



Faculty of Science and Engineering
the Manchester Metropolitan University

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS OF THE
MANCHESTER METROPOLITAN UNIVERSITY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Preference Focussed Many-Objective Evolutionary Computation

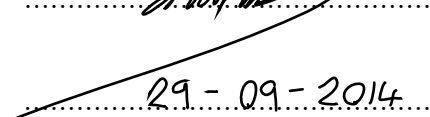
Doctoral Thesis of
Shahin Rostami

2014

Statement of Originality

Unless otherwise stated in the text, the work described in this thesis was carried out solely by the candidate. None of this work has already been accepted for any other degree, nor is it being concurrently submitted in candidature for any degree.

Signed 

Date 

This thesis is dedicated to
Keith Edwards

Acknowledgements

I would like to thank my first supervisor Dr Alex Shenfield, for giving me the opportunity to pursue my doctorate and for the very much appreciated guidance throughout the past few years. I would also like to thank Professor Nick Bowring for adopting me as his Ph.D. student and supporting me through the final hurdles of the thesis write-up and submission process.

I would like to thank my annual reviewer Dr Stephen Sigurnjak for keeping me on track and offering me support throughout the years of research.

I would like to thank those who were on the Ph.D. journey with me, Dean O'Reilly for offering an interesting collaboration between our research, and Peter Delves for the support we gave each other throughout the highs and the lows.

I would of course like to thank my parents for their love and support throughout my research.

I would like to thank Samuel Riddell, for keeping me sane.

Finally I would like to thank my lovely wife, Sam Rostami, for her love, understanding, support, and most of all her patience throughout my research and write-up.

Abstract

Solving complex real-world problems often involves the simultaneous optimisation of multiple conflicting performance criteria, these real-world problems occur in the fields of engineering, economics, chemistry, manufacturing, physics and many more. The optimisation process usually involves some design challenges in the form of the optimisation of a number of objectives and constraints. There exist many traditional optimisation methods (calculus based, random search, enumerative, etc...), however, these only offer a single solution in either adequate performance in a narrow problem domain or inadequate performance across a broad problem domain.

Evolutionary Multi-objective Optimisation (EMO) algorithms are robust optimisers which are suitable for solving complex real-world multi-objective optimisation problems, as they are able to address each of the conflicting objectives simultaneously. Typically, these EMO algorithms are run non-interactively with a Decision Maker (DM) setting the initial parameters of the algorithm and then analysing the results at the end of the optimisation process. When EMO is applied to real-world optimisation problems there is often a DM who is only interested in a portion of the Pareto-optimal front, however, incorporation of DM preferences is often neglected in the EMO literature.

In this thesis, the incorporation of DM preferences into EMO search methods has been explored. This has been achieved through the review of EMO literature to identify a powerful method of variation, Covariance Matrix Adaptation (CMA), and its computationally infeasible EMO implementation, MO-CMA-ES. A CMA driven EMO algorithm, CMA-PAES, capable of optimisation in the presence of many objectives has been developed, benchmarked, and statistically verified to outperform MO-CMA-ES and MOEA/D-DRA on selected test suites. CMA-PAES and MOEA/D-DRA with the incorporation of the novel Weighted Z-score (WZ) preference articulation operator (supporting *a priori*, *a posteriori* or *progressive* incorporation) are then benchmarked on a range of synthetic and real-world problems. WZ-CMA-PAES is then successfully applied to a real-world problem regarding the optimisation of a classifier for concealed weapon detection, outperforming previously published classifier implementations.

Abbreviations

ϵ -MOEA	Domination Based Multi-Objective Evolutionary Algorithm.
AGA	Adaptive Grid Algorithm.
AIS	Artificial Immune System.
ANN	Artificial Neural Network.
CI	Computational Intelligence.
CMA	Covariance Matrix Adaptation.
CMA-ES	Covariance Matrix Adaptation Evolutionary Strategy.
CMA-PAES	Covariance Matrix Adaptation Pareto Archived Evolution Strategy.
CMA-PAES-II	Covariance Matrix Adaptation Pareto Archived Evolution Strategy II.
CWD	Concealed Weapon Detection.
DEMO	Differential Evolution for Multi-objective Optimisation.
DEMO _w SA	Differential Evolution for Multi-objective Optimisation with Self Adaptation.
DM	Decision Maker.
EA	Evolutionary Algorithm.
EC	Evolutionary Computation.
EMO	Evolutionary Multi-Objective Optimisation.
EP	Evolutionary Programming.
ES	Evolution Strategies.
FSM	Finite State Machine.

GA	Genetic Algorithm.
GD	Generational Distance.
HypE	Hypervolume Estimation Algorithm for Multi-Objective Optimisation.
IBC	Indicator Based Conformation.
IBEA	Indicator-Based Evolutionary Algorithm.
IGD	Inverted Generational Distance.
m-CMA-PAES	Multi-tier Covariance Matrix Adaptation Pareto Archived Evolution Strategy.
MO-CMA-ES	Multi-Objective Covariance Matrix Adaptation Evolution Strategy.
MOEA/D	Multi-Objective Evolutionary Algorithm Based on Decomposition.
MOEA/D-DRA	Multi-Objective Evolutionary Algorithm Based on Decomposition with Dynamical Resource Allocation.
MOGA	Multi-Objective Genetic Algorithm.
NFL	No Free Lunch.
NPGA	Niched Pareto Genetic Algorithm.
NSGA	Nondominated Sorting Genetic Algorithm.
NSGA-II	Nondominated Sorting Genetic Algorithm II.
NSGA-III	Nondominated Sorting Genetic Algorithm III.
PAES	Pareto Archived Evolution Strategy.
PBEA	Preference-Based Evolutionary Algorithm.
PESA	Pareto Envelope-based Selection Algorithm.
PPA	Progressive Preference Articulation.
R-NSGA-II	Reference-point-based Nondominated Sorting Genetic Algorithm II.
ROI	Region of Interest.
SMS-EMOA	<i>s</i> -metric Selection Evolutionary Multi-Objective Algorithm.
WZ	Weighted Z-score.

WZ-CMA-PAES	Weighted Z-score Covariance Matrix Adaptation Pareto Archived Evolution Strategy.
WZ-MOEA/D-DRA	Weighted Z-score Multi-Objective Evolutionary Algorithm Based on Decomposition with Dynamical Resource Allocation.

List of Figures

2.1	A mythical effectiveness index across a problem continuum for optimisation schemes.	20
2.2	General execution life-cycle diagram of a basic Evolutionary Algorithm.	21
2.3	An example of a binary encoded chromosome.	22
2.4	An illustration of four solutions subjected to roulette wheel selection.	23
2.5	An example of single-point binary crossover.	24
2.6	Adaptation of the covariance matrix for a multivariate normal distribution.	26
2.7	Proximity, diversity, and pertinence characteristics in an approximation set in bi-objective space.	29
2.8	A Pareto-optimal approximation set containing 10 solutions with ideal diversity.	33
2.9	A Pareto-optimal approximation set containing 10 solutions with undesirable diversity.	33
2.10	An illustration of the trade-off between proximity and diversity to the Pareto-optimal front of an objective function.	35
2.11	An example plot of a population and visualisation of grid divisions managed by an AGA.	36

2.12	An example of the contributing hypervolume indicator in two-dimensional objective space.	38
2.13	Calculation of the crowding-distance.	39
2.14	A Pareto-optimal approximation set containing five solutions with ideal pertinence.	41
2.15	A Pareto-optimal approximation set containing seven solutions with undesirable pertinence.	41
2.16	Execution life-cycle for the PAES optimisation process.	48
2.17	Execution life-cycle for the MO-CMA-ES optimisation process.	50
2.18	Uniformly distributed random weight values generated for use with MOEA/D-DRA.	53
2.19	Plots of the Pareto-optimal fronts for all ZDT test functions excluding ZDT5, from the ZDT test suite.	57
2.20	Plots of the true Pareto-optimal fronts for the bi-objective DTLZ test functions from the DTLZ test suite.	60
2.21	Plots of the true Pareto-optimal fronts for the three-objective DTLZ test functions from the DTLZ test suite.	61
2.22	Plots of the true Pareto-optimal fronts for the bi-objective WFG test functions from the WFG test suite.	63
2.23	Plots of the true Pareto-optimal fronts for the three-objective WFG test functions from the WFG test suite.	64
2.24	Plots of the true Pareto-optimal fronts for the bi-objective CEC09 test functions from the CEC09 competition test suite.	66
2.25	Plots of the true Pareto-optimal fronts for the three-objective CEC09 test functions from the CEC09 competition test suite.	66

2.26	An example of the hypervolume indicator in two-dimensional objective space.	69
2.27	Histogram showing the distribution of the hypervolume indicator values from WZ-MOEA/D-DRA on the WFG6 test problem.	74
2.28	Relationship between SEM and the sample size of hypervolume indicator values from WZ-MOEA/D-DRA on the WFG6 test problem.	75
3.1	Execution life-cycle for the CMA-PAES algorithm.	81
3.2	Box plots of hypervolume indicator results for 2-objective ZDT problems.	86
3.3	Hypervolume indicator values at each generation for CMA-PAES and MO-CMA-ES on the considered ZDT test problems.	87
3.4	An approximation set found using MO-CMA-ES after 300,000 function evaluations on CEC09 UF1.	89
3.5	The true Pareto-optimal front and Pareto-optimal set for CEC09 UF1.	90
3.6	An example of elitist and non-dominated selection, circled points indicate a selected individual.	91
3.7	Execution life-cycle for the m-CMA-PAES algorithm.	93
3.8	An example of the multi-tiered grid selection.	96
3.9	IGD results at each generation visualising performance of m-CMA-PAES and MO-CMA-ES on two-objective test problems.	102
3.10	IGD results at each generation visualising performance of m-CMA-PAES and MO-CMA-ES on three objective test problems.	104
3.11	Box plots of IGD indicator results for two-objective test problems.	105
3.12	Box plots of IGD indicator results for three-objective test problems.	106

3.13	Disjoint true Pareto-optimal fronts plotted for problems ZDT3, UF6, and UF9.	106
4.1	Plot of the hypervolume indicator performance at each generation of CMA-PAES on the WFG3 synthetic test problem.	113
4.2	Execution life-cycle for the IBC mechanism.	117
4.3	Plot of the hypervolume indicator performance at each generation of CMA-PAES-II with and without IBC on the WFG3 test problem. . .	122
4.4	Plot of the hypervolume indicator performance at each generation of CMA-PAES-II with and without IBC and sigma restart on the WFG3 test problem.	124
4.5	Parallel-coordinate plot of the Population X used in the improved AGA example.	127
4.6	One dimensional plots illustrating the grid locations resolved by the AGA scheme for objective values.	128
4.7	Parallel-coordinate plot of the Population X used in the improved AGA example.	131
4.8	Execution life-cycle for the CMA-PAES-II algorithm.	132
4.9	Box plots of hypervolume indicator results from WFG problems for CMA-PAES-II and MOEA/D-DRA.	138
4.10	Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered two-objective WFG test problems.	143

4.11	Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered three-objective WFG test problems.	148
4.12	Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered five-objective WFG test problems.	152
4.13	Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered seven-objective WFG test problems.	155
4.14	Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered 10D WFG test problems. . . .	158
5.1	Basic Z-score preference articulation applied to an initial population generated by MOEA/D-DRA for the synthetic test problem ZDT1. . .	165
5.2	Population generated by MOEA/D-DRA combined with basic Z-score preference articulation for the ZDT1 test problem.	166
5.3	Basic Z-score preference articulation applied to an initial population generated by MOEA/D-DRA for the synthetic test problem WFG5. . .	166
5.4	Population generated by MOEA/D-DRA combined with basic Z-score preference articulation for the WFG5 test problem.	167
5.5	WZ preference articulation applied to an initial population generated by MOEA/D-DRA for the synthetic test problem WFG5.	171
5.6	Population generated by MOEA/D-DRA combined with WZ after 2000 function evaluations for the WFG5 test problem.	172

5.7	Population generated by MOEA/D-DRA combined with WZ after 300 function evaluations for the WFG5 test problem.	173
5.8	Population generated by MOEA/D-DRA without preference articulation after 2000 function evaluations for the WFG5 test problem. . . .	173
5.9	Basic Z-score preference articulation applied to an initial population generated by CMA-PAES for the ZDT4 test problem.	174
5.10	Population generated by CMA-PAES combined with basic Z-score preference articulation for the ZDT4 test problem.	175
5.11	Parallel-coordinate plot of the Population X and preference vector P used in the W-Phase example.	177
5.12	Parallel-coordinate plot of the Population X and preference vector P used in the Z-Phase example.	180
5.13	Box plots of hypervolume indicator results for two-objective ZDT problems.	191
5.14	The mean hypervolume indicator values of WZ-MOEA/D-DRA and MOEA/D-DRA populations at each generation for two-objective ZDT test suite.	193
5.15	Population generated by a run of MOEA/D-DRA and WZ-MOEA/D-DRA on ZDT4 within the ROI.	194
5.16	Population generated by worst run of MOEA/D-DRA and WZ-MOEA/D-DRA on ZDT4 within the ROI.	195
5.17	Box plots of hypervolume indicator results for five-objective WFG test cases.	198
5.18	The mean hypervolume indicator value of populations from considered algorithms at each generation for five-objective test functions.	200

5.19	Population generated by worst run of WZ-MOEA/D-DRA on WFG6.	202
5.20	Population generated by worst run of MOEA/D-DRA on WFG6. . .	203
5.21	The mean hypervolume indicator value of WZ-CMA-PAES (<i>a priori</i>) and WZ-CMA-PAES (PPA) populations at each generation for the ZDT1 PPA test case.	206
5.22	The mean hypervolume indicator value of WZ-CMA-PAES (<i>a priori</i>) and WZ-CMA-PAES (PPA) populations at each generation for the LATCON PPA test case.	207
6.1	System block diagram illustrating the arrangement of the transmitted and receiver horn antennas.	212
6.2	Encoded chromosome for an ANN consisting of 2 hidden layers, input layer, 2 neurons on the output layer, and associated biases.	214
6.3	Typical co polarised radar signals from a body alone, a body with a concealed gun, and finally a body with a concealed knife.	219
6.4	Typical cross polarised radar signals from a body alone, a body with a concealed gun, and finally a body with a concealed knife.	219
6.5	Encoded chromosome for the two-objective ANN consisting of 2 hidden layers, input layer, 1 neuron on the output layer, and associated biases.	221
6.6	Population generated by WZ-MOEA/D-DRA after 500 function evaluations for the two-objective concealed weapons detection classifier. .	223
6.7	Population of candidate solutions presented to the Decision Maker (DM), for selection to be made based on expert knowledge.	224

6.8	Encoded chromosome for the five-objective ANN consisting of 3 hidden layers, input layer, 3 neurons on the output layer, and associated biases.	229
6.9	Encoded chromosome for the seven-objective ANN consisting of 4 hidden layers, input layer, 4 neurons on the output layer, and associated biases.	229
6.10	Parallel-coordinates plot illustrating objective value results for five-objective threat detection.	235
6.11	Colour-map illustrating objective value results for five-objective threat detection, for use by the DM.	235
6.12	Parallel-coordinates plot illustrating objective value results for seven-objective threat detection.	237
6.13	Colour-map illustrating objective value results for seven-objective threat detection, for use by the DM.	237

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Outline of Thesis	5
1.3	Contributions and Objectives	8
1.3.1	Research Objectives	12
2	Review of Evolutionary Multi-Objective Optimisation	15
2.1	History of Evolutionary Computation	16
2.2	Evolutionary Algorithms	19
2.2.1	Genetic Algorithms	22
2.2.2	Evolution Strategies	25
2.2.3	Covariance Matrix Adaptation Evolution Strategy	25
2.3	Multi-Objective Optimisation	27
2.4	Evolutionary Multi-Objective Optimisation	30
2.5	Diversity Preservation in Evolutionary Multi-Objective Optimisation Algorithms	32
2.5.1	Conflicts between Proximity and Diversity	34
2.5.2	Methods of Diversity Preservation	35

2.6	Preference Articulation in Evolutionary Multi-Objective Optimisation	40
2.7	State of the art Evolutionary Multi-Objective Optimisation Algorithms	45
2.7.1	Non-dominated Sorting Genetic Algorithm II	45
2.7.2	Pareto Archived Evolution Strategy	46
2.7.3	Multi-Objective Covariance Matrix Adaptation Strategy	49
2.7.4	Multi-Objective Evolutionary Algorithm Based on Decomposition	51
2.8	Objective Functions	53
2.9	Multi-Objective Test Suites	55
2.9.1	The ZDT Test Suite	55
2.9.2	The DTLZ Test Suite	58
2.9.3	The WFG Tool-kit	61
2.9.4	The CEC2009 Competition Test Suite	65
2.10	Performance Assessment	68
2.10.1	The Hypervolume Indicator	68
2.10.2	The Generational Distance	70
2.10.3	Inverted Generational Distance	71
2.10.4	Non-Parametric Testing	72
2.10.5	Sample Size Sufficiency	74
3	The Covariance Matrix Adaptation Pareto Archived Evolution Strategy	77
3.1	CMA-PAES	78

3.1.1	Comparison Between CMA-PAES and MO-CMA-ES . . .	81
3.1.2	Results	84
3.1.3	Conclusion	88
3.2	m-CMA-PAES	88
3.2.1	New Multi-Tier AGA	94
3.2.2	Comparison Between m-CMA-PAES and MO-CMA-ES . .	96
3.2.3	Results	99
3.2.4	Conclusion	107
3.3	Conclusion	108
4	The Covariance Matrix Adaptation Pareto Archived Evolution	
	Strategy II	111
4.1	CMA-PAES on Many-objective Problems	112
4.2	Indicator Based Conformation	114
4.3	Sigma Restart	122
4.4	Improved Adaptive Grid Algorithm	125
4.5	CMA-PAES-II Design	131
4.6	Comparison between CMA-PAES-II and MOEA/D-DRA	135
4.7	Results	137
4.7.1	Two-Objective Results	140
4.7.2	Three-Objective Results	144
4.7.3	Five-Objective Results	150
4.7.4	Seven-Objective Results	153
4.7.5	Ten-Objective Results	156
4.8	Conclusion	159

5	Weighted Z-Score Preference Articulation	161
5.1	Weighted Z-Score Preference Articulation Operator	163
5.1.1	WZ Preference Articulation Operator Worked-Example . .	176
5.2	Incorporation of the WZ Preference Articulation Operator into CMA-PAES-II	181
5.3	Incorporation of the WZ Preference Articulation Operator into MOEA/D-DRA	183
5.4	Comparison of WZ-CMA-PAES and WZ-MOEA/D-DRA	185
5.4.1	Test Cases	186
5.4.2	Performance Assessment	188
5.5	Results	190
5.5.1	Two-Objective Results	190
5.5.2	Five-Objective Results	196
5.5.3	Progressive Preference Articulation Results	204
5.6	Conclusion	208
6	Application to Concealed Weapon Detection	209
6.1	Concealed Weapon Detection	211
6.2	Encoding the Problem	214
6.3	Comparison to Existing Solution	218
6.3.1	Experiment	220
6.3.2	Results	222
6.4	Detection and Classification of Multiple Types of Threat	225
6.4.1	Experiment	228
6.4.2	Results	234

6.4.3	Conclusion	238
6.5	Conclusion	238
7	Conclusion	241
7.1	Main Findings	242
7.2	Summary of Contributions	244
7.3	Recommendations for Future Research	248
	References	251
	Appendix A Algorithm Pseudo-code	275
A.1	MO-CMA-ES	276
A.2	Weighted Z-score Operator	277
	Appendix B Algorithms not considered	279
B.1	NSGA-III	279
B.2	DEMO and DEMOwSA	282

Chapter 1

Introduction

This chapter is an introduction to the thesis entitled “Preference Focussed Many-Objective Evolutionary Computation”, beginning with a description of the motivation for the research direction in Section 1.1, followed by an outline of the thesis in Section 1.2, and concluding with a listing of the contributions and research objectives in Section 1.3.

1.1 Motivation

Solving complex real-world problems often involves the simultaneous optimisation of multiple conflicting performance criteria, these real-world problems occur in the fields of engineering, economics, chemistry, manufacturing, physics, and many more. Typically the optimisation process involves some design challenges in the form of the optimisation of a number of objectives and constraints. There exist many traditional optimisation methods (calculus based, random search, enumerative, etc...), however, these only offer either adequate performance in a narrow problem domain or inadequate performance across a broad problem domain. The result of such traditional optimisation methods is often a single solution, and in

the presence of multiple conflicting objectives it is highly likely that this solution is bias towards a certain subset of objectives. An ideal solution to a real-world optimisation problem is an approximation set which contains multiple trade-off solutions that satisfy the Decision Maker (DM)'s preferences.

One approach to solving complex real-world problems is to use Evolutionary Multi-Objective Optimisation (EMO) algorithms to address each of the conflicting objectives simultaneously. EMO algorithms are robust optimisers which are suitable for solving multi-objective optimisation problems due to being population based, therefore being able to generate and exploit more than a single solution per iteration of the optimisation process. In addition, EMO techniques do not require any auxiliary or derivative information regarding the problem, do not require aggregation of multiple objectives into a single objective, and are less susceptible to the shape or continuity of the Pareto-optimal front. Typically, these EMO algorithms are run non-interactively with a DM setting the initial parameters of the algorithm and then analysing the results at the end of the optimisation process (which can often take hours or days to complete). This approach has been common since the late 1990s (e.g. in [1, 2, 3, 4, 5]) and will lead to an approximation set of potential solutions distributed across the whole Pareto-optimal set in a single algorithm execution [6].

Whilst classical EMO techniques produce promising results when applied to problems of multiple objectives (involving three or less problem objectives), applying these classical techniques to complex real-world problems of many objectives negatively impacts the behaviour of the EMO process, resulting in poor performance and delayed convergence. One research direction is to consider the use of a performance indicator [7] (e.g. the hypervolume indicator) as an al-

ternative to the commonly employed Pareto dominance concept which widely governs the selection process of many EMO algorithms, because it has been suggested that Pareto dominance struggles to produce a strong selection pressure towards the Pareto-optimal front in the presence of many objectives (e.g. in [8, 9, 10, 11, 12, 13, 14]). In these many-objective cases, the DM is usually more interested in a sub-region of this solution space that satisfies some domain specific criteria. A good EMO algorithm satisfies goals of convergence proximity, diversity preservation, and pertinence to a DM's Region of Interest (ROI) adequately, giving the DM knowledge of the trade-off of solutions within their ROI.

It is therefore desirable to use EMO techniques combined with the incorporation of DM preferences through preference articulation methods. Such a combination would be capable of solving complex real-world many-objective problems and arrive at approximation sets that offer adequate proximity, diversity, and pertinence, whilst reducing the computational cost of the optimisation process so that it does not take an infeasible amount of time to complete the search.

In this thesis, research is presented for enhancing the state of the art in the field of EMO. This is achieved by reviewing the Multi-Objective Covariance Matrix Adaptation Evolution Strategy (MO-CMA-ES), an existing multi-objective implementation of a powerful method for problem variable variation (Covariance Matrix Adaptation (CMA)), which is not feasible past the optimisation of four objectives (due to computational cost), and then developing a new CMA driven EMO algorithm under the name Covariance Matrix Adaptation Pareto Archived Evolution Strategy II (CMA-PAES-II), with the capability to optimise in the presence of many objectives. A new and novel method of preference articulation (the Weighted Z-score (WZ) preference articulation operator) is then

developed and incorporated into CMA-PAES-II and the Multi-Objective Evolutionary Algorithm Based on Decomposition with Dynamical Resource Allocation (MOEA/D-DRA) (a well-regarded EMO algorithm), and a series of test cases and real-world optimisation problems are used to demonstrate the advantages of the WZ preference articulation operator. A real-world optimisation problem involving the optimisation of an Artificial Neural Network (ANN) for the classification of radar signals for the purpose of concealed weapon detection is then optimised, using the Weighted Z-score Covariance Matrix Adaptation Pareto Archived Evolution Strategy (WZ-CMA-PAES) (an implementation of CMA-PAES-II with the incorporation of the WZ preference articulation operator).

1.2 Outline of Thesis

In Chapter 2, a review of evolutionary computation literature is presented, with conceptual emphasis on the use of evolutionary algorithms to solve multi-objective problems. The chapter begins with a brief history of the field of evolutionary computation, before moving onto an introduction to evolutionary algorithms, multi-objective optimisation, and EMO. There is an introduction to diversity preservation and preference articulation, followed by the description of a selection of state of the art EMO algorithms. The chapter concludes with an introduction to objective functions, a description of a selection of multi-objective test suites, and an introduction to EMO performance assessment methods.

In Chapter 3, the development of a new algorithm named the Covariance Matrix Adaptation Pareto Archived Evolution Strategy (CMA-PAES) is described, which is intended by design for fast convergence within a small function evaluation budget. This algorithm builds upon existing EMO algorithms and concepts in the EMO literature (e.g. CMA and Adaptive Grid Algorithm (AGA)) and is shown to perform comparably to MO-CMA-ES (another CMA driven algorithm) without the computational inefficiency which comes from relying entirely on the hypervolume indicator as a selection criterion. The development of a multi-tier variant of CMA-PAES, named the Multi-tier Covariance Matrix Adaptation Pareto Archived Evolution Strategy (m-CMA-PAES), is also described and shown to outperform MO-CMA-ES on a selection of difficult multi-objective synthetic test functions with complex Pareto-optimal sets.

In Chapter 4, CMA-PAES-II is described, combining design elements from CMA-PAES and m-CMA-PAES in combination with new concepts such as In-

indicator Based Conformation (IBC), sigma restart, and an improved AGA, to develop an algorithm designed specifically for the optimisation of many-objective problems. The algorithm is benchmarked against the popular and CEC2009 competition winning MOEA/D-DRA on a selection of many-objective synthetic test problems ranging from two to ten objectives. MO-CMA-ES is no longer considered in the comparison due to its infeasible computational cost on test problems consisting of greater than three objectives.

In Chapter 5, a novel method of preference articulation for EMO is introduced. This method, named the WZ preference articulation operator, is shown to provide improved performance in both the rate of convergence and pertinence in the solutions an EMO algorithm produces. As a demonstration, it is incorporated into two state of the art EMO algorithms and shown to improve both algorithms in the presence of DM preferences, through the pairwise statistical comparison of results from their execution on a selection of synthetic test problems and one real-world problem.

In Chapter 6, the WZ preference articulation operator is incorporated into CMA-PAES-II and used to optimise the architecture of an ANN used for concealed weapon detection in a two-objective, five-objective, and seven-objective problem. The chapter introduces the field of concealed weapon detection, and suggests a method for encoding and decoding the problem for use by the optimisation process. The results of the two-objective problem are compared to previously published results and shown to offer an improvement in performance of the classification of items of threat. The chapter then moves onto experiments on the detection and classification of multiple types of threat.

Finally, Chapter 7 presents a number of conclusions that have been drawn from the research presented in this thesis and suggests a number of directions for future work.

1.3 Contributions and Objectives

Publications and presentations resulting from the pursuit of achieving the research aims and objectives defined in Section 1.3.1 are:

- **Journal Article** Shahin Rostami; Dean O'Reilly; Alex Shenfield; Nick Bowring, "A Novel Preference Articulation Operator for the Evolutionary Multi-Objective Optimisation of Classifiers in Concealed Weapon Detection", DOI: 10.1016/j.ins.2014.10.031, Volume 295, 20 February 2015, Pages 494520, Information Sciences, Elsevier.
- **Conference Paper** Shahin Rostami; Alex Shenfield, "CMA-PAES: Pareto archived evolution strategy using covariance matrix adaptation for Multi-Objective Optimisation." Computational Intelligence (UKCI), 2012 12th UK Workshop on. IEEE, 2012.
- **Conference Paper** Shahin Rostami; Peter Delves; Alex Shenfield, "Evolutionary Multi-Objective Optimisation of an Automotive Active Steering Controller", DOI: 10.13140/2.1.1202.6240 Conference: Manchester Metropolitan University Research Symposium 2013.
- **Seminar Presentation** "Evolutionary Algorithms in Control Systems Engineering", 2011 16th November, Seminar, University of Manchester.
- **Poster Presentation** Shahin Rostami; Alex Shenfield, "Adaptive Grid Archiving Combined with the Covariance Matrix Adaptation Evolution Strategy", Conference: Manchester Metropolitan University Research Symposium 2012.

The main contributions of this thesis resulting from the pursuit of achieving the research aims and objectives defined in Section 1.3.1 are:

- **Development of CMA-PAES, a fast converging EMO algorithm.**

This EMO algorithm is inspired by the Pareto Archived Evolution Strategy (PAES) algorithm structure, the AGA method for diversity preservation, and the CMA scheme for variation, with the aim to be light in computational cost, simple in structure, and provide a fast rate of convergence. CMA-PAES has been shown to outperform MO-CMA-ES in this thesis in regards to the quality of the final approximation set paired with the low computational cost of the algorithm overhead. CMA-PAES has been published in [15] where it is shown to outperform the Nondominated Sorting Genetic Algorithm II (NSGA-II) and PAES, and has been successfully applied to the optimisation of an automotive active steering controller in [16]. A multi-tier variant of CMA-PAES (m-CMA-PAES) is also developed as an EMO algorithm intended for the optimisation of problems containing complex Pareto-optimal sets, by combining a non-elitist AGA based selection scheme with the efficient strategy parameter adaptation of the elitist Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES).

- **Development of CMA-PAES-II, a robust many-objective EMO algorithm.**

This EMO algorithm builds upon the work in CMA-PAES and m-CMA-PAES combined with the new IBC mechanism, sigma restart, and improved AGA. Unlike MO-CMA-ES, CMA-PAES-II allows for the use of CMA in EMO with a computational cost that does not restrict it to

execution on computing clusters or problems consisting of fewer than four objectives.

- **Development of the algorithm agnostic¹ and novel WZ Preference Articulation Operator.** The operator has the flexibility of being incorporated *a priori*, *a posteriori* or *progressively*, and as either a primary or auxiliary fitness operator. The two-phase operator has demonstrated the ability to successfully direct the optimisation process closer in proximity to a DM's expressed ROI, and then proceed to minimise solutions within it. This reduces the computational cost of the optimisation process by reducing the scope of the search-space exploration and produces pertinent optimisation sets.
- **Incorporation of preference articulation techniques into state of the art EMO algorithms.** In this thesis the WZ preference articulation operator has been incorporated into MOEA/D-DRA and CMA-PAES-II, which has shown to improve their performance and the quality in the final approximation set when searching in the presence of DM preferences.
- **Successful optimisation of classifiers used for concealed weapon detection.** Weighted Z-score Multi-Objective Evolutionary Algorithm Based on Decomposition with Dynamical Resource Allocation (WZ-MOEA/D-DRA) and WZ-CMA-PAES are successfully applied to a real-world optimisation problem regarding the optimisation of a classifier for concealed

¹An operator can be referred to as algorithm agnostic if it has been designed to be easily incorporated into any optimisation framework or algorithm.

weapon detection, producing better results than previously published classifier implementations. With the confidence instilled from the successful optimisation of the existing solution, new solutions were designed to allow the classification of radar signals into categories of threat objects (e.g. gun, knife, or explosive), which has produced a classifier which would allow for a better response to the detection of a concealed weapon.

Additional contributions that have arisen as a result of this research but are not included in this thesis are:

- **Creation of the “EMOLibrary” Evolutionary Multi-Objective Optimisation Toolbox for MATLAB.** The EMOLibrary provides many features that can be utilised in the design of new EMO algorithms, implementation of existing EMO algorithms, or conducting pairwise comparisons of EMO algorithms. The library was inspired by [17] which lacks modern features since it was released in 1994. Features of the EMOLibrary include:
 - **Performance Metrics**, such as the hypervolume indicator, spread, epsilon indicator, generational distance, and inverted generational distance.
 - **Selection/Sorting Operators**, such as non-dominated sorting, the contributing hypervolume indicator, AGA, and the WZ preference articulation operator.
 - **Test Problems**, such as problems from the following test suites: ZDT, DTLZ, WFG Toolkit, CEC09, and ELLI/CIGTAB test func-

tions. The objective function used for the design of lateral stability controllers (LATCON) for aircraft is also included.

- **Parameter Settings**, such as problem boundary defaults, suggested reference points, test cases for benchmarking of preference articulation techniques, problem encoding and decoding, problem dimensionality defaults, variable initialisers, and weight generators.
- **True Pareto-optimal Fronts** for the following test suites: ZDT, DTLZ, WFG Toolkit, and CEC09, with the ability to retrieve the portion of the true Pareto-optimal front within a defined ROI.
- **Successful Evolutionary Multi-Objective Optimisation of an Automotive Active Steering Controller.** The presented work [16] investigates the use of EMO to optimise the performance of a closed loop feedback Proportional Integral (PI) vehicle yaw controller on a non-linear vehicle. This is done by comparing results against traditional empirical tuning methods relating to rise time, settling time, overshoot, and steady-state error. The EMO application showed improvement on the original control tuning and also brought to light the difficulty control engineers face with objective interaction for complex problems.

1.3.1 Research Objectives

With the motivation described in Section 1.1, a number of aims and objectives were defined to guide the direction of work throughout the duration of this research. This thesis claims to have achieved every aim and objective in the chapters following. A listing of the aims and objectives are as follows:

Research Aim

To investigate the incorporation of decision maker preferences into multi-objective evolutionary search methods, so as to improve the quality of final solutions produced by the optimisation process.

Research Objectives

1. To produce a critical review of the field of evolutionary computation with a particular emphasis on using evolutionary computation methods to solve multi-objective problems.
2. To develop and benchmark a state of the art evolutionary multi-objective optimisation algorithm for solving real-world engineering problems, and to provide a basis for incorporating decision maker preferences by enhancing the preservation of diversity across the entire approximation set.
3. To develop a novel algorithm for focussing on regions of interest in multi-objective search spaces.
4. To evaluate the effectiveness of incorporating decision maker preferences into current state-of-the-art evolutionary optimisation routines, using statistically rigorous analysis.
5. To benchmark the new algorithms using synthetic test suites and real-world optimisation problems.

Chapter 2

Review of Evolutionary Multi-Objective Optimisation

The evolutionary computation literature is a rich source of knowledge which has been expanded upon every year since it was established as a field of research. In this chapter, a review of the literature within the field of evolutionary computation is presented with conceptual emphasis on the use of evolutionary algorithms to solve multi-objective problems.

The chapter begins with a brief historical overview of the field of evolutionary computation in Section 2.1. Section 2.2 provides a description of evolutionary algorithms, followed by an introduction to multi-objective optimisation (Section 2.3) and Evolutionary Multi-Objective Optimisation (EMO) (Section 2.4). The concept of diversity preservation is introduced in Section 2.5, followed by the concept of preference articulation in Section 2.6. Section 2.7 describes a selection of EMO algorithms considered to be state of the art which are relevant to this thesis. Section 2.8 describes the purpose of objective functions, followed by an overview of multi-objective synthetic test suites. The chapter is concluded with Section 2.10 on methods of EMO performance assessment.

2.1 History of Evolutionary Computation

Evolutionary Computation (EC) refers to a methodology concerning adaptive search and optimisation techniques, derived from the mechanics of natural selection [18] and modern biological genetics [19]. EC is a sub-field of Computational Intelligence (CI) alongside other biologically inspired computing techniques such as Artificial Neural Networks (ANNs) and Artificial Immune Systems (AISs). EC is an interdisciplinary field, bringing together theories of evolutionary biology, computation, mathematics, and physics.

The emergence of EC can be traced back as far as the early 1930s, when the American geneticist Sewall Wright visualised evolution as a search through a landscape of gene combinations graded by their adaptive values [20]. This was later referred to as a “fitness landscape”, containing multiple peaks and valleys representing the fitness of individuals [21]. This provided mathematicians and computer scientists with the notion that evolution is a form of optimisation, and harnessing such an optimisation technique within a computer could potentially solve complex optimisation problems, which traditional algorithms would struggle with.

With evolution portrayed in a manner that was appealing to computer scientists, early contributions were made using computers combined with evolutionary approaches. This dates back as far as the 1950s with work concerning evolutionary robotics [22], an evolutionary method for increasing industrial productivity [23], and evolving sets of machine language instructions to create a learning machine [24, 25]. However, due to the unavailability of powerful computers to the broader scientific community, the field remained unexplored by many.

During the 1960s, the field began to grow exponentially as a result of inexpensive - yet powerful - computers increasing in availability to the scientific community. This led to the pioneering work by three independent groups, each with unique yet related ideas. Ideas of solving real-valued optimisation problems using the evolutionary process were considered by [26] and [27]. These ideas resulted in the formation of a set of algorithms named Evolution Strategies (ES). Simultaneously, an evolutionary framework named Evolutionary Programming (EP) was introduced in [28] and was originally intended to evolve Finite State Machines (FSMs). However, since its introduction to the field, expansion and refinement of the framework has opened the application of EP to problems well beyond evolving FSMs. Holland, inspired by evolutionary processes, proposed a general model of adaptive processes [29]. This led to an initial set of reproductive plans, which formed the basis of what is today referred to as the simple genetic algorithm.

The same basic process involving a fixed-size population of solutions, stochastic events of recombination, and the concept of solution fitness could be recognised in the manifestation of ideas from each of these groups. However, it wasn't until the early 1970s that it became evident that these parameters and their associated rates made a significant impact on the convergence of these algorithms when implemented [30]. This notion was expanded upon by [31] with an investigation into the alteration of parameters and the possible stochastic side-effects. It was concluded that the slight variation of these parameters could provide a more robust overall search at the expense of a slower initial response. This resulted in an overall improvement in the performance of Genetic Algorithms (GAs).

Throughout the 1970s and 1980s the individual models (ES, EP and GA) were developed individually by their respective groups, until the early 1990s. Representatives of each group had begun attending Evolutionary Algorithm (EA) conferences, where they discussed their viewpoints and challenged each others ideas. These interactions resulted in an agreement on the term Evolutionary Computation as the name to unify the general field, as well as the hybridisation of the ideas between the three models [32, 33].

Since the late 1990s and the 21st century many developments in the field have been made. Some of these developments have been made in the form of new EMO algorithms such as the Pareto Archived Evolution Strategy (PAES) [34], the Nondominated Sorting Genetic Algorithm II (NSGA-II) [35], the Multi-Objective Covariance Matrix Adaptation Evolution Strategy (MO-CMA-ES) [36], and the Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [37]. With the development of so many new EMO algorithms, it became necessary to have a standard set of synthetic test functions to allow for the assessment of their performance, this resulted in multi-objective test suites such as ZDT [38], DTLZ [39], and WFG [40].

More recently, hybrid EAs have been realised through embedding local search into the framework of EAs. Memetic Algorithms (which also go under the name Hybrid EAs or Cultural Algorithms), are inspired not only by Darwinian principles of natural evolution, but also by Dawkin's notion of a meme [41]. A meme, analogously to a gene, defines the basic unit of cultural transmission or imitation. Memetic algorithms have shown promise in solving single-objective and multi-objective problems [42]. Hyper-heuristics are a methodology in search and optimisation which are concerned with choosing an appropriate heuristic or algo-

rithm in any given optimisation context [43], and can operate on meta-heuristics. Hybrid algorithms, memetic algorithms, and hyper-heuristics, all indicate the benefits of using an approach which aim to combine existing algorithms and heuristics such that a more general approach can be taken to optimisation.

2.2 Evolutionary Algorithms

EAs are a powerful class of stochastic optimisation techniques that incorporate some of the principles of natural selection and population genetics to converge towards global optima [44]. They provide an iterative and population-based approach to optimisation that is capable of both exploring the search space of a problem and exploiting promising solutions found in previous generations. Typically the exploration of the search space is performed by using variation operators (such as mutation), which introduce an element of stochasticity into the optimisation process and aim to prevent premature convergence to local optima. In contrast, exploitation of promising solutions from previous generations is performed using a selection operator (and in part, recombination operators) that ensures preference is given to solutions that are considered fittest from the previous generation.

The robustness of EAs to multi-modal search landscapes containing many local optima (and other difficulties present in multi-objective search spaces) and the direct use of objective function information (rather than auxiliary knowledge such as derivative information) ensures that EAs are effective when applied to many problem types in which conventional optimisation methods may have difficulty, this is put in perspective in Figure 2.1.

In regards to EAs, there are many different algorithms which exist for the purpose of optimisation, and there exists no “best” algorithm. This is shown in the No Free Lunch (NFL) theorem described in [45], such that if algorithms are averaged over all possible optimisation problems, no algorithm has a performance advantage over any other. This theorem has been confirmed to hold for EAs [46].

The population-based nature of EAs helps to ensure that they are resilient when faced with noisy search spaces, as each generation contains more information about the shape of the fitness landscape than would be available to conventional, non-population based methods such as hill-climbing [47].

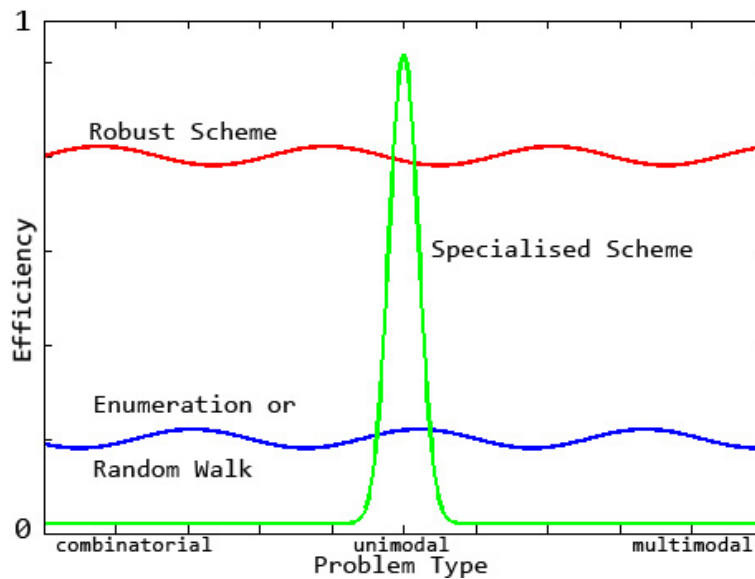


Figure 2.1: “A mythical effectiveness index is plotted across a problem continuum for a specialised scheme, an enumerative scheme, and an idealised robust scheme.”

David E. Goldberg. Genetic Algorithms in Search, Optimisation & Machine Learning, 1989. [44]

The execution life-cycle of a general and basic EA is shown in Figure 2.2. The optimisation process begins by generating an initial population of random

candidate solutions, which are then evaluated using problem specific objective functions and assigned fitness values based on the objective values and other potential indicator values which may be considered. The fitness values of the population solutions are then checked to identify whether any of the current solutions satisfy the termination criteria to terminate the optimisation process, otherwise the process will continue onto the selection of the fittest individuals from the current population, in order to exploit the genetic information contained within the best current solutions. The selected candidate solutions are then used for recombination to exploit the best solution information, and mutation to allow for exploration of the search space beyond the available solution information present in the population and attempts to prevent the possibility of getting stuck in a local optima.

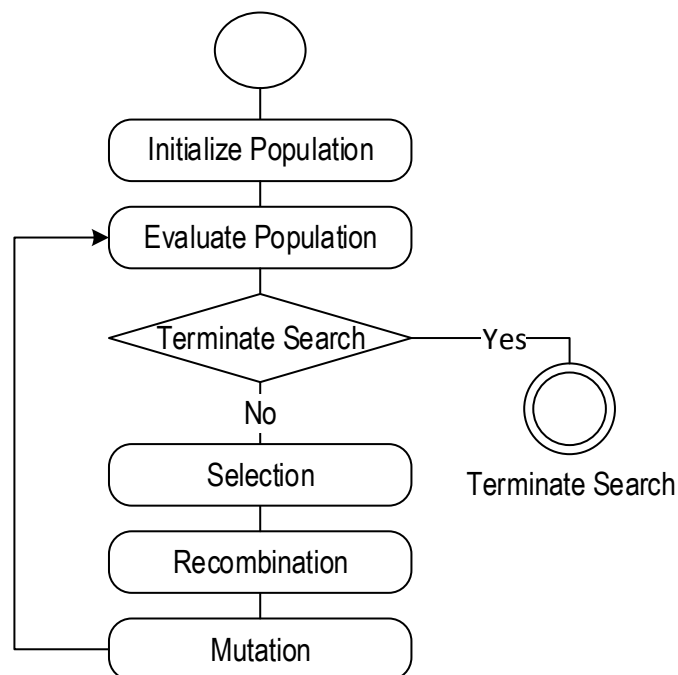


Figure 2.2: General execution life-cycle diagram of a basic Evolutionary Algorithm.

2.2.1 Genetic Algorithms

A Genetic Algorithm is a single-objective optimisation algorithm which operates using a large population of solutions [44]. These solutions are encoded as a string of real numbers or traditionally as a binary bit string (sometimes referred to as a chromosome), an example of a binary chromosome is illustrated in Figure 2.3.



Figure 2.3: An example of a binary encoded chromosome.

Initially, a predefined number of solutions are generated (limited by the population size parameter) and evaluated using a fitness function to identify which solutions are better suited to solving the problem. An example of a simple fitness function is described in Equation 2.1. When putting the solution from Figure 2.3 through the fitness function it is assigned the fitness value 12.83, in a minimisation problem, a smaller value indicates a better solution. The fitness values of all the solutions are evaluated to check whether predefined termination criteria has been satisfied, this can be reaching a threshold fitness value or completing a number of generational iterations. If the criteria has not been satisfied the algorithm continues.

$$fitnessfunction(a, b, c, d, e) = \frac{((a \times 32) + (b \times -26) + (c \times -5) + 50)}{(d + 1) \times (e + 2)} \quad (2.1)$$

Once each solution has been assigned a fitness value, new offspring solutions are typically generated by the selection and recombination of parent solutions. Selection can be achieved through various selection schemes, in this example roulette wheel selection [48] is used. In roulette wheel selection, the probability

of a solution being selected is proportional to their fitness, although it is not certain that the fittest solutions will be selected, only that they have a higher chance. To select an individual, a random number is generated between 0 and 100, this number is then checked against the roulette wheel to see which solution is selected. This has been illustrated in Figure 2.4, where if the randomly generated number is 17, then solution (1) is selected.

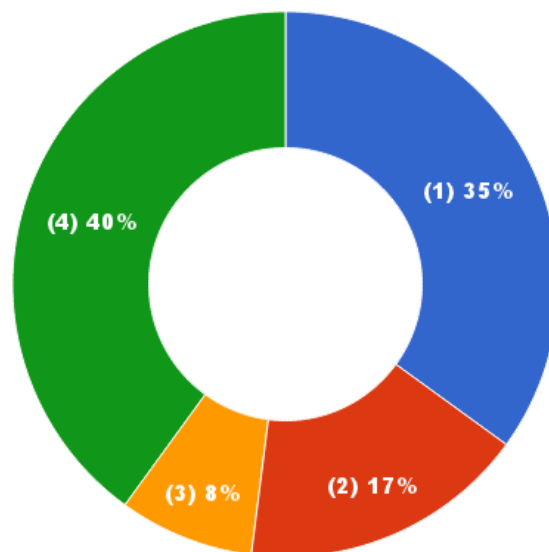


Figure 2.4: An illustration of four solutions subjected to roulette wheel selection.

A random number is then generated between 0 and 1 to decide whether to perform the crossover process on the two selected parent individuals, this random number is checked against a predefined crossover rate parameter (controlling the probability that crossover will occur), and if the randomly generated number is lower, then the crossover is performed. A simple binary crossover regards the swapping of bits between the two binary chromosomes after a certain point in the binary bit string (single point crossover), this has been illustrated in Figure 2.5.

The example given in Figure 2.5 illustrates the crossover and mutation of two parent solutions in order to create two offspring solutions, one of which receives a mutation. In order to determine the position of the crossover point a number between 0 and the length of the binary bit string is randomly generated, the bit strings of the two parent solutions are then swapped at this point to produce two new offspring solutions. A random number between 0 and 1 is then generated and compared to see if it is lower than a pre-defined mutation rate parameter to decide whether mutation is to take place, the mutation rate is often very low for a binary encoded chromosome (e.g. 0.001). If mutation is to take place on either of the offspring solutions, a random position in the binary bit string is selected and the binary value is flipped (either from 0 to 1 or 1 to 0), in this example only Offspring 2 has received a mutation. This process of selection, recombination and mutation continues until the populations solution capacity has been met, and another iteration of the GA continues from the evaluation of solutions.

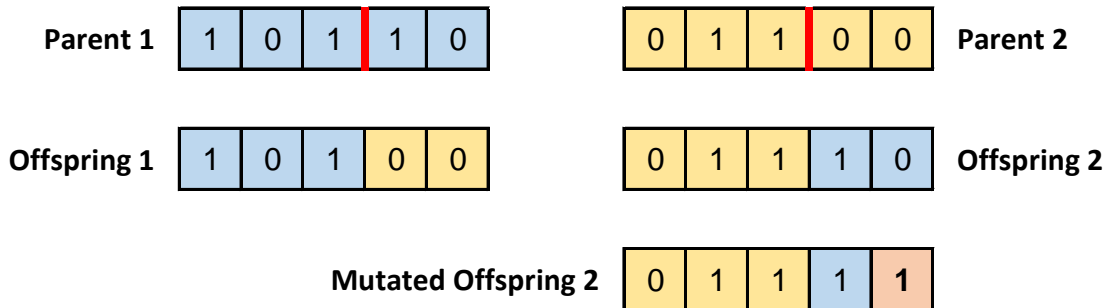


Figure 2.5: An example of single-point binary crossover from position 3 in the binary bit string, and the mutation of binary position 5 in Offspring 2.

2.2.2 Evolution Strategies

Ideas of solving real-valued optimisation problems using an evolutionary process were considered by [26] and [27]. These ideas resulted in the formation of a set of algorithms named Evolution Strategies (ES), which differed from other EA methods in two ways: ES used real parameter values; and they did not use recombination operators, instead the variation of solutions during the optimisation process is driven entirely by mutation. ES were typically implemented in two forms: two-member ES ($1 + 1$), in which a single parent is used to create a single offspring using a mutation operator; and multi-member ES ($\mu + \lambda$) or (μ, λ) , in which more than one parent solution (μ) is used to create λ offspring solutions using a mutation operator. In the $(\mu + \lambda)$ variation of the multi-member ES, both parent and offspring populations are considered in selection for the next parent population, whereas in the (μ, λ) variation, only the offspring population is considered, making the $(\mu + \lambda)$ an elitist procedure.

2.2.3 Covariance Matrix Adaptation Evolution Strategy

The Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) is a state of the art single objective ES first introduced in [49, 50] and later improved upon in [51, 52]. ES typically use a multivariate normal distribution and rank-based selection to apply mutations to a population of solutions in order to continue searching to the next generation. The CMA-ES adapts the mean and full covariance matrix of this multivariate normal distribution (illustrated in Figure 2.6) to direct the search towards new solutions.

The CMA-ES is invariant against linear transformations of the search space, it has been shown to perform extremely well across a broad range of problems

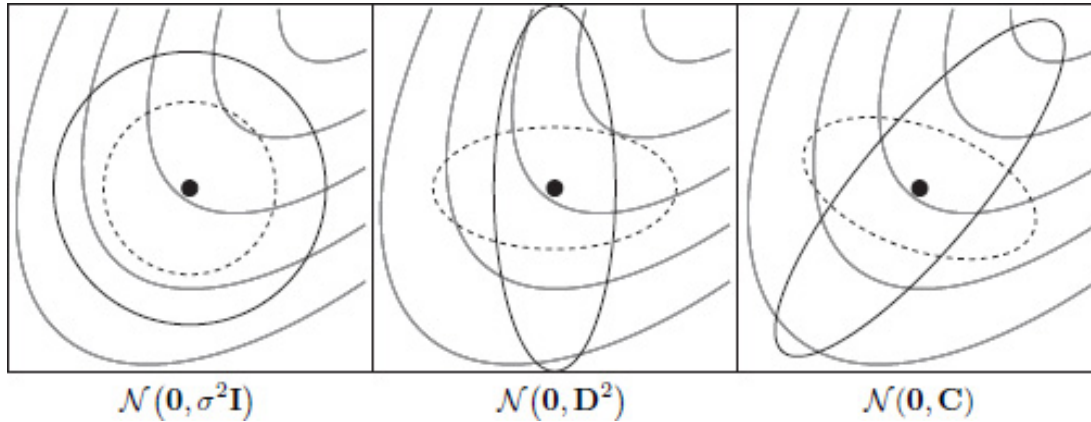


Figure 2.6: “Six ellipsoids, depicting one- σ lines of equal density of six different normal distributions, where $\sigma \in \mathbb{R} + D$ is a diagonal matrix, and \mathbf{C} is a positive definite full covariance matrix. Thin lines depict exemplary objective function contour lines.”

Nikolaus Hansen. The CMA evolution strategy: A comparing review, 2006. [53]

in the continuous domain [54], and it is robust to the initial parameter set used due to its self-adaptive nature. CMA-ES can be used independently as a primary optimiser, but is often incorporated into other optimisation algorithms as a local optimiser. One of the key properties of CMA-ES is the speed at which it can find good approximations to (and in many cases the actual value of) the global minimum.

CMA-ES is an ES and therefore implements mutation as the only scheme for problem variable variation, in contrast to a GA which implements both a recombination and mutation scheme. This difference is an advantage for CMA-ES if it is to be extended for multi-objective and many-objective problems, as recombination operations often become inefficient as the number of problem objectives increase. Solutions become more likely to be distant from each other in objective space as the number of objectives increase, such that two distant parent solutions are likely to produce offspring solutions that are also distant from the

parents [55]. A disadvantage to CMA-ES is that it does not inherit any of the benefits from implementing recombination, which allows the exploitation of genes through the mixing of chromosomes. However, implementing recombination in CMA-ES would disrupt the sophisticated mutation scheme. Additionally, it has been shown in [56] that the NFL theorem does not hold in continuous search domains, meaning CMA-ES, which performs extremely well in the continuous domain, has an advantage over other algorithms.

Through many performance comparisons across different test suites of benchmark problems (e.g. the competition results in [57, 54, 58]) the CMA-ES has been proven to be a powerful and robust optimiser.

2.3 Multi-Objective Optimisation

Multi-objective optimisation, as implied by the name, refers to problems with two or more objective functions, this is often the case with real-world problems in search and optimisation which naturally involve multiple objectives or multiple criteria [59].

A fundamental difference between single-objective optimisation and multi-objective optimisation is that in single-objective optimisation problems, the objective is to find a single solution which represents the global optimum in the entire search space, where as multi-objective optimisation problems often involve conflicts between multiple objectives, and as a result it is unlikely that there exists a single optimal solution. Therefore, in multi-objective optimisation a solution is an approximation set of candidate solutions which offers a representation of the trade-offs between the multiple objectives, where any improvement in one

objective value will result in the degradation in one or more of the other objective values. This notion of “optimum” solutions is called Pareto-optimality.

$$x = (x_1, x_2, \dots, x_n) \quad (2.2)$$

$$\left. \begin{array}{l} \text{optimise } f_m(x), \quad m = 1, 2, \dots, M; \\ \text{subject to } g_j(x) \geq 0, \quad j = 1, 2, \dots, J; \\ \quad \quad \quad h_k(x) = 0, \quad k = 1, 2, \dots, K; \\ \quad \quad \quad x_i^{(L)} \leq x_i \leq x_i^{(U)} \quad i = 1, 2, \dots, n; \end{array} \right\} \quad (2.3)$$

$$f(x) = (f_1(x), f_2(x), f_3(x), \dots, f_M(x)) \quad (2.4)$$

A solution x is defined in Equation 2.2 as a vector of n decision variables. In Equation 2.3, a multi-objective optimisation problem is described in its general form, taken from [59]. There are M objective functions with the definition in Equation 2.4, these objective functions can be either minimised or maximised. The constraint functions $g_j(x)$ and $h_k(x)$ impose inequality and equality constraints that must be satisfied by a solution x in order for it to be considered a feasible solution. Another condition which affects the feasibility of a solution regards the adherence of a solution x to values between the lower $x_i^{(L)}$ and upper $x_i^{(U)}$ boundaries within the decision space.

A set of non-dominated solutions¹ generated by the optimiser is known as an *approximation set* [60] and can be characterised in three key areas [61]. These are illustrated graphically in Figure 2.7 and listed in the following:

¹A solution is termed non-dominated if there exists no other solution in the population that is superior to it in all considered problem objectives.

- Proximity. This tells the Decision Maker (DM) how close the approximation set is to the true Pareto-optimal front. An ideal approximation set should be as close as possible in proximity to the true Pareto-optimal front.
- Diversity. This characterises the distribution of the approximation set both in the extent and uniformity of that distribution. The ideal approximation set should be uniformly distributed across the trade-off surface of the problem.
- Pertinence. This criteria measures the relevance of the approximation set to the DM. Ideally the approximation set should contain a number of solutions which satisfy the DM's expressed preferences.

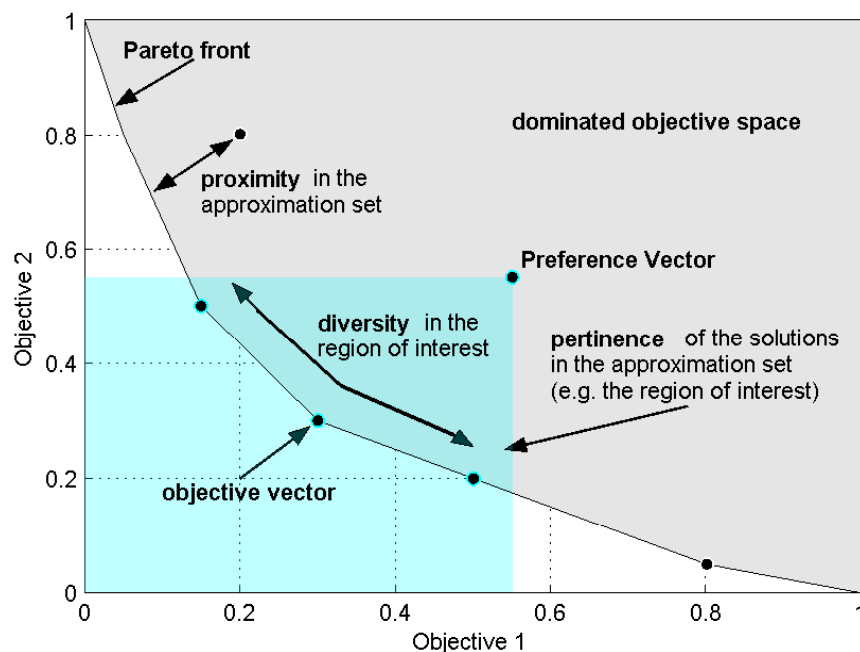


Figure 2.7: Proximity, diversity, and pertinence characteristics in an approximation set in bi-objective space.

Conventional multi-objective optimisation techniques often fail to satisfy these criteria. For example, the goal-attainment method [62] and the weighted-sum method [63] both only provide single solutions to the optimisation problem - thus failing to provide a diverse distribution of solutions. However, EAs are well suited to this kind of multi-objective optimisation since they search a population of candidate solutions and are thus capable of presenting a diverse approximation set to a DM [59].

Many theoretical EMO studies only consider a small number of objectives, with most of the published literature focussing on the bi-objective case. However complex real-world problems often require the consideration of a higher number of objectives. This has led to much interest amongst the research community in *many-objective*² optimisation. The increased scale of a many-objective optimisation problem means that the *pertinence* of the approximation set presented to a DM is especially important. The global trade-off surface for a problem with many conflicting objectives frequently contains many Pareto-optimal solutions, the majority of which may not be in the DM's Region of Interest (ROI) [61]. In this case the exploration of those undesirable regions of the objective space results in inefficiency in terms of the use of computational resources and the quality of the final approximation set produced.

2.4 Evolutionary Multi-Objective Optimisation

Multi-objective optimisation problems had previously been solved by being encapsulated as single-objective optimisation problems using techniques such as the

²The phrase *many-objective* has been used in the operations research community to refer to problems with more than the standard two or three objectives [64].

weighted sum approach [65]. In this approach, different weights are assigned to each objective function based on their configured level of importance and priority, these weighted objectives are then aggregated into a single weighted sum, allowing the use of conventional optimisation techniques to solve the problem. A major disadvantage of using the weighted sum approach and other conventional multi-objective optimisation approaches, is that they can only produce a single candidate solution per execution, and therefore require multiple executions to generate a set of trade-off solutions.

In contrast, EMO algorithms have inherited beneficial properties from the principles upon which they are based. EAs are suitable for solving multi-objective optimisation problems, due to being population based and therefore being able to generate and exploit more than a single solution per generational iteration, this allows EAs to find several solutions in the Pareto-optimal set in a single algorithm execution [6]. In addition, EMO algorithms do not require auxiliary or derivative information regarding the problem, do not require the aggregation of problem objectives into a single objective, and are less susceptible to the shape or continuity of the Pareto-optimal front.

Within the last decade there have been major advances in the field of EMO. Whilst the first generation of Pareto-based EMO's algorithms (such as the Multi-Objective Genetic Algorithm (MOGA), the Niche Pareto Genetic Algorithm (NPGA), and the Nondominated Sorting Genetic Algorithm (NSGA)) were characterised by the simplicity of the algorithms and lack of rigorous methodology for their analysis [66], the latest generation of EMO algorithms have focussed on efficient convergence to the whole of the true Pareto-optimal front. This has been accomplished by incorporating elitism (ensuring that the best solutions are never

lost during the optimisation process) and advanced methods for the preservation of diversity (to ensure a good spread of solutions across the whole Pareto-optimal front) into the selection-for-survival process. There are two main strategies for incorporating elitism into EMO algorithms — maintaining an archive of non-dominated solutions, and using a $(\mu + \lambda)$ type selection-for-survival mechanism.

2.5 Diversity Preservation in Evolutionary Multi-Objective Optimisation Algorithms

After proximity to the true Pareto-optimal front, diversity of solutions in the Pareto-optimal approximation set is the most desired quality in a robust EMO algorithm. The reason for this is because in EMO and multi-objective optimisation in general, there exists no single ideal solution to a problem. Instead there exists many Pareto-optimal solutions, and in the Pareto-optimal approximation set the minimisation of one objective will result in the increase of another objective. For this reason, the DM requires a set of Pareto-optimal solutions that are uniformly spread along the objective space, to allow the DM to observe the trade-off information and use domain specific expert knowledge to select a final solution.

Figure 2.8 presents an ideal Pareto-optimal approximation set of solutions uniformly distributed along the Pareto-optimal front, this is a Pareto-optimal approximation set with both ideal proximity and diversity. In another scenario presented in Figure 2.9, the EMO process has successfully converged to solutions along the Pareto-optimal front, however it has not achieved a satisfactory level of diversity amongst solutions in the Pareto-optimal approximation set. This

scenario does not offer the DM with adequate information to make a well-informed decision.

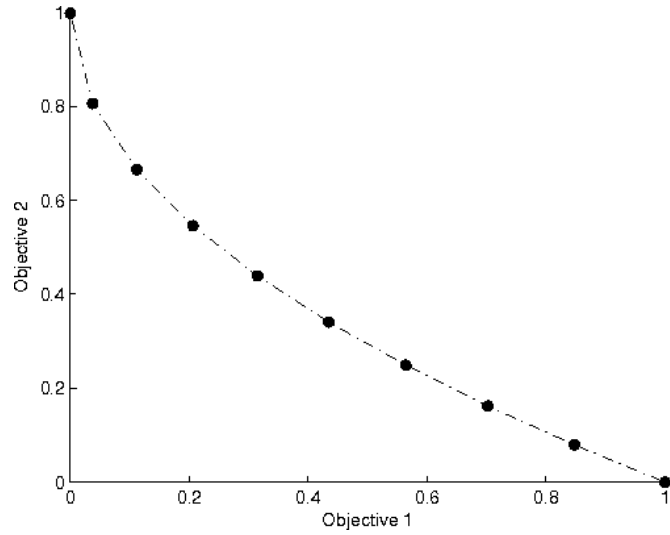


Figure 2.8: A Pareto-optimal approximation set containing 10 solutions with ideal diversity.

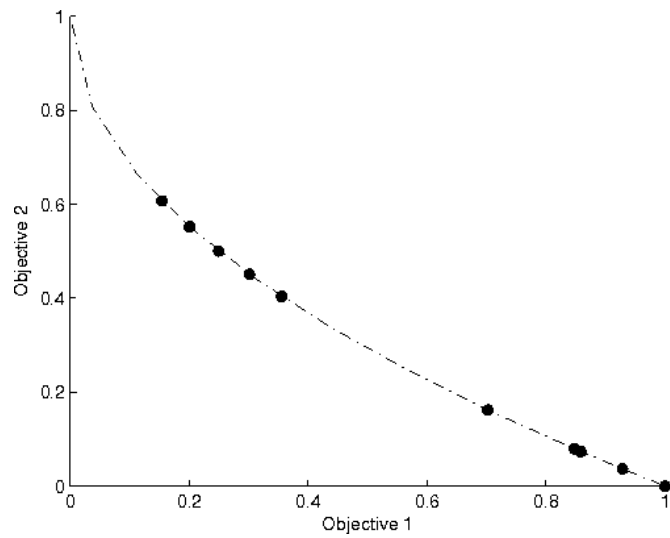


Figure 2.9: A Pareto-optimal approximation set containing 10 solutions with undesirable diversity.

2.5.1 Conflicts between Proximity and Diversity

The EMO process (and multi-objective optimisation process in general) is presented with a multi-objective trade-off of its own. This trade-off arises due to the conflict between attaining ideal proximity and diversity in an approximation set. This is a bi-objective trade-off which exists in most cases where the true Pareto-optimal front is not known. In such a case it is not possible to determine whether the approximation set has converged to the true Pareto-optimal front, and therefore diversity preservation cannot become the focus of the remainder of the search. However, diversity preservation usually comes second to obtaining a good approximation set, as stated in [67]. The goal of diversity preservation is to preserve diversity along an approximation set as close to the Pareto-optimal front as possible.

The example in Figure 2.10 illustrates the trade-off between proximity and diversity. Set 2 has a more diverse population of solutions in comparison to Set 1; however Set 1 is closer in proximity to the Pareto-optimal front than Set 2. In this case, the better diversity offered by Set 2 is not as valuable as the proximity offered by Set 1.

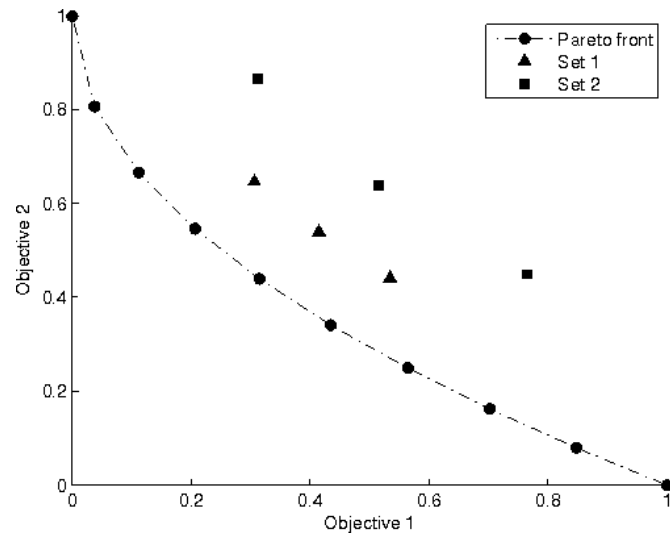


Figure 2.10: An illustration of the trade-off between proximity and diversity to the Pareto-optimal front of an objective function.

2.5.2 Methods of Diversity Preservation

There exist many novel concepts and variants of these concepts for the preservation of diversity in a population throughout the EMO process. In this section a selection of methods of diversity preservation are described.

Adaptive Grid Algorithm

Bounded Pareto archiving (as in the Adaptive Grid Algorithm (AGA) strategy used in the PAES algorithm) is a simple yet powerful diversity preservation scheme which uses an adaptive grid to keep track of the density of solutions within divisions of the objective space [34]. To achieve this, a grid with a pre-set number of divisions is used to divide the objective space and when a solution is generated, its grid location is identified and associated with it. Each grid location is considered to contain its own sub-population, and information on how many

solutions in the archive are located within a certain grid location is available during the optimisation process, this has been illustrated in Figure 2.11.

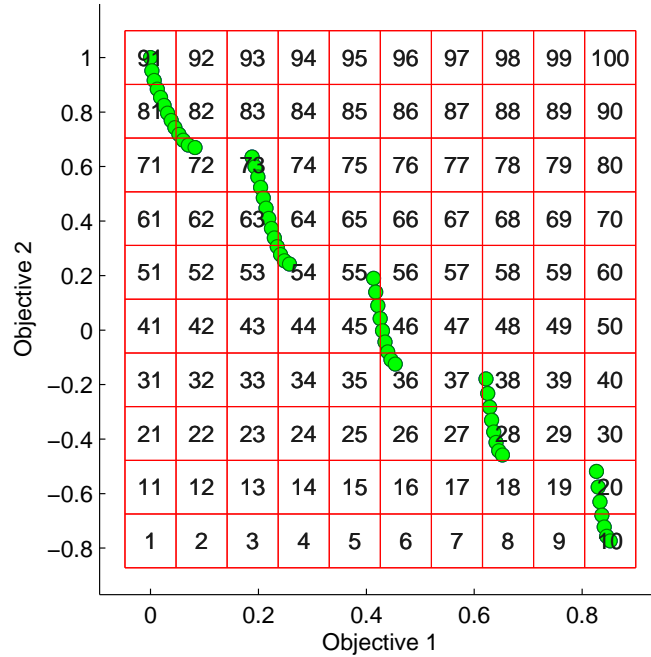


Figure 2.11: An example plot of a population and visualisation of grid divisions managed by an AGA.

When an archive has reached capacity and a new candidate solution is to be archived, the information tracked by the AGA is used to replace a solution in a grid location containing the highest number of solutions. When a candidate solution is non-dominated in regards to the current solution and the archive, the grid information is used to select the solution from the least populated grid location as the current (and parent) solution.

The AGA concept used in PAES (described in Section 2.7.2) later inspired several researchers, and was altered and deployed in multiple EMO algorithms such as the Pareto Envelope-based Selection Algorithm (PESA) (a population

based version of PAES) [68], the Micro Genetic Algorithm [69], and the Domination Based Multi-Objective Evolutionary Algorithm (ϵ -MOEA) [70].

Contributing Hypervolume

The contributing hypervolume indicator is an adaptation of the hypervolume indicator in order to be used as sorting criteria for selection operators, it has been used in the s -metric Selection Evolutionary Multi-Objective Algorithm (SMS-EMOA) [71] and the Hypervolume Estimation Algorithm for Multi-Objective Optimisation (HypE) [72]. The hypervolume indicator works by calculating the size of the objective space that has been dominated by an entire approximation set in regards to a specified reference point, where as the contributing hypervolume indicator assigns each solution in an approximation set with the size of the space covered by each solution exclusively. With this information the population can be sorted by the most dominant and diverse solutions. In addition, most contributing hypervolume indicator selection methods always assign solutions containing the extreme values for an objective with the highest hypervolume value. This has been illustrated in Figure 2.12 in two-dimensional space with a population of three solutions.

Although many state of the art EMO algorithms use the contributing hypervolume as a sorting criteria for selection, its calculation becomes computationally infeasible as the number of problem objectives considered increase. Monte Carlo approximations have been used to speed up the calculation of the contributing hypervolume in [73], which through empirical experiments has shown that the method does not impair the quality of the approximation set. However, the speed increase provided by the Monte Carlo approximation method still results

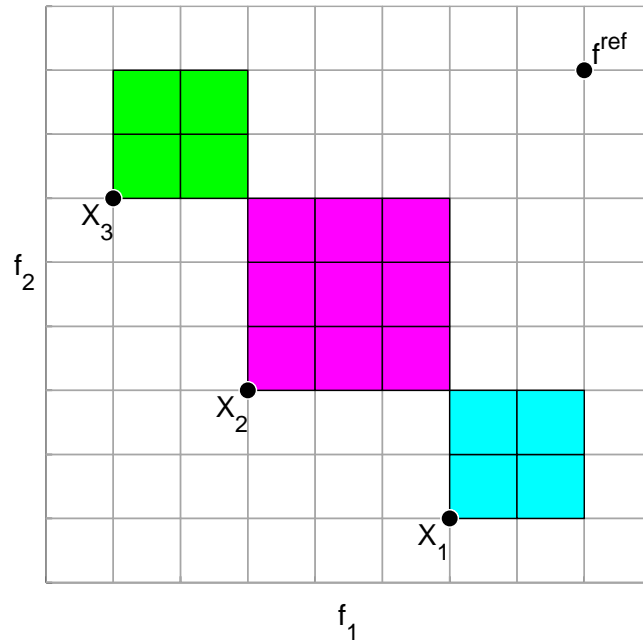


Figure 2.12: An example of the contributing hypervolume indicator in two-dimensional objective space.

in the contributing hypervolume indicator being infeasible on problems consisting of five objectives or more.

This particular measure of diversity preservation can also be used post-optimisation to reduce the size of a final approximation set produced by an optimiser, to a size that will not overwhelm and confuse a DM.

Crowding Comparison Operator

The crowded comparison operator is used in various stages of NSGA-II to guide its selection process towards an approximation set with uniformly spread out solutions. Associated with each individual in a population is two algorithm specific properties: a non-domination rank, in which solutions are ranked by the number of solutions they are dominated by, found using the fast non-dominated

sorting approach; and a local crowding distance, which is an estimation of the density of solutions surrounding a particular solution in the population [35, 74]. An illustration of this measure is given in Figure 2.13.

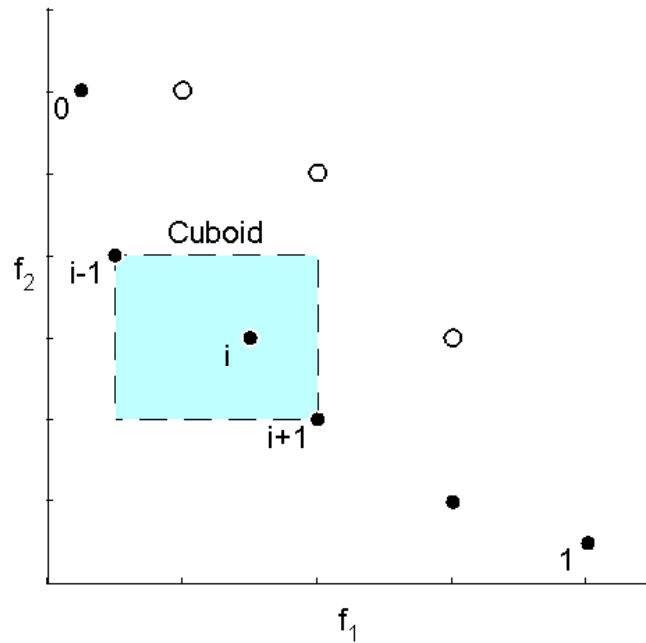


Figure 2.13: Calculation of the crowding-distance — points marked with solid markers are solutions of the same non-dominated rank.

Between two solutions with different non-domination ranks, the solution with the lower rank is given preference. However, if both solutions are of the same domination rank, then the solution which is located in a region with the least number of solutions is given preference.

2.6 Preference Articulation in Evolutionary Multi-Objective Optimisation

When solving real-world multi-objective problems, the ideal optimisation algorithm is one which converges to non-dominated Pareto-optimal solutions within the DM's ROI. This allows for the DM to be presented with a small set of trade-off solutions which are within their ROI (illustrated in Figure 2.14), as opposed to a larger set of trade-off solutions within the entire objective space (illustrated in Figure 2.15). Subsequently, the DM is not overwhelmed with a large set of candidate solutions when using expert knowledge to select their desired solution to the problem. Furthermore, when an ROI is specified by the definition of preferences, the algorithm is able to use this information during the search to discard trade-off solutions which do not fall within the desired region, and to skew the search towards the region by influencing the EMO algorithm's selection operator. This additional preference information ultimately reduces the area of feasible solutions within the objective space, thus reducing the computational effort needed to produce a diverse set of pertinent solutions to aid the DM in making a decision.

The role of the DM in EMO is usually to choose a single compromise solution from the approximation set presented to them. Although there may be a potentially infinite number of Pareto-optimal solutions in the global trade-off surface, in practice the DM will usually only be interested in a small subset of these. Therefore, allowing the DM to focus the optimisation process on relevant areas of the search space both increases the efficiency of the search effort and reduces the amount of irrelevant information the DM has to consider [75].

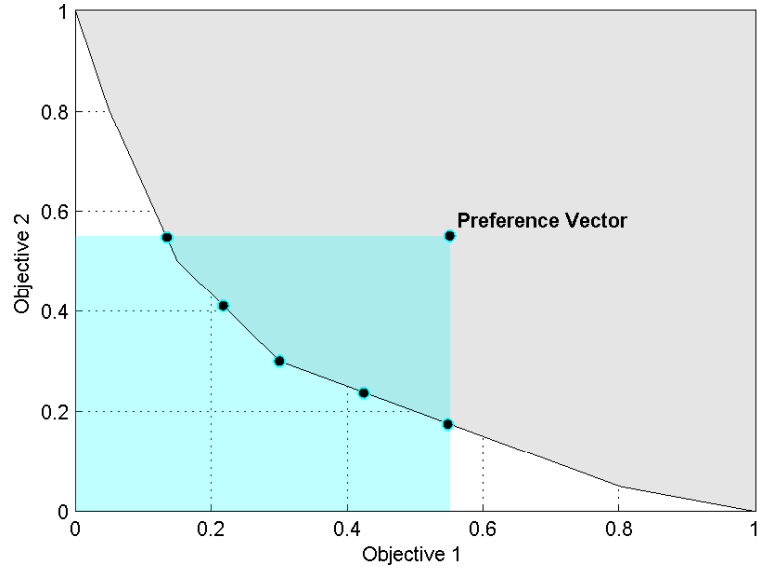


Figure 2.14: A Pareto-optimal approximation set containing five solutions with ideal pertinence.

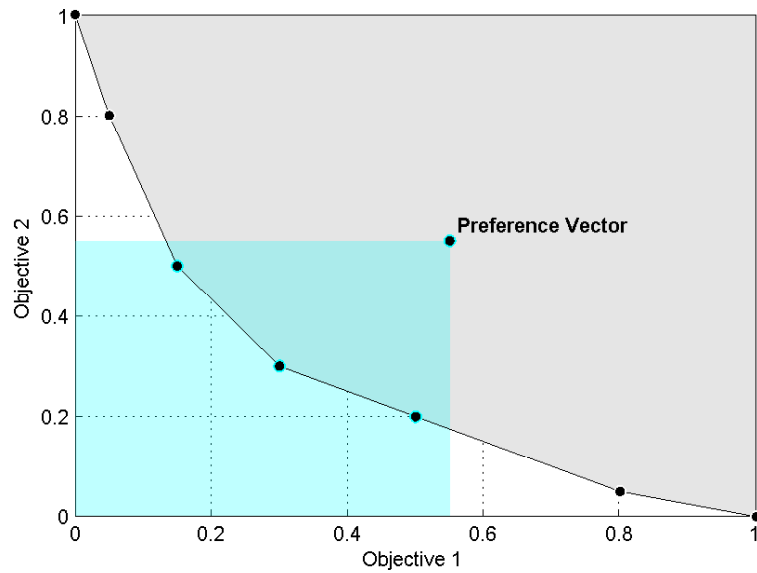


Figure 2.15: A Pareto-optimal approximation set containing seven solutions with undesirable pertinence.

A prerequisite for this type of convergence is the articulation of preferences by the DM. The preferences of a DM can be incorporated into the optimisation process in three ways:

- ***A priori***, in which preferences are defined before the search.
- ***A posteriori***, in which the DM selects a solution after completion of a search.
- **Progressively**, involving interaction with the DM during execution of the search.

A posteriori methods of preference articulation involve the DM selecting a compromise solution from the global approximation set of Pareto-optimal solutions found at the end of the optimisation process, whilst *a priori* and progressive preference articulation methods aim to achieve a good representation of the trade-off surface in the ROI of the DM. The key advantage of *a priori* and progressive preference articulation methods is the reduction in the size of the search space explored by the optimiser because the search is focussed on a sub-set of the global trade-off surface.

In *a priori* articulation of preferences the DM expresses their preferences before the start of the optimisation process. However, often the DM may not be sure of their preferences prior to optimisation, and by stating their preferences *a priori*, the DM may not investigate some areas of the search space which potentially deserve attention. A better method is often progressive articulation of preferences, which enables the DM to alter their preferences during the optimisa-

tion process and thus incorporate information that only becomes available during the search process [76] (such as the exact nature of trade-offs between objectives).

One of the first schemes for progressive preference articulation in EMO algorithms was introduced by [77], and extended the Pareto-based ranking scheme used in the Multiple Objective Genetic Algorithm (MOGA) [78] to allow preferences to be expressed throughout the run of an EMO algorithm. These preferences were then used in a modified version of dominance which combines the concept of Pareto-optimality with a preference operator to rank the candidate solutions according to both preference information and Pareto dominance. This progressive preference articulation method has been used in a wide variety of engineering applications such as the optimisation of robust control strategies for gasifier power plants [79], and the design of lateral stability controllers for aircraft [80].

More recently, the Reference-point-based Nondominated Sorting Genetic Algorithm II (R-NSGA-II) presented in [81], combines a preference based strategy with an EMO methodology, in order to demonstrate how a preferred set of solutions near a number of reference points can be found simultaneously. The paper suggests two approaches for the incorporation of preferences: a modified EMO procedure based on the NSGA-II; and a predator-prey approach based on an original grid based procedure [82]. Both approaches appeared to perform well, with the modified NSGA-II approach performing better overall.

The Preference-Based Evolutionary Algorithm (PBEA) was introduced in [83] in order to address the short-comings of not having preference information in the solution process. The algorithm uses the Indicator-Based Evolutionary Algorithm (IBEA) introduced in [84] as a base, in combination with a binary indicator which

has been modified with an achievement function (based on a reference point) which directly represents the preference information. The experimental results were obtained primarily from bi-objective synthetic test functions from the ZDT synthetic test suite. The authors suggest that the incorporation of preferences results in more relevant approximations throughout the optimisation process.

These preference driven multi-objective optimisers offer promising results, and suggest that the incorporation of DM preferences into multi-objective search can reduce computational cost of the optimisation process and improve the pertinence of the final approximation set presented to the DM.

The approaches introduced appear to lack rigorous benchmarking consisting of many test suites, real-world problems, and test-cases. The approaches introduced also involve the tight integration of the preference method into an existing EMO method. Instead, it would be desirable to have a preference articulation operator which is designed for portability. Such a portable preference articulation operator could be incorporated into any multi-objective optimiser, as different optimisers are more suitable for different problems, and some optimisers become redundant after years of further research.

2.7 State of the art Evolutionary Multi-Objective Optimisation Algorithms

The EMO literature contains descriptions of many different EMO algorithms, some of these algorithms are new designs and introduce new concepts, where as others are variants or combinations of existing EMO algorithms. In this section, four well-regarded and popular EMO algorithms are introduced and described briefly: Section 2.7.1 introduces NSGA-II, Section 2.7.2 introduces PAES, Section 2.7.3 introduces MO-CMA-ES, and Section 2.7.4 introduces MOEA/D and MOEA/D-DRA.

2.7.1 Non-dominated Sorting Genetic Algorithm II

NSGA-II was introduced in [35, 74] as an enhancement to NSGA [85] in order to address some problems with the original algorithm. In particular, NSGA-II uses an enhanced approach to selection whereby a candidate solution is ranked using a much faster non-dominated sorting scheme paired with a crowded comparison operator (described in Section 2.5.2). Not only does NSGA-II take into account the non-domination rank of a candidate solution but also its crowding distance³ during the selection for variation and survival process. Unlike PAES (introduced in Section 2.7.2) which uses an external archive, NSGA-II uses a simple $(\mu + \lambda)$ selection scheme in the survival process. In the EMO literature, NSGA-II is widely used in pairwise comparisons, and as an optimiser against which new optimisers are compared to.

³A measure of the density of solutions surrounding a particular solution in the objective space.

The algorithm begins by initializing a population of randomly generated solutions, and then evaluates the objective values for every solution. Then the generational loop begins, where each solution in the current population is assigned a rank based on its Pareto dominance using the fast non-dominated sorting scheme, and then the crowding distance between every solution of the same rank is calculated. Selection for variance is then carried out using the crowded comparison operator and binary tournament selection. Genetic operators such as the Simulated Binary Crossover (SBX) for recombination and polynomial mutation [86] are then used to introduce variance in the offspring population, these new offspring solutions are then evaluated. The parent population and offspring population are combined and selection for survival takes place, where elitism is ensured because the best solutions from the parent and offspring population are in the population which is subjected to selection. The population for the next generation is populated by solutions from each rank (from best to worse) until the population capacity has been met. If adding all solutions from a rank exceeds the population capacity, then solutions are selected based on their crowding distance in descending order until the capacity is met. This process is repeated for every generation until some termination criteria is met.

2.7.2 Pareto Archived Evolution Strategy

The archiving approach to elitism is typified by PAES which proposes a conceptually simple EMO algorithm capable of producing a diverse approximation set with close proximity to the true Pareto-optimal front [34]. PAES uses a (1+1) ES in conjunction with a novel bounded Pareto archive and AGA (described in Section 2.5.2). This bounded Pareto archive stores only non-dominated solutions

that are discovered during the search, and a non-dominated candidate solution is compared to the archive before it is accepted as a current solution. Once the archive is full, a grid system (whereby the objective space currently covered by non-dominated solutions is divided up into a predefined number of partitions) is used to decide which archived solution to remove to allow space for a new non-dominated solution to be added. Using a set of rules for grid and archive management, diversity is achieved amongst the archive. The execution life-cycle for PAES is illustrated in Figure 2.16. Variations of the AGA approach used in PAES have been used in other EMO algorithms; for example, in the Pareto Envelope-based Selection Algorithm (PESA) [68].

The Pareto Archived Evolution Strategy (PAES) [34] is a simple (1+1) evolution strategy whereby a single parent solution produces a single offspring. PAES uses an archive (with an upper bound on its size) which contains all the non-dominated solutions which have been found during the optimisation process. This archive implements the elitism concept and plays the role of a reference set. The performance of the mutated solution (offspring solution) is assessed by comparing it to the performance of the solutions in the reference set. However, the major feature of PAES is its strategy for promoting diversity in the approximation set. PAES uses an adaptive hyper-grid system in the objective space to divide it into several non-overlapping hyper-boxes. The belonging of a certain solution to a certain region in the hyper-box is determined by the solution's objective values which define the solution's coordinates. In the case where an offspring solution is non-dominated by the reference set, a crowding measure based on the number of solutions residing in a certain grid location is applied to determine whether the offspring solution is to be accepted or not. The major advantage of this diversity

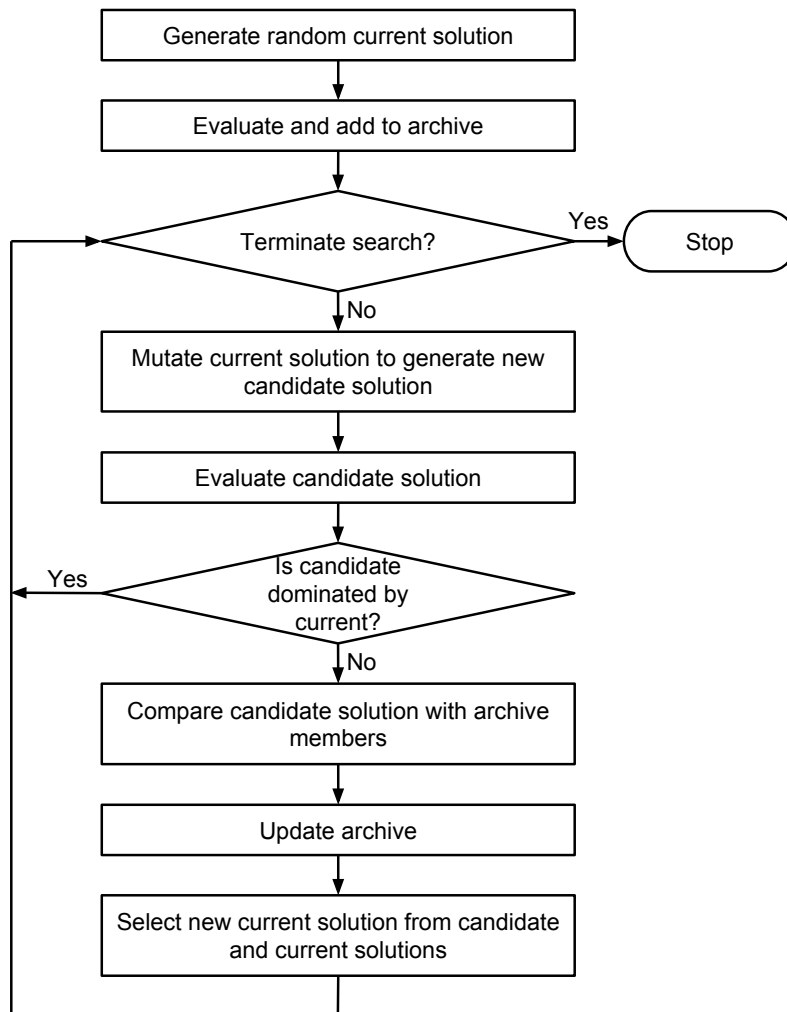


Figure 2.16: Execution life-cycle for the PAES optimisation process.

maintenance technique is that it does not require setting any additional parameters such as the niche size parameter-share in other fitness sharing approaches.

2.7.3 Multi-Objective Covariance Matrix Adaptation Strategy

The MO-CMA-ES is a multi-objective implementation of the powerful single objective CMA-ES designed to solve multi-objective optimisation problems [36]. The MO-CMA-ES maintains a population of elitist solutions that adapt their search strategy depending on the shape of the underlying search landscape. There are two variations of the MO-CMA-ES: the *s*-MO-CMA-ES which achieves diversity using the contributing hyper-volume measure (or *s*-metric) introduced by [87], and the *c*-MO-CMA-ES which achieves diversity using the crowding-distance measure introduced in NSGA-II. Whilst initial results have shown that MO-CMA-ES is extremely promising, it is as yet predominately untested on real-world engineering problems. Some results show that MO-CMA-ES struggles to converge to good solutions on problems with many deceptive locally Pareto-optimal fronts - a feature that can be common in real-world problems [88]. The MO-CMA-ES execution life-cycle has been illustrated in Figure 2.17.

In the original MO-CMA-ES, a mutated offspring solution is considered to be successful if it dominates its parent. In contrast, [88] introduces a new MO-CMA-ES variant which considers a solution successful if it is selected to be in the next parent population, and conducts a comparison of MO-CMA-ES variants on synthetic test functions consisting of up to three objectives, making it the first time MO-CMA-ES has been evaluated on synthetic test functions consisting of more than two objectives. NSGA-II with the hypervolume indicator as a second-level sorting criterion was also considered, and for the first time MO-CMA-ES is shown to out-perform it. MO-CMA-ES with the improved update rule is strongly

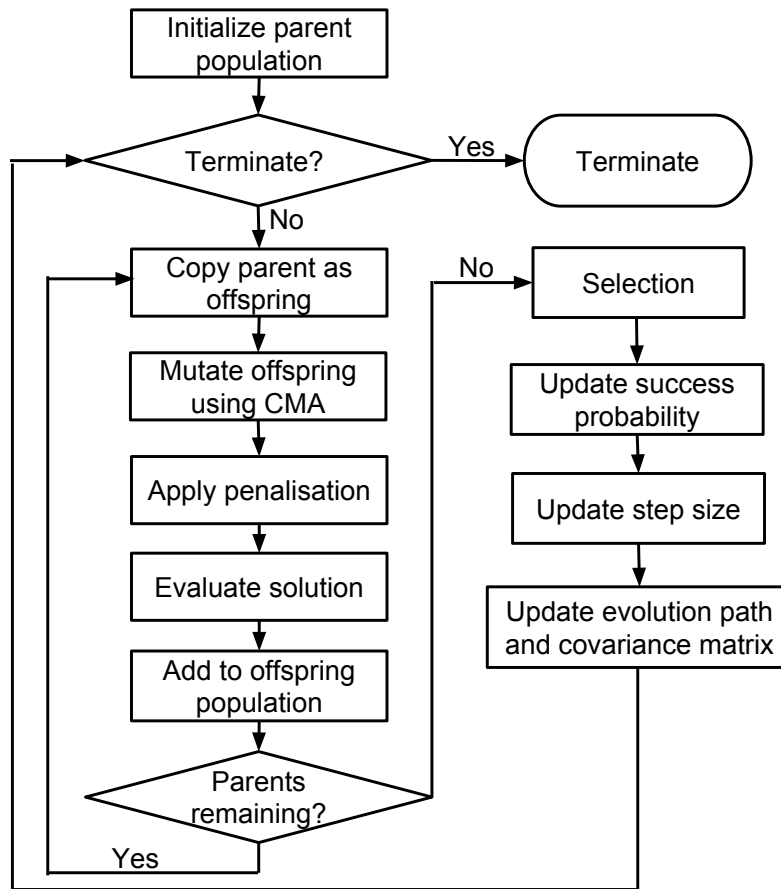


Figure 2.17: Execution life-cycle for the MO-CMA-ES optimisation process.

recommended and is shown to offer better performance than the original, and will be referred to as MO-CMA-ES herein.

MO-CMA-ES is a powerful multi-objective optimiser (empirically evaluated in [36, 89, 88]), but suffers from computational infeasibility when applied to problems consisting of many objectives, this is because of its reliance on the contributing hypervolume indicator as a second-level sorting criterion (described in Section 2.5.2).

2.7.4 Multi-Objective Evolutionary Algorithm Based on Decomposition

The decomposition approach to EMO is typified by the Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D) [37] which decomposes a multi-objective optimisation problem into a number of scalar optimisation sub-problems and optimises them simultaneously. Each of the sub-problems is optimised by only using information from its several neighbouring sub-problems, which results in MOEA/D having a lower computational complexity at each generation when compared to NSGA-II. The use of the Tchebycheff approach [90] to decomposition in MOEA/D is suggested in [37, 91], where MOEA/D either outperforms or performs comparably to NSGA-II on a range of synthetic test functions.

The performance of a new version of MOEA/D, named MOEA/D with Dynamical Resource Allocation (MOEA/D-DRA), was presented as part of the *Special Session on Performance Assessment of Constrained / Bound Constrained Multi-Objective Optimization Algorithms held at CEC09 in Norway*, the synthetic test functions and rules of which are described in [92]. In the previous version of MOEA/D, all sub-problems were treated equally in regards to the computational effort which they were allocated. However, there may be variance in computational difficulty amongst these sub-problems, and because of this MOEA/D-DRA uses dynamic resource allocation to assign different amounts of computational effort to different sub-problems, this is based on the computation of a utility value π^i for each of the sub-problems i . The MOEA/D-DRA algorithm is described in Algorithm 1, with further detail available in [93].

Algorithm 1 MOEA/D-DRA algorithm

Step 1: Initialisation

Step 1.1: Calculate the Euclidean distances between any two weight vectors and then find the closest weight vectors to each weight vector.

Step 1.2: Generate an initial population by uniformly sampling from the weight space.

Step 1.3: Initialize algorithm parameters.

Step 2: Selection of sub-problems using 10-tournament selection based on the utility value π^i , sub-problem indexes are selected.

Step 3: Variation for every sub-problem selected in Step 2:

Step 3.1: Selection of mating/update range.

Step 3.2: Reproduction by generating a new solution using a differential evolution operator, then performing mutation to produce a new solution.

Step 3.3: If an objective value does not conform to the problem boundaries, its value is reset to be a randomly selected value within the boundary.

Step 3.4: Update of Solutions.

Step 4: Terminate optimisation process if stopping criteria is satisfied.

Step 5: Continue the generational loop, continue from Step 2.

Neighbourhood relations amongst sub-problems are defined based on the distances amongst their weight vectors. These weight vectors are generated as random uniformly distributed values, an example of generated weight vectors has been illustrated in Figure 2.18.

Overall, MOEA/D-DRA's performance suggests it is a powerful EMO algorithm capable of producing approximation sets with good proximity and diversity on synthetic test problems which contain optimisation difficulties such as complex Pareto-optimal sets.

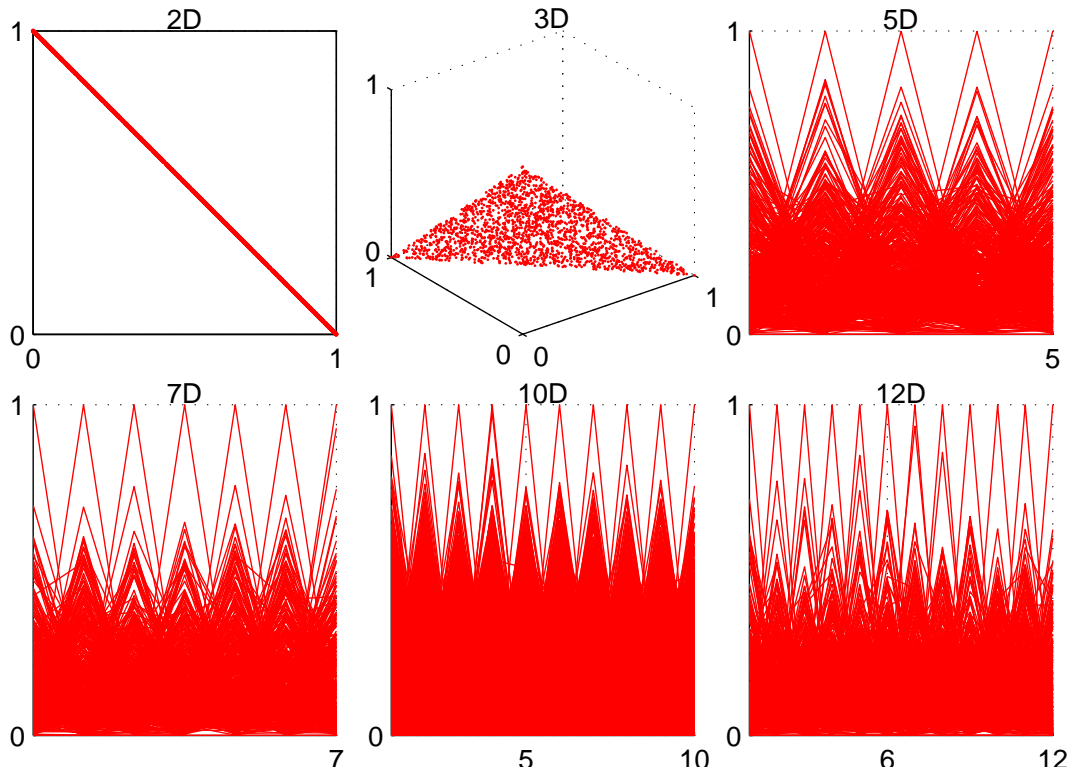


Figure 2.18: Uniformly distributed random weight values generated for 2, 3, 5, 7, 10 and 12 objective problems, for use with MOEA/D-DRA.

2.8 Objective Functions

In mathematics, optimisation is concerned with the selection of optimal solutions to objective functions. An objective function consists of input arguments referred to as problem variables (or genotype) which are computed by one or many mathematical functions to determine the objective value (or phenotype).

Real-world optimisation problems are divided into one (in the case of single objective optimisation) or many (in the case of multi-objective optimisation) objective functions in order to be optimised by an optimisation algorithm. The difficulty of convergence can be reduced by the bounding of problem variables as

this reduces the size of the search domain.

An example objective function can be described mathematically as:

$$\begin{aligned}
 f_1 &= x_1 \\
 f_2 &= g \left(1.0 - \sqrt{\frac{f_1}{g}} \right) \\
 g(x_2, \dots, x_n) &= 1.0 + \frac{9}{n-1} \sum_{i=2}^n x_i \\
 0 &\leq x_i \leq 1, i = 1, \dots, n
 \end{aligned} \tag{2.5}$$

where $\langle x_1, \dots, x_n \rangle$ are the problem variables, f_1 is the first objective value and f_2 is the second objective value for this bi-objective synthetic test function named ZDT1, taken from the ZDT multi-objective test suite (described in Section 2.9.1). This particular objective function is, by design, scalable up to any number of problem variables but is restricted to two problem objectives.

Synthetic test functions which are developed for the purpose of testing the robustness of an optimiser are typically computationally inexpensive and have short execution times. In contrast, real-world problems which have been encapsulated within an objective function in order to be used by an optimiser are often computationally expensive and have long execution times. This is because synthetic test functions are often mathematical equations which aim to cause difficulty for an optimiser when searching for problem variables that produce optimal objective values, whereas real-world problems often involve computationally expensive simulations in order to arrive at the objective values.

2.9 Multi-Objective Test Suites

In order to determine an EMO algorithm robust when solving problems consisting of multiple objectives, its performance must be assessed on the optimisation of test functions which are created for the purpose of testing. These problems may also be used to systematically compare two or more EMO algorithms. These test function often have a scalable number of problem objectives and problem variables as well as a complex Pareto shape, and aim to test algorithms on their ability to converge to an approximation set in the presence of optimisation difficulties which are often present in real-world optimisation problems.

The four most popular multi-objective test suites in the literature are the bi-objective ZDT test suite proposed in [38], the scalable multi-objective DTLZ test suite proposed in [39], the multi-objective CEC09 competition test suite proposed in [92], and the scalable multi-objective WFG test suite proposed in [40]. These test suites aim to incorporate a combination of features in each test problem that an EMO algorithm may potentially find difficult to overcome during the optimisation process, allowing for the assessment of an EMO algorithm's ability to converge toward the true Pareto-optimal front in the presence of such difficulties.

2.9.1 The ZDT Test Suite

The ZDT test suite [38] contains six synthetic test functions which were considered to provide sufficient complexity in the benchmarking of multi-objective optimisers. The test functions are named ZDT1 through to ZDT6, with each test function incorporating a feature that is known to cause the EMO process diffi-

culty when attempting to converge toward the true Pareto-optimal front, and in the maintenance of diversity in the approximation set. Each test function is concerned with the minimisation of two problem objectives, and has been described in the following:

- **ZDT1** is by default a 30 variable problem with a convex Pareto-optimal front;
- **ZDT2** is by default a 30 variable problem with a non-convex Pareto-optimal front;
- **ZDT3** is by default a 30 variable problem with a Pareto-optimal front consisting of non-contiguous convex parts;
- **ZDT4** is by default a 10 variable problem which tests the ability to handle multi-modality with 21^9 local Pareto-optimal fronts;
- **ZDT5** is typically not considered when designing experiments to benchmark modern EMO algorithms due to its requirement of binary encoded problem variables;
- **ZDT6** is by default a 10 variable problem with solutions non-uniformly distributed along the Pareto-optimal front, with the diversity of solutions decreasing near the Pareto-optimal front.

The true Pareto-optimal fronts of the test problems from the ZDT test suite have been plotted and presented in Figure 2.19.

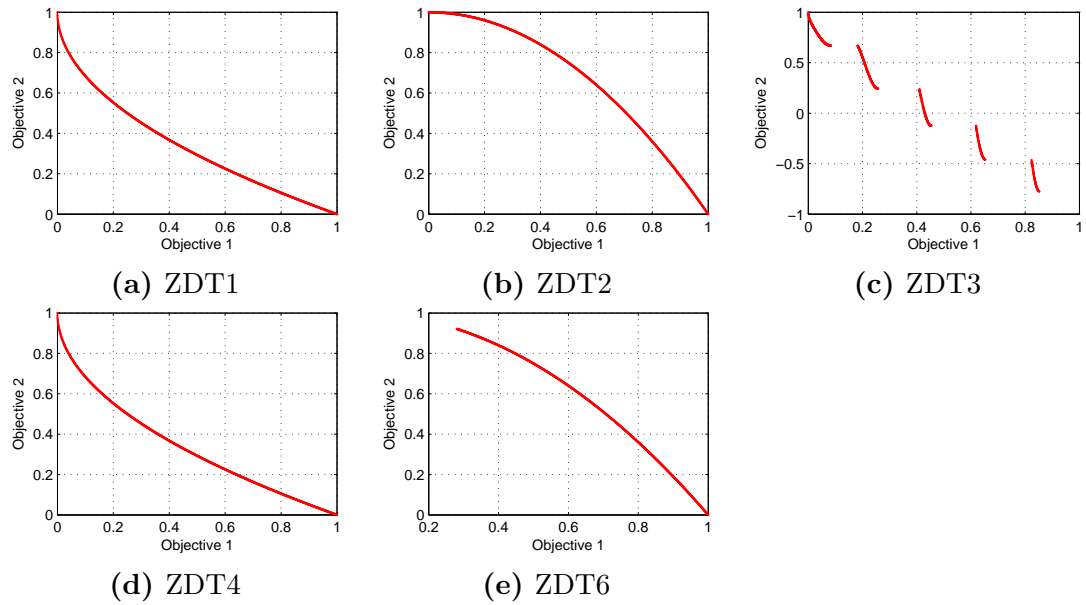


Figure 2.19: Plots of the Pareto-optimal fronts for all ZDT test functions excluding ZDT5, from the ZDT test suite.

The ZDT synthetic test suite has been used for the performance assessment of algorithms in much of the multi-objective optimisation and evolutionary computation literature (e.g. [94, 95, 81, 96, 97]).

2.9.2 The DTLZ Test Suite

Most research on EMO algorithms often employed simple or non-scalable test problems in their benchmarking and comparison. In order to test EMO algorithms on their ability on multi-objective test problems, the DTLZ test suite [39] has been developed with both scalable problem variables and for the first time scalable problem objectives. The test suite consists of seven scalable multi-objective synthetic test problems concerned with minimisation, and have been described in the following:

- **DTLZ1** is by default a 7 variable simple test problem, with objective function values lying on a linear hyperplane. The difficulty is in converging to the Pareto-optimal hyperplane in a search space which contains 11^{k-1} (k recommended to be 5) local Pareto-optimal fronts, each of which can deceive an EMO algorithm and result in premature convergence;
- **DTLZ2** is by default a 12 variable test problem with a spherical Pareto-optimal front, which tests an EMO algorithm's performance when optimising three or more objectives;
- **DTLZ3** is by default a 12 variable test problem based on DTLZ2, which tests an EMO algorithm's ability to converge to a global Pareto-optimal front by altering the DTLZ2 function to introduce many local Pareto-optimal fronts;
- **DTLZ4** is by default a 12 variable test problem based on DTLZ2, which tests for an EMO algorithm's ability to maintain good solution diversity and distribution by altering the DTLZ2 function's meta-variable mapping,

allowing for a dense set of solutions near the $f_m - f_1$ plane. When optimising DTLZ4, the final population is significantly dependant on the initial population;

- **DTLZ5** is by default a 12 variable problem which tests an EMO algorithm's ability to converge to a degenerated curve;
- **DTLZ6** is by default a 12 variable problem which introduces 2^{m-1} disconnected Pareto-optimal regions in the search space, this tests an EMO algorithm's ability to maintain sub-populations in different Pareto-optimal regions;
- **DTLZ7** is by default a 22 variable problem which introduces a Pareto-optimal front consisting of a combination of a straight line and a hyper-plane, this tests an EMO algorithm's ability in finding solutions in both of these regions whilst maintaining good solution diversity and distribution.

The true Pareto-optimal fronts of two-objective and three-objective instances of the test problems from the DTLZ test suite have been plotted and presented in Figure 2.20 and Figure 2.21 respectively.

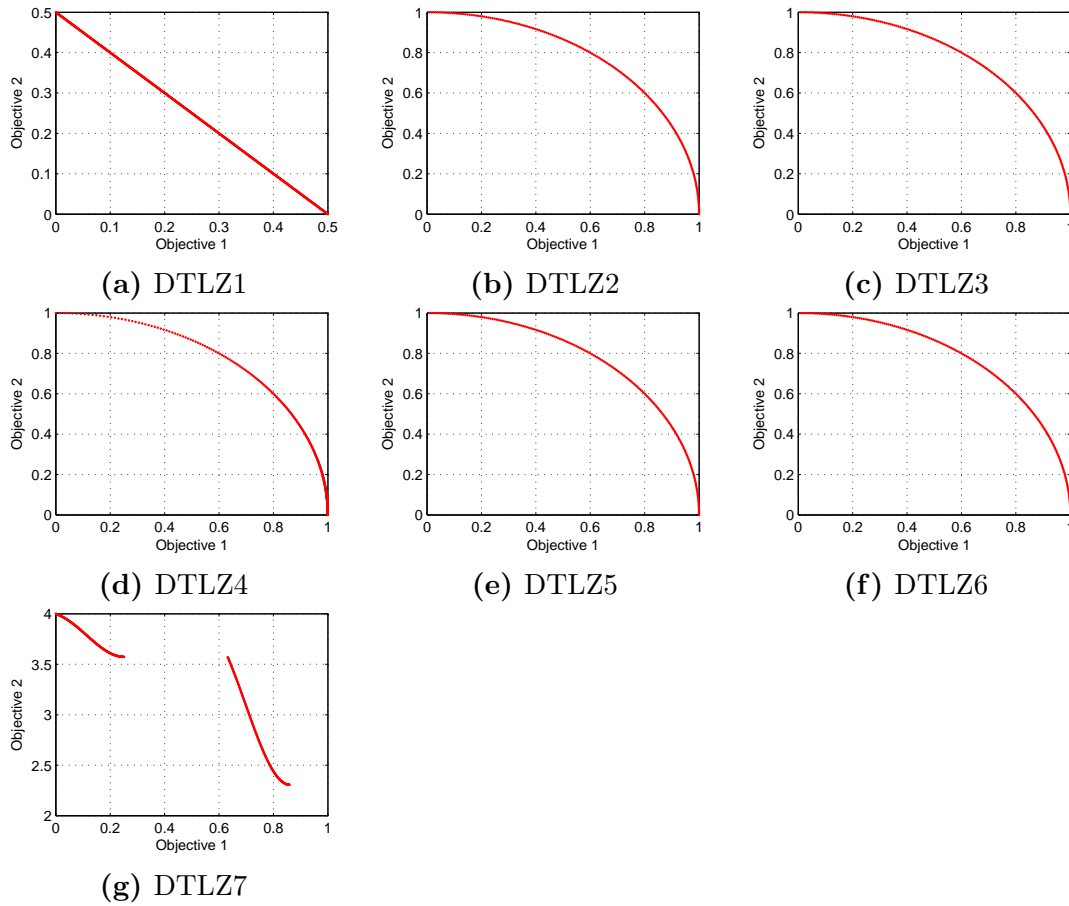


Figure 2.20: Plots of the true Pareto-optimal fronts for the bi-objective DTLZ test functions from the DTLZ test suite.

The DTLZ synthetic test suite has been used for the performance assessment of algorithms in much of the multi-objective optimisation and evolutionary computation literature (e.g. [98, 99, 100, 101, 102]).

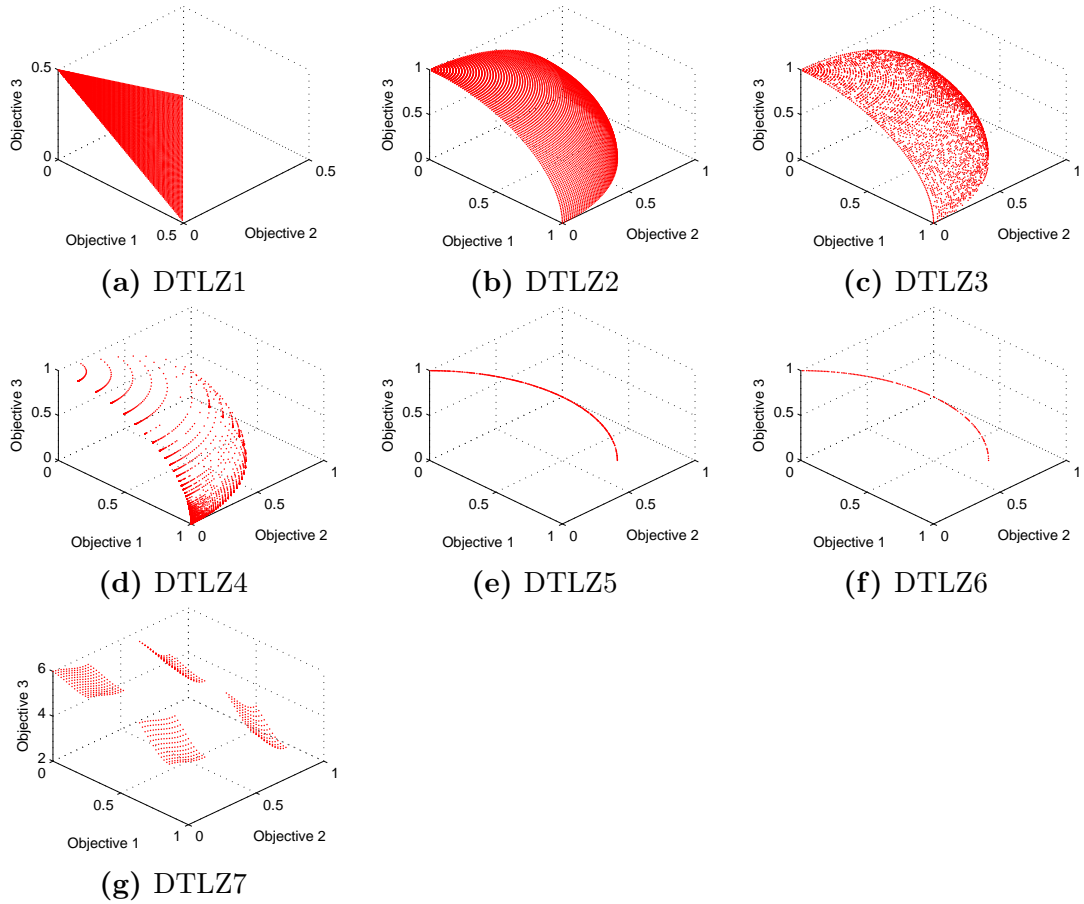


Figure 2.21: Plots of the true Pareto-optimal fronts for the three-objective DTLZ test functions from the DTLZ test suite.

2.9.3 The WFG Tool-kit

Through the analysis of existing test suites, [40] suggests existing test problems are poorly constructed and poorly represent non-separability, and proposes a flexible tool-kit which can be used to construct well-designed synthetic test problems. The tool-kit is demonstrated with the construction of nine multi-objective test problems (WFG1 through to WFG9) referred to as the WFG test suite, with test problems consisting of scalable problem variables and problem objectives. All test problems in the WFG test suite consist of: Pareto-optimal fronts which

have dissimilar trade-off magnitudes, problem variables which have domains of dissimilar magnitude, and Pareto-optimal fronts which are not degenerate (with the exception of WFG3).

The WFG test suite has been described in the following:

- **WFG1** is a separable uni-modal test problem, with a polynomial/flat bias, and a convex mixed geometry;
- **WFG2** is a non-separable test problem with a uni-modal and multi-modal variant, and a convex disconnected geometry;
- **WFG3** is a non-separable uni-modal test problem, with a linear and degenerate geometry consisting of a one dimensional Pareto-optimal front;
- **WFG4** is a separable and multi-modal test problem with a concave geometry;
- **WFG5** is a separable test problem with deceptive modality, and a concave geometry;
- **WFG6** is a non-separable uni-modal test problem with a concave geometry;
- **WFG7** is a separable uni-modal test problem with a parameter dependent bias and a concave geometry;
- **WFG8** is a non-separable uni-modal test problem with a parameter dependent bias and a concave geometry;
- **WFG9** is a non-separable test problem with a multi-modal and deceptive modality, with a parameter dependent bias and a concave geometry.

The true Pareto-optimal fronts of two-objective and three-objective instances of the test problems from the WFG test suite have been plotted and presented in Figure 2.22 and Figure 2.23 respectively.

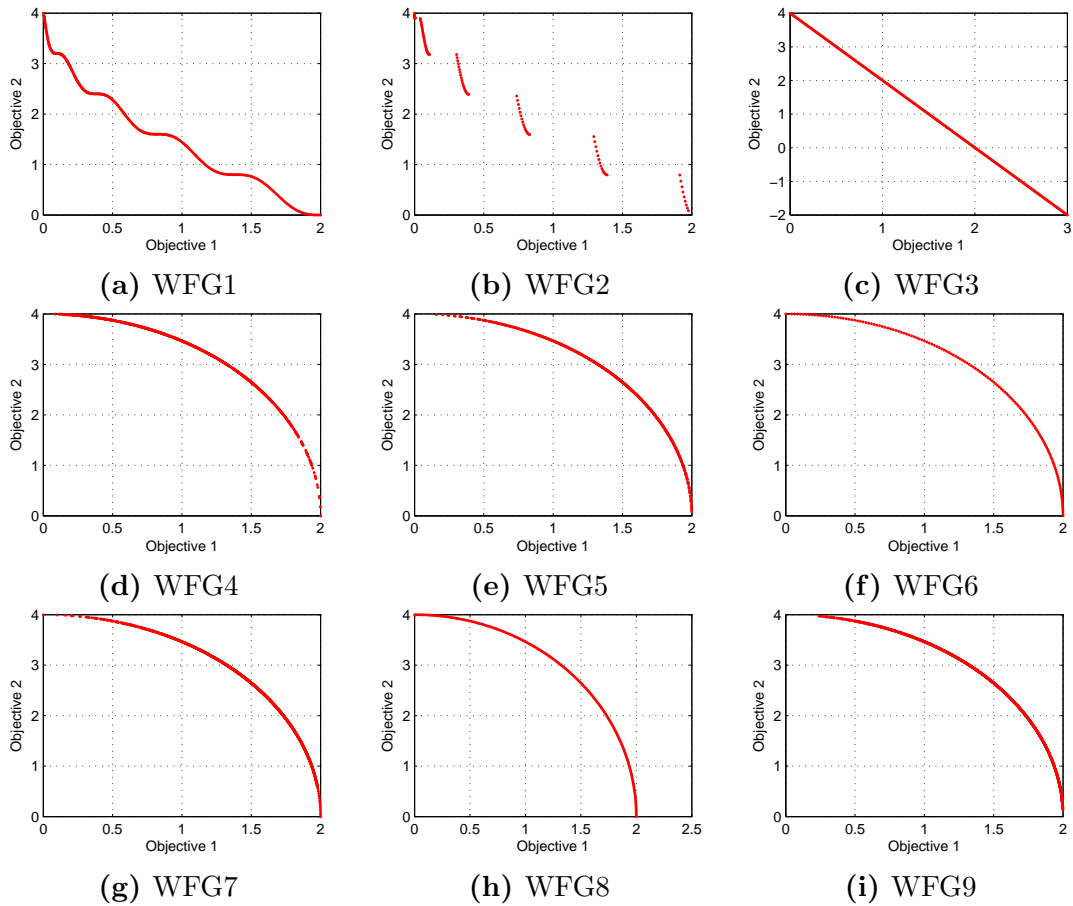


Figure 2.22: Plots of the true Pareto-optimal fronts for the bi-objective WFG test functions from the WFG test suite.

The WFG synthetic test suite has been used for the performance assessment of algorithms in much of the multi-objective optimisation and evolutionary computation literature (e.g. [103, 104, 105, 106, 107]).

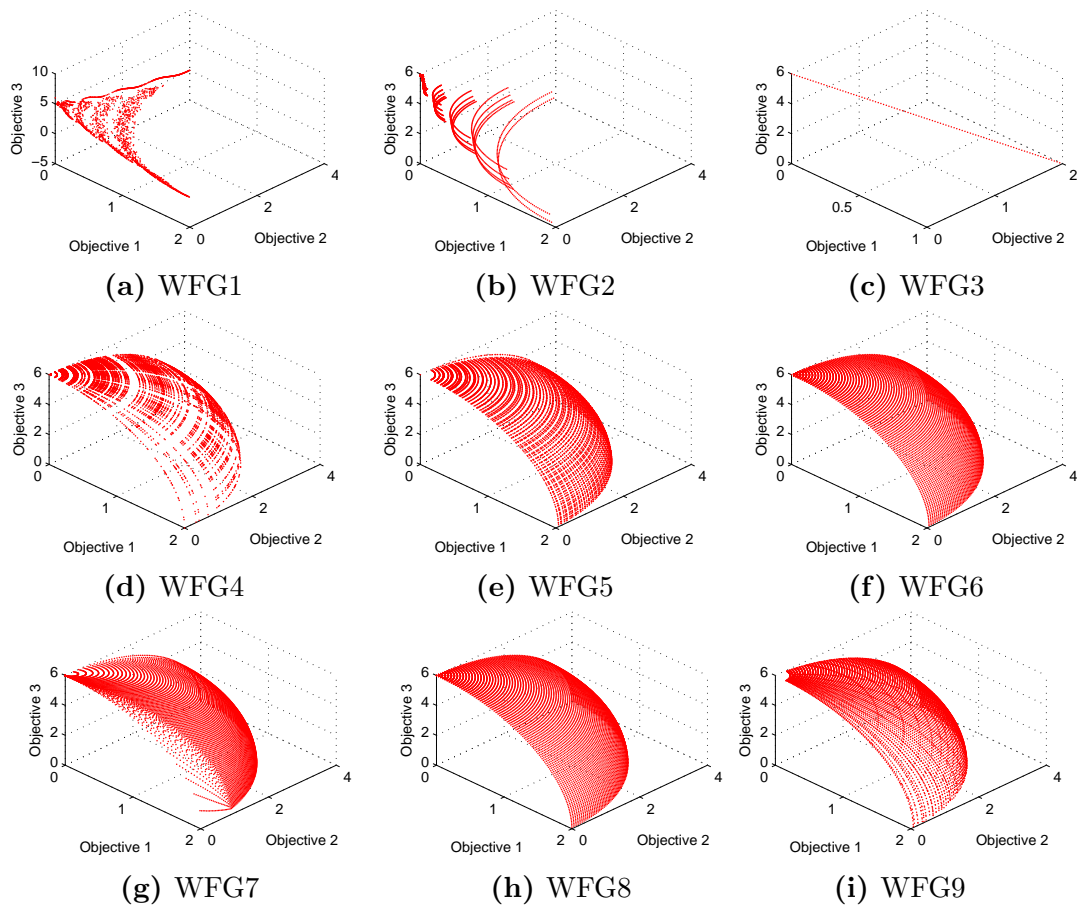


Figure 2.23: Plots of the true Pareto-optimal fronts for the three-objective WFG test functions from the WFG test suite.

2.9.4 The CEC2009 Competition Test Suite

The CEC09 test suite proposed in [92] consists of a series of constrained and unconstrained multi-objective test functions designed for the *2009 IEEE Congress on Evolutionary Computation* competition. Each test problem is intended to resemble complicated real-life problems and declares a required number of problem objectives and problem variables. The unconstrained test functions UF1 to UF10 are of interest from this test suite, with UF1 to UF7 consisting of bi-objective test problems (illustrated in Figure 2.24) and UF8-UF10 consisting of three-objective test problems (illustrated in Figure 2.25), all of which are concerned with their minimisation and have 30 problem variables.

The unconstrained test problems from the CEC09 test suite have been described in the following:

- **UF1** and **UF2** share the same convex Pareto-optimal front, but their Pareto-optimal sets consist of various non-linear complex curves in the decision space;
- **UF3** has the same Pareto-optimal front as UF1 and UF2, but its Pareto-optimal set is a simple curve in the decision space. The test function contains many Pareto-optimal fronts which tests an EMO algorithm's global search ability;
- **UF4** has a non-convex Pareto-optimal front, but its Pareto-optimal set consists of various non-linear complex curves in the decision space;
- **UF5** has a Pareto-optimal front consisting of 21 diversely distributed solutions which lie on a linear hyperplane;

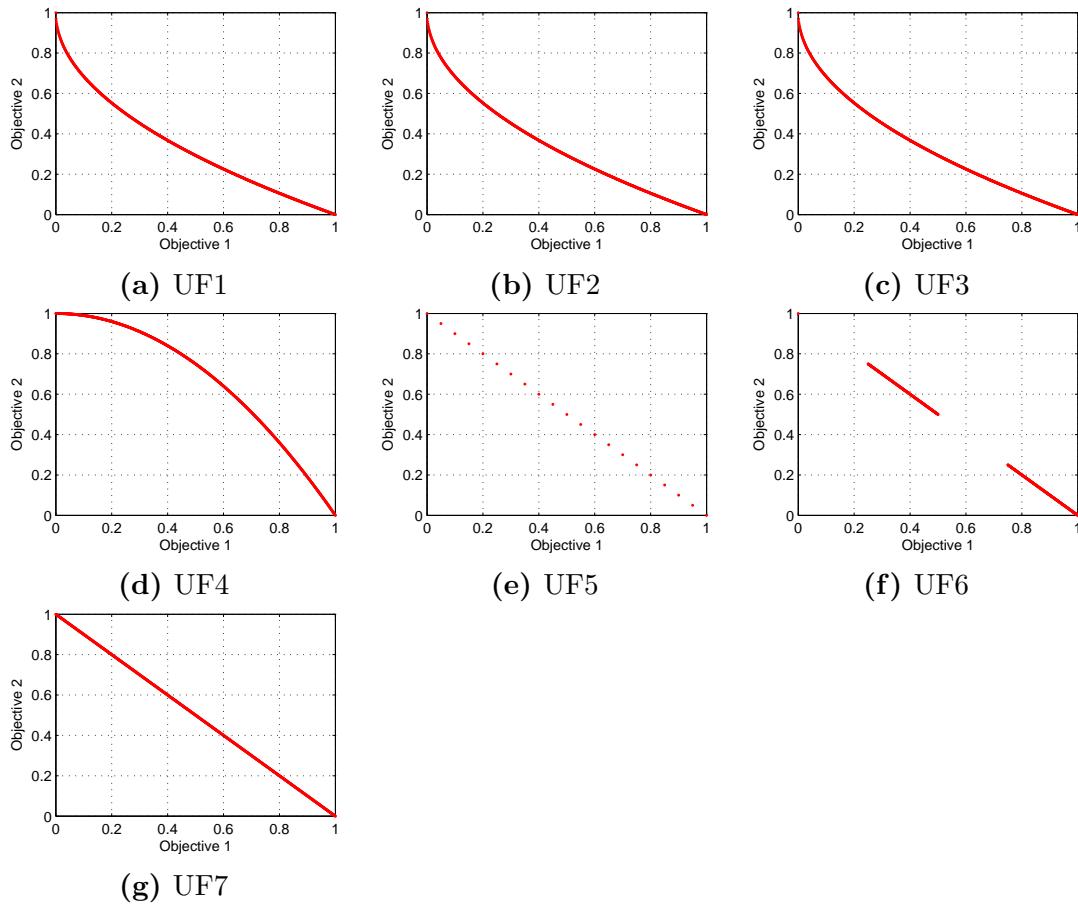


Figure 2.24: Plots of the true Pareto-optimal fronts for the bi-objective CEC09 test functions from the CEC09 competition test suite.

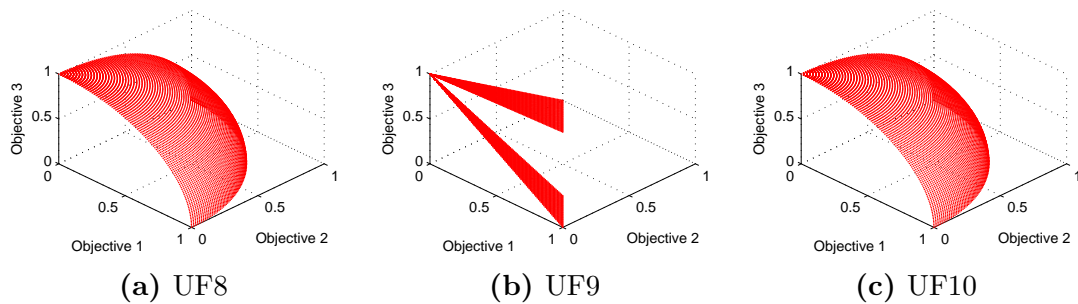


Figure 2.25: Plots of the true Pareto-optimal fronts for the three-objective CEC09 test functions from the CEC09 competition test suite.

- **UF6** has a disjoint Pareto-optimal front of one isolated point $(0, 1)$ and two disconnected parts, which tests an EMO algorithm's ability in finding all the regions of the Pareto-optimal front whilst maintaining good diversity. The Pareto-optimal set consists of thinly distributed disjoint complex curves in the decision space;
- **UF7** has a Pareto-optimal front of solutions which lie on a linear hyperplane, with a Pareto-optimal set which is similar to the one used in UF1;
- **UF8** and **UF10** have a spherical Pareto-optimal front, with a non-linear 2D surface for a Pareto-optimal set;
- **UF9** has a Pareto-optimal front and Pareto-optimal set which have two parts.

The CEC09 competition also has strict configuration guidelines which specify a budget of 300,000 function evaluations, 30 independent executions of the candidate algorithm, the same algorithm parameter settings for test problems consisting of the same number of problem objectives, and the use of the IGD measure for performance assessment (described in Section 2.10.3).

The CEC09 synthetic test suite has been used for the performance assessment of algorithms in much of the multi-objective optimisation and evolutionary computation literature (e.g. [108, 109, 110, 111, 112]).

2.10 Performance Assessment

There have been many contributions in the EMO literature regarding the formulation of performance criteria and methods of performance assessment of EMO algorithms. Selecting a relevant and sufficient method of performance assessment is a necessity when evaluating or comparing EMO algorithms. These methods of performance assessment can be used to gauge an EMO algorithm's performance in regards to the proximity, diversity, and pertinence of the final approximation set. Most methods of performance assessment rely on the availability of a reference front. This dependency is not feasible in real-world problem scenarios, more so in the case where the problem is new and has not yet been subjected to a method of optimisation.

In this chapter, a number of performance metrics are introduced. Section 2.10.1 describes the Hypervolume Indicator metric, Section 2.10.2 describes the Generational Distance metric, and Section 2.10.3 describes the Inverted Generational Distance metric. Section 2.10.4 describes methods of non-parametric testing for statistical analysis of EMO algorithms, and Section 2.10.5 describes the selection of sufficient sample sizes.

2.10.1 The Hypervolume Indicator

The hypervolume indicator (or s -metric) is a performance metric for indicating the quality of a non-dominated approximation set, introduced by [87] where it is described as the “*size of the space covered or size of dominated space*”. It can be defined as [88]:

$$S_{f^{ref}}(X) = \Lambda \left(\bigcup_{X_n \in X} [f_1(X_n), f_1^{ref}] \times \cdots \times [f_m(X_n), f_m^{ref}] \right) \quad (2.6)$$

Where $S_{f^{ref}}(X)$ resolves the size of the space covered by an approximation set X , $f^{ref} \in \mathbb{R}$ refers to a chosen reference point and $\Lambda(\cdot)$ refers to the Lebesgue measure [113]. This has been illustrated in Figure 2.26 in two-dimensional objective space (to allow for easy visualisation) with a population of 3 solutions.

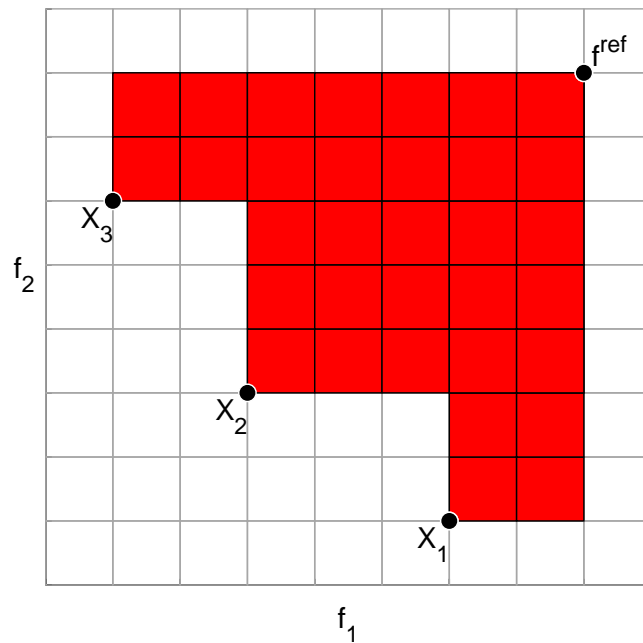


Figure 2.26: An example of the hypervolume indicator in two-dimensional objective space.

The hypervolume indicator is appealing because it is scaling independent and requires no prior knowledge of the true Pareto-optimal front, this is important when working with real-world problems which have not yet been solved. The hypervolume indicator is currently used in the field of multi-objective optimisation as both a proximity and diversity performance metric, and also in the decision making process [114, 115].

A reference vector is required to calculate the hypervolume indicator value. When used for pairwise or multiple comparison of EMO algorithms, this reference vector must be the same, otherwise the resulting hypervolume indicator values are not comparable. This reference vector can be approximated as large values for each problem objective in order for all objective values in any approximation set to be within the reference vector. A more accurate method for selecting a reference vector, is to use the worst objective values from the union of approximation sets produced on a particular test problem for each algorithm which is being considered for comparison.

Various implementations of the hypervolume indicator have been presented in [116, 117, 118, 119, 120], all with the aim to speed up its calculation. The hypervolume indicator has been employed in the performance assessment of algorithms in much of the multi-objective optimisation and evolutionary computation literature (e.g. [8, 121, 122, 96, 106]).

2.10.2 The Generational Distance

The Generational Distance (GD) introduced in [123, 124] measures the proximity of the approximation set to the true Pareto-optimal front in objective space. The GD can be defined as:

$$GD = \frac{\sqrt{\sum_{i=1}^{n^*} d_i^2}}{n^*} \quad (2.7)$$

where n^* is the number of solutions in the approximation set, and d is the Euclidean distance (in objective space) between each solution in the approximation set and the nearest member of the true Pareto-optimal front. A GD value equal to zero indicates that all members of the approximation set are on the

true Pareto-optimal front, and any other value indicates the magnitude of the deviation of the approximation set from the true Pareto-optimal front.

The calculation of the GD is easy and the concept is intuitive, however, knowledge regarding the true Pareto-optimal front is required in order to form a reference set. The selection of solutions for the reference set will have an impact on the results obtained from the GD, and therefore the reference set must be diverse. In addition, the calculation of the GD can be computationally expensive when working with large populations or a high number of problem objectives.

The GD measure has been employed in the performance assessment of algorithms in much of the multi-objective optimisation and evolutionary computation literature (e.g. [125, 126, 127, 128, 129]).

2.10.3 Inverted Generational Distance

Following the suggestion of a reviewer, the Inverted Generational Distance (IGD) was introduced in [130] as an enhancement to the GD measure, measuring the proximity of the approximation set to the true Pareto-optimal front in objective space. The IGD can be defined as:

$$IGD = \frac{\sqrt{\sum_{i=1}^{n'} d_i^2}}{n'} \quad (2.8)$$

where n' is the number of solutions in the reference set, and d is the Euclidean distance (in objective space) between each solution in the reference set and the nearest solution in the approximation set. A GD value equal to zero indicates that all members of the approximation set are on the true Pareto-optimal front, and any other value indicates the magnitude of the deviation of the approximation set from the true Pareto-optimal front. This implementation of the GD solves an

issue in its predecessor so that it will not rate an approximation set with a single solution on the reference set as better than an approximation set which has more non-dominated solutions that are close in proximity to the reference set.

Much like the GD measure, knowledge regarding the true Pareto-optimal front is required in order to form a reference set. The selection of solutions for the reference set will have an impact on the results obtained from the IGD, and therefore the reference set must be diverse. The calculation of the IGD can be computational expensive when working with large reference sets or a high number of objectives.

The IGD measure has been employed in the performance assessment of algorithms in much of the multi-objective optimisation and evolutionary computation literature (e.g. [131, 132, 133, 134, 135]).

2.10.4 Non-Parametric Testing

EAs are inherently stochastic and the initial conditions that ensure the reliability of parametric tests cannot be satisfied [136]. In order to find the significance in contrast amongst the results obtained by algorithms considered for comparison, a non-parametric test (encouraged by [137, 138]) for pairwise statistical comparison can be used. The Wilcoxon signed-ranks [139] non-parametric test (counter-part of the paired t-test) can be used with the statistical significance value ($\alpha = 0.05$), this is able to rank the difference in performance between two algorithms over each approximation set.

The use of non-parametric tests have been used to statistically contrast evolutionary algorithms in many experiments in the literature:

- [137] discusses the basics and gives a survey of a complete set of non-parametric test procedures and encourages the use of non-parametric tests when analysing results obtained by EAs, due to the fact that the initial conditions that guarantee the reliability of the parametric tests are not satisfied.
- In [140] the results of a hybrid EA used for data reduction and a competing algorithm are contrasted through non-parametric statistical tests in order to reinforce the resolved conclusion. Two non-parametric tests for pairwise statistical comparisons of classifiers are employed: the well-known Wilcoxon signed-ranks test [139] and the contrast estimation of medians [141].
- [142] emphasise that it is necessary to distinguish between pairwise tests and multiple comparison tests. This is achieved by demonstrating that when the pairwise Wilcoxon signed-ranks test is employed for multiple comparison the result will lead to overly optimistic solutions. Similar to the t -test, the Wilcoxon signed-ranks test is intended for the contrast in performance of two sets of data.
- Other research employing non-parametric tests in the comparison of EAs can be found in [136, 138, 143, 144].

2.10.5 Sample Size Sufficiency

Selecting a sufficient number of samples when comparing optimisers is critical. The sample size of 25, in order to reduce stochastic noise, is re-occurring in the evolutionary computation literature (e.g. [145, 146, 147, 148, 149]). The sufficiency of this sample size has been tested by producing a large number of hypervolume indicator value samples by executing WZ-MOEA/D-DRA (an optimiser described in Chapter 5) 200 times (the distribution of which has been illustrated in Figure 2.27) on the WFG6 synthetic test problem (described in Section 2.9.3).

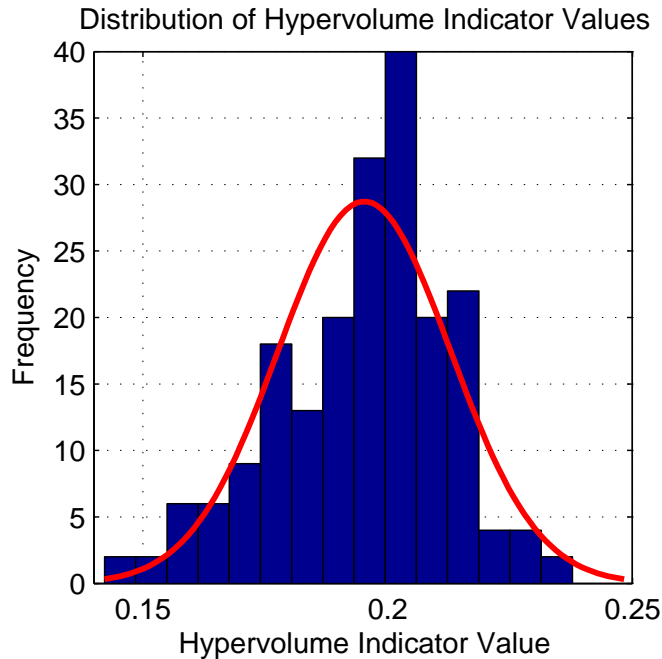


Figure 2.27: Histogram showing the distribution of the hypervolume indicator values from 200 executions of WZ-MOEA/D-DRA on the WFG6 synthetic test problem.

These 200 samples were then used to identify the relationship between the Standard Error of the Mean (SEM) and the sample size using:

$$SEM = \frac{SD}{\sqrt{N}} \quad (2.9)$$

This relationship has been illustrated in Figure 2.28 which shows the limited benefit of more than 25 independent executions of the algorithm on the synthetic test problem.

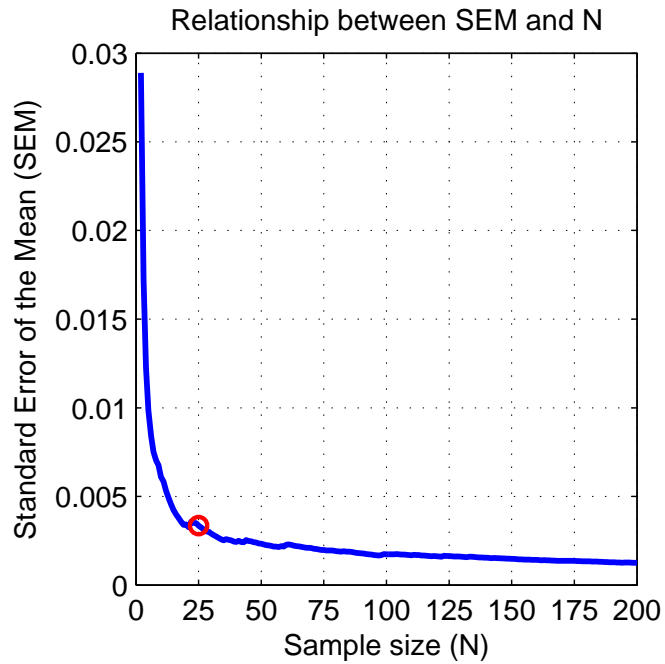


Figure 2.28: Relationship between Standard Error of the Mean (SEM) and the sample size of hypervolume indicator values from 200 executions of WZ-MOEA/D-DRA on the WFG6 synthetic test problem.

Chapter 3

The Covariance Matrix Adaptation Pareto Archived Evolution Strategy

Covariance Matrix Adaptation (CMA) (described in Section 2.2.3) has been selected from the literature as the desired variation operator for the design of an Evolutionary Algorithm (EA) intended for fast convergence within few function evaluations on real-world many-objective problems. The single-objective CMA driven optimiser, the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) (described in Section 2.2.3), has been shown to perform extremely well across a broad range of problems, including the single-objective optimiser performance comparisons in [58, 54].

The Multi-Objective Covariance Matrix Adaptation Evolution Strategy (MO-CMA-ES) (described in Section 2.7.3) is an existing Evolutionary Multi-Objective Optimisation (EMO) algorithm which utilises the CMA variation operator and has been shown to perform well in a number of algorithm variations in [36, 89, 88]. However, MO-CMA-ES relies on the contributing hypervolume (described in Section 2.5.2) indicator as a second-level sorting criterion and therefore suffers

from computational infeasibility on multi-objective problems which consist of more than three problem objectives. Real-world problems are often complex and require the optimisation of many objectives, therefore the CMA operator for variance needs to be incorporated into a new optimisation algorithm if it is to satisfy this requirement and be capable of being utilised for problems consisting of four or more problem objectives. In order to design an EMO algorithm which is driven by the CMA operator and capable of optimisation in the presence of many objectives, subjecting the entire non-dominated population to the contributing hypervolume indicator at each generation of the optimisation life-cycle must be avoided.

This chapter is divided into three sections. First, the Covariance Matrix Adaptation Pareto Archived Evolution Strategy (CMA-PAES) is introduced in Section 3.1 as a fast EMO algorithm which offers comparable performance to MO-CMA-ES, without reliance on the hypervolume indicator. Section 3.2 introduces the Multi-tier Covariance Matrix Adaptation Pareto Archived Evolution Strategy (m-CMA-PAES), an EMO algorithm which uses a multi-tier AGA with a grid-level hypervolume indicator, which outperforms MO-CMA-ES on problems consisting of two and three objectives. The chapter concludes with a summary of the developed algorithms and their intended use in Section 3.3.

3.1 CMA-PAES

The Pareto Archived Evolution Strategy (PAES) (described in Section 2.7.2) is an EMO algorithm which both contains a unique method of diversity preservation in the form of an Adaptive Grid Algorithm (AGA), and an algorithm which does not have a high computational cost due to its simplicity [150]. The simplicity of

PAES has inspired a base framework which can be used to intuitively incorporate the CMA operator, such that an AGA and bounded Pareto archiving scheme will be responsible for diversity preservation and selection for variation and survival, and CMA will be used as the variation operator.

An algorithm inspired by the PAES structure and the CMA scheme for variation has been designed under the name CMA-PAES. With the aim to be light in computational cost (without considering the computational cost of objective function evaluation), simple in structure in order to allow for easy extensibility as the algorithm matures and develops in further work, and the ability to produce approximation sets with performance initially similar to or better than MO-CMA-ES on a test suite for which comparison between CMA-PAES and MO-CMA-ES is feasible (three objectives or lower). The field of Evolutionary Computation (EC) is growing year by year, with many contributions including the introduction of new methods for selection, diversity preservation, variance, etc. The development of a simple and modular framework (CMA-PAES) would allow for easy incorporation of these new methods in any number of combinations, meaning that CMA-PAES can be extended to target specific problems or to incorporate state of the art techniques.

The algorithm execution order for CMA-PAES has been illustrated in Figure 3.1. CMA-PAES begins by initializing the algorithm variables and parameters, these include the number of grid divisions used in the AGA, the archive for storing Pareto-optimal solutions, the parent vector Y and the covariance matrix. An initial current solution is then generated at random, which is evaluated and then the first to be archived (without being subjected to the PAES archiving procedure). The generational loop then begins, the square root of the covariance

matrix is resolved using Cholsky decomposition (as recommended by [151]) which offers a less computationally demanding alternative to spectral decomposition. The λ candidate solutions are then generated using copies of the current solution and the CMA-ES procedure for mutation before being evaluated. The archive is then merged with the newly generated offspring and subjected to Pareto ranking, this assigns a rank of zero to all non-dominated solutions, and a rank reflecting the number of solutions that dominate the inferior solutions. The population is then purged of the inferior solutions so that only non-dominated solutions remain before being fed into the PAES archiving procedure. After the candidate solutions have been subjected to the archiving procedure and the grid has been adapted to the new solution coverage of objective space, the archive is scanned to identify the grid location with the smallest population, this is considered the lowest density grid population (*ldgp*). The solutions from the lowest density grid population are then spliced onto the end of the first $\mu - ldgp$ of the Pareto rank ordered population to be included in the adaptation of the covariance matrix, with the aim to improve the diversity of the next generation by encouraging movement into the least dense area of the grid. After the covariance matrix is updated, the generational loop continues onto its next iteration until the termination criteria is satisfied (maximum number of generations).

CMA-PAES has been benchmarked against the Nondominated Sorting Genetic Algorithm II (NSGA-II) and PAES in [15] in a performance comparison on the ZDT synthetic test suite, using two performance metrics to compare performance in terms of proximity (using the generational distance metric) and diversity (using the spread metric). CMA-PAES displayed superior performance (the significance of which was supported with randomisation testing) in return-

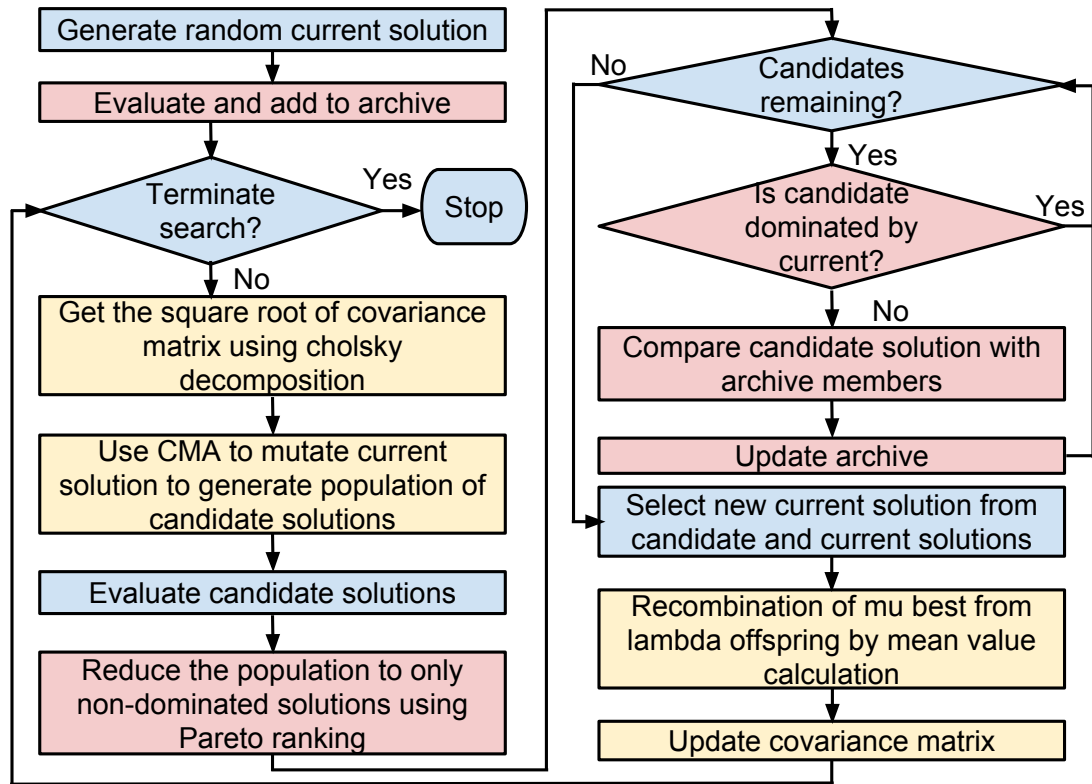


Figure 3.1: Execution life-cycle for the CMA-PAES algorithm.

ing an approximation set close to or on the true Pareto-optimal front as well as maintaining diversity amongst solutions in the set.

The ultimate aim of CMA-PAES development is to utilise the benefits of the CMA operator for variance in an EMO algorithm that is computationally feasible on many-objective problems, and comparable in performance to MO-CMA-ES.

3.1.1 Comparison Between CMA-PAES and MO-CMA-ES

In order to evaluate the performance of CMA-PAES, a pairwise comparison between CMA-PAES and MO-CMA-ES has been conducted. MO-CMA-ES is a popular and powerful EMO algorithm which uses the CMA operator for vari-

ance much like CMA-PAES, for this reason MO-CMA-ES has been selected for performance comparison to assess whether CMA-PAES is capable of comparable performance without relying on the contributing hypervolume indicator.

Both CMA-PAES and MO-CMA-ES have been configured with a function evaluation budget of 25,000 function evaluations per algorithm execution, this is to ensure fair comparison. The algorithm configurations are presented in Table 3.1, and the finer configurations for the CMA operator and population based MO-CMA-ES have been taken from [88], where the version of MO-CMA-ES used incorporates the improved step-size adaptation.

Table 3.1: Parameter configurations used for testing CMA-PAES and MO-CMA-ES.

Parameter	CMA-PAES	MO-CMA-ES
μ / Population	1	100
λ / Offspring	100	100
Generations	250	250
Archive Capacity	100	—
Grid Divisions	10	—
Mutation Rate	1	1

The ZDT test suite has been selected for the benchmarking and comparison of CMA-PAES and MO-CMA-ES, this test suite will pose basic difficulties that can be encountered during multi-objective search, and allows a feasible experiment to be conducted by only containing bi-objective problems. The test functions used for this experiment are ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6. The configurations used for these test functions are shown in Table 3.2.

Table 3.2: Parameter configurations used for the ZDT test suite.

Problem	Number of variables
ZDT1	30
ZDT2	30
ZDT3	30
ZDT4	10
ZDT5	10

The metric used for performance assessment is the hypervolume indicator described in Section 2.10.1. This metric determines the coverage of the objective space (assessing both proximity and diversity) of any given approximation set without the requirement of knowledge of the true Pareto-optimal front. This is a necessary feature of a performance metric in most real-world problems as the true Pareto-optimal front is often not known. The hypervolume indicator will be used at each generation in order to assess performance and compare both algorithms on not just the hypervolume indicator quality of the final approximation set but also the hypervolume indicator quality over time. CMA-PAES and MO-CMA-ES have been executed 25 times on each test function to reduce stochastic noise, this sample size has been selected because of the limited benefit of producing more than 25 samples (discussed in Section 2.10.5).

In this experiment, MO-CMA-ES is at an advantage as it uses the contributing hypervolume indicator for selection and diversity preservation, because of this it is expected that over time MO-CMA-ES will produce better quality approximation sets in regards to hypervolume indicator performance assessment.

3.1.2 Results

The results from the experiments in Section 3.1.1 have been produced and presented in a number of formats in order to allow for a better assessment of each algorithms performance.

Table 3.3 presents the worst, mean, and best hypervolume indicator results for the final approximation set of each algorithm. Overall, CMA-PAES outperformed MO-CMA-ES on three test functions (ZDT3, ZDT4, and ZDT6), and MO-CMA-ES outperformed CMA-PAES on two test functions (ZDT1 and ZDT2).

Table 3.3 also presents information regarding the p -value resolved by the Wilcoxon signed-ranks non-parametric test for the final approximation sets of the considered synthetic test problems, and a symbol indicating the observation of the null hypothesis. A '+' indicates that the null hypothesis was rejected, and CMA-PAES displayed statistically superior performance at the 95% significance level ($\alpha = 0.05$) on the considered synthetic test function. A '-' indicates that the null hypothesis was rejected, and CMA-PAES displayed statistically inferior performance. An '=' indicates that there was no statistically significant difference between both of the considered algorithms on the synthetic test problem. In all cases the null hypothesis was rejected and a statistical significance of greater than 95% was observed.

In regards to the hypervolume indicator results, CMA-PAES significantly outperforms MO-CMA-ES on three of the five considered synthetic test problems, and MO-CMA-ES significantly outperforms CMA-PAES on the remaining two synthetic test problems. The difference in the mean hypervolume indicator results show that CMA-PAES and MO-CMA-ES produce comparable approxima-

tion sets on the considered synthetic test problems, and CMA-PAES in particular shows far superior performance on ZDT4 and ZDT6 in regards to the magnitude of the difference in means of the hypervolume indicator results.

Table 3.3: Hypervolume indicator results from 25 executions of CMA-PAES and MO-CMA-ES on the ZDT test suite with 2 objectives, and results from pairwise comparison of the final approximation sets of both considered algorithms on each synthetic test function using the Wilcoxon signed-ranks non-parametric test.

2D	CMA-PAES			MO-CMA-ES			p -value
	Worst	Mean	Best	Worst	Mean	Best	
ZDT1	0.95713	0.95762	0.95772	0.95777	0.95783	0.95787	1.4e-09 –
ZDT2	0.91625	0.91648	0.91669	0.91674	0.91692	0.91697	1.4e-09 –
ZDT3	1.0096	1.0099	1.01	0.9976	1.0036	1.0051	1.4e-09 +
ZDT4	0.86214	0.89948	0.94049	0.82538	0.85985	0.90769	7.5e-07 +
ZDT6	0.67793	0.67869	0.679	0.65556	0.65558	0.65559	1.4e-09 +

The box plots in Figure 3.2 show that on ZDT1 the median hypervolume indicator value is greater for MO-CMA-ES than that of CMA-PAES, the total range and interquartile range for MO-CMA-ES is smaller, showing a more robust set of results from MO-CMA-ES as well as fewer outliers. ZDT2 appears to be the test function for which both algorithms produced the most outliers, suggesting this test function causes difficulty in achieving consistently robust performance for both algorithms. ZDT3 shows that regardless of MO-CMA-ES being bottom skewed, the interquartile range for CMA-PAES is far greater. On ZDT4, it can be observed that although the hypervolume indicator performance for CMA-PAES is top-skewed and the dispersion is greater, the interquartile range achieves a better hypervolume indicator quality than MO-CMA-ES overall. ZDT6 shows that all results from CMA-PAES clearly outperform MO-CMA-ES.

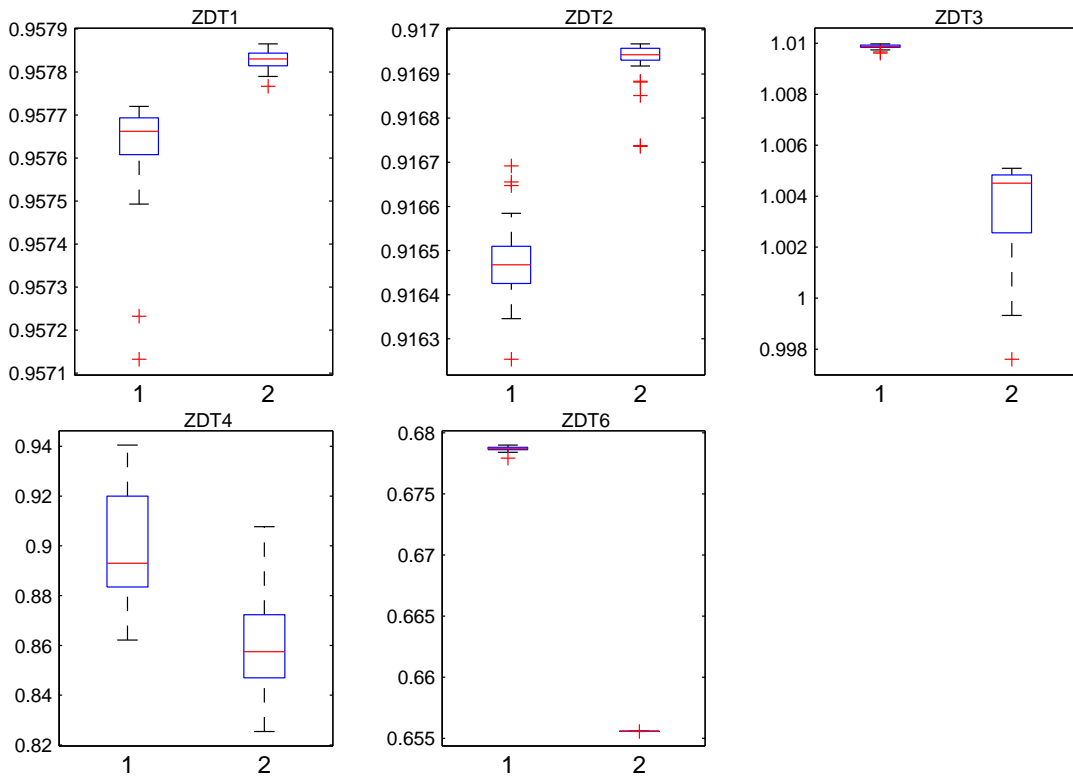


Figure 3.2: Box plots of hypervolume indicator results for 2-objective ZDT problems (1: CMA-PAES; 2: MO-CMA-ES) 25,000 function evaluations, 25 runs.

Figure 3.3 shows the hypervolume indicator performance of each algorithm on all test functions at each generation of the optimisation process. On ZDT1, although MO-CMA-ES converges to a slightly better mean hypervolume indicator value, CMA-PAES converges much faster to a similar hypervolume indicator quality within 75 generations, again on ZDT2 it can be observed that within just 100 generations CMA-PAES has achieved a hypervolume indicator quality similar to that of MO-CMA-ES on its final generation. On ZDT3, CMA-PAES converges to a hypervolume indicator quality that outperforms MO-CMA-ES at just over 50 generations which remains similar throughout the remainder of the search.

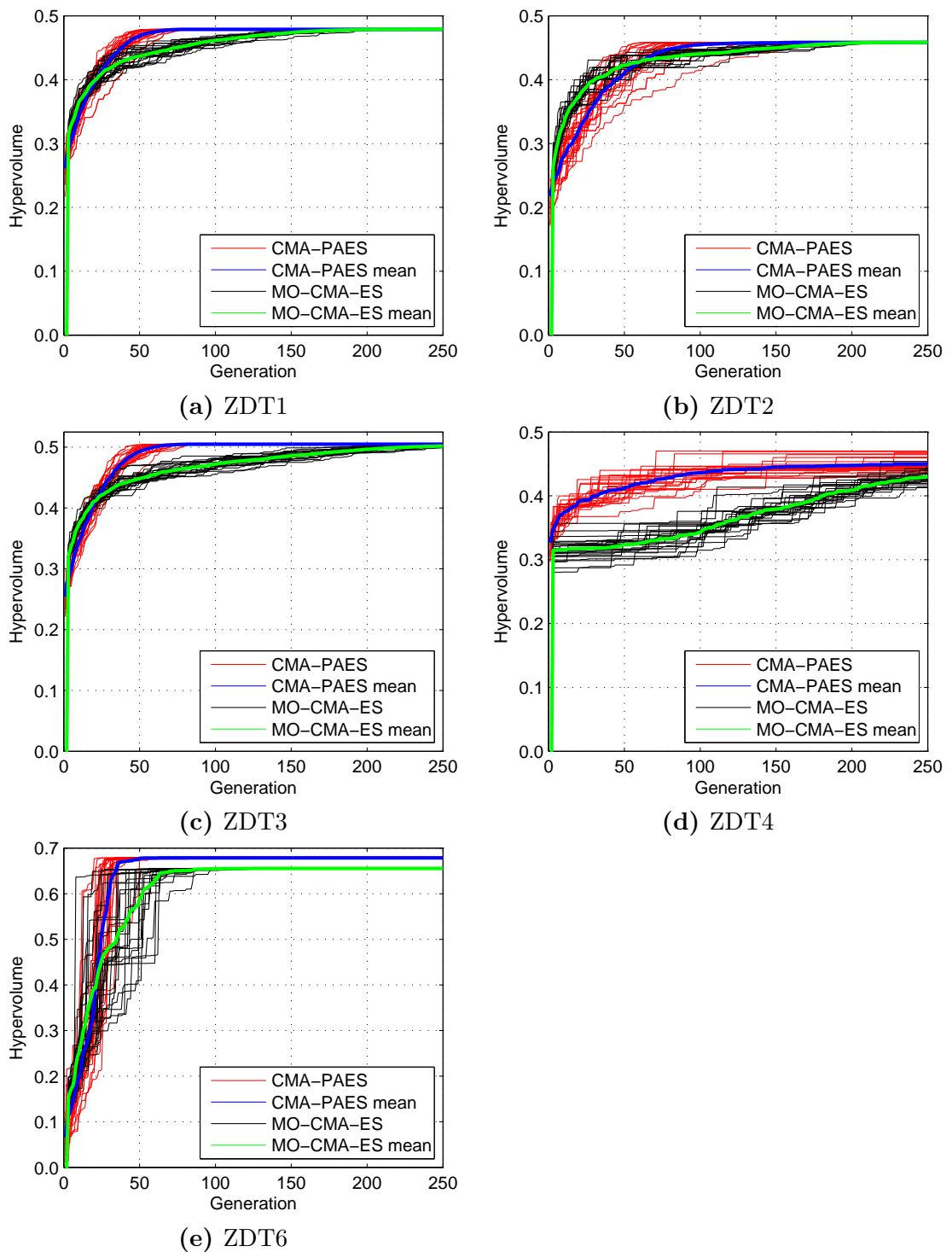


Figure 3.3: Hypervolume indicator values at each generation for CMA-PAES and MO-CMA-ES on the considered ZDT test problems.

3.1.3 Conclusion

The benchmarking and performance analysis of the algorithm returned promising results that suggest on some problems CMA-PAES is faster at converging to an approximation set close to or on the true Pareto-optimal front, as well as returning a diverse set of solutions in regards to points in the objective space.

These observations held in the comparison with MO-CMA-ES on equal function evaluations, however, in this section, no serious attempt was made to find the optimal parameter settings for CMA-PAES. As previously mentioned, CMA-PAES and other CMA driven EMO algorithms fail to perform adequately on ZDT4, further work is required to identify a method for preventing CMA-PAES to be deceived into prematurely converging to locally Pareto-optimal fronts.

Overall CMA-PAES has been designed and developed as an algorithm which utilises CMA as a variation operator, without the need for the hypervolume indicator for selection and diversity preservation, but instead using a computationally lightweight AGA scheme which outperforms MO-CMA-ES when both algorithms are benchmarked on the ZDT test suite.

3.2 m-CMA-PAES

Elitism in EMO algorithms has been shown to improve the rate of convergence by ensuring some or all of the fittest individuals in a population at generation g are inserted into generation $g + 1$. Using this method, it is possible to prevent the loss of the fittest individuals which are considered to have some of the most valuable chromosomes in the population. Many state of the art EMO algorithms use elitism at the core of their population management schemes, for example,

NSGA-II, PAES, and MO-CMA-ES. However, in many multi-objective optimisation problems, solutions exist which may not be considered elite due to their objective value in regards to the population, but may contain useful genetic information. This genetic information can be utilised later in the search to move into unexplored areas of the objective-space, but due to elitism and non-dominated sorting schemes it may be abandoned in the early stages of the search.

The consequences of elitism and non-dominated sorting can be seen in Figure 3.4, where the MO-CMA-ES has produced an approximation set for the CEC09 UF1 [92] test function with a budget of 300,000 function evaluations (in compliance with the CEC09 competition rules).

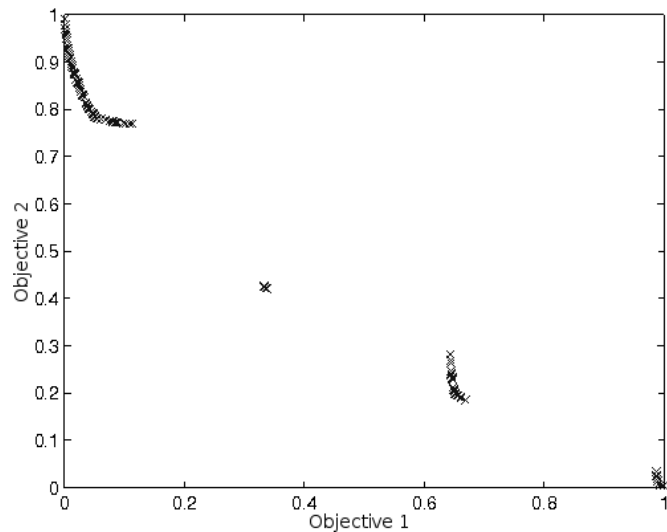


Figure 3.4: An approximation set found using MO-CMA-ES after 300,000 function evaluations on CEC09 UF1.

By observing this two-objective plot of the approximation set, it can be seen that the MO-CMA-ES has converged to an approximation set which is missing three distinct areas containing solutions in comparison to the true Pareto-optimal front plotted in Figure 3.5. The genetic information which would have potentially

found these missing areas was discarded by the MO-CMA-ES during the search process due to the use of elitism and non-dominated sorting. This is a difficulty that occurs in the CEC09 UF1 test problem because of its complicated Pareto-optimal set, making it easier to converge to some areas of the Pareto-optimal front early in the search. In these cases, the MO-CMA-ES will focus selection on these more dominant solutions and converge further into that area of the Pareto-optimal-set, and discard individuals which may have been only a few generations away from producing non-dominated solutions in unexplored areas of the objective-space.

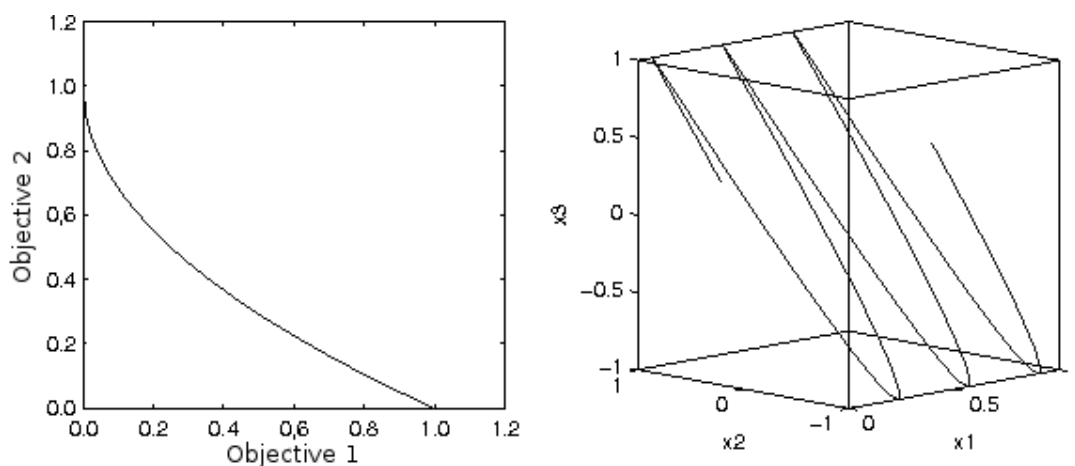


Figure 3.5: The true Pareto-optimal front (left) and Pareto-optimal set (right) for CEC09 UF1.

Figure 3.6 illustrates an example of elitist and non-dominated selection discarding an individual that may contain valuable genetic information, which could have been exploited to produce a better quality approximation set. In this example a Pareto AGA selection scheme has been used to select parent individuals for the next generation. Because of the scheme's elitist nature, the individual between 0.6 and 0.7 on the x-axis has not been selected for reproduction, and

therefore the scheme has discarded genetic information which may have ultimately produced solutions towards the missing area of the approximation set. This behaviour over many generations can lead to convergence to incomplete approximation sets.

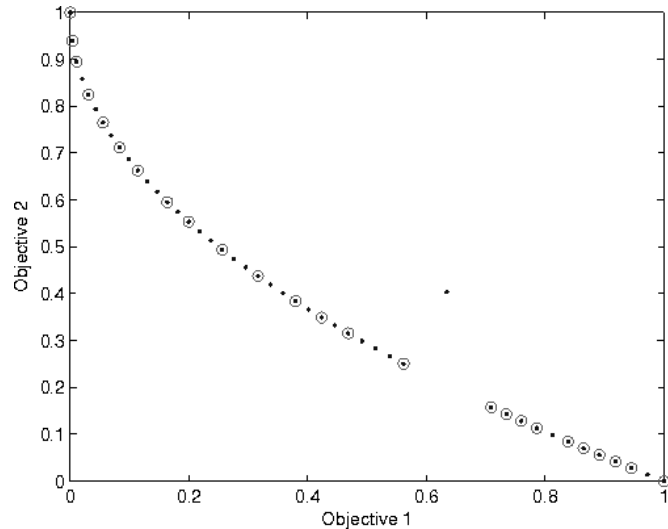


Figure 3.6: An example of elitist and non-dominated selection, circled points indicate a selected individual.

The goal of this study is to counter this negative effect inherited from the elitism in CMA-PAES, and to evaluate its performance on several benchmarking test suites. To achieve this, a new multi-tier adaptive grid selection scheme is developed and combined with the existing CMA-PAES algorithm, in a new algorithm named the Multi-tier Covariance Matrix Adaptation Pareto Archived Evolution Strategy (*m-CMA-PAES*).

CMA-PAES inherited the issues caused by elitism from its contributing algorithms, and has been revised to improve its robustness by allocating a computational budget for dominated and non-elite population individuals at each

generation. The algorithm execution life-cycle for m-CMA-PAES has been illustrated in Figure 3.7.

m-CMA-PAES begins by initialising the initial population, the generational loop then begins by sampling an offspring population. A single offspring is created by copying every member of the parent population into a new population, and then mutating the newly copied offspring's problem variable using its inherited covariance matrix. The new offspring population and old parent population are then merged into a single solution pool to be used in the selection procedure described in Section 3.2.1, where as in MO-CMA-ES the population would be divided into sub-populations by their rank of non-dominance, and then sorted at a secondary level by their contributing hypervolume indicator value. Once the solutions in the selection pool have been assigned their fitness levels, μ individuals are selected to be used as the parent population for the next generation. Before moving onto the next generation, the success probabilities, step-sizes, evolution paths and covariance matrices of the successful solutions are updated.

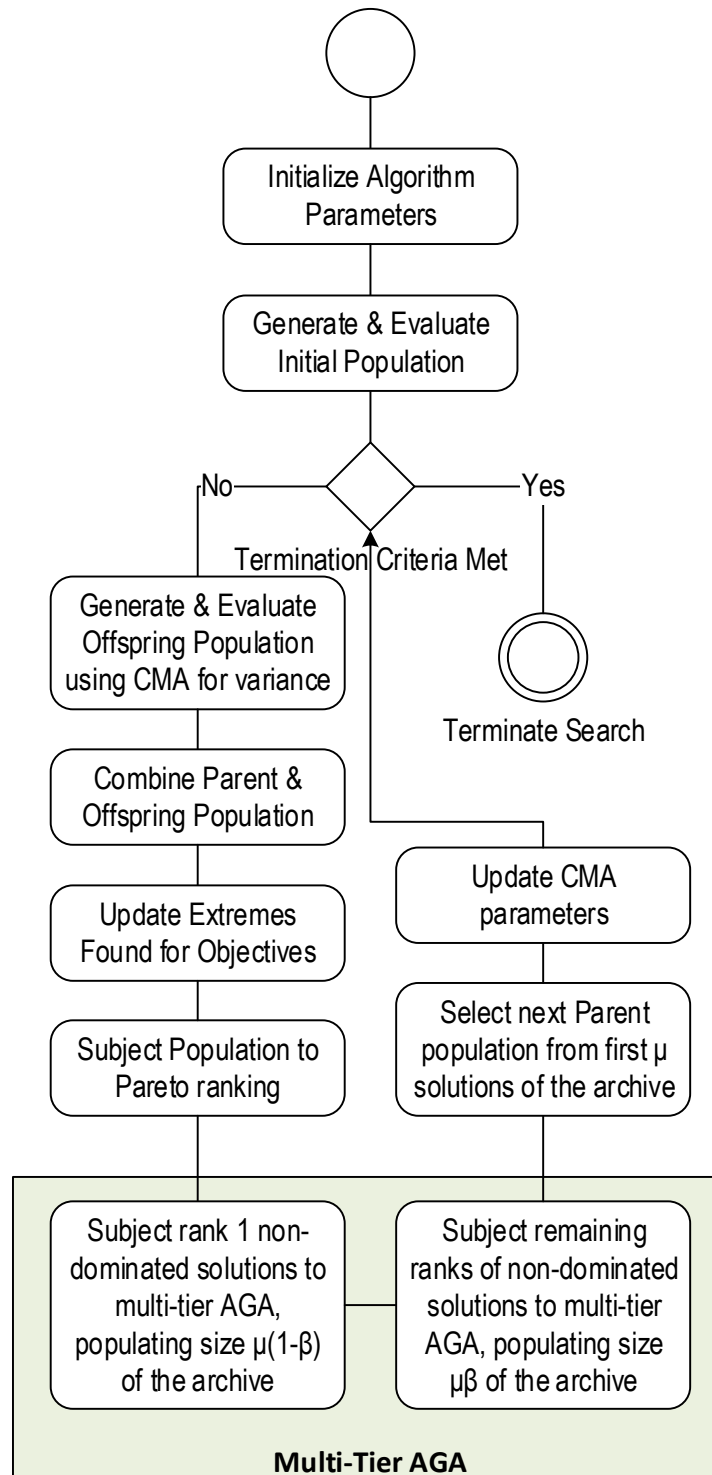


Figure 3.7: Execution life-cycle for the *m*-CMA-PAES algorithm.

3.2.1 New Multi-Tier AGA

The new multi-tier AGA aims to prevent the population prematurely converging due to following only the dominant (i.e. elite) solutions which are discovered early in the optimisation process, as this ultimately results in genetic drift and consequently an approximation set with solutions clustered around these points. This is achieved by dividing the function evaluation budget and investing a percentage of the budget in to non-elite solutions (which can potentially contain genetic information that would contribute to finding undiscovered areas of the objective space later in the search).

The algorithm pseudo-code is listed in Algorithm 2, which is executed from the shaded in “Multi-Tier AGA” stage in Figure 3.7. First, the candidate population is divided into sub-populations based on their non-dominated rank using NSGA-II’s fast non-dominated sort. If the size of any sub-population exceeds μ , then the standard AGA scheme is applied to it with a maximum archive capacity of μ , resulting in a number of rank-ordered archives each with a maximum capacity of μ . Then, a single population of size μ plus the budget for non-elite individuals β is produced, for example if β is set as 10% for a μ population of 100, then a population of size 100×1.10 is to be produced. Next the multi-tier archives containing the first $\mu \times \beta$ solutions are merged with no size restriction (meaning the merged archive size can be greater than $\mu \times \beta$). This merged archive is then subjected to a non-elite AGA (ensuring non-elite solutions are not instantly discarded) with an archive capacity of μ , producing a population of individuals to be selected as parents for the next generation.

Algorithm 2 Pseudo-code of Multi-Tier Adaptive Grid Algorithm

```
1: nonDominatedFronts = nonDominatedSort(population)
2: spaceRemaining =  $\mu \times \beta$ 
3: for all nonDominatedFront in nonDominatedFronts do
4:   if spaceRemaining > 0 then
5:     tierArchive = adaptiveGridSelection(nonDominatedFront,  $\mu$ )
6:     archive = archive + tierArchive
7:     spaceRemaining = spaceRemaining - size(archive)
8:   end if
9:   parentPopulation = adaptiveGridSelection(archive,  $\mu$ )
10: end for
```

The configuration of β is important - if it is too high (for example if it is greater than half of μ), then the majority of the budget is spent on dominated solutions and the search does not progress in a positive direction, and may instead move away from the Pareto-optimal front. However if β is too small, the benefits of investing in non-elite solutions are not achieved. The result of this new grid selection scheme has been illustrated in Figure 3.8, where the solution which may potentially contain valuable genetic information is selected, in contrast to it being discarded in Figure 3.6.

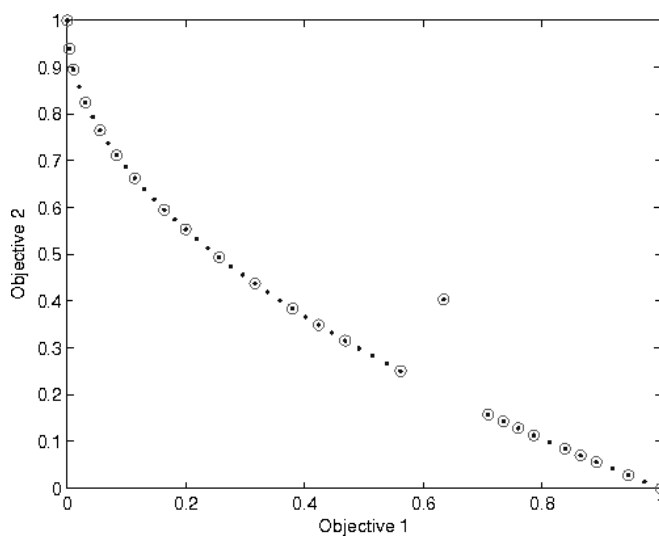


Figure 3.8: An example of the multi-tiered grid selection, circled points indicate a selected individual.

3.2.2 Comparison Between m-CMA-PAES and MO-CMA-ES

In order to evaluate the performance of m-CMA-PAES on multi-objective test problems consisting of up to three problem objectives, a pairwise comparison between m-CMA-PAES and MO-CMA-ES on selected benchmarking problems has been conducted. MO-CMA-ES is a popular and powerful algorithm which uses the CMA operator for variance much like m-CMA-PAES. For this reason MO-CMA-ES is selected for performance comparison to assess whether m-CMA-PAES is capable of comparable performance without relying on the contributing hypervolume indicator at population level.

Both m-CMA-PAES and MO-CMA-ES have been configured with a budget of 300,000 function evaluations per algorithm execution, and were executed 30 times per test function as per the CEC09 competition guidelines. The algorithm configurations are presented in Table 3.1, and the finer configurations for the

CMA operator and MO-CMA-ES have been taken from [88], where the version of MO-CMA-ES used incorporates the improved step-size adaptation.

Table 3.4: Algorithm configurations used when benchmarking MO-CMA-ES and m-MA-PAES.

Parameter	MO-CMA-ES		m-CMA-PAES	
μ	2D(100)	3D(300)	2D(100)	3D(300)
λ	2D(100)	3D(300)	2D(100)	3D(300)
Archive Capacity	—		2D(100)	3D(300)
Multi-tier Budget	—		10%	
Divisions	—		10	

The ZDT, DTLZ, and CEC09 test suites have been selected for the benchmarking and comparison of m-CMA-PAES and MO-CMA-ES. The ZDT and DTLZ test suites will face both algorithms with difficulties likely to be encountered in most real-world multi-objective optimisation problems, in both two-dimensional and three-dimensional objective spaces (allowing for feasible comparison with MO-CMA-ES). The CEC09 competition test suite will face the algorithms with difficulties encountered when optimising in the presence of complex Pareto-optimal sets and Pareto shapes. The test functions used for this experiment are ZDT1 through to ZDT6 (excluding ZDT5), DTLZ1 through to DTLZ7, and CEC09 UF1 through to CEC09 UF10. The configurations used for these test problems are shown in Table 3.5.

Table 3.5: Parameter configurations used for the ZDT, DTLZ and CEC09 test suites.

(a) Three-objective test functions.

Problem	Number of variables
ZDT1	30
ZDT2	30
ZDT3	30
ZDT4	10
ZDT5	10
UF1	30
UF2	30
UF3	30
UF4	30
UF5	30
UF6	30
UF7	30

(b) Three-objective test functions.

Problem	Number of variables
UF8	30
UF9	30
UF10	30
DTLZ1	7
DTLZ2	12
DTLZ3	12
DTLZ4	12
DTLZ5	12
DTLZ6	12
DTLZ7	22

The metric used for performance assessment is the Inverted Generational Distance (IGD) indicator described in Section 2.10.3. The IGD indicator will be used at each generation in order to assess performance, and compare both algorithms on not just the IGD quality of the final approximation set, but also the IGD quality over time. Both m-CMA-PAES and MO-CMA-ES have been executed 30 times on each test function to reduce stochastic noise, this sample size has been selected in order to comply with the CEC09 competition rules described in [92], and is seen as sufficient because of the limited benefit of producing more than 25 samples (discussed in Section 2.10.5).

3.2.3 Results

The results from the experiments in Section 3.2.2 have been produced and presented in a number of formats in order to allow for a better assessment of each algorithm's performance.

The worst, mean, and best IGD indicator results for the final approximation set of each algorithm are presented in Table 3.6 for the two-objective test functions, and in Table 3.7 for the three-objective test functions. Tables 3.6 and 3.7 also present information regarding the p -value resolved by the Wilcoxon signed-ranks non-parametric test for the final approximation sets of the considered synthetic test problems, and a symbol indicating the observation of the null hypothesis. A '+' symbol indicates that the null hypothesis was rejected, and m-CMA-PAES displayed statistically superior performance at the 95% significance level ($\alpha = 0.05$) on the considered synthetic test function. A '-' symbol indicates that the null hypothesis was rejected, and m-CMA-PAES displayed statistically inferior performance. An '=' symbol indicates that there was no statistically significant difference between both of the considered algorithms on the synthetic test problem. Overall, m-CMA-PAES outperformed MO-CMA-ES on all but 3 (ZDT3, UF6 and UF9) of the 22 test functions, producing better performing worst, mean, and best approximation sets.

Table 3.6: IGD results from 30 executions of m-CMA-PAES and MO-CMA-ES on the ZDT and CEC09 test suites with two problem objectives.

m-CMA-PAES				MO-CMA-ES			p -value	
2D	Worst	Mean	Best	Worst	Mean	Best		
ZDT1	0.00628	0.00657	0.00686	0.00813	0.00936	0.01031	1.4e-09	+
ZDT2	0.00592	0.00614	0.00639	0.00989	0.01172	0.01511	1.4e-09	+
ZDT3	0.00574	0.00609	0.00676	0.00552	0.00594	0.00643	0.0625	=
ZDT4	1.80044	6.17983	11.44563	2.85512	8.35397	14.56593	0.0232	+
ZDT6	0.01132	0.01279	0.01406	0.04788	0.08938	0.20901	1.4e-09	+
UF1	0.03762	0.05824	0.06579	0.05044	0.07228	0.12375	1.1e-06	+
UF2	0.01359	0.02006	0.02687	0.02117	0.03496	0.05235	5.5e-08	+
UF3	0.04869	0.07992	0.12647	0.06044	0.08129	0.10133	0.7269	=
UF4	0.05925	0.06431	0.06942	0.07661	0.08261	0.09722	1.4e-09	+
UF5	0.49880	0.72982	1.04816	0.87997	1.04873	1.26644	8.3e-09	+
UF6	0.08817	0.12736	0.22802	0.09314	0.11268	0.22469	0.010432	-
UF7	0.01791	0.02431	0.03226	0.03306	0.06434	0.12773	1.4e-09	+

Table 3.7: IGD results from 30 executions of m-CMA-PAES and MO-CMA-ES on the DTLZ and CEC09 test suites with three problem objectives.

m-CMA-PAES				MO-CMA-ES			p -value	
2D	Worst	Mean	Best	Worst	Mean	Best		
UF8	0.13308	0.18188	0.23023	0.16091	0.23432	0.24924	3.6e-08	+
UF9	0.07381	0.07877	0.08795	0.06755	0.07440	0.07911	5.4e-05	-
UF10	0.64046	0.97907	1.34102	1.33073	1.90805	2.89107	1.6e-09	+
DTLZ1	0.60928	3.11971	5.72913	2.11988	10.1829	20.9531	1.2e-06	+
DTLZ2	0.03919	0.04005	0.04077	0.04207	0.04491	0.04939	1.4e-09	+
DTLZ3	22.4023	50.7571	102.51	171.175	188.531	229.147	1.4e-09	+
DTLZ4	0.02459	0.03090	0.04093	0.03181	0.04411	0.07016	5.5e-08	+
DTLZ5	0.00152	0.00174	0.00201	0.00190	0.00213	0.00259	8.3e-09	+
DTLZ6	0.11059	0.32162	0.65582	0.19705	0.42455	0.71631	0.01701	+
DTLZ7	0.05268	0.05783	0.06449	0.05824	0.06653	0.07449	2.9e-08	+

The mean of the IGD metric at each generation has been plotted and presented in Figure 3.9 for the two-objective test functions, and Figure 3.10 for the three-

objective test functions. These plots illustrate the rate of IGD convergence from the initial population to the final population.

m-CMA-PAES significantly outperforms the *MO-CMA-ES* on most of the test-functions used in this comparison. However, as a consequence of investing a percentage of the maximum number of function evaluations in non-elite solutions, it can be observed in Figures 3.9 and 3.10 that the convergence of the algorithm is slower in most cases (more so in the two-objective test functions). This suggests that in experiments where the number of function evaluations are not constrained to a low number, the *m-CMA-PAES* will outperform *MO-CMA-ES*.

It can be observed in Figures 3.9 and 3.10 that the mean IGD for *MO-CMA-ES* oscillates or rises on some test functions over time. This issue is most visible on UF4 where the mean IGD for *MO-CMA-ES* can be seen to oscillate over time, and in DTLZ3 where the mean IGD for *MO-CMA-ES* can be seen to improve in performance until 200 generations and then worsen gradually until termination. This issue is due to *MO-CMA-ES* being dependent on the hypervolume indicator entirely for diversity preservation, paired with its elitism scheme gradually reducing the IGD quality of an approximation set once a difficult area of the search space is encountered.

The results presented in Table 3.6 and Table 3.7, as well as the box plots presented in Figure 3.11 and Figure 3.12 show that on 18 of the 22 considered test functions, *m-CMA-PAES* significantly outperformed *MO-CMA-ES* in regards to the achieved mean and median IGD. The box plots show that the interquartile ranges for *m-CMA-PAES* results are either lower than the medians or interquartile ranges for *MO-CMA-ES* results. Across all test functions *m-CMA-PAES*

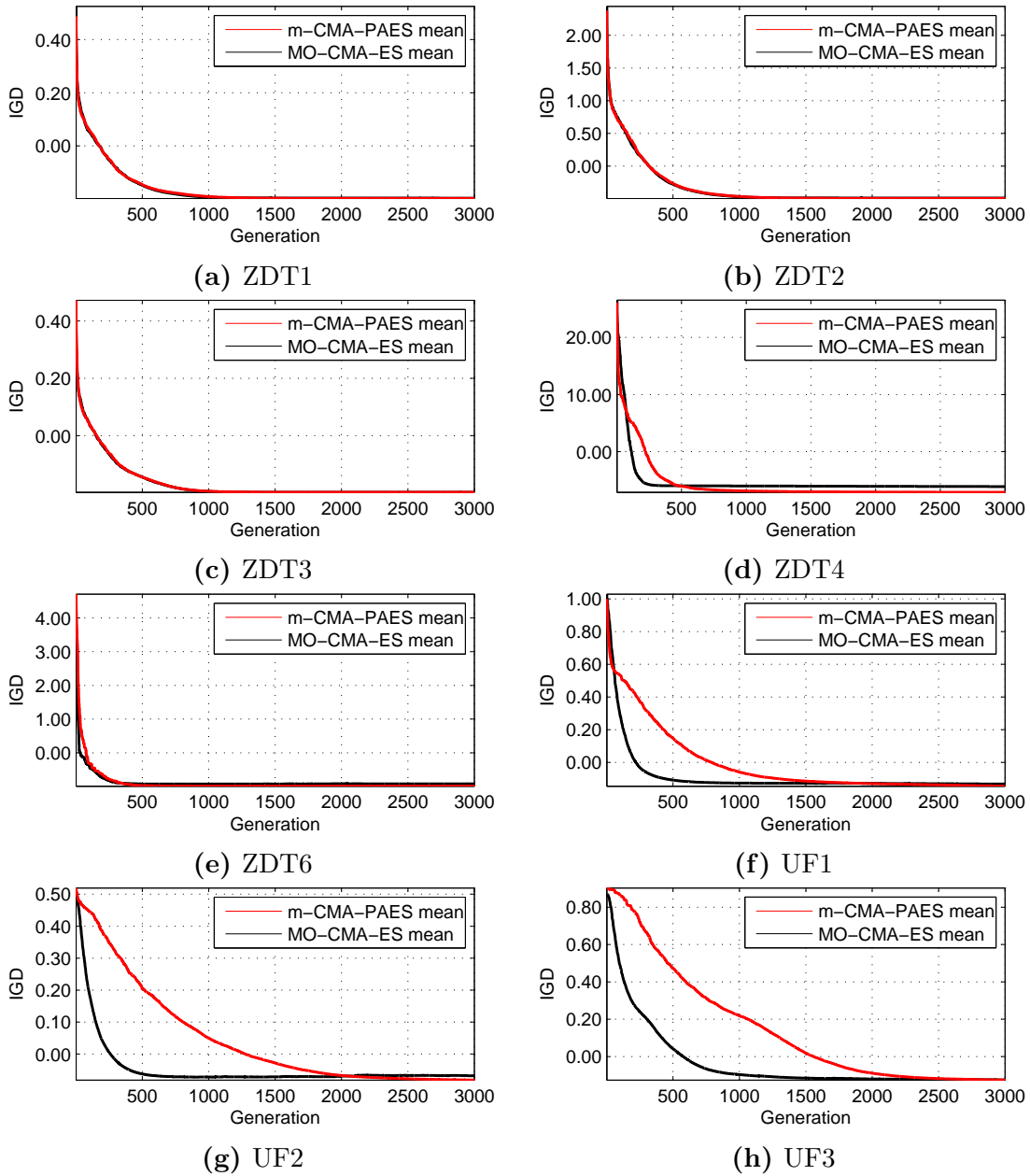


Figure 3.9: IGD results at each generation visualising performance of m-CMA-PAES and MO-CMA-ES over 300,000 function evaluations on two-objective test problems, 30 runs.

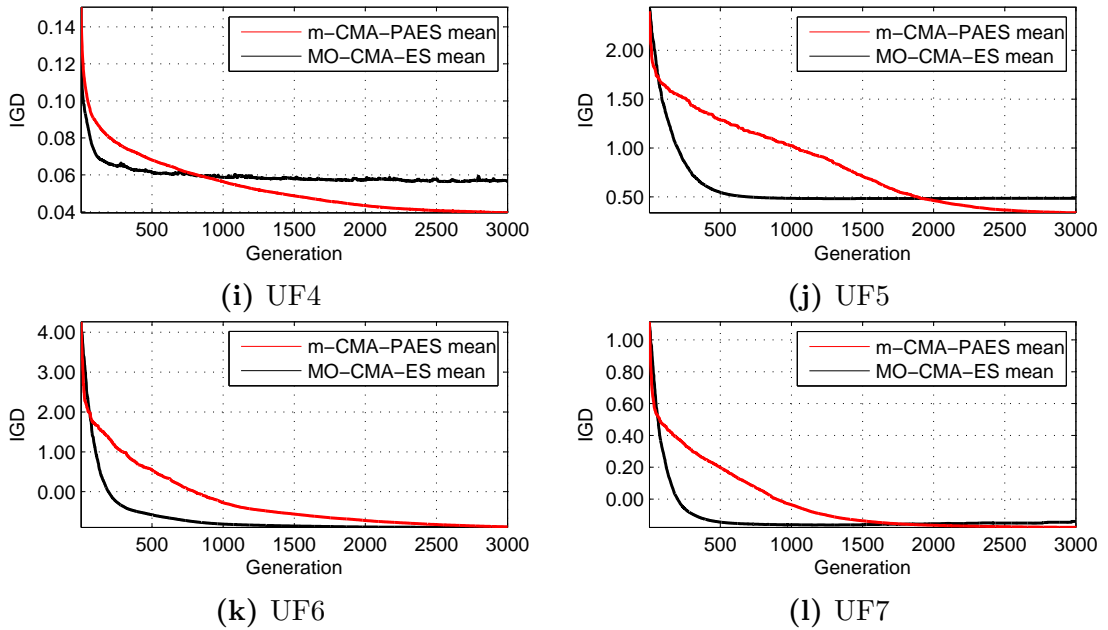


Figure 3.9: IGD results at each generation visualising performance of m-CMA-PAES and MO-CMA-ES over 300,000 function evaluations on two-objective test problems, 30 runs.

produces fewer outliers indicating a more reliable and robust algorithm in comparison to MO-CMA-ES on the considered test functions.

On the UF3 test function, it can be observed in Figure 3.11 that although the MO-CMA-ES median IGD outperforms m-CMA-PAES, m-CMA-PAES achieved a better interquartile range, and a far better total range, achieving the best approximation set for that test function, a similar result to the performance on UF6 where CMA-PAES also achieves the best approximation set but is outperformed by MO-CMA-ES on the median values of the IGD results.

The MO-CMA-ES significantly outperforms the m-CMA-PAES on the UF9, this function (as well as ZDT3 and UF6) consists of disjoint true Pareto-optimal fronts as shown in Figure 3.13. With the comparison in performance on these problems, it can be seen that the m-CMA-PAES has performance issues on some

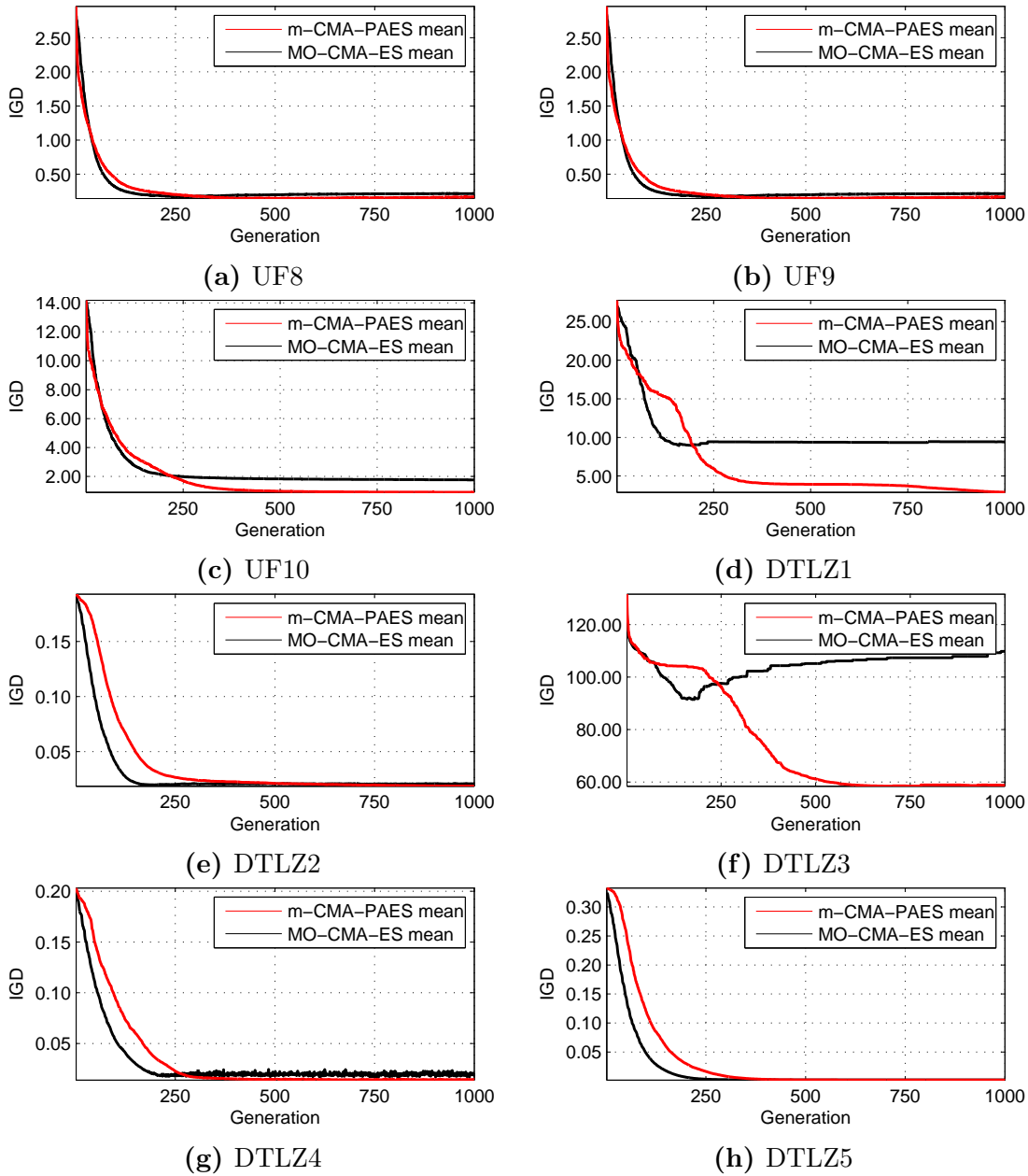


Figure 3.10: IGD results at each generation visualising performance of m-CMA-PAES and MO-CMA-ES over 300,000 function evaluations on three objective test problems, 30 runs.

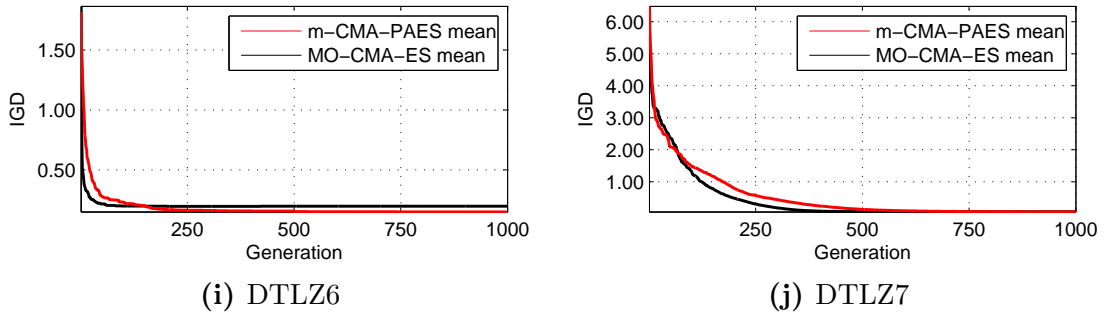


Figure 3.10: IGD results at each generation visualising performance of *m*-CMA-PAES and MO-CMA-ES over 300,000 function evaluations on three objective test problems, 30 runs.

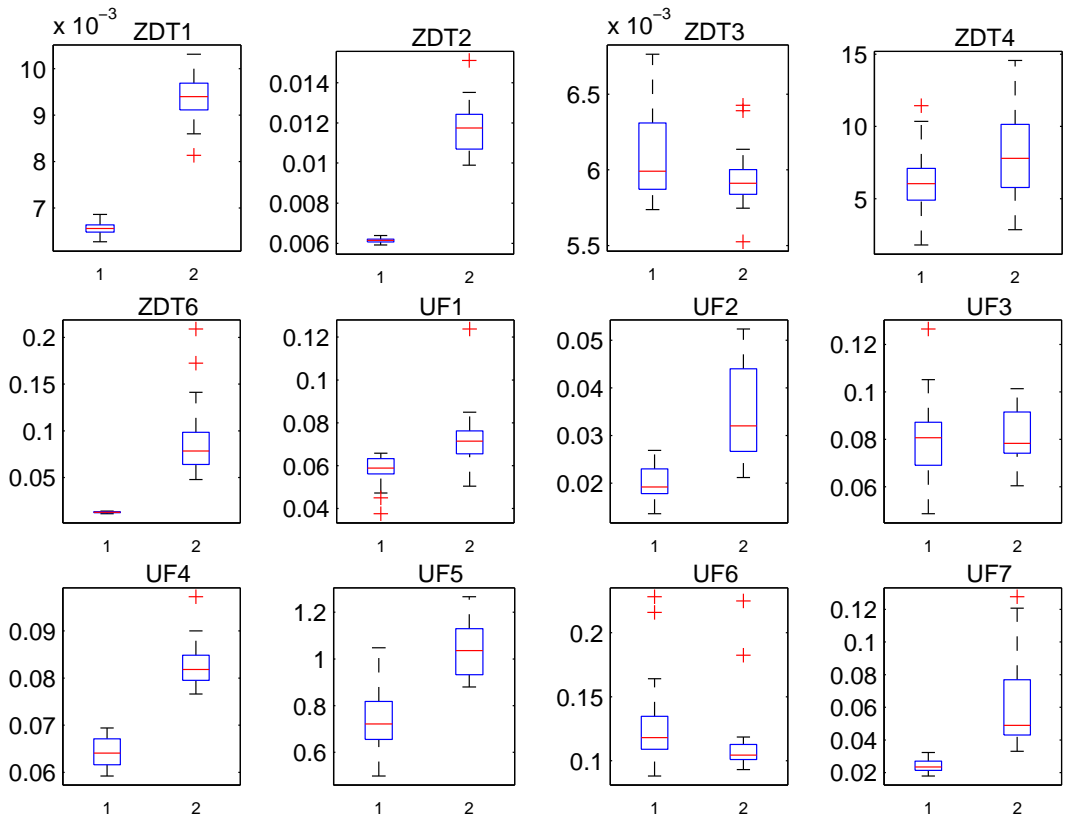


Figure 3.11: Box plots of IGD indicator results for two-objective test problems (1: *m*-CMA-PAES; 2: MO-CMA-ES) 300,000 function evaluations, 30 runs.

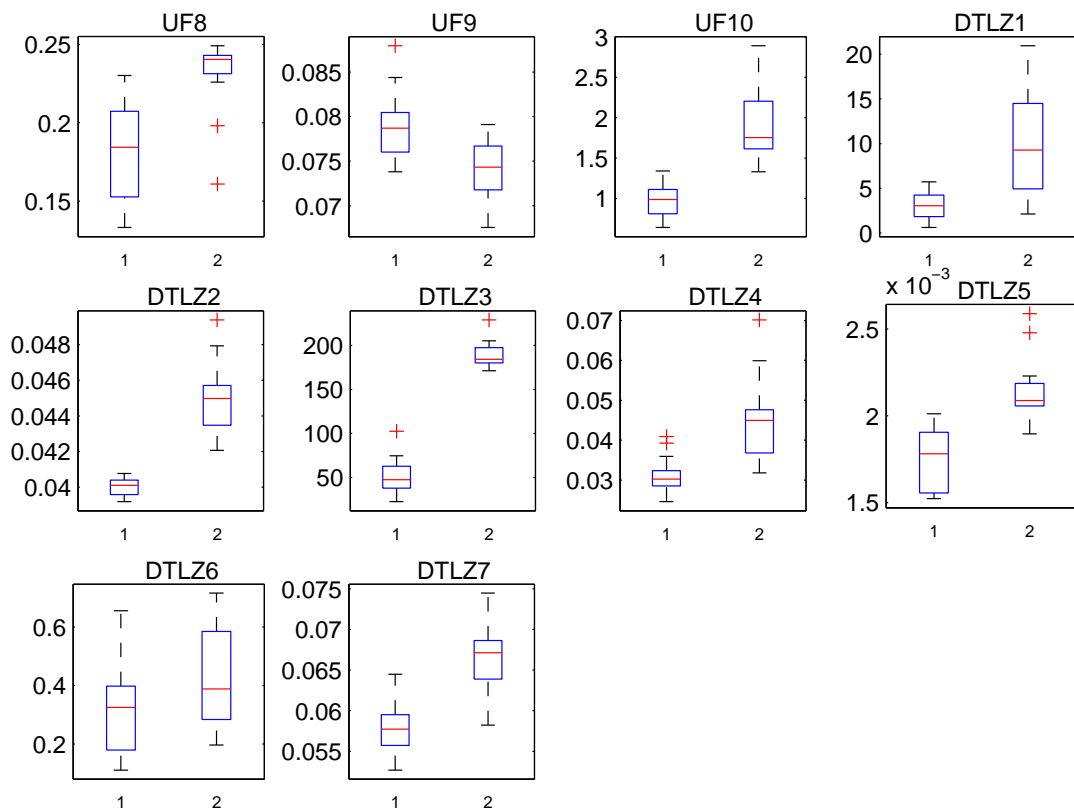


Figure 3.12: Box plots of IGD indicator results for three-objective test problems (1: m-CMA-PAES; 2: MO-CMA-ES) 300,000 function evaluations, 30 runs.

problems consisting of multiple parts in their Pareto-optimal fronts.

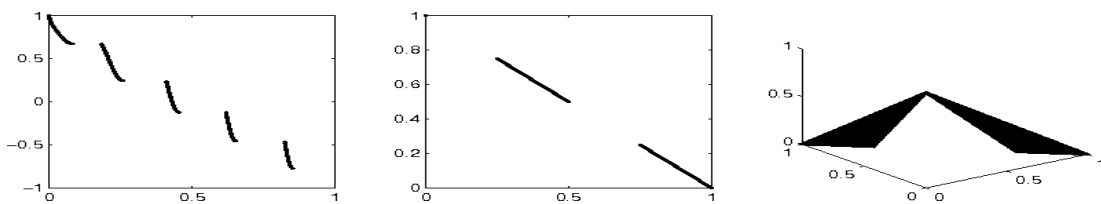


Figure 3.13: True Pareto-optimal fronts plotted for problems ZDT3 (left), UF6 (middle) and UF9 (right).

3.2.4 Conclusion

In this section, a multi-tier AGA scheme has been introduced and incorporated into the CMA-PAES algorithm to create *m*-CMA-PAES. *m*-CMA-PAES improves the quality of the produced final approximation set by investing a percentage of the allowed function evaluation budget in non-elite but potentially successful solutions. Experiments and statistical analysis presented in this study show that with CEC09 competition compliant benchmarking configurations, *m*-CMA-PAES significantly outperforms MO-CMA-ES on all but 4 of the 22 considered synthetic test problems, and out of these 4, MO-CMA-ES only performs significantly better on 2 test functions.

When observing the IGD values at each generation, it can be seen that in some cases the IGD of the final population is higher than some of the generations before it, this is due to the non-elite solutions invested in at each generation being a factor right to the end of the algorithm. This suggests that in further work the algorithm may benefit from either an offline archive which the algorithm selects from at the end of the optimisation process or a final approximation set selection scheme which uses the last two generations of the optimisation process, including non-dominated solutions only.

3.3 Conclusion

In this chapter, two EMO algorithms which utilise the CMA operator for variance have been designed, developed, and benchmarked to assess their performance. Both of these algorithms (CMA-PAES and m-CMA-PAES) are either completely independent of the contributing hypervolume indicator or do not rely on complete contributing hypervolume indicator calculation for the entire population. A summary and comparison of features between CMA-PAES and m-CMA-PAES has been presented in Table 3.8.

Table 3.8: Feature comparison between CMA-PAES and m-CMA-PAES.

	CMA-PAES	m-CMA-PAES
Population Structure	$1 + \lambda$	$\mu + \lambda$
Diversity Preservation	AGA	Multi-Tier AGA
Hypervolume Reliance	None	Grid-level
Suitable # of Objectives	2 Objectives	Up to 3
Suitable Pareto Shape	Simple	Complex
Targeted at	Fast Convergence	Complex Problems

CMA-PAES has been introduced as an EMO algorithm with promising performance on simple test functions containing a low number of objectives, with the ability to converge to approximation sets scoring comparably to MO-CMA-ES in regards to the hypervolume indicator results, without the requirement for using any form of hypervolume indicator calculation during the optimisation process.

m-CMA-PAES has been introduced as an EMO algorithm with promising performance on test functions consisting of complex Pareto-optimal sets with a large function evaluation budget, with the ability to converge to an approximation

set that outperforms MO-CMA-ES in most cases, using the contributing hypervolume indicator at grid-level only, which avoids the computational infeasibility MO-CMA-ES faces.

Chapter 4

The Covariance Matrix Adaptation Pareto Archived Evolution Strategy II

The Covariance Matrix Adaptation Pareto Archived Evolution Strategy (CMA-PAES) and the Multi-tier Covariance Matrix Adaptation Pareto Archived Evolution Strategy (m-CMA-PAES) have shown promising results in Sections 3.1 and 3.2 by either offering similar or better performance than the Multi-Objective Covariance Matrix Adaptation Evolution Strategy (MO-CMA-ES) on considered benchmarks and performance metrics. This performance has been achieved either in complete absence of the hypervolume indicator in CMA-PAES or with grid level use of the hypervolume indicator in m-CMA-PAES.

It is now desirable to use elements from CMA-PAES and m-CMA-PAES to develop an algorithm specifically for the optimisation of many-objective problems, allowing the use of Covariance Matrix Adaptation (CMA) in an extensible framework without the computational infeasibility from using a population-wide hypervolume calculation.

The algorithm will then be benchmarked against the competition winning

Multi-Objective Evolutionary Algorithm Based on Decomposition with Dynamical Resource Allocation (MOEA/D-DRA) on test functions consisting of up to ten objectives. MO-CMA-ES will no longer be considered in comparison as it is not feasible to execute the algorithm on test functions consisting of greater than three objectives [73]. The Nondominated Sorting Genetic Algorithm III (NSGA-III) has not been considered for comparison as it has not been adequately benchmarked with a sufficient sample size, sufficient number of considered test problems, or the appropriate use of non-parametric testing to report the significance of the results. NSGA-III has been discussed in Appendix B.1. Another algorithm not considered but related is Differential Evolution for Multi-objective Optimisation with Self Adaptation (DEMOwSA), this is because it is not focussed on many-objective optimisation which is the focus of this thesis, and does not offer a comparison to any other Evolutionary Multi-Objective Optimisation (EMO) algorithm as a benchmark of its performance. MOEA/D-DRA also uses differential evolution and has also been benchmarked extensively against other algorithms, in particular in the CEC09 competition described in [92]. DEMOwSA has been discussed in Appendix B.2.

4.1 CMA-PAES on Many-objective Problems

An implementation of CMA-PAES including the grid-level hypervolume indicator selection scheme from m-CMA-PAES was executed on the WFG3 test function from the WFG tool-kit. The test function was configured with seven objectives to assess CMA-PAES on a many-objective problem. CMA-PAES was configured with a population and archive capacity of 100, this number was chosen so as to reduce the number of function evaluations per generation and to produce a

final approximation set that would not overwhelm a Decision Maker (DM). The hypervolume indicator performance at each generation has been illustrated in Figure 4.1.

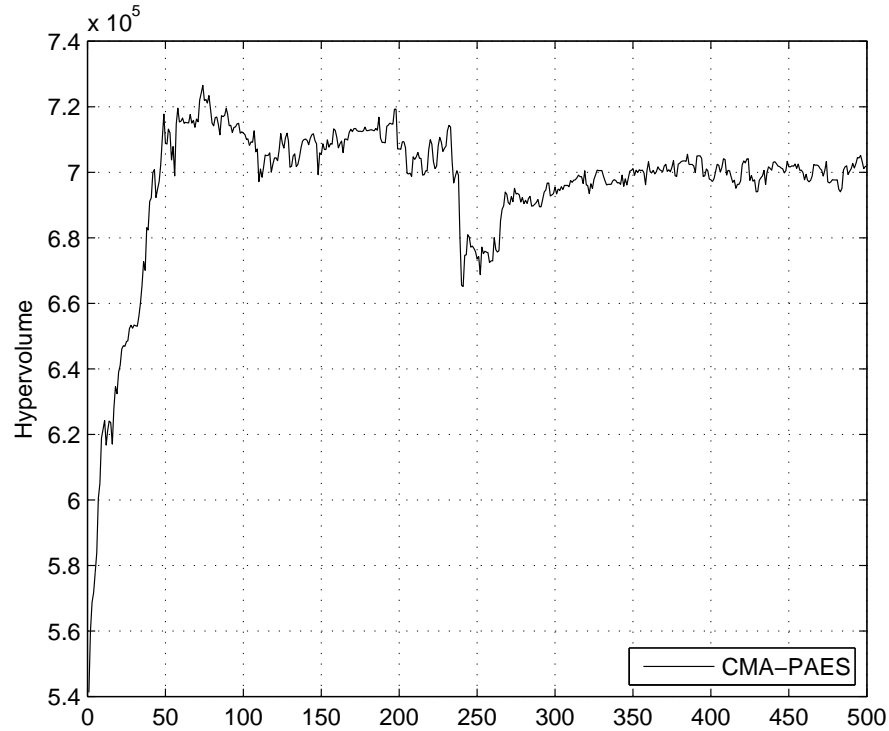


Figure 4.1: Plot of the hypervolume indicator performance at each generation of CMA-PAES on the WFG3 synthetic test problem.

From this figure it can be observed that between generation 50 and 100, CMA-PAES achieves peak hypervolume performance and then regresses to worse performing populations and oscillates in performance until termination, where it ultimately concludes on a population performing worse than some populations preceding it. This is because of several factors. The m-CMA-PAES algorithm uses the contributing hypervolume indicator for selection only at a second level (within grids), therefore it is not designed to improve hypervolume performance of the entire population from generation to generation, this design choice was made

to allow CMA-PAES to be feasible for execution on problems of greater than three objectives. CMA-PAES is also designed to optimise using a small population (for example 100) regardless of the number of objectives being optimised, this means diversity preservation is important because the number of available solutions in the population are limited.

In order to adapt CMA-PAES to perform well on many-objective problems, some modifications to the algorithm were needed. The following sub-sections introduce these modifications, followed by a section describing the full CMA-PAES-II algorithm.

4.2 Indicator Based Conformation

CMA-PAES-II uses the contributing hypervolume indicator as a second-level selection criteria. This is because the hypervolume indicator does not require a reference set - only a reference point (this can be approximated or set as the extremes found for each objective during the optimisation process). This is desirable in real-world optimisation problems where the true Pareto-optimal front is not known.

As the number of grid divisions increase, the accuracy of the contributing hypervolume indicator for second level selection decreases in regards to the overall hypervolume indicator quality of the population. Reducing the number of grid divisions would reduce the accuracy of the first level selection criterion, which is grid location, and also increase the computational cost of the contributing hypervolume indicator calculations by subjecting a higher number of solutions to the hypervolume indicator at once.

It is well known in the EMO literature that although Pareto dominance based algorithms can perform successfully on multi-objective problems [152, 59, 153], they do not always perform well on many-objective problems consisting of three more problem objectives [8, 9, 10, 11, 12, 13, 14]. Pareto dominance struggles to produce a strong selection pressure toward the Pareto-optimal front in the presence of many objectives, as throughout the optimisation process it is likely that the entire population will consist of entirely non-dominated solutions. Indicator-Based Evolutionary Algorithms (IBEAs) have been designed to incorporate performance indicators to produce stronger selection pressure toward the Pareto-optimal front [84, 154, 155], however these also suffer from the computational infeasibility of subjecting an entire population to a performance indicator (i.e. the hypervolume indicator) at each generation of the optimisation process.

One solution is to steer the optimisation process back towards a generation where it was at its peak performance, before it fell into a local optima, non-diverse population, or other fault. This can be seen visually in Figure 4.1 at approximately 75 generations into the optimisation process, where performance dropped from this point onward. As an observer of this illustration, it is possible to decide when it would be wise to revert to a previous population and continue in a different search direction. This can be automated for any performance indicator so that after a number of generations where the indicator performance lowers, the algorithm can exploit a population that conforms to the current executions peak performance. This feature has been named Indicator Based Conformation (IBC).

The IBC mechanism execution-cycle has been illustrated in Figure 4.2, and in the case of CMA-PAES-II, IBC will be based on the hypervolume indicator.

During the optimisation process, each population found at every generation is subjected to the hypervolume indicator, the performance is checked against a variable storing the peak performance of the current execution and replaces it if it is greater. If the hypervolume has been lower than the current execution's peak performance for a number of generations defined by the IBC threshold, then IBC will merge both the current population and the population which has the peak performance into a single intergenerational population. The AGA scheme is then applied to this population using significantly fewer grid divisions than the primary optimisation process. This is slower, but provides better accuracy in terms of hypervolume indicator performance, with the aim to move the next generation in a direction that will improve on the current peak performance.

The IBC threshold, Θ , is a variable which changes throughout the optimisation process but must remain within the boundaries of the IBC minimum threshold, Θ_{min} , and the IBC maximum threshold, Θ_{max} . Θ is initialised as the value of Θ_{max} , which can be set to the maximum number of unsuccessful generations which the DM can afford in the optimisation process, but must not be so low that the IBC mechanism is repeatedly triggered. A recommendation for the value of the IBC minimum threshold is:

$$\Theta_{min} = \frac{G}{100} \quad (4.1)$$

where G is the maximum number of generations configured for the optimisation process. A recommendation for the value of the IBC maximum threshold is:

$$\Theta_{max} = 4\Theta_{min} \quad (4.2)$$

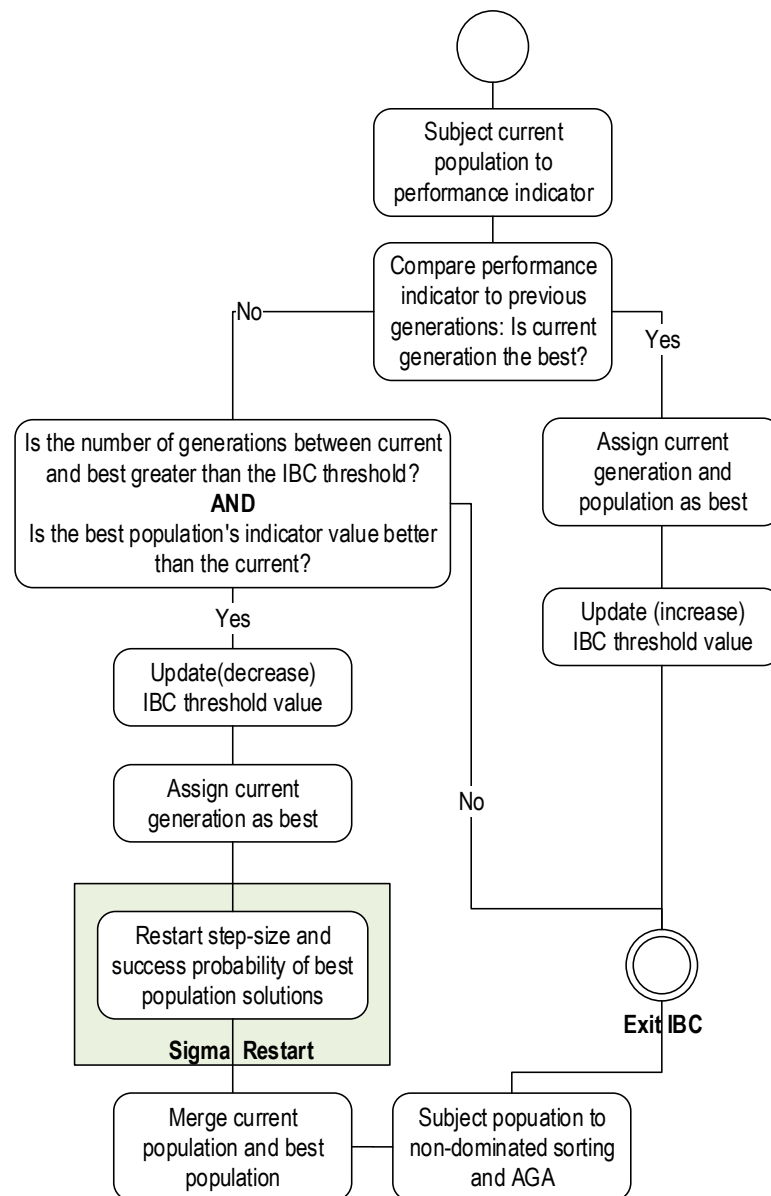


Figure 4.2: Execution life-cycle for the IBC mechanism.

In Figure 4.1 the oscillation in hypervolume indicator performance can be clearly observed, if the IBC mechanism had been incorporated during the optimisation process this negatively performing oscillation may have been prevented. Incorporating the IBC mechanism into the same CMA-PAES implementation and executing the algorithm on the same test problem configuration produces the results presented in Figure 4.3. This plot shows the hypervolume indicator performance of both CMA-PAES with and without IBC, with plotted points indicating the occurrence of the IBC mechanism, the result of which clearly outperforms the implementation of CMA-PAES without IBC. In this example, the maximum number of generations G was set to 500, resulting in a Θ_{min} of 5 and Θ_{max} of 20. In the event that Θ is set to 20 and the IBC conforms to a population which continues to perform poorly for another Θ generations, a total of 40 generations are wasted with no improvement in performance. Therefore it is desirable to adapt Θ so that it is sensitive to success and failure.

For example, the performance i of the current population P is calculated using the hypervolume indicator and then compared to the performance of the best performance found in the current execution B_i . If $i > B_i$, then the current population is stored as the best population found in the current execution, along with the generation it was found within and the performance indicator value:

$$\begin{aligned}
 B_p &= P \\
 B_g &= g \\
 B_i &= i
 \end{aligned}
 \tag{4.3}$$

where B_p is the best population found in the current execution and B_i is the performance indicator value of B_p . This execution path is seen as successful and

as a result the Θ is increased by multiplication with the IBC threshold success variable Θ_{succ} (recommended to be set at 1.05):

$$\Theta = \lceil \Theta_{succ} \Theta \rceil \quad (4.4)$$

and reset to a value between Θ_{min} and Θ_{max} if it falls outside that boundary, before continuing to the next generation of the algorithm execution life-cycle:

$$\Theta = \begin{cases} \Theta_{min}, & \text{if } \Theta < \Theta_{min} \\ \Theta_{max}, & \text{if } \Theta > \Theta_{max} \\ \Theta, & \text{otherwise.} \end{cases} \quad (4.5)$$

However, if the hypervolume indicator value of the current population is less than the hypervolume indicator value of the best found population, and the difference between the number of the current generation and the generation at which the best found population was updated is greater than Θ :

$$(B_i > i) \wedge (g - B_g > \Theta) \quad (4.6)$$

then this execution path is treated as unsuccessful and as a result Θ is decreased by multiplication with the IBC threshold failure variable Θ_{fail} (recommended to be set at 0.5):

$$\Theta = \lceil \Theta_{fail} \Theta \rceil \quad (4.7)$$

and reset to a value between Θ_{min} and Θ_{max} if it falls outside that boundary using Equation 4.5. The number of the current generation g is then stored in B_g and the best found population B_p is copied to a temporary population B_{p^*} before it is merged with the current population P :

$$\begin{aligned}
 P &= P \cup B_{p^*} \\
 B_g &= g
 \end{aligned}
 \tag{4.8}$$

The temporary population B_{p^*} is used later in Section 4.3 for achieving sigma restart as part of the IBC mechanism. The population P is then subjected to non-dominated sorting and the AGA scheme, before the algorithm continues to the next generation. In the event that the hypervolume indicator value of the current population is less than the hypervolume indicator value of the best found population, but the difference between the number of the current generation and the generation at which the best found population was updated is less than Θ , the algorithm simply continues to the next generation.

This process means that once Θ generations have passed without the peak performance improving, IBC will execute and Θ will become the value resolved from Equation 4.7, meaning there will only be an allowance of ten generations of failure to improve upon the peak performance before IBC executes again. This process of division by two will be limited to Θ_{min} , meaning in the worst case scenario IBC will be executed every five generations until a successful optimisation path is found. In contrast, after IBC if the next generation does indeed outperform the current peak performing population, Θ will become the value resolved from Equation 4.4, for example going from 10 to 11, and will continue to do so for every successful (raising peak performance) generation within the limit of Θ_{max} . This will allow more room for failure for populations which have shown promise by raising the peak performance in recent generations.

IBC was incorporated into an implementation of CMA-PAES and again executed on the WFG3 test function configured with seven objectives, with the

algorithm parameter configuration:

$$\begin{aligned}\Theta_{min} &= 5 \\ \Theta_{max} &= 20 \\ IBC_{div} &= 2 \\ AGA_{div} &= 10\end{aligned}\tag{4.9}$$

where IBC_{div} is the number of grid divisions used for the AGA during IBC, and AGA_{div} is the number of grid divisions used for the AGA during the primary optimisation process. In addition, the seed used for generating the random numbers during the optimisation process was kept the same as the seed used in the execution which produced the results in Figure 4.1. The results for this execution have been illustrated in Figure 4.3. During the optimisation process, IBC executed 67 times, and the benefits can be seen in Figure 4.3 where both implementations of CMA-PAES-II with and without IBC are identical in hypervolume indicator performance until approximately 100 generations where the first IBC is executed, from that point the implementation of CMA-PAES-II with IBC continues to achieve better hypervolume indicator performance.

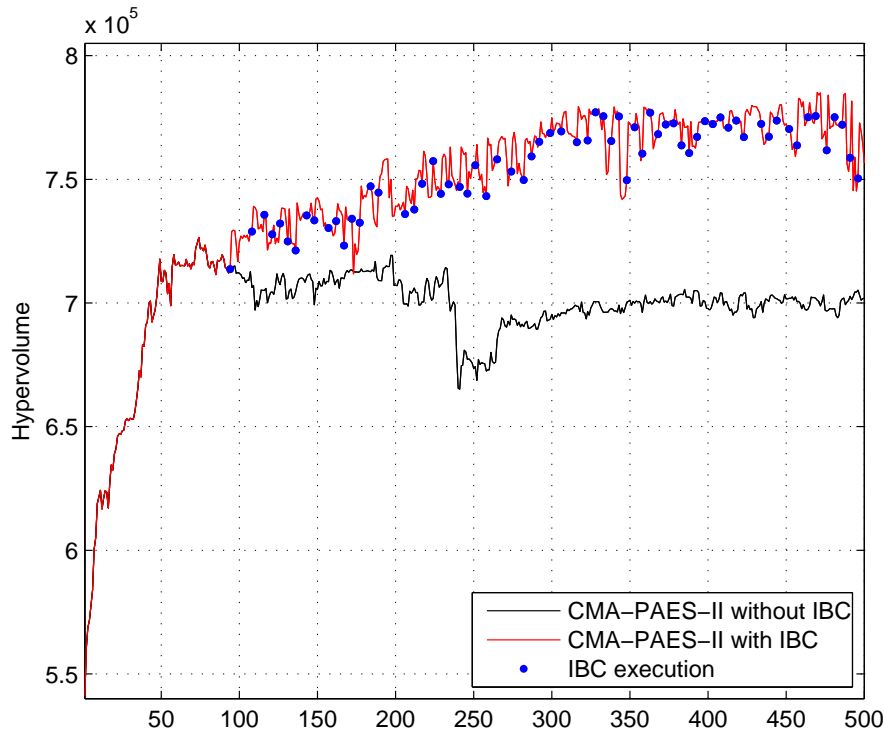


Figure 4.3: Plot of the hypervolume indicator performance at each generation of CMA-PAES-II without IBC and CMA-PAES-II with IBC on the WFG3 synthetic test problem. The generations where IBC was executed have been plotted with blue points.

4.3 Sigma Restart

During an execution of an EMO algorithm with the incorporation of the IBC mechanism, the population which has been stored as the best population can be assumed to be responsible for leading future generations toward populations which score worse performance. One method to prevent repetition of past failure when executing the IBC mechanism (and merging the current population and best population) is to restart the sigma (step-size) values of the population, and allow them to re-propagate to the current state of the search. For example, during the optimisation process the current population would have a lower sigma than

the sigma of the best performing population, as there would have been Θ failed generations in order for the IBC mechanism to execute, and without a sigma restart the solutions in the current population would continue to make very little improvement in performance.

The shaded and labelled area of Figure 4.2 indicates where the sigma restart takes place in the IBC mechanism. This is achieved by setting the sigma and success probability of each solution from the best found population (which as described in Section 4.2 is stored in the temporary population B_{p^*}) to their initial configurations.

$$\begin{aligned}\sigma &= \sigma_{init} \\ \bar{p}_{succ,i} &= psucc_{init}\end{aligned}\tag{4.10}$$

where $\bar{p}_{succ,i} \in [0, 1]$ is the smoothed success probability and $\sigma \in \mathbb{R}_0^+$ is the global step size.

Sigma restart was implemented into the design of CMA-PAES-II and again executed on the WFG3 test function configured with seven objectives, with the configuration $\Theta_{min} = 5$, $\Theta_{max} = 20$, $IBC_{div} = 2$, $AGA_{div} = 10$, where IBC_{div} is the number of grid divisions used for the AGA during IBC, and AGA_{div} is the number of grid divisions used for the AGA during the primary optimisation process. In addition, the seed used for generating the random numbers during the optimisation process was kept the same as the seed used in the execution which produced the results in Figures 4.1 and 4.3. The results for this execution have been illustrated in Figure 4.4. During the optimisation process IBC combined with sigma restart was executed 62 times, which is five times less than the instance without the sigma restart. The benefits can be seen in Figure 4.4 where

both implementations of CMA-PAES-II with and without IBC are identical in hypervolume indicator performance until approximately 100 generations where the first IBC is executed, from that point the implementation of CMA-PAES-II with IBC continues to achieve better hypervolume indicator performance. When comparing these results to the results achieved by IBC without sigma restart, it can be observed that CMA-PAES-II with IBC and sigma restart achieved better hypervolume performance with fewer IBC executions.

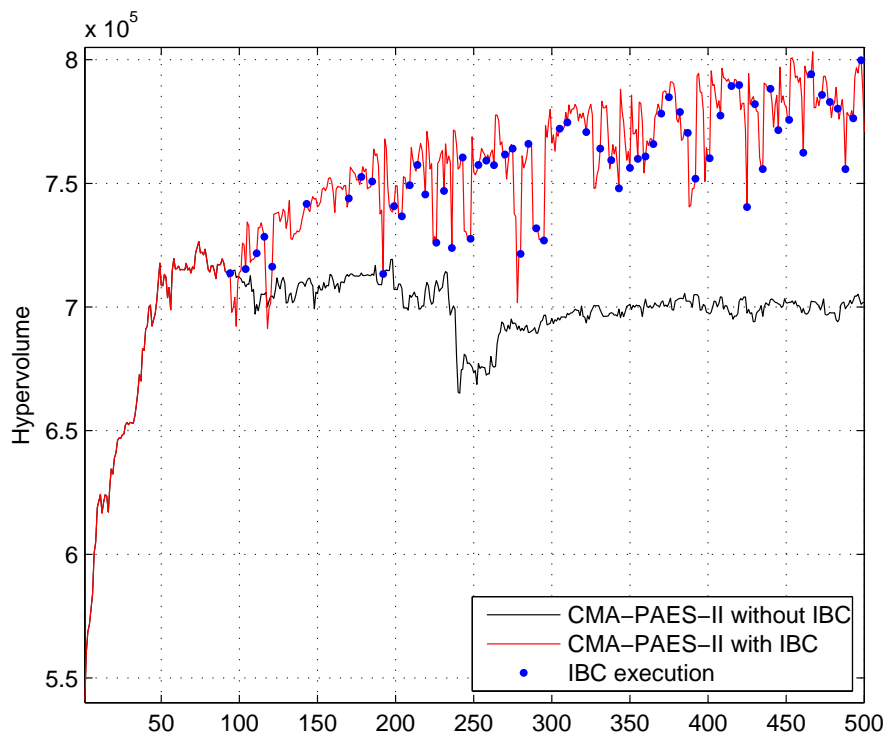


Figure 4.4: Plot of the hypervolume indicator performance at each generation of CMA-PAES-II without IBC and CMA-PAES-II with IBC and sigma restart on the WFG3 synthetic test problem. The generations where IBC was executed have been plotted with blue points.

4.4 Improved Adaptive Grid Algorithm

CMA-PAES-II incorporates the use of an updated AGA implementation containing a number of changes from the AGAs used in CMA-PAES and m-CMA-PAES, in order to make the AGA implementation suitable for many-objective optimisation. These changes consist of:

- A new data structure for storing a solution's grid number.
- An improved scheme for grid selection when searching for a solution to replace.
- Maintenance of global extremes for problem objectives.

The mathematical procedure for the improved AGA in its entirety is described herein. M defines the number of problem objectives and N defines the population size, whilst Δ defines the number of desired grid divisions for a problem objective within the objective space. X is an M by N matrix of entries x_{mn} , where every x_{mn} refers to a solution's objective value:

$$X_n = \langle x_{1n}, x_{2n}, \dots, x_{Mn} \rangle \quad (4.11)$$

Γ is an M by N matrix of entries γ_{mn} , where every γ_{mn} refers to the grid location of an objective value x_{mn} in the divided objective space.

$$\Gamma_n = \langle \gamma_{1n}, \gamma_{2n}, \dots, \gamma_{Mn} \rangle \quad (4.12)$$

To calculate Γ_n , the grid location γ_{mn} of each objective value x_{mn} for each solution X_n needs to be resolved. To calculate a solution's grid location, the padded grid length Λ

$$\Lambda = \langle \lambda_1, \lambda_2, \dots, \lambda_M \rangle \quad (4.13)$$

for each objective needs to be calculated using the lowest and highest objective value for each objective in the population:

$$\begin{aligned} \lambda_m &= \frac{|\min(X_m) - \max(X_m)|}{\Delta} \\ \lambda_m^{lower} &= \min(X_m) - \lambda_m \\ \lambda_m^{upper} &= \max(X_m) + \lambda_m \end{aligned} \quad (4.14)$$

where λ_m^{lower} is the start point of the grid for objective m in the objective space, and λ_m^{upper} is the end point of the grid for objective m in the objective space. With the grid length and range calculated, it is possible to get the grid location of each solution's objective value using:

$$\gamma_{mn} = \left\lceil \frac{x_{mn} - \lambda_m^{lower}}{\frac{\lambda_m}{\Delta}} \right\rceil \quad (4.15)$$

When the entries of Γ_n have been calculated, it can be used to identify the grid location of a solution X_n . In this new method, the grid location Γ_n is defined by a vector rather than a scalar, for example in a five-objective problem a grid location can be described by being at location $\Gamma_n = \langle 2, 4, 1, 1, 2 \rangle$.

As an example, a population X of five ($N = 5$) solutions X_n for a five-objective problem ($M = 5$) has been presented in Table 4.1 and Figure 4.5.

This population X has been subjected to the improved AGA scheme to resolve the grid location Γ_n of each solution X_n , with an AGA configuration of four grid divisions ($\Delta = 4$). The grid locations resolved by the AGA scheme have been presented in Table 4.2 and the objective values x_{mn} have been plotted in their

Table 4.1: An example population X of objective values x_{mn} to be subjected to the improved AGA scheme.

	x_{1n}	x_{2n}	x_{3n}	x_{4n}	x_{5n}
X_1	0.5	0.5	5.0	2.5	1.5
X_2	0.6	0	5.0	3.0	1.4
X_3	0.5	3.5	4.5	2.5	1.5
X_4	0.8	3.2	4.2	3.0	1.2
X_5	1	3	4	2	1

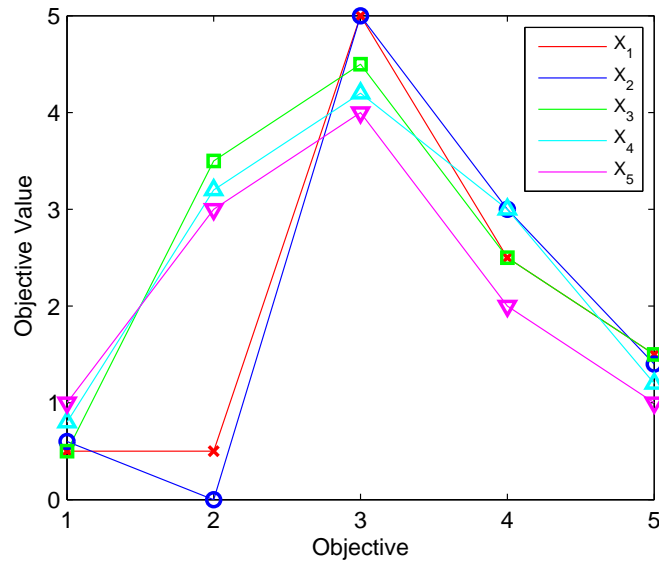


Figure 4.5: Parallel-coordinate plot of the Population X used in the improved AGA example.

respective grid locations γ_m in Figure 4.6, where the plot markers correspond to those used in Figure 4.5.

The results from this example show that the example population does not consist of any solutions which are in the same grid square (otherwise their Γ entries would be identical).

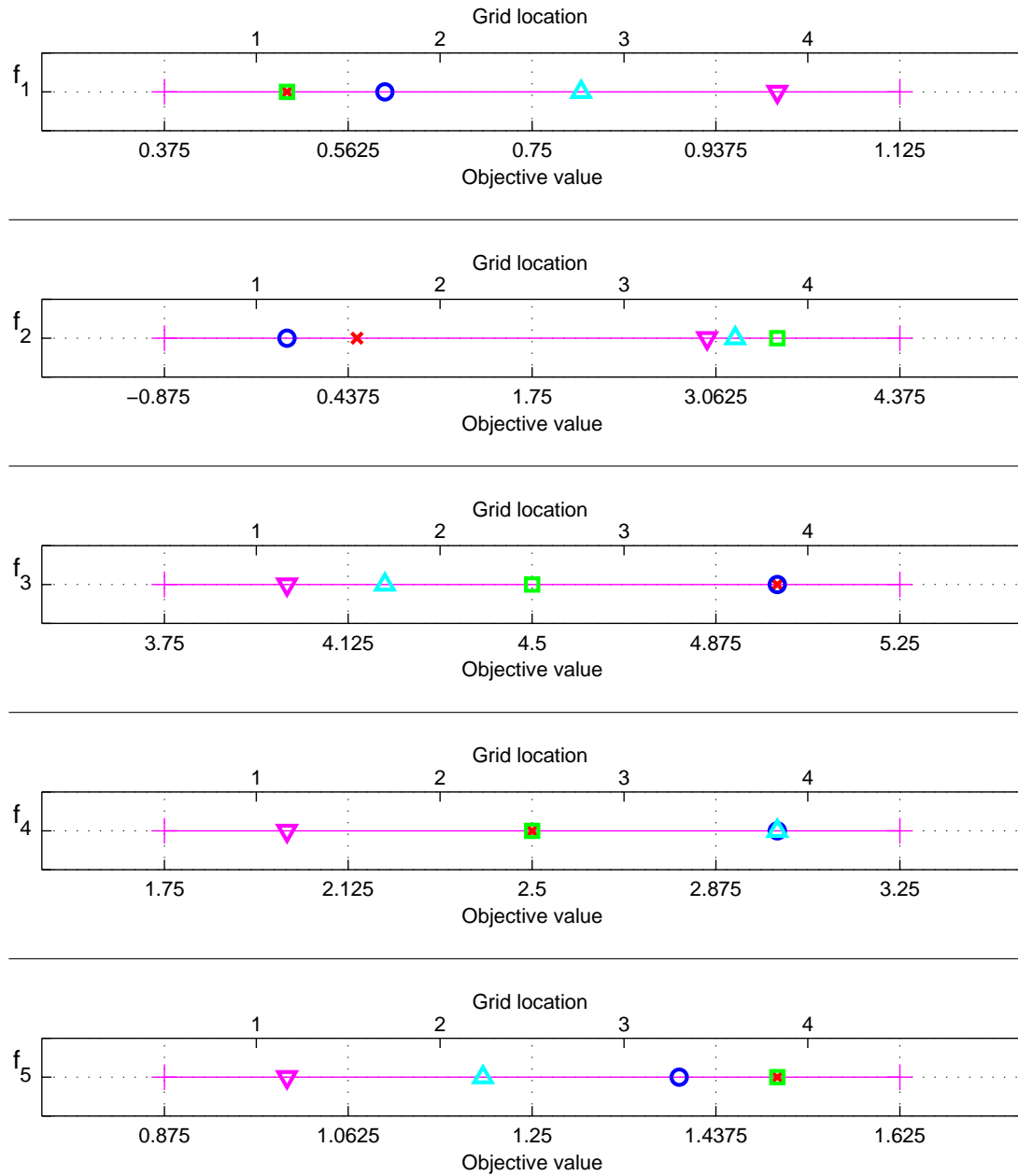


Figure 4.6: One dimensional plots illustrating the grid locations resolved by the AGA scheme for each objective value, where the plot markers correspond to those used in Figure 4.5.

Table 4.2: Grid locations Γ for the example population X of objective values x_{mn} .

	γ_{1n}	γ_{2n}	γ_{3n}	γ_{4n}	γ_{5n}
Γ_1	1	1	4	3	4
Γ_2	2	1	4	4	3
Γ_3	1	4	2	3	4
Γ_4	3	4	2	4	2
Γ_5	4	4	1	1	1

The method for selecting a grid location to replace a solution when the archive is at capacity has also been modified. Previously in CMA-PAES a grid location was selected at random from grid locations which were at the same density, however this could cause genetic drift and doesn't provide the best overall diversity. Instead, it is desirable to find the grid location which is closest in proximity to the candidate solution in the objective space and also at a higher density. CMA-PAES stores grid locations as a single scalar value, this is not helpful when calculating distance between grid locations or for storing grid locations for a many-objective problem. The grid location structure used in the improved AGA scheme described above enables an intuitive method for finding the distance between grid locations. By establishing the grid location which a candidate solution would be assigned if it was part of the archive, it is possible to find the difference between its grid location and other grid locations which are at high density to find out which one it's closest to by summing the grid location vector.

For example, if a new solution $X_6 = \langle 0.6, 0.5, 4, 3, 1.1 \rangle$ was to be included as a candidate solution as part of the improved AGA scheme, it would resolve a grid location of $\Gamma_6 = \langle 2, 1, 1, 4, 2 \rangle$. The distance δ_n between this grid location

and the grid locations of the other solutions can be found by finding the absolute difference of each corresponding entry of the candidate solution's grid location and another solution from the population, and then summing those values.

$$\delta_n = \sum_{n=1}^N |\gamma_m^* - \gamma_{mn}| \quad (4.16)$$

Where γ_m^* is the γ_{mn} values for the candidate solution X_6 . The distances δ_n between the grid location Γ_6 of solution X_6 and all the other solutions in the population presented in Table 4.1 have been presented in Table 4.3. The results show that the solution closest in proximity to solution X_6 is solution X_2 , this has been visualised in Figure 4.7.

Table 4.3: Grid locations Γ for the example population X of objective values x_{mn} .

	Γ_1	Γ_2	Γ_3	Γ_4	Γ_5
δ_6	7	4	8	5	9

One final modification has been made in the improved AGA. Previously, in CMA-PAES, the extreme values for each objective were preserved at grid level. In the new AGA scheme, solutions containing extreme values for problem objectives (with the candidate solution taken into consideration) are removed from the population before it is subjected to the AGA. This ensures candidate solutions are given a better chance of entering the archive than they would have had if they had come up against those solutions containing extreme values. This preserves the overall spread whilst encouraging new solutions to enter the archive.

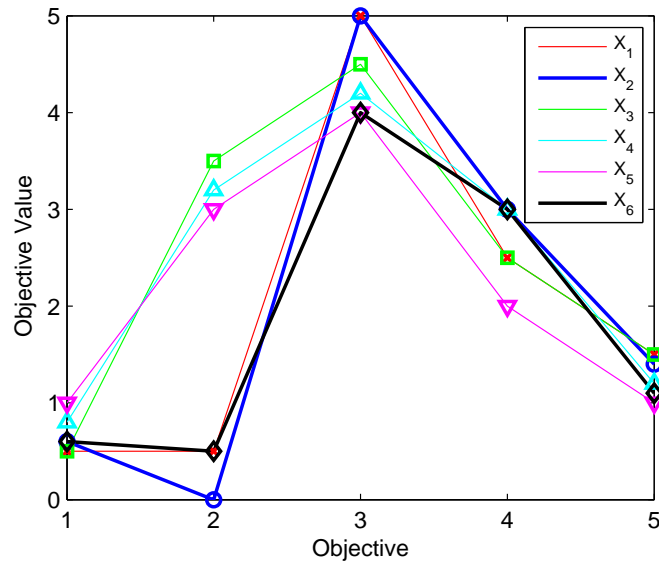


Figure 4.7: Parallel-coordinate plot of the Population X used in the improved AGA example. X_6 and the solution closest in proximity to it, X_2 , have been presented using thicker lines.

4.5 CMA-PAES-II Design

CMA-PAES-II has been designed with the intention of being used for many-objective optimisation. This has been achieved through a combination of features from CMA-PAES, m-CMA-PAES, and with new features which have been developed as a result of preliminary experiments of CMA-PAES on many-objective problems (discussed in Section 4.1). The execution life-cycle for CMA-PAES-II has been presented in Figure 4.8.

CMA-PAES-II begins by initialising algorithm parameters and randomly sampling the search-space to generate an initial parent population, X , of size μ , which is then evaluated by the problem objective function. At this point, the generational loop begins by checking whether the configured termination criteria (a configured maximum number of function evaluations) has been met, and if so the

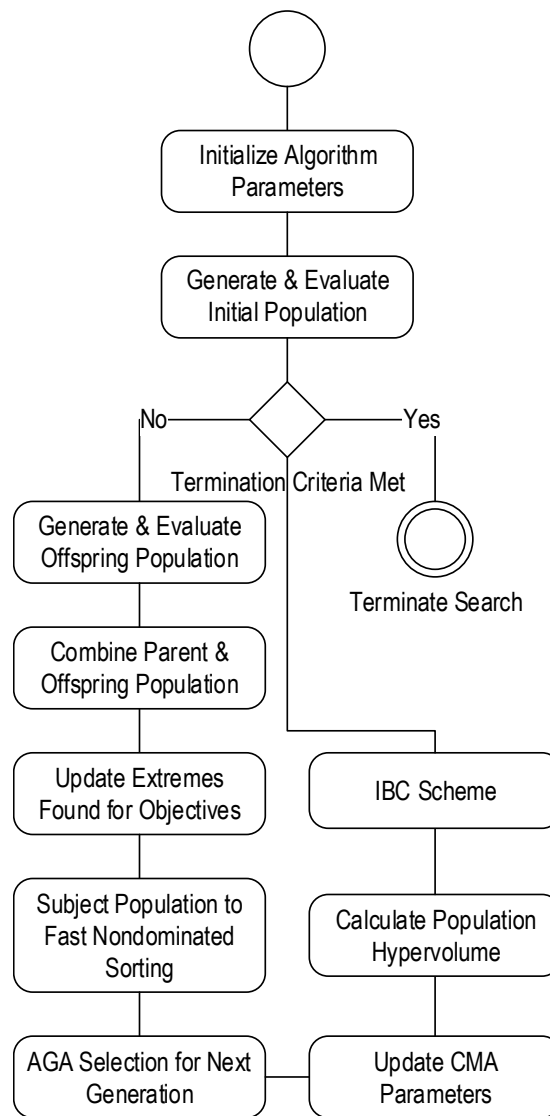


Figure 4.8: Execution life-cycle for the CMA-PAES-II algorithm, where IBC Scheme refers to the execution life-cycle illustrated in Figure 4.2.

EMO process is terminated. X is an M by N matrix of entries x_{mn} , where every x_{mn} refers to a solution's objective value.

$$X_n = \langle x_{1n}, x_{2n}, \dots, x_{Mn} \rangle \quad (4.17)$$

If the termination criteria has not been met, CMA-PAES-II continues to generate and evaluate an offspring population X' of size λ using the CMA operator for variance. The parent population X and offspring population X' are then merged to create an intermediate population X^* to be used in the following processes of the algorithm.

$$X^* = X \cup X' \quad (4.18)$$

The extreme values ϵ_m encountered for each problem objective during the optimisation process are then updated by checking if any objective value x_{mn}^* is higher than a corresponding stored extreme objective value ϵ_m , and if so, replacing it.

$$\epsilon_m = \begin{cases} x_{mn}^* & \text{if } x_{mn}^* > \epsilon_m \\ \epsilon_m & \text{otherwise} \end{cases} \quad (4.19)$$

where E is a vector containing all of the extreme values encountered for each problem objective.

$$E = \langle \epsilon_1, \epsilon_2, \dots, \epsilon_M \rangle \quad (4.20)$$

The intermediate population X^* is then sorted using the fast non-dominated sorting procedure described in Section 2.7.1, before the improved AGA scheme described in Section 4.4 paired with the Sigma restart addition described in Section 4.3 is applied. This is achieved by subjecting each ranked population resolved

from the non-dominated sorting procedure to the improved AGA scheme, with the archive capacity for each execution of the improved AGA being either μ (in the case that the rank-1 non-dominated population is greater than μ) or the size of the rank-1 population otherwise. In the event that the rank-1 population size is not equal to or greater than μ , subsequent lower rank populations are used with an archive capacity setting of μ minus the cumulative size of previous ranked populations subjected to the AGA scheme, until the parent population for the next generation X^{g+1} has been populated with μ solutions. The parameters used for the CMA operator for variance are then updated similarly to that in Section 2.7.3, where solutions are considered successful if they make it from the intermediate population X^* to the parent population for the next generation X^{g+1} .

The performance i of the current generation is then assessed using the hypervolume indicator described in Section 2.10.1 and calculated using Equation 2.6, with the extreme values encountered in each problem objective E as the reference point f^{ref} .

$$i = S_E (X^{g+1}) \quad (4.21)$$

This performance value i is then used in the IBC scheme described in Section 4.2 in order to prevent the algorithm from prematurely converging to local optima. The IBC scheme is only considered after the first 5 generations of CMA-PAES-II, to allow the encountered extreme objective values E to propagate. The optimisation process then continues to the next generational iteration.

4.6 Comparison between CMA-PAES-II and MOEA/D-DRA

CMA-PAES-II and MOEA/D-DRA have been configured with a budget of 50,000 function evaluations per algorithm execution to ensure fair comparison. The algorithm configurations are presented in Table 4.4. Note that MOEA/D-DRA specific configurations have been taken from [93].

Table 4.4: Parameter configurations used for testing CMA-PAES-II and MOEA/D-DRA.

Parameter	CMA-PAES-II	MOEA/D-DRA
μ Population	100	100
λ Offspring	100	100
Niche Population Size	—	20
Archive Capacity	100	—
AGA Grid Divisions	4	—
IBC Grid Divisions	2	—
Θ_{min}	5	—
Θ_{max}	20	—

The WFG tool-kit has been selected for the benchmarking and comparison of CMA-PAES-II and MOEA/D-DRA, these test functions will pose difficulties to the optimisers and also allow for testing the considered algorithms on many-objective problems. The test functions used for this experiment are WFG1, WFG2, WFG3, WFG4, WFG5, WFG6, WFG7, WFG8, and WFG9. The configurations used for these test functions are given in Table 4.5.

Table 4.5: Parameter configurations used for the WFG tool-kit.

Parameter	Value
Number of objectives M	2, 3, 5, 7, 10
Total Variables n	24
Position Related Variables k	Equation 4.22
Distance Related Variables l	$n - k$

Each considered algorithm will be executed 30 times on each test function to reduce stochastic noise for objectives two, three, five, seven, and ten. This sample size has been selected in order to comply with the CEC09 competition rules described in [92] and seen as sufficient because of the limited benefit of producing more than 25 samples (discussed in Section 2.10.5).

The WFG tool-kit requires the configuration of test function parameters such as M for the number of objectives, n for the number of total parameters, k for the number of position related parameters which is resolved using Equation 4.22, and l which is resolved from $n - k$.

$$k = \begin{cases} 2(M-1) & \text{if } M \geq 2 \\ 4 & \text{otherwise} \end{cases} \quad (4.22)$$

The metric used for performance assessment is the hypervolume indicator described in Section 2.10.1. This metric determines the coverage of the objective space (assessing both proximity and diversity) of any given approximation set, without the requirement of knowledge of the true Pareto-optimal front, this is a necessary feature of a performance metric in most real-world problems, as the true Pareto-optimal front is often not known. The hypervolume indicator will be used at each generation in order to assess performance and compare both

algorithms on not just the hypervolume quality of the final approximation set but also the hypervolume quality over time.

4.7 Results

The results from the experiments in Section 4.6 have been produced and presented in a number of formats in order to allow for a better assessment of each algorithms performance.

The worst, mean, and best IGD indicator results for the final approximation set of each algorithm are presented in Tables 4.6, 4.7, 4.8, 4.9, and 4.10, for test functions containing two, three, five, seven and ten objectives respectively. In regards to the mean hypervolume indicator results of the final approximation set produced by each algorithm for each test function, CMA-PAES-II outperformed MOEA/D-DRA on 32 of the 45 test functions considered for the experiment.

Tables 4.6, 4.7, 4.8, 4.9, and 4.10 also present information regarding the p -value resolved by the Wilcoxon signed-ranks non-parametric test for the final approximation sets of the considered synthetic test problems, and a symbol indicating the observation of the null hypothesis. A '+' symbol indicates that the null hypothesis was rejected, and CMA-PAES-II displayed statistically superior performance at the 95% significance level ($\alpha = 0.05$) on the considered synthetic test function. A '-' symbol indicates that the null hypothesis was rejected, and CMA-PAES-II displayed statistically inferior performance. An '=' symbol indicates that there was no statistically significant difference between both of the considered algorithms on the synthetic test problem.

The means of the IGD metric at each generation have been plotted and presented in Figures 4.10, 4.11, 4.12, 4.13, and 4.14, for test functions containing two,

three, five, seven and ten objectives respectively. These plots illustrate the rate of hypervolume indicator convergence from the initial population to the final population. From these plots it can be observed that at two-objective configurations of the WFG tool-kit, MOEA/D-DRA significantly outperforms CMA-PAES-II in regards to rate of convergence, however as the number of objectives increase, CMA-PAES-II becomes the algorithm with the faster and better performing convergence.

The box plots in Figure 4.9 allows for comparison on the dispersion, skew and outliers in the performance for the final approximation set of each algorithm on each considered test function.

Results analysis has been divided into sections based on the number of objectives being optimised for clarity.

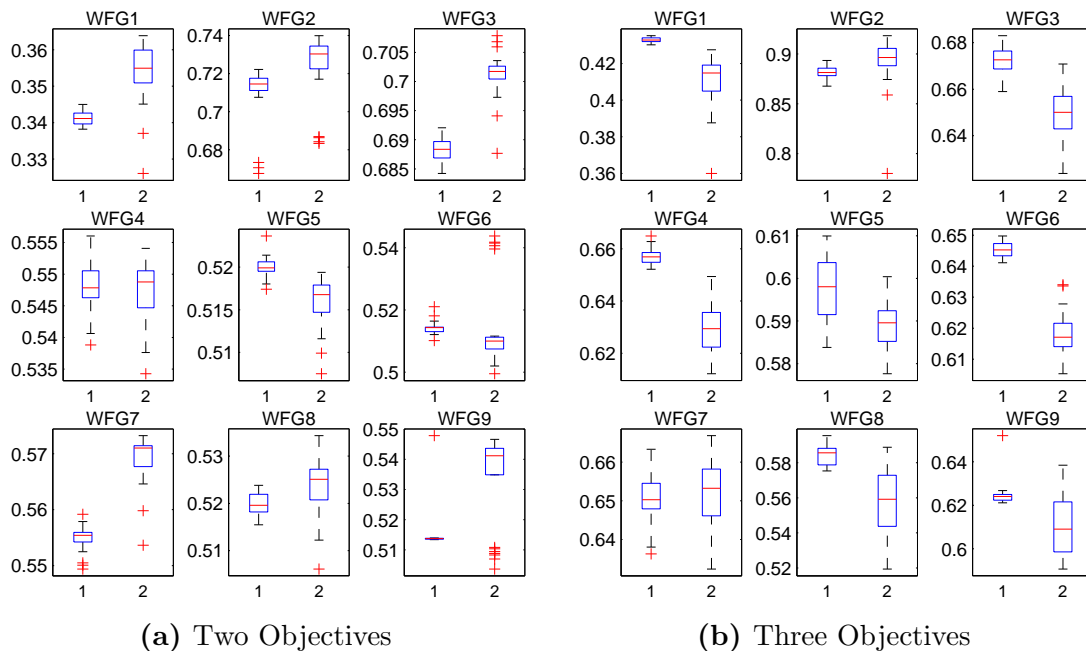
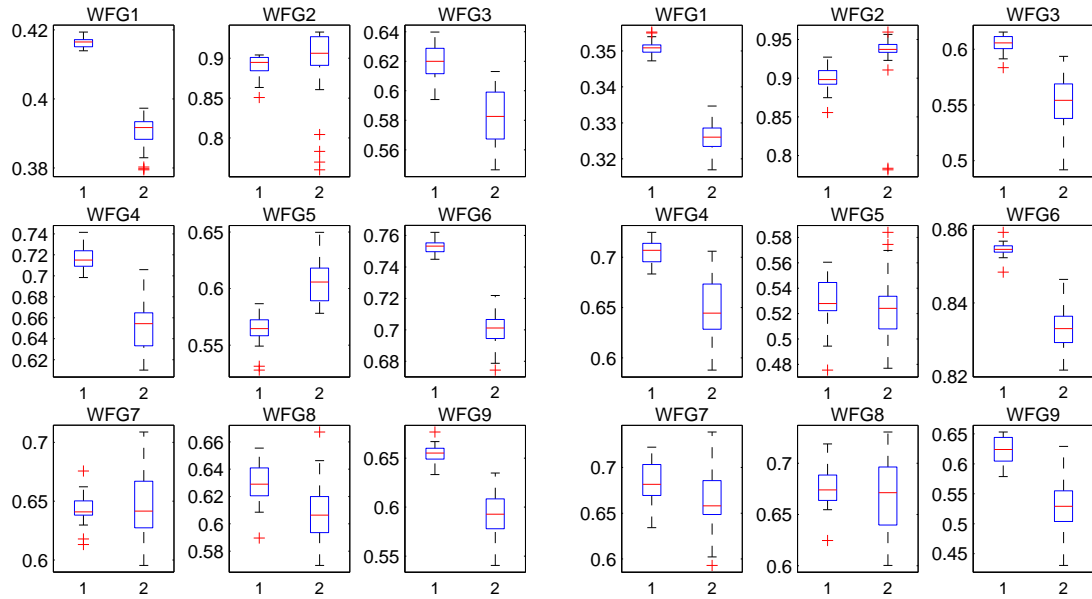
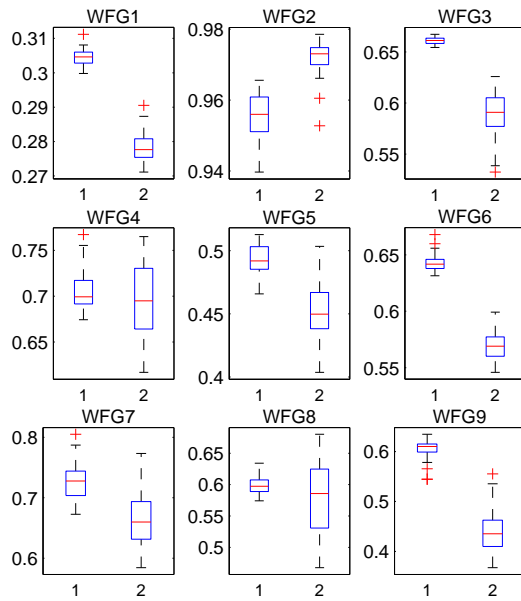


Figure 4.9: Box plots of hypervolume indicator results for WFG problems (set 1: CMA-PAES-II; set 2: MOEA/D-DRA) 50,000 function evaluations, 30 runs.



(c) Five Objectives

(d) Seven Objectives



(e) Ten Objectives

Figure 4.9: Box plots of hypervolume indicator results for WFG problems (set 1: CMA-PAES-II; set 2: MOEA/D-DRA) 50,000 function evaluations, 30 runs.

4.7.1 Two-Objective Results

Overall, it can be observed in Table 4.6 that MOEA/D-DRA outperforms CMA-PAES-II on the considered WFG test functions when each test function is configured to consist of two objectives. MOEA/D-DRA achieves better mean hypervolume indicator performance on six of the nine considered test functions and achieves the best performing final approximation set of the two algorithms on six of the nine considered test functions. However, in regards to worst-case performance, CMA-PAES-II outperforms MOEA/D-DRA by achieving the best performing worst case performance on six of the nine considered test functions.

Table 4.6: Hypervolume indicator results from 30 executions of CMA-PAES-II and MOEA/D-DRA on the WFG test suite with two objectives.

2D	CMA-PAES-II			MOEA/D-DRA			p -value	
	Worst	Mean	Best	Worst	Mean	Best		
WFG1	0.33819	0.34122	0.34499	0.32614	0.35422	0.36457	5.5727e-10	–
WFG2	0.66751	0.71044	0.72216	0.68334	0.72452	0.74016	1.3853e-06	–
WFG3	0.68422	0.68823	0.69205	0.68819	0.70138	0.70787	1.4643e-10	–
WFG4	0.53883	0.54789	0.55604	0.53456	0.54739	0.55347	1	=
WFG5	0.5174	0.52007	0.52365	0.5075	0.51592	0.51936	1.9568e-10	+
WFG6	0.51012	0.51432	0.52103	0.4995	0.51377	0.54229	1.7479e-05	+
WFG7	0.54937	0.55494	0.55917	0.55365	0.56899	0.57322	2.8716e-10	–
WFG8	0.51546	0.52004	0.52381	0.50616	0.52344	0.53442	0.0005264	–
WFG9	0.51339	0.51482	0.5479	0.50359	0.53395	0.5467	0.0013703	–

On WFG1, MOEA/D-DRA outperforms CMA-PAES-II in regards to mean and best hypervolume indicator performance of the final approximation sets. However, it can be seen in the plot of hypervolume indicator performance at each generation in Figure 4.10a that CMA-PAES-II achieves faster initial convergence up until approximately 100 generations, where MOEA-/D-DRA converges

steadily at a faster rate. There is also less dispersion in the individual executions which can be verified in the box-plot in Figure 4.9a, where it can also be observed that although MOEA/D-DRA achieves the two worst executions on WFG1, both executions are considered outliers.

On WFG2, MOEA/D-DRA outperforms CMA-PAES-II in regards to the mean, worst, and best hypervolume indicator performance of the final approximation sets. MOEA/D-DRA also achieves a better convergence rate which can be seen in Figure 4.10b. Both algorithms contain a high number of outliers as seen in Figure 4.9a, this may be a consequence of the WFG2 test function consisting of a disconnected geometry causing some executions of each algorithm to only resolve some parts of the objective space.

On WFG3, MOEA/D-DRA outperforms CMA-PAES-II in regards to the mean, worst, and best hypervolume indicator performance of the final approximation sets. MOEA/D-DRA also achieves a better convergence rate which can be seen in Figure 4.10c. However, Figure 4.9a suggests CMA-PAES-II achieves more robust executions as there is less dispersion in the final approximation sets and MOEA-D/DRA has many outliers both in the top and low end.

On WFG4, CMA-PAES-II outperforms MOEA/D-DRA in regards to the mean, worst, and best hypervolume indicator performance of the final approximation sets. Both algorithms achieve similar rates of convergence and dispersion which can be seen in Figures 4.10d and 4.9a.

On WFG5, CMA-PAES-II outperforms MOEA/D-DRA in regards to the mean, worst, and best hypervolume indicator performance of the final approximation sets. However, MOEA/D-DRA achieves a faster rate of mean convergence up until approximately 250 generations, this can be seen in Figure 4.14e.

CMA-PAES-II offers more robust performance, and Figure 4.9a shows that the interquartile range for CMA-PAES-II achieves better performance than the total range of MOEA/D-DRA.

On WFG6, CMA-PAES-II outperforms MOEA/D-DRA in regards to mean and worst hypervolume indicator performance, on the final approximation sets. Both algorithms achieve similar rates of convergence and dispersion which can be seen in Figure 4.10f, and in Figure 4.9a it is displayed that although MOEA/D-DRA achieves executions with the best performance, they are considered outliers.

On WFG7, MOEA/D-DRA outperforms CMA-PAES-II in regards to the mean, worst and best hypervolume indicator performance of the final approximation sets. MOEA/D-DRA also achieves a better convergence rate which can be seen in Figure 4.10g, and it can be observed in Figure 4.9a that the results for MOEA/D-DRA contain outliers in the bottom end and that the distribution is bottom skewed.

On WFG8, similar to WFG7, MOEA/D-DRA outperforms CMA-PAES-II in regards to mean and best hypervolume indicator performance of the final approximation sets. MOEA/D-DRA also achieves a better convergence rate which can be observed in Figure 4.10h, and Figure 4.9a illustrates that CMA-PAES-II offers more robust performance with a much smaller total range and no outliers.

On WFG9, MOEA/D-DRA outperforms CMA-PAES-II in regards to mean hypervolume indicator performance of the final approximation sets. MOEA/D-DRA also achieves a better convergence rate which can be observed in Figure 4.10i, and CMA-PAES-II achieves the execution with the best performance, though it is considered an outlier which can be shown in Figure 4.9a. MOEA/D-DRA also contains a high number of low performing outliers.

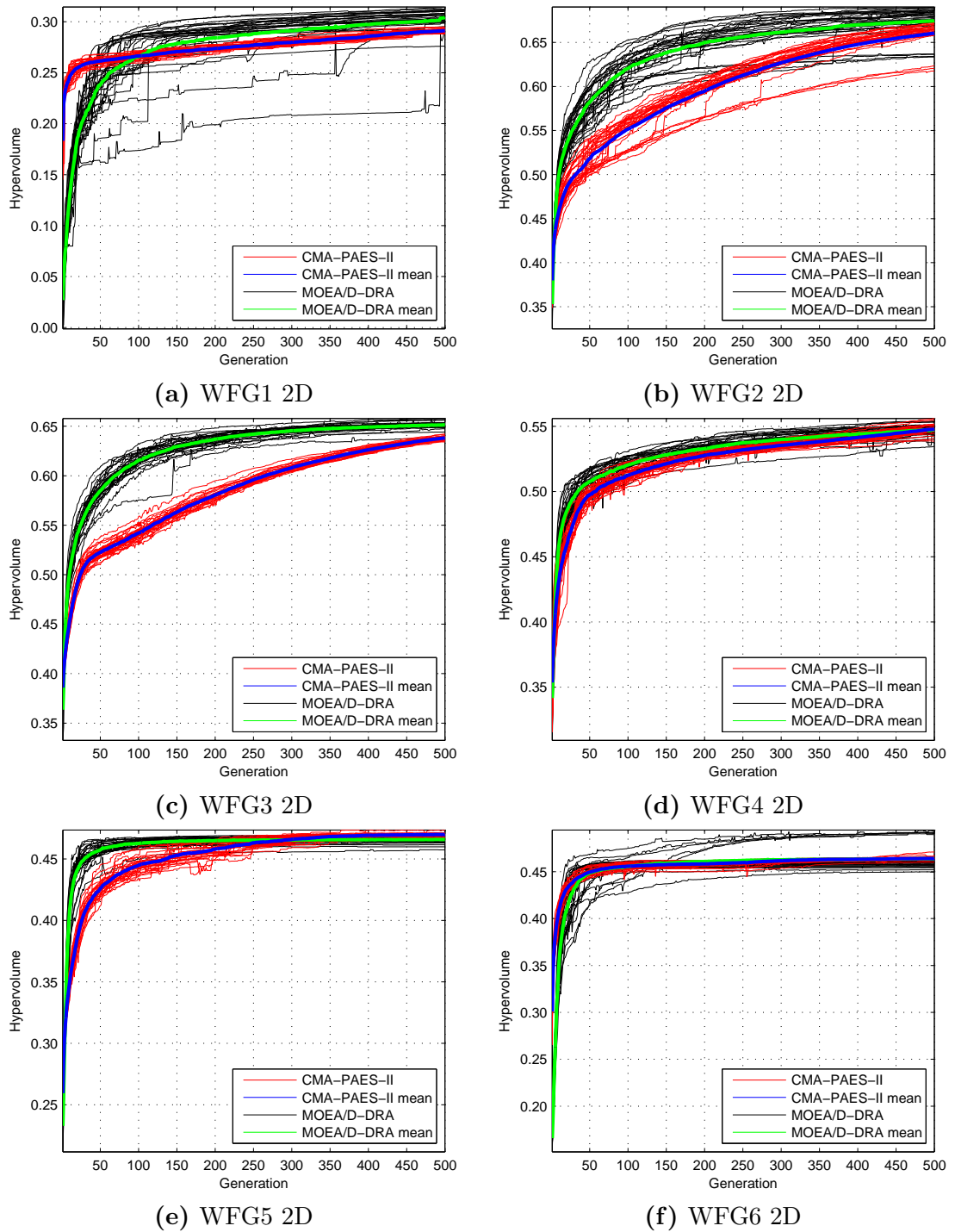


Figure 4.10: Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered two-objective WFG test problems.

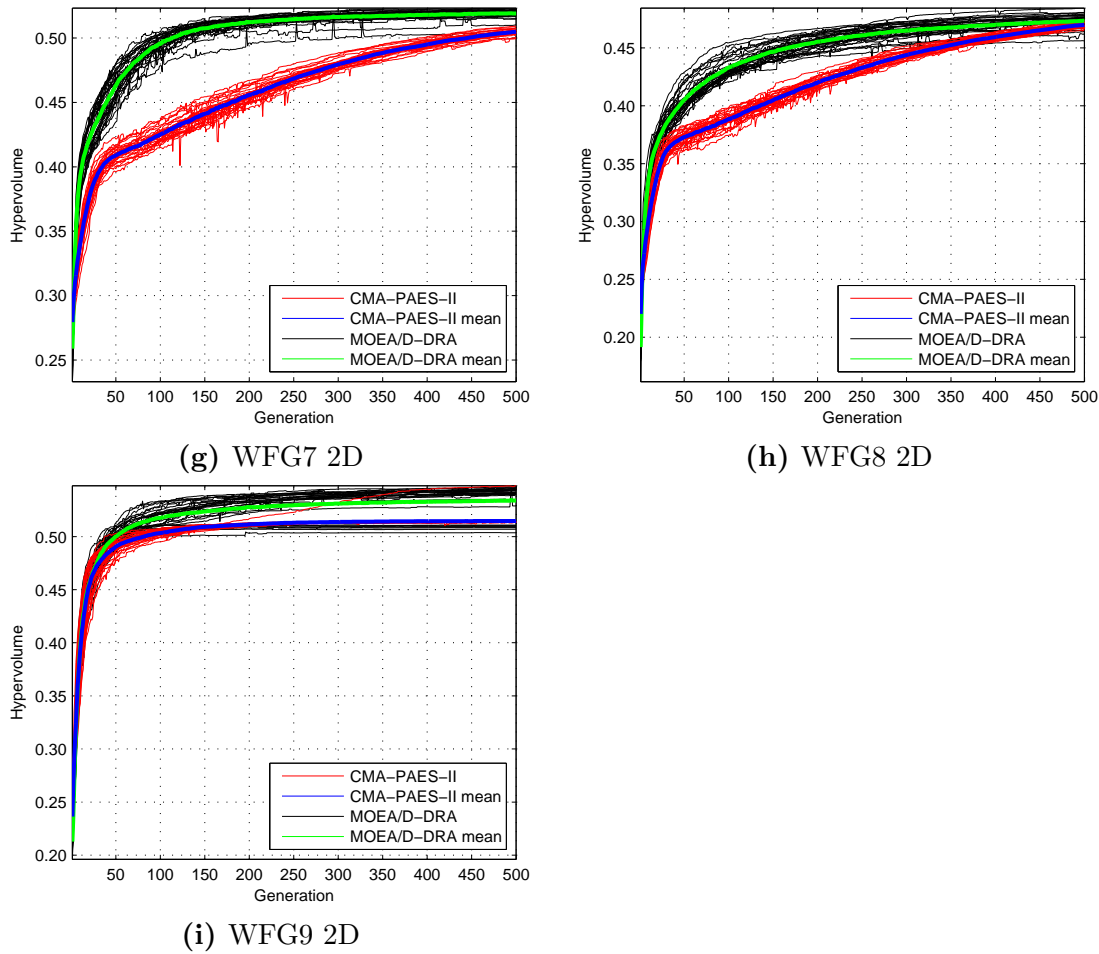


Figure 4.10: Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered two-objective WFG test problems.

4.7.2 Three-Objective Results

Overall, it can be observed in Table 4.7 that CMA-PAES-II outperforms MOEA/D-DRA on the considered WFG test functions when each test function is configured to consist of three objectives. CMA-PAES-II achieves better mean hypervolume indicator performance on seven of the nine considered test functions, it also achieves the best performing final approximation set of two algorithms on six of the nine considered test functions. Furthermore, it achieves the best worst-case

performance on all of the considered test functions.

Table 4.7: Hypervolume indicator results from 30 executions of CMA-PAES-II and MOEA/D-DRA on the WFG test suite with three objectives.

3D	CMA-PAES-II			MOEA/D-DRA			p -value	
	Worst	Mean	Best	Worst	Mean	Best		
WFG1	0.43017	0.43299	0.43517	0.36038	0.41075	0.42752	3.0199e-11	+
WFG2	0.86768	0.8816	0.89345	0.78028	0.89286	0.91835	9.5139e-06	-
WFG3	0.65903	0.67208	0.68303	0.62395	0.65032	0.67108	2.3715e-10	+
WFG4	0.65217	0.65708	0.66489	0.61332	0.62976	0.6491	3.0199e-11	+
WFG5	0.58381	0.59754	0.60997	0.57764	0.58925	0.60018	4.9426e-05	+
WFG6	0.64112	0.64534	0.64974	0.60524	0.61768	0.63362	3.0199e-11	+
WFG7	0.63625	0.65062	0.66336	0.63256	0.6534	0.66692	0.13345	=
WFG8	0.57531	0.58475	0.59543	0.51948	0.55792	0.58836	2.9215e-09	+
WFG9	0.62113	0.62475	0.65209	0.59063	0.61133	0.63881	0.00030059	+

On WFG1, CMA-PAES-II outperforms MOEA/D-DRA in regards to mean, best, and worst hypervolume indicator performance of the final approximation sets. It can be seen in the plot of the hypervolume indicator performance at each generation in Figure 4.11a that CMA-PAES-II achieves faster initial convergence similar to Figure 4.10a when it was executed on WFG1 with two objectives. There is also less dispersion in the individual executions which can be verified in the box-plot in Figure 4.9b.

On WFG2, MOEA/D-DRA again outperforms CMA-PAES-II in regards to the mean and best hypervolume indicator performance of the final approximation sets. MOEA/D-DRA also achieves a better convergence rate which can be seen in Figure 4.11b, however in comparison to the results from WFG2 with two objectives, CMA-PAES-II reaches a similar rate of convergence and performs

better. MOEA/D-DRA contains a number of low performing outliers as seen in Figure 4.9b.

On WFG3, CMA-PAES-II outperforms MOEA/D-DRA in regards to mean, best, and worst hypervolume indicator performance of the final approximation sets. Both algorithms reach similar rates of convergence and dispersion which can be observed in Figure 4.11d, and Figure 4.9b suggests CMA-PAES-II produces more robust executions as there is less dispersion in the final approximation sets.

On WFG4, CMA-PAES-II outperforms MOEA/D-DRA in regards to the mean, worst, and best hypervolume indicator performance of the final approximation sets. Both algorithms reach similar rates of convergence and CMA-PAES-II produces more robust executions as there is less dispersion in the final approximation sets which can be observed in Figures 4.11d and 4.9b.

On WFG5, CMA-PAES-II outperforms MOEA/D-DRA in regards to the mean, worst, and best hypervolume indicator performance of the final approximation sets. However, similar to when optimising WFG5 with two objectives MOEA/D-DRA reaches a faster rate of mean convergence up until approximately 350 generations, this can be seen in Figure 4.11e.

On WFG6, CMA-PAES-II outperforms MOEA/D-DRA in regards to mean, worst, and best hypervolume indicator performance on the final approximation sets. Both algorithms reach similar rates of convergence and dispersion which can be seen in Figure 4.11f. CMA-PAES-II produces more robust executions as there is less dispersion in the final approximation sets which can be seen in Figure 4.9a.

On WFG7, MOEA/D-DRA outperforms CMA-PAES-II in regards to the mean and best hypervolume indicator performance of the final approximation

sets, however, the improvement in performance is not significant. Both algorithms reach similar rates of convergence which can be seen in Figure 4.11g.

On WFG8, CMA-PAES-II outperforms MOEA/D-DRA in regards to mean, worst, and best hypervolume indicator performance of the final approximation sets. Both algorithms reach similar rates of convergence which can be seen in Figure 4.10h. By studying Figure 4.9a, it can be observed that CMA-PAES-II offers more robust performance with a much smaller total range, and an interquartile range which achieves better hypervolume indicator performance than the entire interquartile range for MOEA/D-DRA.

On WFG9, CMA-PAES-II outperforms MOEA/D-DRA in regards to mean, worst and best hypervolume indicator performance of the final approximation sets. Both algorithms reach similar rates of convergence which can be observed in Figure 4.11i, and CMA-PAES-II produces the execution with the best performance, though it is considered an outlier which is shown in Figure 4.9b.

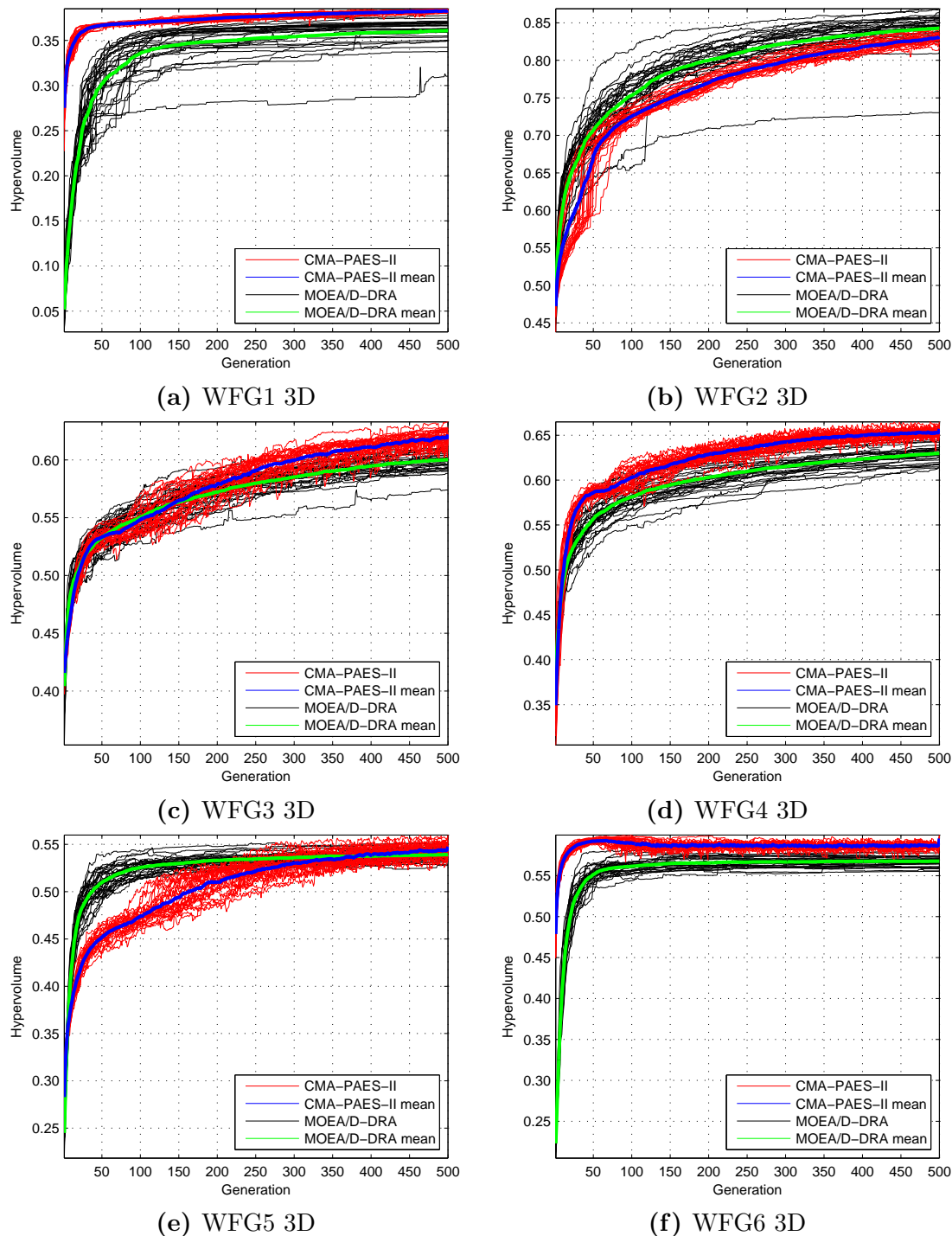


Figure 4.11: Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered three-objective WFG test problems.

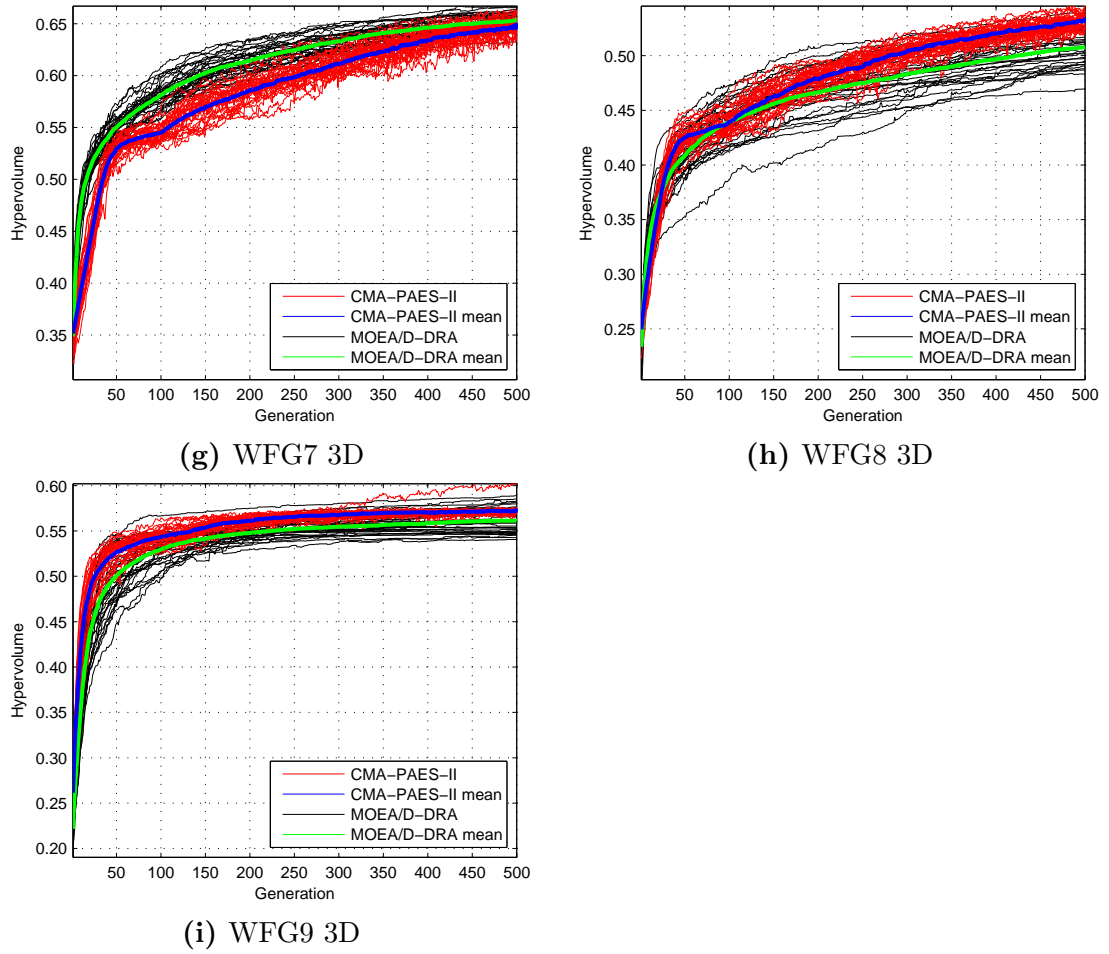


Figure 4.11: Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered three-objective WFG test problems.

4.7.3 Five-Objective Results

Overall, it can be observed in Table 4.8 that CMA-PAES-II outperforms MOEA/D-DRA on the considered WFG test functions when each test function is configured to consist of five objectives. CMA-PAES-II achieves better mean hypervolume indicator performance on seven of the nine considered test functions. Furthermore, CMA-PAES-II produces the best performing final approximation set of the two algorithms on six of the nine considered test functions, and also achieves the best worst-case performance on all of the considered test functions.

Table 4.8: Hypervolume indicator results from 30 executions of CMA-PAES-II and MOEA/D-DRA on the WFG test suite with five objectives.

5D	CMA-PAES-II			MOEA/D-DRA			p -value	
	Worst	Mean	Best	Worst	Mean	Best		
WFG1	0.41395	0.41629	0.41938	0.37947	0.39065	0.3973	3.0199e-11	+
WFG2	0.85062	0.89	0.90419	0.76018	0.89201	0.93239	0.0013017	-
WFG3	0.59406	0.61944	0.6398	0.54642	0.5843	0.61288	2.4386e-09	+
WFG4	0.69843	0.71753	0.74149	0.61096	0.65344	0.70566	4.0772e-11	+
WFG5	0.52795	0.5638	0.58662	0.57799	0.60514	0.65011	6.0658e-11	-
WFG6	0.7449	0.75287	0.76201	0.67458	0.70012	0.72186	3.0199e-11	+
WFG7	0.61305	0.64306	0.6756	0.59533	0.64872	0.70915	0.71719	=
WFG8	0.58967	0.6293	0.65547	0.57004	0.60989	0.66974	0.00020058	+
WFG9	0.63315	0.65487	0.67674	0.5359	0.59016	0.63396	3.3384e-11	+

The rates of convergence for CMA-PAES-II and MOEA/D-DRA on WFG tool-kit test functions configured for five objectives appear to be similar to experiments conducted on the same test functions configured for three objectives in Section 4.7.2, these have been presented in Figure 4.12. The most noticeable difference in hypervolume indicator performance from generation to generation is in the scale of the oscillation from CMA-PAES-II, this is best illustrated in

Figure 4.12c in comparison to Figure 4.11c, where the difference in oscillation on the WFG3 test function increases when moving from three to five objectives.

This oscillation in hypervolume indicator performance from generation to generation is expected to increase in CMA-PAES-II as the number of objectives in the test function increase, this is due to the execution of IBC when trying to maximise hypervolume indicator performance with a small population in many-objective space. It is also expected that where CMA-PAES-II hypervolume indicator performance will oscillate on a greater scale on a higher number of objectives, the hypervolume indicator performance of MOEA/D-DRA in comparison will worsen, increasing the difference in mean performance between the two algorithms.

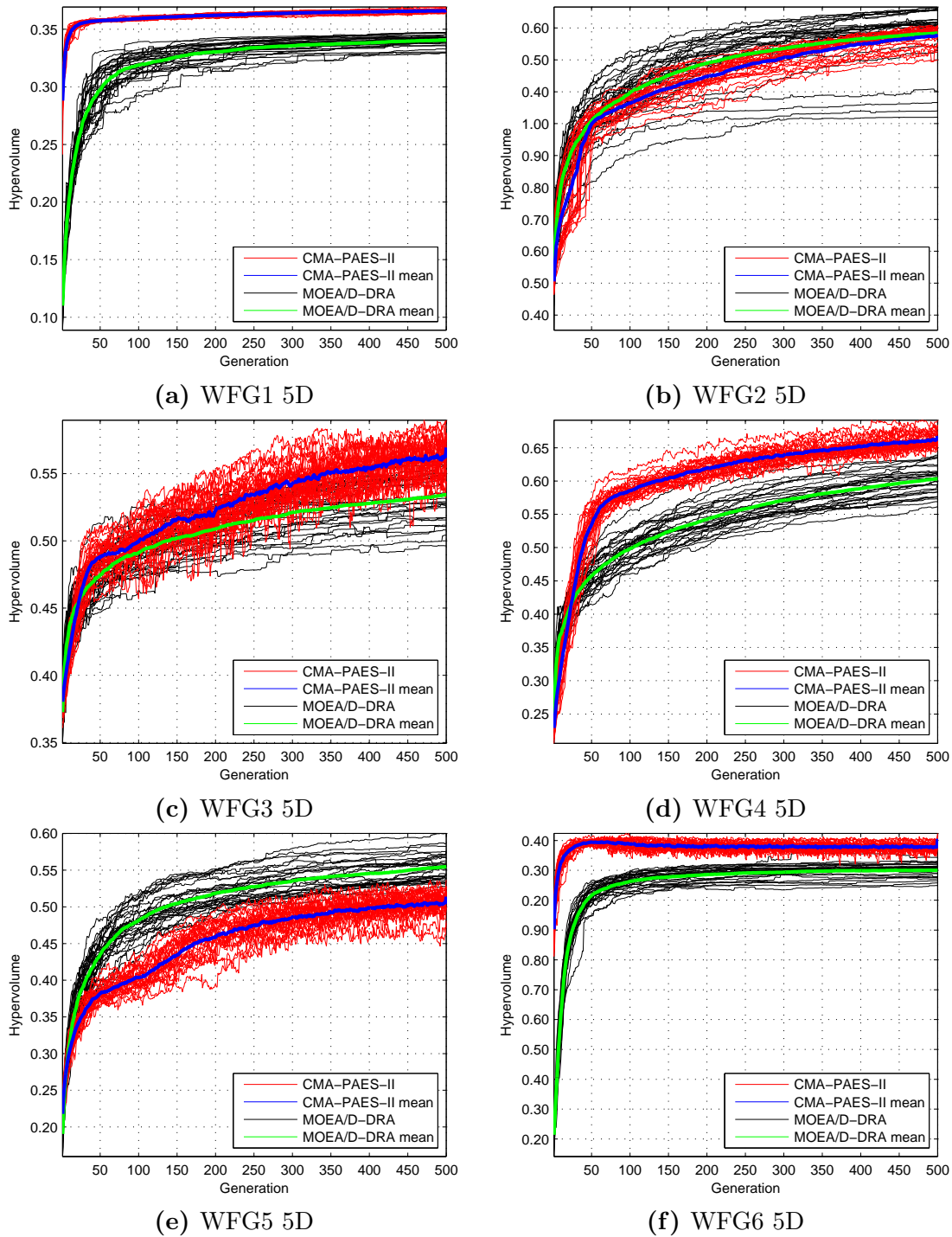


Figure 4.12: Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered five-objective WFG test problems.

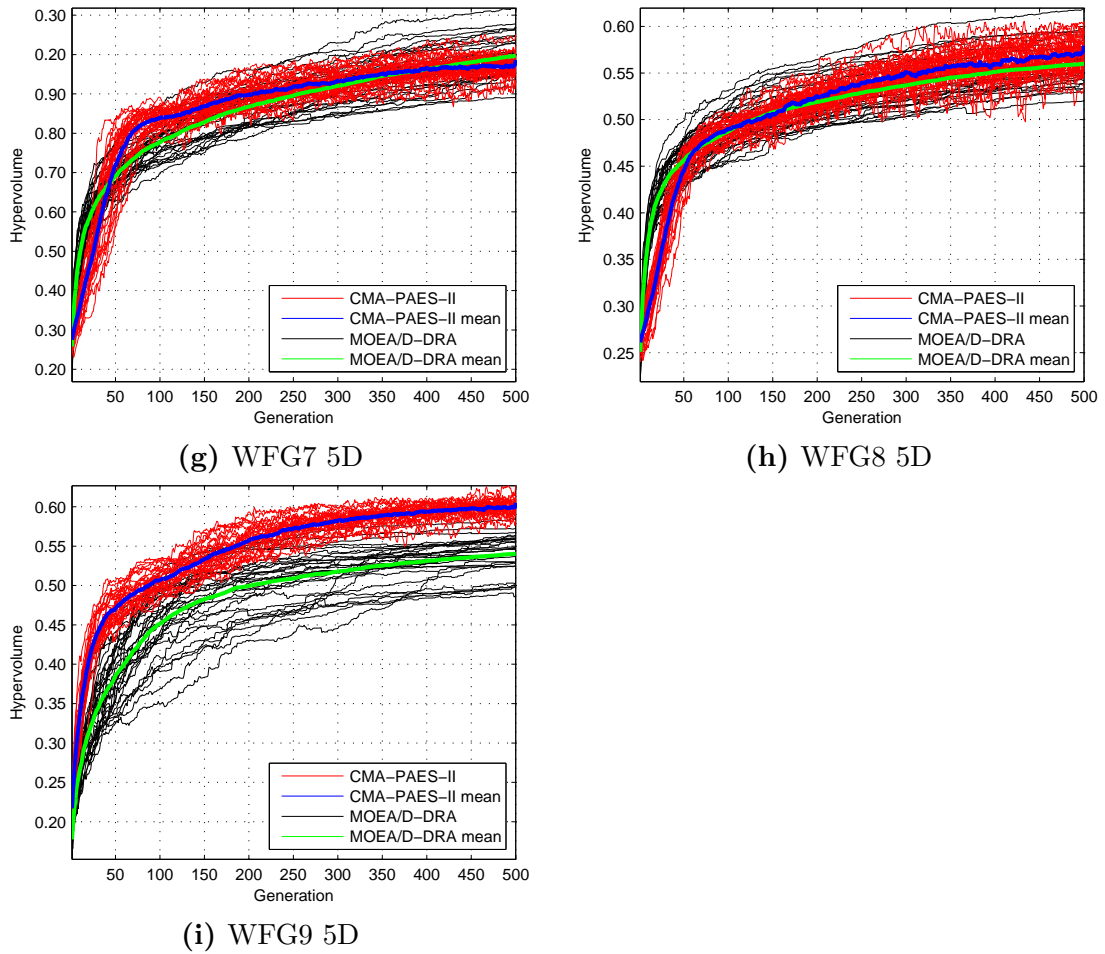


Figure 4.12: Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered five-objective WFG test problems.

4.7.4 Seven-Objective Results

Overall, it can be observed in Table 4.9 that CMA-PAES-II outperforms MOEA/D-DRA on the considered WFG test functions when each test function is configured to consist of seven objectives. CMA-PAES-II achieves better mean hypervolume indicator performance on eight of the nine considered test functions. Furthermore, CMA-PAES-II produces the best performing final approximation set of two algorithms on five of the nine considered test functions, and also achieves the

best worst-case performance on eight of the nine considered test functions.

Table 4.9: Hypervolume results from 30 executions of CMA-PAES-II and MOEA/D-DRA on the WFG test suite with seven objectives.

7D	CMA-PAES-II			MOEA/D-DRA			p -value	
	Worst	Mean	Best	Worst	Mean	Best		
WFG1	0.34726	0.35097	0.35525	0.31918	0.32613	0.33485	3.0199e-11	+
WFG2	0.85555	0.89994	0.92708	0.78171	0.9285	0.95981	2.0152e-08	-
WFG3	0.58351	0.60479	0.61544	0.49175	0.55128	0.59361	4.0772e-11	+
WFG4	0.68342	0.70559	0.72476	0.58811	0.65099	0.70675	6.1177e-10	+
WFG5	0.47554	0.52808	0.56049	0.47747	0.52535	0.5823	0.37108	=
WFG6	0.84837	0.85457	0.85919	0.82161	0.83261	0.84717	3.0199e-11	+
WFG7	0.63425	0.68295	0.72226	0.59571	0.66344	0.73922	0.012731	+
WFG8	0.62454	0.67712	0.71885	0.60134	0.66714	0.73353	0.50114	=
WFG9	0.57861	0.62316	0.65317	0.43025	0.52282	0.63121	1.9568e-10	+

The rates of convergence for CMA-PAES-II and MOEA/D-DRA on WFG tool-kit test functions configured for seven objectives appear to be similar to experiments conducted on the same test functions configured for three and five objectives in Sections 4.7.2 and 4.7.3, these have been presented in Figure 4.13. As expected and predicted in Section 4.7.3 in the comparison of convergence rate between three and five objectives, the oscillation in hypervolume performance has increased in scale, and the difference in performance between CMA-PAES-II and MOEA/D-DRA has increased. This pattern is expected to continue as the number of objectives increase in the next experiment.

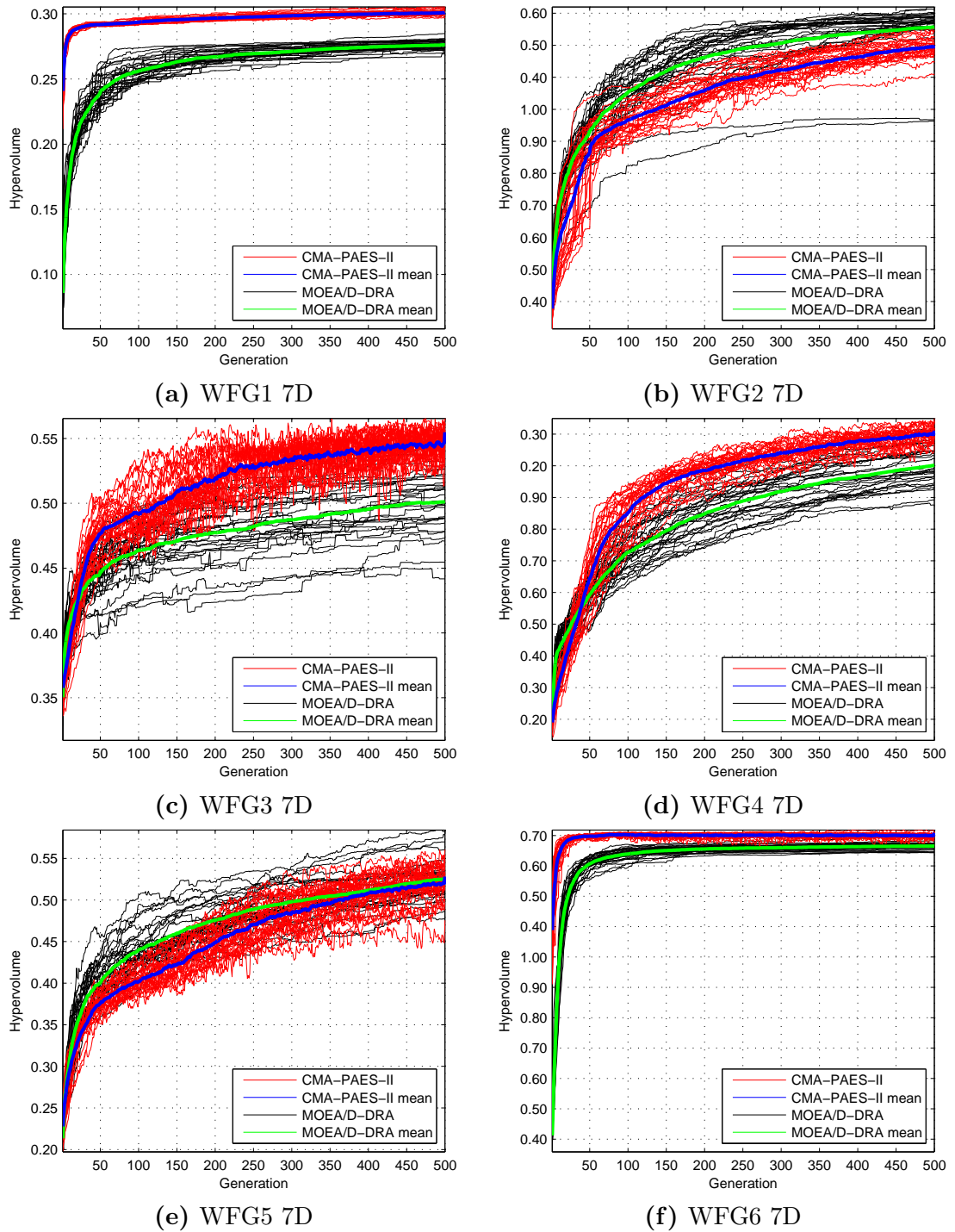


Figure 4.13: Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered seven-objective WFG test problems.

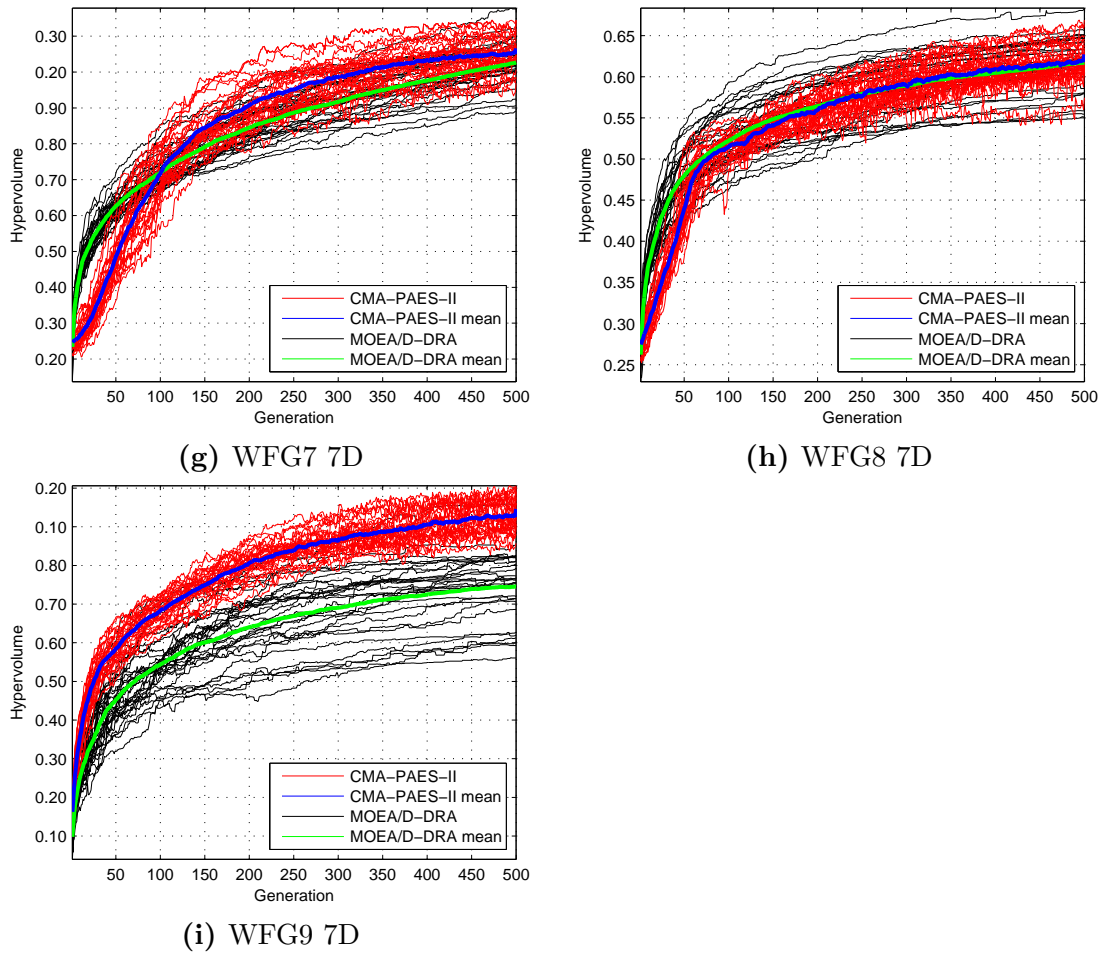


Figure 4.13: Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered seven-objective WFG test problems.

4.7.5 Ten-Objective Results

Overall, it can be observed in Table 4.10 that CMA-PAES-II outperforms MOEA/D-DRA on the considered WFG test functions when each test function is configured to consist of ten objectives. CMA-PAES-II achieves better mean hypervolume indicator performance on eight of the nine considered test functions. Furthermore, CMA-PAES-II achieves the best performing final approximation set of two algorithms on seven of the nine considered test functions, and also achieves the

best worst-case performance on eight of the nine considered test functions.

Table 4.10: Hypervolume results from 30 executions of CMA-PAES-II and MOEA/D-DRA on the WFG test suite with ten objectives.

10D	CMA-PAES-II			MOEA/D-DRA			p -value	
	Worst	Mean	Best	Worst	Mean	Best		
WFG1	0.29982	0.30449	0.31119	0.27127	0.27848	0.29103	3.0199e-11	+
WFG2	0.9397	0.95548	0.96561	0.95163	0.97197	0.97866	5.0723e-10	-
WFG3	0.65453	0.66091	0.66733	0.53156	0.58962	0.62602	3.0123e-11	+
WFG4	0.6744	0.70753	0.76728	0.61893	0.69902	0.765	0.38709	=
WFG5	0.46585	0.49276	0.51274	0.40642	0.45224	0.50403	1.8567e-09	+
WFG6	0.63142	0.64301	0.6679	0.5458	0.56942	0.59811	3.0199e-11	+
WFG7	0.67276	0.72503	0.80515	0.58412	0.66273	0.77383	2.3768e-07	+
WFG8	0.57423	0.59885	0.63409	0.46983	0.5827	0.68128	0.31119	=
WFG9	0.54441	0.60458	0.63477	0.36732	0.44146	0.55746	3.6897e-11	+

The rates of convergence for CMA-PAES-II and MOEA/D-DRA on WFG tool-kit test functions configured for seven objectives appear to be similar to experiments conducted on the same test functions configured for three and seven objectives in Section 4.7.4. As predicted in Section 4.7.4 in the comparison of convergence rate between three, five and seven objectives, the oscillation in hypervolume performance has increased in scale, and the difference in performance between CMA-PAES-II and MOEA/D-DRA has increased. In addition to this performance increase, CMA-PAES-II also outperforms MOEA/D-DRA on more of the worst-case and best-case executions than in previous experiments when test functions were configured to a lower number of problem objectives.

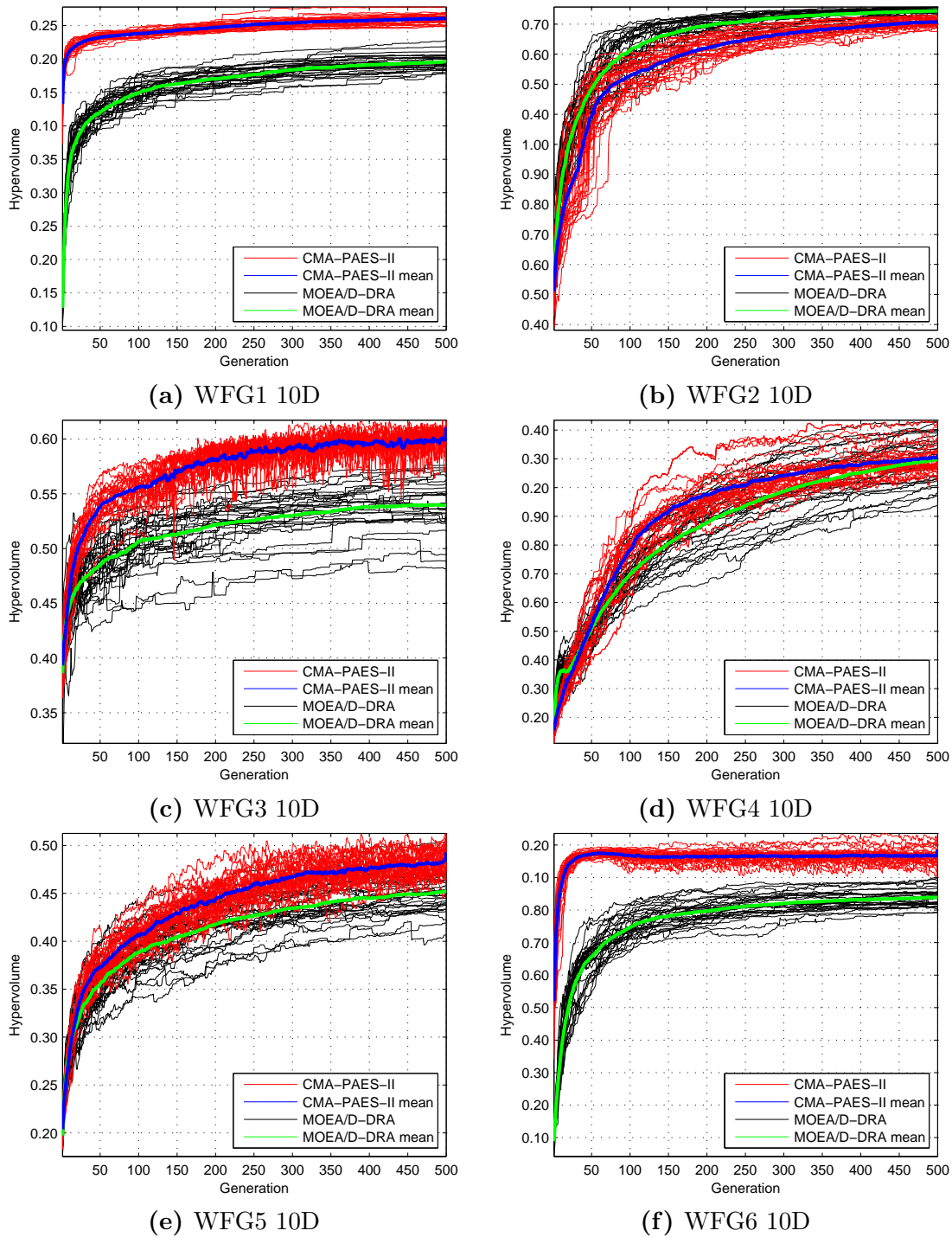


Figure 4.14: Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered 10D WFG test problems.

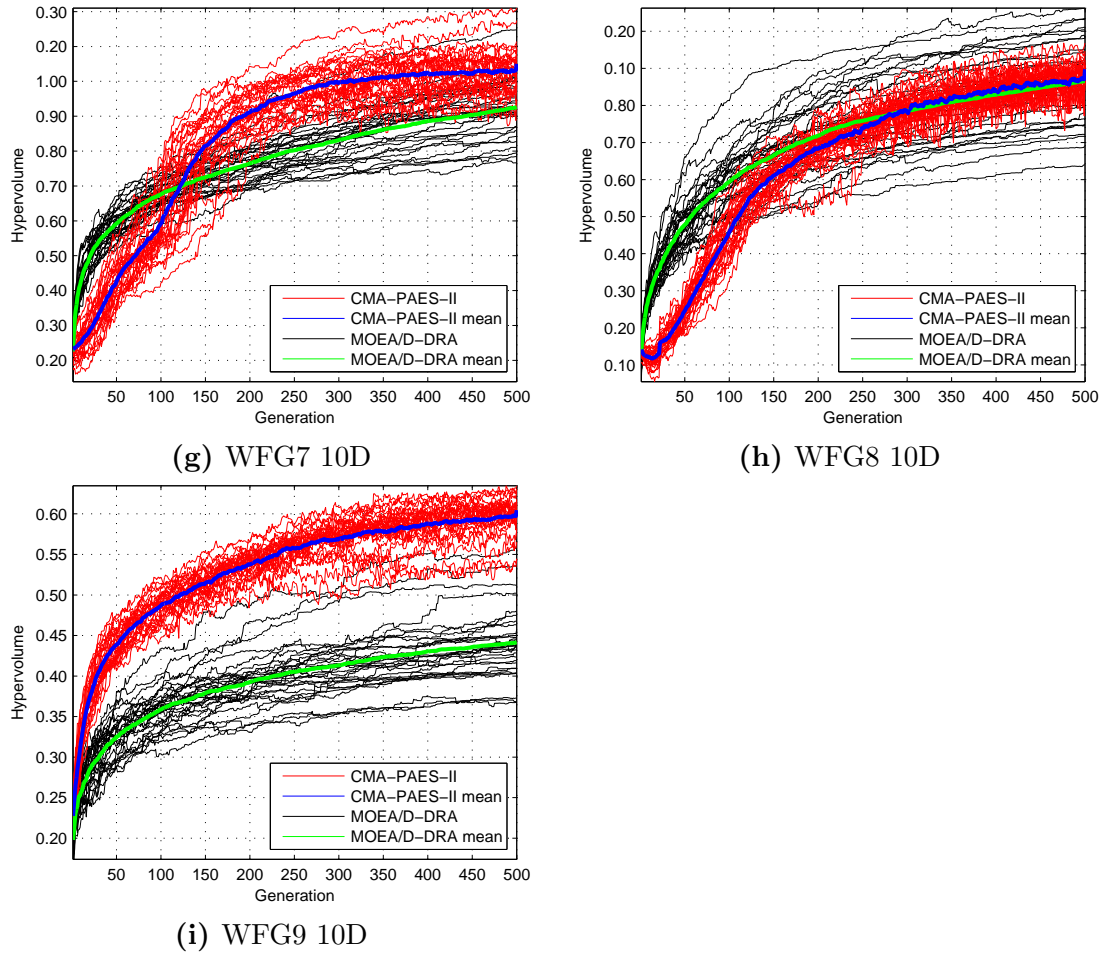


Figure 4.14: Hypervolume indicator values at each generation for CMA-PAES-II and MOEA/D-DRA on the considered 10D WFG test problems.

4.8 Conclusion

Benchmarking and performance analysis of the algorithm returned promising results suggesting that on problems containing many objectives CMA-PAES-II is faster at converging to an approximation set with better hypervolume indicator quality than MOEA/D-DRA. MOEA/D-DRA outperformed CMA-PAES-II on experiments consisting of test functions with two objectives, however, on test

functions consisting of three objectives and more, CMA-PAES-II outperforms MOEA/D-DRA consistently with an increasing difference in the performance gap as the number of objectives scale up.

The IBC mechanism for assisting the convergence of CMA-PAES-II based on continuous assessment using a desired performance indicator, has been successful in ensuring entire executions of the algorithm are not wasted on a local optima or only part of a Pareto-optimal front, this has been illustrated in the plots of hypervolume indicator results over generations in each benchmarking section.

Overall, CMA-PAES-II has been designed and benchmarked as a robust and extensible EMO algorithm for many-objective problems, which has a computational cost that does not restrict it to computing clusters or below four-objective test functions.

Chapter 5

Weighted Z-Score Preference Articulation

One approach to solving complex engineering problems is to use Evolutionary Multi-Objective Optimisation (EMO) algorithms to address each of the conflicting objectives simultaneously. Typically, these EMO algorithms are run non-interactively with a Decision Maker (DM) setting the initial parameters of the algorithm and then analysing the results at the end of the execution process (which can often take hours or days to complete). This approach has been common since the late 1990s [2, 4, 5] and will lead to a set of potential solutions distributed across the whole trade-off surface. Whilst this is often appropriate for problems with a low number of objectives, in real-world problems that involve the consideration of many objectives this trade-off surface can be very large. In these cases, the DM is usually more interested in a sub-region of this solution space that satisfies some domain specific criteria. However, this can be complicated further by a lack of *a priori* knowledge about what trade-offs are achievable. To overcome these problems, Progressive Preference Articulation (PPA) methods have been proposed that take into account DM preferences (such as [156])

but these are frequently difficult to integrate with current state of the art EMO algorithms, and the incorporation of user preferences is frequently disregarded in the EMO literature [157].

This chapter introduces a novel method of progressive preference articulation in Section 5.1 for EMO algorithms which can provide improved performance in both the execution speed of the algorithm and in the quality of the solutions the algorithm produces. This method is then integrated into two state-of-the-art EMO algorithms in Sections 5.2 and 5.3. Section 5.4 provides a full statistical analysis of the results of the integration of the proposed novel progressive preference articulation operator with state of the art EMO algorithms for two suites of benchmark test functions from the literature. The results from the test-cases are then discussed in Section 5.5 and the chapter is then concluded in Section 5.6.

5.1 Weighted Z-Score Preference Articulation Operator

Weighted Z-score (WZ) preference articulation is a novel method of preference articulation based around the use of z-scores¹ (or standard scores) from statistics. Traditional z-score calculations are performed by subtracting the population mean from a datum and then dividing the result by the population standard deviation as can be seen in Equation 5.1. Calculating the z-score in statistics requires knowing the population parameters and not just the parameters of a sample, which is often seen as unrealistic in typical statistics; however this is not an issue in EMO as it is possible to have a complete representation of the population at each generation.

$$z = \frac{(x - \mu)}{\sigma} \quad (5.1)$$

For the z-score to be useful for preference articulation, some modifications are made to the way z is calculated. Instead of using the population mean and population standard deviation to calculate z , the preference information² that has been expressed by the DM is used (as can be seen in Equation 5.2) where ρ_m is the goal for a corresponding objective value x_{mn} , and N is the number of solutions in the population.

$$z_{mn} = \frac{(x_{mn} - \rho_m)}{\sqrt{\frac{\sum_{n=1}^N (x_{mn} - \rho_m)^2}{N}}} \quad (5.2)$$

¹The number of standard deviations a datum is above or below the mean of its data-set.

²A ROI can be defined by a preference vector containing goals for each objective, where if all objective values of a solution satisfy the corresponding objective goal it is considered within the ROI.

This will enable the calculation of z_{mn} for the objective values of each candidate solution in an approximation set, resolving the number of standard deviations each solution is from the DM's expressed Region of Interest (ROI), which will be a positive value when it is outside the ROI, and negative when within the ROI. Once z_{mn} is calculated for every objective value of a solution, the z_{mn} values are aggregated into a single fitness value using Equation 5.3.

$$V_n = \frac{\sum_{m=1}^M z_{mn}}{M} \quad (5.3)$$

As a demonstration, the CEC09 competition winning MOEA/D-DRA [93] EMO is executed for five generations to generate an initial population for the ZDT1 synthetic test problem from the ZDT test suite. Using Equation 5.3 to calculate V_n for each candidate solution, it is possible to order the initial population by the aggregated number of standard deviations from the ROI (0.2 for objective 1 and 1 for objective 2 for this example) and then select a number of solutions (five for this example) to exploit for the next generation. An illustration of this example can be seen in Figure 5.1.

The Z-scores calculated using the simple method in Equation 5.2 can be used in an EMO as either a replacement or addition to the fitness scheme used for selection, to focus the search towards and then within the ROI expressed by the DMs. This works well for test problems where there is a low number of objectives and the Pareto-optimal set is not complicated in shape as seen in Figure 5.2, however its effectiveness is reduced when this basic method is applied to a more complicated problem with a higher number of objectives. To demonstrate this, the same basic Z-score preference articulation method is applied to an initial

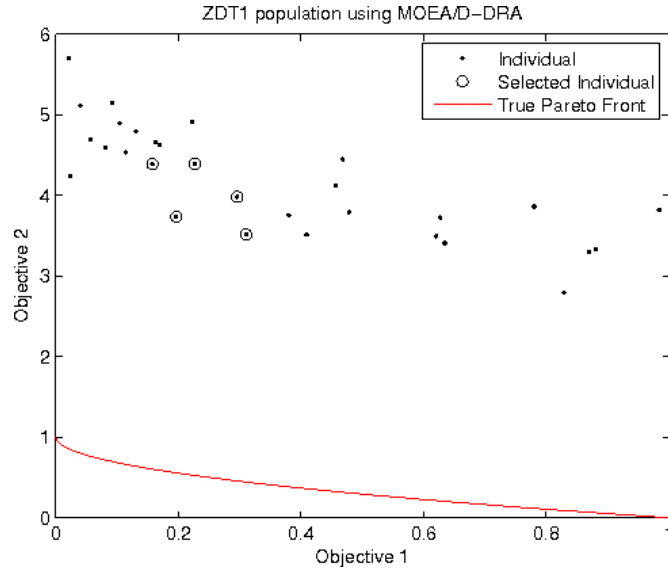


Figure 5.1: Basic Z-score preference articulation applied to an initial population generated by MOEA/D-DRA for the synthetic test problem ZDT1.

population generated by executing MOEA/D-DRA for five generations on a five-objective instance of the WFG5 synthetic test problem from the WFG tool-kit, this is illustrated in Figure 5.3.

This demonstration illustrates that the basic method of Z-score preference articulation falls into the trap of selecting the solutions with the overall lowest z_{mn} values. This results in the minimisation of objectives 1 to 4 as they are already below or close in proximity to the corresponding preference value ρ_m , yielding negative or low positive z values, and due to the conflicting objectives in WFG5, the associated 5th objective value for these selected solutions are the furthest away from the specified ROI. This issue is amplified from generation to generation and the result after 2000 function evaluations with the ROI specified as $\{2, 5, 2, 2, 2\}$ has been illustrated in Figure 5.4, where it can be observed that although objectives 1-4 have been further optimised, there are still no individuals

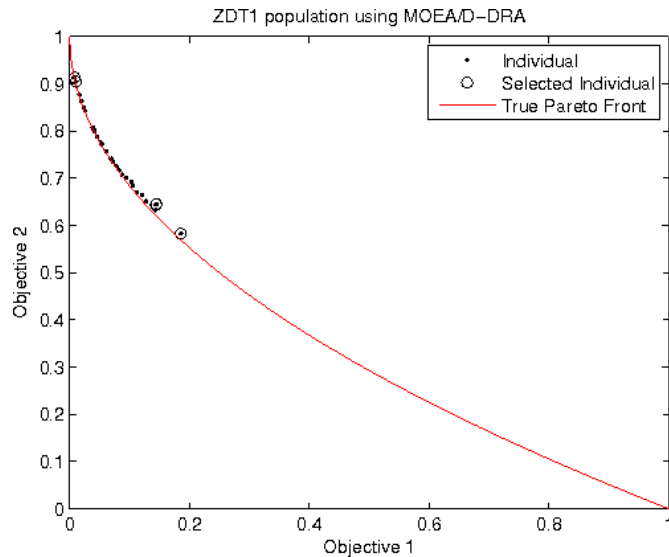


Figure 5.2: Population generated by MOEA/D-DRA combined with basic Z-score preference articulation after 2000 function evaluations for the synthetic test problem ZDT1.

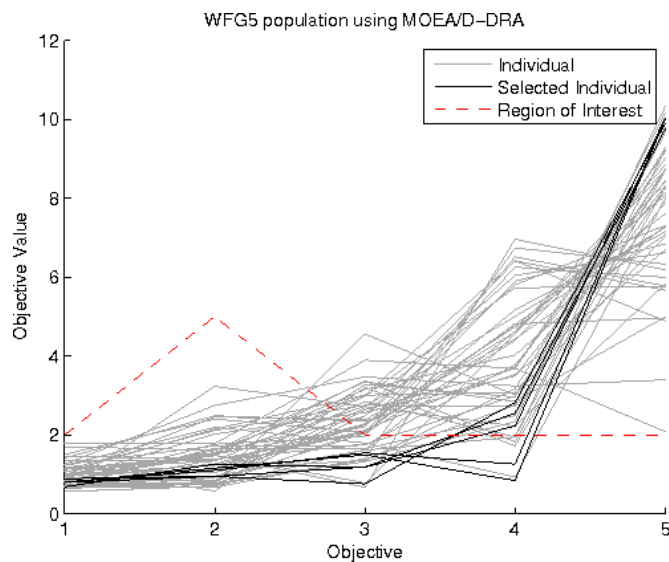


Figure 5.3: Basic Z-score preference articulation applied to an initial population generated by MOEA/D-DRA for the synthetic test problem WFG5.

within the desired ROI due to objective 5.

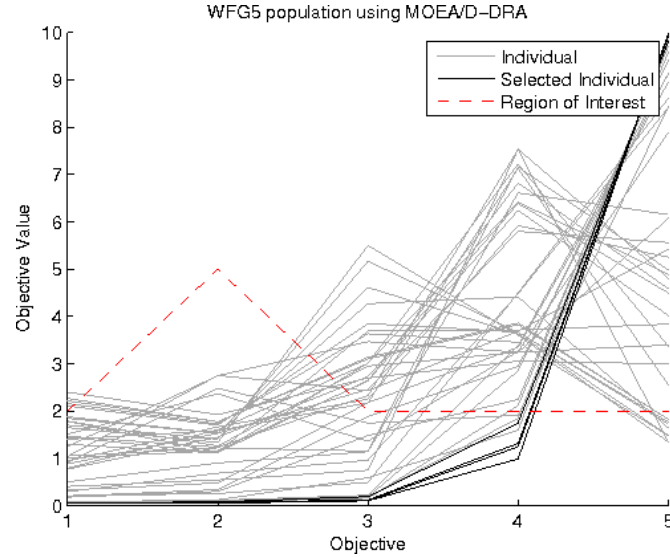


Figure 5.4: Population generated by MOEA/D-DRA combined with basic Z-score preference articulation after 2000 function evaluations for the synthetic test problem WFG5.

To solve this ineffectiveness at higher numbers of objectives and on more difficult problems, a two-phase preference articulation operator has been developed. The first phase (W-phase) focusses the search on bringing all objectives closer in proximity to the desired ROI using absolute and weighted z values. The second phase (Z-phase) takes effect once the criterion for the number of solutions required within the ROI has been met, this phase uses the basic z values demonstrated earlier for minimisation within the ROI.

The mathematical procedure for the WZ preference articulation operator in its entirety is described herein. M defines the number of problem objectives whilst N defines the population size. X is an M by N matrix of entries x_{mn} , where every x_{mn} refers to a solution's objective value:

$$X_n = \langle x_{1n}, x_{2n}, \dots, x_{Mn} \rangle \quad (5.4)$$

Z is an M by N matrix of entries z_{mn} , where every z_{mn} refers to the result of the z-score preference articulation operator applied to a corresponding objective value x_{mn} :

$$Z_n = \langle z_{1n}, z_{2n}, \dots, z_{Mn} \rangle \quad (5.5)$$

To calculate Z , a preference vector P of M entries must be defined, where every entry ρ_m refers to the goal which the corresponding objective values x_m must satisfy:

$$P = \langle \rho_1, \rho_2, \dots, \rho_M \rangle \quad (5.6)$$

S is an M by N matrix of entries s_{mn} where every s_{mn} refers to a logical value indicating whether the corresponding objective value x_{mn} has satisfied the corresponding goal ρ_{mn} ($x_{mn} \leq \rho_m$):

$$S_n = \langle s_{1n}, s_{2n}, \dots, s_{Mn} \rangle \quad (5.7)$$

where s_{mn} is calculated using:

$$s_{mn} = \begin{cases} 1, & \text{if } x_{mn} \leq \rho_m \\ 0, & \text{otherwise.} \end{cases} \quad (5.8)$$

Φ is a vector of N entries, where every ϕ_n refers to a logical value indicating whether all entries of P have been satisfied by a solution X_n .

$$\Phi = \langle \phi_1, \phi_2, \dots, \phi_N \rangle \quad (5.9)$$

where ϕ_n is calculated by the product of the entries of S_n :

$$\phi_n = \prod_{m=1}^M s_{mn} \quad (5.10)$$

The scalar Ψ refers to the number of solutions X_n in the population which have satisfied the preference vector P :

$$\Psi = \sum_{n=1}^N \phi_n \quad (5.11)$$

T defines the required number of solutions which satisfy the preference vector before the search changes phase. Whilst $\Psi < T$ the W-phase of the WZ preference articulation operator takes effect. In this phase, the weighting $(1 - \frac{1}{M})$ is only applied to the z_{mn} value if m corresponds to the entry of Ω with the lowest value. ω_m refers to the number of solutions in the population that have satisfied the corresponding ρ_m :

$$\Omega = \langle \omega_1, \omega_2, \dots, \omega_M \rangle \quad (5.12)$$

ω_m is the sum of columns M in the matrix S and is calculated using:

$$\omega_m = \sum_{n=1}^N s_{mn} \quad (5.13)$$

With the entries of Ω calculated, the M by N matrix of weighted scores E can be defined as:

$$E_n = \langle \epsilon_{1n}, \epsilon_{2n}, \dots, \epsilon_{mN} \rangle \quad (5.14)$$

where the corresponding weighted score ϵ_{mn} for each objective value x_{mn} can be calculated using:

$$\epsilon_{mn} = \begin{cases} z_{mn} \left(1 - \frac{1}{M}\right) & \text{if } f(\omega_m, S_{mn}) = 0 \\ z_{mn} & \text{otherwise.} \end{cases} \quad (5.15)$$

where z_{mn} and ω_m are first normalised to real values between 0 and 1:

$$z_{mn} = f(|z_{mn}|, |Z_m|) \quad (5.16)$$

using the function $f(k, K)$ where:

$$f(k, K) = \frac{k - \min(K)}{\max(K - \min(K))} \quad (5.17)$$

The initial calculation of z_{mn} is the same in both phases (W-phase and Z-phase) and is defined in Equation 5.2. The final score W_n of a single solution is the aggregation of the corresponding ϵ_{mn} entries:

$$W_n = \frac{\sum_{m=1}^M \epsilon_{mn}}{M} \quad (5.18)$$

This two-phase method attempts to move the search towards the production of solutions that are close in proximity to the ROI and within it, but does not attempt to minimise the solutions beyond the edges of the ROI. When the number of solutions within the ROI has satisfied the threshold ($\Psi \geq T$) the Z-phase takes effect. This phase uses Equation 5.2 to calculate Z_n and then Equation 5.3 to aggregate the scores into the scalar V_n , this is because there are adequate solutions (defined by T) that have satisfied all entries of P . These solutions can then be further minimised within the ROI. A full example of both phases of the WZ preference articulation operator is available in Section 5.1.1.

When the WZ preference articulation operator algorithm is applied to the same initial population previously generated for WFG5 by MOEA/D-DRA, solutions that are closest to the ROI are selected with weighted preference for solutions with objectives which have not yet been satisfied. This has been illustrated in Figure 5.5, where it can be seen that solutions with worse values for objective 1-4 have been selected in order to exploit their useful genetic information to bring objective 5 closer to and within the ROI.

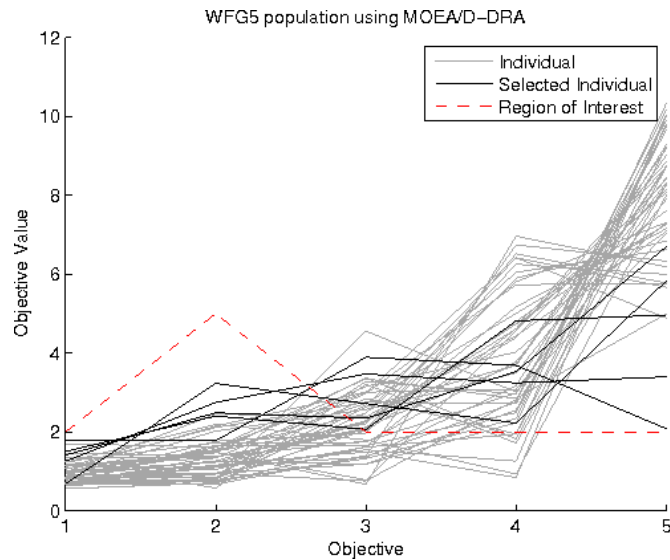


Figure 5.5: WZ preference articulation applied to an initial population generated by MOEA/D-DRA for the synthetic test problem WFG5.

The second phase of the search is activated when the threshold T has been met or exceeded, for this example T is set to 5 solutions (one tenth of the population size) and executed for the same 2000 function evaluations as before, the results of the search have been illustrated in Figure 5.6.

The results from this experiment show that 25 solutions have been found within the desired ROI, further investigation has shown that threshold T of 5 solutions was satisfied at just 300 function evaluations which is when the search

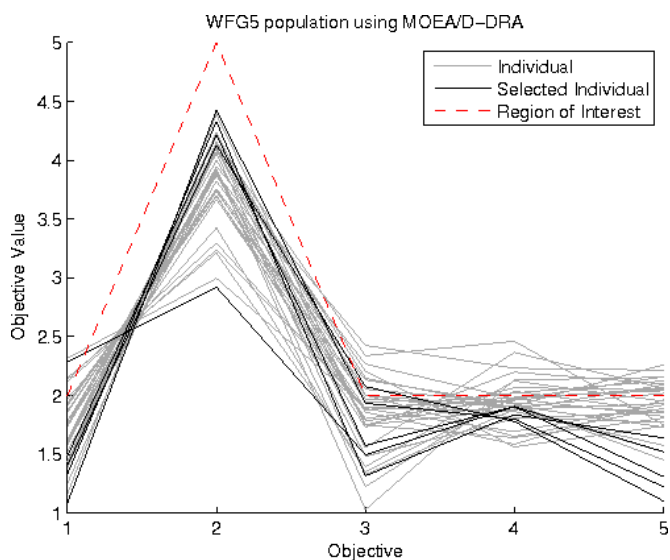


Figure 5.6: Population generated by MOEA/D-DRA combined with WZ preference articulation operator after 2000 function evaluations for the synthetic test problem WFG5. 25 solutions have been found in the desired ROI.

switched from the W-phase to the Z-phase in the WZ preference articulation operator, the results at 300 function evaluations have been illustrated in Figure 5.7.

As a preliminary comparison for proof-of-concept MOEA/D-DRA without preference articulation was executed for 2000 function evaluations on WFG5, the results of which are illustrated in Figure 5.8, for which no solutions were found within the ROI, due to lack of focus toward the desired ROI during the search. This is to be expected in the absence of preference articulation.

The WZ preference articulation operator is algorithm agnostic and can therefore be applied to any EMO algorithm as either a primary or secondary sorting criterion for use by a selection operator. As a demonstration, in Figure 5.9 the WZ preference articulation operator has been combined with an initial population for the ZDT4 synthetic test problem from the ZDT test suite, generated

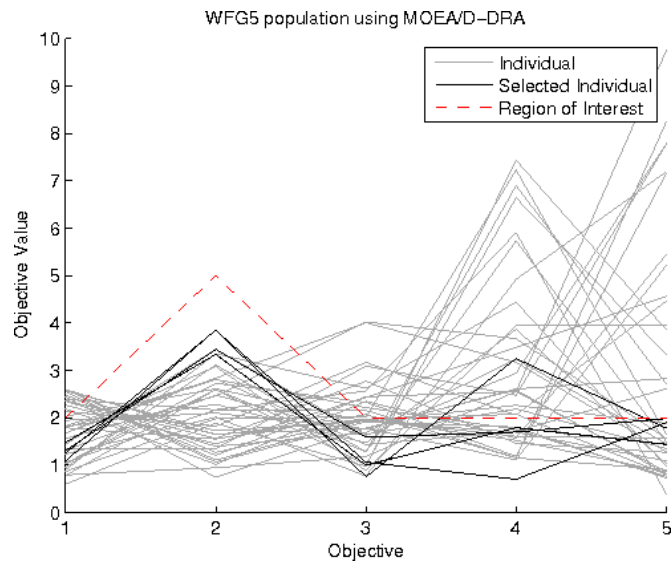


Figure 5.7: Population generated by MOEA/D-DRA combined with the WZ preference articulation operator after 300 function evaluations for the synthetic test problem WFG5. 6 solutions have been found in the desired ROI.

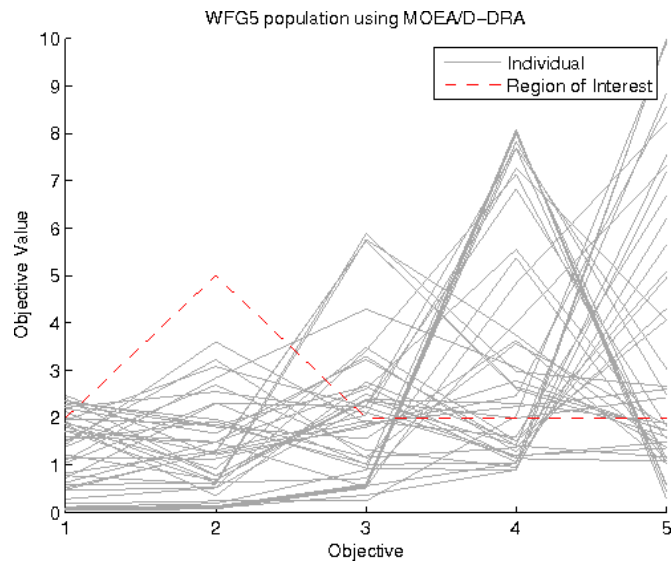


Figure 5.8: Population generated by MOEA/D-DRA without preference articulation after 2000 function evaluations for the synthetic test problem WFG5. No solutions have been found in the desired ROI.

by CMA-PAES [15], an algorithm which uses covariance matrix adaptation for search and adaptive grid archiving for selection and maintenance of a population. After 2000 function evaluations CMA-PAES combined with the WZ preference articulation operator has produced the results illustrated in Figure 5.10. ZDT4 is a test problem with many deceptive Pareto-optimal fronts which makes it computationally expensive for an EMO algorithm to find an approximation set close in proximity to or along the true-front. However, when searching toward and within a specified ROI it is possible to cut down the computational cost of the search by reducing the search space that is explored, whilst still resolving solution individuals which are of interest to the DM.

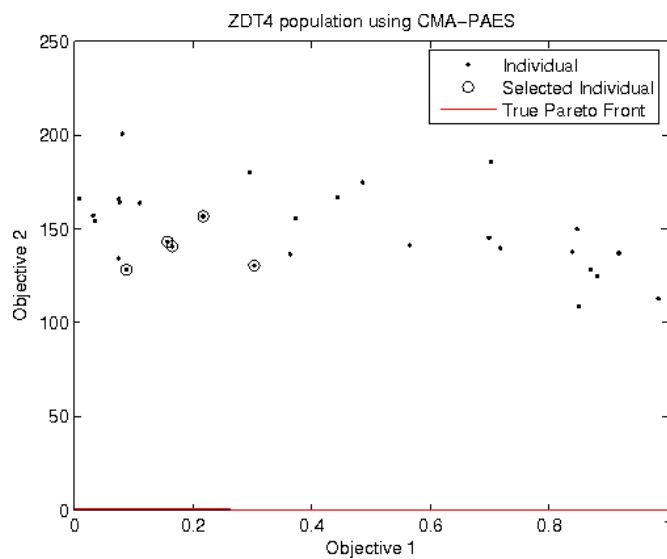


Figure 5.9: Basic Z-score preference articulation applied to an initial population generated by CMA-PAES for the synthetic test problem ZDT4.

The WZ preference articulation operator shows promise in finding solutions within a desired ROI by focussing the search and preventing exploration of areas of the search space that will not be of expressed interest to the DM, and it can be

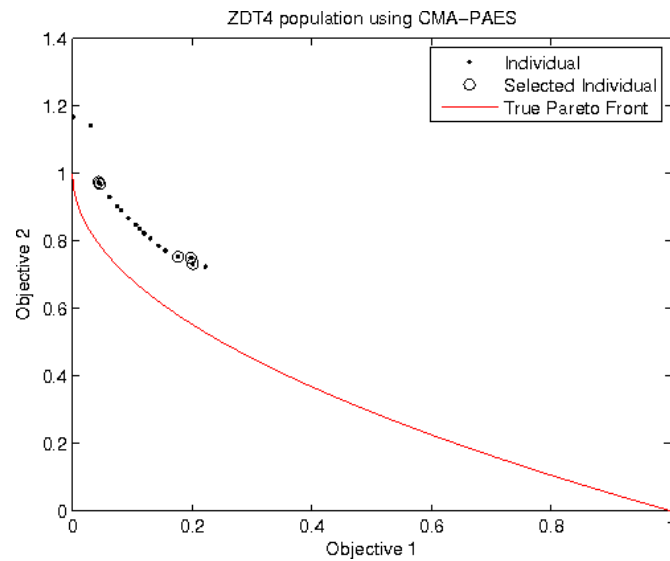


Figure 5.10: Population generated by CMA-PAES combined with basic Z-score preference articulation after 2000 function evaluations for the synthetic test problem ZDT4.

observed from Figures 5.6 and 5.8 that in its absence the performance (in regards to the number of solutions within the ROI) is worse.

Information regarding the number of standard deviations between an objective value and the decision maker's preferred goal is available, this is because the WZ preference articulation operator is based around the use of z-scores from statistics. This information can be used to blacklist solutions from the selection process, in cases where a solution exists with a preferable aggregate W-phase score, but where an objective value is greater than three standard deviations from the preferred goal.

The WZ preference articulation operator is a portable and auxiliary operator which can be incorporated into any host³ algorithm, therefore its performance in finding solutions within the ROI will ultimately be relative to that of its host and

³An algorithm that incorporates the WZ preference articulation operator is referred to as the host.

the method of incorporation (whether it is used as the sole measure of fitness or used in conjunction with other fitness operators).

5.1.1 WZ Preference Articulation Operator Worked-Example

In this section a complete worked-example of both phases of the WZ preference articulation operator (described in Section 5.1) is demonstrated. The objective values used in these examples are not the result of any objection function, they have only been selected for demonstration and ease of replication.

The W-Phase

This example assumes a five-objective problem ($M = 5$) with a population of four ($N = 4$) solutions X_n where $T = 2$, this population has been presented in Table 5.1 and plotted in Figure 5.11. The preference vector $P = \langle 1, 3, 4, 2, 1 \rangle$ defines the ROI for which the solutions are desired to be within. In order to decide which phase of the WZ preference articulation operator takes effect, Ψ needs to be calculated.

Table 5.1: An example population X of objective values x_{mn} .

	x_{1n}	x_{2n}	x_{3n}	x_{4n}	x_{5n}
X_1	0.5	0.5	5.0	2.5	1.5
X_2	0.6	0	5.0	3.0	1.4
X_3	0.5	3.5	4.5	2.5	1.5
X_4	0.8	3.2	4.2	3.0	1.2

Table 5.2 shows that Ψ for the current population against the preference information has been resolved as 0 and because ($\Psi < T$) the W-phase of the WZ

preference articulation operator takes effect. In order to find out which preference entries ρ_m have been satisfied the least by the population, the entries ω of vector Ω are calculated. The entries of Ω are then normalised to values between 0 and 1 using $f(k, K)$ and it can be seen that ρ_3 , ρ_4 and ρ_5 have the least number of solutions satisfied, therefore the scores for those objectives will receive weighting.

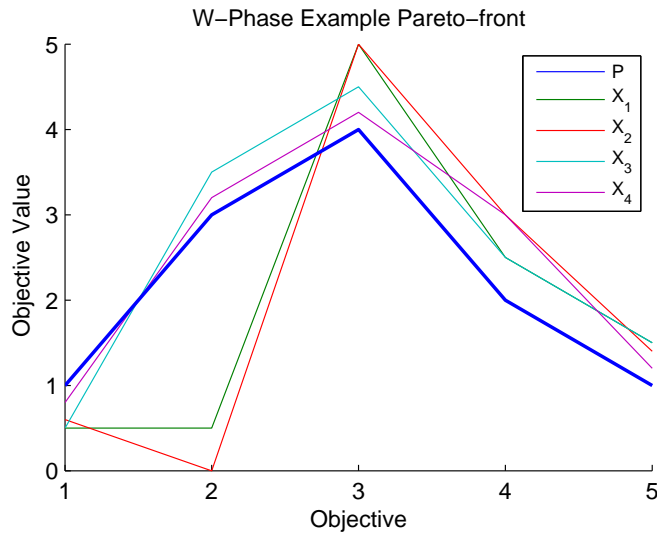


Figure 5.11: Parallel-coordinate plot of the Population X and preference vector P used in the W-Phase example.

Table 5.2: Logical values of matrix S , vector Ω and its normalised values, vector Φ and the scalar Ψ calculated for population X .

	s_{1n}	s_{2n}	s_{3n}	s_{4n}	s_{5n}	$\mathbf{0}$	Ψ
S_1	1	1	0	0	0	0	ϕ_1
S_2	1	1	0	0	0	0	ϕ_2
S_3	1	0	0	0	0	0	ϕ_3
S_4	1	0	0	0	0	0	ϕ_4
Ω	4	2	0	0	0		
$f(\omega, S)$	1	0.5	0	0	0		

In Table 5.3 it can be seen that ϵ_{mn} has been calculated for every solution's objective value and has been aggregated per solution into a single score as the scalar W_n . Sorting the entries of W in ascending order resolves (as calculated by the WZ preference articulation operator) the proximity of each solution X_n to P in the objective space. In this example, the solutions are ordered (presented in order of ascending proximity to P) $\langle X_4, X_3, X_1, X_2 \rangle$.

Table 5.3: The matrix E containing the weighted scores for the population X , and the aggregated weighted score W for each solution X_n .

	ϵ_{1n}	ϵ_{2n}	ϵ_{3n}	ϵ_{4n}	ϵ_{5n}		W
E_1	1.0	0.82	0.8	0	0.8		0.86 W_1
E_2	0.67	1.0	0.8	0.8	0.53		0.95 W_2
E_3	1.0	0.11	0.3	0	0.8		0.55 W_3
E_4	0	0	0	0.8	0		0.2 W_4

In the event that no solutions are found within the ROI, or if $(\Psi < T)$ is not satisfied throughout the optimisation process, then the WZ preference articulation operator will remain in the W-phase until the host algorithm meets its termination criteria. The result of the WZ preference articulation operator remaining in the W-phase is a final approximation set of solutions X_n that are close in proximity to to the preference vector P . This allows the algorithm to still produce an approximation set of feasible solutions that are close in proximity to the DM's ROI in the scenario where no solutions exist within the ROI.

The Z-Phase

This example also assumes a five-objective problem ($M = 5$) with a population of four ($N = 4$) solutions X_n where $T = 2$, this population has been presented

in Table 5.4 and plotted in Figure 5.12. The preference vector $P = \langle 1, 3, 4, 2, 1 \rangle$ defines the ROI for which the solutions are desired to be within. In order to decide which phase of the WZ preference articulation operator takes effect, Ψ needs to be calculated.

Table 5.4: An example population X of objective values x_{mn} .

	x_{1n}	x_{2n}	x_{3n}	x_{4n}	x_{5n}
X_1	1.5	2.2	3.5	1.5	1.0
X_2	0.5	2.5	3.5	1.2	0.5
X_3	0	2.5	1.0	1.5	0
X_4	1.0	1.0	2.0	0.5	1.0

Table 5.5 shows that Ψ for the current population against the preference information has been resolved as 3 and because ($\Psi \geq T$) the Z-phase of the WZ preference articulation operator takes effect. This allows for the minimisation of the solutions within the found discovered by calculating z_{mn} for every x_{mn} and aggregating them into a single score as the scalar V_n , so that the solutions X_n may be sorted in order of descending proximity to P .

Table 5.5: Logical values of matrix S , vector Φ and the scalar Ψ calculated for population X

	s_{1n}	s_{2n}	s_{3n}	s_{4n}	s_{5n}	3	Ψ
S_1	0	1	1	1	1	0	ϕ_1
S_2	1	1	1	1	1	1	ϕ_2
S_3	1	1	1	1	1	1	ϕ_3
S_4	1	1	1	1	1	1	ϕ_4

In Table 5.6 it can be seen that z_{mn} has been calculated for every solution's objective value and has been aggregated per solution as the scalar V_n . Sorting

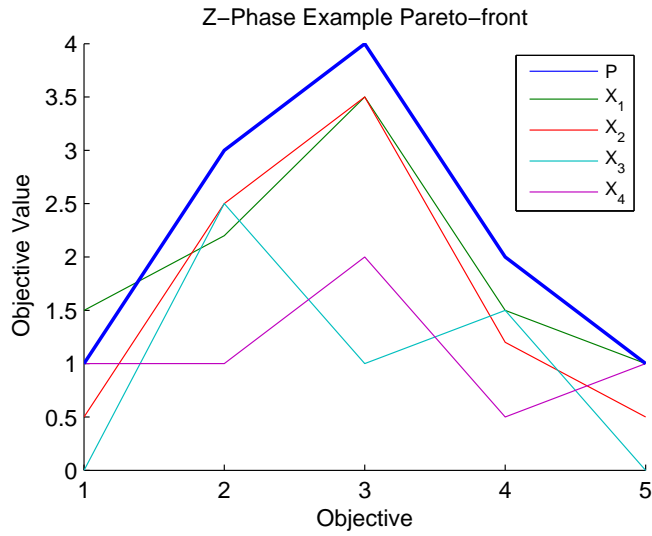


Figure 5.12: Parallel-coordinate plot of the Population X and preference vector P used in the Z-Phase example.

the entries of V in ascending order resolves (as calculated by the WZ preference articulation operator) the solutions X_n within the ROI and furthest in proximity to P in objective space. In this example, the solutions are ordered (presented in order of descending proximity to P) $\langle X_3, X_4, X_2, X_1 \rangle$.

Table 5.6: The matrix Z containing the z-scores for the population X , and the aggregated z-score V for each solution X_n .

	z_{1n}	z_{2n}	z_{3n}	z_{4n}	z_{5n}	V	
Z_1	0.82	-0.71	-0.27	-0.54	0	-0.14	V_1
Z_2	-0.82	-0.44	-0.27	-0.87	-0.89	-0.66	V_2
Z_3	-1.63	-0.44	-1.63	-0.54	-1.79	-1.21	V_3
Z_4	0	-1.76	-1.09	-1.63	0	-0.9	V_4

5.2 Incorporation of the WZ Preference Articulation Operator into CMA-PAES-II

The WZ preference articulation operator has been incorporated into the Covariance Matrix Adaptation Pareto Archived Evolution Strategy II (CMA-PAES-II) (introduced in Chapter 4) in order to both test the portability of the operator itself as well as the feasibility of preference articulation on test functions containing many objectives. CMA-PAES-II has been designed as an algorithm with the optimisation of many-objective test functions in mind, in order to retain the optimisation benefits CMA-PAES-II provides, it is important to incorporate the WZ preference articulation operator in a way that assists the selection process rather than completely replacing it.

An augmented algorithm named the Weighted Z-score Covariance Matrix Adaptation Pareto Archived Evolution Strategy (WZ-CMA-PAES) has been developed, featuring an optimisation scheme that is focussed on optimisation towards and within a DM's ROI. This scheme works in different phases which are activated depending on when certain criteria are satisfied, allowing the optimisation process to efficiently spend the function evaluation budget depending on the current optimisation context.

WZ-CMA-PAES operates in one of four phases, each of which override the "AGA Selection for Next Generation" stage in the execution life-cycle of CMA-PAES-II (illustrated in Figure 4.8). Phase 1 is active whilst there are no solutions in the current approximation set which are within the DM's expressed ROI, this phase uses the WZ clustering algorithm described in Section 5.1 for selection of

individuals that are closest to the DM's expressed ROI as parents for the next generation.

Phase 2 is active whilst there are solutions in the current approximation set which are within the DM's expressed ROI, whilst the number of these solutions is below a threshold Z_{thresh} , this phase continues to use the WZ clustering algorithm described in Section 5.1 whilst retaining solutions in the archive which are within the DM's expressed ROI.

Phase 3 is active when the number of solutions in the current approximation set which are within the DM's expressed ROI equal or exceed Z_{thresh} , this phase uses the Z-score preference articulation method described in Equation 5.2 whilst retaining solutions in the current approximation set which are within the DM's expressed ROI. This phase aims to populate the current archive entirely with solutions that are within the current ROI.

Phase 4 is active when the archive for the current generation is at capacity containing only solutions which are within the DM's expressed ROI. In this phase, when selecting which solutions to use as parents in the next generation, all solutions that are not within the ROI are automatically discarded and the remainder are subjected to the CMA-PAES-II AGA scheme, with the DM's expressed goals as the reference point for the IBC scheme, this encourages diversity and hypervolume indicator quality in the final approximation set.

By using these four phases the optimisation process is able to quickly get as close as possible to the DM's expressed ROI, produce solutions within it, populate an entire archive with solutions only within that ROI, and then converge further into that ROI with a diverse approximation set.

5.3 Incorporation of the WZ Preference Articulation Operator into MOEA/D-DRA

The WZ preference articulation operator has been incorporated into Multi-Objective Evolutionary Algorithm Based on Decomposition with Dynamical Resource Allocation (MOEA/D-DRA) (introduced in Section 2.7.4) in order to both test the portability of the operator itself as well as the feasibility of preference articulation on test functions containing many objectives. MOEA/D-DRA has been designed as an algorithm with the optimisation of test functions consisting of complex Pareto-optimal sets in mind, it is important to retain the benefits of MOEA/D-DRA by incorporating the WZ preference articulation operator in a way that assists the selection process rather than completely replacing it.

The Weighted Z-score Multi-Objective Evolutionary Algorithm Based on Decomposition with Dynamical Resource Allocation (WZ-MOEA/D-DRA) operates in one of two phases (W-phase and Z-phase) dictated by the WZ preference articulation operator, which take effect depending on when certain criteria are satisfied, allowing the optimisation process to efficiently spend the function evaluation budget depending on the current optimisation context.

Whilst the number of solutions satisfying the preference vector P is below the threshold ($\Psi < T$) the W-phase of the WZ preference articulation operator takes effect. In this phase the MOEA/D-DRA's utility selection (step 2 of Algorithm 1) is replaced with a selection of solutions based on their W_n score calculated using Equation 5.18.

If during the optimisation process the threshold ($\Psi \geq T$) is satisfied then the Z-phase of the WZ preference articulation operator takes effect, whilst in this

phase a modified implementation of MOEA/D-DRA's utility selection is used, where the edging sub-problems are no longer considered as elite and solutions that do not satisfy ($\phi_n = 0$) the DM's expressed preferences P are discarded.

Using these two phases WZ-MOEA/D-DRA is able to get close in proximity to the DM's expressed ROI within a small number of function evaluations, and then produce solutions within the ROI and minimise solutions whilst retaining the diversity features of MOEA/D-DRA.

5.4 Comparison of WZ-CMA-PAES and WZ-MOEA/D-DRA

In order to evaluate the performance of the WZ preference articulation operator, test cases have been designed containing both *a priori* preferences and progressive preferences in Section 5.4.1, and the method of performance assessment has been described in Section 5.4.2.

Two test suites are considered in the comparison of WZ-MOEA/D-DRA to MOEA/D-DRA and WZ-CMA-PAES to CMA-PAES-II: the real-valued test problems found in the ZDT bi-objective test suite proposed in [38]; and the scalable WFG multi-objective test suite proposed in [40].

The algorithm configurations for WZ-MOEA/D-DRA and WZ-CMA-PAES are listed in and Tables 5.7 and 5.8 respectively. The algorithm configurations for MOEA/D-DRA and CMA-PAES-II are the same as those used for WZ-MOEA/D-DRA and WZ-CMA-PAES, with the exclusion of the Z_{thresh} parameter.

Table 5.7: Parameter configurations used for WZ-MOEA/D-DRA.

Parameter	Configuration
μ Population	50
Niche	25
Maximum Update Number	10
Z_{thresh}	5
Maximum Function Evaluations	10,000

Table 5.8: Parameter configurations used for WZ-CMA-PAES.

Parameter	Configuration
Archive Capacity	50
Grid Divisions	2
μ Population	10
Θ_{max}	50
Θ_{min}	10
Z_{thresh}	5
Maximum Function Evaluations	10,000

5.4.1 Test Cases

A test case consisting of a chosen ROI has been chosen for each test function, to test the convergence of each algorithm to different areas of objective space. These test cases have been defined in Table 5.9 for two-objective problems and in Table 5.10 for five-objective problems.

Table 5.9: Test cases defining the ROI used for each two-objective test function.

Test Function	Region of Interest
ZDT1	1.0, 0.2
ZDT2	0.3, 1.0
ZDT3	1.0, -0.4
ZDT4	0.2, 1.0
ZDT6	1.0, 0.2

Based on results from previous comparison of the host evolutionary algorithms in Section 4.7.1 and 4.7.3 for two and five objective test functions respectfully, only MOEA/D-DRA and its WZ variant are executed on the bi-objective test cases, where as both MOEA/D-DRA, CMA-PAES-II and their WZ variants are

Table 5.10: Test cases defining the ROI used for each five-objective test function.

Test Function	Region of Interest
WFG1	2.5, 1.0, 1.0, 1.0, 1.0
WFG2	1.0, 0.5, 0.5, 0.5, 2.5
WFG3	0.5, 0.5, 1.0, 2.0, 10
WFG4	2.0, 1.0, 6.0, 1.0, 2.0
WFG5	5.0, 5.0, 1.5, 1.5, 1.5
WFG6	2.0, 0.5, 0.5, 1.0, 11
WFG7	0.6, 0.6, 0.6, 8.5, 0.6
WFG8	0.5, 0.5, 0.5, 9.0, 0.5
WFG9	3.0, 1.0, 1.0, 2.0, 1.5

executed on the five-objective test cases. This is because CMA-PAES-II is intended as a many-objective EMO and this study is not a direct comparison between the two individual algorithms, but instead a comparison between an EMO with and without incorporation of the WZ preference articulation operator.

In Table 5.11 two test cases have been designed to demonstrate WZ preference articulation when applied to scenarios involving a change in preferences during the optimisation process. Selected for this demonstration is the ZDT1 test function as it is a bi-objective problem which is easily visualised, and LAT-CON (multi-objective design optimisation of an aircraft lateral control system) a seven-objective test function taken from [158].

Table 5.11: Progressive Preference Articulation Test Cases.

Test Function	Generation	Region of Interest
ZDT1	0 to 399	1.0, 0.1
	400 to 699	0.6, 0.5
	700 to 1000	0.2 0.9
LATCON	0 to 399	0.75, -360, -0.01, -3.75, -0.45, -1, -90
	400 to 599	0.75, -1200, -0.01, -3.75, -0.45, -1, -90
	600 to 1000	0.75, -800, -0.01, -3.75, -0.45, -1, -200

5.4.2 Performance Assessment

For the two-objective and five-objective test functions, all considered algorithms (MOEA/D-DRA, WZ-MOEA/D-DRA, WZ-CMA-PAES, and CMA-PAES-II) have been executed with an allowance of 10,000 function evaluations 25 times on each test function to reduce stochastic noise, and the population at each generation of each execution has been scored using the hypervolume indicator. This sample size is seen as sufficient because of the limited benefit of producing more than 25 samples (discussed in Section 2.10.5).

The hypervolume indicator is selected because it is scaling independent and requires no prior knowledge of the true Pareto-optimal front, this is important when working with real-world problems which have not yet been solved. The hypervolume indicator is currently used in the field of multi-objective optimisation as both a performance metric and in the decision making process [114, 115].

The hypervolume indicator allows performance comparison between WZ-MOEA/D-DRA and MOEA/D-DRA based on which algorithm covers the greatest amount of the search space within a specified ROI, by using the preference vector P described in Equation 5.6 in place of f^{ref} when calculating the hypervolume in-

indicator value. This means until the optimisation process has found solutions within the ROI, the algorithms current execution is considered to be performing at zero. This will be the basis for identifying which algorithm outperforms the other, an example of this measure has been given in Figure 2.26.

Oscillation in hypervolume indicator performance from generation to generation is expected on some test functions as neither MOEA/D-DRA or CMA-PAES-II are entirely hypervolume indicator driven algorithms, with MOEA/D-DRA not utilising the hypervolume indicator at all during the optimisation process and CMA-PAES-II only utilising it at grid level. WZ variants of both CMA-PAES-II and MOEA/D-DRA are of course considered to outperform their non-WZ counter-parts, in particular because their non-WZ counter-parts will be attempting to maintain diversity across the entire objective space where as the WZ variants will be attempting to maintain diversity across the ROI.

For the progressive preference articulation test cases, WZ-CMA-PAES has been compared to CMA-PAES-II. For WZ-CMA-PAES the population prior to a change in preferences or algorithm termination has been scored using the relevant reference points taken from Table 5.11 and compared to a final population generated by CMA-PAES-II within the same function evaluation budget.

5.5 Results

The discussion of results for the test-cases defined in Section 5.4.1 has been divided into three sections: Section 5.5.1 for the two-objective test-cases, Section 5.5.1 for the five-objective test-cases, and Section 5.5.3 for the progressive preference articulation test-cases.

5.5.1 Two-Objective Results

From a general observation of the box plots in Figure 5.13 it can be seen that WZ-MOEA/D-DRA outperforms MOEA/D-DRA on each of the ZDT test cases using the hypervolume indicator. WZ-MOEA/D-DRA succeeds in finding solutions within the specified ROI in fewer function evaluations than MOEA/D-DRA, even when MOEA/D-DRA fails to find any solutions in the ROI throughout the entire search process (within the function evaluation budget). The populations found by WZ-MOEA/D-DRA on each test function cover more of the hypervolume within the specified ROI than MOEA/D-DRA, and because of this the DM will have a more diverse set of candidate solutions to make a decision from, giving them a better idea of the trade-offs within the specified ROI with a better spread throughout the approximated front.

Table 5.12 presents information regarding the p -value resolved by the Wilcoxon signed-ranks non-parametric test for the considered synthetic test problems, and a symbol indicating the observation of the null hypothesis. A '+' indicates that the null hypothesis was rejected, and WZ-MOEA/D-DRA displayed statistically superior performance at the 95% significance level ($\alpha = 0.05$) on the considered synthetic test function. A '-' indicates that the null hypothesis was rejected,

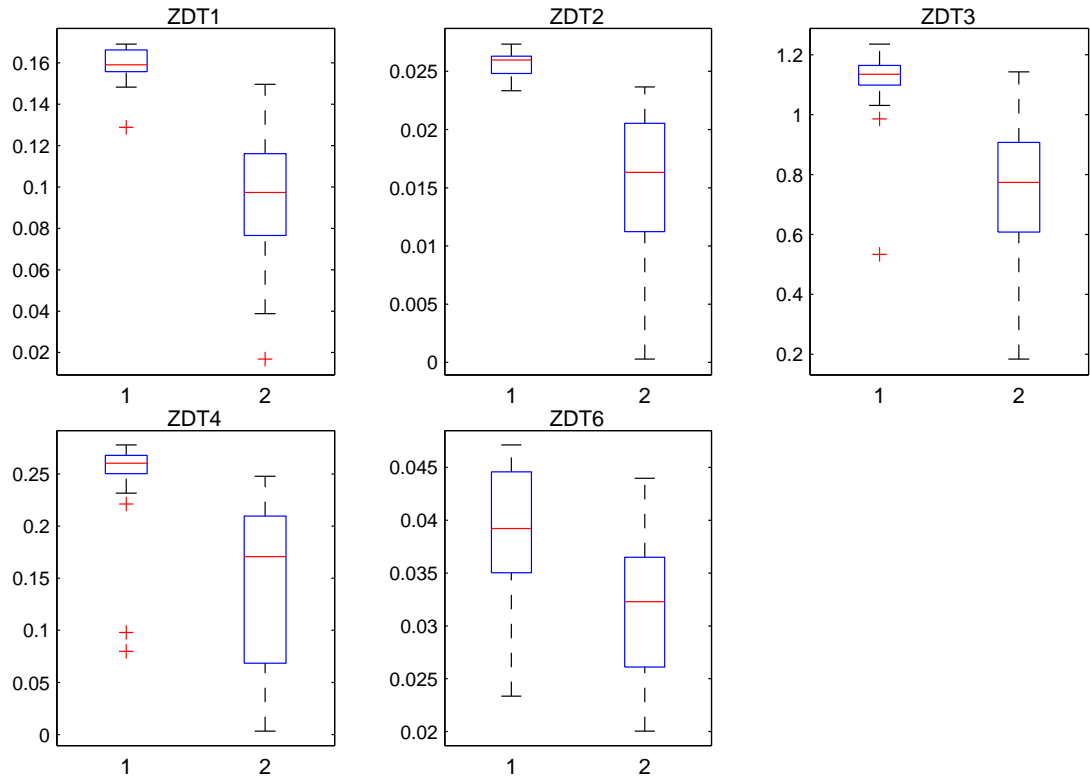


Figure 5.13: Box plots of hypervolume indicator results for two-objective ZDT problems (1: WZ-MOEA/D-DRA; 2: MOEA/D-DRA) 10,000 function evaluations, 25 runs.

and WZ-MOEA/D-DRA displayed statistically inferior performance. An '=' indicates that there was no statistical difference between both of the considered algorithms on the synthetic test function.

In all cases the null hypothesis was rejected and a statistical significance of greater than 95% was observed. In addition, in all test cases WZ-MOEA/D-DRA scores a median greater than the interquartile range when compared to MOEA/D-DRA, and in all cases but ZDT6, WZ-MOEA/D-DRA's entire interquartile range is robust and outperforms MOEA/D-DRA's.

Plots for the results of the test cases have been illustrated in Figure 5.14 for

Table 5.12: Results from pairwise comparison of the final approximation sets of both considered algorithms on each two-objective synthetic test function using the Wilcoxon signed-ranks non-parametric test.

Test Function	p -value	
ZDT1	2.2857e-09	+
ZDT2	2.5742e-09	+
ZDT3	1.1773e-07	+
ZDT4	1.309e-07	+
ZDT6	0.00051446	+

two-objective test problems. Each plot presents the mean hypervolume covered within the ROI at each generation for 25 executions on a test function, for both MOEA/D-DRA and WZ-MOEA/D-DRA.

The results show that the test function posing the greatest difficulty in finding a single solution within the specified ROI is ZDT4, this can be seen in Figure 5.14d where the plot of the mean hypervolume indicator value over the number of generations indicates that of all the test functions, ZDT4 required each algorithm to search for more generations before finding solutions within the ROI. This is to be expected as a result of ZDT4's multi-frontal nature [159] which is a struggle for most EMO algorithms; however, when comparing WZ-MOEA/D-DRA to MOEA/D-DRA it is clear that WZ-MOEA/D-DRA can get to the specified ROI faster and produce a final approximation set of better hypervolume indicator quality. In Figure 5.15 the populations for WZ-MOEA/D-DRA and MOEA/D-DRA have been plotted for an execution for which the hypervolume indicator value was close to the mean of each respective algorithms 25 executions, these plots show that WZ-MOEA/D's final approximation set has converged through more of the deceptive Pareto-optimal fronts than MOEA/D-DRA, with more

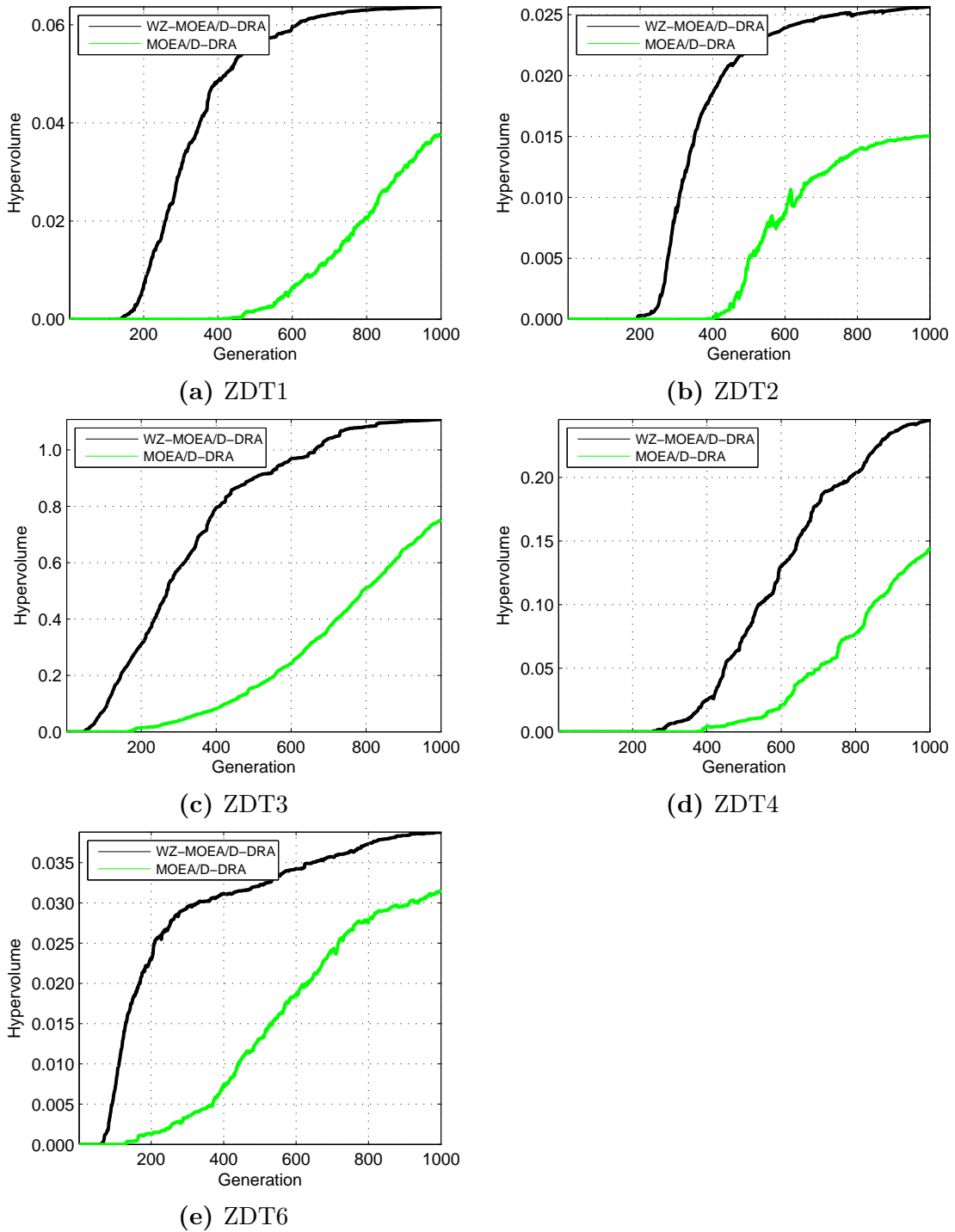


Figure 5.14: The mean hypervolume indicator values of WZ-MOEAD-DRA and MOEA/D-DRA populations at each generation for two-objective ZDT test suite.

candidate solutions, better diversity, and also including more solutions near the extremes of the ROI. A similar result can be seen in Figure 5.16 where the worst of the 25 populations for WZ-MOEA/D-DRA and MOEA/D-DRA have also been plotted for ZDT4, in this case however MOEA/D-DRA has only found a single solution within the ROI where as WZ-MOEA/D-DRA has completed with better proximity and diversity.

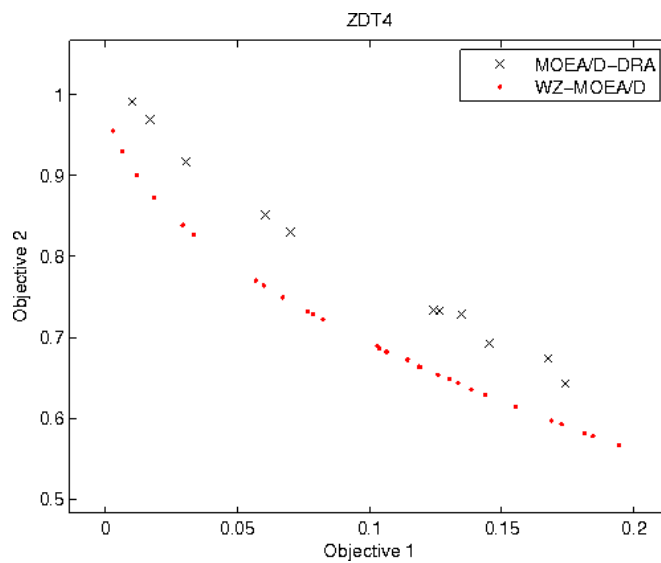


Figure 5.15: Population generated by a run of MOEA/D-DRA and WZ-MOEA/D-DRA on ZDT4 within the ROI, with a hypervolume indicator value close to the mean of the 25 executions.

The plots in Figure 5.14 show that in every test function considered, the WZ variant of MOEA/D-DRA not only finds solutions in the ROI many generations prior to MOEA/D-DRA, but it also achieves MOEA/D-DRA's peak and final hypervolume metric performance much earlier in the optimisation process and then proceeds to performs far better.

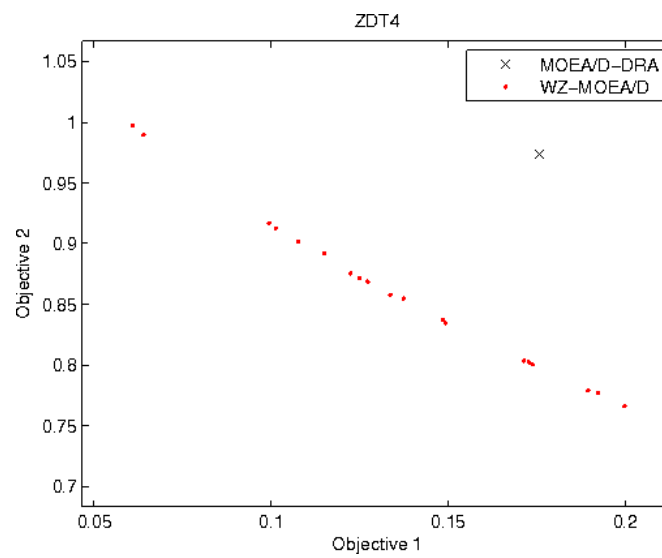


Figure 5.16: Population generated by worst run of MOEA/D-DRA and WZ-MOEA/D-DRA on ZDT4 within the ROI, with a hypervolume indicator value of 0.000063568 and 0.0159 respectively.

5.5.2 Five-Objective Results

From a general observation of the box plots in Figure 5.17 it can be seen that WZ variants of MOEA/D-DRA and CMA-PAES-II outperform their original implementations on each of the WFG test cases using the hypervolume metric. The WZ variants succeed in finding solutions in the specified ROI in fewer function evaluations than their original implementations, even when the originals fail to find any solutions in the ROI throughout the entire search process (within the function evaluation budget). The populations found by the WZ variants on each test function cover more of the hypervolume within the specified ROI than the original algorithms, and because of this the DM will have a more diverse set of candidate solutions to make a decision from, giving them a better idea of the trade-offs within the specified ROI with a better spread throughout the approximated front.

Tables 5.13 and 5.14 present information regarding the p -value resolved by the Wilcoxon signed-ranks non-parametric test for the considered synthetic test problems, and a symbol indicating the observation of the null hypothesis. A '+' indicates that the null hypothesis was rejected, and WZ algorithm variant displayed statistically superior performance at the 95% significance level ($\alpha = 0.05$) on the considered test synthetic test function. A '-' indicates that the null hypothesis was rejected, and the WZ algorithm variant displayed statistically inferior performance. An '=' indicates that there was no statistical difference between both of the considered algorithms on the synthetic test function. In all cases the null hypothesis was rejected and a statistical significance of greater than 95% was observed.

Table 5.13: Results from the pairwise comparison of the final approximation sets of both WZ-MOEA/D-DRA and MOEA/D-DRA on each five-objective synthetic test function using the Wilcoxon signed-ranks non-parametric test.

Test Function	p -value	
WFG1	7.6832e-08	+
WFG2	0.022656	+
WFG3	6.132e-07	+
WFG4	0.0012886	+
WFG5	5.806e-09	+
WFG6	5.8255e-05	+
WFG7	0.00023161	+
WFG8	1.0932e-06	+
WFG9	1.7323e-06	+

Table 5.14: Results from the pairwise comparison of the final approximation sets of both WZ-CMA-PAES and CMA-PAES-II on each five-objective synthetic test function using the Wilcoxon signed-ranks non-parametric test.

Test Function	p -value	
WFG1	1.4157e-09	+
WFG2	2.4712e-10	+
WFG3	1.4157e-09	+
WFG4	4.4598e-08	+
WFG5	1.2695e-09	+
WFG6	1.5967e-09	+
WFG7	1.4144e-09	+
WFG8	3.93e-10	+
WFG9	2.883e-09	+

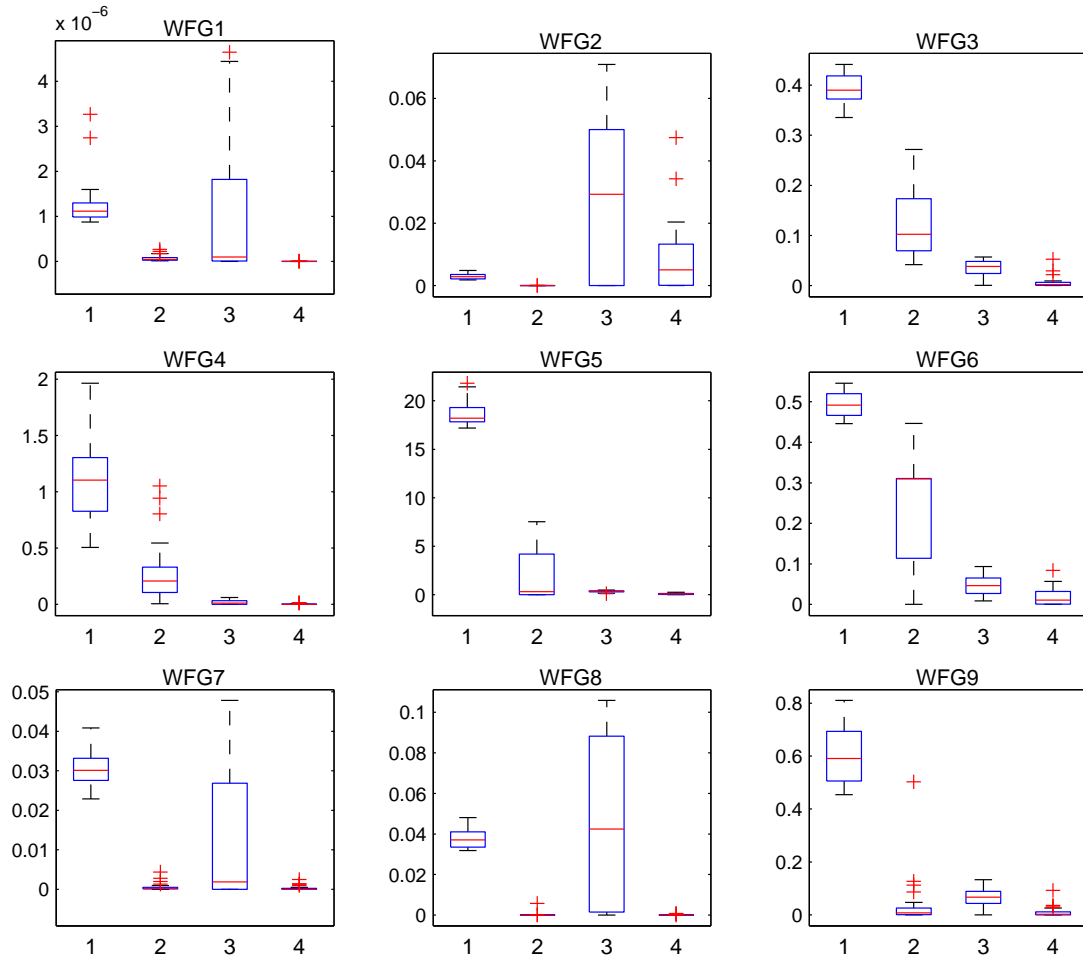


Figure 5.17: Box plots of hypervolume indicator results for five-objective WFG test cases (1: WZ-CMA-PAES; 2: CMA-PAES-II; 3: WZ-MOEA/D-DRA; 4: MOEA/D-DRA) 10,000 function evaluations, 25 runs.

In all test cases, WZ variants of MOEA/D-DRA and CMA-PAES-II score a median greater than the original, with WZ-CMA-PAES scoring the highest median hypervolume indicator result on 7 of the 9 test functions and WZ-MOEA/D-DRA scoring the highest median on the remaining 2. In some test cases (WFG3, WFG4, WFG5 and WFG6), CMA-PAES-II and WZ-CMA-PAES outperform MOEA/D-DRA and WZ-MOEA/D-DRA, this is because of CMA-PAES-II's improved performance in general on five-objective test functions in comparison to

MOEA/D-DRA as seen in Section 4.7.3.

Plots for the results of the test cases have been illustrated in Figure 5.18 for five-objective test problems. Each plot presents the mean hypervolume covered within the ROI at each generation for 25 executions on a test function for WZ-MOEA/D-DRA, MOEA/D-DRA, WZ-CMA-PAES, and CMA-PAES-II.

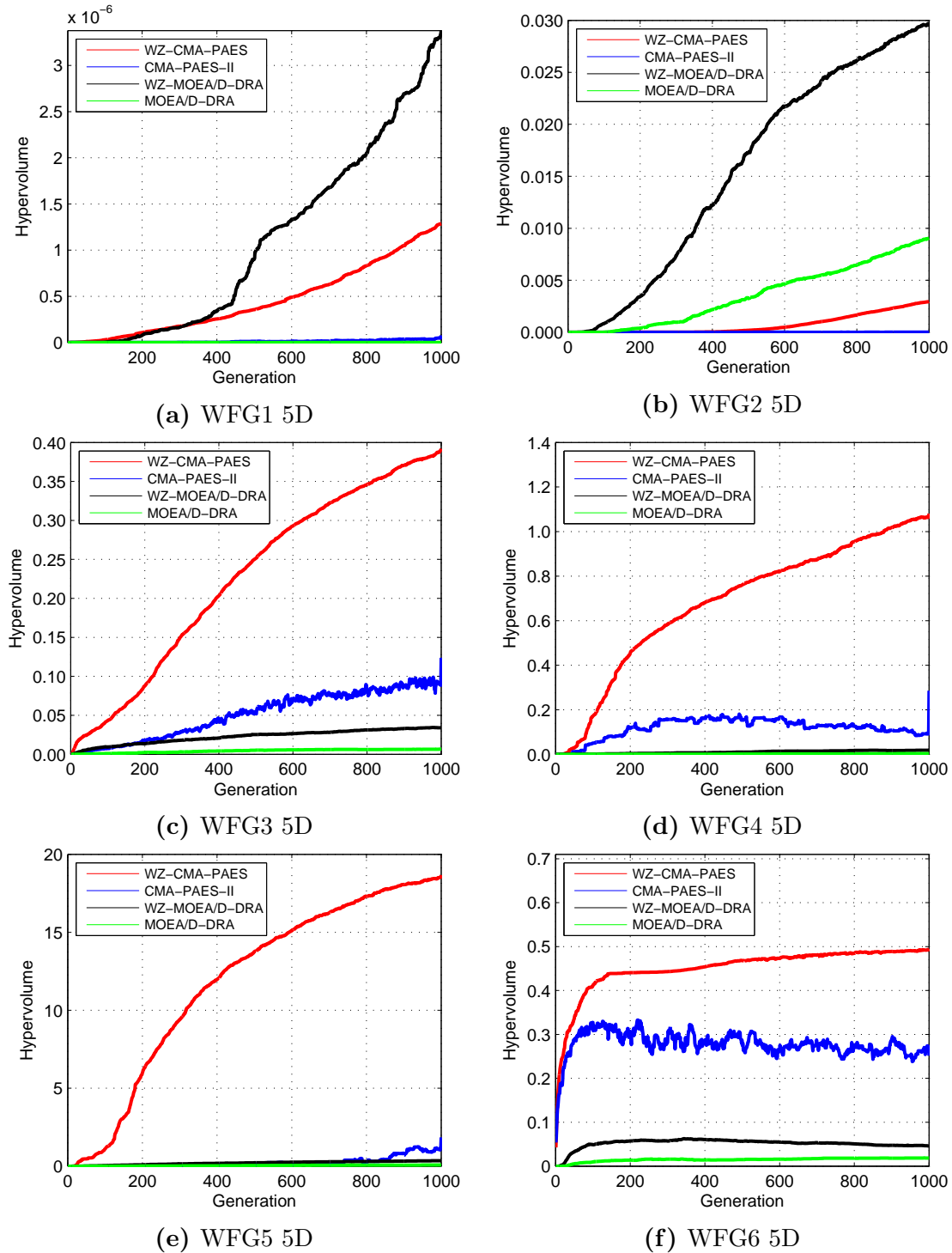


Figure 5.18: The mean hypervolume indicator value of WZ-MOEA/D, MOEA/D-DRA, WZ-CMA-PAES, and CMA-PAES-II populations at each generation for five-objective test functions WFG1, WFG2, WFG3, WFG4, WFG5, and WFG6.

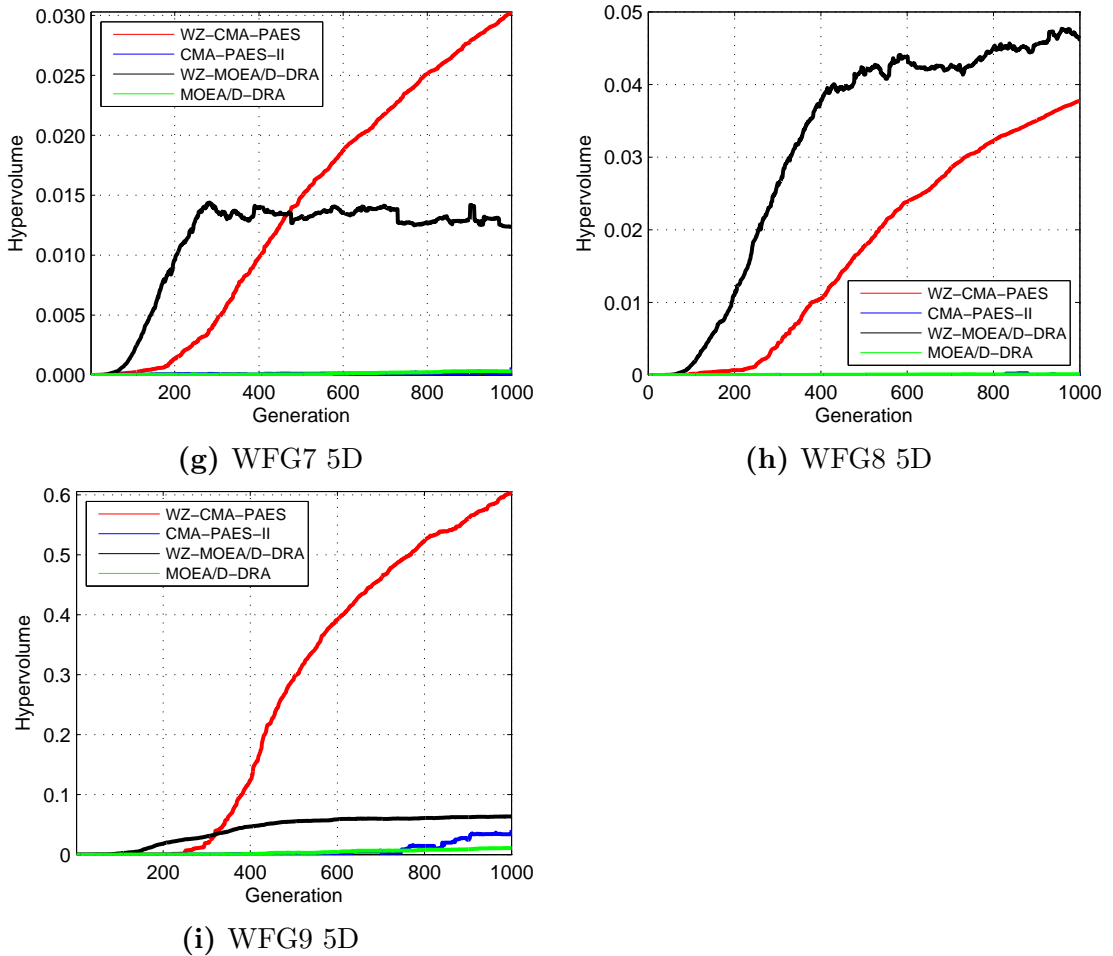


Figure 5.18: The mean hypervolume indicator value of WZ-MOEA/D, MOEA/D-DRA, WZ-CMA-PAES, and CMA-PAES-II populations at each generation for five-objective test functions WFG7, WFG8 and WFG9.

Similar to the results when CMA-PAES-II was compared to MOEA/D-DRA in Section 4.7.3, Both WZ-CMA-PAES and CAM-PAES-II continue to perform poorly on WFG2, though WZ-CMA-PAES performs far better than CMA-PAES-II on the test case. This is to be expected as the WZ operator compliments its host algorithm and aims to reduce the number of function evaluations wasted and improve the overall hypervolume indicator quality of the population, it does not replace the variation operator.

The results for WFG6 show that all considered algorithms find solutions within the ROI very early in the search, taking less than 50 generations on average. This is the result of relaxed preferences used in the test case, in particular for the 5th objective where the preference was set to 11 or below. In general, all considered algorithms find the ROI in a similar number of function evaluations, with WZ variants of the algorithms converging to populations with better hypervolume quality very early on in the search. In the worst performing (in regards to hypervolume) execution of each algorithm on WFG6, the plots for the final population show that WZ-MOEA/D-DRA in Figure 5.19 finds many solutions within the ROI with good diversity to offer the DM with an idea of the trade-offs, whereas MOEA/D-DRA in Figure 5.20 fails to find a solution in the ROI.

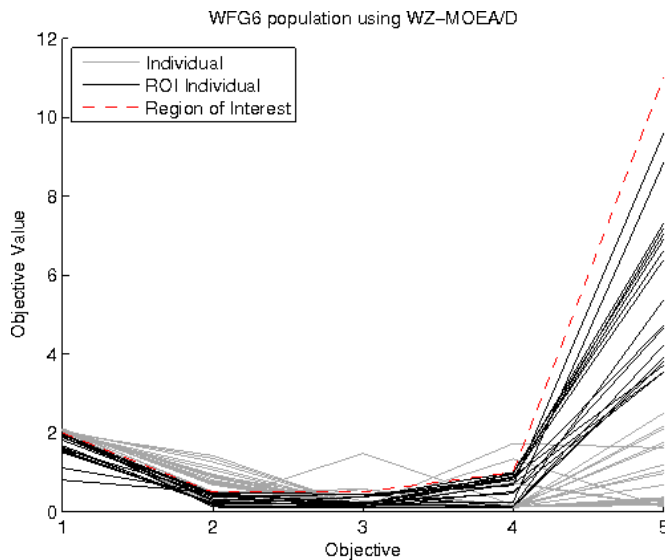


Figure 5.19: Population generated by worst run of WZ-MOEA/D-DRA on WFG, with a hypervolume indicator value of 0.0485.

In some of the results it can be seen that the achieved hypervolume indicator value oscillates throughout the optimisation process, for example in WFG4 and WFG7. This occurs because both MOEA/D-DRA and WZ-MOEA/D-DRA are

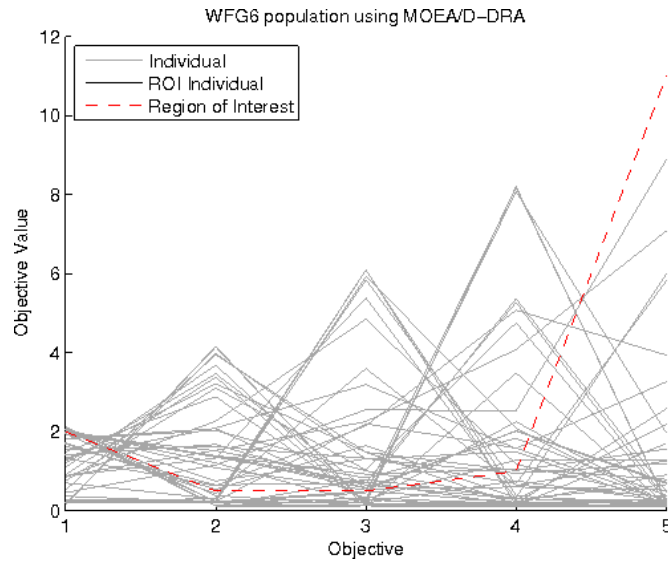


Figure 5.20: Population generated by worst run of MOEA/D-DRA on WFG6, with a hypervolume indicator value of 0.

not hypervolume indicator based algorithms and instead use a weighted selection method, therefore the hypervolume indicator is not taken into account in any point of the optimisation process, unlike algorithms such as CMA-PAES and MO-CMA-ES, which take the contributing hypervolume indicator into account during their selection process.

Table 5.15 shows the number of occurrences in which each considered algorithm did not find a solution in the ROI on each test case out of the 25 executions. From this table it can be seen that there are no test cases where WZ-CMA-PAES fails to find a solution within the expressed ROI, this is expected to be because of the promising performance of CMA-PAES-II as seen in Section 4.7.3 paired with the IBC scheme described in 4.2 translating intuitively from using a maximum hypervolume indicator reference point to using the ROI for each test case. WZ-MOEA/D-DRA also performs well when assessing the number of times the algorithm failed to find a solution in the ROI, and though it did not perform

as well as WZ-CMA-PAES in this regard, it did outperform WZ-CMA-PAES on some test cases in regards to mean and median hypervolume indicator performance. CMA-PAES-II and MOEA/D-DRA perform similarly in regards to the number of times no solution was found in the ROI, which is to be expected as they are not algorithms that incorporate a method of preference articulation.

Table 5.15: The number of occurrences in which each considered algorithm did not find a solution in the ROI on the WFG test suite with 5 objectives.

	WZ-CMA-PAES	CMA-PAES-II	WZ-MOEA/D-DRA	MOEA/D-DRA
WFG1	0	0	2	18
WFG2	0	22	6	4
WFG3	0	0	0	3
WFG4	0	0	3	8
WFG5	0	9	0	4
WFG6	0	1	0	6
WFG7	0	2	3	14
WFG8	0	20	5	20
WFG9	0	0	1	9

5.5.3 Progressive Preference Articulation Results

Overall, the results from the PPA test cases in Table 5.11 show promise in evolutionary optimisation when using the WZ preference articulation operator in the presence of online changes in DM preferences. The results predictably show that the exploitation of existing solutions that have undergone optimisation for some number of generations prior to a change in the ROI allows for the new ROI to be found in fewer function evaluations than if the optimisation process were to be restarted with the new preferences expressed *a priori*.

This has been demonstrated for the two-objective test function ZDT1 from the ZDT test suite and the seven-objective LATCON real-world problem. In order to compare the effectiveness of an algorithm execution where WZ is incorporated progressively as a PPA operator, against an algorithm where WZ is incorporated *a priori*, a single algorithm execution where WZ is used progressively is compared to a number of executions per number of preferences changes where WZ is used *a priori*.

It is expected that both algorithms will perform identically in any measure of performance up until the first change in preference, which for both the test cases is the 400th generation. This has proven true in Figures 5.21a and 5.22a where both WZ with PPA and WZ with *a priori* preference articulation has achieved identical hypervolume indicator performance from generation 0 to generation 399.

After the change in preferences, it can be seen in Figure 5.21b that WZ-CMA-PAES with PPA finds the new ROI in almost 150 generations fewer than a new execution of WZ-CMA-PAES to find the same ROI, this is because WZ-CMA-PAES with PPA was able to use the existing population with optimal information for a different ROI to get to the new ROI quicker. Similarly in Figure 5.22b it can be seen that in the LATCON test-case that not only does WZ-CMA-PAES with PPA find the new ROI in fewer generations, but it finds it instantly as the ROI for generations 400 to 699 is within the initial ROI, this allows the existing population to be exploited far more efficiently.

After the final change in preferences, WZ-CMA-PAES with PPA finds solutions in the new ROI in fewer generations or instantly when compared to a new execution of WZ-CMA-PAES. In the third ZDT1 test-case, a solution is found instantly (shown in Figure 5.21c) due to the closeness of the old ROI and the new

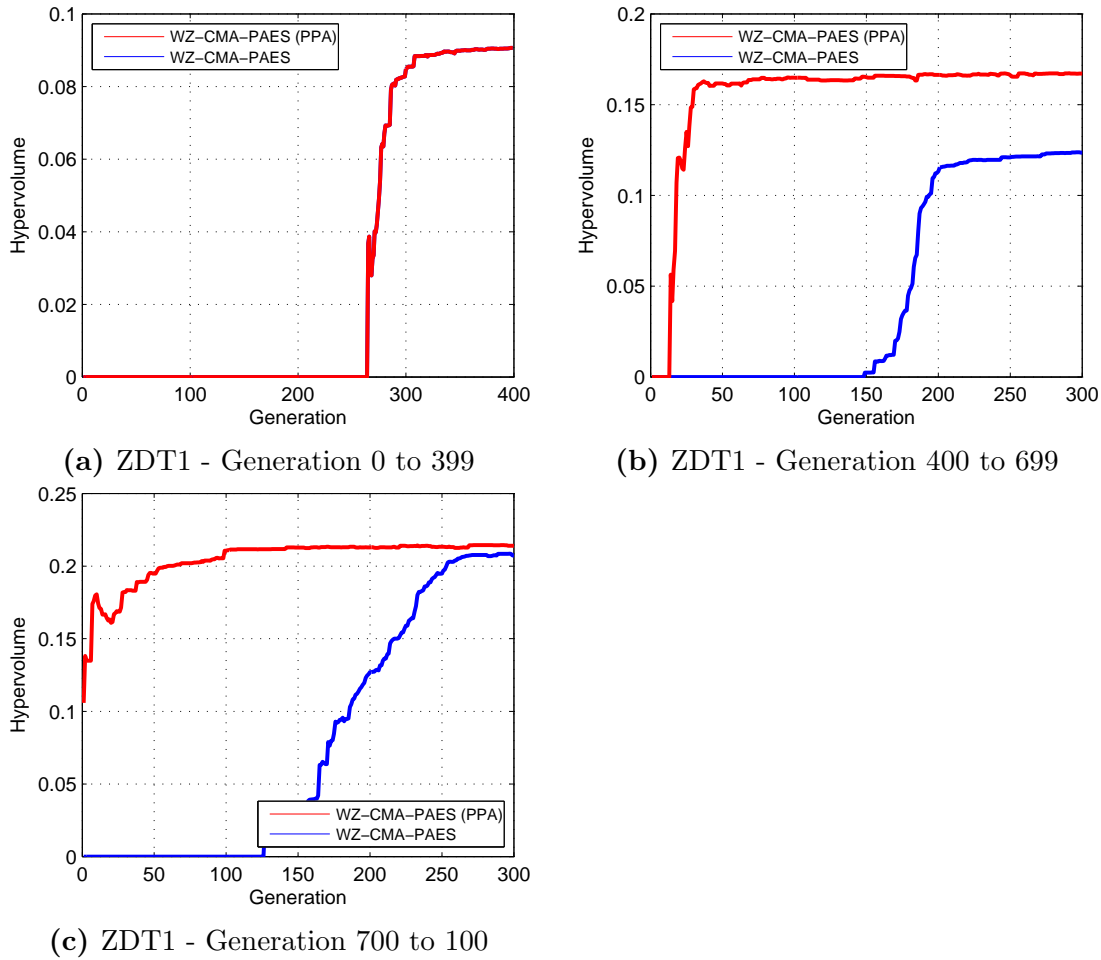


Figure 5.21: The mean hypervolume indicator value of WZ-CMA-PAES (*a priori*) and WZ-CMA-PAES (PPA) populations at each generation for the ZDT1 PPA test case.

ROI and the easiness of the ZDT1 test function. In the third LATCON test-case a solution is found instantly (shown in Figure 5.22c) when the preferences for the second objective is relaxed and the preferences for the fifth objective made more strict, in this case it is likely that either a number of solutions that satisfied both sets of preferences existed in the generation prior to the change in preferences, or the change in ROI was an easy one.

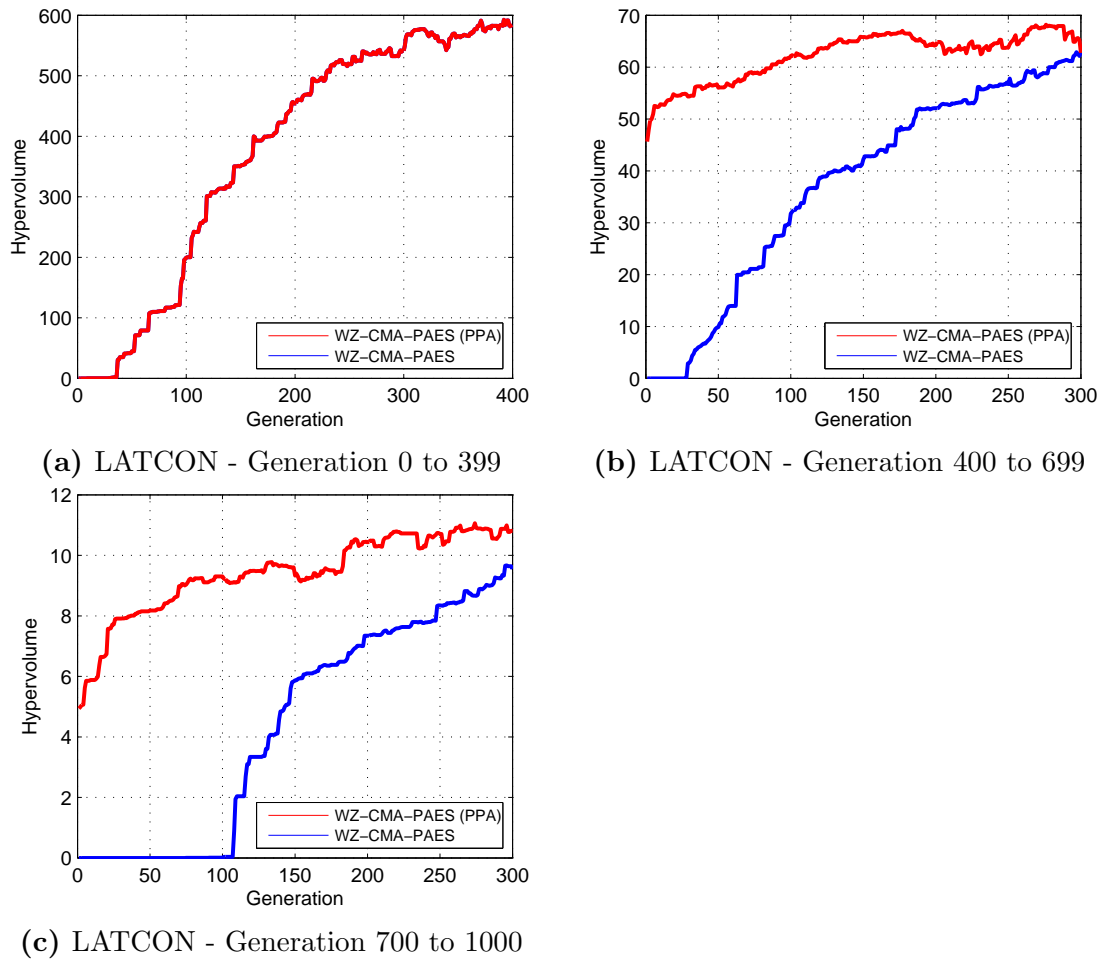


Figure 5.22: The mean hypervolume indicator value of WZ-CMA-PAES (*a priori*) and WZ-CMA-PAES (PPA) populations at each generation for the LATCON PPA test case.

5.6 Conclusion

In this chapter a novel method of preference articulation has been introduced in the form of the Weighted Z-score (WZ) preference articulation operator. The WZ preference articulation method has been incorporated into an implementation of MOEA/D-DRA and CMA-PAES-II and benchmarked on two-objective and five-objective test functions and compared to their originals. The results show that the WZ preference articulation operator has successfully improved the performance of their host algorithms (MOEA/D-DRA and CMA-PAES-II) when searching towards an expressed ROI, by producing more solutions within the ROI, producing solutions in the ROI within fewer function evaluations, and also producing populations of better hypervolume indicator quality.

With the development WZ-MOEA/D-DRA and WZ-CMA-PAES, an interesting question is raised as to the classification of these algorithms. The WZ preference articulation operator has been developed independently and then incorporated into MOEA/D-DRA and CMA-PAES-II, this may allow the resulting algorithms to be classified as hybrid algorithms, and may encourage hybridisation of other algorithms if the WZ operator is required in any other host algorithm.

In addition, the WZ preference articulation operator has been demonstrated successfully in scenarios involving progressive preference articulation, showing promising results for reducing the number of function evaluations required when exploring a new ROI, by not requiring a completely fresh execution of the optimisation process but instead continuing from any existing search and exploiting existing solutions.

Chapter 6

Application to Concealed Weapon Detection

The optimisation of the accuracy and efficiency of classifiers in pattern recognition is a complex problem that is often poorly understood. For example, whilst numerous techniques exist for the optimisation of weights in Artificial Neural Networks (ANNs) (such as the Widroff-Hoff least mean squares algorithm and back propagation), there do not exist any hard and fast rules for choosing the structure of an ANN - in particular for choosing both the size (in terms of the number of neurons) and the number of hidden layers used in the network. However, this internal structure is one of the key factors in determining the efficiency of the network and the accuracy of the classification. In recent years there has been some interest in using soft computing techniques such as Evolutionary Algorithms (EAs) to provide a solution to this problem [160], focussing on evolving the structure of an ANN to solve function approximation problems. However, complex classification problems often involve trade-offs between classification objectives that are not well suited to this kind of single-objective approach.

This chapter presents the optimisation of the ANN architecture used for con-

cealed weapon detection in a two-objective, five-objective, and seven-objective problem. The two-objective optimisation is performed on an ANN that is classifying the radar signals into two groups which are *threat* and *non-threat*. The five-objective and seven-objective optimisation are ambitious in the sense that they attempt to optimise the architecture of a number of ANNs each of which are trained to detect a specific threat item. These many-objective optimisations are the more difficult of the problems but will give a greater level of information to the user of the detection system. This will allow the security forces to react to specific threats in a more controlled manner as they will know the type of threat presented. The optimisation of concealed weapon detection classifiers is important because even a marginal gain in performance can improve the safety and security for the area in which the system is implemented.

The chapter begins with an overview of the method of concealed weapon detection used in the experiments in Section 6.1, followed by a description of how the problem is encoded into a real-value chromosome for use by the evolutionary multi-objective optimiser in Section 6.2. Section 6.3 regards the two-objective optimisation of a currently implemented and published system, and Section 6.4 regards the five-objective and seven-objective optimisation of systems able to detect multiple categories of threat. The chapter concludes with a summary of the research in Section 6.5

6.1 Concealed Weapon Detection

Concealed Weapon Detection (CWD) is an important area of research in the defence and security community. This is due to a number of high profile terrorist attacks which have resulted in loss of life and damage to public infrastructure.

The threat faced by the security forces is diversifying and current technology is struggling to meet new requirements. The technologies currently in use at airports include metal detection portals, millimetre wave imaging systems, x-ray scanners and ion mobility spectrometers. These technologies are all designed to detect specific threats to security and collectively they are used to satisfy the current requirements for screening in the aviation industry. The use of multiple technologies in airports has led to choke points with lengthy waits at security checkpoints, making this method of screening undesirable and it is only used in places where it is absolutely necessary. Recently the threat of terrorism has spread and many more public areas and government buildings are becoming targets, even sporting events have now become targets. This presents a real problem as the best examples of mass screening are demonstrated at airports and as mentioned previously this type of screening leads to choke points and delays. Therefore there is a requirement for a fast method of screening people in crowded areas, which is capable of detecting a diverse range of threats.

One method of detecting a diverse of range concealed weapons in crowded areas in real-time is to use small portable radars. A number of radar systems have been developed for this purpose [161] and [162]. These radars use multiple methods of detecting concealed weapons such as time domain reflectometry and

the exploitation of polarisation changes induced by complex objects concealed on the human torso under clothing.

The radar used in this work is constructed of a Vector Network Analyser (VNA) with pyramid horn antennas connected to the VNA using suitable cabling. A laptop is used to control the VNA and then classify the signals. The radar signals are analysed on the laptop using pattern recognition applied to the time resolved signals in the form of an ANN, this set-up has been illustrated in Figure 6.1. This method has been discussed in detail in a previous publication [163].

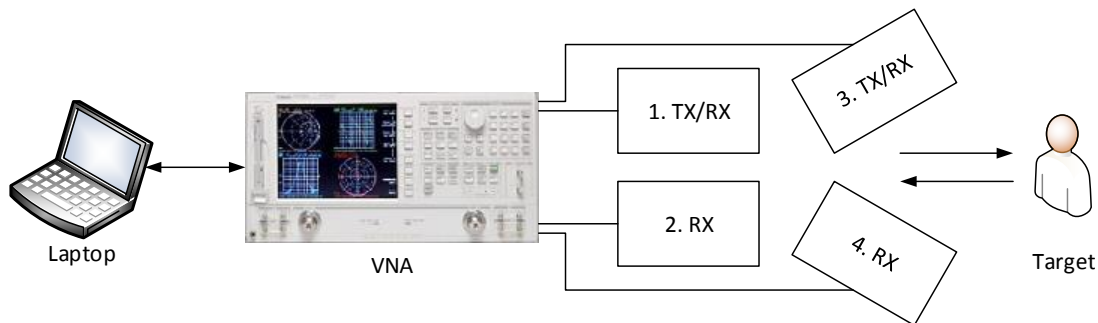


Figure 6.1: System block diagram illustrating the arrangement of the transmitted and receiver horn antennas.

One of the shortcomings of this method is that the optimisation of the ANN architecture has previously been performed using trial and error. This has been done by increasing the number of hidden layers in the ANN and also increasing the number of neurons on each of these layers. A set of validation data was fed into the ANN each time a new layer or neuron was added, and the true positive and false positive rates were recorded. The best architecture was selected by weighing the achieved true positive rate against the cost in false alarm rate. Another issue exists with the training of the ANN, which tends to be inconsistent. This is caused when an initial guess at the weights and biases is taken. As a result of the

randomness of this guess the convergence can be to a local minima rather than the global.

It is of great importance that the false alarm rate is kept low, typically below a few percent, for security screening of large volumes of people. This is due to the action that must be taken once a potential threat has been identified. This action could range from further investigation, e.g. stop and search, to the evacuation of a crowded public area. If the false positive rate goes above a few percent the inconvenience to the security forces and general public would render the method ineffective. Therefore the primary objective in optimising the ANN architecture must be the reduction of the false positive rate.

The second objective in the optimisation of the ANN architecture must be the preservation of the true positive rate for targets of interest. The targets that should be detected by the radar include knives and guns. It is unfortunate that knives and guns are seized by the security forces far too frequently and pose a significant threat to the safety of the general public. The damage that can be caused with these weapons is considerable and these targets are easily concealed upon the human body.

6.2 Encoding the Problem

In order to use evolutionary optimisation to optimise the topology and weights of the ANN classifier for concealed weapon detection, the ANNs topology and weights must be encoded into a real-valued chromosome, which can then be subjected to the various evolutionary operators used in the optimisation process and then decoded for evaluation. Figure 6.2 illustrates the chromosome structure used to store the problem parameters for an ANN with 2 output neurons, a maximum of 2 layers, and 8 input neurons.

HL1.NEURONS	HL2.NEURONS	IL.WEIGHTS	HL1.WEIGHTS
HL2.WEIGHTS	HL1.BIAS	HL2.BIAS	OL.BIAS

Figure 6.2: Encoded chromosome for an ANN consisting of 2 hidden layers (HL), input layer (IL), 2 neurons on the output layer (OL), and associated biases.

Parameter boundaries are also required to restrict the number of hidden layers, neurons per hidden layer, and ranges for the weights and biases within a lower and upper limit. All hidden layers but the last can contain a number of neurons ranging from none to twice the number of input neurons as seen in Equation 6.1, and the last hidden layer must contain a minimum of neurons equal to the number of input neurons as seen in Equation 6.2, this means each candidate network generated by the optimiser must have at least one hidden layer, preventing the generation of benign networks which would waste function evaluations throughout the entire optimisation process. Finally, each weight or bias is restricted to the same boundary shown in Equation 6.3.

$$b(1...(HL - 1)) = \{x \in \mathbb{Z} \mid 0 \leq x \leq 2i\} \quad (6.1)$$

$$b(HL) = \{x \in \mathbb{Z} \mid i \leq x \leq 2i\} \quad (6.2)$$

$$w = \{x \in \mathbb{R} \mid -5 \leq x \leq 5\} \quad (6.3)$$

The algorithm for generating the parameter boundaries used for solutions during the optimisation process is listed in Algorithm 3. The algorithm requires an input of: maximum number of hidden layers, minimum number of neurons per hidden layer, and the number of output neurons.

Algorithm 3 ANN Solution Boundary Algorithm

```

1: function range = annboundary(layers, size, n_out)1
2: lb = [lb; (ones((size × 2) × size, 1) × -5)];
3: for i = 1 : layers do
4:   if i == layers then
5:     lb = [lb; (ones((size × 2) × n_out,1) × -5)];
6:   else
7:     lb = [lb; (ones((size × 2) × (size × 2),1) × -5)];
8:   end if
9: end for
10: lb = [lb; (ones((layers × size × 2) + n_out, 1) × -5)];
11: ub = lb × - 1;
12: lb = [zeros(layers, 1); lb];
13: ub = [(ones(layers, 1) × size × 2); ub];
14: lb(layers - 1) = size - 1;
15: lb(layers) = size;
16: range = [lb ub];

```

For the ANN used in this network, each candidate solution contains 452 variables, with the first 2 defining the number of hidden layers and the number of

¹This is near functional MATLAB 2012a code but may be interpreted as pseudo-code.

¹The function ones(n, m) returns an n-by-m matrix of ones.

neurons on each respectively, the following 128 variables defining the weights for the input layer, 256 for the first hidden layer, and 32 for the final hidden layer.

Regardless of the topology of the candidate solution ANN (which in this case is defined by the first 2 genes of the encoded chromosome) the maximum number of weights and biases will be stored with each chromosome, however not all genotypes will manifest and be expressed as phenotypes as only the weights and biases required to configure the candidate solutions ANN topology will be decoded and used. These unused weights and biases will remain unexpressed in the phenotype until the first two genes allow them to manifest and can go through many generations as dormant genes. This introduces the interesting feature of atavism² into this problem.

At each function evaluation, a chromosome is decoded from its encoded state described in Figure 6.2 and used to instantiate an ANN. This ANN is then used to classify the training data and the objective information is extracted and used to assess the chromosome's fitness based on the ANN result set.

²“Atavism is the tendency to revert to ancestral type. In biology, an atavism is an evolutionary throwback, such as traits reappearing which had disappeared generations before.”

An algorithm for decoding a chromosome conforming to the structure defined in Algorithm 3 has been suggested in Algorithm 4. In this algorithm listing, *top* contains the topology of the ANN, *HL* contains the weights for the hidden layers of the ANN, *bias* contains all the layer biases of the ANN, *IL* contains the weights for the input layer of the ANN, *chrom* contains the real-value encoded chromosome for an ANN, *layers* is the number of hidden layers, *size* is the minimum number of neurons per hidden layer, and *n_out* contains the number of neurons on the output layer.

Algorithm 4 ANN Chromosome Decode Algorithm

```

1: function [top HL bias IL] = anndecode(chrom, layers, size,
    n_out3)
2: top = chrom(1:layers);
3: chrom(1:layers) = [];
4: IL = chrom(1:size × 2 × size);
5: chrom(1:size × 2 × size) = [];
6: for (i = 1 : layers) do
7:   if i == layers then
8:     HL{i} = chrom(1:(size × 2) × n_out);
9:     chrom(1:(size × 2) × n_out) = [];
10:  else
11:    HL{i} = chrom(1:(size × 2)2);
12:    chrom(1:(size × 2)2) = [];
13:  end if
14: end for
15: for (i = 1 : layers) do
16:   bias{i} = chrom(1:size × 2);
17:   chrom(1:size × 2) = [];
18: end for
19: bias{layers + 1} = chrom;
20: top = ceil(top);

```

³This is near functional MATLAB 2012a code but may be interpreted as pseudo-code.

6.3 Comparison to Existing Solution

To assess the performance of the optimisation algorithm when applied to this novel real-world problem, a benchmark must be taken. The performance of the optimised ANN is compared to previously published data [163], the experimental methodology used to obtain this data is explained herein.

The radar used in [163] is a novel multi-polarimetric frequency modulated continuous wave radar operating in the K Band (14-40 GHz). The radar illuminates a person using two linear polarisations which are switched. The use of two illuminating polarisations has been shown to significantly improve detection rates [163]. The reflected signal is then recorded in four polarisations, which are co and cross polar to each of the illuminating polarisations. The gain and phase of the reflected signals are then recorded with respect to the outgoing signal, in the frequency domain. Once collected the gain and phase information is used to generate an array of complex numbers that are then time resolved using an Inverse Fast Fourier Transform (IFFT). Examples of typical radar signals from the human body with and without concealed weapons are given in Figures 6.3 and 6.4.

The radar signals from the body with and without a concealed weapon vary with the aspect in which the person and weapon are presented in the illuminating radar beam. To compensate for this variance many scans must be taken whilst a person is moving in the beam and an accurate representation of the real operating conditions are given to the ANN. The large amounts of data obtained in the collection of training scans makes the problem difficult for the ANN to converge to a solution that is robust but without over training the ANN. In [163] a Principal

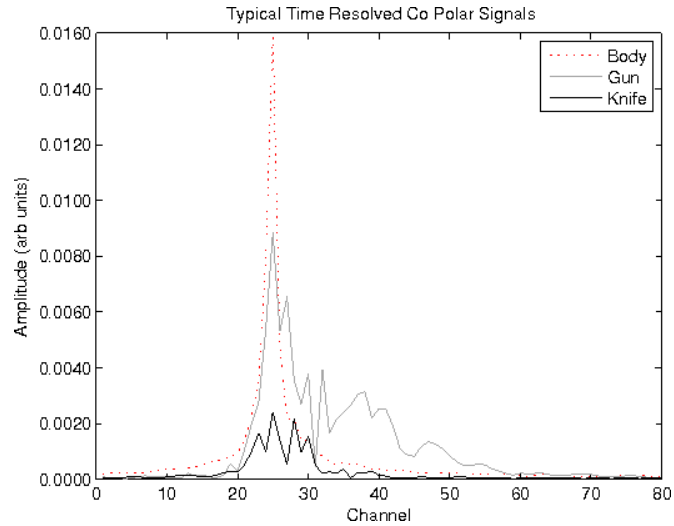


Figure 6.3: Typical co polarised radar signals from a body alone, a body with a concealed gun, and finally a body with a concealed knife.

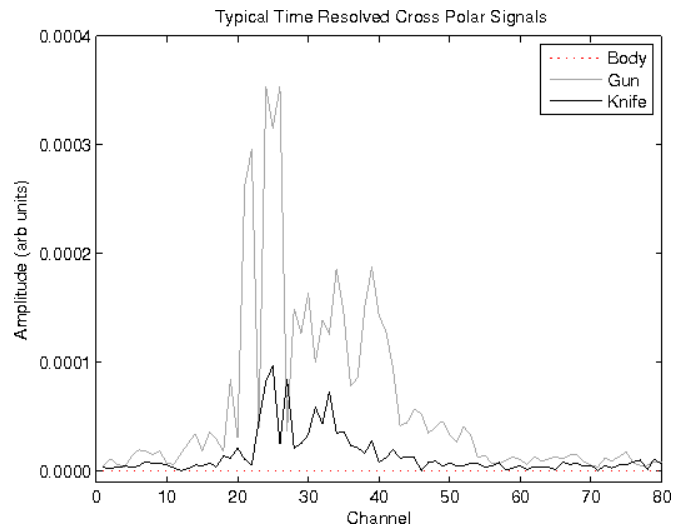


Figure 6.4: Typical cross polarised radar signals from a body alone, a body with a concealed gun, and finally a body with a concealed knife.

Component Analysis (PCA) data reduction technique was integrated into the classification algorithm. The PCA data reduction was applied to the time resolved radar signals to obtain a set of Eigenvalues and Eigenvectors. The Eigenvalues were then used to train the ANN to classify the data into two classes, namely 'threat' and 'non-threat'. A threshold was then applied to the output neuron of the ANN and an alarm was triggered when the output value became larger than the threshold.

In [163] the architecture of the ANN is a 3 layer feed-forward network with as many input neurons as there are significant principal components (typically 5 or 6), the hidden layer has one extra neuron than the input layer and the output layer has a single neuron. The ANN was trained using a constant gradient descent back-propagation method. The training set was constructed of Eigenvalues corresponding to 700 multi-polarimetric radar scans. Of these 700 scans, 100 were taken with a body without a concealed weapon and the other 600 scans with a concealed weapon. There were two different weapons used in the training set, which were a knife and a gun (300 scans of each). The validation of the ANN was performed using a dataset with the same number of scans and same distribution of body with and without a concealed weapon. The validation dataset was constructed using radar scans, which the ANN had not seen *a priori*.

6.3.1 Experiment

This section will describe the encoding of the problem for the optimiser, the optimiser used and its configuration, and the method of performance assessment used when comparing the existing solution to the proposed solution.

The problem has been encoded into a real-valued chromosome using the method described in Section 6.2 with the parameters listed in Table 6.1, the structure of which has been illustrated in Figure 6.5.

Table 6.1: Parameter configurations used for instantiating the ANN.

Parameter	Configuration
Hidden Layers	2
Output Layer Neurons	1
Minimum Hidden Layer Neurons	8
Maximum Hidden Layer Neurons	16
Training Set Size	700
Test Set Size	700

HL1.NEURONS Variables: 1	HL2.NEURONS Variables: 1	IL.WEIGHTS Variables: 128	HL1.WEIGHTS Variables: 256
HL2.WEIGHTS Variables: 16	HL1.BIAS Variables: 16	HL2.BIAS Variables: 16	OL.BIAS Variables: 1

Figure 6.5: Encoded chromosome for the two-objective ANN consisting of 2 hidden layers (HL), input layer (IL), 1 neuron on the output layer (OL), and associated biases, totalling to 435 variables.

The proposed solution uses an implementation of MOEA/D-DRA enabled with the ability to express Decision Maker (DM) preferences named WZ-MOEA/D-DRA, the algorithm itself is described and benchmarked in Chapter 5. WZ-MOEA/D-DRA was selected to allow the optimisation process to search within a Region of Interest (ROI), which will be above a true positive rate of 0.8 and

below false positive rate of 0.04. This ROI was selected from the existing solution's performance, this will allow the algorithm to optimise toward and possibly beyond a known benchmark. MOEA/D-DRA specific configurations have been taken from [93], and those configurations which differ are listed in Table 6.2.

Table 6.2: Parameter configurations used for WZ-MOEA/D-DRA.

Parameter	Configuration
μ Population	300
Number of Variables	435
Number of Objectives	2
Niche	60
Maximum Update Number	6
Z_{thresh}	50
Maximum Function Evaluations	500
Region of Interest	0.8, 0.04

The hypervolume indicator described in Section 2.10.1 will be used as the method of performance assessment, this will be conducted similar to the performance assessment in Section 5.4.2, where the ROI is used as the reference point that is provided to the hypervolume indicator allowing for performance assessment based on how much of the search space within the ROI has been covered.

6.3.2 Results

WZ-MOEA/D-DRA has been used to optimise the classifier parameters for the classification of concealed weapon detection with two objectives: true positives and false positives. The final population produced by the optimiser has been plotted in Figure 6.6, where it can be seen that a number of trade-off solutions have been found with the ROI, resulting in a Pareto-optimal approximation set.

When compared to the benchmark plot it can be observed that it is Pareto-dominated by every candidate solution in the final population, this means that the optimiser has found a diverse set of solutions which are all an improvement on the existing solution. The same candidate solutions have also been plotted in Figure 6.7 without conversion of all objectives for minimisation, this plot is what was presented to the DM when selection of a successful candidate solution was required.

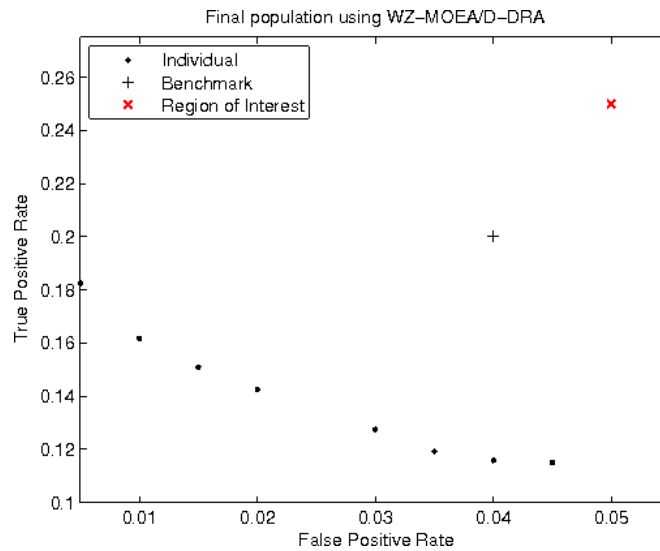


Figure 6.6: Population generated by MOEA/D-DRA combined with Weighted Z-score preference articulation after 500 function evaluations for the two-objective concealed weapons detection classifier.

The plot in Figure 6.7 shows a number of optimised solutions. One of these solutions must be selected, by the DM, to be applied in a concealed weapon detection radar. As can be seen from the distribution of the solutions in Figure 6.7 each solution offers a distinctly different true positive and false positive rate. This allows the DM to use these solutions in a manner that will give control over the behaviour of the weapon detector. This will enable the DM to choose a

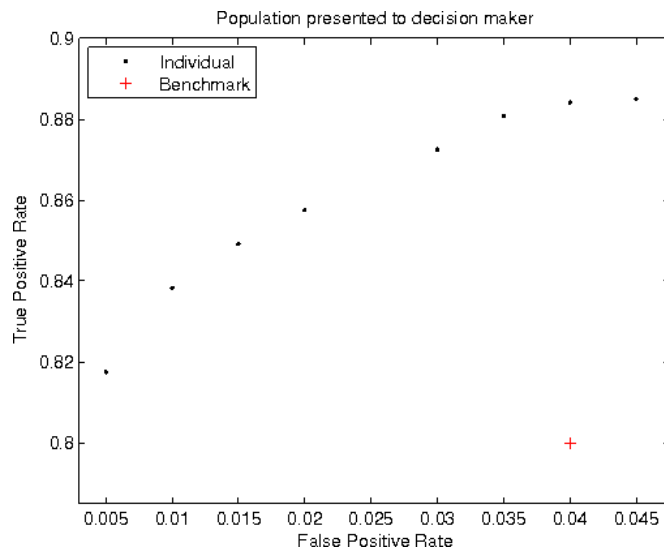


Figure 6.7: Population of candidate solutions presented to the DM, for selection to be made based on expert knowledge.

solution that increases the system's sensitivity when there is a heightened threat to security. Equally the false alarm rate could be reduced by choosing another solution.

The benchmark shown in Figure 6.7 was taken from [163], this solution offers a true positive rate of 0.8 at a cost in false positive rate of 0.04. Given the requirement for false alarm reduction which was identified as the primary objective earlier, the solution from Figure 6.7 that was chosen by the DM was the solution offering a true positive rate of 0.82 and a false positive rate of 0.005. The chosen solution has provided a false alarm reduction of 0.035 at the same time as increasing the true positive rate by 0.02. To put this into perspective previously 40 people in one thousand would have been wrongly accused of carrying a concealed weapon whereas the number of wrongly accused with the new optimised solution would be less than one in a thousand. Not only does this reduce the inconvenience to the security forces and general public, it builds confidence in

the ability and robustness of the weapon detection system.

6.4 Detection and Classification of Multiple Types of Threat

A weapon detection system that is capable of classifying a detected threat into target groups would be an extremely valuable tool to security forces. Such a system would enable the authorities to react to a detection in a controlled and proportional manner. The action that must be taken to confront an individual concealing a threat object depends very much on the threat object itself. An individual carrying a knife could be dealt with easier than an individual with an improvised explosive device.

Besides obvious threat objects such as guns and knives, the detection of objects such as mobile phones is desirable as they can be seen as a threat in the case of a controlled courtroom (where photographing witnesses and communicating with witnesses waiting to testify is an issue) where they are banned and when entering a controlled site.

The extent of the ROI will be determined by the DM based on some pre-determined criteria, for example the radar may be deployed in an environment where the client has specified a minimum detection rate and maximum false alarm rate. It is possible that no solutions may be found within an ROI which has been confined based on a client's specifications, in which case the specification would be deemed beyond the performance of the radar and another solution would be required.

As the number of targets of interest increases so does the risk of increasing the number of conflicting objectives. The extent of which this method can handle conflicting objectives will be explored in more detail in subsequent publications.

It should be noted that the radar beam is confined to a spot size which is commensurate in size with the weapons of interest, therefore it is infeasible to measure more than one weapon in a given location on the body. The authors suggest that the radar operator would scan the radar beam over the person and would be able to find multiple weapons concealed on different parts of the body and then address the situation based on the most severe threat detected.

To investigate whether the developed radar based weapon detection system is capable of classifying the reflected signal into target groups, two experiments have been conducted. In the first experiment, radar signals from a body without a concealed weapon, a body with a concealed knife, and a body with a concealed gun have been recorded by the radar. In the second experiment, radar signals from a body without a concealed weapon, a body with a concealed knife, a body with a concealed gun, and a body with a concealed mobile phone device have been recorded by the radar.

Again, the radar signals were recorded in the frequency domain with both amplitude and phase relative to the illuminating beam measured. The amplitude and phase were used to generate an array of complex numbers that were temporally resolved using an IFFT, the resulting signal was then reduced using PCA as described earlier. For the first experiment the training set was constructed of 300 sets of Eigenvalues corresponding to the radar scans, 100 scans of each target scenario were taken. That is 100 scans from body alone, 100 with a concealed knife, and 100 with a concealed gun. For the second experiment the training

set was constructed of 500 sets of Eigenvalues correspond to the radar scans, and the addition of 100 scans of a body with a concealed mobile phone device. The validation set was constructed of the same number of scans and the same distribution of targets. The scans used to construct the validation set had not been seen by the ANN *a priori*. The architecture (in terms of the number of layers and number of neurons on each layer) of the ANN was determined by the optimisation algorithm and the weights and biases were also determined by the optimisation algorithm.

To assess the statistical significance of these results the area under the observed Receiver Operating Characteristic (ROC) curve (as observed in [163]) was used alongside the chosen sample sizes to calculate the standard error.

$$\begin{aligned}\theta_1 &= 0.8362 \\ n_A &= 200 \\ n_N &= 100\end{aligned}\tag{6.4}$$

θ_1 is the area under the curve for the observed ROC, n_A is the number of scans with a weapon, and n_N is the number of scans without a weapon. The standard error is calculated as follows:

$$SE = \sqrt{\frac{\theta_1(1 - \theta_1) + (n_A - 1)(Q_1 - \theta_1^2) + (n_N - 1)(Q_2 - \theta_1^2)}{n_A n_N}}\tag{6.5}$$

Where Q_1 and Q_2 are calculated using the following equations:

$$Q_1 = \frac{\theta_1}{2 - \theta_1}\tag{6.6}$$

$$Q_2 = \frac{2\theta_1^2}{1 + \theta_1}\tag{6.7}$$

The standard error was calculated to be 0.0225, this shows a high level of certainty that the results are statistically significant. It can be shown through testing a null hypothesis (that the results happened by chance) that the sample size chosen gives a high level of confidence in the significance of these results. To do this the area under the curve is set at 0.5, this represents a special case of a ROC which is by chance and has no useful classification abilities. The difference between the area under the observed ROC and the area under the curve for the null hypothesis is divided by the standard error to give a z -score. A z -score of 1.645 relates to a 5% one-sided test of significance and a 95% power.

$$z = \frac{\theta_1 - \theta_2}{SE} \quad (6.8)$$

$$\theta_2 = 0.5 \quad (6.9)$$

The value of the z -score calculated for this test was 14.9, this shows that the sample sizes chosen were much larger than required and thus result in a high level of statistical significance.

6.4.1 Experiment

This section will describe the encoding of the problems for the optimiser, the optimisers used and their configuration, and the method of performance assessment used when comparing the final solutions. The problems have been encoded into real-valued chromosomes using the method described in Section 6.2 with the parameters listed in Table 6.3, the structures of which have been illustrated in Figures 6.8 and 6.9.

HL1.NEURONS Variables: 1	HL2.NEURONS Variables: 1	HL3.NEURONS Variables: 1	IL.WEIGHTS Variables: 128
HL1.WEIGHTS Variables: 256	HL2.WEIGHTS Variables: 256	HL3.WEIGHTS Variables: 48	HL1.BIAS Variables: 16
HL2.BIAS Variables: 16	HL3.BIAS Variables: 16	OL.BIAS Variables: 3	

Figure 6.8: Encoded chromosome for the five-objective ANN consisting of 3 hidden layers (HL), input layer (IL), 3 neurons on the output layer (OL), and associated biases, totalling to 742 variables.

HL1.NEURONS Variables: 1	HL2.NEURONS Variables: 1	HL3.NEURONS Variables: 1	HL4.NEURONS Variables: 1
IL.WEIGHTS Variables: 128	HL1.WEIGHTS Variables: 256	HL2.WEIGHTS Variables: 256	HL3.WEIGHTS Variables: 256
HL4.WEIGHTS Variables: 64	HL1.BIAS Variables: 16	HL2.BIAS Variables: 16	HL3.BIAS Variables: 16
HL4.BIAS Variables: 16	OL.BIAS Variables: 4		

Figure 6.9: Encoded chromosome for the seven-objective ANN consisting of 4 hidden layers (HL), input layer (IL), 4 neurons on the output layer (OL), and associated biases, totalling to 1032 variables.

Table 6.3: Parameter configurations used for instantiating the ANN.

Parameter	5D Configuration	7D Configuration
Maximum Hidden Layers	3	4
Output Layer Neurons	3	4
Minimum Hidden Layer Neurons	8	8
Maximum Hidden Layer Neurons	16	16
Training Set Size	300	400
Test Set Size	300	400

The proposed solution in the first experiment uses an implementation of MOEA/D-DRA enabled with the ability to express DM preferences named WZ-MOEA/D-DRA, the algorithm itself is described and benchmarked in Chapter 5. WZ-MOEA/D-DRA was selected to allow the optimisation process to search within a ROI with the preferences defined as:

- True positive rate on classification of bodies at or above 95%.
- True positive rate on classification of guns at or above 60%.
- True positive rate on classification of knives at or above 50%.
- False positive rate on classification of guns at or below 5%.
- False positive rate on classification of knives at or below 5%.

This ROI was selected by the DM based on the performance of the two-objective results, this will allow the algorithm to optimise toward and possibly beyond an unknown goal. The MOEA/D-DRA specific configurations have been taken from [93], and those configurations which differ are listed in Table 6.4.

Table 6.4: Parameter configurations used for WZ-MOEA/D-DRA.

Parameter	Configuration
μ Population	100
Number of Variables	742
Number of Objectives	5
Niche	30
Maximum Update Number	3
Z_{thresh}	50
Maximum Function Evaluations	1000
Region of Interest	0.05, 0.4, 0.5, 0.05, 0.05

The proposed solution in the second experiment uses an implementation of CMA-PAES-II enabled with the ability to express DM preferences named WZ-CMA-PAES. The algorithm itself is described and benchmarked in Chapters 4 and 5 where it is shown to perform well on problems containing many objectives, this is a requirement for this problem as it consists of seven problem objectives. WZ-CMA-PAES was selected to allow the optimisation process to search within a ROI with the preferences defined as:

- True positive rate on classification of bodies at or above 95%.
- True positive rate on classification of guns at or above 60%.
- True positive rate on classification of knives at or above 50%.
- True positive rate on classification of mobile phones at or above 50%.
- False positive rate on classification of guns at or below 5%.
- False positive rate on classification of knives at or below 5%.

- False positive rate on classification of mobile phones at or below 5%.

This ROI was selected by the DM based on the performance of the two-objective results, this will allow the algorithm to optimise toward and possibly beyond an unknown goal. The MOEA/D-DRA specific configurations have been taken from [93], and those configurations which differ are listed in Table 6.4.

Additionally, this problem regarding concealed weapon detection is in the continuous domain, meaning that WZ-CMA-PAES is advantageous as it uses Covariance Matrix Adaptation (CMA) from the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) for its variation. This is because it has been shown in [56] that the No Free Lunch (NFL) theorem does not hold in continuous search domains, meaning CMA-ES which performs extremely well in the continuous domain has an advantage over other algorithms. The configurations for WZ-CMA-PAES have been listed in Table 6.5.

Table 6.5: Parameter configurations used for WZ-CMA-PAES.

Parameter	Configuration
Archive Capacity	50
Grid Divisions	3
μ Population	10
Θ_{max}	50
Θ_{min}	10
Number of Variables	1032
Number of Objectives	7
Z_{thresh}	50
Maximum Function Evaluations	1000
Region of Interest	0.05, 0.4, 0.5, 0.5, 0.05, 0.05, 0.05

The hypervolume indicator described in Section 2.10.1 will be used as the method of performance assessment, this will be conducted similar to the performance assessment in Section 5.4.2 where the ROI is used as the reference point that is provided to the hypervolume indicator allowing performance assessment based on how much of the search space within the ROI has been covered.

The contributing hypervolume indicator described in Section 2.5.2 is used to reduce the size of the final approximation set produced by the optimiser to a size that will not overwhelm and confuse the DM.

6.4.2 Results

This section is divided into two parts. The first part presents and discusses the results for the five-objective problem which concerns the optimisation of a classifier aiming to categorise signals into bodies, guns, and knives. The second part presents and discusses the results for the seven-objective problem which concerns the optimisation of a classifier aiming to categorise signals into bodies, guns, knives, and mobile phones.

Five-Objective Problem

The population of solutions from the WZ-MOEA/D-DRA optimisation results are plotted in Figure 6.10, this plot shows the solutions for the five-objective problem. The five objectives are split into true positives for the body, gun, and knife, and the false positives for the gun and knife. In this problem the reduction of both the false positive rates are the two main objectives, these should be weighted equally. The remaining objectives are the maximization of the true positive rates for each target included in the training and validation sets. Also presented is a colour-map, Figure 6.11, this was used to aid the DM in choosing a solution from the population. In each of the presented plots there are five candidate solutions, one of which should be chosen by the DM to be implemented in the weapon detection system.

The colour-map was found to be useful tool in the selection of a solution from the final candidate population. When a population of 5 solutions was presented to the DM in the minimisation format presented in Figure 6.10, it was difficult to visualise the different trade-offs between candidate solutions. The visualisation of the trade-off between solutions is much clearer in the colour-map in Figure

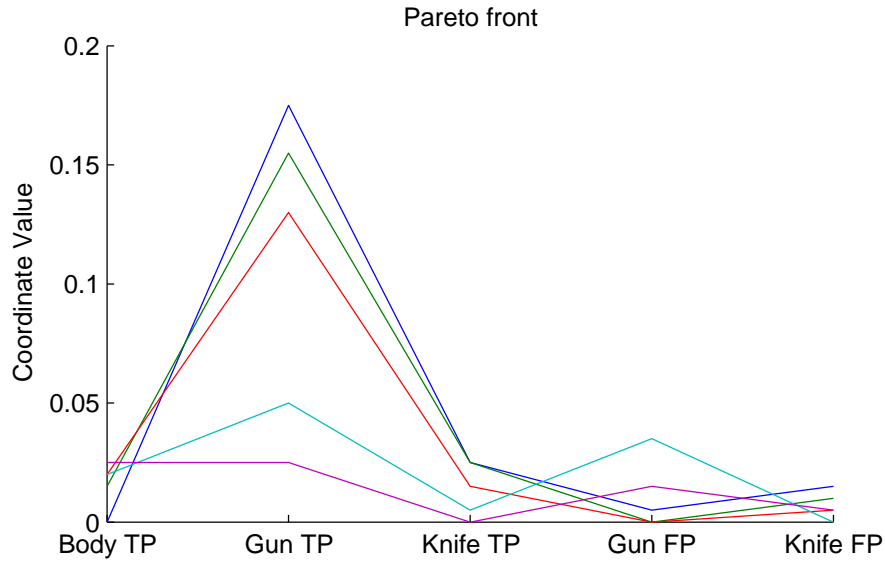


Figure 6.10: Parallel-coordinates plot illustrating objective value results for five-objective threat detection.

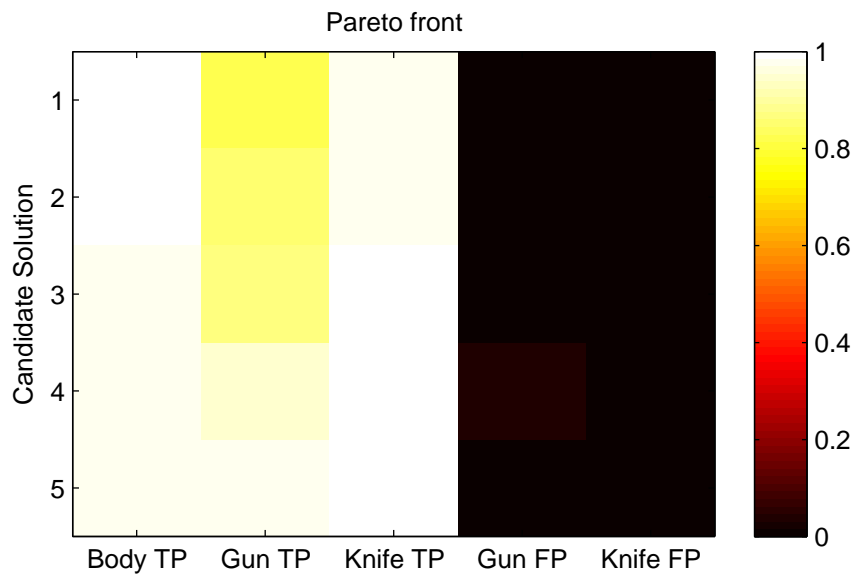


Figure 6.11: Colour-map illustrating objective value results for five-objective threat detection, for use by the DM.

6.11. Figure 6.11 shows that all solutions provide a means to reliably classify the signals into target classes which are body, gun, and knife; where the true positive rates are high and the false positive rates are low for all target classes. This can also be seen in the minimisation format presented in Figure 6.10. In Figure 6.11 it is clear to see that there are subtle differences in the trade-offs between the five objectives in the final candidate population. As the false positive rates for both target classes are effectively equal for all candidate solutions, the solution taken by the DM was chosen as the solution with near equal true positive rate for all targets. This solution is candidate solution number 5 in Figure 6.11.

Seven-Objective Problem

The population of solutions from the WZ-CMA-PAES optimisation are plotted in Figure 6.12, this plot shows the solutions for the seven-objective problem. The seven objectives are split into true positives for the body, gun, knife, mobile phone and the false positives for the gun, knife, and mobile. In this problem the reduction of false positive rates are the three main objectives, these should be weighted equally. The remaining objectives are the maximization of the true positive rates for each target included in the training and validation sets. Also presented is a colour-map, Figure 6.13, this was used to aid the DM in choosing a solution from the population. In each of the presented plots there are 7 candidate solutions, one of which should be chosen by the DM to be implemented in the weapon detection system.

Figure 6.13 shows that all solutions provide a means to reliably classify the signals into target classes which are body, gun, knife, and mobile phone; where the true positive rates are high and the false positive rates are low for all target

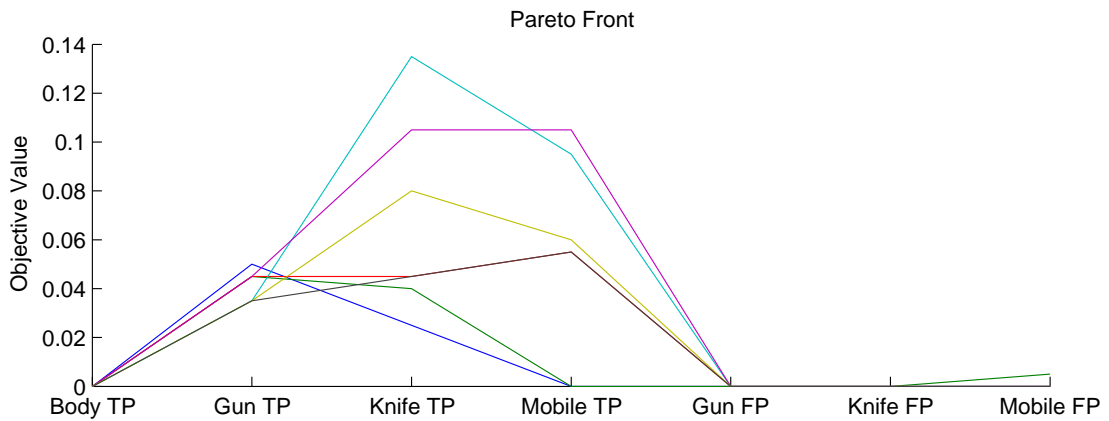


Figure 6.12: Parallel-coordinates plot illustrating objective value results for seven-objective threat detection.

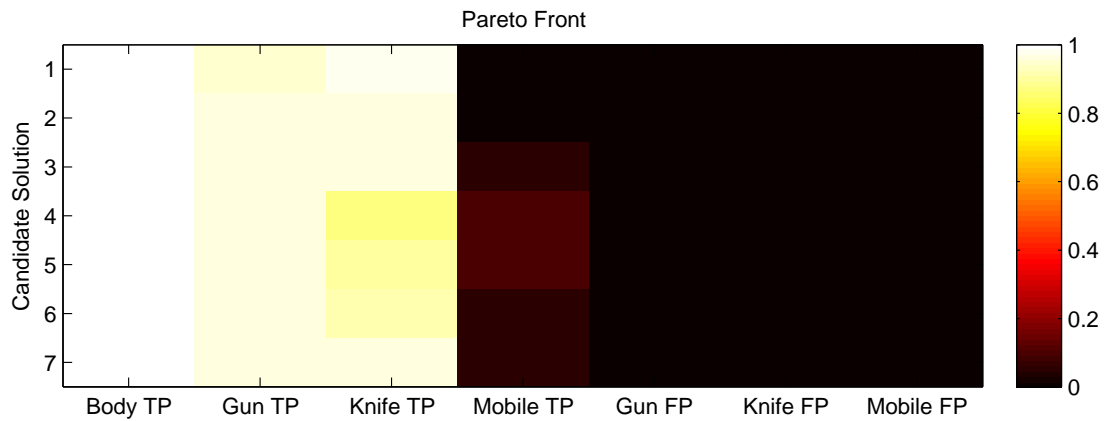


Figure 6.13: Colour-map illustrating objective value results for seven-objective threat detection, for use by the DM.

classes. This can also be seen in the minimisation format presented in Figure 6.12. In Figure 6.13 it is clear to see that there are subtle differences in the trade-offs between the seven objectives in the final candidate population. As the false positive rates for the three target classes are effectively equal for all candidate solutions, the solution taken by the DM was chosen as the solution with near equal true positive rate for all targets. This solution is candidate solution number 2 in Figure 6.13.

6.4.3 Conclusion

In this section, preference driven Evolutionary Multi-Objective Optimisation (EMO) has been used to design classifiers that are capable of classifying signals into many categories of threat. These classifiers were presented to the DM in the form of a colour-map in order for the preferred solution to be selected using expert knowledge.

The method used for encoding the problem into a chromosome proved feasible and applicable to both the five-objective and seven-objective problem, and both WZ-MOEA/D-DRA and WZ-CMA-PAES were able to produce solutions that satisfied the DM's ROI within the function evaluation budget defined by the experiment.

6.5 Conclusion

In this chapter two preference driven EMO (WZ-MOEA/D-DRA and WZ-CMA-PAES) have successfully optimised the design of three signal classifiers used for concealed weapon detection. By using preference articulation, the final approximation set is more pertinent and the computational cost of the optimisation process has been reduced by requiring less function evaluations and searching within an expressed ROI, this has been shown in Chapter 5.

The application of the WZ-MOEA/D-DRA and WZ-CMA-PAES optimisation algorithms to the training and optimisation of the topology of an ANN intended for use in concealed weapon detection has been presented. The performance of the optimised ANN has been benchmarked against previously published data, in which an ANN had been trained using back-propagation and its topology

determined by trial-and-error. The optimisation has been shown to provide the DM with a number of solutions (trained ANNs) that all have independent trade-offs which are equally distributed across the Pareto-optimal front. The DM then selected an optimised solution which provided a reduction in false alarm rate and an increase in detection rate.

This evolutionary method showed promising results in Section 6.3 where the proposed solution was compared to an existing published solution. The comparison resulted in the evolutionary method producing better solutions and justifying further experiments in Section 6.4, where a more difficult classifier which is able to classify signals into multiple threat categories was designed and tested with results satisfying the DM's preferences.

Although the results have shown that the weapon detection radar can achieve high detection rates with very little cost in terms of false alarm rates, it should be noted that these values only apply in the scenario used to collect the data within this thesis. It is anticipated that the system performance will fall away and fewer detections will be made in other scenarios, for example if the person being screened is non-co-operative then the operator may be unable to make a full scan of the person and therefore a concealed weapon may go undetected. The application of radar to weapon detection is novel and therefore all operational procedures and system performance have not yet been evaluated, this will be subject to further research.

Overall the evolutionary optimisation of classifiers used for concealed weapon detection showed an increase in performance when compared to the existing solution, and proves a feasible method of optimising topology, weights and biases of ANNs. The use of an EMO algorithm for the design of such a classifier will save

time that would be spent using a trial-and-error manual hill-climbing method, and the marginal performance gains that an evolutionary method can provide may end up detecting more threats to security.

Chapter 7

Conclusion

In this thesis the incorporation of Decision Maker (DM) preferences into Evolutionary Multi-Objective Optimisation (EMO) search methods has been explored. This has been achieved through the development and statistically verified benchmarking of an extensible modular EMO framework, the Covariance Matrix Adaptation Pareto Archived Evolution Strategy (CMA-PAES), with the incorporation of the novel Weighted Z-score (WZ) preference articulation operator. This WZ driven algorithm (WZ-CMA-PAES) is then benchmarked against the WZ driven Multi-Objective Evolutionary Algorithm Based on Decomposition with Dynamical Resource Allocation (MOEA/D-DRA) variant (WZ-MOEA/D-DRA) before it is applied to solving a real-world concealed weapon detection optimisation problem.

This chapter is structured with an overview of the main findings in Section 7.1, a summary of the contributions of this research in Section 7.2, and suggestions of future research directions in Section 7.3.

7.1 Main Findings

A clear outline of the main findings in this thesis is presented in the following:

- Development of CMA-PAES, a fast converging EMO algorithm.
- Development of m-CMA-PAES, an EMO algorithm tailored for problems with complex Pareto sets.
- Development of CMA-PAES-II, an algorithm specifically tailored towards many-objective problems.
- A new method of preference articulation (the Weighted Z-score operator), to involve the DM in any EMO process.
- Successful optimisation of a classifier used for concealed weapon detection, which outperforms an existing and published classifier.

Through these findings, it has been shown that:

- Using an AGA in place of the contributing hypervolume indicator can produce comparable results to MO-CMA-ES.
- Investing a percentage of the function evaluation budget in dominated solutions can prevent discarding potentially valuable genetic material.
- CMA can be used in many-objective spaces and can outperform state-of-the-art approaches.
- The contributing hypervolume indicator can be feasibly incorporated into many-objective optimisation, when used as a narrow-phase sorting criterion.

-
- Approaches such as IBC can keep the evolutionary process on track and improve the indicator quality of the final approximation set.
 - Methods for incorporating decision maker preferences do not have to be dependant on existing search operators.
 - Methods for incorporating decision maker preferences may appear to perform well on multiple objectives, for example 2 or 3, but unexpected optimisation difficulties may occur when moving to many-objective space.
 - Optimisers with and without a non-dominated approach can easily incorporate the same portable preference articulation operator.
 - the Hypervolume Indicator can be used as a performance metric for assessing an optimisers approximation set in regards to its pertinence. This is achieved by using the goals or preferences as the reference point.
 - EMO can be applied to the field of concealed weapon detection and produce better performing solutions to a state-of-the-art and current problem, which determines peoples safety.
 - Preference driven many-objective optimisers can solve problems which were before seen as too difficult. For example the parameter settings and topology for a multi-output neuron ANN.
 - Preference driven many-objective optimisers can produce specialist classifiers by focussing on regions of interest and exploring classification trade-offs.

7.2 Summary of Contributions

This thesis entitled “Preference Focussed Many-Objective Evolutionary Computation” aims to investigate the incorporation of DM preferences into EMO search methods in order to solve real-world problems, so as to improve the quality of the final solutions produced by the optimisation process. In order to achieve this aim, a number of research objectives were defined and accomplished throughout the duration of this research.

A critical review of the field of evolutionary computation with particular emphasis on using evolutionary computation methods to solve multi-objective problems was conducted in Chapter 2. The review of the literature identified Covariance Matrix Adaptation (CMA) as a powerful technique for the variation of solutions, which had been originally implemented within a single-objective evolution strategy named the Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES). The multi-objective implementation of CMA-ES, named the Multi-Objective Covariance Matrix Adaptation Evolution Strategy (MO-CMA-ES), unfortunately offered poor performance in terms of computational efficiency on problems consisting of four or more objectives. The cause of this computational inefficiency on many-objective problems was identified to be a result of the algorithm’s complete reliance on the contributing hypervolume indicator (described in Section 2.5.2) as a sorting criterion during the selection process. During the sorting for selection stage of MO-CMA-ES, the population at the current generation is sorted into ranks using the non-dominated sorting algorithm, and then sorted at a second level using the contributing hypervolume indicator. The computational effort of this scheme increases exponentially throughout the optimisation process,

as the number of dominated solutions in the population reduces, and only non-dominated solutions are produced. This results in all solutions in the population being assigned the same rank and therefore all solutions being subjected to the contributing hypervolume indicator at once.

The computationally lightweight EMO algorithm, named the Pareto Archived Evolution Strategy (PAES), appeared to offer a simple structure, intuitive design, and computationally efficient method of diversity preservation through the adaptive grid technique. The attractive simplicity and effectiveness of PAES, and the powerful variation and convergence offered by CMA set the premise for developing a CMA driven EMO algorithm capable of feasibly optimising problems consisting of many objectives. Through the EMO literature a number of performance assessment methods were also identified, allowing the performance comparison of EMO algorithms and statistical verification of their results.

Chapter 3 concerns the development of a new EMO algorithm, named CMA-PAES. The design of the algorithm consists of a PAES-like structure, adaptive grid inspired diversity preservation method, and CMA driven variation. The ambition of this design was to exploit the powerful CMA variation technique in a multi-objective implementation, without the infeasibility of optimising problems consisting of many objectives which MO-CMA-ES suffers from. CMA-PAES is shown to perform comparably with MO-CMA-ES (and to outperform NSGA-II and PAES in [15]) on the two-objective ZDT synthetic test suite. A multi-tier variant of CMA-PAES, named the Multi-tier Covariance Matrix Adaptation Pareto Archived Evolution Strategy (m-CMA-PAES), is developed which employs the contributing hypervolume indicator at grid-level and is shown to outperform MO-CMA-ES on all but 3 of 22 synthetic test functions from the ZDT, DTLZ, and

CEC09 synthetic test suites. m-CMA-PAES outperforms MO-CMA-ES on these synthetic test functions in the presence of their intended optimisation difficulties and complex Pareto-optimal sets on up to three objectives. No more than three objectives were considered due to the infeasibility of MO-CMA-ES on problems consisting of many objectives.

With the confidence that CMA-PAES is an EMO algorithm comparable to MO-CMA-ES in performance but without the infeasibility of optimisation in the presence of many objectives (due to the reliance on population-wide subsection to the contribution hypervolume indicator), Chapter 4 aims to assess CMA-PAES on synthetic test problems consisting of many objectives. Design elements from CMA-PAES and m-CMA-PAES were used to implement a new EMO algorithm, named the Covariance Matrix Adaptation Pareto Archived Evolution Strategy II (CMA-PAES-II), which is enhanced with new features intended for robustness on many-objective optimisation problems, such as Indicator Based Conformation (IBC), sigma restart, and an improved Adaptive Grid Algorithm (AGA). CMA-PAES-II is shown to perform comparably to MOEA/D-DRA on the two-objective configuration of the scalable WFG synthetic test suite, and is shown to significantly outperform MOEA/D-DRA on the three-objective, five-objective, seven-objective, and ten-objective configurations of the same test suite.

A novel method of preference articulation has been developed, in order to improve the pertinence of an approximation set produced by an EMO algorithm. This novel method increases the rate of an EMO algorithm's convergence by directing the search towards a Region of Interest (ROI) expressed by a DM, without wasting computational effort exploring regions of undesirable objective space. Chapter 5 introduces the two-phase WZ preference articulation operator,

a portable operator capable of focussing the optimisation process toward a ROI and producing pertinent approximation sets using preferences expressed by a DM. The WZ preference articulation operator has been incorporated into CMA-PAES-II and MOEA/D-DRA and has shown in Chapter 5 to provide significant performance enhancement in the optimisation process in the presence of DM preferences. The results indicate WZ-CMA-PAES provides overall better performance, suggesting it is more suited to the incorporation of portable operators due to its extensible design and many-objective optimisation enhancements.

The promising results from the performance assessment and statistical analysis of these algorithms, suggest that CMA-PAES-II and the WZ preference articulation operator offer robust preference driven many-objective optimisation on problems containing a range of optimisation difficulties. In Chapter 6, WZ-CMA-PAES is applied to an optimisation problem regarding the optimisation of classifiers intended for concealed weapon detection. The results show that WZ-CMA-PAES produced an approximation set of non-dominated solutions which outperform the previously published [163] ANN solution. WZ-CMA-PAES is then used to optimise many-objective concealed weapon detection problems and successfully produces classifiers which are able to categorise radar signals into categories of threat (e.g. gun, knife, explosive), rather than simply threat or non-threat. The success of this application suggests that WZ-CMA-PAES is a preference driven EMO algorithm capable of the optimisation of real-world problems, and also demonstrates that security forces can benefit from the optimisation of their systems in order to increase the safety and security of the area in which it is implemented.

7.3 Recommendations for Future Research

There are many opportunities for future research directions in the field of Evolutionary Computation, and its application to concealed weapon detection. Some research directions which are relevant to this thesis are listed in the following:

- There is an opportunity for further work in using the WZ preference articulation operator in combination with an offline archive to explore the effects of changing preferences progressively during the optimisation process and using the offline archive to search for solutions matching the new preferences, to identify whether this type of combination would speed up search in the presence of shifting preferences, without the need for re-starts.
- There is potential for incorporating the WZ preference articulation operator into many state of the art EMO algorithms in a performance comparison study in order to identify which algorithm is best suited for preference articulation and what mode of incorporation is the most beneficial.
- There exists a requirement for the development of a multi-objective test suite targeting the performance assessment of the ability for an optimiser to incorporate DM preferences, this would also provide a standard for future comparison and assessment of preference driven optimisers.
- Although the results have shown that the weapon detection radar can achieve high detection rates with very little cost in terms of false alarm rates, it should be noted that these values only apply in the scenario used to collect the data published within in this thesis. It is anticipated that the system performance will fall away and fewer detections will be made in other

scenarios, for example if the person being screened is non-co-operative then the operator may be unable to make a full scan of the person and therefore a concealed weapon may go undetected. The application of radar to weapon detection is novel and therefore all operational procedures and system performance have not yet been evaluated, this will be subject to further research.

- The optimisation of classifiers for concealed weapon detection in Chapter 6 is a novel application, further work is recommended to explore the application of preference focussed EMO in the real-time execution of a concealed weapon detection security system. The premise of the recommendation is that the WZ preference articulation operator can be incorporated as a progressive preference articulation method, and with every radar signal that is scanned by the concealed weapon detection system, the training, test, and verification sets can be expanded upon. This will create a change in the calculation of the objective function, therefore allowing for the real-time evolution of a classifier for concealed weapon detection. A real-time EMO for the optimisation of a classifier for concealed weapon detection, with the ability to change preferences of the system in real-time (e.g. assign greater preference in lower false alarm rates) is a challenging and novel research direction.
- The Indicator Based Conformation mechanism presented in Section 4.2 shows promising results in preventing CMA-PAES-II from succumbing to local-optima or “stagnating” during the optimisation process. There is an opportunity for the exploration of the incorporation of the IBC mechanism

into other EMO algorithms and the use of other indicators in the mechanism.

References

- [1] Z. Kowalczyk, P. Suchomski, and T. Bialaszewski, “Evolutionary multi-objective pareto optimisation of diagnostic state observers,” *Applied Mathematics and Computer Science*, vol. 9, pp. 689–710, 1999.
- [2] C. Vlachos, D. Williams, and J. Gomm, “Genetic approach to decentralised pi controller tuning for multivariable processes,” *IEE Proceedings-Control Theory and Applications*, vol. 146, no. 1, pp. 58–64, 1999.
- [3] P. Schroder, B. Green, N. Grum, and P. Fleming, “On-line evolution of robust control systems: an industrial active magnetic bearing application,” *Control Engineering Practice*, vol. 9, no. 1, pp. 37–49, 2001.
- [4] Y. Wang, B. Li, and T. Weise, “Two-stage ensemble memetic algorithm: Function optimization and digital iir filter design,” *Information Sciences*, vol. 220, pp. 408–424, 2013.
- [5] E. Besada-Portas, L. De La Torre, A. Moreno, and J. L. Risco-Martín, “On the performance comparison of multi-objective evolutionary uav path planners,” *Information Sciences*, vol. 238, pp. 111–125, 2013.

-
- [6] C. A. C. Coello *et al.*, “A comprehensive survey of evolutionary-based multiobjective optimization techniques,” *Knowledge and Information systems*, vol. 1, no. 3, pp. 129–156, 1999.
- [7] S. F. Adra, *Improving convergence, diversity and pertinency in multiobjective optimisation*. PhD thesis, University of Sheffield, Department of Automatic Control and Systems Engineering, 2008.
- [8] E. J. Hughes, “Evolutionary many-objective optimisation: many once or one many?,” in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 1, pp. 222–227, IEEE, 2005.
- [9] E. J. Hughes, “Msops-ii: A general-purpose many-objective optimiser,” in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pp. 3944–3951, IEEE, 2007.
- [10] H. Ishibuchi, N. Tsukamoto, Y. Hitotsuyanagi, and Y. Nojima, “Effectiveness of scalability improvement attempts on the performance of nsga-ii for many-objective problems,” in *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pp. 649–656, ACM, 2008.
- [11] A. Jaszkiewicz, “On the computational efficiency of multiple objective metaheuristics. the knapsack problem case study,” *European Journal of Operational Research*, vol. 158, no. 2, pp. 418–433, 2004.
- [12] V. Khare, X. Yao, and K. Deb, “Performance scaling of multi-objective evolutionary algorithms,” in *Evolutionary Multi-Criterion Optimization*, pp. 376–390, Springer, 2003.

-
- [13] R. C. Purshouse and P. J. Fleming, “On the evolutionary optimization of many conflicting objectives,” *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 770–784, 2007.
- [14] X. Zou, Y. Chen, M. Liu, and L. Kang, “A new evolutionary algorithm for solving many-objective optimization problems,” *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 38, no. 5, pp. 1402–1412, 2008.
- [15] S. Rostami and A. Shenfield, “Cma-paes: Pareto archived evolution strategy using covariance matrix adaptation for multi-objective optimisation,” in *Computational Intelligence (UKCI), 2012*, pp. 1–8, IEEE, 2012.
- [16] S. Rostami, P. Delves, and A. Shenfield, “Evolutionary multi-objective optimisation of an automotive active steering controller,” in *Science and Engineering Research Symposium, 2013*, pp. 1–3, Manchester Metropolitan University, 2013.
- [17] A. Chipperfield, P. Fleming, H. Pohlheim, and C. Fonseca, “Genetic algorithm toolbox for use with matlab,” tech. rep., Department of Automatic Control and Systems Engineering, University of Sheffield, 1994.
- [18] C. Darwin, “On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life,” *New York: D. Appleton*, 1859.
- [19] G. Mendel, J. Bennett, and S. R. A. Fisher, *Experiments in plant hybridization*. Oliver & Boyd, 1965.

-
- [20] S. Wright, "The roles of mutation, inbreeding, crossbreeding, and selection in evolution," in *Proc of the 6th International Congress of Genetics*, vol. 1, pp. 356–366, 1932.
- [21] D. C. Dennett, "Darwin's dangerous idea: Evolution and the meanings of life," *Synthese*, vol. 27, no. 3, 1995.
- [22] G. J. Friedman, *Selective feedback computers for engineering synthesis and nervous system analogy*. Masters thesis, UCLA, 1956.
- [23] G. E. Box, "Evolutionary operation: A method for increasing industrial productivity," *Applied Statistics*, vol. 6, no. 2, pp. 81–101, 1957.
- [24] R. M. Friedberg, "A learning machine: Part I," *IBM Journal of Research and Development*, vol. 2, no. 1, pp. 2–13, 1958.
- [25] R. M. Friedberg, B. Dunham, and J. H. North, "A learning machine: Part II," *IBM journal of research and development*, 1959.
- [26] I. Rechenberg, "Cybernetic solution path of an experimental problem," in *Royal Aircraft Establishment Translation No. 1122, B. F. Toms, Trans.*, Farnborough Hants: Ministry of Aviation, Royal Aircraft Establishment, Aug. 1965.
- [27] H.-P. Schwefel, "Experimentelle optimierung einer zweiphasenduse, teil i," Technical Report No. 35 of the Project MHD–Staustahlrohr 11.034/68, AEG Research Institute, Berlin, October 1968.
- [28] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial intelligence through simulated evolution*, vol. 26. Wiley New York, 1966.

-
- [29] J. H. Holland, “Nonlinear environments permitting efficient adaptation,” *Computer and Information Sciences*, vol. 2, pp. 147–164, 1967.
- [30] D. R. Frantz, *Nonlinearities in Genetic Adaptive Search*. PhD thesis, Ann Arbor, MI, USA, 1972. AAI7311116.
- [31] K. A. De Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*. PhD thesis, Ann Arbor, MI, USA, 1975. AAI7609381.
- [32] T. Back and H. P. Schwefel, “An overview of evolutionary algorithms for parameter optimization,” *Evolutionary computation*, vol. 1, no. 1, pp. 1–23, 1993.
- [33] Z. Michalewicz, *Genetic Algorithms Plus Data Structures Equals Evolution Programs*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2nd ed., 1994.
- [34] J. Knowles and D. Corne, “The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation,” in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 1, IEEE, 1999.
- [35] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii,” *Lecture notes in computer science*, vol. 1917, pp. 849–858, 2000.
- [36] C. Igel, N. Hansen, and S. Roth, “Covariance matrix adaptation for multi-objective optimization,” *Evolutionary computation*, vol. 15, no. 1, pp. 1–28, 2007.

-
- [37] Q. Zhang and H. Li, “Moea/d: A multiobjective evolutionary algorithm based on decomposition,” *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 712–731, 2007.
- [38] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: Empirical results,” *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [39] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable multi-objective optimization test problems,” in *Proceedings of the Congress on Evolutionary Computation (CEC-2002), (Honolulu, USA)*, pp. 825–830, Proceedings of the Congress on Evolutionary Computation (CEC-2002), (Honolulu, USA), 2002.
- [40] S. Huband, P. Hingston, L. Barone, and L. While, “A review of multi-objective test problems and a scalable test problem toolkit,” *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 5, pp. 477–506, 2006.
- [41] R. Dawkins, *The selfish gene*. No. 199, Oxford university press, 2006.
- [42] F. Neri and C. Cotta, “Memetic algorithms and memetic computing optimization: A literature review,” *Swarm and Evolutionary Computation*, vol. 2, pp. 1–14, 2012.
- [43] E. Burke, G. Kendall, J. Newall, E. Hart, P. Ross, and S. Schulenburg, “Hyper-heuristics: An emerging direction in modern search technology,” in *Handbook of metaheuristics*, pp. 457–474, Springer, 2003.

-
- [44] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1st ed., 1989.
- [45] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 67–82, 1997.
- [46] D. W. Corne and J. D. Knowles, “No free lunch and free leftovers theorems for multiobjective optimisation problems,” in *Evolutionary Multi-Criterion Optimization*, pp. 327–341, Springer, 2003.
- [47] Z. Michalewicz and D. B. Fogel, *How to Solve It: Modern Heuristics*. Berlin: Springer, 2000.
- [48] D. E. Goldberg and K. Deb, “A comparative analysis of selection schemes used in genetic algorithms,” *Urbana*, vol. 51, pp. 61801–2996, 1991.
- [49] N. Hansen and A. Ostermeier, “Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation,” in *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, pp. 312–317, IEEE, 1996.
- [50] N. Hansen and A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies,” *Evolutionary computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [51] N. Hansen, S. D. Müller, and P. Koumoutsakos, “Reducing the time complexity of the derandomized evolution strategy with covariance matrix

- adaptation (cma-es),” *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.
- [52] A. Auger and N. Hansen, “A restart cma evolution strategy with increasing population size,” in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 2, pp. 1769–1776, IEEE, 2005.
- [53] N. Hansen, “The cma evolution strategy: a comparing review,” in *Towards a new evolutionary computation*, pp. 75–102, Springer, 2006.
- [54] A. Auger and N. Hansen, “Performance evaluation of an advanced local search evolutionary algorithm,” in *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 2, pp. 1777–1784, IEEE, 2005.
- [55] 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,” *Evolutionary Computation, IEEE Transactions on*, vol. 18, pp. 577–601, Aug 2014.
- [56] A. Auger and O. Teytaud, “Continuous lunches are free!,” in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pp. 916–922, ACM, 2007.
- [57] N. Hansen and S. Kern, “Evaluating the cma evolution strategy on multimodal test functions,” in *Parallel Problem Solving from Nature-PPSN VIII*, pp. 282–291, Springer, 2004.
- [58] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Pošík, “Comparing results of 31 algorithms from the black-box optimization benchmarking bbob-2009,”

- in *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation*, pp. 1689–1696, ACM, 2010.
- [59] K. Deb, “Multi-objective optimization,” *Multi-objective optimization using evolutionary algorithms*, pp. 13–46, 2001.
- [60] E. Zitzler, L. Thiele, M. Laumanns, C. M., and V. G. da Fonseca, “Performance assessment of multiobjective optimizers: An analysis and review,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.
- [61] R. C. Purshouse, *On the Evolutionary Optimisation of Many Objectives*. PhD thesis, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, UK, S1 3JD, 2003.
- [62] F. W. Gembicki, *Vector Optimization for Control with Performance and Parameter Sensitive Indices*. PhD thesis, Case Western Reserve University, Cleveland, Ohio, 1974.
- [63] C.-L. Hwang and A. S. M. Masud, *Multiple Objective Decision Making - Methods and Applications*, vol. 164 of *Lecture Notes in Economics and Mathematical Systems*. Berlin: Springer-Verlag, 1979.
- [64] M. Farina and P. Amato, “On the optimal solution definition for many-criteria optimization problems,” in *Proceedings of the NAFIPS-FLINT International Conference* (J. Keller and O. Nasraoui, eds.), pp. 233 – 238, 2002.

-
- [65] W. Jakob, M. Gorges-Schleuter, and C. Blume, “Application of genetic algorithms to task planning and learning,” in *PPSN*, pp. 293–302, 1992.
- [66] C. A. Coello Coello, “Evolutionary multi-objective optimization: a historical view of the field,” *Computational Intelligence Magazine, IEEE*, vol. 1, no. 1, pp. 28–36, 2006.
- [67] P. A. Bosman and D. Thierens, “The balance between proximity and diversity in multiobjective evolutionary algorithms,” *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 2, pp. 174–188, 2003.
- [68] D. W. Corne, J. D. Knowles, and M. J. Oates, “The pareto envelope-based selection algorithm for multiobjective optimization,” in *Parallel Problem Solving from Nature PPSN VI*, pp. 839–848, Springer, 2000.
- [69] C. A. Coello Coello and G. T. Pulido, “A micro-genetic algorithm for multiobjective optimization,” in *Evolutionary multi-criterion optimization*, pp. 126–140, Springer, 2001.
- [70] K. Deb, M. Mohan, and S. Mishra, “Evaluating the ϵ -domination based multi-objective evolutionary algorithm for a quick computation of pareto-optimal solutions,” *Evolutionary computation*, vol. 13, no. 4, pp. 501–525, 2005.
- [71] M. Emmerich, N. Beume, and B. Naujoks, “An emo algorithm using the hypervolume measure as selection criterion,” in *Evolutionary Multi-Criterion Optimization*, pp. 62–76, Springer, 2005.

-
- [72] J. Bader and E. Zitzler, “Hype: An algorithm for fast hypervolume-based many-objective optimization,” *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.
- [73] K. Bringmann, T. Friedrich, C. Igel, and T. Voß, “Speeding up many-objective optimization by monte carlo approximations,” *Artificial Intelligence*, vol. 204, pp. 22–29, 2013.
- [74] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.
- [75] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, “Many objective optimization: An engineering perspective,” in *Proceedings of the International Conference on Evolutionary Multi-Objective Optimization (EMO2005)* (C. A. Coello Coello, A. H. Aguirre, and E. Zitzler, eds.), vol. 3470 of *Lecture Notes in Computer Science*, (Berlin), pp. 14 – 32, Springer-Verlag, 2005.
- [76] C. A. Coello Coello, “Handling preferences in evolutionary multiobjective optimization: A survey,” in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 1, pp. 30–37, IEEE, 2000.
- [77] C. M. Fonseca and P. J. Fleming, “Multiobjective optimization and multiple constraint handling with evolutionary algorithms - Part I: A unified formulation,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 28, no. 1, pp. 26 – 37, 1998.
- [78] C. M. Fonseca and P. J. Fleming, “Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization,” in *Proceedings*

- of the Fifth International Conference on Genetic Algorithms* (S. Forrest, ed.), pp. 416–423, Morgan Kaufmann, 1993.
- [79] I. A. Griffin, P. Schroder, A. J. Chipperfield, and P. J. Fleming, “Multi-objective optimization approach to the alstom gasifier problem,” *Proceedings of the institute of mechanical engineers*, vol. 214, no. 1, pp. 453–468, 2000.
- [80] A. Shenfield, P. J. Fleming, and M. Alkarouri, “Computational steering of a multi-objective evolutionary algorithm for engineering design,” *Engineering Applications of Artificial Intelligence*, vol. 20, no. 8, pp. 1047–1057, 2007.
- [81] K. Deb, J. Sundar, N. Udaya Bhaskara Rao, and S. Chaudhuri, “Reference point based multi-objective optimization using evolutionary algorithms,” *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 273–286, 2006.
- [82] M. Laumanns, G. Rudolph, and H.-P. Schwefel, “A spatial predator-prey approach to multi-objective optimization: A preliminary study,” in *Parallel Problem Solving from Nature PPSN V*, pp. 241–249, Springer, 1998.
- [83] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, “A preference-based evolutionary algorithm for multi-objective optimization,” *Evolutionary Computation*, vol. 17, no. 3, pp. 411–436, 2009.
- [84] E. Zitzler and S. Künzli, “Indicator-based selection in multiobjective search,” in *Parallel Problem Solving from Nature-PPSN VIII*, pp. 832–842, Springer, 2004.

-
- [85] N. Srinivas and K. Deb, “Multiobjective optimization using nondominated sorting in genetic algorithms,” *Evolutionary computation*, vol. 2, no. 3, pp. 221–248, 1994.
- [86] R. B. Agrawal, K. Deb, and R. B. Agrawal, “Simulated binary crossover for continuous search space,” 1994.
- [87] E. Zitzler and L. Thiele, *An evolutionary algorithm for multiobjective optimization: The strength pareto approach*. Citeseer, 1998.
- [88] T. Voß, N. Hansen, and C. Igel, “Improved step size adaptation for the mo-cma-es,” in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pp. 487–494, ACM, 2010.
- [89] C. Igel, T. Suttorp, and N. Hansen, “Steady-state selection and efficient covariance matrix update in the multi-objective cma-es,” in *Evolutionary Multi-Criterion Optimization*, pp. 171–185, Springer, 2007.
- [90] K. Miettinen, *Nonlinear multiobjective optimization*, vol. 12. Springer, 1999.
- [91] H. Li and Q. Zhang, “Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii,” *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 2, pp. 284–302, 2009.
- [92] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, “Multiobjective optimization test instances for the cec 2009 special session and competition,” *University of Essex, Colchester, UK and Nanyang Technological University, Singapore, Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, Technical Report*, 2008.

-
- [93] Q. Zhang, W. Liu, and H. Li, “The performance of a new version of moea/d on cec09 unconstrained mop test instances,” in *Evolutionary Computation, 2009. CEC’09. IEEE Congress on*, pp. 203–208, IEEE, 2009.
- [94] K. Deb, M. Mohan, and S. Mishra, “Towards a quick computation of well-spread pareto-optimal solutions,” in *Evolutionary multi-criterion optimization*, pp. 222–236, Springer, 2003.
- [95] T. Robič and B. Filipič, “Demo: Differential evolution for multiobjective optimization,” in *Evolutionary Multi-Criterion Optimization*, pp. 520–533, Springer, 2005.
- [96] 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 *Evolutionary Multi-Criterion Optimization*, pp. 495–509, Springer, 2009.
- [97] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, “Theory of the hypervolume indicator: optimal μ -distributions and the choice of the reference point,” in *Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pp. 87–102, ACM, 2009.
- [98] G. T. Pulido and C. A. C. Coello, “The micro genetic algorithm 2: Towards online adaptation in evolutionary multiobjective optimization,” in *Evolutionary Multi-Criterion Optimization*, pp. 252–266, Springer, 2003.
- [99] P. K. Tripathi, S. Bandyopadhyay, and S. K. Pal, “Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients,” *Information Sciences*, vol. 177, no. 22, pp. 5033–5049, 2007.

- [100] M. Gong, L. Jiao, H. Du, and L. Bo, “Multiobjective immune algorithm with nondominated neighbor-based selection,” *Evolutionary Computation*, vol. 16, no. 2, pp. 225–255, 2008.
- [101] A. Lara, G. Sanchez, C. A. Coello Coello, and O. Schutze, “Hcs: a new local search strategy for memetic multiobjective evolutionary algorithms,” *Evolutionary Computation, IEEE Transactions on*, vol. 14, no. 1, pp. 112–132, 2010.
- [102] J. Chen, Q. Lin, and Z. Ji, “A hybrid immune multiobjective optimization algorithm,” *European Journal of Operational Research*, vol. 204, no. 2, pp. 294–302, 2010.
- [103] R. L. Becerra and C. A. Coello Coello, “Solving hard multiobjective optimization problems using ε -constraint with cultured differential evolution,” in *Parallel Problem Solving from Nature-PPSN IX*, pp. 543–552, Springer, 2006.
- [104] A. J. Nebro, J. J. Durillo, C. A. Coello Coello, F. Luna, and E. Alba, “A study of convergence speed in multi-objective metaheuristics,” in *Parallel Problem Solving from Nature-PPSN X*, pp. 763–772, Springer, 2008.
- [105] S. F. Adra, T. J. Dodd, I. A. Griffin, and P. J. Fleming, “Convergence acceleration operator for multiobjective optimization,” *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 4, pp. 825–847, 2009.
- [106] R. C. Purshouse, C. Jalbă, and P. J. Fleming, “Preference-driven co-evolutionary algorithms show promise for many-objective optimisation,” in *Evolutionary Multi-Criterion Optimization*, pp. 136–150, Springer, 2011.

-
- [107] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, “Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications,” *Theoretical Computer Science*, vol. 425, pp. 75–103, 2012.
- [108] L.-Y. Tseng and C. Chen, “Multiple trajectory search for unconstrained/constrained multi-objective optimization,” in *Evolutionary Computation, 2009. CEC’09. IEEE Congress on*, pp. 1951–1958, IEEE, 2009.
- [109] H.-l. Liu and X. Li, “The multiobjective evolutionary algorithm based on determined weight and sub-regional search,” in *Evolutionary Computation, 2009. CEC’09. IEEE Congress on*, pp. 1928–1934, IEEE, 2009.
- [110] A. Mousa, M. El-Shorbagy, and W. Abd-El-Wahed, “Local search based hybrid particle swarm optimization algorithm for multiobjective optimization,” *Swarm and Evolutionary Computation*, vol. 3, pp. 1–14, 2012.
- [111] R. Akbari, R. Hedayatzadeh, K. Ziarati, and B. Hassanizadeh, “A multi-objective artificial bee colony algorithm,” *Swarm and Evolutionary Computation*, vol. 2, pp. 39–52, 2012.
- [112] O. Schutze, X. Esquivel, A. Lara, and C. A. Coello Coello, “Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization,” *Evolutionary Computation, IEEE Transactions on*, vol. 16, no. 4, pp. 504–522, 2012.
- [113] H. Lebesgue, “Intégrale, longueur, aire,” *Annali di matematica pura ed applicata*, vol. 7, no. 1, pp. 231–359, 1902.

-
- [114] M. Helbig and A. P. Engelbrecht, “Performance measures for dynamic multi-objective optimisation algorithms,” *Information Sciences*, vol. 250, pp. 61–81, 2013.
- [115] K. Van Moffaert, M. M. Drugan, and A. Nowé, “Hypervolume-based multi-objective reinforcement learning,” in *Evolutionary Multi-Criterion Optimization*, pp. 352–366, Springer, 2013.
- [116] L. While, P. Hingston, L. Barone, and S. Huband, “A faster algorithm for calculating hypervolume,” *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 1, pp. 29–38, 2006.
- [117] C. M. Fonseca, L. Paquete, and M. López-Ibáñez, “An improved dimension-sweep algorithm for the hypervolume indicator,” in *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pp. 1157–1163, IEEE, 2006.
- [118] Q. Yang and S. Ding, “Novel algorithm to calculate hypervolume indicator of pareto approximation set,” in *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques*, pp. 235–244, Springer, 2007.
- [119] K. Bringmann and T. Friedrich, “Approximating the least hypervolume contributor: Np-hard in general, but fast in practice,” in *Evolutionary Multi-Criterion Optimization*, pp. 6–20, Springer, 2009.
- [120] J. Bader, K. Deb, and E. Zitzler, “Faster hypervolume-based search using monte carlo sampling,” in *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems*, pp. 313–326, Springer, 2010.

- [121] Y. Tang, P. Reed, T. Wagener, *et al.*, “How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration?,” *Hydrology and Earth System Sciences Discussions*, vol. 10, no. 2, pp. 289–307, 2006.
- [122] A. J. Nebro, J. Durillo, J. Garcia-Nieto, C. A. Coello Coello, F. Luna, and E. Alba, “Smpso: A new pso-based metaheuristic for multi-objective optimization,” in *Computational intelligence in multi-criteria decision-making, 2009. mcdm'09. iee symposium on*, pp. 66–73, IEEE, 2009.
- [123] D. A. Van Veldhuizen and G. B. Lamont, “Evolutionary computation and convergence to a pareto front,” in *Late breaking papers at the genetic programming 1998 conference*, pp. 221–228, Citeseer, 1998.
- [124] D. A. Van Veldhuizen, “Multiobjective evolutionary algorithms: classifications, analyses, and new innovations,” tech. rep., DTIC Document, 1999.
- [125] G. T. Pulido and C. A. Coello Coello, “Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer,” in *Genetic and Evolutionary Computation—GECCO 2004*, pp. 225–237, Springer, 2004.
- [126] J. Knowles and D. Corne, “On metrics for comparing nondominated sets,” in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, vol. 1, pp. 711–716, IEEE, 2002.
- [127] A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, and E. Alba, “Mocell: A cellular genetic algorithm for multiobjective optimization,” *International Journal of Intelligent Systems*, vol. 24, no. 7, pp. 726–746, 2009.

-
- [128] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, “Handling multiple objectives with particle swarm optimization,” *Evolutionary Computation, IEEE Transactions on*, vol. 8, no. 3, pp. 256–279, 2004.
- [129] C. A. Coello Coello and M. R. Sierra, “A coevolutionary multi-objective evolutionary algorithm,” in *Evolutionary Computation, 2003. CEC’03. The 2003 Congress on*, vol. 1, pp. 482–489, IEEE, 2003.
- [130] C. A. Coello Coello and N. C. Cortés, “Solving multiobjective optimization problems using an artificial immune system,” *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, 2005.
- [131] Q. Zhang, A. Zhou, and Y. Jin, “Rm-meda: A regularity model-based multiobjective estimation of distribution algorithm,” *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 1, pp. 41–63, 2008.
- [132] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, “Expensive multiobjective optimization by moea/d with gaussian process model,” *Evolutionary Computation, IEEE Transactions on*, vol. 14, no. 3, pp. 456–474, 2010.
- [133] S. Tiwari, G. Fadel, P. Koch, and K. Deb, “Performance assessment of the hybrid archive-based micro genetic algorithm (amga) on the cec09 test problems,” in *Evolutionary Computation, 2009. CEC’09. IEEE Congress on*, pp. 1935–1942, IEEE, 2009.
- [134] C.-M. Chen, Y.-p. Chen, and Q. Zhang, “Enhancing moea/d with guided mutation and priority update for multi-objective optimization,” in *Evolutionary Computation, 2009. CEC’09. IEEE Congress on*, pp. 209–216, IEEE, 2009.

-
- [135] M. Nasir, A. K. Mondal, S. Sengupta, S. Das, and A. Abraham, “An improved multiobjective evolutionary algorithm based on decomposition with fuzzy dominance,” in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pp. 765–772, IEEE, 2011.
- [136] K. Li, S. Kwong, J. Cao, M. Li, J. Zheng, and R. Shen, “Achieving balance between proximity and diversity in multi-objective evolutionary algorithm,” *Information Sciences*, vol. 182, no. 1, pp. 220–242, 2012.
- [137] J. Derrac, S. García, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms,” *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.
- [138] M. Epitropakis, V. Plagianakos, and M. Vrahatis, “Evolving cognitive and social experience in particle swarm optimization through differential evolution: A hybrid approach,” *Information Sciences: an International Journal*, vol. 216, pp. 50–92, 2012.
- [139] F. Wilcoxon, “Individual comparisons by ranking methods,” *Biometrics bulletin*, pp. 80–83, 1945.
- [140] J. Derrac, C. Cornelis, S. García, and F. Herrera, “Enhancing evolutionary instance selection algorithms by means of fuzzy rough set based feature selection,” *Information Sciences*, vol. 186, no. 1, pp. 73–92, 2012.
- [141] S. García, A. Fernández, J. Luengo, and F. Herrera, “Advanced nonparametric tests for multiple comparisons in the design of experiments in com-

- putational intelligence and data mining: Experimental analysis of power,” *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.
- [142] B. Trawiński, M. Smetek, Z. Telec, and T. Lasota, “Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms,” *International Journal of Applied Mathematics and Computer Science*, vol. 22, no. 4, pp. 867–881, 2012.
- [143] A. Hatamlou, “Black hole: A new heuristic optimization approach for data clustering,” *Information Sciences*, vol. 222, pp. 175–184, 2013.
- [144] P. Civicioglu, “Backtracking search optimization algorithm for numerical optimization problems,” *Applied Mathematics and Computation*, vol. 219, no. 15, pp. 8121–8144, 2013.
- [145] Z. Yang, K. Tang, and X. Yao, “Large scale evolutionary optimization using cooperative coevolution,” *Information Sciences*, vol. 178, no. 15, pp. 2985–2999, 2008.
- [146] A. Zamuda, J. Brest, B. Boskovic, and V. Zumer, “Differential evolution for multiobjective optimization with self adaptation.,” in *IEEE Congress on Evolutionary Computation*, pp. 3617–3624, 2007.
- [147] I. De Falco, A. Della Cioppa, D. Maisto, U. Scafuri, and E. Tarantino, “Biological invasion–inspired migration in distributed evolutionary algorithms,” *Information Sciences*, vol. 207, pp. 50–65, 2012.
- [148] S. García, D. Molina, M. Lozano, and F. Herrera, “A study on the use of non-parametric tests for analyzing the evolutionary algorithms behaviour:

- a case study on the cec2005 special session on real parameter optimization,” *Journal of Heuristics*, vol. 15, no. 6, pp. 617–644, 2009.
- [149] Y. Wang, Z. Cai, and Q. Zhang, “Differential evolution with composite trial vector generation strategies and control parameters,” *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 55–66, 2011.
- [150] J. D. Knowles and D. W. Corne, “Approximating the nondominated front using the pareto archived evolution strategy,” *Evolutionary computation*, vol. 8, no. 2, pp. 149–172, 2000.
- [151] H. Beyer and B. Sendhoff, “Covariance matrix adaptation revisited—the cmsa evolution strategy—,” in *Parallel Problem Solving from Nature—PPSN X*, pp. 123–132, Springer, 2008.
- [152] C. A. Coello Coello and G. B. Lamont, *Applications of multi-objective evolutionary algorithms*, vol. 1. World Scientific, 2004.
- [153] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications: Algorithms and Applications*. Springer, 2006.
- [154] H. Ishibuchi, N. Tsukamoto, Y. Sakane, and Y. Nojima, “Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing functions,” in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pp. 527–534, ACM, 2010.
- [155] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, “Multiobjective evolutionary algorithms: A survey of the state of the art,” *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

- [156] S. Chaudhuri and K. Deb, “An interactive evolutionary multi-objective optimization and decision making procedure,” *Applied Soft Computing*, vol. 10, no. 2, pp. 496–511, 2010.
- [157] C. A. Coello Coello, “Research directions in evolutionary multi-objective optimization,” *Evolutionary Computation Journal*, vol. 3, no. 3, pp. 110–121, 2012.
- [158] D. Tabak, A. Schy, D. Giesy, and K. Johnson, “Application of multiobjective optimization in aircraft control systems design,” *Automatica*, vol. 15, no. 5, pp. 595–600, 1979.
- [159] J. J. Durillo, A. J. Nebro, C. A. Coello Coello, F. Luna, and E. Alba, “A comparative study of the effect of parameter scalability in multi-objective metaheuristics,” in *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pp. 1893–1900, IEEE, 2008.
- [160] G. Howard, E. Gale, L. Bull, B. de Lacy Costello, and A. Adamatzky, “Evolution of plastic learning in spiking networks via memristive connections,” *Evolutionary Computation, IEEE Transactions on*, vol. 16, no. 5, pp. 711–729, 2012.
- [161] N. Bowring, M. Southgate, D. Andrews, N. D. Rezgui, S. Harmer, and D. O’Reilly, “Development of a longer range standoff millimetre wave radar concealed threat detector,” in *Proc. SPIE 8714, Radar Sensor Technology XVII*, pp. 1–8, SPIE, 2013.

-
- [162] D. O'Reilly, N. Bowring, and S. Harmer, "Signal processing techniques for concealed weapon detection by use of neural networks," in *IEEE 27th Convention of Electrical and Electronic Engineers in Israel IEEE*, pp. 1–4, IEEE, 2012.
- [163] D. O'Reilly, N. Bowring, N. D. Rezgui, D. Andrews, and S. Harmer, "Remote concealed threat detection by novel classification algorithms applied to multipolarimetric uwb radar," in *Proc. SPIE 8714, Radar Sensor Technology XVII*, pp. 1–8, SPIE, 2013.
- [164] S. Rostami, D. O'Reilly, A. Shenfield, and N. Bowring, "A novel preference articulation operator for the evolutionary multi-objective optimisation of classifiers in concealed weapons detection," *Information Sciences*, vol. 295, no. 0, pp. 494 – 520, 2015.

Appendix A

Algorithm Pseudo-code

A.1 MO-CMA-ES

The algorithm pseudo-code for MO-CMA-ES [88].

Algorithm 5 MO-CMA-ES algorithm pseudo-code

```

1:  $g \leftarrow 0$ 
2: initialize parent population  $Q^0$ 
3: while termination criteria not met do
4:   for  $k = 1, \dots, \lambda$  do
5:      $i_k \leftarrow k$ ;
6:      $a_k^{(g+1)} \leftarrow a_{i_k}^{(g)}$ 
7:      $x_k^{(g+1)} \sim x_{i_k}^{(g)} + \sigma_{i_k}^{(g)} \mathcal{N}\left(0, C_{i_k}^{(g)}\right)$ ;
8:      $Q^{(g)} \leftarrow Q^{(g)} \cup \left\{a_k^{(g+1)}\right\}$ ;
9:   end for
10:  for  $k = 1, \dots, \lambda$  do
11:     $\bar{p}_{succ,k}^{(g+1)} \leftarrow (1 - c_p)\bar{p}_{succ,k}^{(g)} + c_p succ_{Q^{(g)}}\left(a_k^{(g)}, a_k^{(g+1)}\right)$ ;
12:     $\sigma_k^{(g+1)} \leftarrow \sigma_k^{(g)} \exp\left(\frac{1}{d} \frac{\bar{p}_{succ,k}^{(g+1)} - p_{succ}^{target}}{1 - p_{succ}^{target}}\right)$ ;
13:    if  $\bar{p}_{succ,k}^{(g+1)} < p_{thresh}$  then
14:       $p_{c,k}^{(g+1)} \leftarrow (1 - c_c)p_{c,k}^{(g)} + \sqrt{c_c(2 - c_c)} \frac{x_k^{(g+1)} - x_{i_k}^{(g)}}{\sigma_k^{(g)}}$ ;
15:       $\mathbf{C}_k^{(g+1)} \leftarrow (1 - c_{cov})\mathbf{C}_k^{(g)} + c_{cov} p_{c,k}^{(g+1)} p_{c,k}^{(g+1)\top}$ 
16:    else
17:       $p_{c,k}^{(g+1)} \leftarrow (1 - c_c)p_{c,k}^{(g)}$ ;
18:       $\mathbf{C}_k^{(g+1)} \leftarrow (1 - c_{cov})\mathbf{C}_k^{(g)} + c_{cov} \left( p_{c,k}^{(g+1)} p_{c,k}^{(g+1)\top} + c_c(2 - c_c)\mathbf{C}_k^{(g)} \right)$ 
19:    end if
20:     $\bar{p}^{(g)}_{i_k} \leftarrow (1 - c_p)\bar{p}^{(g)}_{i_k} + c_p succ_{Q^{(g)}}\left(a_{i_k}^{(g)}, a_{i_k}^{(g+1)}\right)$ ;
21:     $\sigma_{i_k}^{(g)} \leftarrow \sigma_{i_k}^{(g)} \exp\left(\frac{1}{d} \frac{\bar{p}_{succ,i_k}^{(g)} - p_{succ}^{target}}{1 - p_{succ}^{target}}\right)$ ;
22:  end for
23:   $g \leftarrow g + 1$ ;
24:   $Q^{(g)} \leftarrow \left\{Q_{\prec:i}^{(g-1)} \mid 1 \leq i \leq \mu\right\}$ ;
25: end while

```

A.2 Weighted Z-score Operator

The algorithm pseudo-code describing the WZ preference articulation operator proposed in [164].

Algorithm 6 Weighted Z-score pseudo-code

```
1: nSolutionsSatisfyObjectives =  
   PopulationObjectives <= RegionOfInterest  
2: for i = 1 : nObjectives do  
3:   WeightedZscore(i) = AbsZscore(PopulationObjectives(i),  
   RegionOfInterest(i))  
4:   if nSolutionsSatisfyObjectives(i) == 0 then  
5:     WeightedZscore(i) = WeightedZscore(i) × (1 - 1 ×  
     nObjectives)  
6:   end if  
7: end for  
8: for i = 1 : nSolutions do  
9:   SummedWeightedZScore = sum(WeightedZscore(i))  
10: end for
```

Appendix B

Algorithms not considered

B.1 NSGA-III

The Nondominated Sorting Genetic Algorithm III (NSGA-III) is a multi-objective algorithm which builds upon the popular Nondominated Sorting Genetic Algorithm II (NSGA-II) framework in order to tackle many-objective problems.

NSGA-III uses structured weights, which the authors refer to as “reference-points”, in order to achieve the diversity characteristic throughout the optimisation process. This approach is similar to the approach taken in the Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D), such that both NSGA-III and MOEA/D can be initialised with the same set of weights, and results in a pre-defined guided mechanism to achieve diversity based on the distribution and structure of these weights. Another feature of NSGA-III is the use of niching in order to achieve efficient recombination, without this feature two distant parents are likely to produce offspring which are also distant from the parents.

The EMO algorithm is proposed in [55], and although the authors suggest “algorithms must be tested on other more challenging problems than the usual

normalized test problems such as DTLZ problems”, NSGA-III is compared to MOEA/D on a limited set of test functions: DTLZ1, DTLZ2, DTLZ3, DTLZ4, WFG6, WFG7. The sample size used for this comparison is not proven to be sufficient or greater than a sample size of one, and there is no statistical verification to indicate the significance of the results obtained.

NSGA-III also aims to allow the incorporation of preference information, however, the incorporation of preferences is implemented by simply providing different weights or “aspiration-points” when initialising the algorithm. This approach is not user-friendly, as there is no way for a decision maker to provide a goal or “reference-points”, they must instead design a structure of weights which are distributed to reflect the preferences.

The NSGA-III pseudo-code has been listed in Algorithm 7, however, there is currently no complete and fully functioning first-party implementation of the NSGA-III algorithm available, and the available third-party implementation does not function on problems consisting of more than three problem objectives.

For these reasons, NSGA-III has not been considered for comparison in this thesis.

Algorithm 7 Generation t of NSGA-III algorithm pseudo-code

- 1: **Input:** H structured reference points Z^S or supplied aspiration points Z^a , parent population P^t .
 - 2: **Output:** P_{t+1}
 - 3: $S_t = \Phi$, $i = 1$
 - 4: $Q_t = \text{Recombination+Mutation}(P_t)$
 - 5: $R_t = P_t \cup Q_t$
 - 6: $(F_1, F_2, \dots) = \text{Non-dominated sort}(R_t)$
 - 7: **do**
 - 8: $S_t = S_t \cup F_i$
 - 9: $i = i + 1$
 - 10: **while** $|S_t| \geq N$ Last front to be included: $F_l = F_i$
 - 11: **if** $|S_t| = N$ **then**
 - 12: $P_{t+1} = S_t$, break
 - 13: **else**
 - 14: $P_{t+1} = \cup_{j=1}^{l-1} F_j$
 - 15: Points to be chosen from F_l :: $K = N - |P_{t+1}|$
 - 16: Normalise objectives and create reference set Z^r :
Normalise(f^n, S_t, Z^r, Z^s, Z^a)
 - 17: Associate each member s of S_t with a reference point: $[\pi(s), d(s)] = \text{Associate}(S_t, Z^r)$.
 - 18: Compute niche count of reference point $j \in Z^r$: $\rho_j = \sum S \in S_t / F_l ((\pi(s) = j) ? 1 : 0)$
 - 19: Choose K members one at a time from F_l to construct P_{t+1} :
Niching($K, \rho_j, \pi, d, Z^r, F_l, P_{t+1}$)
 - 20: **end if**
-

B.2 DEMO and DEMOwSA

The Differential Evolution for Multi-objective Optimisation (DEMO) is an Evolutionary Multi-Objective Optimisation (EMO) algorithm which is based around differential evolution, it has been extended in Differential Evolution for Multi-objective Optimisation with Self Adaptation (DEMOwSA) to include self-adaptation which was inspired by the self-adaptation in Evolution Strategies (ES).

DEMO builds upon differential evolution which is a simple but powerful Evolutionary Algorithm (EA) with many successful applications. DEMO aims to combine the advantages provided by differential evolution with mechanisms of Pareto-based ranking and crowding distance for sorting.

DEMO is introduced and benchmarked on the ZDT synthetic test suite in [95] in three variants:

- **DEMO/parent**, immediately replaces a parent solution with the candidate solution that dominates it. Uses non-dominated sorting and crowding distance metric.
- **DEMO/closest/dec**, works the same way as DEMO/parent, with an exception, such that a candidate solution replaces the most similar solution in the decision space if it dominates it.
- **DEMO/closest/obj**, works the same way as DEMO/closest/dec, except candidate solutions are compared in the objective space rather than the decision space.

The authors conclude that DEMO/closest/dec and DEMO/closest/obj are too computationally expensive and do not offer any important advantage over

DEMO/parent, and therefore recommend the DEMO/parent variant be used in future experiments.

The DEMO pseudo-code has been listed in Algorithm 8

Algorithm 8 DEMO algorithm pseudo-code

```

1: Evaluate initial population  $P$  of random individuals.
2: while termination criteria not met do
3:   for each individual  $P_i (i = 1, \dots, popSize)$  from  $P$  do
4:     Create candidate  $C$  from parent  $P_i$ 
5:     Evaluate candidate
6:     if  $C > P_i$  then
7:        $P_i \leftarrow C$ 
8:     else if  $C < P_i$  then
9:       discard  $C$ 
10:    else
11:      insert  $C$  into  $P$ .
12:    end if
13:  end for
14:  if  $size(P) > popSize$  then
15:    truncate( $P$ )
16:  end if
17:  Randomly enumerate the individuals in  $P$ .
18: end while

```

DEMOwSA is introduced in [146], and is benchmarked on a number of synthetic test functions including test functions from the ZDT and WFG synthetic test suites.

DEMO and DEMOwSA are not focussed on many-objective optimisation, and depend on non-dominated sorting and the crowded comparison metric which does not scale feasibly for many-objective problems. They also do not offer a comparison to any other state-of-the-art EMO algorithm as a benchmark of their performance.

For these reasons, DEMO and DEMOwSA have not been considered for comparison in this thesis. However, MOEA/D which also uses differential evolution during the optimisation process has been considered for comparison, as it is a competition winning and thoroughly benchmarked state-of-the-art algorithm.