2 and 4.3 indicated that the simple and discontinuous problem categories exhibited the stronger convergence behaviour the most.

Figure 4.9 shows the performance over time curves for MOEA/D-DE and U-NSGA-III on the ZDT simple set of problems. For each problem, the baseline algorithm and the epigenetic variants reach similar optimal solutions after 100,000 function evaluations as indicated by the overlapping flat lines. However, the epigenetic variants are able to

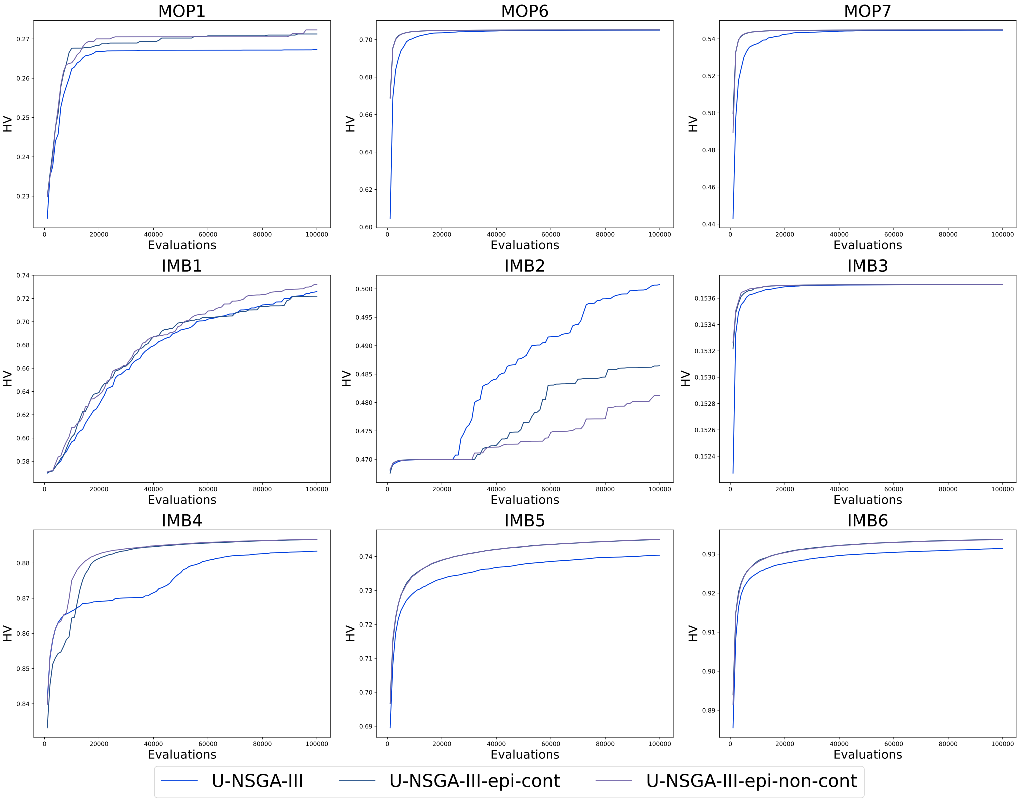


Figure 4.8: Median HV values over time of selected benchmark problems from Category VI. Imbalanced problems for U-NSGA-III.

find those solutions earlier. The difference is much more obvious for U-NSGA-III, with a clear gap between the baseline algorithm and the epigenetic variants. The differences in performance mostly come from 10,000 – 30,000 function evaluations, similar to previous examples, showing the recurring behavioural pattern across a number of different problems. In some problems such as ZDT4 and ZDT6, the difference in performance is larger and last longer until 50,000 function evaluations. Both the contiguous and non-contiguous epigenetic masks perform similarly. The same differences are not as clear for MOEA/D-DE. The epigenetic variants still exhibit faster convergence, but the performance of the baseline algorithm is much closer. Furthermore, there is a noticeable difference between the use of a contiguous and non-contiguous mask for MOEA/D-DE. For example, the contiguous epigenetic mask has the best performance on ZDT4 and ZDT6, although the non-contiguous mask finds early success for ZDT6 in the first 5,000 – 10,000 evaluations.

The results for discontinuous problems shows the same performance with a few interesting aspects. Figure 4.10 shows the IGD values rather than the HV values as the results on MOP4 are far from the true Pareto front resulting in low HV values. The fast convergence behaviour is seen for example on UF6 for MOEA/D-DE and DTLZ7 for U-NSGA-III. Interestingly, the use of a contiguous and non-contiguous mask plays a larger role and its effect differs based on the algorithm. The contiguous mask achieves higher performance on WFG2 for U-NSGA-III but has no difference on the same problem for

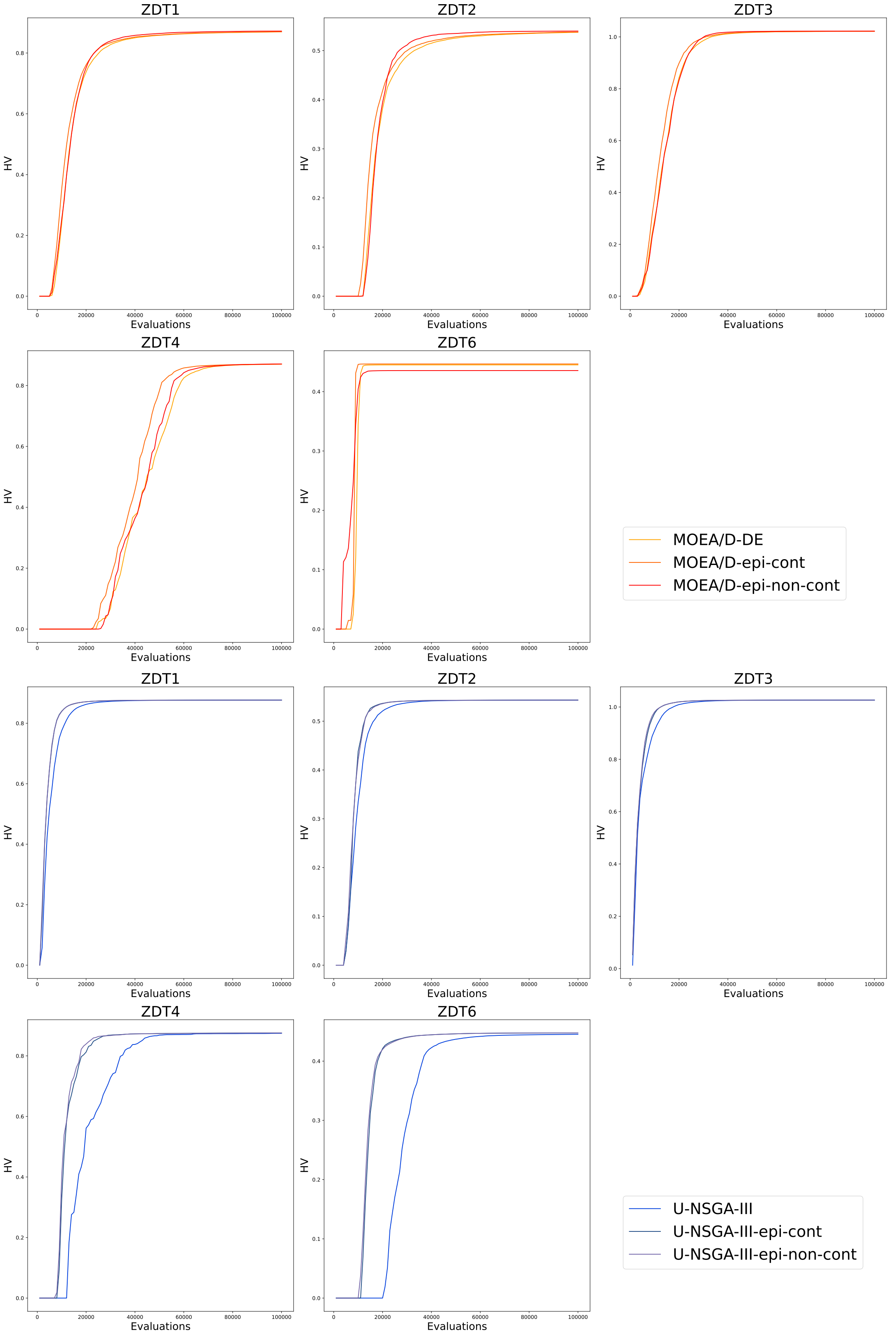


Figure 4.9: Median HV values over time of selected benchmark problems from Category I. Simple problems.

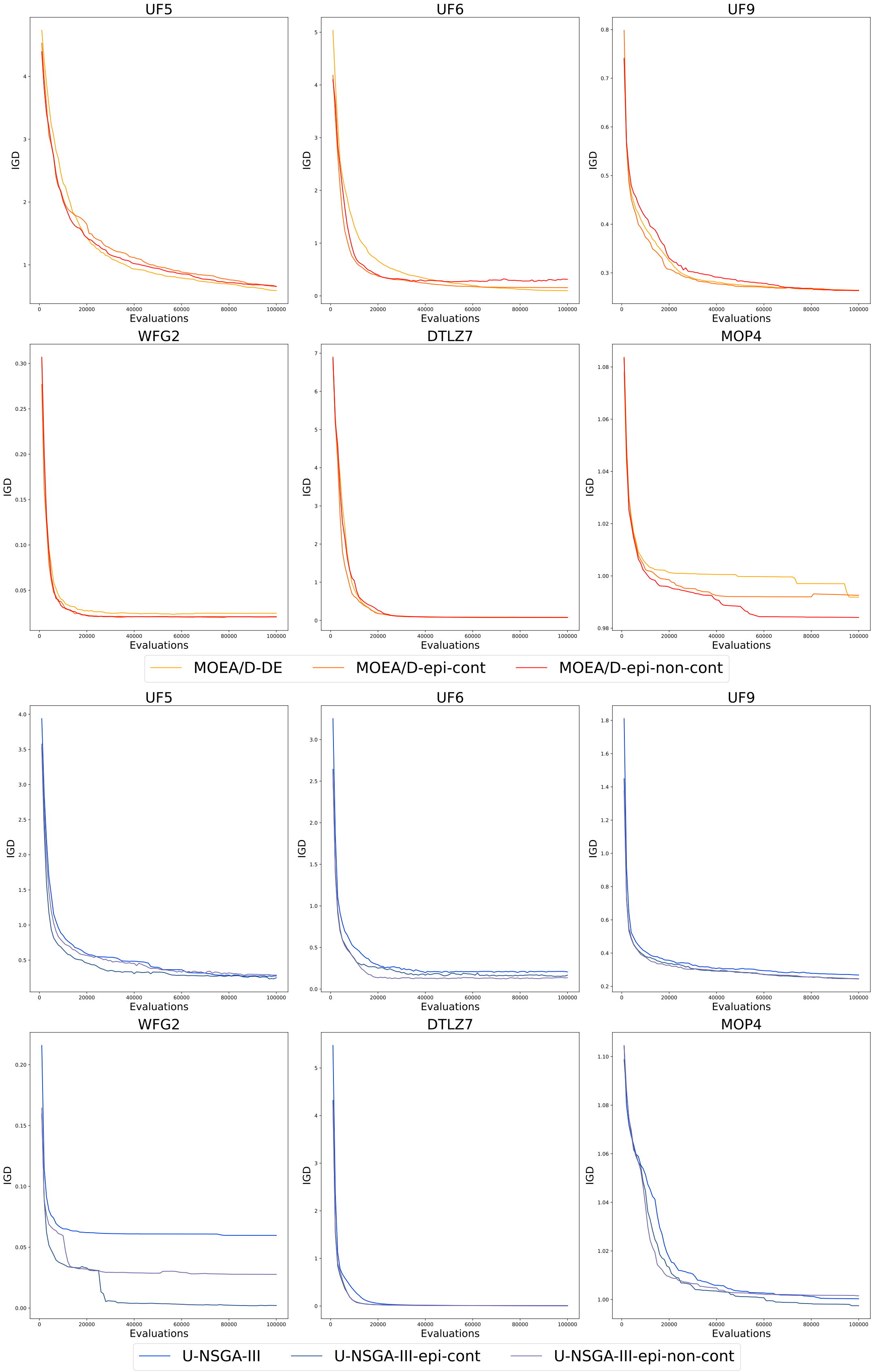


Figure 4.10: IGD values over time of selected benchmark problems from Category V. Discontinuous problems.

MOEA/D-DE. The non-contiguous mask is better on MOP4 for MOEA/D-DE, yet the contiguous mask achieves that performance for U-NSGA-III.

4.4.3 Differences between a contiguous and non-contiguous epigenetic mask

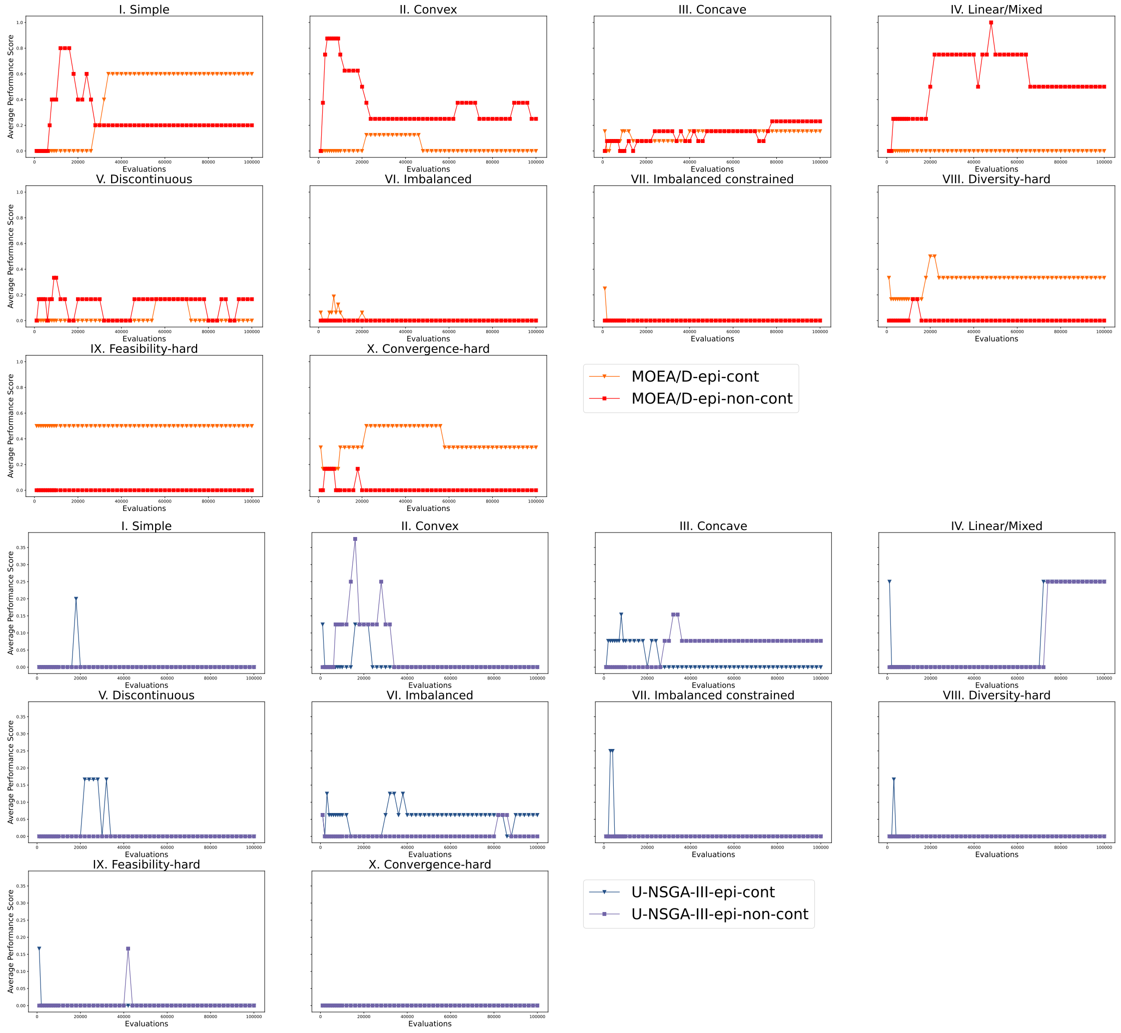


Figure 4.11: Average performances scores comparing the contiguous and non-contiguous epigenetic masks

The use of an epigenetic blocking mechanism is able to provide performance benefits in the MOEA/D-DE and U-NSGA-III genetic algorithms on a number of static benchmark problems. Two epigenetic variants are tested, one with a contiguous epigenetic mask and one with a non-contiguous epigenetic mask. Both show similar performance improvements compared to the baseline algorithms. Next the two variants are compared to each other to evaluate their differences and determine if one variant outperforms another, or if the effectiveness of a contiguous or non-contiguous mask is problem-dependent.

The average performance scores of the two epigenetic variants are shown in Figure 4.11. In general there are few differences between the two variants. If there is no statistically significant difference between the two epigenetic variants, both would score a 0 on the performance score calculation. For MOEA/D-DE, four out of the ten categories show low average performance scores for both variants, indicating little to no performance differences for concave, discontinuous, imbalanced, and imbalanced constrained problems. Additionally, of the other problem categories where more significant differences are found, different variants may outperform another depending on the problem category. For example, in convex and linear/mixed problems, the contiguous epigenetic mask outperforms the non-contiguous epigenetic mask, maintaining an average performance score of 0 throughout almost the entire 100,000 function evaluations. Conversely, for the DASCMOP set of problems in diversity-hard, feasibility-hard and convergence-hard problems, the use of a non-contiguous epigenetic masks provides performance benefits over the contiguous mask, again throughout all 100,000 evaluations. As discussed earlier, these patterns of performance are based on the way the benchmark problems are structured, leading to one epigenetic variant outperforming the other on different categories of problems. The DAS-CMOP problems are complex with hard feasibility, convergence, and diversity properties. Many of the feasible region are disconnected or constrained which is advantageous for the non-contiguous due to its ability to block non-adjacent variables, helpful to escape infeasible areas and exploring with more diversity. In contrast, the convex and linear problem structures reward contiguous masking, as variables are set in sequence and blocks of different variables, such as position and distance variables.

The results on the simple set of ZDT problems are the most interesting, where the contiguous mask performs best at the early stage of optimisation, before 30,000 evaluations, and the non-contiguous mask performs better at the later stages. This highlights the potential difference between the epigenetic variants relating to exploration and exploitation, where the non-contiguous mask favours exploration and the contiguous mask favours exploitation. These findings suggest that the choice of epigenetic mask should depend on the problem structure and decision variable design. Contiguous masks are better suited to correlated search spaces with decision variables relating in sequence to each other, while non-contiguous masks are better suited in scenarios with complex constraints or irregular and unrelated decision variables.

A similar result is seen for U-NSGA-III. Notably the average performance scores for U-NSGA-III are in much lower than that of MOEA/D-DE. At least six of the ten categories show no performances differences except a few small points of anomalies. Of the other four categories where some performance differences are seen, the contiguous mask outperforms the non-contiguous mask on concave and early portions of convex problems, while the non-contiguous mask outperforms on the imbalanced problems. In the linear/mixed problems, both variants have the same higher average performance scores after 70,000 function evaluations, which shows both have outperformed the other on a different problem in the same problem category. These results demonstrate the difference the underlying algorithm can make to the effectiveness of the epigenetic blocking mechanism. U-NSGA-III use of niching and reference-point selection reduces the possible variance between the two epigenetic variants, leading to similar positive results for both variants against the baseline.

4.4.4 Discussion of epigenetic performance on static problems

The epigenetic approaches are shown to outperform the baseline algorithms on a number of different problems and problem categories. A larger effect is seen on the 3-objective problems compared to the 2-objective problems, suggesting the mechanism is capable of handling more complex problems, and fewer benefits are found for simpler problems. The most notable gains are observed in concave and imbalanced problems, where the epigenetic approach consistently outperforms the baseline algorithms throughout all function evaluations. For simpler and discontinuous problems, the epigenetic mechanism provides an early advantage, increasing convergence between 10,000 and 60,000 function evaluations. However, as the optimisation continues, the baseline algorithms often close the performance gap, and in some cases, even surpass the epigenetic variants. This trend suggests that while epigenetics can help guide the initial search efficiently, its impact diminishes over time with more evaluations.

Few significant differences are found between the contiguous and non-contiguous epigenetic masks, as both variants are able to find success on a range of different problems. Non-contiguous masks are more effective in complex problems such as the diversity-hard, feasibility-hard, and convergence-hard cases from the DASCMOP test set. In contrast, contiguous masks achieve better results in convex and linear problems, suggesting that the effectiveness of each epigenetic approach is dependent on how the problem is structured and the relation between decision variables, and further suggests that the configuration of using a contiguous or non-contiguous mask could be considered hyperparameter of the mechanism. If the order of variables were randomised within a benchmark problem, the advantages of the contiguous mask may be lost, so the design of decision variable sequence plays a role in the effectiveness of each epigenetic mask variant.

Overall, the results highlight the potential of epigenetic mechanisms to Evolutionary Algorithms. However, the epigenetic approaches might require additional fine-tuning for the best results. Developing more adaptive approaches to mask selection and hyperparameter tuning could help improve the benefits across a wider range of problems, on both early-stage convergence and better sustained long-term performance.

The epigenetic adaptations lead to faster convergence properties akin to those seen in biological mathematical models of epigenetic effects [87]. In dynamic changing environments, epigenetics can aid organisms to achieve faster adaptations with high convergence. The adaptations are carried into future generations, and can be lost or reversed if the environment returns to being stable [55]. Computationally with static multi-objective benchmark problems, the faster convergence properties are noticeable, but the benefits are not fully realised in a non-changing environment. Next, the epigenetic mechanism is benchmarked against dynamic multi-objective problems, where more improvements to performance can be expected compared to the static problems

4.5 Performance on dynamic multi-objective optimisation problems

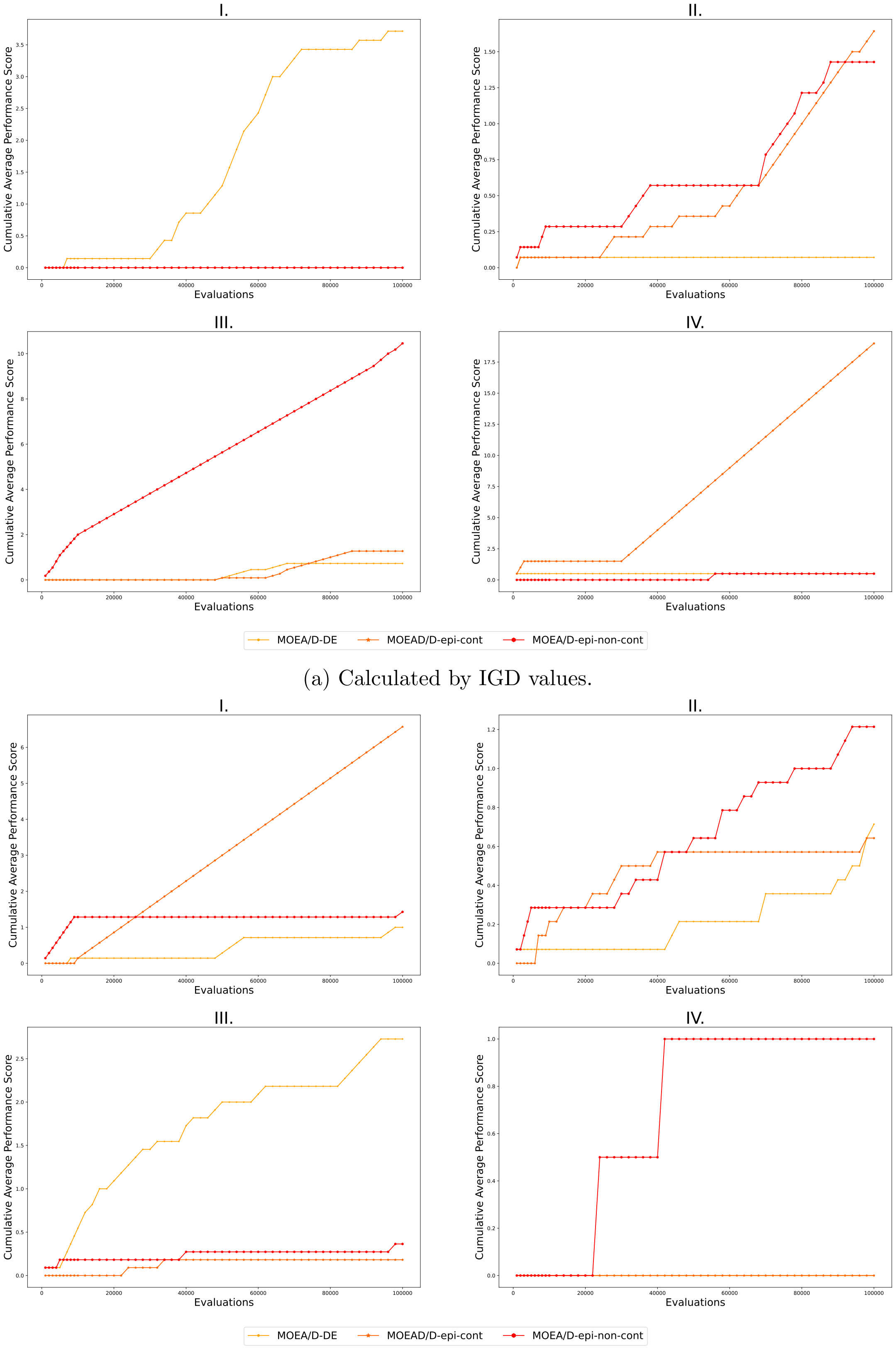
Next the dynamic benchmark problems are tested, with problems that have changing Pareto fronts, Pareto sets or both at the same time. Figures 4.12 and 4.13 show the cumulative average performance score of MOEA/D-DE and D-NSGA-II. Unlike the static benchmark problems where there was a clear distinction between the baseline algorithms and the algorithms with epigenetics, the performance differences for the dynamic problems are mixed and there are interesting differences between the performance scores calculated by HV and IGD metrics. The cumulated sum of average performance scores are presented as the plots of the average performance scores are difficult to interpret with frequent fluctuations between the tested algorithms and epigenetic variants.

The cumulative average performance score for an algorithm A_i is calculated as

$$\sum_{t=1}^{T} APS(A_i) \tag{4.3}$$

where T is the current evaluation counter and m is the number of problems. The evaluation counter represents the recorded performance values, which occurs every 2000 evaluations. The maximum cumulative average performance score is therefore the evaluation counter multiplied by the number of other algorithms. However, it is important to note achieving a maximum cumulative performance score indicates that the algorithm outperforms all other algorithms on all problems at every evaluation counter. In the dynamic problems explored here, the maximum cumulative average performance score is $50 \cdot 2 = 100$ as there are 50 evaluation points at which performance metrics are recorded, and two other algorithms to compare against.

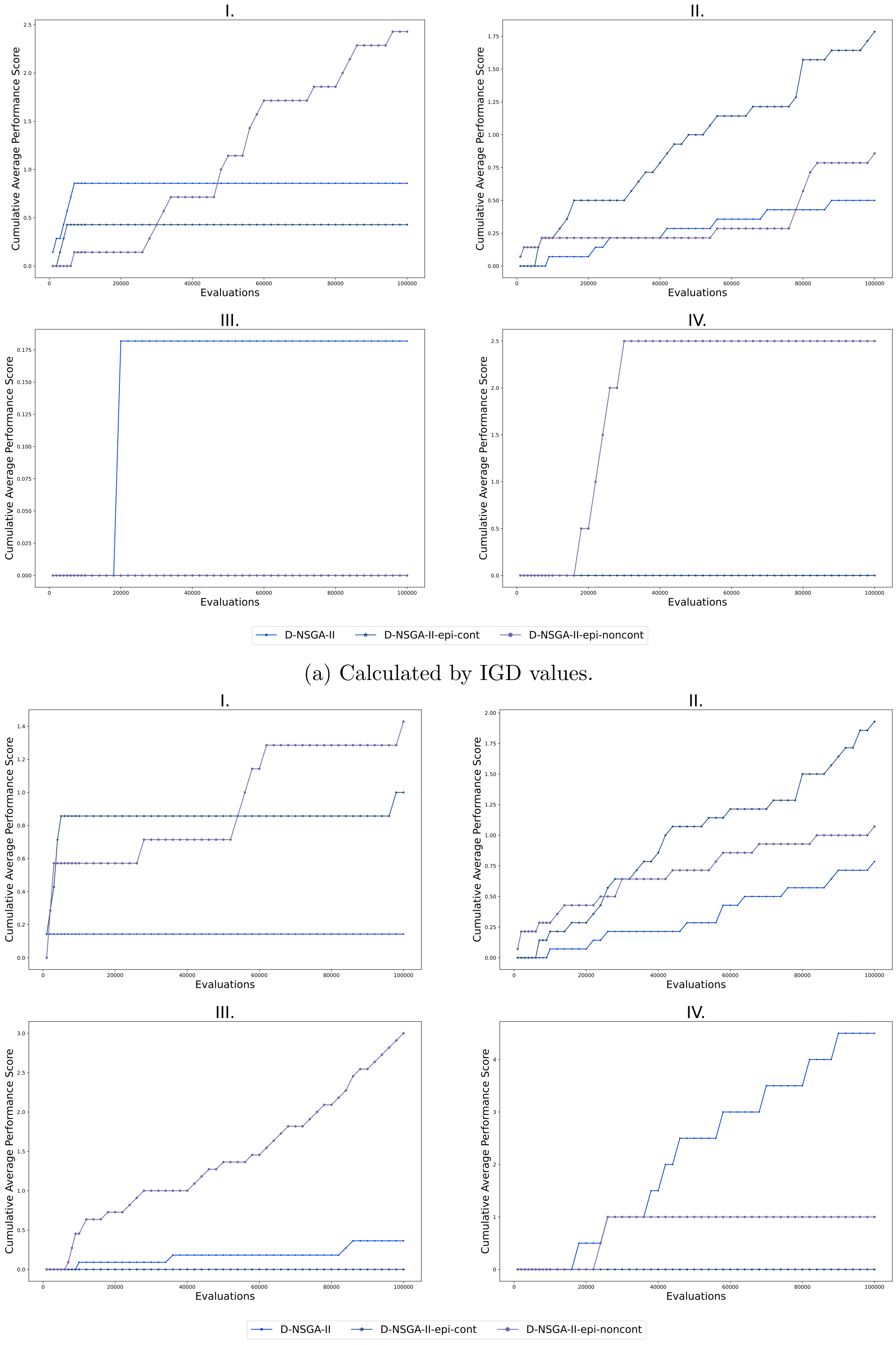
Notably, the epigenetic mechanism stands out in three cases: type I dynamic problems for MOEA/D-DE on IGD values, type III dynamic problems for MOEA/D-DE on HV values, and type IV dynamic problems for D-NSGA-II on HV values. In these cases, there is some part of the problem that remains stable across environmental changes, whether it is because only one of the Pareto set or front changes, or only the constraints that change in type IV problems, which allows the epigenetic mask to block useful variables. For example, in type I problems, the Pareto front remains static while only the Pareto set shifts. The epigenetic mask is able to help retain promising decision variables and enable the algorithm to adapt rapidly to a shifted Pareto set. It should be noted that, the performance difference



(a) Calculated by IGD values.

(b) Calculated with HV values.

Figure 4.12: Cumulative average performance score for MOEA/D-DE.



(a) Calculated by IGD values.

(b) Calculated with HV values.

Figure 4.13: Cumulative average performance score for D-NSGA-II.

is very small compared to the maximum possible cumulative average performance score, indicating very few problems where one approach outperforms another. However, type IV problems with MOEA/D-DE with IGD values has the contiguous epigenetic mask at 19.0 cumulative average performance score at the end of the optimisation, showing its poor performance compared to the non-contiguous mask and baseline MOEA/D-DE.

The type II of dynamic problems where both the Pareto set and Pareto front changes over time shows mixed average performance score for all algorithms and performance metrics, with the epigenetic mechanism occasionally showing poor performance and small success, albeit with a small cumulative average performance score of 1 – 2 compared to the maximum of 100, which indicates the variants performing on average 1 – 2% worse than the baseline approach. Here, the shifts of both the Pareto set and front mean that previously effective decision variables that were blocked in the epigenetic mask could now contain misleading or out of date information, negatively affecting the adaptation to new environments once both the optimal set and front have moved. For example, a contiguous epigenetic mask on MOEA/D-DE when looking at IGD values has good early performance up to 30,000 function evaluations, but as the problem shifts more substantially, outdated decision variables spread through epigenetics degrades the performance over time, with the baseline MOEA/D-DE algorithm maintaining lower average performance score after 45,000 evaluations. For D-NSGA-II, only the type IV of constrained dynamic problems shows strong HV performance for epigenetics. In many other dynamic problem types, one epigenetic variant performs significantly worse while the other variant only shows small differences to the baseline D-NSGA-II.

Overall, these findings suggest that the effectiveness of epigenetic mechanisms in dynamic multi-objective optimisation is dependent on environmental stability. The mechanism performs best when previously learned decision variables within the epigenetic mask remain relevant, but becomes an issue when rapid disruptive changes cause the historical information to become outdated. This highlights the need for methods to adaptive the masking dynamically to better handle rapidly changing dynamic problems. As the performance across all four dynamic problem types is mixed between the baseline algorithms and the epigenetic variants, and the performance differences are small compared to the maximum possible cumulative average performance score, the best and worst problems for epigenetics will be discussed to highlight the kinds of problems where the epigenetics mechanism performs well, and the problems where the mechanism performs poorly. There are also a number of problems where there is no statistical significant difference between the baseline algorithm and the epigenetic variants.

4.5.1 Epigenetic strengths

Figure 4.14 displays the problems where the epigenetic mechanism outperforms the baseline algorithms. Both algorithms have different problems where the epigenetic mechanism

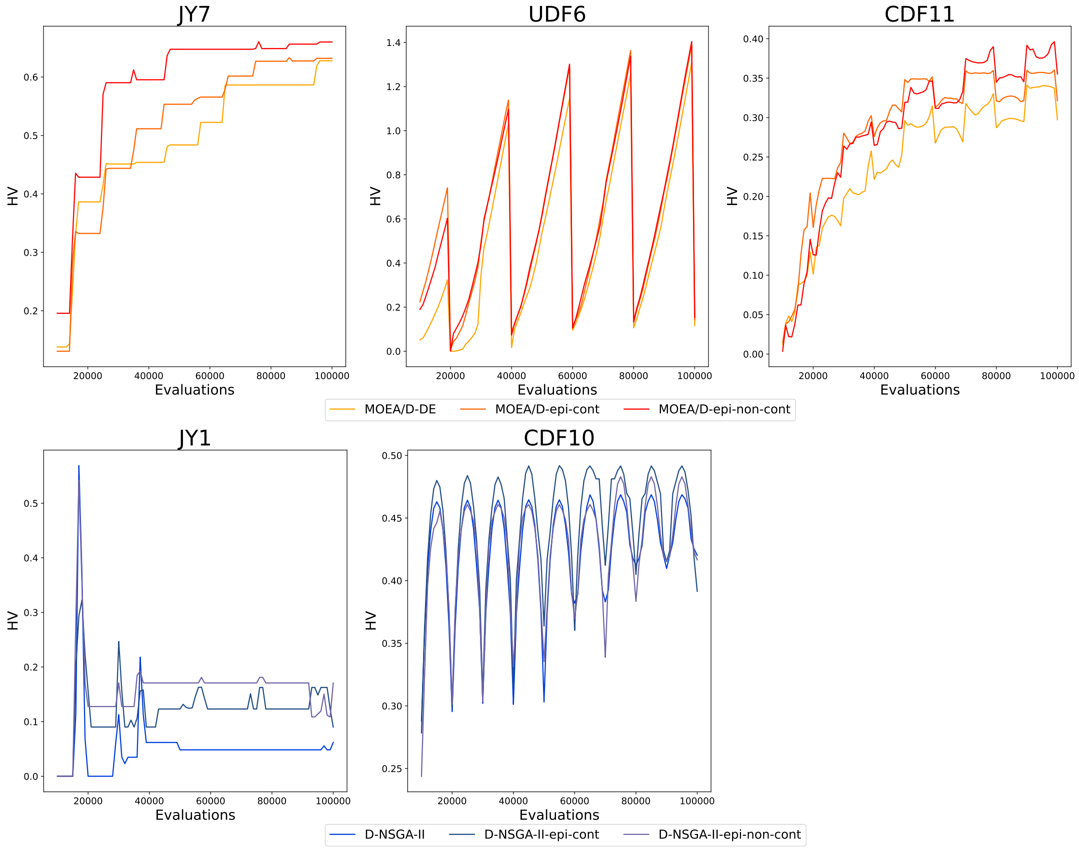


Figure 4.14: Median HV values on dynamic problems where the epigenetic mechanism performs well.

performs best. For MOEA/D-DE, JY7 shows a clear difference between the epigenetic variants and the baseline algorithm, especially after 50,000 function evaluations. UDF6 and CDF11 show interesting patterns of performance with the dynamic shifts visible on the with the HV values over time. In UDF6, the baseline MOEA/D-DE algorithm performance is lower at each dynamic peak, with the largest difference seen at the beginning at 20,000 function evaluations. At the later peaks at 40,000, 60,000, 80,000, and 100,000 evaluations, the performances are much closer though a gap between the epigenetic variants and the baseline can be seen. The same pattern is seen for CDF11, where the dynamic shifts are only visible after 50,000 function evaluations.

In the two problems where D-NSGA-II with epigenetic performs well, JY1 is a continuous type I problem where only the Pareto set changes, and CDF10 is a continuous type III problem where only the Pareto front changes. There is not an obvious or clear pattern for D-NSGA-II performing better specific types of dynamic problems, but some performance similarities to MOEA/D-DE with epigenetics can be seen. In JY1, there is a degradation in performance after 20,000 function evaluations and the epigenetic approach is able to maintain better final solutions compared to the original baseline D-NSGA-II. For CDF10, higher peaks for the contiguous epigenetic mask are seen at each dynamic cycle. Furthermore, higher peak HV values are seen for the non-contiguous mask on the last three cycles, indicating the effect of inherited epigenetic masks in the non-contiguous approach which was showing the same performance as the baseline algorithm until that point.

4.5.2 Epigenetic weaknesses

Figure 4.15 displays the problems where the baseline algorithms outperform the epigenetic mechanism. All three problems where MOEA/D-DE with epigenetic performs worse than the baseline algorithm are type III problems where only the Pareto front changes. In all three problems there is a clear distinction between the baseline algorithm and the epigenetic variants through most if not all of the 100,000 function evaluations. It should be noted the difference in HV value is low in some cases, for example there is a difference of 0.0052 between the baseline MOEA/D-DE and the epigenetic mechanism with a contiguous mask for JY5, and a difference of 0.0159 between the baseline and the epigenetic mechanism with a non-contiguous mask. The low difference in HV values contrasts some of the differences seen when the epigenetic mechanism performs better as described earlier, for example with a 0.070 and 0.164 difference in HV value for JY7 at 50,000 evaluations for a contiguous and non-contiguous mask respectively, and a difference of 0.418 and 0.280 difference for contiguous and non-contiguous epigenetic masks respectively in HV value for UDF6 at 20,000 evaluations. Regardless, in all three problems, the baseline algorithm is able to outperform the epigenetic variants. The poor performance of the epigenetic mechanism on type III problems is supported by Figure 4.12 where the baseline MOEA/D-DE has a consistently lower average performance score compared to the epigenetic variants. Problems with only a change in the Pareto front would be analogous to environmental changes in the biological literature, as the objectives, or fitness, of the current environment changes. The poor performance of epigenetics on this set of problems is therefore surprising, suggesting a possible mismatch in the hyperparameters or the process in which the epigenetic masks are being activated or inherited. As the epigenetic mechanism has shown to increase convergence for a range of static problems, there may be more details or processes required to apply the mechanism well to dynamic problems.

For D-NSGA-II, the epigenetic variants performed most poorly on CDF2. There is a degradation in performance over time, similar to JY1, for both the baseline algorithm and the epigenetic approaches, but the epigenetic approaches are unable to stabilise after 50,000 evaluations. The lack of stability compared to the baseline algorithm suggests the initial hyperparameters chosen may not be suitable in this particular case compared to the baseline algorithm, as other problems did not display the same issue and the baseline algorithm remains relatively stable. This could be caused by an overly aggressive blocking of the decision variables, or poor choices of epigenetic masks particularly in the contiguous case that leads to a lack of diversity. Therefore, specific tuning of the mechanism may be required for different problems rather than a single configuration for dynamic problems.

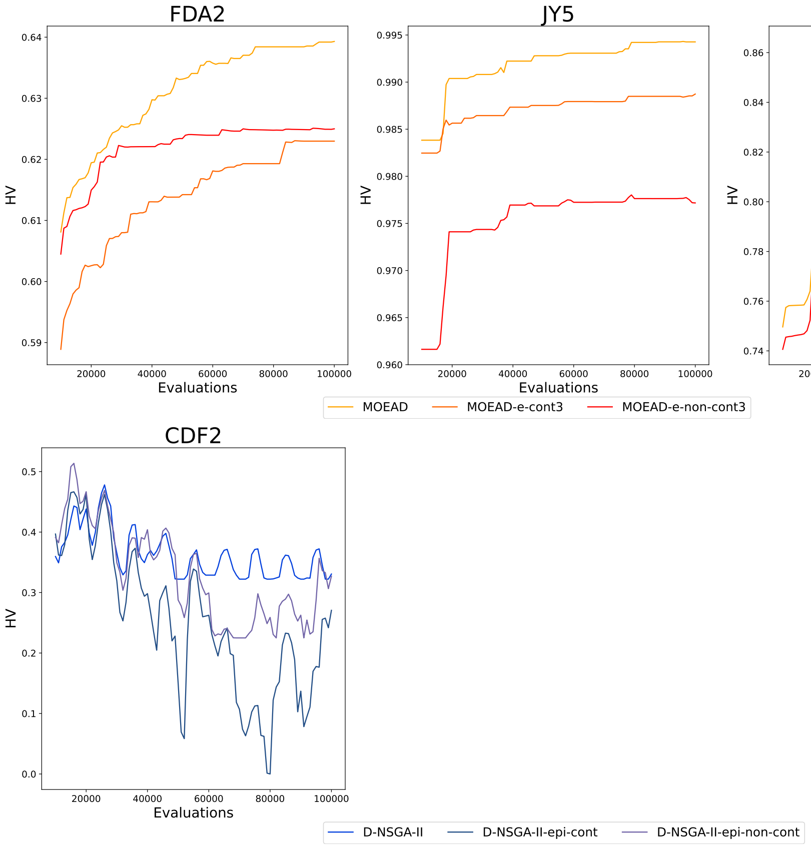


Figure 4.15: Median HV values on dynamic problems where the epigenetic mechanism performs poorly. Evaluations begin at 10,000 to highlight the performance difference.

4.5.3 Epigenetic performance on dynamic problems

The results on dynamic multi-objective optimisation problems show mixed results compared to the performance on static problems. Unlike the clearer benefits seen in static problems, the performance differences in dynamic problems are more varied. The cumulative average performance scores of MOEA/D-DE and D-NSGA-II indicate that while epigenetics does provide advantages on certain problems, its effectiveness is highly dependent on the nature of the dynamic changes. In cases where the problem retains some elements of stability, such as a static Pareto front in type I problems or only changing constraint in type IV problems, the epigenetic blocking mechanism can preserve effective decision variables to enable rapid re-adaptation. However, when both the Pareto set and front shifts like in type II problems, the information carried forward from being blocked by the epigenetic mask can become out of date and worsen performance for dynamic optimisation. Furthermore, the performance results differ based on the metric used, with HV and IGD providing contrasting results, indicating a difference in the spread of solutions across the Pareto front.

Additionally, the results further demonstrate the difference between the two epigenetic mask approaches. In some scenarios, such as a contiguous mask on type III dynamic problems, the epigenetic mechanism is able to outperform the baseline algorithm and non-contiguous mask approach. However, the non-contiguous mask outperforms the contiguous mask on type I dynamic problems.

4.6 Summary of the epigenetic mechanism

After testing the newly introduced epigenetic blocking mechanism on both static and dynamic multi-objective optimisation problems, it is shown the mechanism improves performance on the static problems while providing limited benefits in performance for dynamic problems. For static problems where the number of function evaluations are limited, the epigenetic mechanism can improve performance due to its convergence properties, often leading to problems where both the baseline and epigenetic variants of the algorithm reach similar final solutions, but the epigenetic mechanism finds those solutions with fewer function evaluations. In dynamic problems, the performance is mixed and both the baseline algorithms and the epigenetic variants perform similarly across many problems, with only a few problems standing out with clear better or worse performance, indicating a limited effect of the epigenetic approach. The findings highlight a need for more adaptive epigenetic approaches and investigation into the hyperparameters of the mechanism. As the evolutionary biological research shows there are tangible benefits of epigenetic effects in changing environments [87] based on their adaptive capabilities, the next step is to study the details necessary to improve the mechanism further, by investigating aspects such as dynamically changing and selecting the probability, mask size, and masking regions.

Chapter 5

Variants on the epigenetic blocking mechanism

This chapter explores further alterations to the epigenetic blocking mechanism introduced previously. The initial mechanism uses a constant probability, like the probability of crossover or mutation, to trigger epigenetic blocking. The length of the epigenetic mask is a constant value based on the number of variables of the problem, and the decision variables selected in the mask are randomly chosen. These mechanisms are further explored through:

- variants that alter the probability and mask length dynamically
- investigating the use and effectiveness of variable gradients to determine which decision variables to block
- utilising hyperparameter optimisation to tune the probability and mask length according to the problem

The different investigations are used to determine the most effective methods to enhance performance of the epigenetic mechanism, with a focus on dynamic multi-objective optimisation problems. As epigenetics is dynamic in nature, the epigenetic mechanism should also contain dynamic changes to match the biological inspiration, and to match the nature of dynamic optimisation problems. Furthermore, the exact probabilities and mask lengths are difficult to determine manually, and may have strong effects on different problems and algorithms.

It is not clear which decision variables should be included in the epigenetic mask. Each problem will have a different subset of variables which have a larger impact on the final objectives and not all variables may contribute equally. Due to the inheritance of the epigenetic mask and the selection pressure of an evolutionary algorithm, the epigenetic mechanism should trend towards choosing the more impactful variables over time, as epigenetic masks with poor variable choices lead to lower fitness and are less likely to be inherited. However, perhaps there is an approach to bias the selection of decision variables in the epigenetic mask. The use of gradient values at each variable aims to study such an approach. By calculating partial derivatives and finding the gradient for every decision variable, the additional information could be used to enhance the epigenetic mechanism.

5.1 Experimental setup

The experimental setup for all the epigenetic variants tested in this Chapter will follow the same methodology as the dynamic problem experiments in the previous chapter, with the multi-objective dynamic problems sets of UDF [20], JY [96], CDF [73], and FDA [60] being tested, the details of which can be found in Appendix A. The results use 20 independent runs for each algorithm and problem with a total of 100,000 function evaluations. IGD and HV values are measured every 1,000 function evaluations. A n_t value of 10 and τ_t value of 5 is used as parameters for the dynamic problem changes. All solutions are stored in a separate unbounded non-dominated archive that is re-evaluated at each time step to ensure it remains up to date with dynamic Pareto set or Pareto front changes.

The average performance score is calculated to measure statistically significant differences at every 1,000 function evaluations by performing a Wilcoxon ranked-sum test with a p-value threshold of 0.05. To account for statistical type I errors, the Bonferroni correction is applied, adjusting the p-value threshold based on the number of pair-wise comparisons made. This provides the performance over the during of the optimisation. A lower average performance score for an algorithm or epigenetic variant indicates fewer other approaches outperform it and it has better performance. As there are two algorithms tested, MOEA/D-DE and D-NSGA-II, average performance scores of each are calculated separately so the performance differences are measured between the baseline algorithms and the epigenetic variants, and not measured between the two algorithms which could inflate the performance scores. The cumulative average performance score is used to show the performance difference over time. The maximum cumulative average performance score indicates that can algorithm outperforms another every 1,000 function evaluations. The hyperparameters used for MOEA/D-DE and D-NSGA-II are shown in Table 5.1.

5.2 Effects of changing mask length and probability

To determine how the changes in the block rate and the probability of change affect the performance, the two variants of the epigenetic mechanism are investigated. The difference between each variant, the original mechanism, and the baseline algorithm are then compared to demonstrate the improvement in performance.

Table 5.1: Hyperparameters for the baseline algorithms used. N is the population size. MR and CR are the mutation and crossover rates respectively. F is the differential weight for differential evolution. NS is the neighbourhood size and NP is the neighbourhood selection probability for MOEA/D-DE.

Algorithm	**Hyperparameters**
D-NSGA-II	$N = 128, MR = 1/d, CR = 0.9$
MOEA/D-DE	$N = 100, MR = 1/d, CR = 0.9, F = 0.5, NS = 3, NP = 0.9$

Table 5.2: Hyperparameters for the epigenetic variants

Algorithm	**Hyperparameters**
EIB	$ER = 0.1, s = \frac{evals}{max\_evals} \cdot d$
EIP	$ER = \frac{evals}{max\_evals} \cdot 0.8, s = \frac{d}{5}$

Two parameters are altered: the probability for the mechanism to trigger, and the number of variables blocked in the process (mask length). A summary of the variants is as follows:

- **EIB** - with a constant probability of 0.1 and a varying mask length from 1 up to the total number of variables in the problem.
- **EIP** - with a varying probability from 0 up to a maximum of 0.8 with a constant mask length of the number of variables divided by 5. A maximum epigenetic rate of 0.8 is used to prevent the mechanism from triggering too often.

Both the contiguous and non-contiguous masks are used.

The gradual increase in either probability or mask length is intended to increase the convergence of the population, as epigenetic blocking more prevents diverse changes. The maximum probability is limited to 0.8 to prevent stagnation where the blocking occurs too often. Increases to probability are rounded to 0.01, increasing every 2 generations to the maximum probability. The mask length increases are rounded to the nearest whole number and depends on the number of variables. For example, a problem with 30 variables and a population of 500 will increase the mask length every 4 generations. Continuing to increase the probability and mask length in a second dynamic cycle of change in each problem exaggerates the difference that can be achieved with a more aggressive epigenetic mechanism. It is expected to see degraded performance towards the end of the optimisation when the probability and mask sizes are large, as many decision variables will be continually blocked during the reproduction process. However, the effect of changing each parameter can be analysed in more detail.

Details of these hyperparameters, can be found in Table 5.2. ER is the epigenetic rate, or the probability for the epigenetic mechanism to trigger and s is the mask length. The Inverted Generational Distance (IGD) [165] and HyperVolume (HV) [221] metrics are used to show the performance of the epigenetic variants to the MOEA/D-DE algorithm. The average performance score is used to compare performance between each variant. If any variant performs statistically significantly better than another, the variants it outperforms increment their average performance score. A lower score therefore indicates an algorithm that is not outperformed by any other algorithm. The original variants **E** and baseline algorithms without epigenetics are included in the average performance score calculations to compare every variant with each other and the baseline.

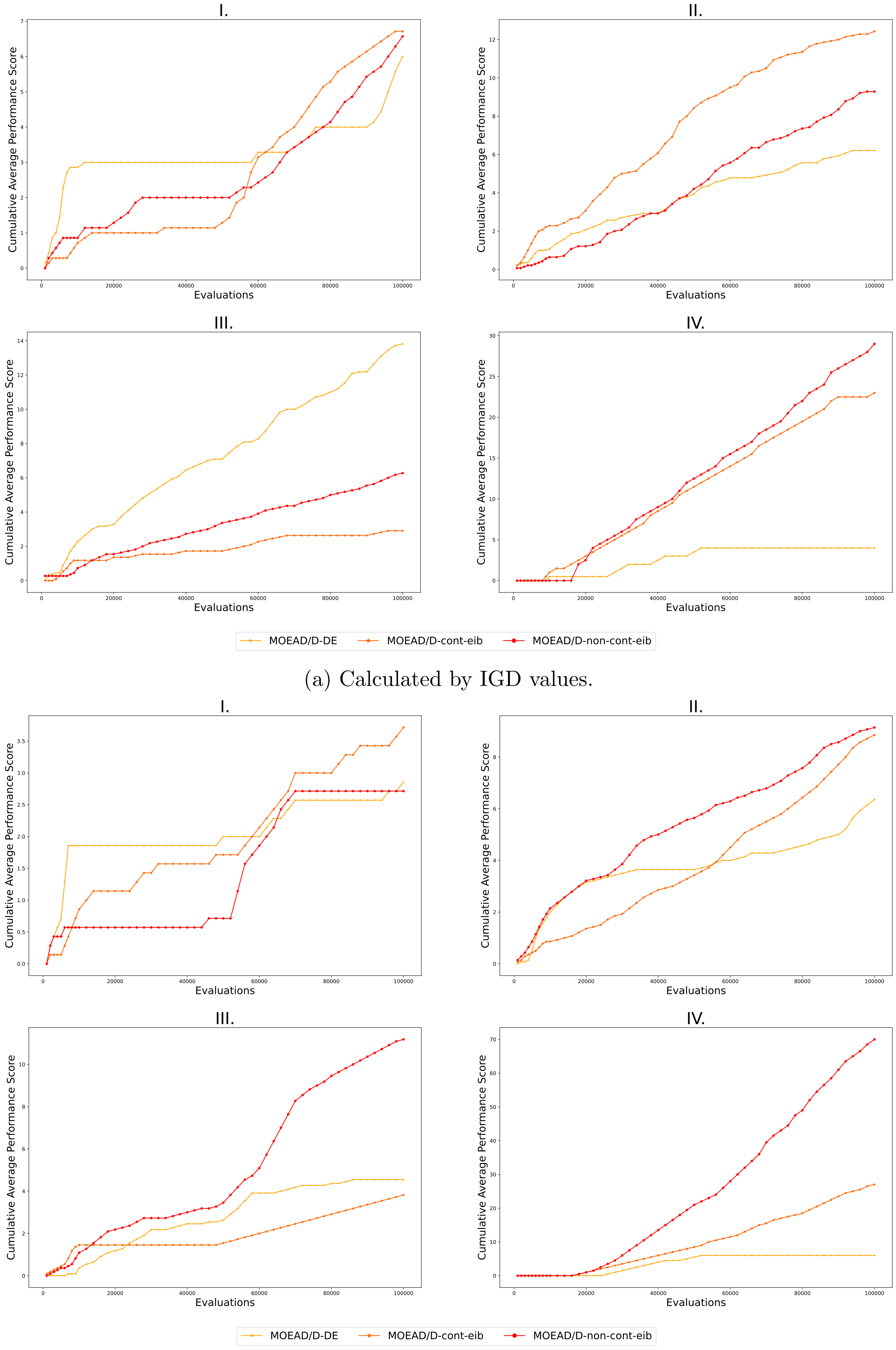
The maximum cumulative average performance score is 100, as each variant has a contiguous and non-contiguous configuration, and are being compared to the baseline independently from the other variants, making the number of other algorithms in the comparison 2.

5.2.1 Performance of EIB against the baseline

First, the average performance scores are presented to show the performance of the new epigenetic approaches to the baseline algorithm. The cumulative average performance scores are shown for clarity. The HV and IGD values are shown as differences in performance with the two metrics are seen, and indicate each variants differences in convergence and diversity for each dynamic problem type. There are a number of interesting observations that can be seen for each algorithm, and there are visible performance differences between the two algorithms in some categories.

For the **EIB** variant where the size of the epigenetic mask increases over time, MOEA/D-DE shows strong early performance on type I dynamic problems, mixed performance for type II and III problems, and poor performance for type IV problems, as shown in Figure 5.1. Figure 5.2 displays the D-NSGA-II results, which show generally poor performance for epigenetics except for type III problems. Since **EIB** uses a constant probability with a gradually growing block size, as the optimisation continues, progressively more decision variables will be blocked by the epigenetic mask, shifting the search from early exploration to faster convergence.

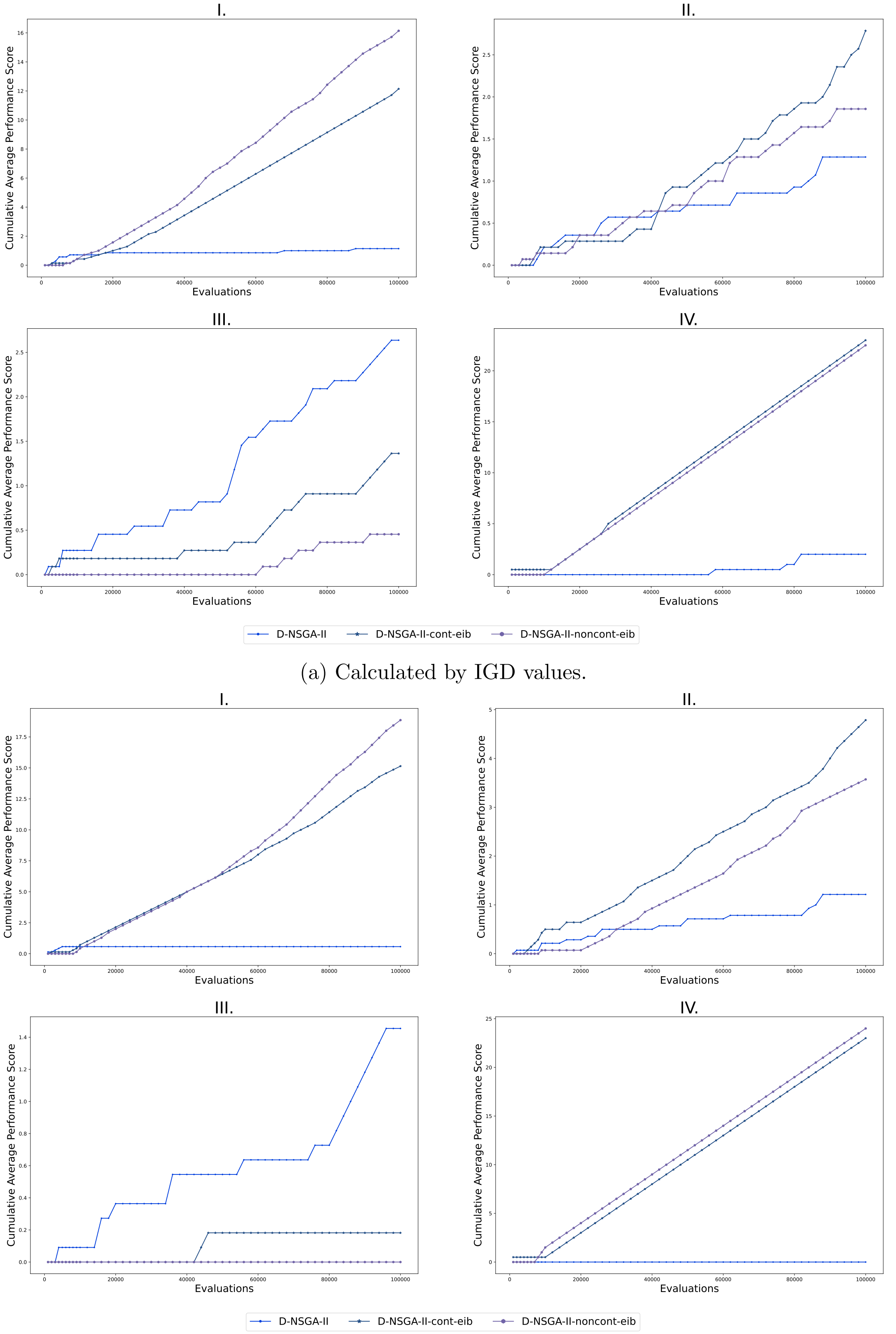
On type I problems, both IGD and HV metrics show similar results on both algorithms. MOEA/D-DE with epigenetics performs better than the baseline at the beginning of the optimisation, up to 50,000 evaluations when the performance starts to degrade. Due to **EIB**'s growing size of the epigenetic mask, more decision variables are blocked at later stages of the optimisation. There is a larger difference average performance score for IGD compared to HV, indicating the convergence properties of the epigenetic mechanism and potential lack of diversity, a pattern that will be seen again later. IGD tends to reward a close distance to the true Pareto front, while HV, being Pareto-compliant, is



(a) Calculated by IGD values.

(b) Calculated with HV values.

Figure 5.1: Cumulative average performance score for MOEA/D-DE against the **EIB** epigenetic variant.



(a) Calculated by IGD values.

(b) Calculated with HV values.

Figure 5.2: Cumulative average performance score for D-NSGA-II against the **EIB** epigenetic variant.

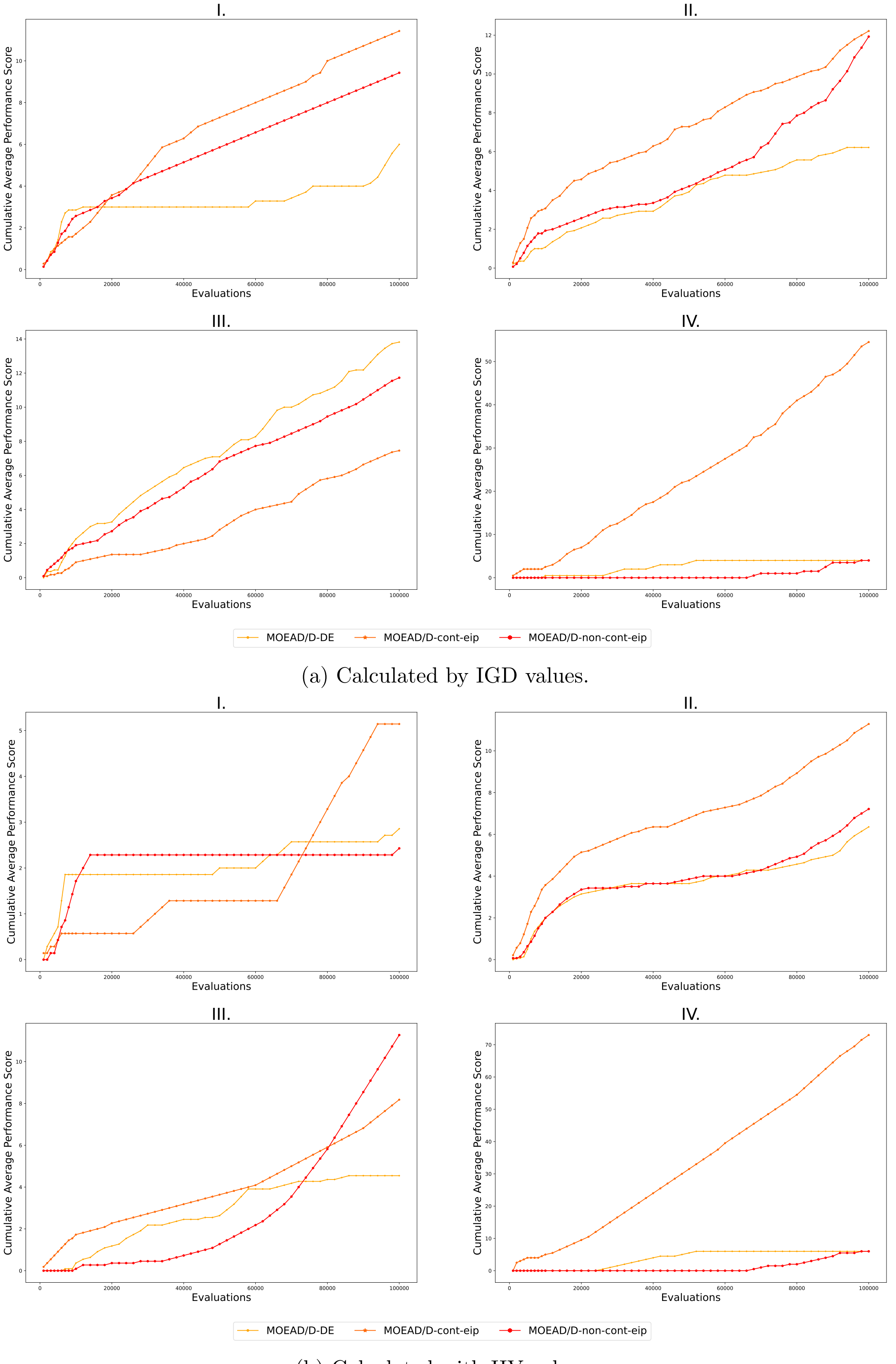
more sensitive to the spread of solutions across the front. The pattern of better IGD but worse HV therefore indicates that **EIB** promotes convergence but reduces spread as the block size increases.

The strong early performance for **EIB** is seen for D-NSGA-II at a smaller scale for only the first 10,000 evaluations. Poor performance is seen for both algorithms with **EIB** on type II dynamic problems, though some strong early performance is seen once again. Different results are found between the use of IGD and HV metrics here. As the IGD measures the Euclidean distance between solutions found and the true Pareto front, good IGD values indicates strong convergence properties in the algorithm. Conversely, HV is Pareto compliant and therefore takes into account the diversity and spread of solution across the Pareto front. The use of a contiguous mask for **EIB** with MOEA/D-DE shows good diversity across the Pareto front on type II problems, with lower average performance scores on HV, while the non-contiguous mask shows better convergence as seen on the IGD calculated average performance scores. In both cases, the epigenetic variant can out-perform the baseline algorithm in the first 50,000 function evaluations before the performance worsens. The timing of this performance differences suggests that as the block size grows too large, the epigenetic mechanism becomes a detriment to the optimisation. This means an adaptive means to adaptive the block size, or resetting the block size could preserve the early performance gains while avoiding early convergence.

5.2.2 Performance of EIP against the baseline

In the first type of dynamic problems where only the Pareto set changes, there is a significant difference for D-NSGA-II between the baseline algorithm and both **EIB** and **EIP** variants. Comparatively, there is a much smaller difference for MOEA/D-DE in the same type of problems. Both **EIB** and **EIP** for D-NSGA-II perform poorly compared to the baseline, with both contiguous and non-contiguous masks performing poorly and the non-contiguous mask performing the worse. This contrasts many other categories and variants where mixed performance is seen. Similar to the static problems explored in Chapter 4, the difference between MOEA/D-DE and D-NSGA-II comes from the latter's use of non-dominated sorting and crowding distance, which prioritise diversity. **EIB** and **EIP**'s increasing block size and increasing trigger probability, aim to retain fixed decision variables may no longer align with the new Pareto set, hence misguiding the algorithm. As MOEA/D-DE decomposes the search into sub-problems, its neighbourhood replacement makes the epigenetic bias less disruptive when only the Pareto set shifts.

Interestingly, there is a small window in the first 5,000 – 10,000 function evaluations where the epigenetic variants perform better than the baseline on both **EIB** and **EIP**, indicating the initial hyperparameter settings of the epigenetic mechanism for D-NSGA-II were already suitable, and the increased probability and mask lengths impacted performance even from early on after 10,000 function evaluations. A similar pattern is seen in



(a) Calculated by IGD values.

(b) Calculated with HV values.

Figure 5.3: Cumulative average performance score for MOEA/D-DE against the **EIP** epigenetic variant.

the second and fourth categories for D-NSGA-II.

There is noise and mixed performances in the second type of problem where both the Pareto front and Pareto set changes, but there is a slow increase in average performance score in later function evaluations for both **EIB** and **EIP**. The fourth type of constrained problems for D-NSGA-II further shows the poor performance of the epigenetic mechanism with both a high probabilities and mask lengths. The same performance is achieved as the baseline algorithm up to 5,000 – 7,000 function evaluations, after which neither epigenetic variant performed well.

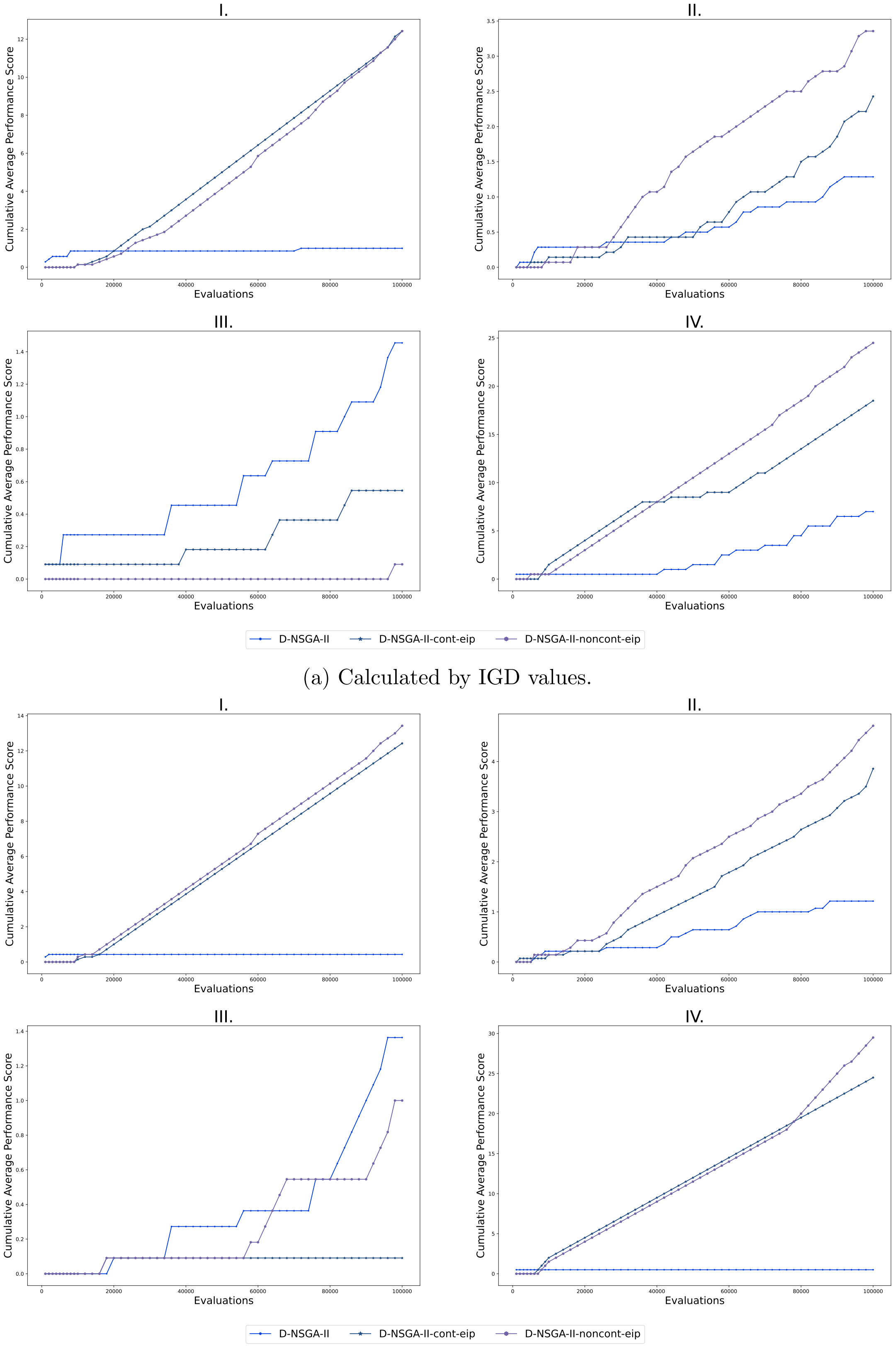
Finally, there are mixed performances for both **EIB** and **EIP** on the third type problems, where only the Pareto front changes. The performance on the epigenetic variants differs on MOEAD/D-DE compared to D-NSGA-II. Firstly, there is not a significant gap between the the baseline and the epigenetic variants, with the average performance low and consistent throughout the optimisation. **EIP** showed some worsening performance from 65,000 – 95,000 function evaluations but the average performance score is still low at 0.3. Furthermore there is a spike for the baseline MOEA/D-DE at 6,000 and 7,000 function evaluations but there is no indicative pattern of behaviour as the baseline has low performance score otherwise throughout.

The **EIP** variation on the epigenetic mechanism increases the probability of triggering the epigenetic mechanism over time. This mechanism favours frequent epigenetic blocking at later stages, which promotes convergence when suitable blocks of variables exist but can cause an over-biasing to a local optima when the Pareto front changes. While some differences between IGD and HV values are shown for MOEA/D-DE, the results for D-NSGA-II on both metrics are similar. Figure 5.3 and 5.4 display the cumulative average performance scores of MOEA/D-DE and D-NSGA-II respectively with **EIP**.

5.2.3 Conclusions of changing mask length and probability

The epigenetic approaches, **EIB** and **EIP**, offer interesting insights in the performance of the epigenetic blocking mechanism, but their performance varies significantly with problem types and algorithms. **EIB**, with its increasing epigenetic mask lengths, demonstrates strong early performances on type I dynamic problems for MOEA/D-DE, indicating that smaller masks can initially increase convergence. However, this performance diminishes further into the optimisation, suggesting that the increasing mask lengths hinder at later stages of optimisation. On the other hand, D-NSGA-II with **EIB** struggles across problem types, except for type III, where the larger masks seem to improve performance.

The **EIP** variant, which increases the probability of triggering the epigenetic mechanism over time, shows a different pattern. In the first category of dynamic problems, the baseline D-NSGA-II outperforms **EIP**, possibly due to the higher probabilities leading to over-convergence, the higher initial average performance score of the baseline on the



(a) Calculated by IGD values.

(b) Calculated with HV values.

Figure 5.4: Cumulative average performance score for D-NSGA-II against the **EIP** epigenetic variant.

IGD metric supporting this possibility. Both algorithms and both **EIB** and **EIP** initially outperform the baseline for type I problems. The point at which the baseline overtakes the epigenetic variants depends on the algorithm and variant, but the trend implies that the increasing probabilities and mask lengths are beneficial for problems with changing Pareto sets, but require additional tuning to find better advantages.

These results highlight the partial success of the epigenetic mechanism, showing that it can be fine tuned for better balance between convergence, with larger masks and higher probabilities, and diversity, with smaller masks and lower probabilities. The results further suggest a focus on adaptive mechanisms to adjust mask lengths and triggering probabilities would be beneficial to more consistent performance.

5.3 Using gradients to determine epigenetic blocking locations

To study an adaptive mechanism of the length and locations for the epigenetic mask, additional problem-specific information could be based on the sensitivity of each decision variable to the final solution of the problem, following the ideas of gradient descent to find the derivatives of the objective function. The difference here is the partial derivatives of each variable are used instead, as each of their gradients are needed to inform the epigenetic mechanism on whether to block that variable or not. The effect of using gradient information of variables is not well known, though there are many algorithms using gradient descent methods in machine learning [137]. Gradient information on objectives can show whether the slope of the objective function is tending towards a maximum or minimum, but it is more difficult to understand what the gradient of each variable represents.

There are two key aspects to the use of gradient information that is of interest to the epigenetic mechanism. First, is there a difference between positive and negative gradients for epigenetic blocking? Does utilising only positive or negative gradients in decision variables correspond to the problems that maximise or minimise the objective? Secondly, how significant is the magnitude of the gradients? Variables with larger gradients may indicate their relative importance to the optimisation problem, making them better or worse candidates for the epigenetic mechanism.

5.3.1 Calculating the gradients

Figure 5.5 shows the steps of the process for working out the partial derivatives and calculating the gradients. The equations for each objective function of a problem are taken and the partial derivatives calculated with respect to each variable. At every evaluation step, the variable values are evaluated for every solution to find each variable's gradient.

Then, the values of the gradients are used to aid the epigenetic mechanism determine when to block out that specific variable. Note that the use of gradient information moves the optimisation from a black-box framework, where the details of the underlying problem are unknown, to a white-box one, by accessing and calculating the function derivatives. This approach only applies to optimisation problems with known function equations.

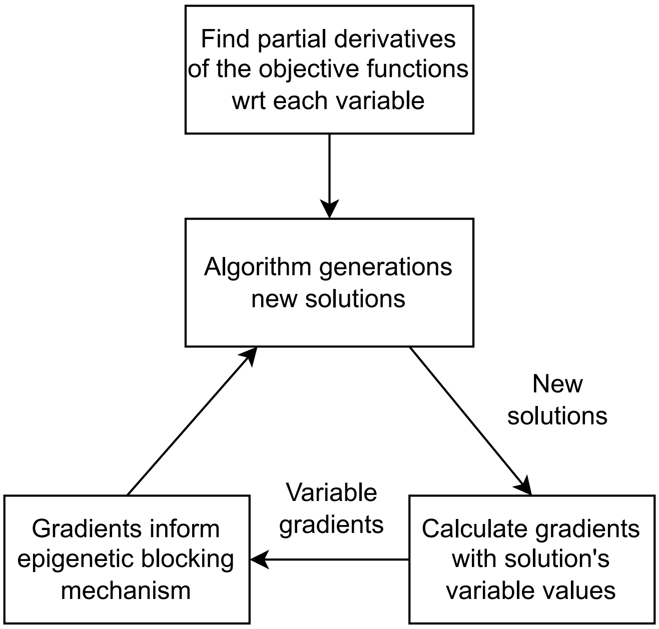

Find partial derivatives of the objective functions wrt each variable
Algorithm generations new solutions
New solutions
Calculate gradients with solution's variable values
Variable gradients
Gradients inform epigenetic blocking mechanism

Figure 5.5: How the gradient calculation fits into the algorithm with epigenetics.

For example in the JY1 problem, the equation for the first objective is as follows,

$$f_0(x) = (1 + \sum_{x=1}^{n}(x_i - Gt)^2) \cdot (x_0 + At \cdot sin(Wt \cdot \pi \cdot x_0)) \tag{5.1}$$

where Gt, At, and Wt are constants, some of which change over time depending on the problem. For JY1, only Gt changes over time. In this example, the derivative wrt x_0 would be

$$(1 + \sum_{x=1}^{n}(x_i - Gt)) + \pi \cdot At \cdot Wt \cdot cos(\pi \cdot Wt \cdot x_0) \tag{5.2}$$

and the derivatives wrt x_i would be

$$x_0 \cdot (2x_i - 2Gt) \tag{5.3}$$

The values of x_0 and all of x_i are then calculated for every solution to find the gradients for each decision variable. The gradients are summed to form a single gradient value per decision variable, for a simple and computationally efficient method. The approach acts as a scalarisation of the objectives into a single directional value, which would prioritise the objective with the larger magnitude. However, this approach loses the nuance of opposing objectives, which can degrade the effectiveness of the approach, for example if two opposing derivatives have a similar magnitude, the summed gradient would show a low value indicating less importance of that decision variable.

Four variants are tested based on the value of the gradient at each variable:

- **GP1** - All variables with positive gradients are blocked.
- **GN1** - All variables with negative gradients are blocked.
- **PP1** - A constant mask length is used and weighted probabilities are calculated based on the value of the gradients is used to choose a variable. Larger values have a higher probability. For each decision variable x_i in the decision vector $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ with an associated gradient g_i, the probability p_i of x_i being chosen is

$$p_i = \frac{(g_i - g_{min} + 1)^2}{\sum_{i=1}^{n}(g_i - g_{min} + 1)^2} \tag{5.4}$$

- **PN1** - The same as **PP1** except inversed such that larger negative gradients have the higher probability and large positive values have lower probability.

The probability for the epigenetic mechanism to trigger still applies, with a constant probability of 0.1 used in all gradient-based experiments. Because of this, not every solution will have its variables blocked by the epigenetic mask, but the choice of which variables and the number of variables blocked are based on the calculated gradients. Therefore there are not any contiguous or non-contiguous variants either.

5.3.2 Results of gradient-based epigenetic masks

The performance of the gradient-based methods are tested against the baseline algorithms to determine the effects of choosing particular decision variables in the epigenetic mask. The average performance score is calculated with all variants so the differences between each variant is taken into account. First the overall performance averaged across all dynamic problems is presented, then the performance is broken down into each dynamic problem type to find the scenarios where gradient-based epigenetic masks are the most effective. Here, the maximum cumulative average performance score is 200, as there are 50 evaluation counters and 4 other approaches compared against.

Figure 5.6 shows the cumulative average performance score for MOEA/D-DE with all the gradient variants average across all the dynamic optimisation benchmark problems. In the overall results, MOEA/D-DE with gradient epigenetic variants performs well on both HV and IGD metrics. **GN1** and **GP1** show the best results with the lowest average

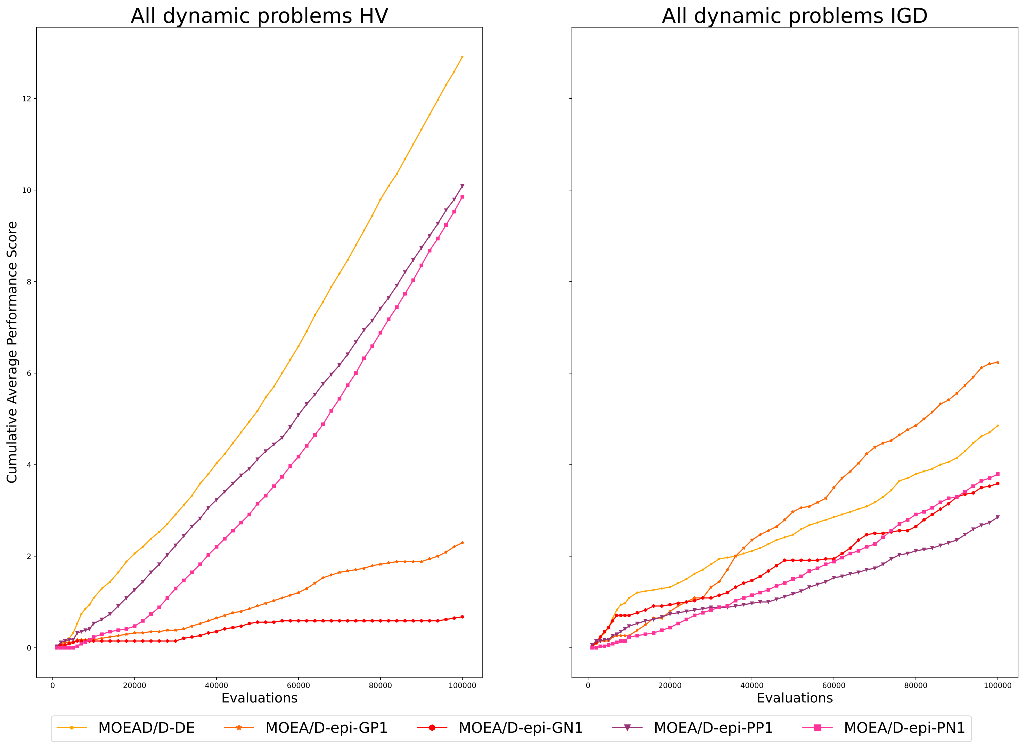


Figure 5.6: Cumulative average performance score for MOEA/D-DE against all gradient-based epigenetic variants for all dynamic problems.

performance scores based on HV, indicating the positive effect of blocking all decision variables with either a positive or negative gradient, with negative gradients having the best effect. Results from IGD metrics are less clear, with three gradient approaches performing similarly and **GP1** as the only approach that performs worse than the baseline. Note that the cumulative average performance score is quite low, at 6 out a possible 200, indicating there is little significance between the approaches, but as the score is averaged over all dynamic problems, a maximum score of 200 means that the approach outperformed all other approaches on every dynamic problem at every evaluation counter, which is highly unlikely.

For D-NSGA-II, poor performance of the epigenetic blocking mechanism is seen in general, across both metrics, as shown in Figure 5.7. This discrepancy is similar to the difference between U-NSGA-III and MOEA/D-DE on the static benchmark problems, where U-NSGA-III's ranking-based selection provides a stronger selection pressure that negates the effectiveness of epigenetic blocking, compared to the decomposition approach of MOEA/D-DE. Due to the poor performance, there is little to differentiate between the different gradient variants for D-NSGA-II. **GP1** has the highest cumulative average performance score, and **GN1** has the lowest, demonstrating the same trend as MOEA/D-DE where using negative gradients has a greater impact on performance.

Next, when grouping the results across the four dynamic problem types, the strengths and weaknesses of the gradient-based approaches can be further analysed. The results for

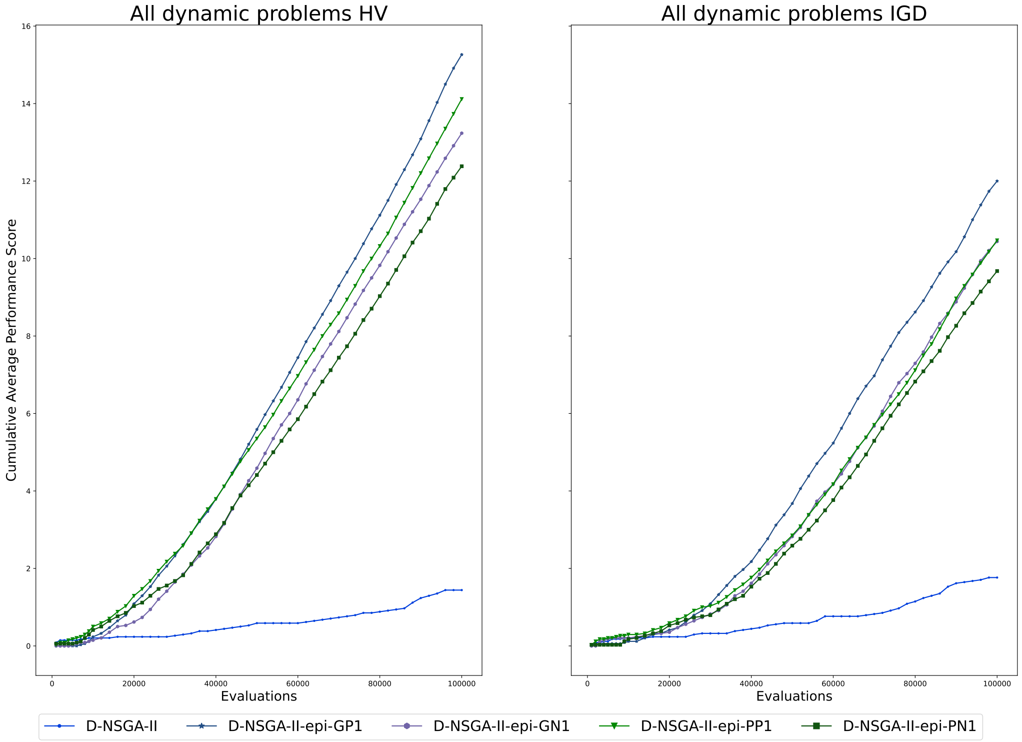
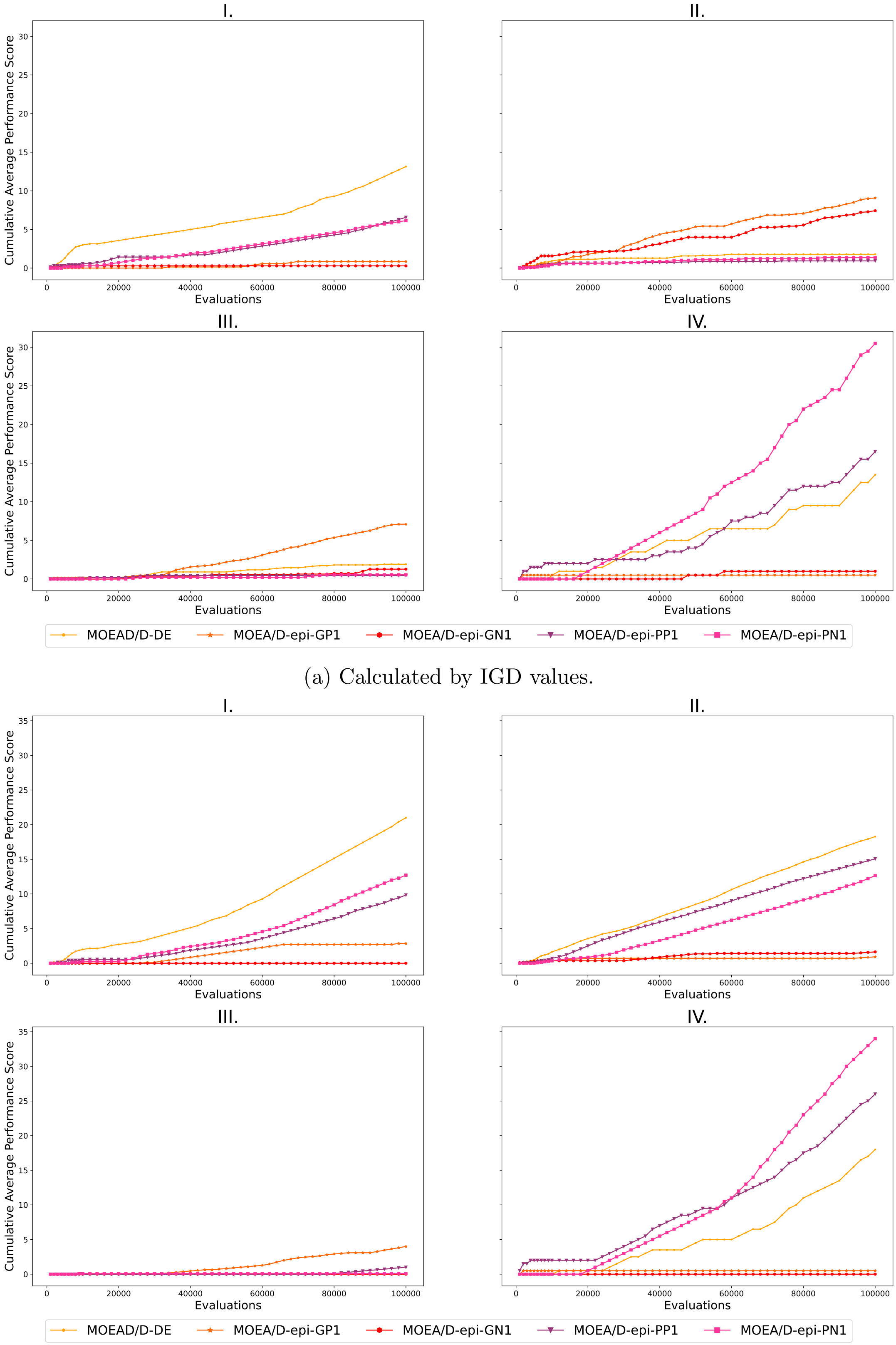


Figure 5.7: Cumulative average performance score for D-NSGA-II against all gradient-based epigenetic variants for all dynamic problems.

MOEA/D-DE are presented in Figure 5.8. All variants outperform the baseline for type I problems where only the Pareto set changes. Based on HV metrics, **GP1** and **GN1** outperform the baseline for type II and IV problems, with **GP1** performing the worse on type III problems and other variants performing similarly to the baseline.

However, when calculating based on IGD, none of the variants outperform the baseline for type II problems, and **GP1** and **GN1** now perform poorly compared to **PP1** and **PN1**. Table 5.3 shows the magnitude of final cumulative average performance scores for all gradient-based approaches with MOEA/D-DE. The low cumulative average performance scores demonstrates the effectiveness of the gradient-based epigenetic methods, for example **GN1** has a cumulative APS of 0.29 across all 100,000 function evaluations on type I problems, meaning it was almost never outperformed. In contrast, even though the baseline algorithm has the best results on the IGD metric for type II problems, its cumulative APS is the highest at 6.14 out of all the best performing results. The **GN1** variant shows the best consistent results, outperforming the baseline in all cases except on IGD for type II problems.

This pattern indicates that gradient-based blocking, especially when all positive or negative gradients are blocked, is much more effective than using a probability to block based on the gradient magnitude. This stems from the probabilistic method not consistently blocking important variables, occasionally blocking unimportant variables, and discrepancy from normalising the block probability, leading to many variables not blocked if there



(a) Calculated by IGD values.

(b) Calculated with HV values.

Figure 5.8: Cumulative average performance score for MOEA/D-DE against all gradient-based epigenetic variants, grouped by dynamic problem type.

Algorithm	**HV cumulative APS**				**IGD cumulative APS**			
	I.	II.	III.	IV.	I.	II.	III.	IV.
MOEA/D-DE	67.57	34.21	3.45	40.0	61.0	**6.14**	13.30	32.0
GP1	6.57	**5.86**	21.64	**1.0**	5.71	23.0	26.36	**1.50**
GN1	**0.29**	7.43	**1.27**	2.5	**2.29**	18.5	6.18	14.0
PP1	22.71	22.14	7.18	63.0	22.29	9.50	8.36	41.50
PN1	20.14	19.43	7.55	56.05	12.57	7.71	**3.36**	57.50

Table 5.3: Final cumulative average performance scores for MOEA/D-DE with epigenetic gradient-based approaches.

are one or two large gradient values, or most variables blocked if all gradient values are simiarly low. A reason why the positive and negative gradients both have similar effects, despite pointing in opposite directions comes from the summing of the objectives. Due to the summing approach, is objectives have similar values pointing in different directions, the nuance is lost. Whether the derivative value is small or large, both are treated equally in the **GP1** and **GN1** approaches. This dilutes the effect of decision variables with large gradient values, as small gradient values are treated the same, making both approaches act similarly to each other. In the context of minimisation-based dynamic multi-objective problems, which all the dynamic benchmark problems tested here are, positive gradients indicate decision variables where increases to the value would worsen the objective values, but the inverse is true that decreases to the value would improve the objective. In this case, blocking positive and negative gradients have a similar effect, but the increase or decrease of the variable value is not taken advantage of, as evolutionary algorithms do not explicitly increase and decrease variable values.

The performance measured by the HV and IGD scores are consistent for MOEA/D-DE except for type II problems. For HV, **GP1** and **GN1** perform the best across all the problem types, with **GP1**'s poor performance on type III problems reflected in both HV and IGD scores. On type II problems, **GP1** and **GN1** performs the best when measured with HV but perform the worst when measured by IGD. This indicates reduced convergence when both Pareto front and set change if the epigenetic mechanism blocks all positive or all negative gradients, but a retention in diversity and spread across the Pareto front.

In general, using all the positive or negative gradients in the epigenetic mask is more effective than a scaling probability of each decision variable being included based on its value, due to the inconsistency of blocking variables..

The strong results of epigenetics on MOEA/D-DE are not replicated for D-NSGA-II to the same extent. The gradient-based epigenetic variants perform poorly with D-NSGA-II except on type III problems. On type III problems, both the HV and IGD metrics show the epigenetic variants with strong early performance and maintaining the good performance for the rest of the optimisation except for **PN1**. Furthermore, D-NSGA-II using the gradient approaches finds very poor results for type I and IV problems, where

Algorithm	HV cumulative APS				IGD cumulative APS			
	I.	II.	III.	IV.	I.	II.	III.	IV.
D-NSGA-II	**2.0**	**8.21**	4.73	**9.0**	**3.71**	11.0	6.91	**15.0**
GP1	50.71	16.50	2.36	72.50	45.29	14.79	3.46	60.50
GN1	40.0	16.0	2.45	65.0	37.86	13.14	2.45	61.50
PP1	49.43	12.0	**1.91**	79.50	49.14	9.86	**1.45**	62.0
PN1	45.57	9.0	6.55	70.5	40.71	**8.43**	8.73	61.50

Table 5.4: Cumulative average performance scores for D-NSGA-II with epigenetic gradient-based approaches.

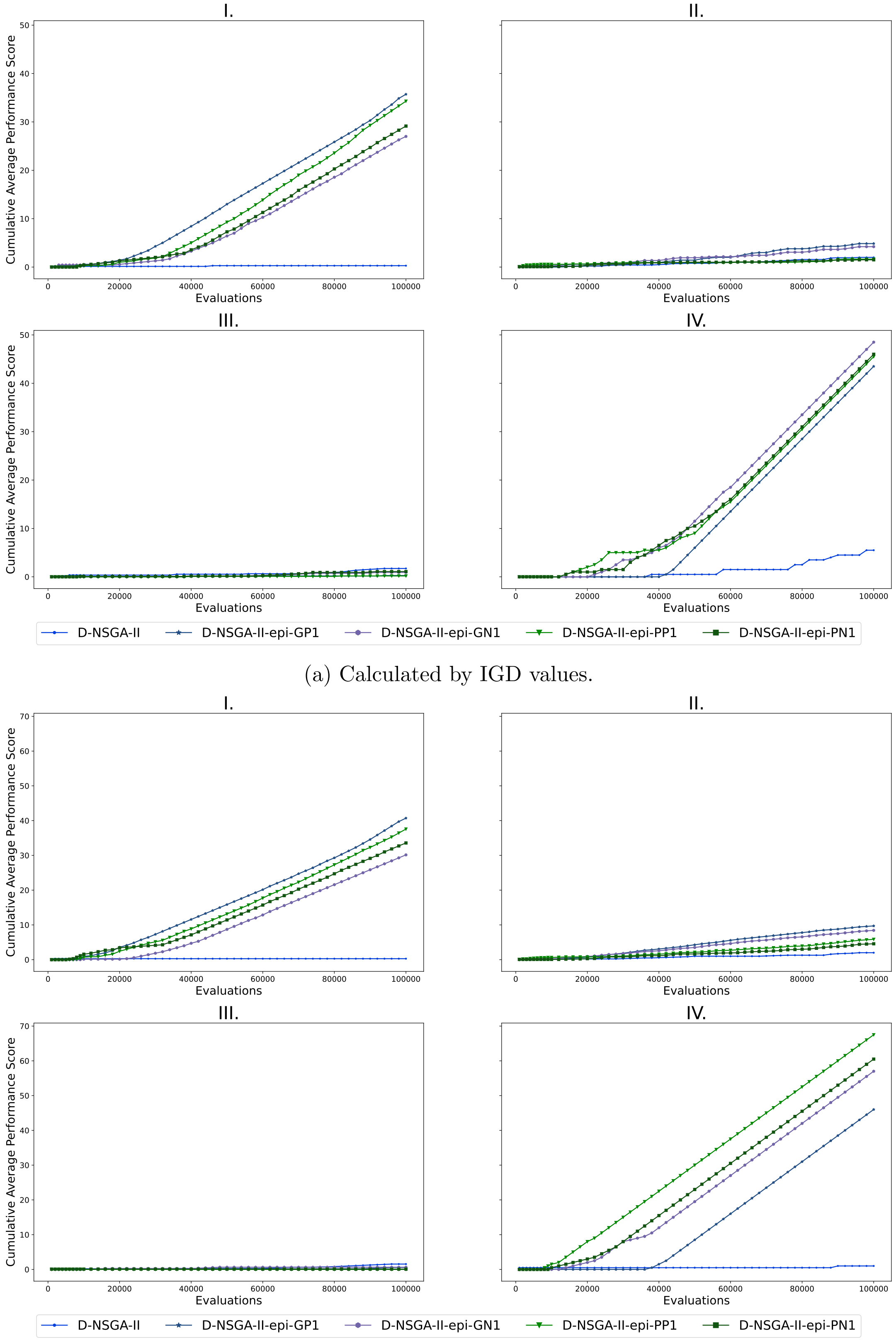
the cumulative APS scores are high relative to the baseline algorithm's scores, as shown in Table 5.4, such as a 2.0 cumulative score for baseline D-NSGA-II with cumulative scores of 40.0 and more for the gradient variants.

Similar to MOEA/D-DE, the performances calculated by the HV and IGD metrics are consistent for D-NSGA-II with the exception of type II problems. With performance based on IGD, **PP1** and **PN1** shows good performance and the baseline algorithm performance results between the two gradient-based approaches after 100,000 function evaluations. When using HV to calculate performance, the order of performance of the gradient-based approaches remains the same with **PN1** performing best and **GP1** performing the worst, but now the baseline algorithm outperforms all the gradient methods. This demonstrates the convergence properties of the epigenetic mechanism resulting in better IGD scores and worse HV scores.

5.3.3 Computation issues with the gradient method

The time taken and computation required to calculate gradients is significantly costly compared to other methods. Despite the derivative being pre-computed beforehand, the gradient of each variable has to be recalculated for every individual in every generation when the epigenetic mechanism is active. The computation costs quickly adds up, making this method relatively expensive for improving performance of the epigenetic mechanism, though with promising results for MOEA/D-DE.

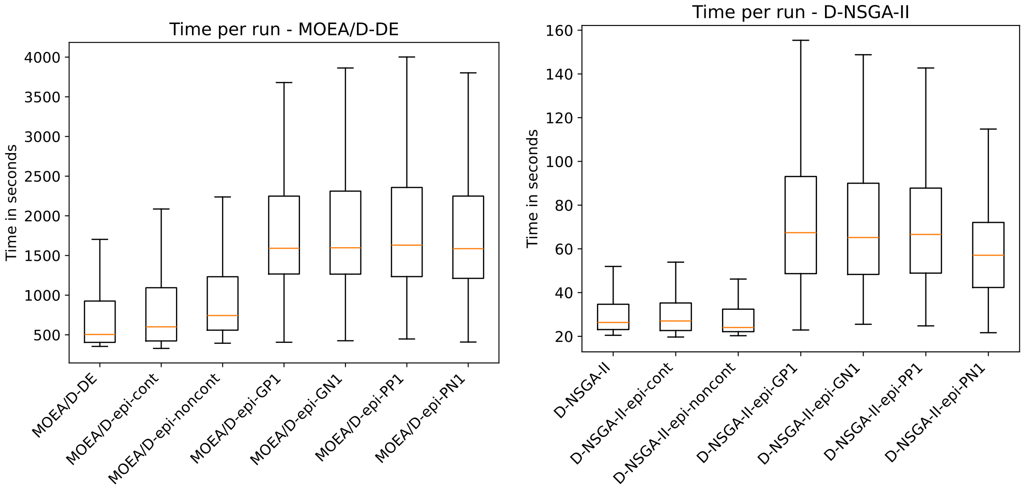
Although real time used in performing computation can have high variability due to hardware and other programs running at the same time, all computation is run on the IRIDIS 5 High Performance Computing System to maintain equal and fair hardware usage as much as possible. IRIDIS 5 uses Intel compute nodes with dual 2.0 GHz Intel Xeon Gold 6138 processors. Figure 5.10 shows the original baseline algorithms and original epigenetic variants taking similar amounts of time when averaged across all the runs and benchmark problems. There is a noticeable increase in average time taken for the gradient-based approaches at about three times the time required. For example, MOEA/D-DE and the contiguous and non-contiguous epigenetic variants have a median time taken of 504, 600, and 703 seconds, compared to the gradient-based variants which all have a median time



(a) Calculated by IGD values.

(b) Calculated with HV values.

Figure 5.9: Final cumulative average performance score for D-NSGA-II against all gradient-based epigenetic variants, grouped by dynamic problem type



(a) MOEA/D-DE

(b) D-NSGA-II

Figure 5.10: Box-plot of time taken per run for each algorithm and epigenetic variant. The wallclock time is used to record the time taken for each algorithm.

taken over 1500 seconds at 1591, 1597, 1630, and 1585 seconds.

While the time taken is not the most important metric to measure the performance of the algorithms, it can impact an algorithms real world application if a three times increase in computation is required. A key consideration is then whether the improved performance with the gradient-based approaches can result in fewer total evaluations.

5.3.4 Conclusions for gradient-based epigenetic blocking

The investigation into gradient-based epigenetic blocking highlights the potential of using gradient information to guide the epigenetic mechanism. By calculating partial derivatives of decision variables, the value of the gradient at each variable is used to selectively apply epigenetic blocking. Four variations are tested: **GP1**, **GN1**, **PP1**, and **PN1**, which use the gradient information differently when choosing to block decision variables.

The results demonstrate that gradient-based epigenetic approaches can significantly enhance performance in specific problem types, particularly with all positive or negative gradients blocked for MOEA/D-DE. Blocking variables with negative gradients in **GN1** consistently yields the best results across problem types. On the other hand, D-NSGA-II exhibits poor performance with gradient-based epigenetics, except for type III problems, where the approaches show little difference to each other. The difference between positive and negative gradients is diluted due to the summing of objectives, losing some nuance between each objective's gradient direction. Furthermore, positive gradient values can indicate improvement in the objective if the variable value is increased. However, as evolutionary algorithms do not explicitly take advantage of this information, the difference between blocking positive or negative gradients has the same effect.

The computation cost of the gradient-based method is a consideration that must be taken into account. The time required for gradient calculations is approximately three times higher than running the baseline algorithms. While the improved performance can justify the increased computation cost, it may not be suitable in applications with a high number of small evaluations where this additional cost is more significant. However, in applications with an existing high computation cost, the additional cost of computing gradients may not be as noticeable.

5.4 Epigenetics with hyperparameter optimisation

Hyperparameter optimisation (HPO) is a common concept within artificial intelligence, and especially machine learning, to optimise performance with hyperparameter choices. In machine learning, tuning hyperparameters has been shown to improve performance compared to default settings [124, 163]. The use of Evolutionary Computation for optimising and tuning machine learning algorithms has also been extensively studied [111, 186, 213] However, the effect of applying the same hyperparameter optimisation process to Evolutionary Computation itself is not well documented. Tuning Genetic Algorithms has been studied with manual methods [132], and there have been examples of specific problems where hyperparameter optimisation has been used successfully [33].

There are many existing tools developed for hyperparameter optimisation such as SMAC3 (Sequential Model-Based Algorithm Configuration) [117], MIP-EGO (Mixed-Integer Parallel Efficient Global Optimisation) [176], ISAC++ [7], and irace [122]. In Evolutionary Computation, these tools have been explored on specific problems. For example, in using a Genetic Algorithm to identify ambulance locations [33], SMAC3 was used to choose appropriate hyperparameter values. Ye at al. [212] compared the results of multiple hyperparameter optimisation approaches, specifically on a configurable framework for GAs they previously developed, not using any specific state-of-the-art algorithm. Their results showed mixed performance for all the compared approaches, with no single hyperparameter optimisation approach outperforming the others, and a number of instances where the tuned configurations did not outperform the default configuration. The use of Bayesian Optimisation for hyperparameter optimisation has been shown to be effective on Evolutionary Algorithms compared to other hyperparameter tuning methods such as MIP-EGO and irace, which require more evaluations especially when it is expensive to compute a single hyperparameter configuration [161]. Further, SMAC3 has been shown to be an effective hyperparameter optimisation solution for Evolutionary Algorithms compared to packages such as irace and ISAC++ [8]. As the different approaches gave similar performance increases, and SMAC3 utilises Bayesian optimisation to reduce computation costs and has previously shwon to be effective on Evolutionary Algorithms, it was chosen as the hyperparameter optimisation library used. The library and repository for SMAC3 can be found on GithHub (https://github.com/automl/SMAC3).

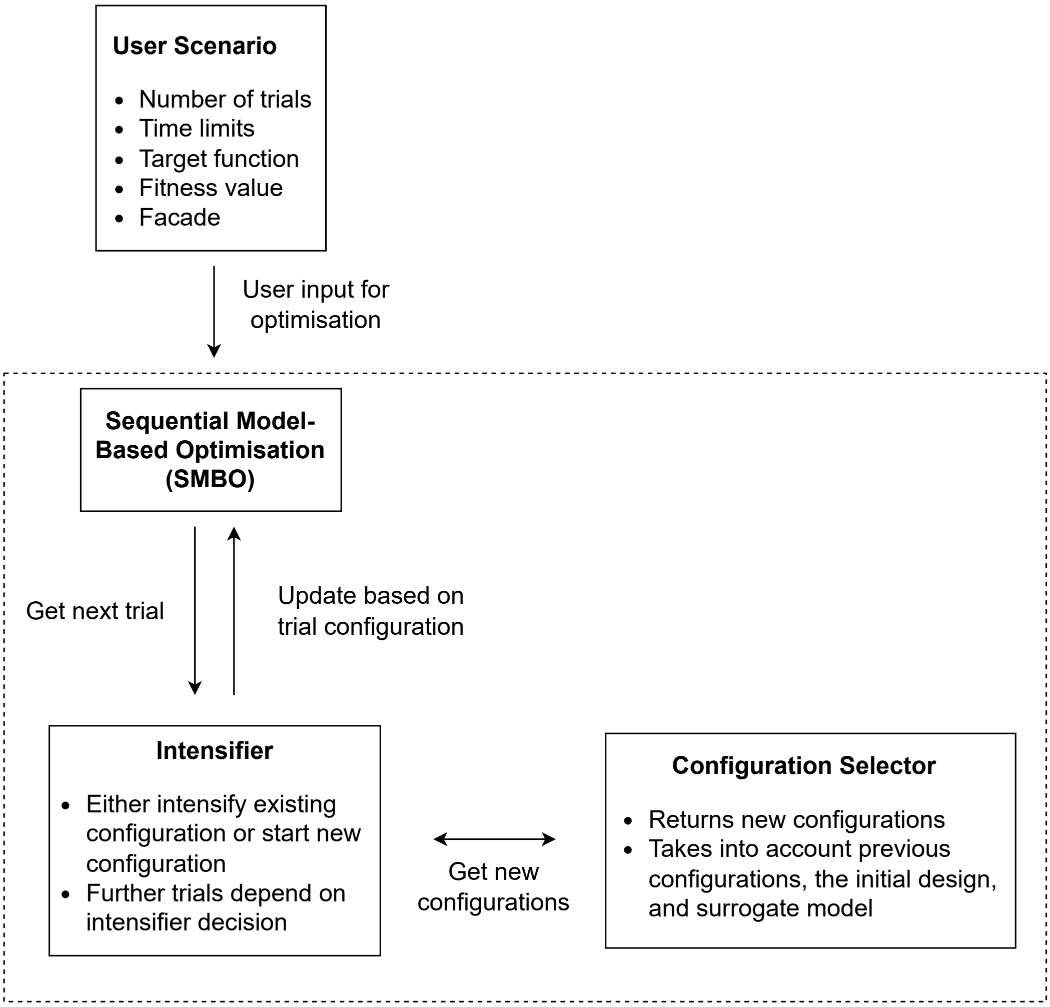

User Scenario
Number of trials
Time limits
Target function
Fitness value
Facade
User input for optimisation
Sequential Model-Based Optimisation (SMBO)
Get next trial
Update based on trial configuration
Intensifier
Either intensify existing configuration or start new configuration
Further trials depend on intensifier decision
Get new configurations
Configuration Selector
Returns new configurations
Takes into account previous configurations, the initial design, and surrogate model

Figure 5.11: Components within SMAC3

SMAC3 uses a Bayesian optimisation loop, keeping track of the number of trials and wallclock time it has remaining. Figure 5.11 shows an overview of the different components within the SMAC3 optimisation loop. A target function returns the fitness value for the objective. In the case of optimising hyperparameters for Evolutionary Computation, the fitness value is measured using a performance metric such as IGD or HV. The configuration selector returns new hyperparameter configurations based on the initial design, surrogate models such as random forests, target function, and history or previously tried configurations. The configurations are evaluated and either intensified via aggressive racing, successive halving or Hyperband, or new configurations are generated. The loop then repeats, with a history of configurations stored to inform both the configuration selector and the intensifier.

Table 5.5: Hyperparameters and the ranges used for the epigenetic mechanism.

Hyperparameter	**Value range**	**Value type**	**Comment**
Crossover rate	0 – 1	Continuous	
Mutation rate	0 – 1	Continuous	
Mutation factor F	0 – 1	Continuous	Only for MOEA/D-DE
Neighbourhood size	2 – 100	Discrete	Only for MOEA/D-DE
Neighbourhood selection probability	0 – 1	Continuous	Only for MOEA/D-DE
Epigenetic rate	0 – 1	Continuous	Only for epigenetics
Mask length	1 – number of variables	Discrete	Only for epigenetics

5.4.1 Configuring the hyperparameter optimisation process

To setup hyperparameter optimisation, hyperparameters that are manually chosen such as crossover rate and mutation rate are used for the HPO process. Table 5.5 shows the range of values for each hyperparameter that the SMAC3 library explores. The algorithm configuration facade for SMAC3 was chosen as it supports multi-objective and multi-fidelity optimisation, while the hyperparameter optimisation facade does not have the same support. Hyperparameters are tuned per problem so that each algorithm variant has the best chance to succeed. Making each configuration of hyperparameters problem-specific also emulates how it would be done for real world problems, where an algorithm is chosen and then tuned to that specific problem.

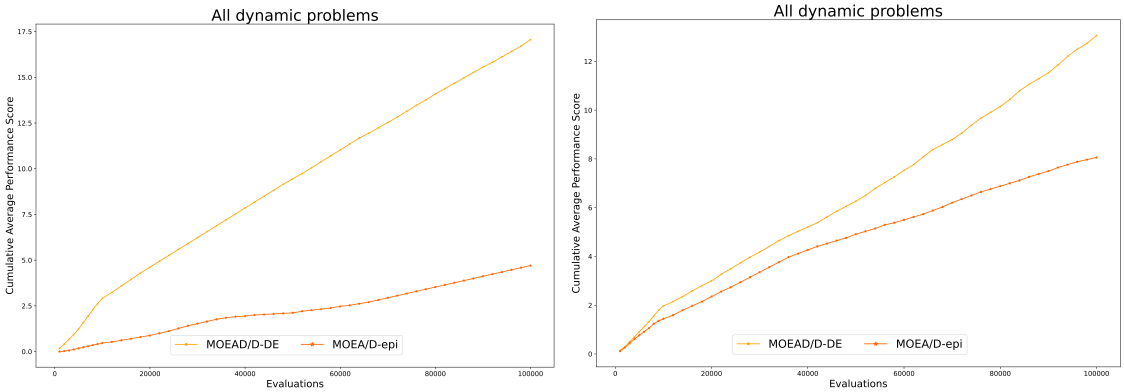
To explore the use of hyperparameter optimisation with the epigenetic mechanism, the probability and mask length of the mechanism are set as additional hyperparameters in the SMAC3 library in addition to the baseline algorithm's hyperparameters. As the variations on dynamically changing the probability and mask length led to mixed results, with some very impressive performances along with some quite poor performances, perhaps the use hyperparameter optimisation can better provide suitable parameters for the epigenetic mechanism.

The optimisation process is limited to a maximum trial-time of 24 hours, and a maximum trial count of 100. To keep the process fair, other hyperparameters such as population size and crossover rate are optimised again for dynamic problems and with the epigenetic mechanism. This leads to separate hyperparameters for the algorithm with and without the mechanism, so the optimisation is not biased towards inclusion of the mechanism or not. Due to the stochastic nature of Evolutionary Computation, a single SMAC3 trial uses objective values from 10 independent runs of the algorithm configuration, taking the average performance metric values of the runs. Fewer independent runs are used compared to the full benchmarking process to allow more trials to be performed in time. Here, trials refers to a single hyperparameter configuration that is selected and tested, and a run is any time an algorithm goes through the optimisation process on a given problem. Hyperparameters are optimised for 100,000 function evaluations per problem, with the mean IGD and HV values across all 100,000 evaluations used as the metrics to be optimised.

The set of hyperparameters optimised are displayed in Table 5.5. The length of the epigenetic mask and the epigenetic rate are hyperparameters specifically used by the epigenetic mechanism. Some other hyperparameters are specific to each algorithm, for example the neighbourhood size and selection for MOEA/D-DE. After hyperparameters are obtained through the SMAC3 process, they are used in 20 independent runs for each algorithm-problem configuration. Here the maximum cumulative average performance score is 50, as there is only one other algorithm that is compared against, the baseline compared to the epigenetic approach, both with hyperparameter tuning.

5.4.2 Hyperparameter optimisation leads to strong performance

Using hyperparameter optimisation with the epigenetic mechanism provided a strong performance increase, compared to the baseline algorithms with hyperparameter optimisation.



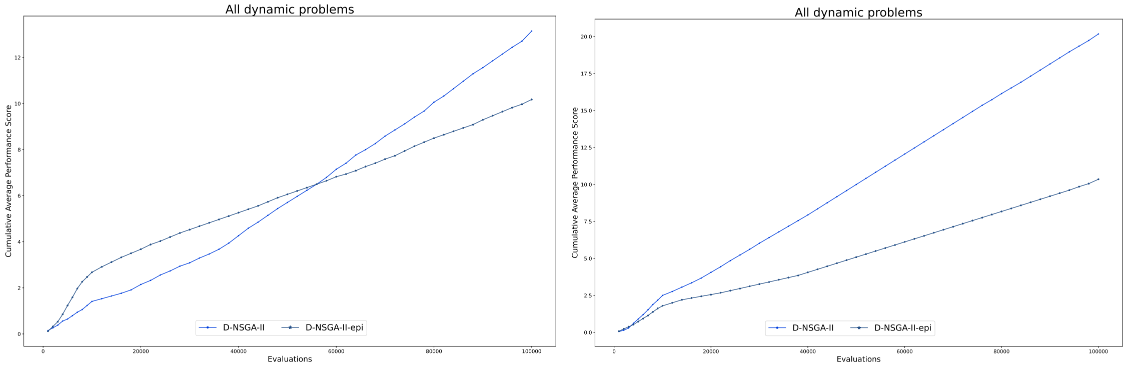
(a) Calculated with HV values.

(b) Calculated by IGD values.

Figure 5.12: Cumulative average performance score for MOEA/D-DE with and without epigenetics using hyperparameter optimisation.

Figures 5.12 and 5.13 display the cumulative average performance scores averaged across all the dynamic problems for MOEA/D-DE and D-NSGA-II respectively. MOEA/D-DE with epigenetics shows a significant performance increase on all problems on the HV metric, with a final cumulative average performance score of 4.7 compared to 17.1 for the baseline algorithm, and a smaller but noticeable difference on IGD, with final cumulative average performance scores of 8.1 for the epigenetic approach compared to 13.1. The results are similar to the gradient results, with the epigenetic methods achieving better HV values than IGD values compared to the baseline, indicating good diversity and spread of solutions across the Pareto front with reduced convergence rates.

D-NSGA-II finds similar results when looking at the HV values, at 10.3 cumulative average performance score for epigenetics compared to 20.2. The larger scores indicate there are more problems where one approach outperforms the other. When using IGD calculations for D-NSGA-II, there is less performance gains made by the epigenetic mechanism, as its early performance is worse but its later performance after 55,500 evaluations is better. Both algorithms show better HV values with the epigenetic mechanism, demonstrating its increased diversity properties with fine tuned hyperparameters while retaining competitive on convergence as shown by the IGD metric.

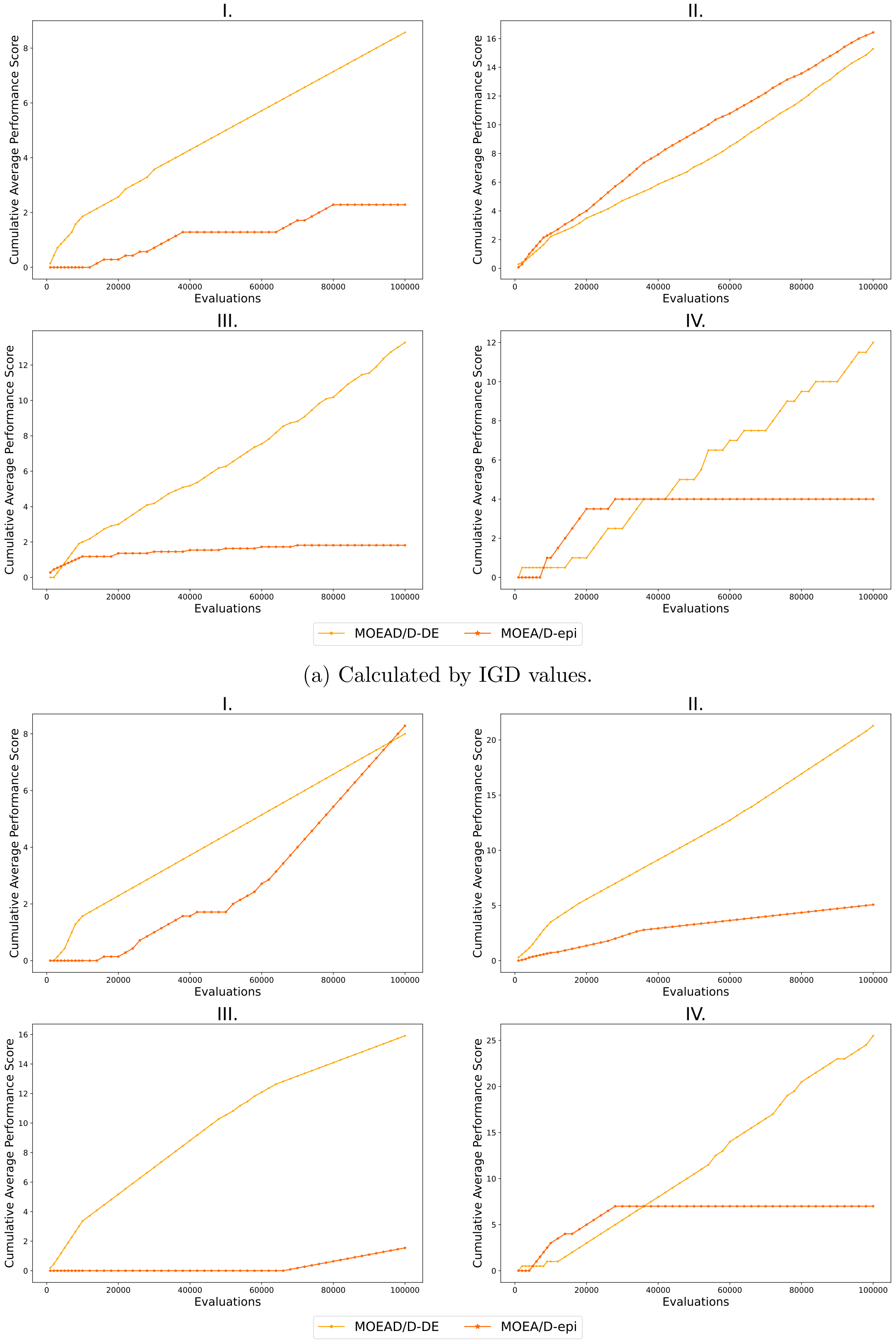


(a) Calculated by IGD values.

(b) Calculated with HV values.

Figure 5.13: Cumulative average performance score for D-NSGA-II with and without epigenetics using hyperparameter optimisation.

Next, the cumulative average performance scores are calculated and grouped by the dynamic problem type. The strengths and weaknesses of each dynamic type can be seen for MOEA/D-DE in Figure 5.14 and D-NSGA-II in Figure 5.15. Both algorithms show similar behaviour with epigenetics on a number of problem types. For example on type IV dynamic problems, both algorithms on both IGD and HV show early worse performance for epigenetics, then from 20,000 – 40,000 function evaluations the epigenetic approach has 0 average performance score until the end of the optimisation, meaning it is always the best compared to the baseline. The two algorithms differ at when the epigenetic approach remains at 0 performance score. MOEA/D-DE achieves this point later at 28,000 and 30,000 for IGD and HV respectively while D-NSGA-II achieves this point earlier at 12,000 evaluations for both IGD and HV metrics. D-NSGA-II with epigenetics is able to achieve 0 average performance score at an earlier point in the optimisation. However, before reaching this score, D-NSGA-II exhibits higher cumulative average performance scores of 9 and 8 based on HV and IGD respectively, while MOEA/D-DE scores 4 and 7 on the same metrics. These results show that D-NSGA-II with epigenetics is worse earlier on in the first 10,000 evaluations compared to the baseline, a trade-off to MOEA/D-DE where it takes longer to reach 0 average performance score but is closer to the baseline before that



(a) Calculated by IGD values.

(b) Calculated with HV values.

Figure 5.14: Cumulative average performance score for MOEA/D-DE with and without epigenetics using hyperparameter optimisation grouped by dynamic problem type.

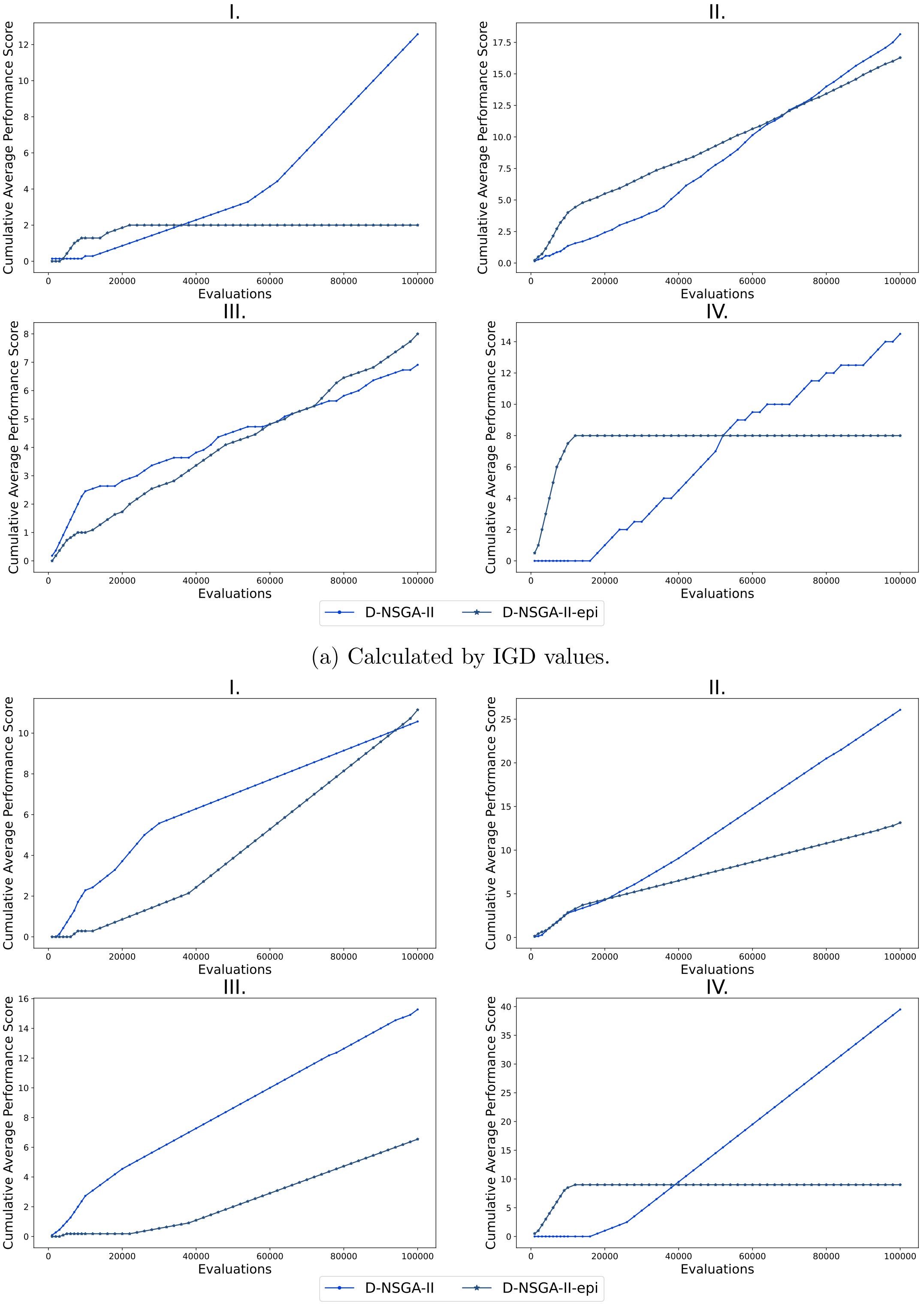
point. Notably for D-NSGA-II, the baseline algorithm also achieves 0 average performance score for the first 10,000 evaluations.

On the other dynamic problem types, MOEA/D-DE with epigenetics demonstrates strong overall performance with a low cumulative average performance score for type II and III dynamic problems with HV. The same strong performance is seen for IGD values for type II problems, though the IGD for type III problems is more mixed, with both the baseline and epigenetic approaches having high average performance scores and the baseline outperforming by a little. Finally, type I problems show good convergence for epigenetics based on IGD values and good early performance based on HV. However, the epigenetic performance degrades towards the end of the optimisation, with both approaches having a similar total average performance score after 100,000 function evaluations. Further, the cumulative average performance score is low at 8 compared to the much higher scores in the type II and III problems. As the maximum cumulative average performance score here is 50, the scores of 20+ for type II and type IV problems with MOEA/D-DE indicate its significant difference compared to performance on type I problems.

Interestingly, D-NSGA-II displays the same behaviour on type I problems based on HV, with good performance early and similar total average performance score to the baseline at the end. For the other problem types with D-NSGA-II, the HV values also show good performance for epigenetics for type II and type III performance, but the IGD values show mixed performance. Notably the mixed performance still leads to a high cumulative average performance scores, indicating that rather than both the baseline and epigenetic approaches having no significant differences, both approaches have different problems where one approach is statistically significantly better than the other. This suggests the tuning of the epigenetic probability and mask lengths greatly increase diversity of the mechanism at the cost of convergence, particularly when both the Pareto front and Pareto set changes.

5.4.3 Specific strengths and weaknesses for fine-tuned epigenetics

On some problems, certain behavioural patterns can be observed with epigenetics. The HV values between MOEA/D-DE with and without the epigenetic mechanism, across the full 100,000 function evaluations for problems where the mechanism performs significantly better are shown in Figure 5.16. UDF4, CDF5, and CDF6 are type II problems, and CDF12 and JY5 are type III problems. In these examples, the algorithm with epigenetic blocking achieves significantly higher HV values than the baseline algorithm. In all cases, the epigenetic mechanism performs well early on, and the performance becomes similar towards the end of the 100,000 function evaluations. In particular, CDF6 and JY5 show significant differences in the first 20,000 − 30,000 evaluations with and without the epigenetic mechanism. After 50,000 evaluations the HV values of both the baseline and the epigenetic approach are close to each other until the end. CDF5 is an exaggerated example



(a) Calculated by IGD values.

(b) Calculated with HV values.

Figure 5.15: Cumulative average performance score for D-NSGA-II with and without epigenetics using hyperparameter optimisation grouped by dynamic problem type.

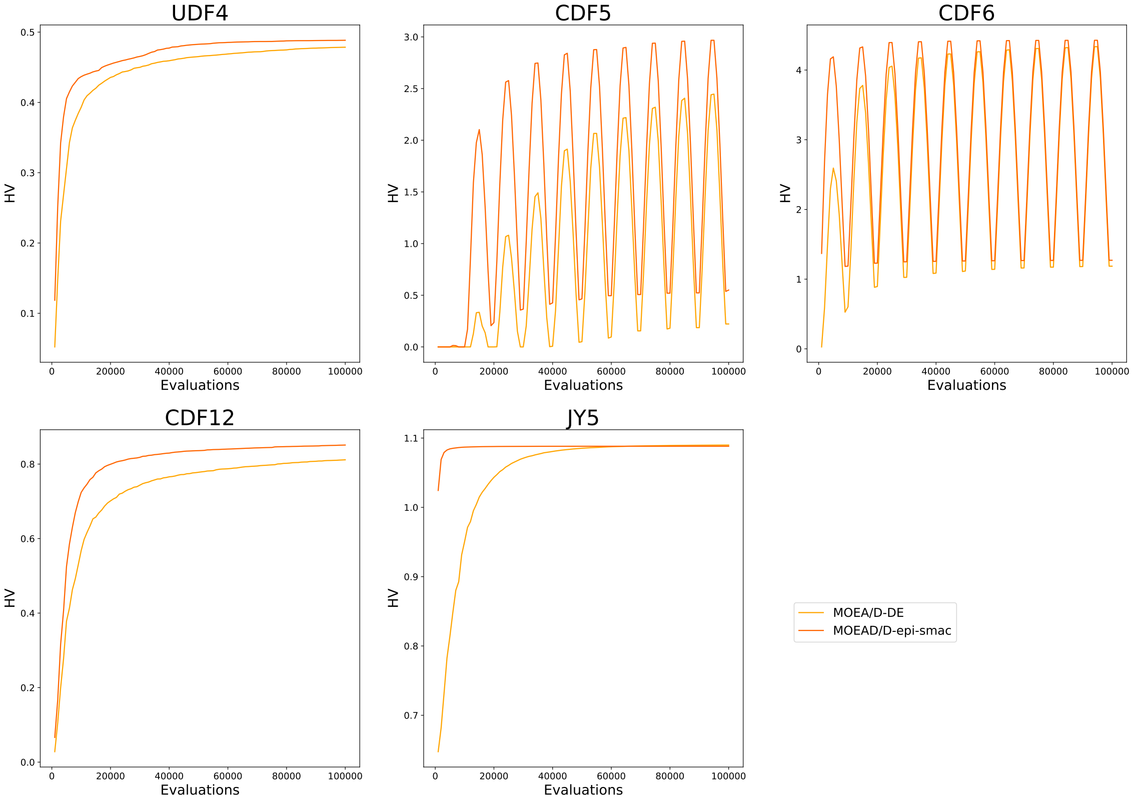


Figure 5.16: Median HV values on dynamic problems for MOEA/D-DE where the epigenetic mechanism performs well when both the mechanism and the baseline algorithm have fine-tuned hyperparameters.

of the same epigenetic behaviour found in CDF6. The baseline algorithm performs poorly at the beginning, and the epigenetic approach continues to outperform until the end of the optimisation. Furthermore, the peaks and troughs are always higher for epigenetics, indicating its rapid adaptive properties to dynamic changes and improving in solution quality after further changes.

Figure 5.17 shows the same benchmark problems for D-NSGA-II. Similar performance patterns are seen, but the effect of the epigenetic mechanism is reduced. For example, CDF6 shows no significant differences between the two approaches and only the first two peaks show a visible different for CDF5. For JY5, MOEA/D-DE with epigenetics maintained its strong performance until 50,000 evaluations, but for D-NSGA-II, the two approaches reach the same HV values by 20,000 evaluations. It is important to note in this case, the performance of the epigenetic mechanism is the same for both algorithms, reaching an HV value of 1.08 in 5,000 for MOEA/D-DE and 6,000 for D-NSGA-II. So the performance difference comes from the baseline D-NSGA-II algorithm showing better performance than the baseline MOEA/D-DE algorithm with optimised hyperparameters.

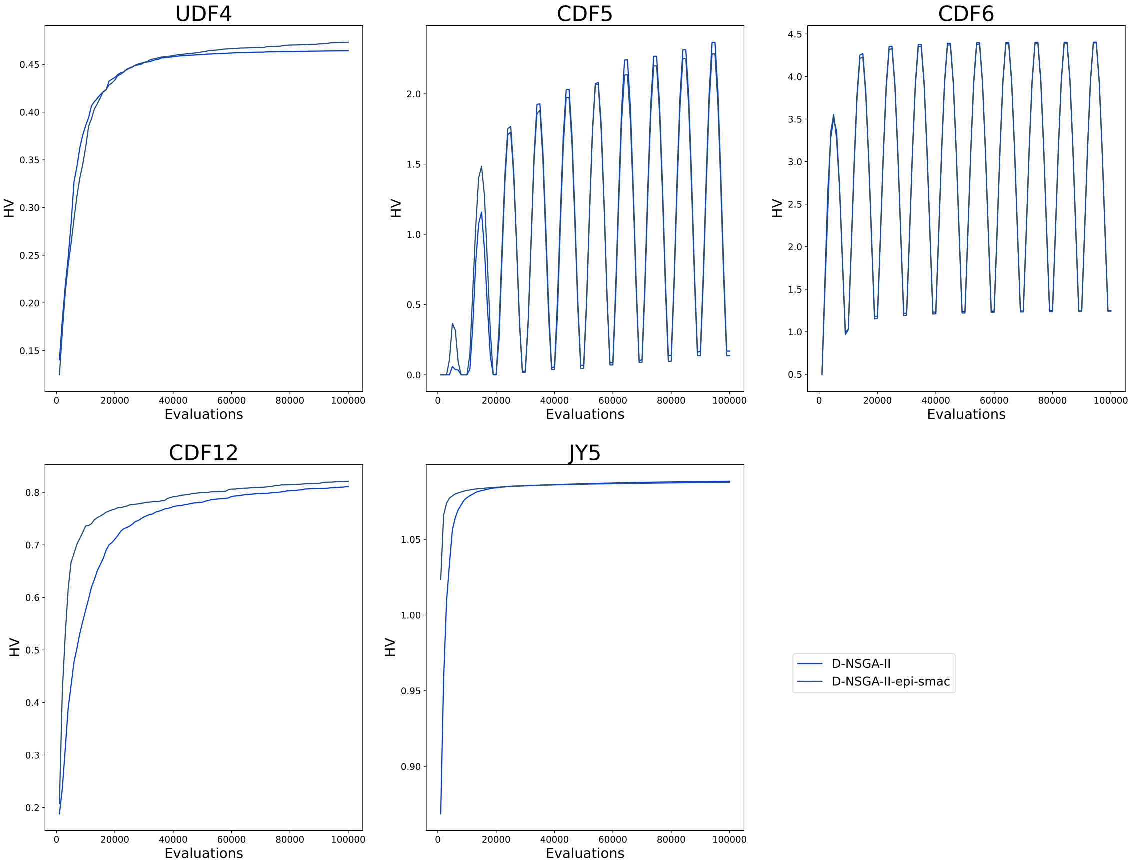
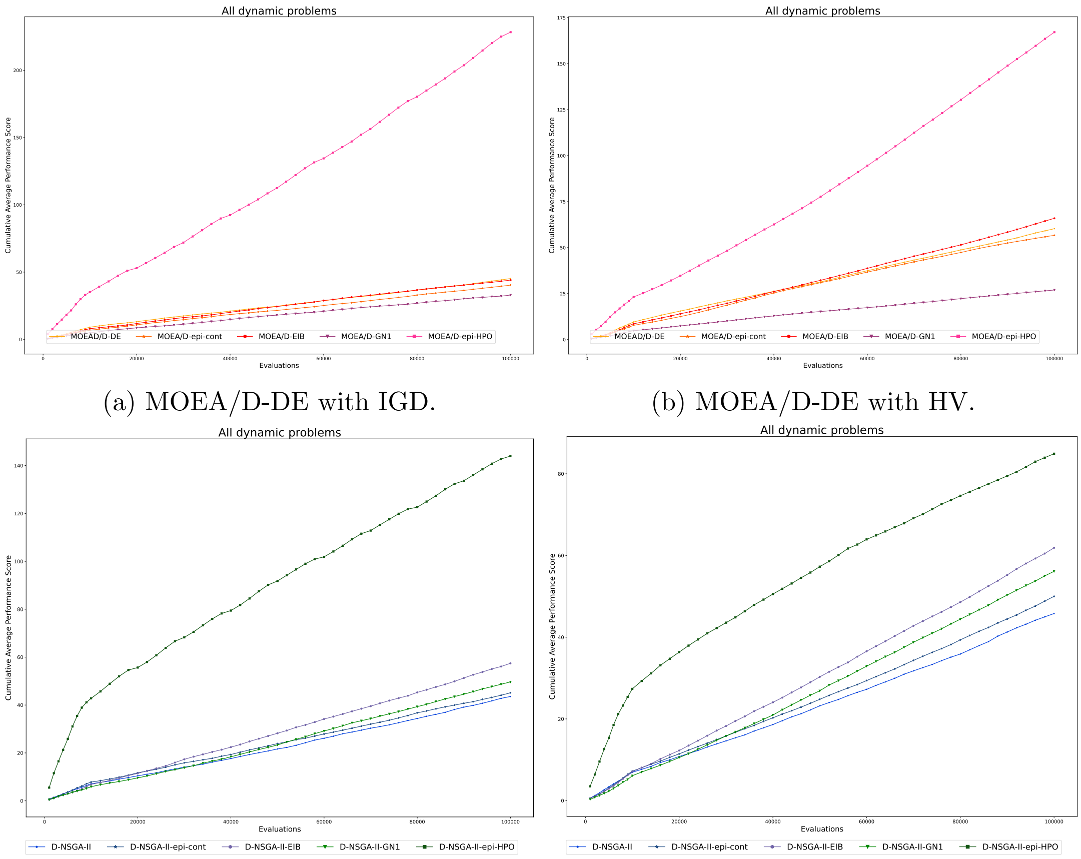


Figure 5.17: Median HV values on dynamic problems for D-NSGA-II where the epigenetic mechanism performs well when both the mechanism and the baseline algorithm have fine-tuned hyperparameters.

5.5 Comparison of the epigenetic mechanism's variations

The use of hyperparameter optimisation leads to a strong positive increase in performance for the epigenetic mechanism, but how strong is this effect compared to the other variations explored previously?

Figures 5.18 shows the cumulative average performance score with all the different epigenetic variations explored compared together. For clarity, only a select few variations from the previous sections are used where there are similar performances between the variations, namely **epi-cont** for the original variations with a contiguous mask, **EIB** for the variation with increasing mask length, **GN1** for the gradient variation, and finally **HPO** for the epigenetic mechanism with optimised hyperparameters. The use of hyperparameter optimisation performing much more poorly compared to all the other variations including the original baseline is surprising. When compared to the hyperparameter-optimised baseline, the epigenetic approach found significantly improved performance. This suggests the hyperparameter optimisation approach is not suitable to dynamic problems, as the SMAC3



(a) MOEA/D-DE with IGD.

(b) MOEA/D-DE with HV.

(c) D-NSGA-II with IGD.

(d) D-NSGA-II with HV.

Figure 5.18: Cumulative average performance score for each epigenetic variant across all dynamic problems.

framework optimises for a single metric value, which is averaged over the optimisation in dynamic problems. The other approaches all show competitive results to each other, with the gradient-based approach showing the best performance for MOEA/D-DE and the baseline algorithm without epigenetics showing the best performance of D-NSGA-II.

Next when dividing the problems into the four dynamic problem types, more interesting performance behaviours are shown. Figure 5.19 first shows the cumulative average performance scores for MOEA/D-DE and its epigenetic variants. In the first dynamic problem type with only changing Pareto sets, the gradient-based approach, but more notably the hyperparameter optimisation approach perform extremely well compared to the other methods. The gradient-based approach with negative gradients in the epigenetic mask demonstrates the best consistent performance across all four dynamic problem types. The performance difference is reduced on type II and III problems, where all other variants and the baseline MOEA/D-DE perform similarly, but a gap can be seen that exhibits the gradient approach's superior performance. The HPO approach retains low average performance score throughout the optimisation for type I problems. This is interesting because the hyperparameter optimisation shows the worst performance when the average performance score is taken from all problems. Its poor performance is continued to be

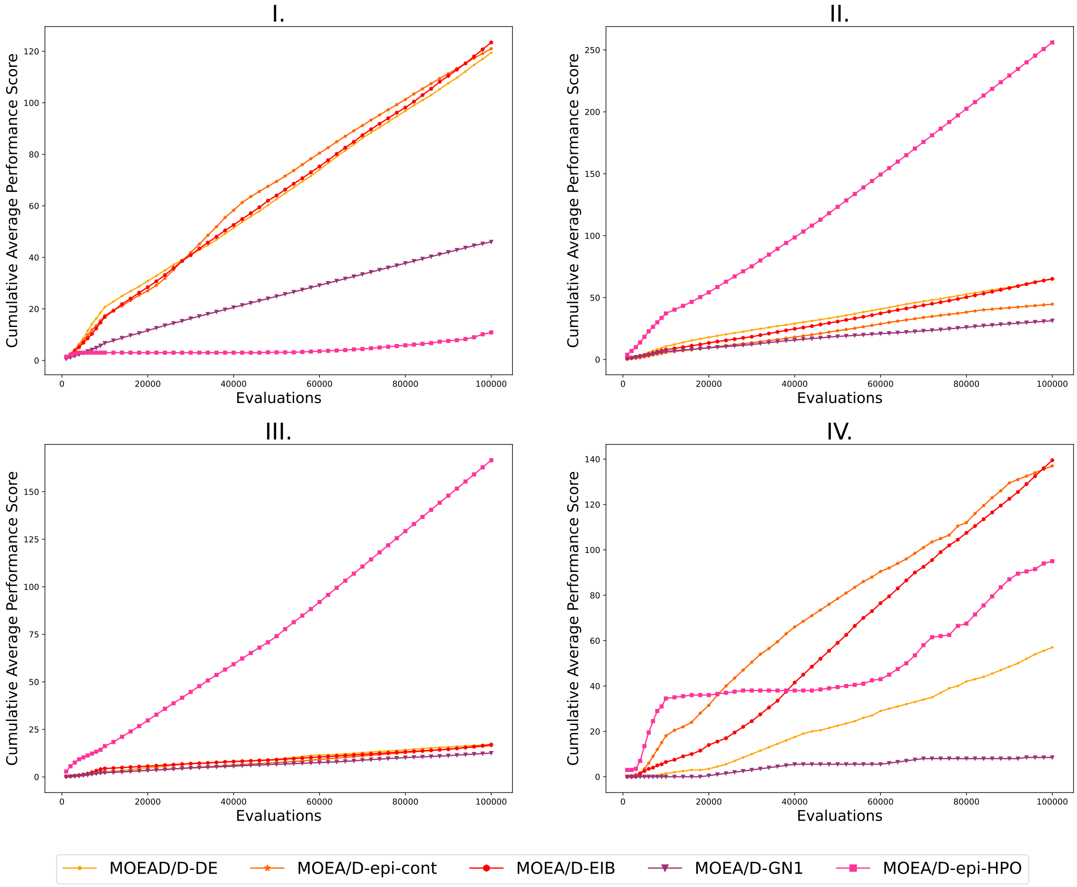


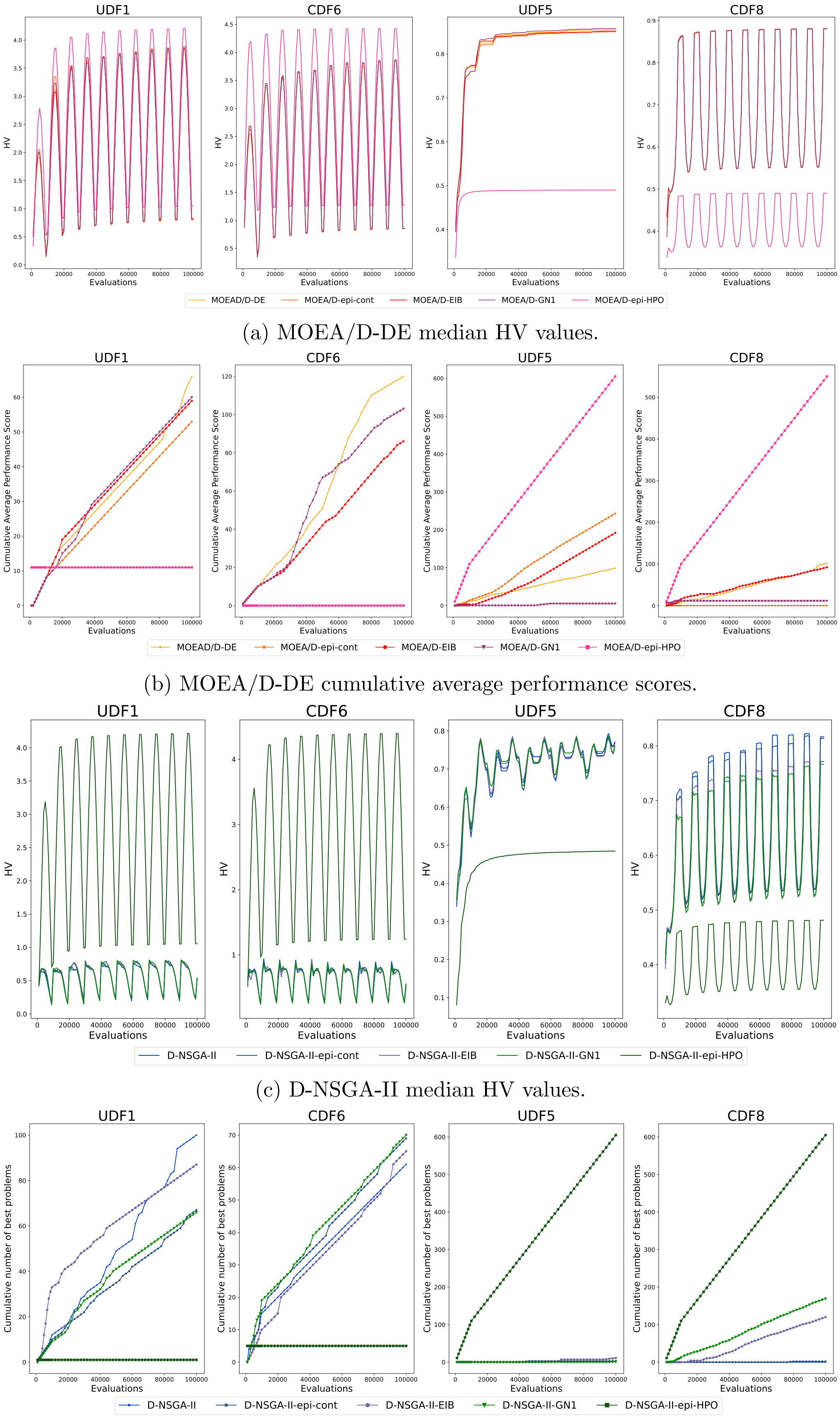
Figure 5.19: Cumulative average performance scores for MOEA/D-DE and each epigenetic variant split by dynamic problem type.

seen in the type II and III dynamic problems, both of which feature dynamically changing Pareto fronts. Additionally, epigenetics with hyperparameter optimisation is below average among the variants for the fourth type of dynamic problems with dynamic constraints. The results point to hyperparameter optimisation being less suitable for problems with changing Pareto fronts.

When applying the epigenetic mechanism to D-NSGA-II, the same performance gains are not seen, suggesting some Evolutionary Algorithms are better suited to the epigenetic mechanism than others. In each dynamic problem type, one or two of the epigenetic variants may have a lower cumulative average performance score than the baseline D-NSGA-II, but the baseline consistently performs well on all types of dynamic problems.

5.5.1 Hyperparameter optimisation's 'poor' performance

Delving deeper into the individual problems to determine the cause of the poor performance from hyperparameter optimisation, Figure 5.20 shows both the median HV values and the cumulative average performance scores for four example type II problems. UDF1



(a) MOEA/D-DE median HV values.

(b) MOEA/D-DE cumulative average performance scores.

(c) D-NSGA-II median HV values.

(d) D-NSGA-II cumulative average performance scores.

Figure 5.20: Type II dynamic problems where the HPO variant performs best and worst in.

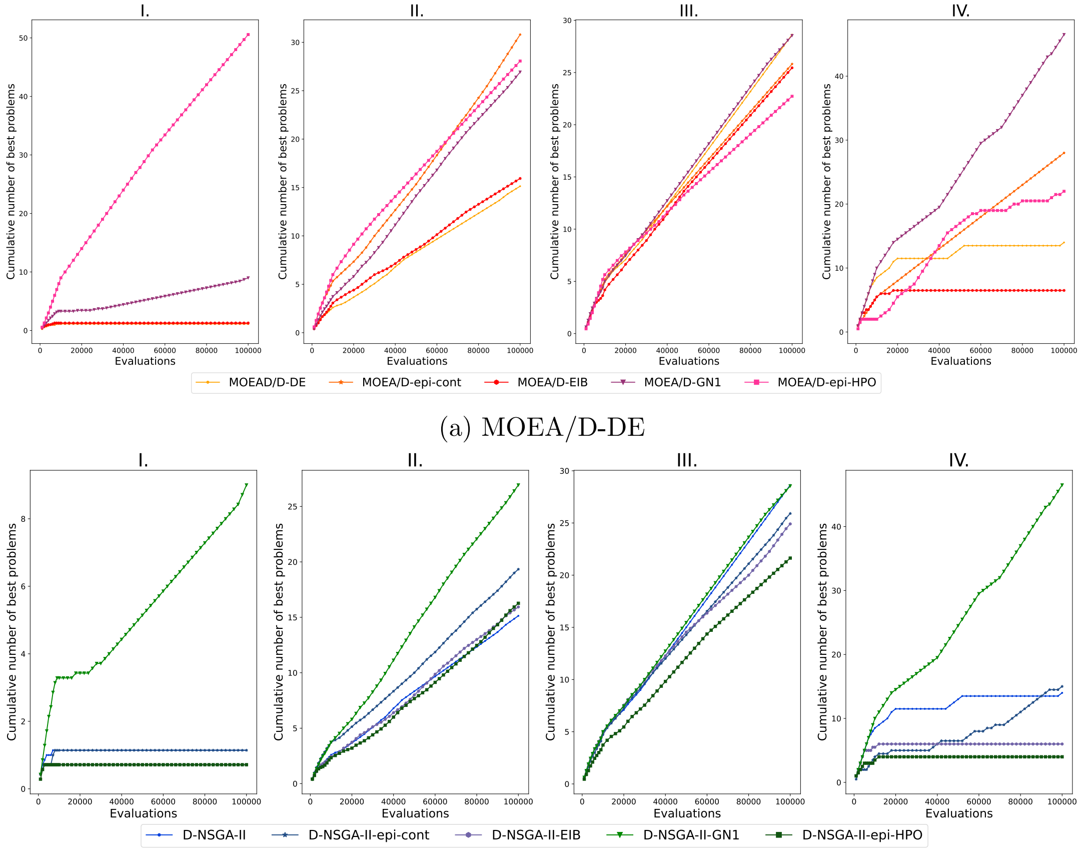
and CDF6 are problems with diagonal shifts while UDF5 and CDF8 are problems with angular shifts. There is a significant performance contrast between the problems where HPO performs well and poorly. However notably, the cumulative average performance scores for UDF5 and CDF8 for the worst performing variant are significantly higher, at over 500, than the scores in UDF1 and CDF6, at less than 120. The HPO approach's poor performance on angular shifting problems points to an overfitting of the hyperparameters, improving performance on suitable problem while losing robustness. The angular shifting nature of the true Pareto front means not only a positional shift of objectives, but also a directional rotational change, more heavily impacting parameters fit for one direction of the front and not another. Another technical reason for the poor performance is other variants all showing similar median HV values and so when they all perform worse against the HPO variant but not each other, their performance scores only increment by 1 at each time step. When the HPO variant performs worse, its performance score increments by 4 or more as all the other variants perform better, leading to its average performance score across all dynamic problems to be much higher than its performance indicates. The same effect is seen for both MOEA/D-DE and D-NSGA-II.

Note that this is an issue stemming from the large number of variants added in the average performance score calculations, which affects the HPO variant more severely than in previous comparisons using the average performance score when only 2 or 3 variants, or 5 in the gradient-based approaches, are compared together at one time. In this case, 13 variations are compared together, leading to the inflation of average performance scores.

Instead of averaging and summing the inflated performance scores, Figure 5.21 shows a cumulative count of the number of problems that each variant performs best in, every 1,000 function evaluations. Here, each variant receives a count of 1 if its performance score is the lowest at every 1,000 function evaluations to show how many problems each performs best in. Now it can be seen that the use of hyperparameter optimisation for MOEA/D-DE maintains the best performance for type I dynamic problems, its performance is competitive for type II problems only just below the other variants for type III problems. Additionally, the **GN1** gradient-based approach continues to show its good performance and surprisingly the contiguous mask with the basic epigenetic hyperparameters achieves a high performance count for type II problems. For D-NSGA-II, the **GN1** variant exhibits superior performance compared to what the cumulative average performance scores suggests, and the HPO approach remains poor performing.

5.5.2 Best general performance

The gradient-based epigenetic approach to block decision variables with negative gradients demonstrates the best performance for dynamic multi-objective optimisation problems. The approach comes at the cost of additional computation time required to compute the gradient values.



(a) MOEA/D-DE

(b) D-NSGA-II

Figure 5.21: Cumulative count of problems each epigenetic variant performs best in (i.e, has the lowest performance score.

The hyperparameter optimisation approach displays extremely strong performances on problems without changing Pareto fronts, because the static objective ensures that tuned parameters remain effective throughout the optimisation. However, on problems with angular shifting Pareto fronts, this method can lead to poor performance as the fine-tuned parameters do not dynamically adapt to the shifting front and the epigenetic blocking mechanism becomes misaligned, making the epigenetic masks unsuitable throughout as the hyperparameter optimisation process takes the average performance as its tuning metric. Angular shifts in the Pareto front are particularly difficult for the HPO approach because it is not only the objectives that change, but the direction of the trade-off, as vertical and horizontal shifts still maintain the same trade-off directions. This makes the HPO approach a high risk strategy, where the properties of the dynamic problem should be known beforehand to utilise it well. The other epigenetic approach in **EIB** and **EIP** for increasing probability and epigenetic mask lengths can show some good performance but often shows similar and mixed performances to the baseline without epigenetics. Therefore in general cases, the gradient-based approach is recommended if the additional computation cost is acceptable relative to the gains in performance and convergence. In specific dynamic problems where only the Pareto set changes over time, a hyperparameter optimisation approach is recommended, and if a full hyperparameter optimisation process is not feasible due to resource, time, or computation constraints, a less granular grid search to fine-tune the epigenetic hyperparameters can be used.

Chapter 6

Application on a real-world voyage optimisation problem

To apply and test the epigenetic mechanism on a real-world problem, a commercial voyage optimisation problem is run based on a Voyage Optimisation Software tool (T-VOS). The problems are performed on a test set initially developed by Khan et al. [101]. Two objectives are optimised in this problem: fuel consumption and voyage time. By minimising fuel consumption, both the cost of a vessel's voyage, and emissions per voyage can be reduced. Further, using voyage time as an objective allows the Pareto front to suggest a variety of arrival times to best suit a particular voyage. Met-ocean data models are used to provide information such as wind speeds, ocean currents, and weather updates, all of which feed into the calculation for fuel consumption. As a ship travels on different routes, the kinds of weather and ocean data changes, resulting in a complex optimisation problem with irregular weather constraints. This makes the epigenetic mechanism a potentially suitable mechanism in this application.

6.1 The voyage optimisation problem

A total of 10 voyages are tested, 6 have calm weather conditions and 4 have more severe weather conditions. These problems are considered constrained multi-objective optimisation problems. A grid of nodes are generated to form the route, with constraints added to prevent grounding, provide ship safety in severe weather, and adhere to ship traffic and routing schemes in certain regions. A mesh is generated over the voyage to limit the search space within the relevant ocean routes based on the First Order Approximation (FOA) route. By default, the width of the mesh is 2000km, 1000km on each side of the FOA route, which means that optimiser can route the vessel up to 1000km from the shortest great-circle route. The decision variables are made up of the sequence of nodes, forming the waypoints of the route contained within the mesh, and the speed of the vessel between each pair of waypoints. Re-routing may occur depending on the weather conditions during the voyage, requiring new changes to the route. More severe weather would require more re-routing compared to the calmer weather voyages. Historical Met-ocean data is used to provide the weather information, which may differ from the real-time use of the software that utilises weather forecasts instead, but is a useful approximation to how the software adapts re-routing in changing weather. The met-ocean constraints were set to 6 metres for wave heights and 40 knots for wind speed. The T-VOS system uses a Speed Over Ground (SOG) operational mode, which means that the SOG is optimised for each waypoint on the route and then assumed to be constant between waypoints.

Epigenetic rate	**Mask length**	**Min fuel**	**Min time**	**Overall**
0	0	10.5833	13.6667	12.1250
0.333	0.1	9.5833	9.8333	9.7083
0.666	0.1	8.5000	9.3333	8.9167
0.9	0.1	12.2500	9.9167	11.0833
0.333	0.4	9.9167	11.6667	10.7917
0.666	0.4	11.2500	11.6667	11.4583
0.666	0.7	14.6667	11.2500	12.9583
0.75	0.1	8.4167	8.5000	8.4583
0.75	0.2	**4.6667**	7.3333	**6.0000**
0.75	0.3	9.1667	11.6667	10.4167
0.75	0.25	11.0000	10.2500	10.6250
0.75	0.15	9.5000	10.5000	10.0000
0.71	0.2	15.5000	12.4167	13.9583
0.8	0.2	11.1667	12.0833	11.6250
0.7	0.2	12.5000	12.0000	12.2500
0.666	0.2	14.3333	15.7500	15.0417
0.69	0.2	8.1667	13.2500	10.7083
0.74	0.2	12.3333	8.6667	10.5000
0.76	0.2	11.3333	**7.2500**	9.2917

Table 6.1: The hyperparameter grid search done to find the best performing epigenetic rate and mask length. The minimum fuel and time solutions are ranked, and the average rank is displayed to find the best hyperparameters. The grid search is limited due to the computational resources of the T-VOS platform. The epigenetic rate and mask lengths of 0 represent the standard algorithm without epigenetics. The algorithm can be quite sensitive to changes in the parameters, as small changes can lead to a large degradation of performance.

Voyages with severe weather make the optimisation problem more difficult due to more frequent re-routing and safety constraints of the vessel. Each voyage and algorithm configuration were benchmarked over 7 runs. A total of 664 generations are generated in each run. The number of runs and generations are due to the computational resources and operational costs of running the T-VOS system. There is a termination condition on the number of evaluations, with a soft limit at 400,000 evaluations and a hard limit at 800,000. The optimisation is always run up to the soft limit, and if no navigable and safe route is found within that limit, the optimisation is then run up until the hard limit. All routes here finished within the 400,000 evaluation soft limit. For the epigenetic blocking mechanism, a grid search was used to determine the best probability and mask length as it was less feasible to run hyperparameter optimisation libraries such as SMAC3 with the the expensive computations involved. The proprietary nature of the voyage optimisation software further made it difficult to implement a gradient-based approach with the epigenetic mechanism. An epigenetic rate of 0.2 and a mask length of of 0.75 of the number of decision variables was found to provide the best results. A non-uniform grid search was done across the parameter space, with initial exploratory values of 0.3, 0.6, and 0.9 for the epigenetic rate, and 0.1, 0.4, and 0.7 for the mask length. Then a more focused dense search was done for mask lengths in the range of $[0.6, 0.8]$, and epigenetic rates in the range of $[0.1, 0.2]$. Table 6.1 shows the detailed hyperparameters that were tested, and how they ranked compared to each other and the standard approach without epigenetics. As the problem contains decision variables in a correlated sequence, a contiguous epigenetic mask is used.

The T-VOS software uses cMLSGA [74] as the underlying evolutionary algorithm. A variation on the SBX crossover method is used and by default the optimiser sets waypoints spaced 250km apart in open waters, and changes speed approximately every 5 waypoints. While near coasts or within traffic schemes, this restriction is relaxed. The default restrictions can be changed based on route and client preferences, but are put in place to reduces the number of nodes necessary to explore in the route-finding mesh, reducing computational needs. Furthermore, end-user experience from captains and routing operators do not prefer overly detailed and jagged routes with constant turning and speed changes, as it creates questions and uncertainty as to why so many routing adjustments are necessary, reducing adoption and trust of the voyage optimisation software. Therefore, the restriction of the optimisation search space not only helps to reduce computational resources, but reflects human preferences.

Both IGD and HV values are calculated as performance metrics. The IGD values are calculated by taking the best route found by the voyage optimisation software from all historic runs as the “true” Pareto front. However, only a sample of 14 Pareto optimal fronts are used in the reference set from the historical best runs stored on the system as some of the older best solutions were lost. Therefore the “true” Pareto front used in the IGD calculations may not be adequate and should be interpreted cautiously, not as a final metric that definitively determines performance on the problems.

6.2 Performance of epigenetics in voyage optimisation

Table 6.2 displays the averaged final IGD and HV values for the best route found in each run in the voyage optimisation problem. The calm and severe weather voyages and listed below, along with their associated reference points for Hypervolume calculation, which are 10% worse than the nadir points found for each route, in the form (Time in hours, Fuel in tonnes).

Voyages with calm weather conditions:

- Houston – Halifax (358.80, 238.38)
- Houston – Cuxhaven (789.06, 526.12)
- Recife – Bangkok (1640.40, 972.09)
- Corpus Christi – Cape Town (1386.40, 760.41)
- Rio de Janeiro – Ningbo (1924.07, 1098.56)
- Ras Tanura – Mailiao (813.58, 573.68)

Voyages with severe weather conditions:

- London – Sunny Point (647.68, 376.51)
- London – Sunny Point 2 (676.06, 388.22)
- San Francisco – Kushiro (714.41, 413.27)
- Cape Horn – Tauranga (813.89, 436.77)

HyperVolume reference points are taken from the nadir points of the reference Pareto fronts, which takes all routes from historical runs to create a Pareto optimal front using non-dominated sorting, forming a single reference front.

None of the six calm weather voyage showed a statistically significant difference between the epigenetic and non-epigenetic approach, indicating there is limited if any difference between the algorithms by the end of the optimisation. The epigenetic approach neither improves or detracts from the original approach on these calm weather problems.

On the four severe weather routes, there is a statistical significance between the two approaches on all the routes, with the epigenetic approach finding the best results in all four routes for both IGD and HV. There are significant differences between the metric values with IGD, for example in the Cape Horn to Tauranga voyage, the epigenetic approach has a final IGD value of 2.736 compared to 4.8187 for the baseline algorithm. Due to the rough IGD calculation on few optimal fronts, it is not indicative of performance, but does suggest performance improvements when compared to the values of the two approaches on calm weather voyages. The HV values further support the performance improvements made by the epigenetic mechanism, as the same results are shown for both metrics.

Figure 6.1 displays the trade-off fronts between the fuel and time objectives for problems with a statistically significant IGD and HV value. In each of these routes, non-dominated solutions are found with epigenetics that show an improvement in fuel savings

Route	**Epi**	**IGD**			**HV**		
		Average	Std	Sig	Average	Std	Sig
Houston – Halifax	✓	0.4731	0.2391	×	0.0593	0.0005	×
	×	0.6394	0.2419		0.0590	0.0005	
Houston – Cuxhaven	✓	1.8391	1.1269	×	0.0379	0.0008	×
	×	3.0688	0.6180		0.0369	0.0004	
Recife – Bangkok	✓	2.2563	0.6646	×	0.0356	0.0003	×
	×	2.1264	0.7793		0.0356	0.0003	
CorpusChristi – CapeTown	✓	1.5868	0.3358	×	0.0415	0.0002	×
	×	1.6203	0.3970		0.0415	0.0003	
Rio – Ningbo	✓	2.9052	1.3648	×	0.0354	0.0005	×
	×	2.6012	1.2843		0.0355	0.0005	
RasTanura – Mailiao	✓	1.3803	0.5660	×	0.0139	0.0003	×
	×	1.5356	0.3515		0.0138	0.0002	
London – SunnyPoint	✓	**1.0574**	0.1907	✓	**0.0482**	0.0001	✓
	×	1.4005	0.1761		0.0476	0.0001	
London – SunnyPoint2	✓	**0.8322**	0.1252	✓	**0.0443**	0.0002	✓
	×	1.2221	0.1583		0.0438	0.0002	
San Francisco – Kushiro	✓	**0.9649**	0.2364	✓	**0.0346**	0.0002	✓
	×	1.9946	0.6014		0.0335	0.0005	
Cape Horn – Tauranga	✓	**2.7360**	0.5269	✓	**0.0266**	0.0005	✓
	×	4.8187	1.1710		0.0257	0.0007	

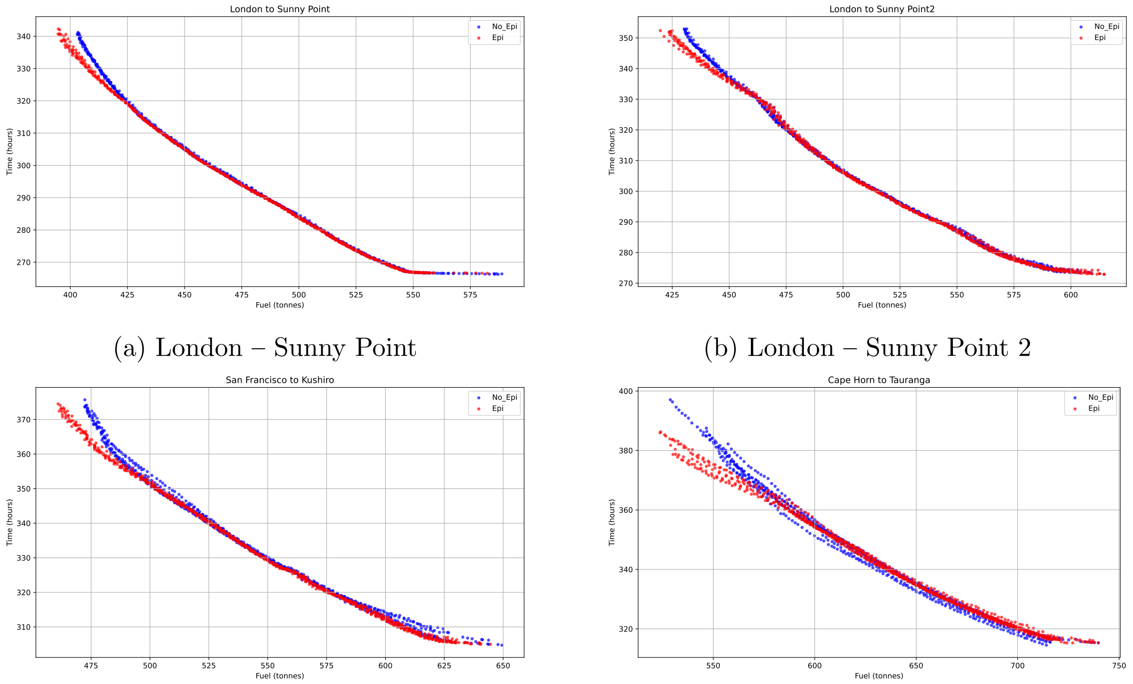
Table 6.2: Final IGD and HV values for each voyage. Statistical significance is calculated with a Wilcoxon rank sum test. The best values for each voyage are highlighted in bold. A total of

without an increase in the time required for the journey. In some cases such as the Cape Horn to Tauranga route, the epigenetic algorithms finds solutions that dominate in both fuel saved and voyage time objectives, though there is a trade-off where the baseline method finds solutions that save time for a lower fuel cost after 575 tonnes of fuel.

6.2.1 Fuel savings

As it is difficult to determine performance improvements solely on IGD and HV metrics, the objective of fuel consumption on a route is analysed to determine further performance differences between the two approaches. The average saving of fuel consumption, across all voyages, with the epigenetic approach is 1% compared to the voyage optimisation without epigenetics. The potential savings are significantly higher when operating on severe weather conditions, with additional savings of 1.7 – 3.3%. Additionally, for all voyages, the epigenetic approach has a higher convergence rate, reaching higher fuel savings in fewer generations. This provides a considerable benefit, reducing the cost of providing these simulations in a real world application.

The calm weather problems are relatively simple requiring limited re-routing. In these problems, the most optimal points are found after 300 – 400 generations by both variants.



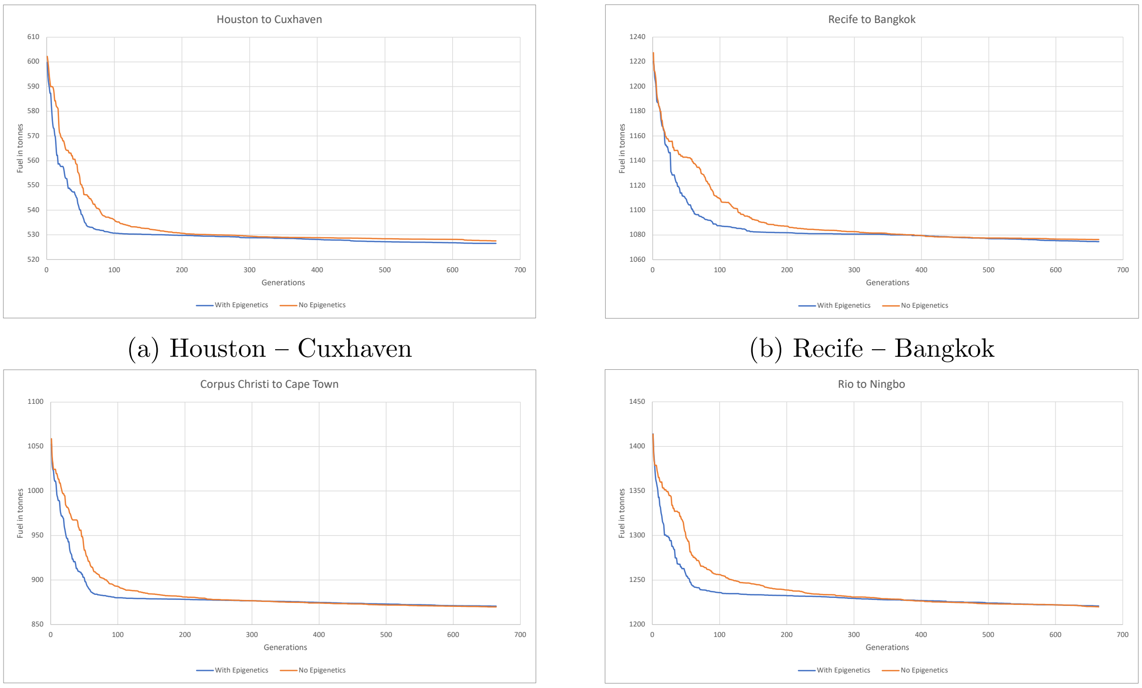
(a) London – Sunny Point

(b) London – Sunny Point 2

(c) San Francisco – Kushiro

(d) Cape Horn – Tauranga

Figure 6.1: Approximate Pareto fronts found in all runs by T-VOS with and without epigenetics at the end of the optimisation.



(a) Houston – Cuxhaven

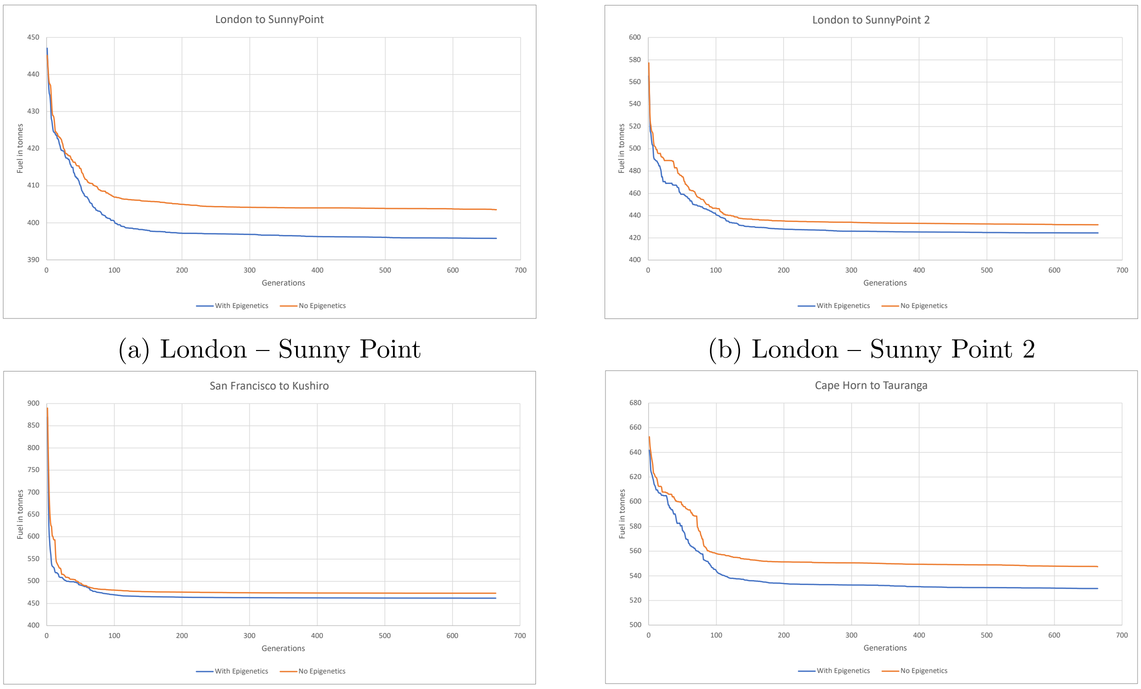
(b) Recife – Bangkok

(c) Corpus Christi – Cape Town

(d) Rio de Janeiro – Ningbo

Figure 6.2: Routes found with the best fuel consumption per generation in calm weather condition voyages.

Therefore no significant savings are observed when looking at the final approximated Pareto fronts. The IGD and HV values further showed no statistical significance on the calm weather routes except for the Houston to Cuxhaven route in the final approximated Pareto fronts. However, the higher convergence rate of the epigenetic mechanism can be seen when viewing the fuel usage over time in Figure 6.2. The best single route with the lowest fuel consumption after 664 generations is shown, as that is the route that matters most a voyage optimisation problem trying to save fuel and emissions. Both the epigenetic and non-epigenetic algorithms converge after 300 – 400 generations with negligible differences after, but the epigenetic approach is able to converge faster within the first 100 – 200 generations. A gap is seen between the two approaches before the optima are reached, showing the epigenetic mechanism's rapid adaptation even in calm weather conditions. On some voyages such as Rio de Janeiro to Ningbo, the epigenetic mechanism shows faster convergence as early as the first 20 generations of optimisation. In practical terms, the faster convergence means a lower computational soft limit can be used, saving on computational resources and operational costs. This in turn impacts the scalability of the system, allowing more routes and vessels to be computed with the same computational resources.



(a) London – Sunny Point

(b) London – Sunny Point 2

(c) San Francisco – Kushiro

(d) Cape Horn – Tauranga

Figure 6.3: Routes found with the best fuel consumption per generation in severe weather condition voyages.

In severe weather conditions, the safety of the vessel is threatened due to the constantly changing weather and thus more re-routing is required. Figure 6.3 shows the epigenetic approach finding better optimal routes once both approaches converge, leading to significant fuel savings in these problems. The fast convergence properties are maintained, with lower fuel consumption routes found in fewer generations. For example in the Cape Horn to Tauranga voyage, the epigenetic mechanism finds routes that consume 540 tonnes of fuel or less by 100 generations of computation, compared to the algorithm without epigenetics which never finds a route with less than 540 tonnes of fuel even after all 664 generations. The epigenetic approach maintains a higher convergence rate in addition to better optimal routes. The requirements for more complex re-routing enables the adaptive nature of epigenetics to improve final fuel savings. The results demonstrates the strengths of the epigenetic approach in problems that are more complex.

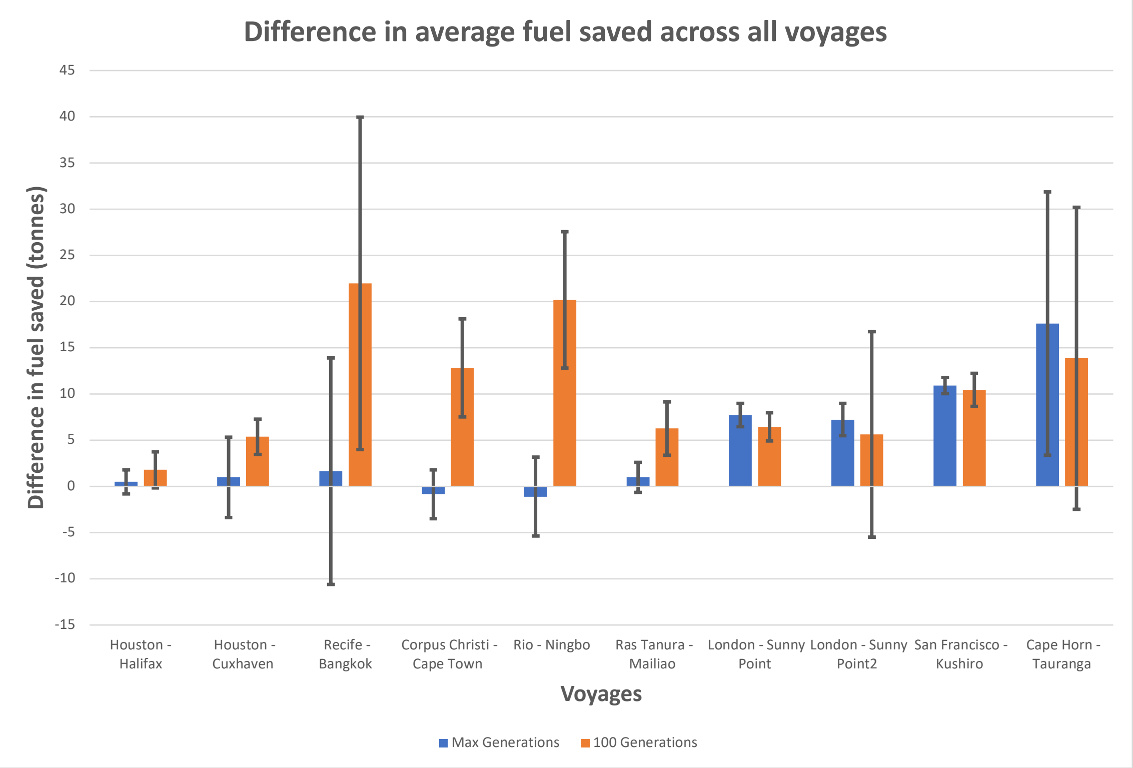


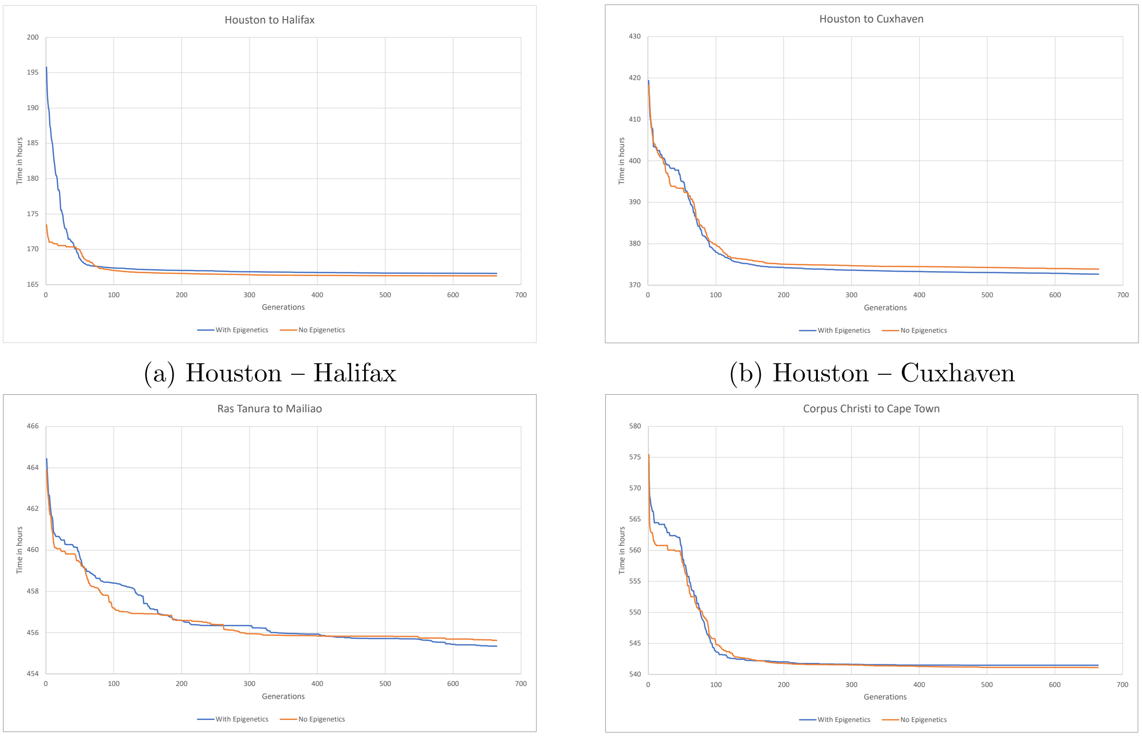
Figure 6.4: Fuel saved with the epigenetic approach compared to the baseline without epigenetics. Positive values mean fuel saved with the epigenetic approach and negative values mean fuel saved using the baseline algorithm.

The difference in fuel savings between calm and severe weather voyages is shown in Figure 6.4. The first six voyages are the calm weather voyages and the last four are the severe weather voyages. Negative values represent voyages where the baseline approach without epigenetics saves more fuel, for example Corpus Christi to Cape Town at maximum generations. The relative fuel savings at 100 generations and the maximum 664 generations are both shown for comparison of convergence.

On the calm weather voyages at the maximum number of generations, there is little to no fuel savings found between the baseline approach and the epigenetic approach, with no voyage having over 2 tonnes of fuel saved in favour of the baseline or epigenetic approach. When taking the fuel differences at 100 generations, significant fuel savings are found for the calm weather voyages using the epigenetic mechanism, demonstrating the benefits of the higher convergence rate. In systems with less computational resources, or voyages with higher computational demands, the ability to find the best routes in fewer generations is crucial. Furthermore, fuel savings are found for the severe weather voyages after both 100 and the maximum 664 generations after the algorithm converges, demonstrating not only the increased convergence rate, but the epigenetic mechanism's ability to find more optimal voyage routes under more complex situations which require more re-routing.

6.2.2 Time savings

The other objective that is optimised in this voyage optimisation problem is the time it takes to reach the destination port. Compared to the performance improvements found in fuel usage, the epigenetic approach finds insignificant differences to the baseline approach at 0.07% across all voyages, and only 0.15% on only the severe weather voyages.



(a) Houston – Halifax

(b) Houston – Cuxhaven

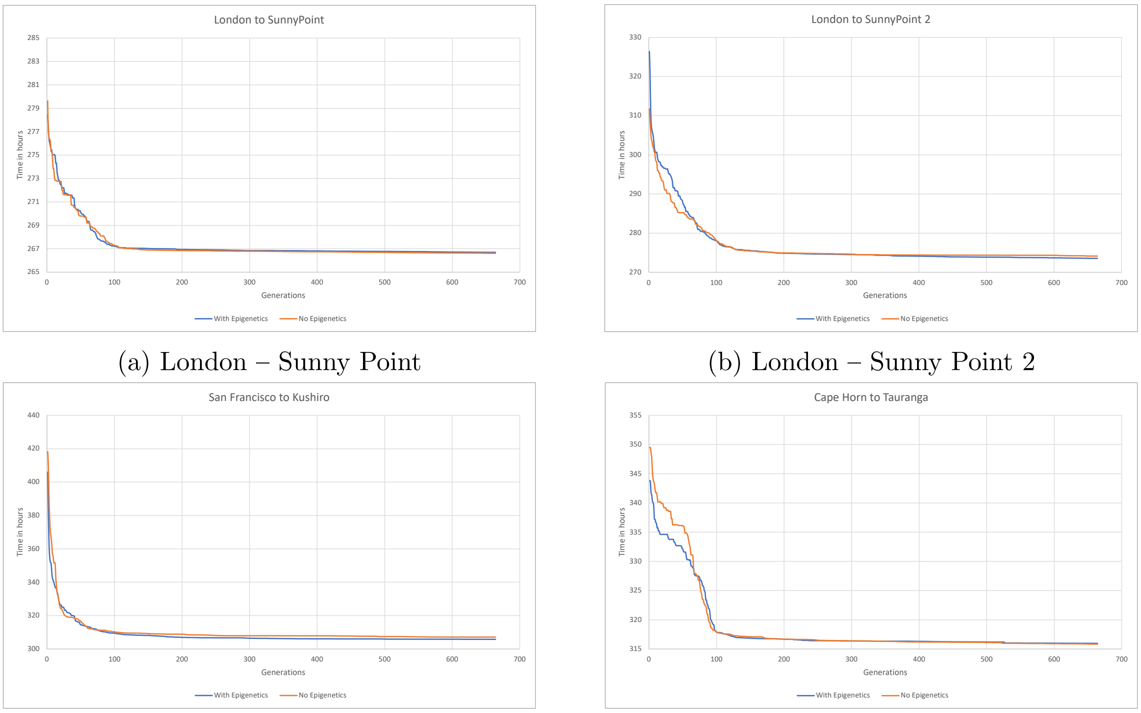
(c) Ras Tanura – Mailiao

(d) Corpus Christi – Cape Town

Figure 6.5: Routes found with the best voyage time per generation in calm weather condition voyages.

When looking at the results over time for voyage time, no clear performance improvements are seen for the epigenetic approach. Figure 6.5 shows the voyage time over time. Generally the epigenetic approach has slower convergence in the first 100 generations and both approaches converge within 1 hour of the other. For example, the Houston to Cuxhaven route converges at 373.9 hours for the baseline and 372.6 hours for the epigenetic approach, and the Corpus Christi to Cape Town route ends at 541.1 hours for the baseline and 541.4 hours for the epigenetic approach. These small differences make little to no impact on the final routes used by the vessels.

Furthermore, the range of voyage time values is quite small between the most and least optimised routes, indicating there is less possible room for optimisation. The average percentage differences between the routes from generation 1 and generation 664 is 11% for the voyage time objective and 24% for the fuel consumption objective. This is compounded by the fact that 1 ton of fuel saved represents a significantly larger economic cost than 1 hour of time in commercial shipping.



(a) London – Sunny Point

(b) London – Sunny Point 2

(c) San Francisco – Kushiro

(d) Cape Horn – Tauranga

Figure 6.6: Routes found with the best voyage time per generation in severe weather condition voyages.

When examining the voyage times for the severe weather routes, there are no significant performance differences for the epigenetic approach over the baseline method. Figure 6.6 illustrates the voyage time comparison over all generations. On the London to Sunny Point 2 voyage, the epigenetic algorithm shows slower convergence while on the Cape Horn to Tauranga route, the baseline algorithm shows slower convergence, both within the first 100 generations. The final voyage times after all 664 generations show no significant difference between the approaches, with an average percentage difference of 0.15%. Similar to the calm weather voyages, both approaches performed comparably, further indication that there is less possible optimisation for voyage time compared to fuel consumption.

6.3 Epigenetic effects on voyage optimisation

The application of the epigenetic mechanism to a real-world voyage optimisation problem shows promising results in both calm and severe weather conditions to reduce fuel consumption in commercial shipping. Comparable results are found when analysing the voyage time objective, and good results are found for the objective of fuel consumption. Increased convergence properties are notable for the calm weather voyages, where close to optimal fuel savings are found within the first $100 - 200$ generations of optimisation, compared to the $300 - 400$ generations required for the baseline algorithm without epigenetics. Better final solutions for fuel savings are found in addition to the increased convergence in severe weather voyages. The IGD and HV performance metrics further demonstrate statistically significant results for the epigenetic approach on severe weather voyages, with no statistically significant results on all but one calm weather voyages, where the epigenetic approach outperforms the baseline.

In practical terms, although voyages time are drastically longer than computation times, hundreds of hours compared to a few minutes, there is still a practical need for efficient and fast optimisation algorithms. First, platforms such as T-VOS have end-users which expect fast responses from the software and not have to wait 10+ minutes or over an hour if a brute force or inefficient approach was used. Secondly, in practical scenarios, vessels require periodic updates typically every twenty-four hours, but potentially as frequently as every six hours, to account for changing weather, port, or traffic constraints. For large-scale operations where thousands of ships all utilise the same system, the computational efficiency of an algorithm optimising routes becomes important. For example, if a fleet of 1,000 ships requires re-routing every 24 hours, and the system takes 10 minutes per ship, the total computation time would be 600,000 seconds, or 167.67 hours every 24 hours. This becomes a significant operational cost when scaling up to more ships and more frequent updates, and highlights the practical impact of efficient optimisation approaches. The results demonstrate the epigenetic mechanism's application to a real world multi-objective optimisation problem providing impact on both theoretical performance from IGD and HV metric, and practical reductions of fuel and emissions for planned shipping voyages.

Chapter 7

Conclusions

On an initial glance, the field of Evolutionary Computation looks to be full of diverse mechanisms and algorithms. There are different families and flavours of Evolutionary Algorithms and Genetic Algorithms, and a large number of Swarm Intelligence algorithms inspired by diverse forms of different animal behaviour. However, from an evolutionary perspective, most of the existing algorithms and mechanisms use the same few evolutionary concepts. Swarm Intelligence in particular has many algorithms with the same fundamental mechanics of position updates to create new generations of solutions, differing only in the update function or exploration mechanics, leaving a gap in Evolutionary Computation to explore well researched contemporary evolutionary mechanisms.

This thesis set out to investigate the non-genetic evolutionary concept of epigenetics, from the Extended Evolutionary Synthesis, with the hypothesis that it would improve on dynamic multi-objective optimisation problems by improving convergence, as epigenetics in nature captures the rapid adaptation of organisms to changing environments. It was found that while a probabilistic epigenetic blocking mechanism can improve on a variety of static optimisation benchmark problems, additional mechanics are required to improve its performance on dynamic problems.

7.1 Key findings

The main contributions of the work in this thesis can be split into two main sections:

A biological framework is developed to analyse and break down Evolutionary Computation algorithms based on evolutionary theory from the Extended Evolutionary Synthesis. Then, a focused benchmarking study on Genetic Algorithms and Particle Swarm Optimisation algorithms illustrates their differences and analogies to evolutionary theory.

- In Swarm Intelligence, all algorithms use some form of social learning and communication. While the details of the methods may be different, the fundamental mechanism is the same. With the exception of Ant Colony Optimisation, the other Swarm Intelligence algorithms all utilise a form velocity or positional update, with differences in the update function such as using the top three global best solutions in the Grey Wolf Optimiser, or drawing from a Lévy distribution for Cuckoo Search.

- The performance of Genetic Algorithms and Particle Swarm Optimisation algorithms exhibit different strengths and weaknesses depending on the specific algorithm, rather than a fundamental performance gap. Additional strategies such as incorporating multiple swarms in PSOs can improve both convergence and diversity on a different set of problems than the original algorithm, at the cost of reduced effectiveness on previously well-performing problems.

An epigenetic blocking mechanism is developed and thoroughly tested. The blocking mechanism takes inspiration from epigenetic gene silencing, where genes are inherited but not expressed into the phenotype. The aspect of inheritance is incorporated, allowing the positions of blocked decision variables to be transferred and reused in future generations. The initial iteration of the mechanism shows shows strong convergence properties for static problems, but mixed performance on dynamic problems, where epigenetics would have been expected to improve an organism's fitness in the natural world. The use of a contiguous mask works best on problems with decision variables that are correlated in sequence. In contrast, a non-contiguous mask is more suitable for problems with irregular or uncorrelated decision variables. Investigating further into the conditions and variations on the epigenetic blocking mechanism for dynamic problems found that:

- The use of gradient values to inform the epigenetic mechanism on which decision variable to be blocked results in significantly improved performance at the cost of additional computation required to compute the variable gradients. However, due to Evolutionary Algorithms not having explicit mechanisms to increase or decrease variable values, the advantage of using positive or negative gradients becomes nullified, making both approaches perform similarly well to each other.
- Optimising the hyperparameters on a per-problem basis can lead to the best results when using the same hyperparameter optimisation process on the same algorithm without epigenetics. When compared to the other epigenetic variations, hyperparameter optimisation can lead to the best results on problems with changing Pareto sets, or diagonal shifts in the Pareto front. However, the hyperparameter optimisation process struggles on problems with angular Pareto front shifts, leading to the worst results on these types of problems. The properties of the dynamic problem must therefore be known beforehand to utilise hyperparameter optimisation for the epigenetic mechanism.

Finally, the application of the epigenetic mechanism in a real world voyage optimisation software system led to significant performance improvements. The epigenetic mechanism demonstrated faster convergence in calm weather voyages and improved final solutions on more dynamic severe weather voyages, indicating the strength of its adaptive properties, and exhibiting potential economic and environmental benefits in fuel savings.

7.2 Discussion and future work

The results of this thesis show the potential of epigenetics as a modern evolutionary mechanism applied in Evolutionary Algorithms, with promising results in solving both static and dynamic multi-objective optimisation problems. Additional modifications to the approach, such as the use of gradient information led to improved performance with the epigenetic mechanism, but there are areas where more adaptive methods could be used:

1. **Dynamic adaptation to calculating decision variable gradients**

   The results from the gradient-based approach found that choosing to block variables with positive or negative gradients in the epigenetic mask provides the most consistent performance improvements. However, the meaning behind positive or negative values is lost due to the lack of explicit mechanisms to act on decision variable values. Furthermore, the computational cost of computing every variable gradient is expensive. To build on this approach, the magnitude of variable gradients can be used instead. The probabilistic approach showed inconsistent performance, indicating that a threshold of gradient value should be used instead, and all decision variables above the threshold should be blocked. Methods to reduce the number of gradient calculations such as gradient approximation or calculating the gradients less frequently can be explored. The summing of objectives to form the gradient value also loses nuance of objectives trade-offs in opposite directions and only favours the objective with the largest magnitude. Incorporating all objectives could further increase the performance improvement gained.

2. **Creating an epi-genotype**

   The epigenetic masks created and inherited by the epigenetic blocking mechanism incorporates elements of inheritability and memory. In evolutionary biology, epigenetic changes occur due to environmental changes, and can be reversed when the environment reverts back to its original properties. The current epigenetic blocking mechanism can make changes to increase convergence, inherit the changes in the next generation, and inherit the epigenetic mask to remember which decision variables are to be blocked, but does not revert the changes. Additional information such as the values of the masked parent's previous decision variables, or previous gradient information could be stored by creating an epi-genotype that records and marks epigenetic changes, allowing them to be reversed or turned off when necessary. This further relates to an improved adaptive approach, where the epigenetic mechanism is triggered from environmental cues during optimisation such as dynamic shifts or premature convergence rather than probabilistically.

3. **An element of memory**

   The current inheritance mechanism of the epigenetic masks enables future generations to store the epigenetic masks of its parents, creating a trans-generational epigenetic effect. However, there is no means to remove or forget the epigenetic masks, which may lead the population to inherit masks which are biased towards the early stages of optimisation. Selection pressure can alleviate part of the issue, as solutions with poor epigenetic masks are less likely to be chosen to reproduce the next generation, but this only applied when the epigenetic mechanism is active and the mask is actively silencing decision variables. Otherwise, poor epigenetic masks can continue to be carried onto future generations like a recessive gene. To counteract this, additional mechanisms can be put in place to wipe the epigenetic memory from the population. A naive approach could remove epigenetic masks after a set number of generations. More advanced approaches may incorporate the epi-genotype to evaluate and encode the fitness of an epigenetic mask independent to the fitness of the parent or child solutions.

Furthermore, other epigenetic processes should also be considered and studied, alongside the epigenetic blocking mechanism to determine their cooperative effects. For example, genomic imprinting [200] where decision variables from a parent are imprinted onto child solutions, but not expressed until the epigenetic mechanism is triggered. Instead of blocking the decision variable immediately during reproduction, the imprints are carried on by the child and only utilised later.

Appendix A

Benchmark problem properties

Selected state-of-the-art static and dynamic multi-objective benchmark problems. The problems are categorised based on their properties. The problem categories are based on the categories in [76] where additional problems such as LZ_09 [114] and DTLZ [48] are added.

- d is the number of variables.
- o is the number of objectives.
- Categories I – VI are static unconstrained problems.
- Categories VII – X are static constrained problems.
- Types I – IV are the dynamic problems.

A.1 Benchmark problem categories

Table A.1: Category I. Simple unconstrained problems

Problem	**d**	**o**	**Properties**
ZDT1	30	2	Convex
ZDT2	30	2	Concave
ZDT3	30	2	Disconnected
ZDT4	10	2	Convex, multimodal
ZDT6	10	2	Biased, concave, multimodal

Table A.2: Category II. Convex unconstrained problems

Problem	d	o	Properties
UF1	30	2	Complex Pareto set, non-linear
UF2	30	2	Complex Pareto set, non-linear
UF3	30	2	Complex Pareto set, multimodal
LZ09_F1	10	2	Complex Pareto set, convex
LZ09_F2	30	2	Complex Pareto set, convex
LZ09_F3	30	2	Complex Pareto set, convex
LZ09_F4	30	2	Complex Pareto set, convex
LZ09_F5	30	2	Complex Pareto set, convex
LZ09_F6	10	3	Complex Pareto set, convex
LZ09_F7	10	2	Complex Pareto set, convex, multimodal
LZ09_F8	10	2	Complex Pareto set, convex, multimodal

Table A.3: Category III. Concave unconstrained problems

Problem	d	o	Properties
UF4	30	2	Concave, complex Pareto set
UF8	30	3	Disconnected
UF10	30	3	Complex Pareto set
WFG4	22	3	Concave, multimodal
WFG5	22	3	Concave, deceptive
WFG6	22	3	Concave, non-separable
WFG7	22	3	Biased, concave
WFG8	22	3	Biased, concave, non-separable
WFG9	22	3	Biased, concave, non-separable
DTLZ2	12	3	Concave
DTLZ3	12	3	Concave, multimodal
DTLZ4	12	3	Biased, concave, degenerate
DTLZ5	12	3	Degenerate
DTLZ6	12	3	Biased, disconnected
LZ09_F9	30	2	Concave, non-linear, Complex Pareto set

Table A.4: Category IV. Linear/Mixed unconstrained problems

Problem	d	o	Properties
UF7	30	2	Complex Pareto set, linear
WFG1	22	3	Biased, Mixed
WFG3	22	3	Non-separable, linear, degenerate
DTLZ1	7	3	Linear, multimodal

Table A.5: Category V. Discontinuous unconstrained problems

Problem	**d**	**o**	**Properties**
UF5	30	2	Complex Pareto set, discrete points, linear
UF6	30	2	Complex Pareto set, disconnected
UF9	30	3	Complex Pareto set
WFG2	22	3	Convex, disconnected, non-separable
DTLZ7	22	3	Biased, disconnected, multimodal
MOP4	10	2	Discontinuous

Table A.6: Category VI. Imbalanced unconstrained problems

Problem	**d**	**o**	**Properties**
MOP1	10	2	Convex
MOP2	10	2	Convex
MOP3	10	2	Concave
MOP5	10	2	Convex
MOP6	10	3	Linear
MOP7	10	3	Concave
IMB1	10	2	Convex
IMB2	10	2	Linear
IMB3	10	2	Concave
IMB4	10	3	Linear
IMB5	10	3	Concave
IMB6	10	3	Linear
IMB7	10	2	Convex, non-separable
IMB8	10	2	Linear, non-separable
IMB9	10	2	Concave, non-separable
IMB10	10	3	Linear

Table A.7: Category VII. Imbalanced constrained problems

Problem	**d**	**o**	**Properties**
IMB11	10	2	Convex
IMB12	10	2	Linear
IMB13	10	2	Concave
IMB14	10	3	Linear

Table A.8: Category VIII. Diversity-hard problems

Problem	**d**	**o**	**Properties**
DAS-CMOP1(5)	30	2	Concave, disconnected
DAS-CMOP2(5)	30	2	Mixed, continuous
DAS-CMOP3(5)	30	2	Disconnected, linear, multimodal
DAS-CMOP4(5)	30	2	Concave, disconnected
DAS-CMOP5(5)	30	2	Mixed, disconnected
DAS-CMOP6(5)	30	2	Discrete points, disconnected

Table A.9: Category IX. Feasibility-hard constrained problems

Problem	**d**	**o**	**Properties**
DAS-CMOP1(6)	30	2	Concave, disconnected
DAS-CMOP2(6)	30	2	Mixed, continuous
DAS-CMOP3(6)	30	2	Disconnected, linear, multimodal
DAS-CMOP4(6)	30	2	Concave, disconnected
DAS-CMOP5(6)	30	2	Mixed, disconnected
DAS-CMOP6(6)	30	2	Discrete points, disconnected

Table A.10: Category X. Convergence-hard constrained problems

Problem	**d**	**o**	**Properties**
DAS-CMOP1(7)	30	2	Concave, disconnected
DAS-CMOP2(7)	30	2	Mixed, continuous
DAS-CMOP3(7)	30	2	Disconnected, linear, multimodal
DAS-CMOP4(7)	30	2	Concave, disconnected
DAS-CMOP5(7)	30	2	Mixed, disconnected
DAS-CMOP6(7)	30	2	Discrete points, disconnected

Table A.11: Type I dynamic problems, where only the Pareto set changes.

Problem	**Continuity**	**PS shift**	**PF shift**
JY1	Continuous	Vertical shift	No change
JY6	Continuous	Vertical shift	No change
FDA1	Continuous	Vertical shift	No change
CDF1	Continuous	Curvature change	No change
CDF2	Continuous	Curvature change, horizontal shift	No change
CDF3	Discontinuous	Curvature change	No change
CDF4	Discontinuous	Vertical shift	No change

Table A.12: Type II dynamic problems, where both the Pareto set and Pareto front changes.

Problem	**Continuity**	**PS shift**	**PF shift**
UDF1	Continuous	Vertical shift	Diagonal shift
UDF2	Continuous	Curvature change, vertical shift	Diagonal shift
UDF4	Continuous	Horizontal shift	Angular shift
UDF5	Continuous	Vertical shift	Angular shift
UDF8	Continuous	Random shift	Random shift
UDF9	Continuous	Random shift	Random shift
JY2	Continuous	Curvature change	Angular shift
JY3	Continuous	Curvature change	Angular shift
JY7	Continuous	Vertical shift	Angular shift
FDA3	Continuous	Vertical shift	Vertical shift
CDF5	Continuous	Diagonal shift	Diagonal shift
CDF6	Continuous	Diagonal shift	Diagonal shift
CDF7	Discontinuous	Curvature change, vertical shift	Diagonal shift
CDF8	Discontinuous	Curvature change	Angular shift
CDF13	Discontinuous	Random shift	Random shift

Table A.13: Type III dynamic problems, where only the Pareto front changes.

Problem	**Continuity**	**PS shift**	**PF shift**
UDF3	Discontinuous	No change	Diagonal shift
UDF6	Distinct points	No change	Angular shift, diagonal shift
UDF7	Continuous	No change	Radial shift
JY4	Discontinuous	No change	Discontinuity shift
JY5	Continuous	No change	Angular shift
JY8	Continuous	No change	Angular shift
FDA2	Continuous	No change	Angular shift
CDF9	Continuous	No change	Curvature change, diagonal shift
CDF10	Continuous	No change	Curvature change
CDF11	Discontinuous	No change	Horizontal shift
CDF12	Discontinuous	No change	Curvature change, angular shift

Table A.14: Type IV dynamic problems, where the constraints change over time.

Problem	**Continuity**	**PS shift**	**PF shift**
CDF14	Discontinuous	No change	No change
CDF15	Discontinuous	No change	No change

A.2 Reference set sizes

The sizes of the reference Pareto fronts used to calculated the IGD values of each problem are displayed in tables

Problem	Reference set size	Source	Reference point
ZDT1 – 6	1000	jMetal library	(1.1, 1.1)
UF1 – 4, 6, 7	1000	jMetal library	(1.1, 1.1))
UF5	21		(1.1, 1.1)
UF8 – 10	10000		(1.1, 1.1, 1.1)
LZ09_F1 – 5, 7 – 9	500	jMetal library	(1.1, 1.1)
LZ09_F6	990		(1.1, 1.1, 1.1)
DTLZ1	999	jMetal library	(0.55, 0.55, 0.55)
DTLZ2, 3, 7			(1.1, 1.1, 1.1)
DTLZ4	1000		(1.1, 1.1, 1.1)
DTLZ5, 6	200		(0.78, 0.78, 1.1)
IMB1 – 14	3000	Self-generated	(1.1, 1.1)
MOP1 – 7	2000	Self-generated	(1.1, 1.1)
WFG1	1113	jMetal library	(2.2, 4.4)
WFG2	119		
WFG3	916		
WFG4	1326		
WFG5	837		
WFG6	426		
WFG7	2494		
WFG8	527		
WFG9	2600		
DASCMOP1,2,4,5(5)	504	Original paper	(1.1, 1.1)
DASCMOP3,6(5)	11		(1.1, 1.1)
DASCMOP1,2,4,5(6)	1000		(1.65, 1.65)
DASCMOP3,6(6)	124		(1.65, 1.65)
DASCMOP1,4(7)	717		(1.1, 1.1)
DASCMOP2,5(7)	1000		(1.1, 1.1)
DASCMOP3,6(7)	124		(1.1, 1.1)

Table A.15: Reference set sizes for the static benchmark problems.

Problem	**Reference set size**	**Source**	**Reference point**
UDF1,2,4,5,7,8	10000	Self-generated	(1.1, 1.1) – (2.2, 2.2)
UDF3,6,	22		(1.1, 1.1) – (2.2, 2.2)
CDF1,2,5,6,10	10000	Self-generated	(1.1, 1.1)
CDF3,7,11	20		(1.1, 1.1) – (2.2, 2.2)
CDF4	1065		(1.1, 1.1)
CDF5,6	10000		(1.1, 1.1) – (2.2, 2.2)
CDF8	4668 – 6149		(1.1, 1.1) – (0.88, 1.1)
CDF9	6722 – 10000		(1.1, 1.1) – (2.2, 2.2)
CDF12	6217 – 6251		(1.1, 1.1)
CDF13	7501		(1.1, 1.1)
CDF14	1000 – 10000		(1.1, 1.1)
CDF15	3414 – 6412		(1.1, 1.1)
JY1 – 8	10000	Self-generated	(1.1, 1.1) – (2.2, 2.2)
FDA1 – 3	10000	Self-generated	(1.1, 1.1) – (2.2, 2.2)

Table A.16: Reference set sizes for the dynamic benchmark problems.

Bibliography

[1] A. Ahmadi-Javid. "Anarchic Society Optimization: A human-inspired method". In: *2011 IEEE Congress of Evolutionary Computation (CEC)*. 2011, pp. 2586–2592.

[2] R. Akbari, V. Zeighami, and K. Ziarati. "MLGA: A Multilevel Cooperative Genetic Algorithm". In: *2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*. 2010, pp. 271–277.

[3] R. Akbari, R. Hedayatzadeh, K. Ziarati, and B. Hassanizadeh. "A multi-objective artificial bee colony algorithm". In: *Swarm and Evolutionary Computation* 2 (2012), pp. 39–52. ISSN: 2210-6502. URL: https://www.sciencedirect.com/science/article/pii/S2210650211000411.

[4] P. Alberch. "From genes to phenotype: dynamical systems and evolvability". In: *Genetica* 84 (1 1991), pp. 5–11.

[5] R. Albertson and K. T.D. "Genetic and development basis of cichlid trophic diversity". In: *Heredity* 97 (3 2006), pp. 211–221.

[6] P. J. Angeline. "Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences". In: *Evolutionary Programming VII*. Ed. by V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 601–610. ISBN: 978-3-540-68515-9.

[7] C. Ansótegui, J. Gabàs, Y. Malitsky, and M. Sellmann. "MaxSAT by improved instance-specific algorithm configuration". In: *Artificial Intelligence* 235 (2016), pp. 26–39. ISSN: 0004-3702.

[8] C. Ansótegui, J. Pon, and M. Sellmann. "Boosting evolutionary algorithm configuration". In: *Annals of Mathematics and Artificial Intelligence* 90 (7 2022), pp. 715–734.

[9] C. Aranha, C. L. C. Villalón, F. Campelo, M. Dorigo, R. Ruiz, M. Sevaux, K. Sörensen, and T. Stützle. "Metaphor-based metaheuristics, a call for action: the elephant in the room". In: *Swarm Intelligence* 16.1 (2022), pp. 1–6. ISSN: 1935-3820. URL: https://doi.org/10.1007/s11721-021-00202-9.

[10] A. Ashe, V. Colot, and B. P. Oldroyd. "How does epigenetics influence the course of evolution?" In: *Philos Trans R Soc Lond B Biol Sci* 376 (1826 2021).

[11] T. Bäck, D. B. Fogel, and Z. Michalewicz. *Handbook of Evolutionary Computation.* 1st. GBR: IOP Publishing Ltd., 1997. ISBN: 0750303921.

[12] J. Bader. "Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods". PhD thesis. ETH Zurich, Jan. 2010.

[13] J. Bader and E. Zitzler. "Hype: An algorithm for fast hypervolume-based many-objective optimization". In: *Evol. Comput.* 19.1 (Mar. 2011), pp. 45–76. ISSN: 1063-6560. URL: `https://doi.org/10.1162/EVCO_a_00009`.

[14] R. Baragona and F. Battaglia. "Evolutionary Computing in Statistical Data Analysis". In: *Foundations of Computational Intelligence Volume 3: Global Optimization.* Ed. by A. Abraham, A.-E. Hassanien, P. Siarry, and A. Engelbrecht. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 347–386. ISBN: 978-3-642-01085-9.

[15] W. Bateson, G. Mendel, and W. M. Wheeler. *Mendel's principles of heredity; a defence.* Cambridge,University press, 1902, p. 236.

[16] A. Benítez-Hidalgo, A. J. Nebro, J. García-Nieto, I. Oregi, and J. D. Ser. "jMetalPy: A Python framework for multi-objective optimization with metaheuristics". In: *Swarm and Evolutionary Computation* (2019), p. 100598. ISSN: 2210-6502.

[17] B. E. Bernstein, A. Meissner, and E. S. Lander. "The Mammalian Epigenome". In: *Cell* 128.4 (2007), pp. 669–681. ISSN: 0092-8674.

[18] H.-G. Beyer and H.-P. Schwefel. "Evolution strategies – A comprehensive introduction". In: *Natural Computing* 1 (1 2002), pp. 3–52.

[19] S. Biswas and C. M. Rao. "Epigenetic tools (The Writers, The Readers and The Erasers) and their implications in cancer therapy." In: *European Journal of Pharmacology* 837 (2018), pp. 8–24.

[20] S. Biswas, S. Das, P. Suganthan, and C. Coello. "Evolutionary multiobjective optimization in dynamic environments: A set of novel benchmark functions". In: *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014.* July 2014, pp. 3192–3199.

[21] D. Bolnick, P. Amarasekare, M. Araújo, R. Bürger, J. Levine, M. Novak, V. Rudolf, S. Schreiber, M. Urban, and D. Vasseur. "Why intraspecific trait variation matters in community ecology". In: *Trends in ecology & evolution* 26 (Feb. 2011), pp. 183–92.

[22] E. Bonabeau, M. Dorigo, and G. Theraulaz. "Swarm Intelligence : From Natural to Artificial Systems". In: *Oxford University press* (1999).

[23] C. T. Brown, L. S. Liebovitch, and R. Glendon. "Lévy Flights in Dobe Ju'hoansi Foraging Patterns". In: *Human Ecology* 35 (1 2007), pp. 129–138.

[24] R. L. Carroll. "Towards a new evolutionary synthesis". In: *Trends in Ecology & Evolution* 15.1 (2000), pp. 27–32. ISSN: 0169-5347.

[25] F. A. Champagne. “Epigenetic mechanisms and the transgenerational effects of maternal care”. In: *Front Neuroendocrinol* 29 (3 2008), pp. 386, 397.

[26] H. S. Chandra and V. Nanjundiah. “The evolution of genomic imprinting”. In: *Development* 108.Supplement (Apr. 1990), pp. 47–53.

[27] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff. “A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization”. In: *IEEE Transactions on Evolutionary Computation* 20.5 (2016), pp. 773–791.

[28] T. Chiang and Y. Lai. “MOEA/D-AMS: Improving MOEA/D by an adaptive mating selection mechanism”. In: *2011 IEEE Congress of Evolutionary Computation (CEC)*. 2011, pp. 1473–1480.

[29] O. Chikumbo, E. Goodman, and K. Deb. “Approximating a multi-dimensional Pareto front for a land use management problem: A modified MOEA with an epigenetic silencing metaphor”. In: *2012 IEEE Congress on Evolutionary Computation*. 2012, pp. 1–9.

[30] K. Chrominski, M. Tkacz, and M. Boryczka. “Epigenetic Modification of Genetic Algorithm”. In: *Computational Science – ICCS 2020*. Cham: Springer International Publishing, 2020, pp. 267–278. ISBN: 978-3-030-50417-5.

[31] G. Ciro, F. Dugardin, F. Yalaoui, and R. Kelly. “A NSGA-II and NSGA-III comparison for solving an open shop scheduling problem with resource constraints”. In: *IFAC-PapersOnLine* 49 (Dec. 2016), pp. 1272–1277.

[32] Z. Cui, M. Zhang, H. Wang, X. Cai, and W. Zhang. “A hybrid many-objective cuckoo search algorithm”. In: *Soft Computing* 23.21 (Nov. 2019), pp. 10681–10697. ISSN: 1433-7479. URL: https://doi.org/10.1007/s00500-019-04004-4.

[33] F. Da Ros, L. Di Gaspero, D. La Barbera, V. Della Mea, K. Roitero, L. Deroma, S. Licata, and F. Valent. “A Multi-objective Biased Random-Key Genetic Algorithm for the Siting of Emergency Vehicles”. In: *Metaheuristics*. Ed. by L. Di Gaspero, P. Festa, A. Nakib, and M. Pavone. Cham: Springer International Publishing, 2023, pp. 449–456. ISBN: 978-3-031-26504-4.

[34] C. Dai and Y. Wang. “A new decomposition based evolutionary algorithm with uniform designs for many-objective optimization”. In: *Applied Soft Computing* 30 (2015), pp. 238–248. ISSN: 1568-4946.

[35] É. Danchin. “Avatars of information: towards an inclusive evolutionary synthesis”. In: *Trends in Ecology & Evolution* 28.6 (2013), pp. 351–358. ISSN: 0169-5347.

[36] E. Danchin, A. Charmantier, F. A. Champagne, A. Mesoudi, B. Pujol, and S. Blanchet. “Beyond DNA: integrating inclusive inheritance into an extended theory of evolution”. In: *Nature Reviews Genetics* 12 (7 2011), pp. 475–486.

[37] E. Danchin and R. H. Wagner. “Inclusive heritability: combining genetic and non-genetic information to study animal behavior and culture”. In: *Oikos* 119.2 (2010), pp. 210–218.

[38] C. Darwin. *On the Origin of Species by Means of Natural Selection.* London: Murray, 1859.

[39] C. Darwin. *The Formation of Vegetable Mould Through the Action of Worms, with Observations on their Habits.* John Murray, London, 1881.

[40] C. Darwin and A. Wallace. “On the Tendency of Species to form Varieties; and on the Perpetuation of Varieties and Species by Natural Means of Selection.” In: *Journal of the Proceedings of the Linnean Society of London. Zoology* 3.9 (1858), pp. 45–62.

[41] R. Dawkins. *The Selfish Gene.* Oxford University Press, Oxford, UK, 1976.

[42] K. Deb and R. B. Agrawal. “Simulated Binary Crossover for Continuous Search Space”. In: *Complex Systems* 9 (1995).

[43] K. Deb and H. Jain. “An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints”. In: *IEEE Transactions on Evolutionary Computation* 18.4 (2014), pp. 577–601.

[44] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* USA: John Wiley & Sons, Inc., 2001. ISBN: 047187339X.

[45] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *IEEE Transactions on Evolutionary Computation* 6.2 (Apr. 2002), pp. 182–197. ISSN: 1089778X.

[46] 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.* Ed. by S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 803–817.

[47] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. “Scalable multi-objective optimization test problems”. In: *Proc. of 2002 IEEE Congress on Evolutionary Computation* 1 (June 2002), pp. 825–830.

[48] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. “Scalable Test Problems for Evolutionary Multiobjective Optimization”. In: *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications.* Ed. by A. Abraham, L. Jain, and R. Goldberg. London: Springer London, 2005, pp. 105–145. ISBN: 978-1-84628-137-2.

[49] J. Del Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanz, D. Camacho, S. Das, P. N. Suganthan, C. A. Coello Coello, and F. Herrera. “Bio-inspired computation:

Where we stand and what's next". In: *Swarm and Evolutionary Computation* 48 (2019), pp. 220–250. ISSN: 2210-6502.

[50] D. C. Dolinoy. "The agouti mouse model: an epigenetic biosensor for nutritional and environmental alterations on the fetal epigenome". In: *Nutrition reviews* 66 Suppl 1 (2008), S7–S11.

[51] M. Dorigo, M. Birattari, and T. Stützle. "Ant Colony Optimization". In: *Computational Intelligence Magazine, IEEE* 1 (Dec. 2006), pp. 28–39.

[52] M. Doumpos and C. Zopounidis. "Multi-objective optimization models in finance and investments". In: *Journal of Global Optimization* 76.2 (2020), pp. 243–244.

[53] J. J. Durillo, J. García-Nieto, A. J. Nebro, C. A. Coello Coello, F. Luna, and E. Alba. "Multi-objective particle swarm optimizers: An experimental comparison". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* Vol. 5467 LNCS. Springer, Berlin, Heidelberg, 2010, pp. 495–509. ISBN: 3642010199.

[54] D. H. Erwin. "Macroevolution of ecosystem engineering, niche construction and diversity". In: *Trends in Ecology & Evolution* 23.6 (2008), pp. 304–310. ISSN: 0169-5347.

[55] T. H. G. Ezard, R. Prizak, and R. B. Hoyle. "The fitness costs of adaptation via phenotypic plasticity and maternal effects". In: *Functional Ecology* 28.3 (2014), pp. 693–701. ISSN: 02698463, 13652435.

[56] J. Falcón-Cardona and C. Coello. "iMOACO $_{\mathbb{R}}$: A New Indicator-Based Multi-objective Ant Colony Optimization Algorithm for Continuous Search Spaces". In: *Parallel Problem Solving from Nature.* Vol. 9921. Sept. 2016, pp. 389–398. ISBN: 978-3-319-45822-9.

[57] Z. Fan, H. Li, Caimin Wei, W. Li, Han Huang, X. Cai, and Z. Cai. "An improved epsilon constraint handling method embedded in MOEA/D for constrained multi-objective optimization problems". In: *2016 IEEE Symposium Series on Computational Intelligence (SSCI).* 2016, pp. 1–8.

[58] Z. Fan, Y. Fang, W. Li, X. Cai, C. Wei, and E. Goodman. "MOEA/D with angle-based constrained dominance principle for constrained multi-objective optimization problems". In: *Applied Soft Computing* 74 (2019), pp. 621–633. ISSN: 1568-4946.

[59] Z. Fan, W. Li, X. Cai, H. Li, K. Hu, Q. Zhang, K. Deb, and E. D. Goodman. "Difficulty Adjustable and Scalable Constrained Multi-objective Test Problem Toolkit". In: *CoRR* abs/1612.07603 (2016). arXiv: 1612.07603.

[60] M. Farina, K. Deb, and P. Amato. "Dynamic multiobjective optimization problems: test cases, approximations, and applications". In: *IEEE Transactions on Evolutionary Computation* 8.5 (2004), pp. 425–442.

[61] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili. "Grey wolf optimizer: a review of recent variants and applications". In: *Neural Computing and Applications* 30.2 (July 1, 2018), pp. 413–435.

[62] R. A. Fisher. "XV.—The Correlation between Relatives on the Supposition of Mendelian Inheritance." In: *Transactions of the Royal Society of Edinburgh* 52.2 (1919), pp. 399–433.

[63] P. J. Fleming and A. J. Chipperfield. "Evolutionary Algorithms for Multiple Criteria Decision Making in Control". In: *IFAC Proceedings Volumes* 31.13 (1998). IFAC Workshop on Nonsmooth and Discontinuous Problems of Control, Optimization and Applications (NDPCO'98), Chelyabinsk, Russia, 17-20 June 1998, pp. 227–234. ISSN: 1474-6670.

[64] D. Futuyama. *Evolutionary Biology.* Sinauer Associates, 2006.

[65] Y. Gajpal and P. Abad. "Multi-ant colony system (MACS) for a vehicle routing problem with backhauls". In: *European Journal of Operational Research* 196.1 (2009), pp. 102–117. ISSN: 0377-2217. URL: `https://www.sciencedirect.com/science/article/pii/S0377221708002300`.

[66] J. Gerhart and M. Kirschner. "The theory of facilitated variation". In: *Proceedings of the National Academy of Sciences* 104.suppl 1 (2007), pp. 8582–8589.

[67] A. Godínez, L. Espinosa, and E. Mezura-Montes. "An Experimental Comparison of Multiobjective Algorithms: NSGA-II and OMOPSO". In: *2010 IEEE Electronics, Robotics and Automotive Mechanics Conference.* Nov. 2010, pp. 28–33.

[68] C. Goh and K. C. Tan. "A Competitive-Cooperative Coevolutionary Paradigm for Dynamic Multiobjective Optimization". In: *IEEE Transactions on Evolutionary Computation* 13.1 (2009), pp. 103–127.

[69] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* 1st. USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN: 0201157675.

[70] C. J. Goodnight and L. Stevens. "Experimental Studies of Group Selection: What Do They Tell US About Group Selection in Nature?" In: *The American Naturalist* 150.S1 (1997), s59–s79. ISSN: 00030147, 15375323. (Visited on 03/20/2023).

[71] J. I. Griffiths, D. Z. Childs, R. D. Bassar, T. Coulson, D. N. Reznick, and M. Rees. "Individual differences determine the strength of ecological interactions". In: *Proceedings of the National Academy of Sciences* (2020). ISSN: 0027-8424.

[72] P. A. Grudniewski and A. J. Sobey. "Behaviour of Multi-Level Selection Genetic Algorithm (MLSGA) using different individual-level selection mechanisms". In: *Swarm and Evolutionary Computation* 44 (2019), pp. 852–862. ISSN: 2210-6502.

[73] P. A. Grudniewski and A. J. Sobey. “Benchmarking the performance of genetic algorithms on constrained dynamic problems”. In: *Natural Computing* 21 (2022), pp. 109–125.

[74] P. A. Grudniewski and A. J. Sobey. “cMLSGA: co-evolutionary Multi-Level Selection Genetic Algorithm”. In: *Swarm and Evolutionary Computation* (2019).

[75] P. A. Grudniewski and A. J. Sobey. “Do general Genetic Algorithms provide benefits when solving real problems?” In: *2019 IEEE Congress on Evolutionary Computation (CEC)* (2019), pp. 1822–1829.

[76] P. A. Grudniewski. “Improving the applicability of genetic algorithms to real problems”. PhD thesis. 2021.

[77] R. Al-Haddad, N. Karnib, R. A. Assaad, Y. Bilen, N. Emmanuel, A. Ghanem, J. Younes, V. Zibara, J. S. Stephan, and S. F. Sleiman. “Epigenetic changes in diabetes”. In: *Neuroscience Letters* 625 (2016), pp. 64–69.

[78] D. Haig. “The Kinship Theory of Genomic Imprinting”. In: *Annual Review of Ecology and Systematics* 31.1 (2000), pp. 9–32.

[79] J. B. S. Haldane. *The causes of evolution.* Macmillan, 1932.

[80] N. Hansen, S. D. Müller, and P. Koumoutsakos. “Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES)”. In: *Evolutionary Computation* 11.1 (2003), pp. 1–18.

[81] K. Harada, S. Hiwa, and T. Hiroyasu. “Adaptive weight vector assignment method for MOEA/D”. In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI).* 2017, pp. 1–9.

[82] G. Harik, F. Lobo, and D. Goldberg. “The Compact Genetic Algorithm”. In: *Evolutionary Computation, IEEE Transactions on* 3 (Dec. 1999), pp. 287–297.

[83] M. Hauschild and M. Pelikan. “An introduction and survey of estimation of distribution algorithms”. In: *Swarm and evolutionary computation* 1.3 (2011), pp. 111–128.

[84] J. Holland. “Adaptive plans optimal for payoff-only environments”. In: *Second Hawaii Conference on Systems Sciences.* 1969.

[85] J. Holland. *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: University of Michigan Press, 1975.

[86] D. Houle. “Comparing evolvability and variability of quantitative traits”. In: *Genetics* 130 (1992), pp. 195–204.

[87] R. Hoyle and T. H. G. Ezard. “The benefits of maternal effects in novel and stable environments”. In: *Journal of the Royal Society, Interface / the Royal Society* 9 (May 2012), pp. 2403–13.

[88] S. Huband, P. Hingston, L. Barone, and L. While. "A review of multiobjective test problems and a scalable test problem toolkit". In: *Evolutionary Computation, IEEE Transactions on* 10 (Nov. 2006), pp. 477–506.

[89] J. Huxley. New York ; London: Harper & brothers, 1943.

[90] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima. "Performance comparison of NSGA-II and NSGA-III on various many-objective test problems". In: *2016 IEEE Congress on Evolutionary Computation (CEC)*. 2016, pp. 3045–3052.

[91] G. J. and K. M. *The plausibility of life: resolving Darwin's Dilemma*. New Haven, CT: Yale University Press, 2005.

[92] B. P. Z. Jack Bosworth Norman Foo. "Comparison of genetic algorithms with conjugate gradient methods". In: *NASA Contractor Reports* (Aug. 1972).

[93] M. A. Jan and Q. Zhang. "MOEA/D for constrained multiobjective optimization: Some preliminary experimental results". In: *2010 UK Workshop on Computational Intelligence (UKCI)*. 2010, pp. 1–6.

[94] S. Jiang, S. Yang, Y. Wang, and X. Liu. "Scalarizing Functions in Decomposition-Based Multiobjective Evolutionary Algorithms". In: *IEEE Transactions on Evolutionary Computation* 22.2 (2018), pp. 296–313.

[95] S. Jiang, M. Kaiser, S. Yang, S. Kollias, and N. Krasnogor. "A Scalable Test Suite for Continuous Dynamic Multiobjective Optimization". In: *IEEE Transactions on Cybernetics* 50.6 (2020), pp. 2814–2826.

[96] S. Jiang and S. Yang. "Evolutionary Dynamic Multiobjective Optimization: Benchmarks and Algorithm Comparisons". In: *IEEE Transactions on Cybernetics* 47.1 (2017), pp. 198–211.

[97] B. M. Jones and G. E. Robinson. "Genetic accommodation and the role of ancestral plasticity in the evolution of insect eusociality". In: *Journal of Experimental Biology* 221.23 (2018). ISSN: 0022-0949.

[98] D. Karaboga and B. Basturk. "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm". In: *Journal of Global Optimization* 39 (3 2007), pp. 459–471. ISSN: 1573-2916.

[99] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga. "A comprehensive survey: artificial bee colony (ABC) algorithm and applications". In: *Artificial Intelligence Review* 42 (1 2014), pp. 21–57.

[100] J. Kennedy and R. Eberhart. "Particle swarm optimization". In: *Giornale di Malattie Infettive e Parassitarie* 46.10 (1995), pp. 1942–1948. ISSN: 19353812. eprint: 9780201398298.

[101] S. Khan, P. Grudniewski, Y. S. Muhammad, and A. J. Sobey. "The benefits of co-evolutionary Genetic Algorithms in voyage optimisation". In: *Ocean Engineering*

245 (2022), p. 110261. ISSN: 0029-8018. URL: `https://www.sciencedirect.com/science/article/pii/S0029801821015705`.

[102] M. Kirschner and J. Gerhart. “Evolvability”. In: *Proceedings of the National Academy of Sciences of the United States of America* 95 (15 July 1998), pp. 8420–8427.

[103] H. Kitano. “Designing Neural Networks Using Genetic Algorithms with Graph Generation System”. In: *Complex Syst.* 4 (1990).

[104] J. Knowles and D. Corne. “M-PAES: a memetic algorithm for multiobjective optimization”. In: *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512).* Vol. 1. 2000, 325–332 vol.1.

[105] B. Kuijper and R. A. Johnstone. “Parental effects and the evolution of phenotypic memory”. In: *Journal of Evolutionary Biology* 29.2 (2016), pp. 265–276.

[106] B. Kuijper, R. Johnstone, and S. Townley. “The Evolution of Multivariate Maternal Effects”. In: *PLoS computational biology* 10 (Apr. 2014), e1003550.

[107] K. Laland, B. Matthews, and M. W. Feldman. “An introduction to niche construction theory”. In: *Evolutionary Ecology* 30 (2 2016), pp. 191–202.

[108] R. Lande and S. Shannon. “The Role of Genetic Variation in Adaptation and Population Persistence in a Changing Environment”. In: *Evolution* 50.1 (1996), pp. 434–437. ISSN: 00143820, 15585646.

[109] M. D. Laubichler and J. Maienschein. *From Embryology to Evo-Devo: A History of Developmental Evolution.* Cambridge: MIT Press, 2007.

[110] T. Lenaerts, A. Defaweux, and B. Manderick. “Modeling Artificial Multilevel Selection”. In: *AAAI Spring Symposium on Computational Synthesis* (2003).

[111] F. Leung, H. Lam, S. Ling, and P. Tam. “Tuning of the structure and parameters of a neural network using an improved genetic algorithm”. In: *IEEE Transactions on Neural Networks* 14.1 (2003), pp. 79–88.

[112] R. C. Lewontin. “The Units of Selection”. In: *Annual Review of Ecology and Systematics* 1 (1970), pp. 1–18. ISSN: 00664162. (Visited on 03/20/2023).

[113] H. Li, M. Ding, J. Deng, and Q. Zhang. “On the use of random weights in MOEA/D”. In: *2015 IEEE Congress on Evolutionary Computation (CEC).* 2015, pp. 978–985.

[114] H. Li and Q. Zhang. “Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II”. In: *IEEE Transactions on Evolutionary Computation* 13.2 (2009), pp. 284–302.

[115] M. Li, S. Yang, and X. Liu. “Pareto or Non-Pareto: Bi-Criterion Evolution in Multiobjective Optimization”. In: *IEEE Transactions on Evolutionary Computation* 20.5 (2016), pp. 645–665.

[116] Q. Lin, J. Chen, Z.-H. Zhan, W.-n. Chen, C. Coello, Y. Yin, C.-M. Lin, and J. Zhang. "A Hybrid Evolutionary Immune Algorithm for Multiobjective Optimization Problems". In: *IEEE Transactions on Evolutionary Computation* 20 (Jan. 2015), pp. 1–1.

[117] M. Lindauer, K. Eggensperger, M. Feurer, A. Biedenkapp, D. Deng, C. Benjamins, T. Ruhkopf, R. Sass, and F. Hutter. "SMAC3: A Versatile Bayesian Optimization Package for Hyperparameter Optimization". In: *Journal of Machine Learning Research* 23.54 (2022), pp. 1–9. URL: http://jmlr.org/papers/v23/21-0888.html.

[118] D. Liu, K. Tan, C.-K. Goh, and W. Ho. "A Multiobjective Memetic Algorithm Based on Particle Swarm Optimization". In: *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society* 37 (Mar. 2007), pp. 42–50.

[119] H. Liu, L. Chen, K. Deb, and E. D. Goodman. "Investigating the Effect of Imbalance Between Convergence and Diversity in Evolutionary Multiobjective Algorithms". In: *IEEE Transactions on Evolutionary Computation* 21.3 (2017), pp. 408–425.

[120] H. Liu, F. Gu, and Q. Zhang. "Decomposition of a Multiobjective Optimization Problem Into a Number of Simple Multiobjective Subproblems". In: *IEEE Transactions on Evolutionary Computation* 18.3 (2014), pp. 450–455.

[121] Q. Liu, X. Liu, J. Wu, and Y. Li. "An Improved NSGA-III Algorithm Using Genetic K-Means Clustering Algorithm". In: *IEEE Access* 7 (2019), pp. 185239–185249.

[122] M. López-Ibáñez, J. Dubois-Lacoste, L. Pérez Cáceres, M. Birattari, and T. Stützle. "The irace package: Iterated racing for automatic algorithm configuration". In: *Operations Research Perspectives* 3 (2016), pp. 43–58. ISSN: 2214-7160.

[123] X. Ma, Q. Zhang, G. Tian, J. Yang, and Z. Zhu. "On Tchebycheff Decomposition Approaches for Multiobjective Evolutionary Optimization". In: *IEEE Transactions on Evolutionary Computation* 22.2 (2018), pp. 226–244.

[124] R. Mantovani, T. Horváth, R. Cerri, J. Vanschoren, and A. de Carvalho. "Hyperparameter tuning of a decision tree induction algorithm". English. In: *5th Brazilian Conference on Intelligent Systems, BRACIS 2016; Recife, Pernambuco; Brazil; 9 October 2016 through 12 October 2016.* 5th Brazilian Conference on Intelligent System - BRACIS 2016, At Recife, Pernambuco, Brazil, BRACIS 2016 ; Conference date: 09-10-2016 Through 12-10-2016. United States: Institute of Electrical and Electronics Engineers, Feb. 2017, pp. 37–42.

[125] S. Martínez-Jaramillo, A. García-Almanza, B. Alexandrova-Kabadjova, and T. P. Centeno. "Evolutionary Computation in Finance". In: *Encyclopedia of Machine Learning.* Ed. by C. Sammut and G. I. Webb. Boston, MA: Springer US, 2010, pp. 344–353. ISBN: 978-0-387-30164-8.

[126] J. Maynard-Smith and E. Szathmary. *The Major Transitions in Evolution.* Oxford: Oxford University Press, 1995.

[127] A. Mesoudi. “Cultural Evolution”. In: American Cancer Society, 2018, pp. 1–8. ISBN: 9780470015902.

[128] T. Miconi, K. Stanley, and J. Clune. “Differentiable plasticity: training plastic neural networks with backpropagation”. In: *Proceedings of the 35th International Conference on Machine Learning.* Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 3559–3568. URL: `https://proceedings.mlr.press/v80/miconi18a.html`.

[129] S. Mirjalili, S. M. Mirjalili, and A. Lewis. “Grey Wolf Optimizer”. In: *Advances in Engineering Software* 69 (2014), pp. 46–61. ISSN: 0965-9978.

[130] S. Mirjalili, S. Saremi, S. M. Mirjalili, and L. dos S. Coelho. “Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization”. In: *Expert Systems with Applications* 47 (2016), pp. 106–119. ISSN: 0957-4174. URL: `https://www.sciencedirect.com/science/article/pii/S0957417415007435`.

[131] K. L. Moran, Y. Shlyakhtina, and M. M. Portal. “The role of non-genetic information in evolutionary frameworks”. In: *Critical Reviews in Biochemistry and Molecular Biology* 56.3 (2021), pp. 255–283.

[132] M. Mosayebi and M. Sodhi. “Tuning Genetic Algorithm Parameters Using Design of Experiments”. In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion.* GECCO ’20. Cancún, Mexico: Association for Computing Machinery, 2020, pp. 1937–1944. ISBN: 9781450371278. URL: `https://doi.org/10.1145/3377929.3398136`.

[133] W. Muir. “Group Selection for Adaptation to Multiple-Hen Cages: Selection Program and Direct Responses1,2”. In: *Poultry Science* 75.4 (1996), pp. 447–458. ISSN: 0032-5791.

[134] G. B. Müller. “Evo–devo: extending the evolutionary synthesis”. In: *Nature Reviews Genetics* 8 (12 2007), pp. 943–949.

[135] D. Navarro-Mateu and A. Cocho-Bermejo. “Evo-Devo Algorithms: Gene-Regulation for Digital Architecture”. In: *Biomimetics (Basel)* 4 (3 Aug. 2019).

[136] A. J. Nebro, J. J. Durillo, G. Nieto, C. A. Coello Coello, F. Luna, and E. Alba. “SMPSO: A new pso-based metaheuristic for multi-objective optimization”. In: *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, MCDM 2009 - Proceedings.* 2009, pp. 66–73. ISBN: 9781424427642.

[137] P. Netrapalli. “Stochastic gradient descent and its variants in machine learning”. In: *Journal of the Indian Institute of Science* 99.2 (2019), pp. 201–213.

[138] M. A. Nowak, C. E. Tarnita, and E. O. Wilson. “The evolution of eusociality”. In: *Nature* 466 (7310 2010), pp. 1057–1062.

[139] F. J. Odling-Smee, K. N. Laland, and M. W. Feldman. *Niche Construction: The Neglected Process in Evolution (MPB-37)*. Princeton University Press, 2003. ISBN: 0691044376.

[140] V. Oduguwa, A. Tiwari, and R. Roy. “Evolutionary computing in manufacturing industry: an overview of recent applications”. In: *Applied Soft Computing* 5.3 (2005). Application Reviews, pp. 281–299. ISSN: 1568-4946.

[141] B. P. Oldroyd, R. J. Reid, A. Ashe, and E. J. Remnant. “Honey Bees, Royal Jelly, Epigenetics”. In: *Encyclopedia of Reproduction (Second Edition)*. Ed. by M. K. Skinner. Second Edition. Oxford: Academic Press, 2018, pp. 722–727. ISBN: 978-0-12-815145-7.

[142] C. Ozturk, E. Hancer, and D. Karaboga. “A novel binary artificial bee colony algorithm based on genetic operators”. In: *Information Sciences* 297 (2015), pp. 154–170. ISSN: 0020-0255.

[143] K. E. Parsopoulos and M. N. Vrahatis. “Particle swarm optimization method in multiobjective problems”. In: *Proceedings of the 2002 ACM Symposium on Applied Computing*. SAC ’02. Madrid, Spain: Association for Computing Machinery, 2002, pp. 603–607. ISBN: 1581134452. URL: https://doi.org/10.1145/508791.508907.

[144] K. Parsopoulos and M. Vrahatis. “Particle swarm optimization method in multiobjective problems”. In: *Proceedings of the ACM Symposium on Applied Computing* (Mar. 2002).

[145] M. Pelikan and H. Mühlenbein. “Marginal distributions in evolutionary algorithms”. In: *Proceedings of the international conference on genetic algorithms mendel*. Vol. 98. Citeseer. 1998, pp. 90–95.

[146] S. Periyasamy, A. Gray, and P. Kille. “The Epigenetic Algorithm”. In: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. 2008, pp. 3228–3236.

[147] M. Pigliucci. “Is evolvability evolvable?” In: *Nature Reviews Genetics* 9 (1 2008), pp. 75–82.

[148] M. Pigliucci. “Do We Need an Extended Evolutionary Synthesis?” In: *Evolution* 61.12 (2007), pp. 2743–2749.

[149] M. Pigliucci and G. Muller. *Evolution – the Extended Synthesis*. MIT Press, 2010.

[150] A. Piotrowski, M. Napiórkowski, J. Napiórkowski, and P. Rowiński. “Swarm Intelligence and Evolutionary Algorithms: Performance versus speed”. In: *Information Sciences* 384 (Dec. 2016).

[151] R. H. Plasterk. “RNA silencing: the genome’s immune system”. In: *Science* 296 (2002), pp. 1263–1265.

[152] S. Pllana, S. Memeti, and J. Kolodziej. “Customizing Pareto Simulated Annealing for Multi-objective Optimization of Control Cabinet Layout”. In: *CoRR* abs/1906.04825 (2019). arXiv: `1906.04825`.

[153] M. A. Potter and K. A. De Jong. “A cooperative coevolutionary approach to function optimization”. In: *Parallel Problem Solving from Nature — PPSN III.* Ed. by Y. Davidor, H.-P. Schwefel, and R. Männer. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 249–257. ISBN: 978-3-540-49001-2.

[154] T. Price, A. Qvarnström, and D. Irwin. “The role of phenotypic plasticity in driving genetic evolution”. In: *Proceedings. Biological sciences / The Royal Society* 270 (Aug. 2003), pp. 1433–40.

[155] S. R. Proulx and H. Teotónio. “What Kind of Maternal Effects Can Be Selected For in Fluctuating Environments?” In: *The American Naturalist* 189.6 (2017). PMID: 28514627, E118–E137.

[156] W. Reik. “Stability and flexibility of epigenetic gene regulation in mammalian development”. In: *Nature* 447 (7143 2007), pp. 425–432.

[157] A. Reza and K. Ziarati. “A multilevel evolutionary algorithm for optimizing numerical functions”. In: *International Journal of Industrial Engineering Computations* 2 (Apr. 2011).

[158] N. Riquelme, C. Von Lücken, and B. Baran. “Performance metrics in multi-objective optimization”. In: *2015 Latin American Computing Conference (CLEI).* 2015, pp. 1–11.

[159] V. Roberge, M. Tarbouchi, and G. Labonte. “Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning”. In: *Industrial Informatics, IEEE Transactions on* 9 (Feb. 2013), pp. 132–141.

[160] C. Rollo. *Phenotypes: Their epigenetics, ecology and evolution.* New York: Chapman and Hall, 1994.

[161] I. Roman, J. Ceberio, A. Mendiburu, and J. A. Lozano. “Bayesian optimization for parameter tuning in evolutionary algorithms”. In: *2016 IEEE Congress on Evolutionary Computation (CEC).* 2016, pp. 4839–4845.

[162] C. D. Rosin and R. K. Belew. “New Methods for Competitive Coevolution”. In: *Evol. Comput.* 5.1 (Mar. 1997), pp. 1–29. ISSN: 1063-6560.

[163] S. Sanders and C. G. Giraud-Carrier. “Informing the Use of Hyperparameter Optimization Through Metalearning”. In: *2017 IEEE International Conference on Data Mining (ICDM)* (2017), pp. 1051–1056.

[164] R. Santer. “Cellular Mechanisms of Aging”. In: *Brocklehurst’s Textbook of Geriatric Medicine and Gerontology (Seventh Edition)*. Ed. by H. M. Fillit, K. Rockwood, and K. Woodhouse. Seventh Edition. Philadelphia: W.B. Saunders, 2010, pp. 42–50. ISBN: 978-1-4160-6231-8.

[165] H. Sato, H. E. Aguirre, and K. Tanaka. “Local dominance using polar coordinates to enhance multiobjective evolutionary algorithms”. In: *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)* 1 (2004), 188–195 Vol.1.

[166] H. Seada and K. Deb. “U-NSGA-III: A Unified Evolutionary Optimization Procedure for Single, Multiple, and Many Objectives: Proof-of-Principle Results”. In: *Evolutionary Multi-Criterion Optimization*. Ed. by A. Gaspar-Cunha, C. Henggeler Antunes, and C. C. Coello. Cham: Springer International Publishing, 2015, pp. 34–49. ISBN: 978-3-319-15892-1.

[167] M. Shafiee, A. Berglund, E. Z. Berglund, E. D. Brill, and G. Mahinthakumar. “Evolutionary Computation-based Decision-making Framework for Designing Water Networks to Minimize Background Leakage”. In: *Procedia Engineering* 89 (2014). 16th Water Distribution System Analysis Conference, WDSA2014, pp. 118–125. ISSN: 1877-7058.

[168] M. Shehab, A. T. Khader, and M. A. Al-Betar. “A survey on applications and variants of the cuckoo search algorithm”. In: *Applied Soft Computing* 61 (2017), pp. 1041–1059. ISSN: 1568-4946. URL: `https://www.sciencedirect.com/science/article/pii/S1568494617301278`.

[169] C. E. Sheppard, R. Inger, R. A. McDonald, S. Barker, A. L. Jackson, F. J. Thompson, E. I. K. Vitikainen, M. A. Cant, and H. H. Marshall. “Intragroup competition predicts individual foraging specialisation in a group-living mammal”. In: *Ecology Letters* 21.5 (2018), pp. 665–673.

[170] Y. Shi and R. Eberhart. “A modified particle swarm optimizer”. In: *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*. 1998, pp. 69–73.

[171] O. M. Shir. “Niching in Evolutionary Algorithms”. In: *Handbook of Natural Computing*. Ed. by G. Rozenberg, T. Bäck, and J. N. Kok. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 1035–1069. ISBN: 978-3-540-92910-9. URL: `https://doi.org/10.1007/978-3-540-92910-9_32`.

[172] M. R. Sierra and C. A. Coello Coello. “Improving PSO-based Multi-Objective optimization using crowding, mutation and ϵ-dominance”. In: *Lecture Notes in Computer Science*. Vol. 3410. Springer, Berlin, Heidelberg, 2005, pp. 505–519.

[173] A. Soltoggio, K. O. Stanley, and S. Risi. "Born to learn: The inspiration, progress, and future of evolved plastic artificial neural networks". In: *Neural Networks* 108 (2018), pp. 48–67. ISSN: 0893-6080.

[174] K. Sörensen. "Metaheuristics – the metaphor exposed". In: *International Transactions of Operations Research* 22 (1 Jan. 2013), pp. 3–18.

[175] A. Starkey, H. Hagras, S. Shakya, and G. Owusu. "A comparison of particle swarm optimization and genetic algorithms for a multi-objective Type-2 fuzzy logic based system for the optimal allocation of mobile field engineers". In: July 2016, pp. 5068–5075.

[176] B. van Stein, H. Wang, and T. Bäck. "Automatic Configuration of Deep Neural Networks with Parallel Efficient Global Optimization". In: *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2019, pp. 1–7.

[177] T. Stewart, O. Bandte, H. Braun, N. Chakraborti, M. Ehrgott, M. Göbelt, Y. Jin, H. Nakayama, S. Poles, and D. Stefano. "Real-World Applications of Multiobjective Optimization". In: *Multiobjective Optimization: Interactive and Evolutionary Approaches* 5252 (Jan. 2008), pp. 285–327.

[178] D. H. Stolfi and E. Alba. "Epigenetic algorithms: A New way of building GAs based on epigenetics". In: *Information Sciences* 424 (2018), pp. 250–272. ISSN: 0020-0255.

[179] R. Storn and K. Price. "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces". In: *Journal of Global Optimization* 11 (Jan. 1997), pp. 341–359.

[180] R. Storn and K. Price. "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces". In: *Journal of Global Optimization* 11 (4 1997), pp. 341–359.

[181] A. Suppapitnarm, K. A. Seffen, G. T. Parks, and P. J. Clarkson. "A Simulated Annealing Algorithm for Multiobjective Optimization". In: *Engineering Optimization* 33.1 (2000), pp. 59–85.

[182] D. T. and B. R. "Intralocus sexual conflict can drive the evolution of genomic imprinting". In: *Genetics* 167 (2004), pp. 1537–1546.

[183] 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". In: *Computers & Operations Research* 40.6 (2013). Emergent Nature Inspired Algorithms for Multi-Objective Optimization, pp. 1648–1660. ISSN: 0305-0548.

[184] R. Tanabe and A. Fukunaga. "Success-history based parameter adaptation for Differential Evolution". In: *2013 IEEE Congress on Evolutionary Computation*. 2013, pp. 71–78.

[185] R. Tanabe and A. Oyama. “Benchmarking MOEAs for multi- and many-objective optimization using an unbounded external archive”. In: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO ’17. Berlin, Germany: Association for Computing Machinery, 2017, pp. 633–640. ISBN: 9781450349208. URL: `https://doi.org/10.1145/3071178.3079192`.

[186] L. Tani, D. Rand, C. Veelken, and M. Kadastik. “Evolutionary algorithms for hyperparameter optimization in machine learning for application in high energy physics”. In: *The European Physical Journal C* 81.2 (Feb. 2021).

[187] P. Tonelli and J.-B. Mouret. “Using a map-based encoding to evolve plastic neural networks”. In: *2011 IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS)*. 2011, pp. 9–16.

[188] A. M. Turing. “I.—COMPUTING MACHINERY AND INTELLIGENCE”. In: *Mind* LIX.236 (Oct. 1950), pp. 433–460. ISSN: 0026-4423.

[189] L. G. Valiant. “Evolvability”. In: *Mathematical Foundations of Computer Science 2007*. Ed. by L. Kučera and A. Kučera. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 22–43. ISBN: 978-3-540-74456-6.

[190] M. W. Vaughn, M. Tanurdžić, Z. Lippman, H. Jiang, R. Carrasquillo, P. D. Rabinowicz, N. Dedhia, W. R. McCombie, N. Agier, A. Bulski, V. Colot, R. Doerge, and R. A. Martienssen. “Epigenetic Natural Variation in Arabidopsis thaliana”. In: *PLOS Biology* 5.7 (June 2007), pp. 1–13.

[191] C. Violle, B. J. Enquist, B. J. McGill, L. Jiang, C. H. Albert, C. Hulshof, V. Jung, and J. Messier. “The return of the variance: intraspecific variability in community ecology”. In: *Trends in Ecology & Evolution* 27.4 (2012), pp. 244–252. ISSN: 0169-5347.

[192] M. J. Wade. “The Primary Characteristics of Tribolium Populations Group Selected for Increased and Decreased Population Size”. In: *Evolution* 33.2 (1979), pp. 749–764. ISSN: 00143820, 15585646. (Visited on 03/20/2023).

[193] G. P. Wagner and L. Altenberg. “Complex adaptations and the evolution of evolvability”. In: *Evolution* 50 (1996), pp. 967–976.

[194] L. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao. “Constrained Subproblems in a Decomposition-Based Multiobjective Evolutionary Algorithm”. In: *IEEE Transactions on Evolutionary Computation* 20.3 (2016), pp. 475–480.

[195] Z. Wang, Q. Zhang, M. Gong, and A. Zhou. “A replacement strategy for balancing convergence and diversity in MOEA/D”. In: *2014 IEEE Congress on Evolutionary Computation (CEC)*. 2014, pp. 2132–2139.

[196] M. West-Eberhard. “Phenotypic Plasticity”. In: *Encyclopedia of Ecology*. Ed. by S. E. Jørgensen and B. D. Fath. Oxford: Academic Press, 2008, pp. 2701–2707. ISBN: 978-0-08-045405-4.

[197] R. While, L. Bradstreet, and L. Barone. "A Fast Way of Calculating Exact Hypervolumes". In: *IEEE Trans. Evolutionary Computation* 16 (Feb. 2012), pp. 86–95.

[198] J. B. Wolf. and R. Hager. "A maternal-offspring coadaptation theory for the evolution of genomic imprinting". In: *PLoS Biology* 4 (2006), pp. 2238–2243.

[199] D. H. Wolpert and W. G. Macready. "No Free Lunch Theorems for Optimization". In: *Trans. Evol. Comp* 1.1 (Apr. 1997), pp. 67–82. ISSN: 1089-778X.

[200] A. J. Wood and R. J. Oakey. "Genomic imprinting in mammals: emerging themes and established theories". In: *PLoS Genetics* 2 (11 2006), e147.

[201] J. Wright. "Epigenetics: Reversible tags". In: *Nature* 498 (7455 2013), S10–S11.

[202] S. Wright. "Evolution in Mendelian Populations". In: 16 (2 1931), pp. 97–159.

[203] X. Hu and R. Eberhart. "Multiobjective optimization using dynamic neighborhood particle swarm optimization". In: *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600).* Vol. 2. 2002, 1677–1681 vol.2.

[204] X. Hu, R. C. Eberhart, and Yuhui Shi. "Particle swarm with extended memory for multiobjective optimization". In: *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706).* 2003, pp. 193–197.

[205] Q. Xu, Z. Xu, and T. Ma. "A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition: Variants, Challenges and Future Directions". In: *IEEE Access* 8 (2020), pp. 41588–41614.

[206] X. Yang and Suash Deb. "Cuckoo Search via Lévy flights". In: *2009 World Congress on Nature Biologically Inspired Computing (NaBIC).* 2009, pp. 210–214.

[207] X.-S. Yang. "Firefly Algorithms for Multimodal Optimization". In: *Stochastic Algorithms: Foundations and Applications.* Ed. by O. Watanabe and T. Zeugmann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 169–178.

[208] X.-S. Yang. *Nature-Inspired Metaheuristic Algorithms.* Luniver Press, July 2008.

[209] X.-S. Yang and S. Deb. "Multiobjective cuckoo search for design optimization". In: *Computers & Operations Research* 40.6 (2013). Emergent Nature Inspired Algorithms for Multi-Objective Optimization, pp. 1616–1624. ISSN: 0305-0548. URL: https://www.sciencedirect.com/science/article/pii/S0305054811002905.

[210] Y. Yang, G. Wu, J. Chen, and W. Dai. "Multi-objective optimization based on ant colony optimization in grid over optical burst switching networks". In: *Expert Systems with Applications* 37.2 (2010), pp. 1769–1775. ISSN: 0957-4174. URL: https://www.sciencedirect.com/science/article/pii/S0957417409006897.

[211] X. Yao. "Evolving artificial neural networks". In: *Proceedings of the IEEE* 87.9 (1999), pp. 1423–1447.

[212] F. Ye, C. Doerr, H. Wang, and T. Back. “Automated Configuration of Genetic Algorithms by Tuning for Anytime Performance”. In: *IEEE Transactions on Evolutionary Computation* 26.6 (Dec. 2022), pp. 1526–1538.

[213] S. R. Young, D. C. Rose, T. P. Karnowski, S.-H. Lim, and R. M. Patton. “Optimizing Deep Learning Hyper-Parameters through an Evolutionary Algorithm”. In: *Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments.* MLHPC ’15. Austin, Texas: Association for Computing Machinery, 2015. ISBN: 9781450340069. URL: `https://doi.org/10.1145/2834892.2834896`.

[214] X. Yu and X. Zhang. “Multiswarm comprehensive learning particle swarm optimization for solving multiobjective optimization problems”. In: *PLOS ONE* 12.2 (Feb. 2017). Ed. by W.-B. Du, e0172033. ISSN: 1932-6203.

[215] Y. Zeng and Y. Sun. “Comparison of multiobjective particle swarm optimization and evolutionary algorithms for optimal reactive power dispatch problem”. In: *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014* (July 2014), pp. 258–265.

[216] Z. H. Zhan, J. Li, J. Cao, J. Zhang, H. S. H. Chung, and Y. H. Shi. “Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems”. In: *IEEE Transactions on Cybernetics* 43.2 (Apr. 2013), pp. 445–463.

[217] Q. Zhang and H. Li. “MOEA/D: A multiobjective evolutionary algorithm based on decomposition”. In: *IEEE Transactions on Evolutionary Computation* 11.6 (Dec. 2007), pp. 712–731. ISSN: 1089778X.

[218] Q. Zhang, A. Zhou, S. Zhao, P. Suganthan, W. Liu, and S. Tiwari. “Multiobjective optimization Test Instances for the CEC 2009 Special Session and Competition”. In: *Mechanical Engineering* (Jan. 2008).

[219] Y. Zhang, R. Yang, J. Zuo, and X. Jing. “Enhancing MOEA/D with uniform population initialization, weight vector design and adjustment using uniform design”. In: *Journal of Systems Engineering and Electronics* 26.5 (2015), pp. 1010–1022.

[220] E. Zitzler, J. D. Knowles, and L. Thiele. “Quality Assessment of Pareto Set Approximations”. In: *Multiobjective Optimization.* 2008.

[221] E. Zitzler and L. Thiele. “Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach”. In: *IEEE Transactions on Evolutionary Computation* 3.4 (1999), pp. 257–271.

[222] E. Zitzler, K. Deb, and L. Thiele. “Comparison of Multiobjective Evolutionary Algorithms: Empirical Results”. In: *Evolutionary computation* 8 (Feb. 2000), pp. 173–95.

[223] E. Zitzler and S. Künzli. “Indicator-Based Selection in Multiobjective Search”. In: *Parallel Problem Solving from Nature - PPSN VIII*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 832–842. ISBN: 978-3-540-30217-9.

[224] I. B. Zovkic, M. C. Guzman-Karlsson, and J. D. Sweatt. “Epigenetic regulation of memory formation and maintenance”. In: *Learning & memory* 20 (2 2013), pp. 61–74.