Performance of multi-objective algorithms on varying test problem features using the WFG toolkit

Liang Zhao

This report is submitted as partial fulfilment
of the requirements for the Honours Programme of the
School of Computer Science and Software Engineering,
The University of Western Australia,
2007

Abstract

Multi-objective optimisation is widely useful in solving many real world problems. We are interested in how well multi-objective optimisation evolutionary algorithms (MOEA) perform on problems of certain types of features. The WFG toolkit is a very flexible test problem toolkit which allows the user to easily construct and manipulate test problems with differing feature transformations.

This project aims to investigate the performance of three popular MOEAs (NSGA-II, SPEA2 and IBEA) with the example WFG test problems. By varying test problem features we aim to demonstrate the “no free lunch” theorem, that there is no generally superior MOEA.

Experimentation is carried out by systematically varying dimension and features transformations of the example WFG problems, by comparing the results after each test we build knowledge of which features present difficulties for certain MOEAs. We can then combine a selection of test problem features together to show how the performance of a MOEA change, this combining of feature transformations can be accomplished easily with the WFG toolkit.

The results show that IBEA outperforms SPEA2 and NSGA-II in the default example WFG test problems for two, three and five dimensions. However through investigations and by adding transformations and changing shape functions within the WFG toolkit which we believe create difficulties for IBEA, we are able to construct new test problems where IBEA's performance lead decreases and eventually loses to NSGA-II and SPEA2.

In this project we have successfully demonstrated an example of the “no free lunch” theorem by tweaking various features of a test problem, as well as find a set of test problem features which present difficulties for IBEA. Through our testing we have also shown the versatility and usefulness of the WFG toolkit.

Acknowledgements

I would like to thank my supervisor Luigi Barone for his continued support, encouragement and guidance which has helped me tremendously. As well I owe a big thanks to my parents and friends for their constant support and love. A special acknowledgement to Lucas Bradstreet, who has helped me with setup of PISA and use of his fast hypervolume implementation. And finally I thank God for his mercy, wisdom and for answering my prayers.

Contents

List of Tables

List of Figures

1 Introduction

Evolutionary algorithms (EAs) provide a simple yet intuitive way of solving complex optimisation problems [1], without prior knowledge of the characteristics or features of a problem. In optimisation the aim is to find the best possible solution based on a set of constraints, given one objective we only judge a solution based on one criteria, thus we are clear on how good this solution is. However many real world optimisation problems contain multiple objectives or are multiobjective, where solutions are judged based on multiple criteria, and the new aim is to find solution sets with diverse tradeoffs between the objectives.

It is difficult to extend conventional optimisation techniques to the true multi-objective case, as they were not designed with multiple objectives in mind [14]. However evolutionary algorithms are positively well-suited to multi-objective optimisation [14], and are used in many real-world problems [10], such as in scheduling [2], nuclear fuel management [3] and telecommunications network design [4]. The growing success of MOEAs has given rise to many newly developed MOEAs [5]-[9], which raises the question of whether there exists a generally superior algorithm.

The performance MOEAs can be benchmarked using test problems, test problems are useful as they allow researchers to test the performance MOEAs in a controlled environment. For this purpose artificially constructed test problems have significant advantages over real-world problems, as they can be designed to be fast, easy to describe, easy to understand and visualise, easy to implement, and their optima known in advance [10].

The “no free lunch” theorem [11] states that for any optimization algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class. Thus we can deduce that there does not exist a superior MOEA, or for any given MOEA there must exist some class of test problems or performance measures in which the MOEA will perform poorly.

This motivates the purpose of this project, we aim to investigate the performance of several popular MOEAs on test problems with varying features. We also aim to demonstrate the "no free lunch" theorem by finding certain test problem features which present difficulties for particular MOEAs.

This dissertation is structured as follows. Section two gives an overview of evolutionary algorithms, how it works, some popular MOEAs and performance measures. Section three describes test problem characteristics, popular test problem suites in literature, and a framework for MOEA implementation and performance testing. Section four presents our experimental methodology, experimental results and analysis of those results. Section five is the conclusion.

2 Background

2.1 Evolutionary Algorithms

Optimisation aims at locating a good solution given a set of constraints [13]. Many real world problems can be modelled and optimised. However, optimisation of complex systems by checking the performance of all parameters is often impossible as a consequence of non-separability, where parameters are dependent upon each other and adjusting one parameter affects others. Evolutionary algorithms are particularly well suited in solving complex optimisation problems [bi1] due to their adaptability and non-assumption of the underlying fitness landscape.

Many terms used in EAs are related to natural selection process. For example individuals in the context of EAs refer to a single solution, whilst a population represents a set of solutions. Parent solutions refers to solutions in the initial population or the population after any number of generations, whilst offspring solutions are new solutions created in each generation from parent solutions based on 'genetic operators', such as mutation and recombination. The former generates offspring solutions by modifying individual parameters of the parent solution, whilst the latter combines the parameters of two parent solutions to form an offspring solution. Evolutionary algorithms imitate the process of natural evolution in computers [1]. In the context of optimisation, this involves the following:

1. Generation of an initial random population of solutions.
2. Applying the fitness function to each solution.
3. Selection of quality solutions as candidates for parents.
4. Generation of new solutions from parents using genetic operators such as mutation and recombination.
5. Iterate through steps 2-4 until solutions reach a desired level of fitness.

The first step represents creation of a diverse solution set by forming solutions with random parameters. In the second step, a fitness function is used to determine the quality of each solution. If the quality of a solution in each objective can be quantified into a vector x then a naïve fitness function can determine the sum of x as a representation of its quality or fitness, modern MOEAs use other methods discussed in Section 2.6. In step 3, the fitness calculations from step 2 are used as a reference to selection of solution candidates for reproduction. Individuals with a higher fitness score are stronger candidates as parents, the binary tournament selection method directly compares random pairs of solutions in the population and chooses solutions with a higher fitness score. A stochastic approach such as the probability tournament method more mimics the natural selection, where random pairs of solutions are compared but there is a probability that the losing solution is selected. Using the stochastic approach leads to a more diverse selection process.

In step 4, parent candidates selected in step 3 are used to generate offspring solutions. Genetic operators such as recombination and mutation are used with the parent population to generate a new population of offspring solutions. Recombination takes two parent solutions and combines them to create one new offspring solution, the frequency of recombination is controlled by the crossover probability parameter. The mutation operator mimics evolution by modifying the chromosome or individual elements in a solution vector, the mutation probability determines how often mutation occurs. When the mutation probability is equal to 1 then the offspring solution will always mutate from a parent solutions. Step 5 iterates through steps 2 and 4, every iteration is termed a generation, and the population evolves upon each generation.

One disadvantage of EAs is that it is not clear when the evolution of a population should terminate. One way to approach this problem is to visually graph the solution front after each generation, where a solution front is a set of solutions in a population representative of its shape in fitness space. By observing the changes in the solution front after each generation, we can make a decision on the number of generations needed based on the rate of convergence of the solution front towards the optimal front.

2.2 Multiobjective optimisation

Optimisation problems which contain more than one objective are known as a multi-objective optimisation problem (MOP), where the task is to finding an optimal solution to a problem in which candidate solutions are judged according to multiple criteria that conflict with each other to some degree [29]. Because multiple criteria are conflicting, optimal solutions cannot be improved in one objective without losing out in another objective. Instead we aim to find a set of solutions or a solution front, which best represents the 'Pareto optimal set' containing only solutions of optimal tradeoffs between objectives. A decision can then be made on which solution to use, based on specific preferences for the objectives.

A generic multiobjective problem can be defined by a search space of n parameters $x_1,\ldots,x_n$, with m objective functions $f_1(x_1,\ldots,x_n),\ldots,f_m(x_1,\ldots,x_n)$, where the objective functions map parameter vector into fitness space.

2.3 Problem of incomparability

MOEAs solve problems by mimicking the natural selection process just as single objective evolutionary algorithms do, in each generation the algorithm makes progress by firstly assigning fitness to its solutions [15]. Unlike the single objective case where solutions can be directly compared based on the fitness, when optimising a problem with multiple objectives a deterministic fitness function for the quality of a solution based on the performance in all its objective does not usually exist, and so we cannot compare different solutions directly.

2.4 Pareto Optimum

Instead other approaches are taken to define fitness functions which work in the multi-objective case. One approach is to rank a solution based on the concept of domination. By definition a solution *a* dominates solution *b* iff *a* performs at least as good as *b* in all objectives and *a* beats *b* in at least one objective. A solution *a* is non-dominated with respect to a set of solutions X iff there are no solutions in X which dominates *a*. Two solutions are incomparable if neither dominates the other, and a non-dominated solution set X is formed iff all solutions in X are non-dominated with respect to all solutions in X. Non-dominated solutions in a solution set form a 'solution front' representing the best solutions in solution set, it can be visualised with three or less objectives by plotting the solution vectors and linking the non-dominated solutions.

The concept of domination leads to the concept of Pareto optimum. By definition a solution *a* is Pareto optimal iff it remains non-dominated with respect to the solution set X representing all possible solutions. Pareto optimal solutions have the characteristic that improvements in one objective cannot occur without the decline in at least one other objective. The set of all Pareto optimal solutions form the Pareto optimal set X representing the best solutions for a given problem, since all solutions in X are non-dominated the solutions in the Pareto optimal set form the Pareto optimal front. Due to the stochastic nature of evolutionary algorithms, there is no guarantee that a MOEA will reach a set of Pareto optimal solutions, rather it attempts to find the best solution given limited resources.

2.5 Performance measures

We are interested in how well MOEAs are able to perform, however often it is impossible to compare a resultant front with the Pareto optimal front because it is unknown. However we are able to compare the relative performance of several MOEAs to judge the ability of a MOEA. Several performance indicators have been proposed [16] to measure the change in performance between different solution populations. Performance measures are useful as they translate the quality of a solution set into a scalar quantity, which can be used in comparison.

One such measure is the epsilon indicator [17], [29]. Consider a minimization problem, by definition the binary epsilon indicator $I_{\epsilon}(\mathbf{X},\mathbf{Y})$ for solution sets **X** and **Y** finds the minimum factor epsilon, for which any objective vector in **Y** is ϵ-dominated by at least one objective vector in **X**, where solution **a** ϵ-dominates solution **b** if **a** <

ϵ*$\mathbf{b}$. More generally the epsilon indicator measures the factor or by which a solution set **Y** is worse than the solution set **X** with respect to all objectives, or the scalar distance between **Y** and **X.**

An alternative indicator is the hypervolume indicator or the s-metric. It measures the 'volume' of the objective space dominated by a solution set X as shown in Figure 1 [19], this metric is more computationally expensive than the epsilon indicator. The binary hypervolume indicator[17] IH2(**a**,**b**) measures the hypervolume of the subspace weakly dominated by solution **a** but not solution **b**. Hypervolume is often preferred to the epsilon indicator as it capture the change in volume as opposed to the change in distance. Although hypervolume is computationally much more expensive, major progress has been made by [19], [20] to dramatically increase hypervolume calculation speed.

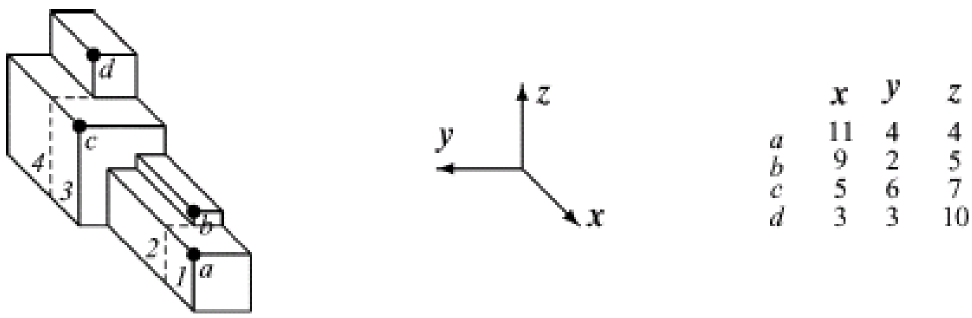


Figure 1: Example of hypervolume for four solutions a b c and d in three dimensions, taken from [19].

A different performance indicator is the R metric [18] proposed by Hansen and Jaczkiewicz. The r metric is built on the concept of outperformance, where solution **a** outperforms solution **b** subject to a set U of utility functions, if there exists some utility functions in set U that achieve better values in **a** than in **b**. The utility function maps each point in objective space into a value of utility, Up is a set of utility functions based on the form defined in (1).

$$u_p\left(\mathbf{z},\mathbf{z}^*,\Lambda,p\right)=-\left(\sum_{j=1}^{J}\lambda_i\left(z_j^*-z_j\right)^p\right)^{1/p},\ p\in\{1,2,\ldots\}+\{\infty\}. \qquad (1)$$

Where **z** is a set of objective functions, **z*** represents the best attainable objective function values, Λ is a weight vector and p is utility function index. The *u**(A) function is defined to represent the maximum utility value of solutions in a solution set. Three versions of the R metric R1, R2 and R3 are proposed in [18]. Define **A** and **B** as solution sets, R1 measures the probability that the *u****(**A**) is better than *u****(**B**), R2 is defined by the expected value *u**(**A**) minus *u**(**B**), and R3 measures expected value of the ratio of u*(**B)** minus *u**(**A**) over *u**(**B**) the utility of a minus the expected value of the utility of b. The R metric provides a different way of performance measurement based on utility functions.

2.6 Multiobjective Evolutionary Algorithms

In this section we will investigate some popular MOEAs used in EA literature. We chose NSGA-II by Deb et al. [9], SPEA2 by Zitzler et al.[7], and IBEA by Zitzler et al.[7]. These algorithms differ significantly in their choice of fitness selection and diversity preservation.

2.6.1 NSGA II

The non-dominated sorting genetic algorithm 2 (NSGA II) [9] is a popular MOEA which improves on the original NSGA [21]. NSGA II assesses the fitness of a solution based on the number of other solutions which dominate it, in each generation NSGA II calculates for each solution the number of other solutions which dominate it, the higher this value is the worse the fitness assigned to the solution, thus non-dominated solutions are assigned the best fitness.

As NSGA II strives towards the Pareto optimal solutions by discarding solutions of lower fitness, eventually there will be many solutions of equal fitness. NSGA II employs a diversity preservation mechanism known as crowding distance to assess between solutions of equal fitness and keep diversification. The crowding distance is calculated for a particular solution by averaging the distance between its two closest neighbouring solutions of equal fitness, see Figure 2 [9]. Solutions with the lowest values of crowding distance are discarded when required as they are considered more clustered.

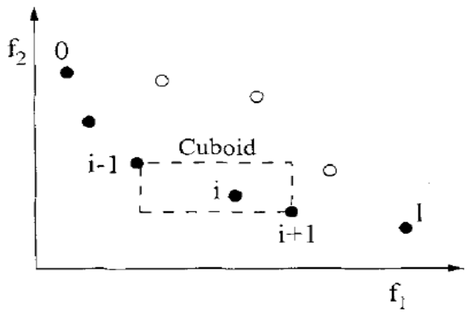


*Figure 2: Example of crowding distance for solution **i**, taken from [9].*

The NSGA II makes several improvements on the original NSGA with emphasis on efficiency and speed. Most noticeably it reduced the computational complexity of non-dominated sorting from $O(MN^3)$ to $O(MN^2)$ (M as the number of objectives and N as population size) by introducing a smarter sorting algorithm. NSGA II also introduced the use of elitism maintaining separate archive of non-dominated solutions which is used as a reference for the selection mechanism, the use of elitism prevents loss of non-dominated solutions and also shown [22] to achieve better convergence in MOEAs. Finally NSGA II replaces the old diversity preservation mechanism with the new crowding distance which does not require the specification of a sharing parameter.

2.6.2 SPEA2

The strength Pareto evolutionary algorithm 2 (SPEA2) [7] is another popular MOEA which is based on its predecessor SPEA [8]. SPEA operates by keeping a regular population as well as an archive of best solutions. For each generation all non-dominated solutions in the regular population are taken and combined with the archived solutions to update it, in this process any dominated or duplicate solutions are removed from the archive. When the archive reaches a pre-defined maximum size solutions are discarded based on a clustering technique in order to keep the characteristics of the non-dominated solution front.

In each generation new fitness values are calculated for both the population and archived solution sets. Firstly strength values are calculated for each solution x in the archived set, which is calculated by taking the number of population solutions y that are dominated by or equal to x, divided by the total population size plus one. Then, the fitness value of a solution x in the population set is the summation of the strength values of all archived set solutions y which dominate or are equal to x added by one, thus non-dominated solutions have the lowest fitness values.

After fitness assignment offspring are produced by selecting solutions from the combined population and archived set, parents are selected by means of binary tournaments [8]. Solutions with lower fitness values are given more importance, and solutions from the archived set are more likely to be chosen as a result of this. The final offspring population replaces the original population set to form the new population set for the next generation.

SPEA2 improves upon the original SPEA to address a few weaknesses in the algorithm. Firstly SPEA2 incorporates more precise fitness assignment by calculating strength values for both population and archived set based on solutions from not one but both solution sets, this is done to improve decision making between solutions of equal fitness. Secondly, in SPEA2 the archived solution set is fixed in size and always contains the same number of solutions, when there are not enough non-dominated solutions to accommodate, dominated solutions are used. This is done to prevent the situation where there is only one archived solution which causes all population solutions to have the same fitness value. Thirdly in SPEA2 only solutions from the archived set are used for the selection of parents for reproduction.

2.6.3 IBEA

The indicator-based evolutionary algorithm (IBEA) [6] is designed as a generic MOEA which uses various performance measures as its selection mechanism. Instead of using fixed fitness functions as with SPEA2 and NSGA II, IBEA can incorporate an arbitrary performance indicator in its fitness function.

IBEA uses binary performance indicator 'I' which compare the difference in performance between two solutions, these are outlined in 2.5. The indicator also has to be dominance preserving, which by definition means that solution **a** dominates solution **b** implies I(**a**,**b**) is less than I(**b**,**a**), as well if solution **a** dominates solution **b** then this implies that I(**c**,**a**) > I(**c**,**b**) for all solution **a**, **b** and **c** in the solution set. By having the dominance preserving property a fitness function **F** can be defined such that if solution **a** dominates solution **b**, this implies that **F**(**a**) > **F**(**b**). This result allows IBEA to have a simple environmental selection process whereby the solution with the lowest fitness value is removed. Two binary performance indicators which fit these criteria and implemented in IBEA are the epsilon and hypervolume binary indicators.

The underlying EA used by IBEA employs a binary tournament selection scheme for choosing parents, and uses both recombination and mutation operators to generate offspring. Experiments of IBEA [6] with test problems DTLZ2, DTLZ6, ZDT6 and EXPO2 all show IBEA winning significantly over SPEA2 and NSGA-II.

3 Test problems

In order to better solve real world problems we must have a detailed understanding of problem characteristics, this section aims investigates these characteristics in detail. As well we look at the usefulness of test problems in reflecting fitness landscape features, and as benchmarks for the performance testing of MOEAs. We then investigate several popular test problem packages. This section follows the methodology outlined in [10], which provides a very thorough analysis.

3.1 Fitness Landscape

The fitness landscape of a test problem is defined by the mapping of objective parameters to the fitness space. Different multiobjective optimisation problems can be characterised by specific features which exist in the fitness landscape, these features represent different hurdles during solution search.

One example of a feature for a fitness landscape is whether the mapping between parameter vectors and objective vectors is many-to-one or one-to-one. The former case is named a many-to-one fitness landscape whilst the latter a one-to-one fitness landscape. A many-to-one fitness landscape presents more problems during solution search, as because many solutions map to a single objective vector, it is unclear which solution to chose. A special case of a many-to-one fitness landscape exists if a joint and open subset of parameter vectors map to one objective vector, such a feature is named a flat region where small alterations of parameter vectors have no effect on the objective vector, the lack of gradient information provides a hurdle for MOEAs.

Another feature of optimisation problems is modality. A problem with an objective function which contains one optimum is unimodal, whilst an objective function containing more than one optima is multimodal. A deceptive feature is a special case of modality as defined by Deb[24], where there exists at least two optima with one being the true optima and the other a deceptive optima, see Figure 3 [10]. During solution search an algorithm can be deceived by getting stuck at deceptive optimums.

Contrary to flat regions where there is no gradient information available, with deceptive objective functions incorrect gradient information can provide a major hurdle for MOEAs.

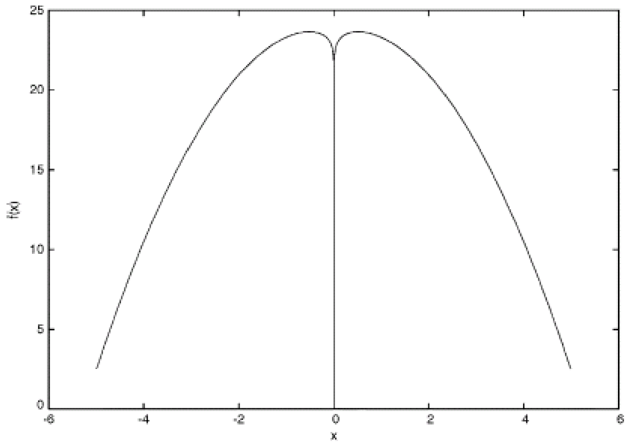


Figure 3: Example of a deceptive and multimodal objective, the true optima is at x=0 and deceptive optima are at the two sides, taken from [10]

The characteristics of a fitness landscape can also be described by the degree of evenness of the distribution for a set of objective vectors mapped from an evenly distributed set of parameter vectors. When there is a significant difference in the distribution from mapping search space to fitness space, then this problem can be described to have a biased characteristic. Bias is hard to define mathematically, but can be clearly seen by graphing parameter vectors in fitness space, see Figure 4 [10]. Unlike the previous two characteristics which are hurdles to solution search, in problems with significant bias the mapping between the Pareto optimal set and the Pareto optimal front is also biased. It is then difficult to determine whether to optimize evenly distributed solutions in the search space or in the fitness space.

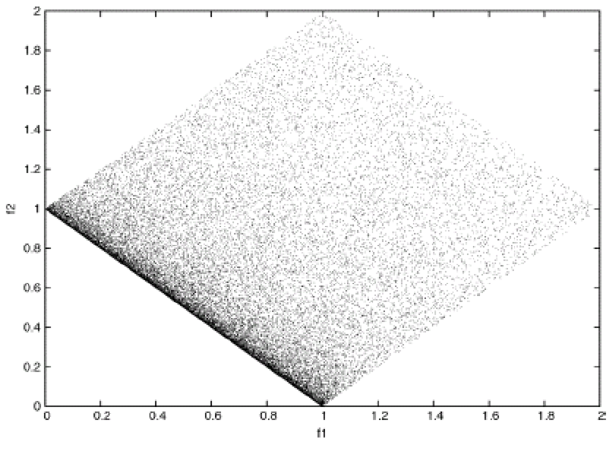


Figure 4: Example of a biased landscape, the objective vector that corresponds to 40 000 randomly selected parameter vectors from the biased two objective problem, the objective vectors are denser toward the Pareto optimal front at f1=f2=1. [10]

Another important problem feature is whether the objectives are separable, or whether parameter dependencies exist in the optimisation problem. A definition for objective-separability is given in [10], where given a single objective O, a parameter x, and index i, we define a derived problem $P_{o,x,i}$ as the problem of optimising O by varying only x_i. This is a single objective problem with a single parameter. Also define $P^*_{O,x,i}$ to be the set of global optima in parameter space for each subproblem. If $P^*_{O,x,i}$ is the same for all values of x, then x_i is separable on O. Otherwise x_i is nonseparable on O. A problem where all its objectives are separable is termed a separable problem, otherwise it is a non-separable problem.

Separable multiobjective problems are easier to optimise. By considering each objective separately, then taking the cross-product of the optimal sets for each optimised parameter gives the global optimal parameter vector.

The relationship between individual parameters and the fitness landscape is affected by the classification of the parameter. For a given problem, these relations allow us to distinguish between the convergence of solution sets and the spread of solution sets. A parameter is classified as a distance parameter iff given all parameter vectors **a**, modifying a parameter x_i in **a** results in a new parameter vector that dominates **a**, is equivalent to **a**, or is dominated by **a** [10], see Figure 5 [10]. Another classification is termed a position parameter, which differs from the distance parameter whereby modifying a parameter x_i in **a** can only results in a parameter vector that is incomparable or equivalent to **a** [10].

In multiobjective problems, Pareto optimal fronts are no longer a single point as in single objective problems, instead they become a large collection of mixed geometries. A set is defined as *concave* iff it covers a concave shape, and similarly a set is defined as *convex* iff it covers a convex shape, see Figure 6 [10]. A set is defined as *strictly concave* if it covers a concave shape but not a convex shape, and conversely a set is defined as *strictly convex* if it covers a convex shape but not a concave shape. When a set exhibits both concave and convex shapes it is defined as a *linear* set. When a solution front is a combination of connected subsets which are individually and differing strictly convex, strictly concave, or linear, this is termed a *mixed front*.

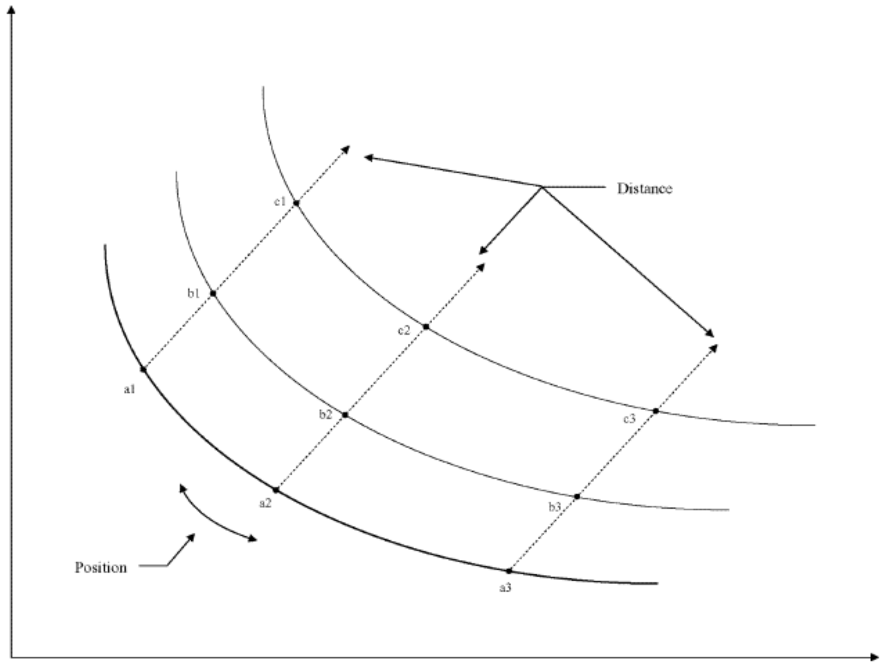


Figure 5: The difference between distance and position. Objective vectors labeled with the same letter occur at a different position on the same non-dominated front, whereas objective vectors labeled with the same number occur at different distances to the Pareto optimal front (at the same position on different fronts). [10]

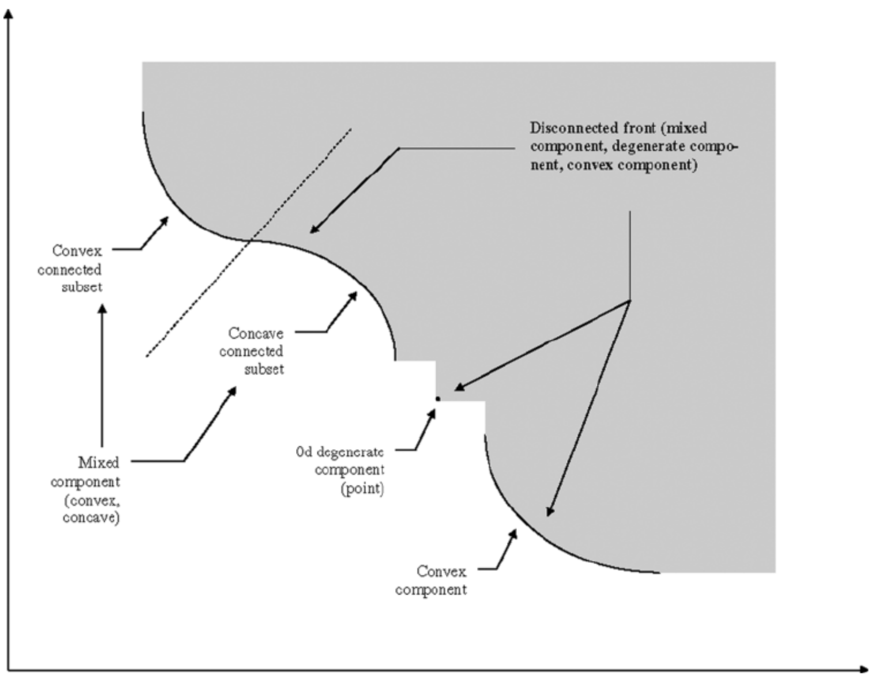


Figure 6: Sample geometry of a disconnected, mixed front that consists of: a half-convex half-concave component, a degenerate zero dimensional point, and a convex component. [10]

A degenerate front (see Figure 7 [10]) is a front that is lower dimension than the dimension of the objective space of the problem, minus one. A three dimensional problem containing a front consisting of a line segment is an example of a degenerate problem, however a three dimensional problem containing a two dimensional front no longer contains a degenerate front. During multi-objective search a degenerate front can cause problems with algorithms which attempt to maximise spread, but not adapt to the unexpected lower dimension of the front. Another feature of Pareto optimal geometry is whether the solution front is *connected* set. A set is defined as *discontinuous* if the set is disconnected.

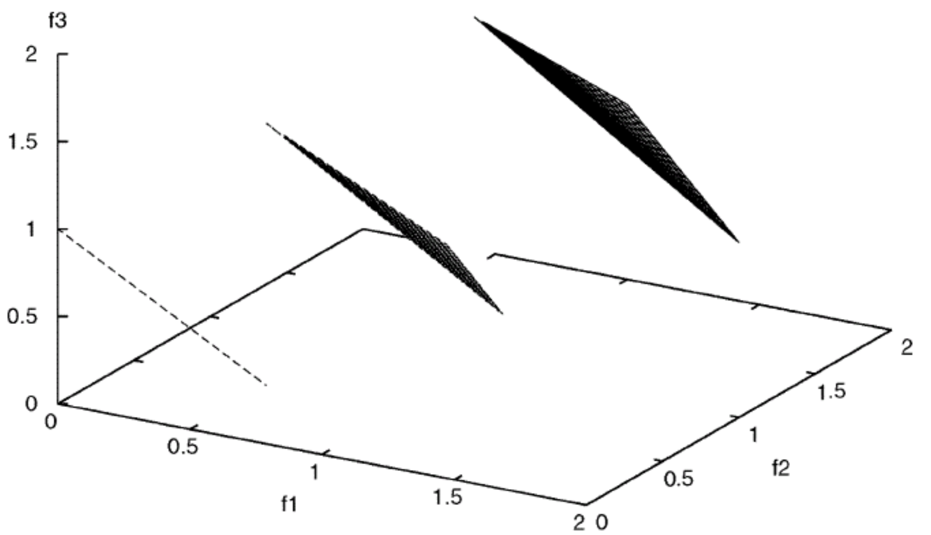


Figure 7: Three different non-dominated fronts taken from a degenerate problem. The front closest to the origin is the Pareto optimal front and is a degenerate line. [10]

3.2 Test Problems

Test problems are more useful than real world problems for comparison of MOEAs [10]. We are interested in investigating the construction and characteristics of test problems, test problems allow us to gain more insight into how better to solve real world problems, as well as evaluate the performance of MOEAs. By doing the latter, we can determine specifically which types of problem features present more difficulties for a MOEA, or which MOEAs are better suited to solving problems with a specific set of features, this way former is also fulfilled.

We will categorise the various features of the fitness landscape similar to that in [10] as shown in Table 1. This allows us to more easily classifying test problem features.

Table 1: List of test possible problem features.	
F1 Pareto optimal geometry	Combination of: Concave, convex, mixed, linear, degenerate, disconnected.
F2 Parameter dependency	Separable or non-separable
F3 Bias	Biased or unbiased
F4 Many-to-one mappings	Pareto one-to-one or many-to-one, flat regions, isolated optima
F5 Modality	Uni-modal, multi-modal, or deceptive multi-modal

3.3 Test Problem Suites

Test problem suites should aim to include a variety of test problems that collectively capture a wide variety of characteristics, this is not easy to achieve in the multiobjective domain [10]. The many possible combinations of features listed in Table 1 make it hard to capture all Pareto geometries, and existing test problem suites struggle with this. In this section we will study prominent work by [24], [22] and [25].

3.3.1 Deb's Toolkit

Introduced in [24], Deb's toolkit for two objective test problems aims to provide the user flexibility in constructing custom test problems. Constructing a problem using Deb's toolkit is accomplished by selecting three functions f_1, g and h. The f_1 function represents the distribution function and affects an algorithm's ability find a diverse solution front. The g function is the distance function which tests how well an algorithm can converge to the Pareto optimal front. Finally the h function reflects the shape function for the Pareto optimal front.

One advantage of Deb's toolkit is its flexibility in the types of test problems which can be constructed. It also makes it easy for practitioners to construct test problems of varying characteristics by separating the construction of the problem into three functions. Deb has also provided a list of example functions as shown in [10] and listed in appendix A1. Another advantage is that Deb's toolkit is the first to use separated distance and position parameters, as in many problems mixed parameters are uncharacteristic. In this way Pareto optimal solutions can be obtained by minimizing the g function, which is dependent only upon the distance parameters. As a result of this separation Deb's toolkit makes it easy to conduct analysis and the determination of the Pareto optimal front, where finding the minimal value for the g function can derive the Pareto optimal front in parametric form [10].

Deb's toolkit also have several drawbacks, the most significant one being that it is only able to construct test problems of two objectives, many real world problems have more than two objectives, and it is ideal to be able to test the performance of MOEAs in higher dimensions. Deb's toolkit also lacks the ability to incorporate important test problem features such as flat regions and deceptive landscapes.

3.3.2 ZDT Test Suite

The ZDT test suite is introduced in [22] by Zitzler and is widely employed as a benchmark test suite in the EA literature. The test suite contains 6 problems ZDT1-ZDT6, as shown in [10] and listed in appendix A2, with ZTD5 omitted as it is binary encoded. ZDT1 and ZDT4 both use convex shape functions, ZTD2 and ZTD6 use the concave shape function, whilst ZTD3 uses a disconnected shape both in the Pareto optimal set and front. All ZDT test problems are separable and ZDT3, ZTD4 and ZTD6 are multimodal. ZTD6 also contains bias and has a many-to-one mapping. The ZDT test suite contains a good variety of shape functions but lack features such as non-separability, an analysis of these features are listed in Table 2 from [10], which displays for each test problem the presence of features F1-F5 from Table 1, R1-R5 represent test problem recommendations put forth by [10].

Table 2: Analysis of ZTD Problems. [10]

Name	Objective	R3: # Parameters	F2: Separability	F5: Modality	R1: No Extremal	R2: No Medial	R5: Diss. Domains	R6: Diss. Ranges	R7: Optima Known	F1: Geometry	F3: Bias	F4a: Pareto Many-to-one	F4b: Flat Regions
ZDT1	f_1	1	S	U	×	✓	×	×	✓	convex	−	−	−
	f_2	✓	S	U									
ZDT2	f_1	1	S	U	×	✓	×	×	✓	concave	−	−	−
	f_2	✓	S	U									
ZDT3	f_1	1	S	U	×	✓	×	✓	✓	disconnected*	−	−	−
	f_2	✓	S	M									
ZDT4	f_1	1	S	U	✓	×	×*	×	✓	convex	−	−	−
	f_2	✓	S	M									
ZDT6	f_1	1	S	M	×	✓	×	✓	✓	concave	+	+	−
	f_2	✓	S	M									

The ZDT test suite can almost entirely be constructed using Deb's toolkit and it also shares many of its advantages and disadvantages. All of the ZDT test problems use only one position parameter, which implies that the f_1 function only contains one parameter. Similar to Deb's toolkit, the advantage of the ZDT test suite is that it has well defined Pareto optimal fronts, and there are many instances of test results using ZDT in the EA literature.

However, the ZDT toolkit still only allows test problems of two objectives, and is not scalable in higher dimensions. Its test problems also lack important landscape features like flat regions and non-separability. Although the ZDT toolkit is extensively used as a MOEA benchmark, it does not cover a wide range of test problem characteristics.

3.3.3 DTLZ Test Suite

The more recent DTLZ test suite is detailed in [25] by Deb et al, an important innovation of this test suite is that its test problems are scalable in any number of objectives. This facilitates the benchmarking of MOEAs in higher objectives which greatly increases the usefulness of test problems as a tool for the study of MOEAs. As well DTLZ1-DTLZ6 are scalable in the number of distance parameters, position parameters are fixed to M – 1 where M is the number of objectives.

The DTLZ original test suite contains 9 test problems, DTLZ8 and DTLZ9 contain side constraints and are not covered, appendix A3 [10] provides a list of DTLZ test problem functions. The DTLZ test suite contains a variety of shape functions, DTLZ1 is linear shape, DTLZ2-4 uses a concave shape function, DTLZ5-DTLZ6 has degenerate shape (however according to [10] this falls apart after 3 dimensions), and DTLZ7 uses a disconnected shape function. Table 3 from [10] lists a feature analysis of DTLZ1-DTLZ7. DTLZ1, DTZL3 and DTLZ7 contain multimodality. DTLZ1-DTLZ6 all have many-to one mappings, as well DTLZ4 and DTLZ6 contain bias. The DTLZ test suite offers a moderate coverage of Pareto geometries and coverage of features. However all DTLZ1-DTLZ7 problems lack the non-separability feature.

Table 3: Analysis of DTLZ problems [10].

Name	Objective	R3: # Parameters	F2: Separability	F5: Modality	R1: No Extremal	R2: No Medial	R5: Diss. Domains	R6: Diss. Ranges	R7: Optima Known	F1: Geometry	F3: Bias	F4a: Pareto Many-to-one	F4b: Flat Regions
DTLZ1	$f_{1:M}$	✓*	S*	M	✓	×	×	×	✓	linear	−	+	−
DTLZ2*	$f_{1:M}$	✓*	S*	U	✓	×	×	×	✓	concave	−	+	−
DTLZ3	$f_{1:M}$	✓*	S*	M	✓	×	×	×	✓	concave	−	+	−
DTLZ4	$f_{1:M}$	✓*	S*	U	✓	×	×	×	✓	concave	+	+	−
DTLZ5*	$f_{1:M}$	✓*	?	U	?	?	×	?	×	?	−	+	−
DTLZ6*	$f_{1:M}$	✓*	?	U	?	?	×	?	×	?	+	+	−
DTLZ7	$f_{1:M-1}$	1	NA	U	×	✓	×	×*	✓	disconnected*	−	−	−
	f_M	✓	S	M									

DTLZ test suite differs from the original Deb's toolkit in that the distance function now influence all objective functions, objective functions are no longer dependent only on position parameters. The advantages of the DTLZ are quite clear, it offers the ability to test beyond two objectives. And similarly to Deb's toolkit, in DTLZ we are clear about problem characteristics and information of the Pareto optimal front.

Despite the advantages DTLZ has to offer, it still lacks important problem features such as flat regions, deceptiveness and in particular, non-separability. It also has problems with the location of optimal solutions, some are at the origin which makes it easy for “averaging” algorithms where solutions averaged to the origin find the optima, and some optima are at the extrema which are good for “capping” algorithms where solutions can overshoot and get capped back to the optima.

DTLZ allows flexibility in the number of objectives but loses flexibility with problem features and problem construction, which was achievable in Deb's toolkit. For the purposes of this project we require a test problem toolkit which offers flexibility with respect to the number of objectives and problem construction.

3.4 WFG toolkit

The WFG toolkit is introduced in [26], aims to provide improvement on the quality and rigor of MOEA testing by providing researchers with the ability to construct test problems with easily customisable features, and remain scalable in the number of objectives.

The WFG toolkit defines a problem [26] in terms of an underlying vector of parameters x, where x is always associated with a simple underlying problem that defines fitness space. The vector x is determined through a series of transition vectors, from the vector of working parameters z. Each transition vector increases complexity of the underlying problem, by adding problem features such as the multimodality and non-separability. The MOEA directly manipulates z, through which x is indirectly manipulated.

The WFG toolkit allows practitioners the ability to control the features present in a test problem through a series of layered transformations. To construct a new test problem, a shape function is first chosen for the Pareto optimal geometry, then use a number of transformation functions to facilitate the creation of transition vectors. Transformation functions must be carefully designed such that the underlying Pareto optimal front remains intact relatively easily with the Pareto optimal set. The WFG toolkit provides a set of predefined shape and transformation functions to ensure this is the case.

Shape functions in the WFG toolkit determine the Pareto optimal geometry, it also maps input parameters with domain [0,1] onto range [0,1]. Five predefined shape functions presented in [10] are listed in Appendix A4. Note that each of h1:M must be associated with a shape function. For example, if all h1:M is associated with the linear shape function, then the Pareto optimal front becomes a linear hyperplane.

Table 4: Example WFG test problems. [10]

THE WFG TEST SUITE. THE NUMBER OF POSITION-RELATED PARAMETERS k MUST BE DIVISIBLE BY THE NUMBER OF UNDERLYING POSITION PARAMETERS, $M-1$ (THIS SIMPLIFIES REDUCTIONS). THE NUMBER OF DISTANCE-RELATED PARAMETERS l CAN BE SET TO ANY POSITIVE INTEGER, EXCEPT FOR WFG2 AND WFG3, FOR WHICH l MUST BE A MULTIPLE OF TWO (DUE TO THE NATURE OF THEIR NONSEPARABLE REDUCTIONS). TO ENHANCE READABILITY, FOR ANY TRANSITION VECTOR $\mathbf{t}^i$, WE LET $\mathbf{y} = \mathbf{t}^{i-1}$. FOR $\mathbf{t}^1$, LET $\mathbf{y} = \mathbf{z}_{[0,1]} = \{z_1/2, \ldots, z_n/(2n)\}$

Problem	Type	Setting
All	Constants	$S_{m=1:M} = 2m$
		$D = 1$
		$A_1 = 1$
		$A_{2:M-1} = \begin{cases} 0, & \text{for WFG3} \\ 1, & \text{otherwise} \end{cases}$
		The settings for $S_{1:M}$ ensures the Pareto optimal fronts have dissimilar tradeoff magnitudes, and the settings for $A_{1:M-1}$ ensures the Pareto optimal fronts are not degenerate, except in the case of WFG3, which has a one dimensional Pareto optimal front.
All	Domains	$z_{i=1:n,\max} = 2i$
		The working parameters have domains of dissimilar magnitude.
WFG1	Shape	$h_{m=1:M-1} = \text{convex}_m$
		$h_M = \text{mixed}_M$ (with $\alpha = 1$ and $A = 5$)
	$\mathbf{t}^1$	$t^1_{i=1:k} = y_i$
		$t^1_{i=k+1:n} = \text{s\_linear}(y_i, 0.35)$
	$\mathbf{t}^2$	$t^2_{i=1:k} = y_i$
		$t^2_{i=k+1:n} = \text{b\_flat}(y_i, 0.8, 0.75, 0.85)$
	$\mathbf{t}^3$	$t^3_{i=1:n} = \text{b\_poly}(y_i, 0.02)$
	$\mathbf{t}^4$	$t^4_{i=1:M-1} = \text{r\_sum}(\{y_{(i-1)k/(M-1)+1}, \ldots, y_{ik/(M-1)}\}, \{2((i-1)k/(M-1)+1), \ldots, 2ik/(M-1)\})$
		$t^4_M = \text{r\_sum}(\{y_{k+1}, \ldots, y_n\}, \{2(k+1), \ldots, 2n\})$
WFG2	Shape	$h_{m=1:M-1} = \text{convex}_m$
		$h_M = \text{disc}_M$ (with $\alpha = \beta = 1$ and $A = 5$)
	$\mathbf{t}^1$	As $\mathbf{t}^1$ from WFG1. (Linear shift.)
	$\mathbf{t}^2$	$t^2_{i=1:k} = y_i$
		$t^2_{i=k+1:k+l/2} = \text{r\_nonsep}(\{y_{k+2(i-k)-1}, y_{k+2(i-k)}\}, 2)$
	$\mathbf{t}^3$	$t^3_{i=1:M-1} = \text{r\_sum}(\{y_{(i-1)k/(M-1)+1}, \ldots, y_{ik/(M-1)}\}, \{1, \ldots, 1\})$
		$t^3_M = \text{r\_sum}(\{y_{k+1}, \ldots, y_{k+l/2}\}, \{1, \ldots, 1\})$
WFG3	Shape	$h_{m=1:M} = \text{linear}_m$ (degenerate)
	$\mathbf{t}^{1:3}$	As $\mathbf{t}^{1:3}$ from WFG2. (Linear shift, non-separable reduction, and weighted sum reduction.)
WFG4	Shape	$h_{m=1:M} = \text{concave}_m$
	$\mathbf{t}^1$	$t^1_{i=1:n} = \text{s\_multi}(y_i, 30, 10, 0.35)$
	$\mathbf{t}^2$	$t^2_{i=1:M-1} = \text{r\_sum}(\{y_{(i-1)k/(M-1)+1}, \ldots, y_{ik/(M-1)}\}, \{1, \ldots, 1\})$
		$t^2_M = \text{r\_sum}(\{y_{k+1}, \ldots, y_n\}, \{1, \ldots, 1\})$
WFG5	Shape	$h_{m=1:M} = \text{concave}_m$
	$\mathbf{t}^1$	$t^1_{i=1:n} = \text{s\_decept}(y_i, 0.35, 0.001, 0.05)$
	$\mathbf{t}^2$	As $\mathbf{t}^2$ from WFG4. (Weighted sum reduction.)
WFG6	Shape	$h_{m=1:M} = \text{concave}_m$
	$\mathbf{t}^1$	As $\mathbf{t}^1$ from WFG1. (Linear shift.)
	$\mathbf{t}^2$	$t^2_{i=1:M-1} = \text{r\_nonsep}(\{y_{(i-1)k/(M-1)+1}, \ldots, y_{ik/(M-1)}\}, k/(M-1))$
		$t^2_M = \text{r\_nonsep}(\{y_{k+1}, \ldots, y_n\}, l)$
WFG7	Shape	$h_{m=1:M} = \text{concave}_m$
	$\mathbf{t}^1$	$t^1_{i=1:k} = \text{b\_param}(y_i, \text{r\_sum}(\{y_{i+1}, \ldots, y_n\}, \{1, \ldots, 1\}), \frac{0.98}{49.98}, 0.02, 50)$
		$t^1_{i=k+1:n} = y_i$
	$\mathbf{t}^2$	As $\mathbf{t}^1$ from WFG1. (Linear shift.)
	$\mathbf{t}^3$	As $\mathbf{t}^2$ from WFG4. (Weighted sum reduction.)
WFG8	Shape	$h_{m=1:M} = \text{concave}_m$
	$\mathbf{t}^1$	$t^1_{i=1:k} = y_i$
		$t^1_{i=k+1:n} = \text{b\_param}(y_i, \text{r\_sum}(\{y_1, \ldots, y_{i-1}\}, \{1, \ldots, 1\}), \frac{0.98}{49.98}, 0.02, 50)$
	$\mathbf{t}^2$	As $\mathbf{t}^1$ from WFG1. (Linear shift.)
	$\mathbf{t}^3$	As $\mathbf{t}^2$ from WFG4. (Weighted sum reduction.)
WFG9	Shape	$h_{m=1:M} = \text{concave}_m$
	$\mathbf{t}^1$	$t^1_{i=1:n-1} = \text{b\_param}(y_i, \text{r\_sum}(\{y_{i+1}, \ldots, y_n\}, \{1, \ldots, 1\}), \frac{0.98}{49.98}, 0.02, 50)$
		$t^1_n = y_n$
	$\mathbf{t}^2$	$t^2_{i=1:k} = \text{s\_decept}(y_i, 0.35, 0.001, 0.05)$
		$t^2_{i=k+1:n} = \text{s\_multi}(y_i, 30, 95, 0.35)$
	$\mathbf{t}^3$	As $\mathbf{t}^2$ from WFG6. (Non-separable reduction.)

Transformation functions in the WFG toolkit also map input parameters with domain [0,1] to range [0,1]. All transformations map a vector of parameters (called the primary parameters) and map them to a single value. Transformation functions may also use constants and secondary parameters that further influence the mapping. A set of predefined transformation functions [10] are shown in Appendix A5.

The bias transformation functions introduce bias to the fitness landscape. Shift transformation functions shift the location of the optima. In the absence of shift transformations, all distance-related parameters become extremal parameters with an optimal value of zero. To avoid extremal parameters [26] recommends the use of at least one shift transformation function. Both the bias and shift transformations can use only one primary parameter. This is different to reduction transformations which can use many parameters as opposed to just one primary parameter. By introducing additional secondary parameters, dependencies can be produced between position and distance-related parameters, increasing the non-separability of the problem. See Appendix A5 for a list of transformation function restrictions [10]

3.5 Example WFG Test Suite: WFG1-WFG9

The example WFG test suite includes well-designed [26] test problems with a variety of characteristics which aim to provide a thorough benchmark for MOEAs. The WFG1-WFG9 problems [10] are listed in Table 4. The WFG test suite use a variety of shape functions, WFG1 uses a convex and mixed shape function, WFG2 uses a convex and disconnected shape function, WFG3 has Pareto optimal geometry which is linear and degenerate, and WFG4-WFG9 all use the concave shape function.

The WFG test problems also employ a wide variety of features including multimodality, deceptiveness, bias and non-separability. Further we're able to easily turn these features on and off through the addition or removal of feature transitions within the toolkit. As such the WFG toolkit is well suited for the purposes of this project.

3.6 PISA

Many different test problems and MOEAs have been developed in the EA literature, researchers require understanding of the design of MOEAs as well as programming knowledge to integrate them with test problems, which can be a very time consuming task. A Platform and programming language independent Interface for Search Algorithms (PISA) framework proposed in [23] aims to design a standardized, extensible and easy to use framework for the implementation and testing of MOEAs. PISA was designed with the goals of simplicity and small overhead, in order to realise an easy and readily applicable test framework.

One feature of the PISA framework is the separation of selectors (MOEAs) and variators (problems) as shown in Figure 8 [23]. This is very useful as the modular design allows researchers to "plug in" new MOEAs and test problems. Another important feature is in the implementation of communication between variator and selector, under PISA this is done entirely through the file system, which allows for platform, programming language and operating system independence.

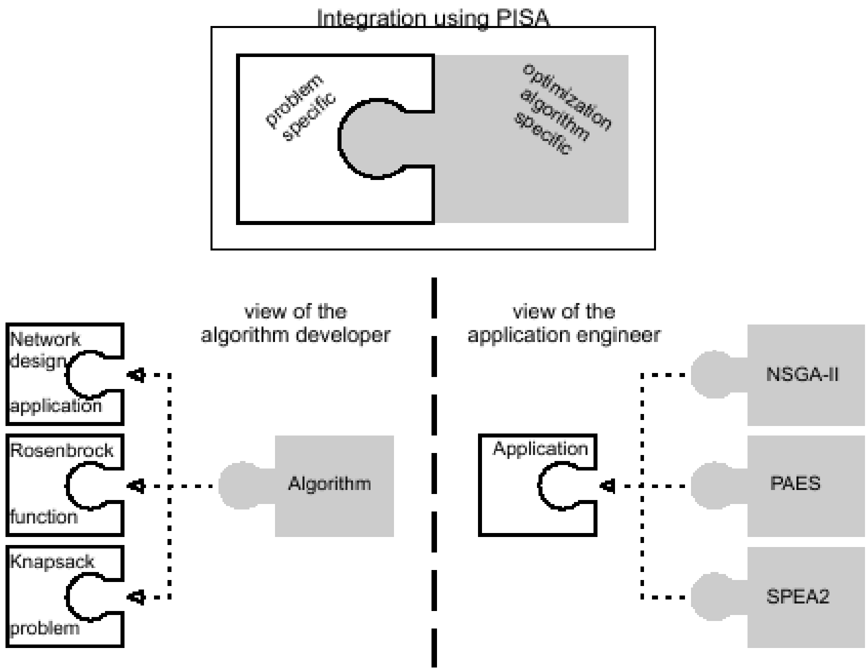


Figure 8: Integration using PISA [23].

Control flow during execution in PISA is achieved through a state system, in this model there is a consistent state for the entire optimisation process, and only one module is active at one time. The basic states in PISA range from zero to three, as shown in Figure 9 [23], in state zero and one the variator and selector are initialised, respectively. The main optimisation process occurs through iterations of states two and three, first the selector chooses a set of parent individuals and sends them to the variator, the variator then generate new offspring solutions in accordance with the set of parent individuals, the offspring solutions are then passed back to the selector. Higher states are responsible for terminating the selector and variator, and resetting them.

PISA comes with a wide range of ready implemented selectors and variators, including the MOEAs IBEA, NSGA-II and SPEA2. It also includes performance and statistical testing modules, including the performance indicators epsilon, hypervolume and R. Statistical tests such as the Mann-Whitely U test are also included. The PISA framework provides the ideal environment for MOEA testing, and is a suitable testbed for our experiments.

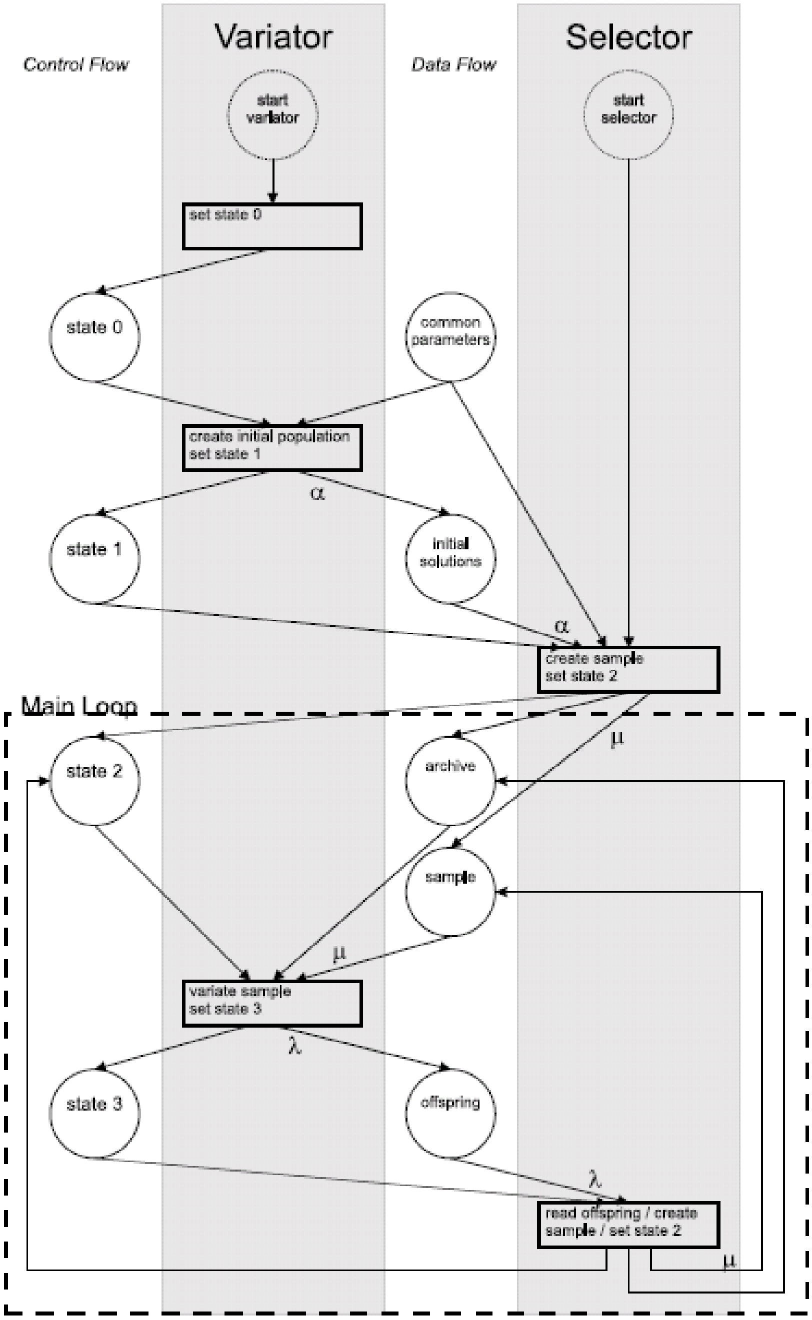


Figure 9: State transition diagram for states 0-3 in PISA, taken from [23].

4 Experiments and Results

In this section we will run extensive testing to compare the performance of MOEAs and investigate which class of features present difficulties during optimisation, and aim to demonstrate the “no free lunch” theorem where there does not exist a generally superior algorithm, by systematically varying test problem features we hope to construct test problems where strong performing algorithms will become weak. The process for comparing MOEAs involves:

1. Choose MOEAs to compare
2. Choose criteria to compare MOEAs
3. Choose test problems to use
4. Execute problems using MOEAs and obtain results
5. Compare results
6. Draw conclusion

For our experiments we will choose three popular MOEAs to compare, the NSGAII, SPEA2 and IBEA (using hypervolume binary indicator as selection mechanism). The performance of MOEAs will be judged by using three performance indicators, the epsilon, hypervolume and R metric (we use the R2 metric in experiments). These indicators take certain features of a final solution front and provide a single value evaluation for performance. We note that using hypervolume as a performance metric is somewhat unfair as IBEA uses the hypervolume indicator within its selection mechanism, however we believe hypervolume is a quality metric which is favoured by many people [19], [27] and [28].

The MOEAs are run on test problems from the WFG toolkit, as it allows the user to easily change test problem transformations functions as well as shape functions, this is very useful for the purposes of our experiments, by observing the performance characteristics of MOEAs towards test problems with certain features we can hypothesis about which features pose difficulties for the MOEA and adjust those features in the next experiment.

Execution of problems is done under the PISA framework, where the MOEAs NSGAII, SPEA2 and IBEA are available and standard implementations as selectors in PISA. The WFG toolkit is also set up under the PISA framework as the variator. PISA provides performance assessment of selectors and we used the implementations of the epsilon, hypervolume and R metric under PISA. The Mann-Whitney U test is a statistical test which tests the null hypothesis that the probability distribution of two samples is equal. The Mann-Whitney U test is used to check the significance of performance indicators

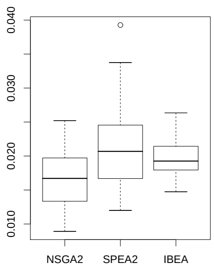
Using the PISA framework all experiments were repeated 40 times, with all selectors (MOEAs) using a (100+100) populations scheme running 750 generations. The WFG variator used a fixed 20 distance parameters and 4 position parameters, with real-parameter SBX crossover with probability 1.0 and η-recombination equals to 10. Variable-wise polynomial mutation probability is set to 1/n where n equals the sum of distance and position parameters. In our experiments we do not vary the distance and position parameters which can increase problem difficulty, because results in [12] show it does not significantly affect overall results.

After each batch of experiments some observations are made and test problem parameters and features are modified, and we iterate through steps 3-5 to observe the behaviour of MOEAs with varying test problem features.

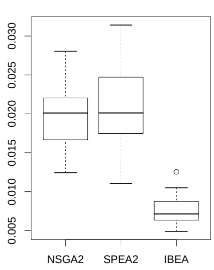
4.1 Experiment 1, 2d

1) Aim and Experimental Setup: In this experiment we aim to establish the baseline performance of MOEAs under 2 objectives, and use this as a basis for comparison in further experiments. Experiments were run with all selectors using WFG2-9 test problems under two dimensions. Through many trials we could not obtain results of the selector IBEA against variator WFG1, which consistently produced an assertion error indicating a problem with the parameters of default transitions in WFG1, as such WFG1 will not be included in further experiments.

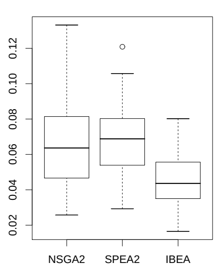
Figure 10: WFG2-WFG9, two dimensions, epsilon plots.



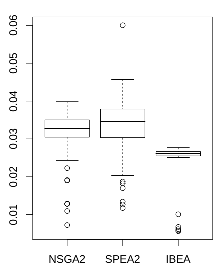
a) WFG2 epsilon



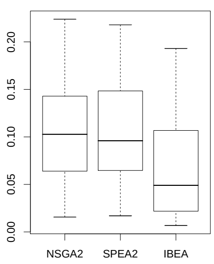
b) WFG3 epsilon



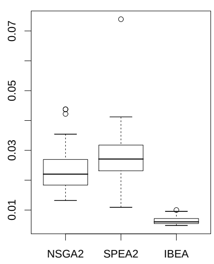
c) WFG4 epsilon



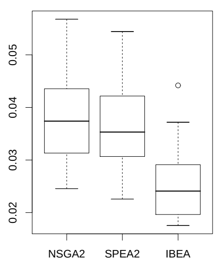
d) WFG5 epsilon



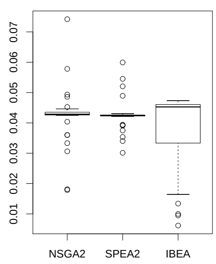
e) WFG6 epsilon



f) WFG7 epsilon

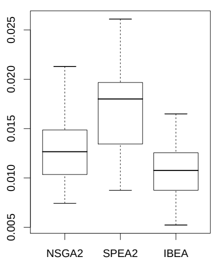


g) WFG8 epsilon

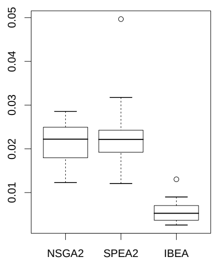


h) WFG9 epsilon

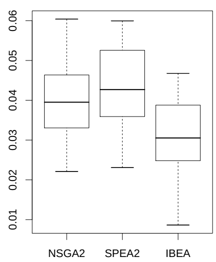
Figure 11: WFG2-WFG9, two dimensions, hypervolume plots.



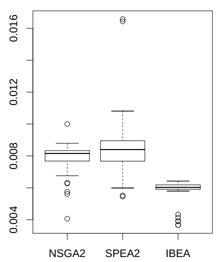
a) WFG2 hypervolume



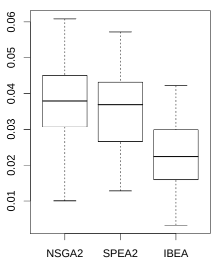
b) WFG3 hypervolume



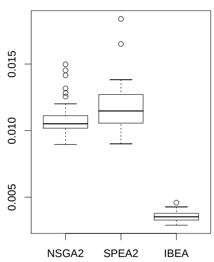
c) WFG4 hypervolume



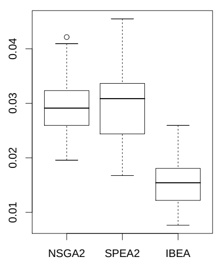
d) WFG5 hypervolume



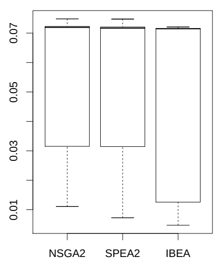
e) WFG6 hypervolume



f) WFG7 hypervolume

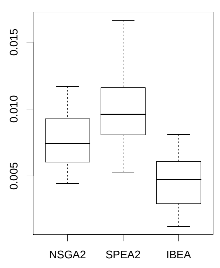


g) WFG8 hypervolume

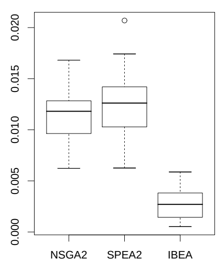


h) WFG9 hypervolume

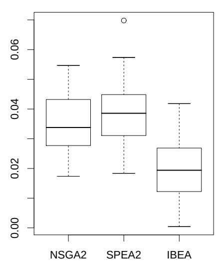
Figure 12: WFG2-WFG9, two dimensions, R plots.



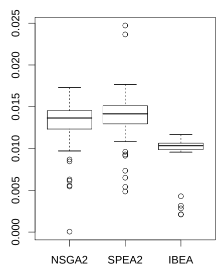
a) WFG2 R



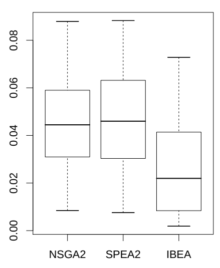
b) WFG4 R



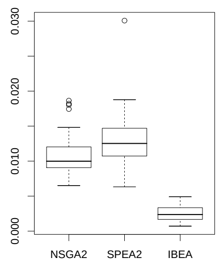
c) WFG5 R



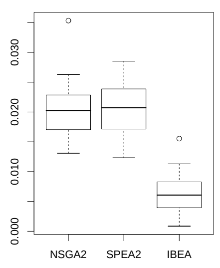
d) WFG6 R



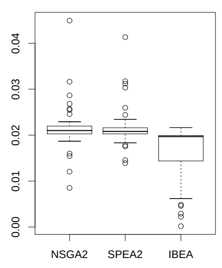
e) WFG6 R



f) WFG7 R



g) WFG8 R



h) WFG9 R

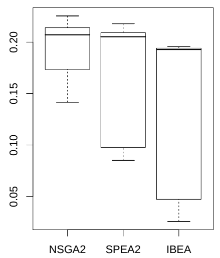
2) Results: Figure 10-12 show box-and-whisker plots for values of three performance indicators, the epsilon, hypervolume, and R metrics. Specifically it shows for each performance metric, the performance difference of each selector on each variator relative to a combined non-dominated solution front generated by all selectors, thus smaller is better. The box-and-whisker plots show the distribution of samples, the box represents the upper and lower quartiles of the sample with the middle bar representing the median, the two tails represent lowest and highest observations, with the circles considered outliers. We can see that in two dimensions IBEA is dominant winning on almost all WFG2-9 problems on all indicators against NSGA-II and SPEA2, a surprising result which seems to challenge the “no free lunch” theorem, where IBEA seems to demonstrate superiority in all problem classes tested.

In the battle for second place NSGA-II is a close winner, in most cases performing at least as good as SPEA2, the performance gap between these two MOEAs is much smaller than against IBEA. This is consistent with results from [12] showing NSGA-II performing better overall in two dimensions.

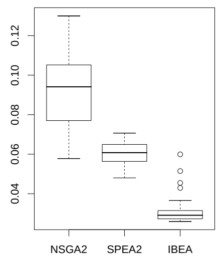
4.2 Experiment 2, 3d

1) Aim and Experimental Setup: Following the results of objectives, we aim to determine the baseline performance of MOEAs under 3 objectives, by increasing the difficulty we expect changes in the relative performance between the three MOEAs. For example Zitzler's [7] paper finds crowding distance used by NSGA-II as a diversity preservation mechanism degenerates in high dimensions, so we expect NSGA-II to perform worse as than in two dimensions. Experiments were run with all selectors using WFG2-9 test problems under three dimensions.

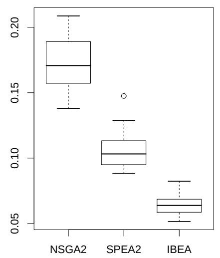
Figure 13: WFG2-WFG9, three dimensions, epsilon plots.



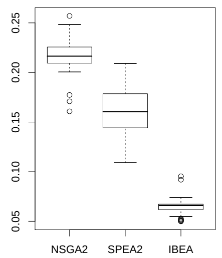
a) WFG2 epsilon



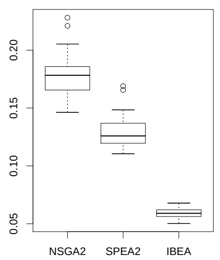
b) WFG3 epsilon



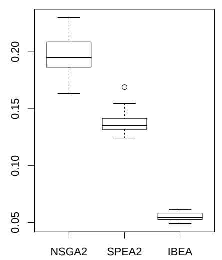
c) WFG4 epsilon



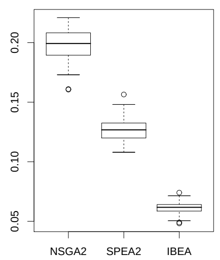
d) WFG5 epsilon



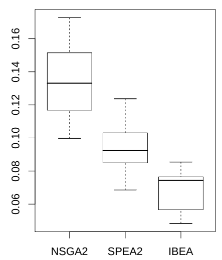
e) WFG6 epsilon



f) WFG7 epsilon

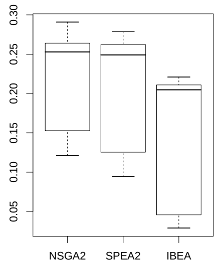


g) WFG8 epsilon

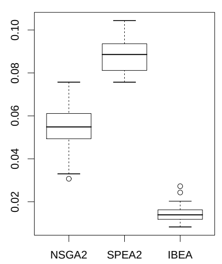


h) WFG9 epsilon

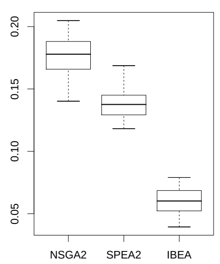
Figure 14: WFG2-WFG9, three dimensions, hypervolume plots.



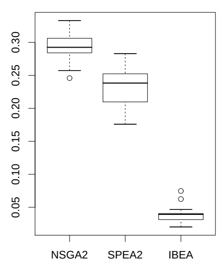
a) WFG2 hypervolume



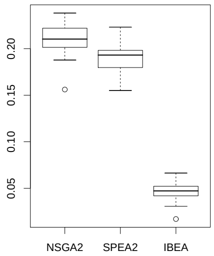
b) WFG3 hypervolume



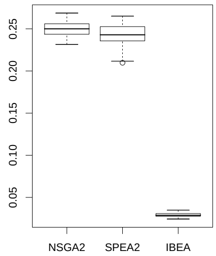
c) WFG4 hypervolume



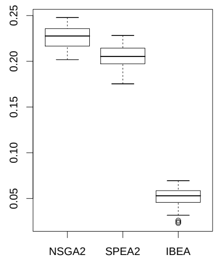
d) WFG5 hypervolume



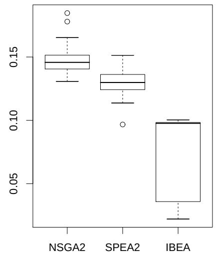
e) WFG6 hypervolume



f) WFG7 hypervolume

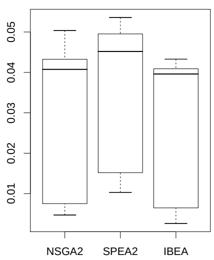


g) WFG8 hypervolume

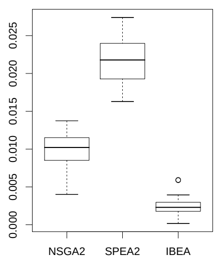


h) WFG9 hypervolume

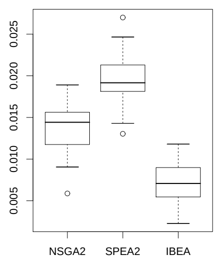
Figure 15: WFG2-WFG9, three dimensions, R plots.



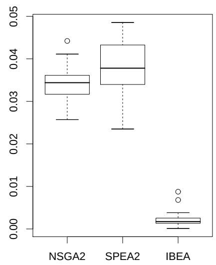
a) WFG2 R



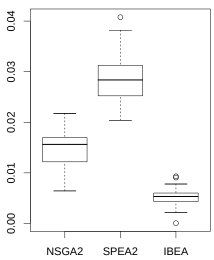
b) WFG4 R



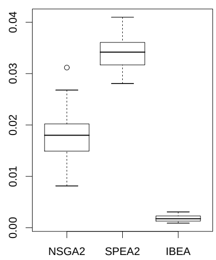
c) WFG5 R



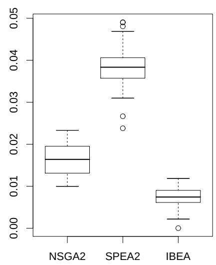
d) WFG6 R



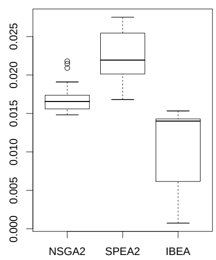
e) WFG6 R



f) WFG7 R



g) WFG8 R



h) WFG9 R

2) Results: The box-and-whisker plots in Figure 13-15 show that again in three dimensions IBEA is dominant, winning on all WFG2-9 problems on all indicators against NSGA-II and SPEA2 by a large margin, another surprising result which seems to go against the “no free lunch” theorem.

The battle for second place is again between NSGA-II and SPEA2, but this time SPEA2 is the overall winner. This result is consistent with [12] and [7], this also supports the theory that crowding distance breaks down in higher dimensions. In WFG3 we see that NSGA-II is still able to outperform SPEA2, further investigation from [12] shows the linear shape used by WFG3 favours NSGA-II, and performance can be reversed by replacing the linear shape function with concave shape function. In three dimensions the performance gap between IBEA against NSGA-II and SPEA2 is still quite large.

4.3 Experiment 3, 5d

1) Aim and Experimental Setup: Following the results of objectives, we aim to further investigate the performance of MOEAs under 5 objectives. By increasing the difficulty again we expect IBEA to maintain its lead over NSGA-II and SPEA2, and NSGA-II loses to SPEA2 due to underperformance of crowding-distance in NSGA-II. Experiments were run with all selectors using WFG2-9 test problems under five dimensions. Because of the long computation time in calculating hypervolume in higher dimensions, the hypervolume calculations were performed using a faster algorithm [20]. The calculation of the combined non-dominated front from three selectors is left out, this does not affect the overall result as the hypervolume of the combined front is fixed. Plots of the hypervolume metric show only hypervolume obtained by each selector, so higher values are better.

Figure 16: WFG2-WFG9, five dimensions, epsilon plots.

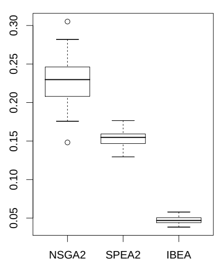
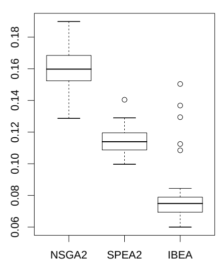
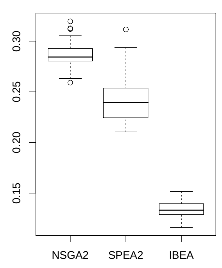
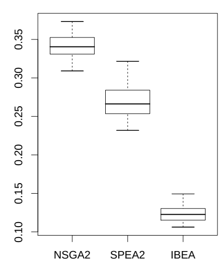
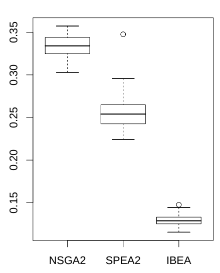
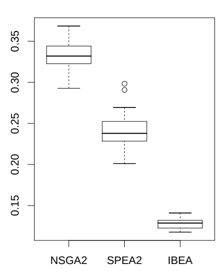
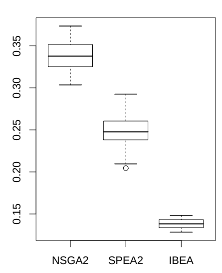
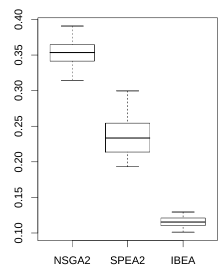
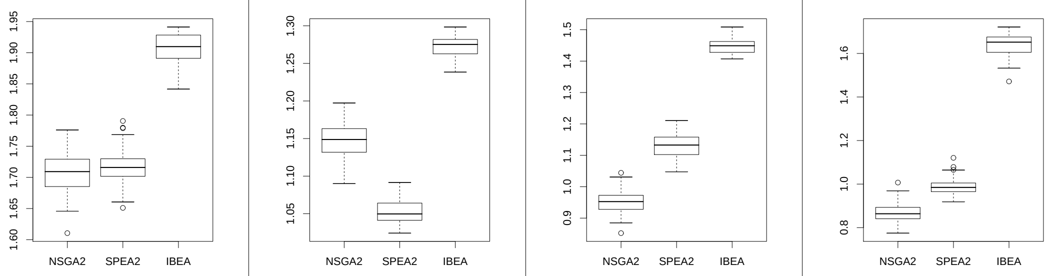
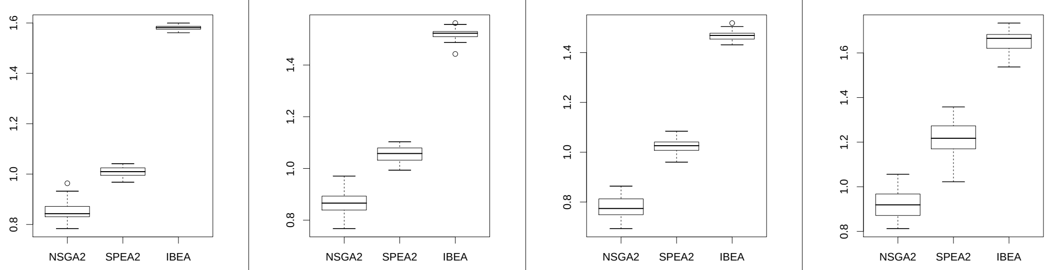
			
a) WFG2 epsilon	b) WFG3 epsilon	c) WFG4 epsilon	d) WFG5 epsilon
			
e) WFG6 epsilon	f) WFG7 epsilon	g) WFG8 epsilon	h) WFG9 epsilon

Figure 17: WFG2-WFG9, five dimensions, hypervolume plots.

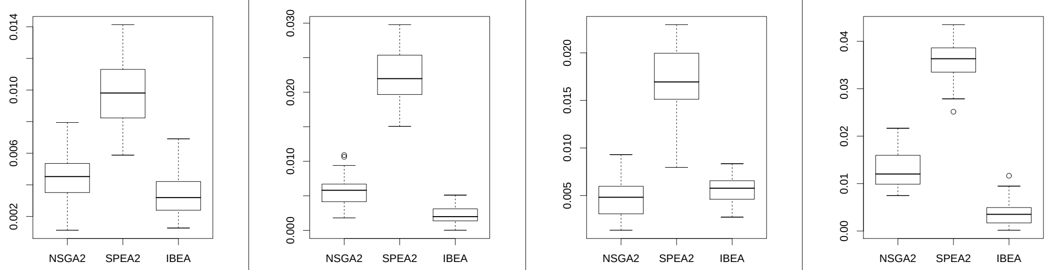


a) WFG2 hypervolume | b) WFG3 hypervolume | c) WFG4 hypervolume | d) WFG5 hypervolume

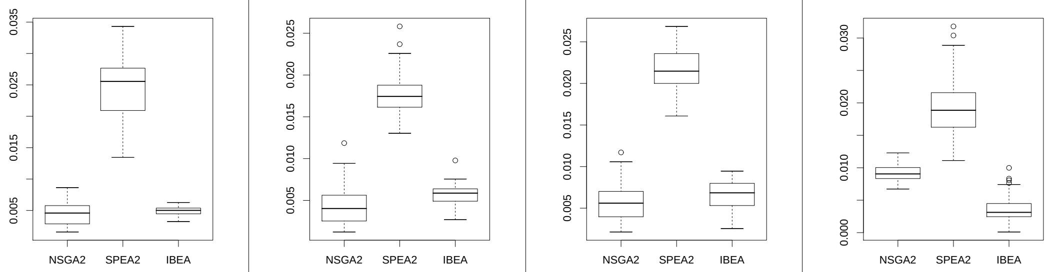


e) WFG6 hypervolume | f) WFG7 hypervolume | g) WFG8 hypervolume | h) WFG9 hypervolume

Figure 18: WFG2-WFG9, five dimensions, R plots.



a) WFG2 R | b) WFG3 R | c) WFG4 R | d) WFG5 R



e) WFG6 R | f) WFG7 R | g) WFG8 R | h) WFG9 R

2) Results: From the box-and-whisker plots in Figure 16-18 we see that for epsilon and hypervolume metrics the results mimic those of three objectives, for all problems IBEA wins over NSGA-II and SPEA2, and NSGA-II loses to SPEA2 with the exception of WFG3. This again shows that SPEA2 is stronger in higher dimensions due to the underperformance of crowding distance for NSGA-II. The R metric provides interesting results, showing SPEA2 winning over IBEA in WFG4, WFG7 and WFG8. This could be explained by a greater deviation between characteristics represented by hypervolume and R metrics in high dimensions.

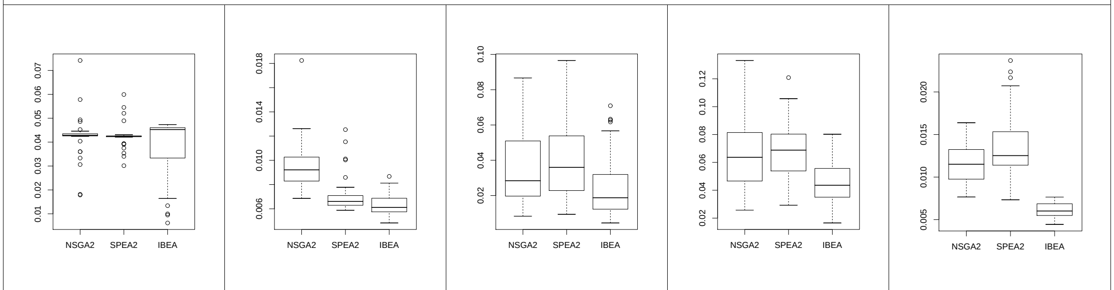
4.4 Experiment 4, Investigate transformations

1) Aim and Experimental Setup: Following the results of the baseline experiments where the performance of IBEA is dominant, we wish to investigate the effect of changing test problem features on the performance of IBEA, and gain insight into possible ways to beat IBEA as a demonstration of the "no free lunch" theorem. Due to the long run time of up to 5 hours for each test problem, it is unrealistic to exhaustive investigate transformations for all test problems. Instead we aim to make informed guesses on which transformations pose more difficulties for IBEA. We also choose to make these decisions based on the hypervolume metric, which we believe capture more performance information of a solution front as it measures the volume change as opposed to a scalar change. By examining the results of baseline 2d and 3d experiments, we can see the performance lead of IBEA on WFG2, WFG3, WFG4 and WFG9 is comparably less. Further examining the feature transformations present in these problems we see that multimodality is the only common transformation present in both WFG4 and WFG9, with WFG9 having an additional deceptive multimodality transformation. In this experiment we systematically remove multimodality transformations in WFG4 and WFG9 test problems, experiments are carried out for tests shown in Table 5 in both two and three objectives.

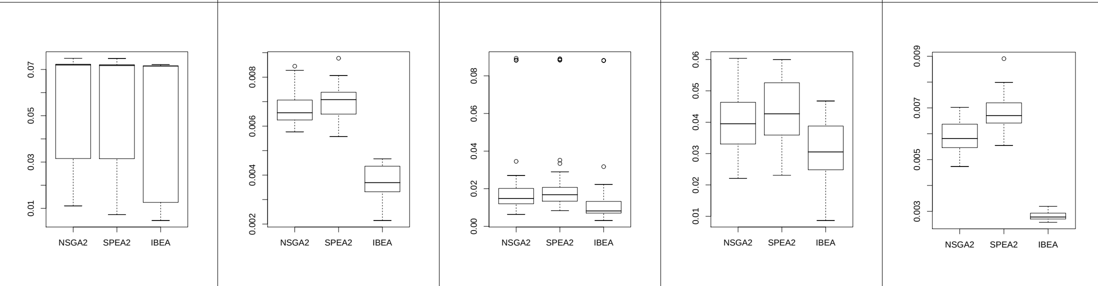
Table 5: Experiment four tests.

Test 1	WFG9, remove transformation 2 (multimodality and deceptiveness)
Test 2	WFG9, remove deceptiveness transformation in transformation 2
Test 3	WFG4, remove transformation 1(multimodality)

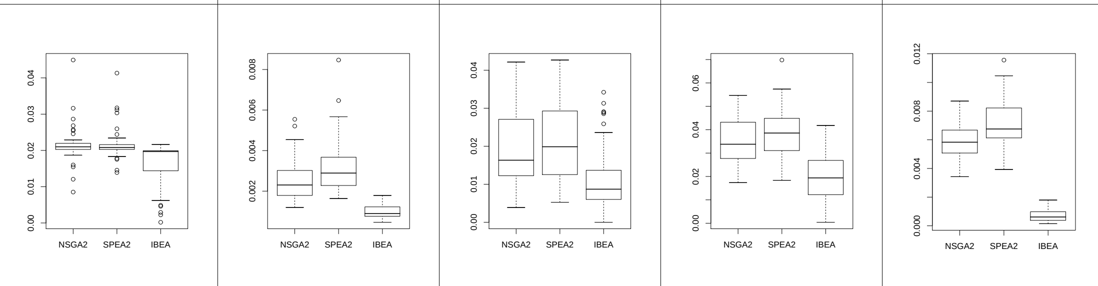
Figure 19: Experiment four, two dimensions, performance plots.



a) WFG9 baseline epsilon	b) Test 1 epsilon	c) Test 2 epsilon	d) WFG4 baseline epsilon	e) Test 3 epsilon

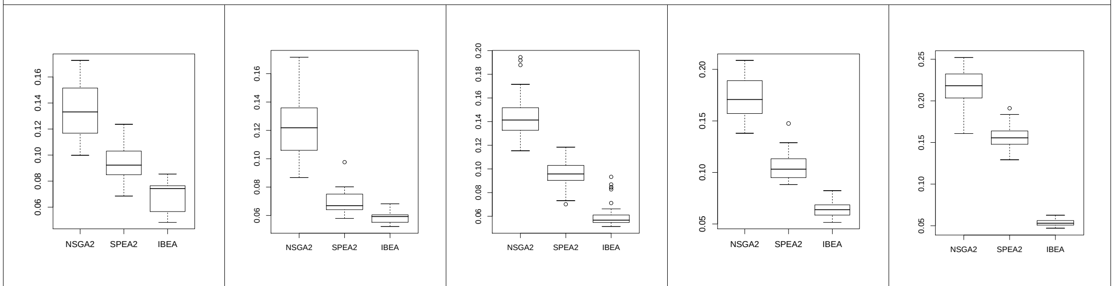


f) WFG9 baseline hypervolume	g) Test 1 hypervolume	h) Test 2 hypervolume	i) WFG4 baseline hypervolume	j) Test 3 hypervolume

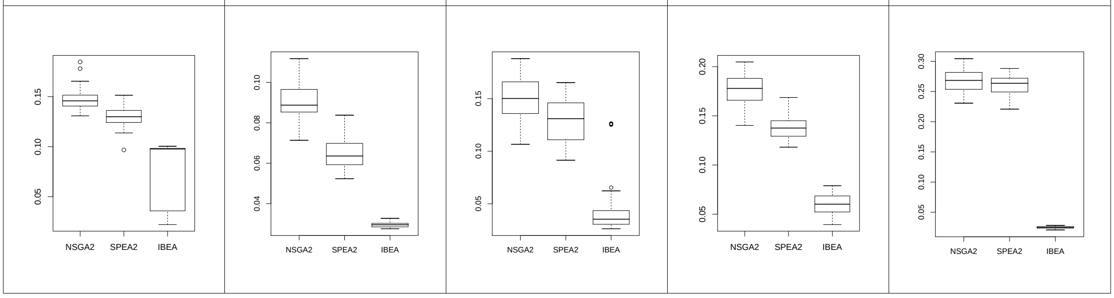


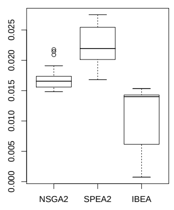
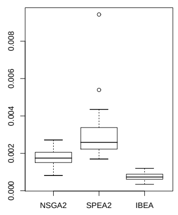
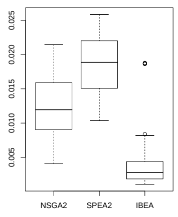
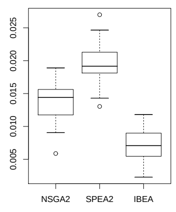
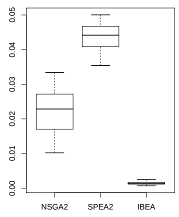
k) WFG9 baseline R	l) Test 1 R	m) Test 2 R	n) WFG 4 baseline R	o) Test 3 R

Figure 20: Experiment four, three dimensions, performance plots.



a) WFG9 baseline epsilon	b) Test 1 epsilon	c) Test 2 epsilon	d) WFG4 baseline epsilon	e) Test 3 epsilon



f) WFG9 baseline hypervolume	g) Test 1 hypervolume	h) Test 2 hypervolume	i) WFG4 baseline hypervolume	j) Test 3 hypervolume
				
k) WFG9 baseline R	l) Test 1 R	m) Test 2 R	n) WFG 4 baseline R	o) Test 3 R

2) Results: Firstly for two dimensions, Figure 19 shows the performance of Test 1 and Test 2 with altered transformations in WFG9. Compared to the baseline WFG9, we can see that in Test 1 removing transformation 2 (consisting of a multimodality transformation as well as a deceptiveness transformation) significantly increases the performance lead of IBEA. Whilst in Test 2 just removing the deceptiveness transformation in transformation 2 results in less spread in performance indicators but does not increase performance lead of IBEA as significantly, compared to Test 1. Test 3 again shows this where upon removal of the multimodality transformation the performance lead of IBEA increases significantly. These results suggest that introducing the multimodality transformation poses difficulties for IBEA.

In three objectives the results mimic those in two dimensions, Figure 20 shows that in Test 1 and Test 3 removing the multimodality transformations dramatically increases the performance lead of IBEA compared to the baseline performance, whereas removing only the deceptiveness transformation with Test 2 increases the spread of performance, but does not altar the lead of IBEA over NSGA-II and SPEA, relative to the baseline.

One way to explain this is that IBEA solely relies on hypervolume for diversity preservation, however when there is a multimodal landscape solutions tend to cluster around the top of “hills”, it is difficult for these solutions to move to other “hills” as in the process of doing so hypervolume will decrease or remain unchanged, which gives no incentive. Algorithms like NSGA-II which rely crowding distance for diversity preservation may make better decisions as it attempts to maximise distance between solutions which gives incentive to find other “hills”.

This experiment shows clearly that multimodality, a problem features which increases problem difficulty, poses difficulties for the IBEA algorithm.

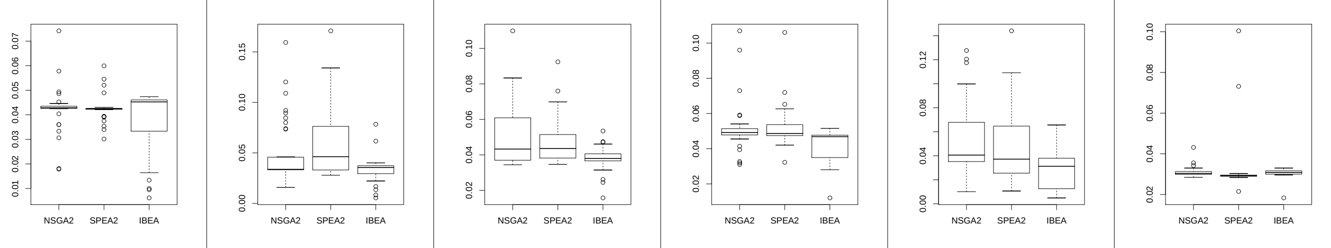
4.5 Experiment 5, Varying Multimodality

1) Aim and Experimental Setup: This experiment aims to further investigate how varying degree multimodality impacts upon the performance of IBEA. The implementation of the multimodality transformation in the WFG toolkit is in the form of multi(A,B,C) where A defines the number of minima, B controls the magnitude of the “hill sizes”, and C is in the range of [0,1] determining the location of the global optima. We want to test a wide range of values to see if extreme parameter values change the behavior of IBEA significantly. In [12] it states larger values of A and smaller values of B create more difficult problems, and this is reflected in Test 2. Tests 1-5 in Table 6 are completed in both two dimensions and three dimensions.

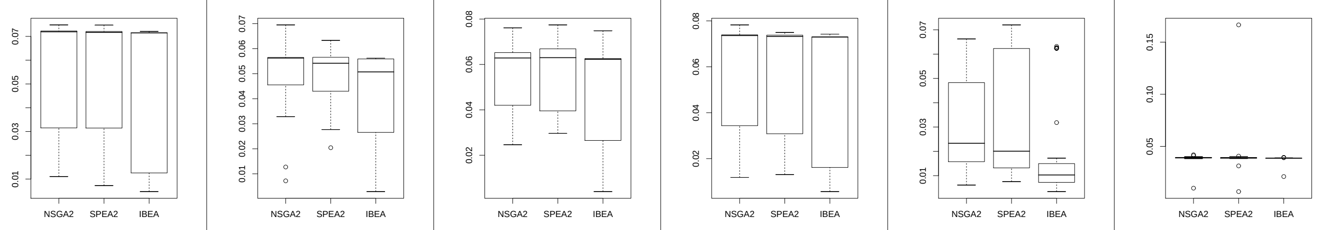
Table 6: Experiment five tests.

Test 1	WFG9, change transformation 2, multi(5, 5, 0.35)
Test 2	WFG9, change transformation 2, multi(95, 30, 0.35)
Test 3	WFG9, change transformation 2, multi(190, 190, 0.35)
Test 4	WFG9, change transformation 2, multi(30, 95, 0.05)
Test 5	WFG9, change transformation 2, multi(30, 95, 0.95)

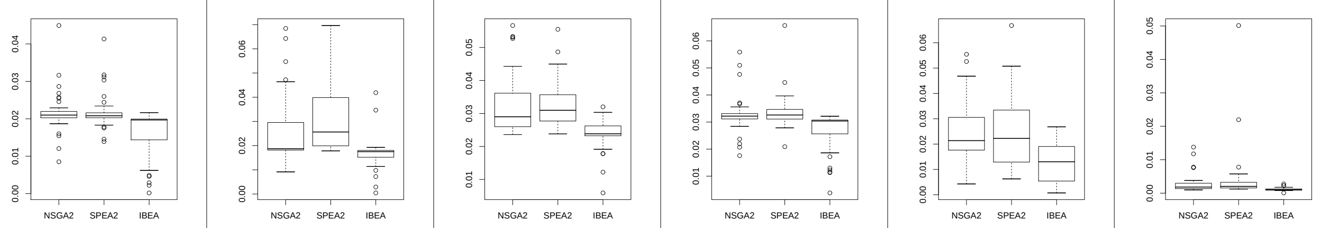
Figure 21: Experiment five, two dimensions, performance plots.



a) WFG9 baseline epsilon	b) Test 1 epsilon	c) Test 2 epsilon	d) Test 3 epsilon	e) Test 4 epsilon	f) Test 5 epsilon

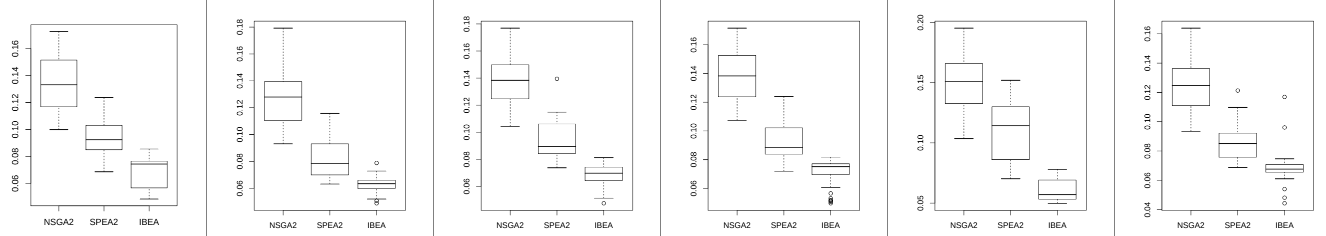


g) WFG9 baseline hypervolume	h) Test 1 hypervolume	i) Test 2 hypervolume	j) Test 3 hypervolume	k) Test 4 hypervolume	l) Test 5 hypervolume

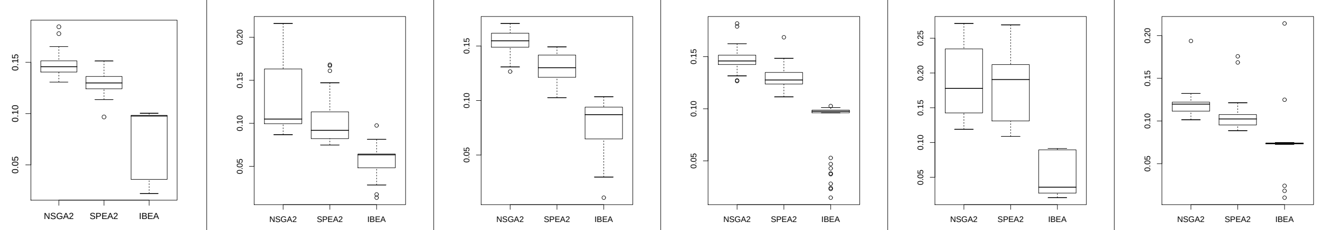


m) WFG9 baseline R	n) Test 1 R	o) Test 2 R	p) Test 3 R	q) Test 4 R	r) Test 5 R

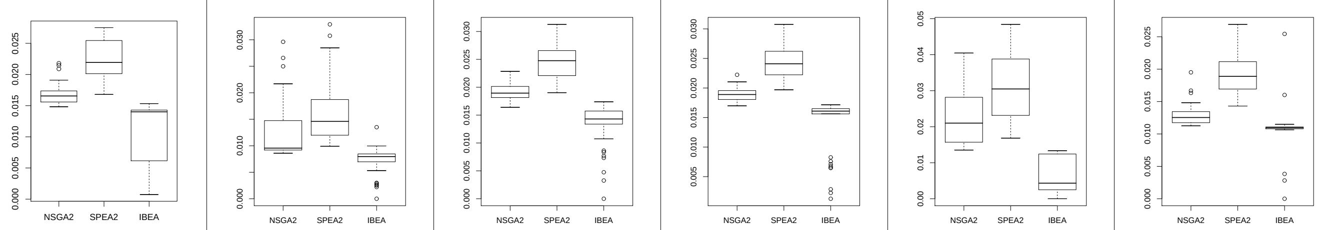
Figure 22: Experiment five, three dimensions, performance plots.



a) WFG9 baseline epsilon	b) Test 1 epsilon	c) Test 2 epsilon	d) Test 3 epsilon	e) Test 4 epsilon	f) Test 5 epsilon



g) WFG9 baseline hypervolume	h) Test 1 hypervolume	i) Test 2 hypervolume	j) Test 3 hypervolume	k) Test 4 hypervolume	l) Test 5 hypervolume



m) WFG9 baseline R	n) Test 1 R	o) Test 2 R	p) Test 3 R	q) Test 4 R	r) Test 5 R

2) Results: By examining the box-whisker-plots in Figure 21 it can be seen that in 2d Test 3 and Test 5 the performance lead of IBEA is equal or less to the baseline performance for WFG9, with Test 5 presenting more problems for IBEA with performance indicators for hyprevolume showing the three MOEAs performing roughly on par, as well in Test 5 the epsilon indicator shows SPEA2 taking the lead over IBEA, something not evident in the other tests.

In three objectives the results are very close, but again Figure 22 shows Test 3 and Test 5 giving IBEA the most trouble, in Test 5 SPEA2 no longer beats IBEA in the epsilon indicator. In Test 3 the box-whisker-plot shows a few low outliers for IBEA, which indicates that IBEA is able to find a few very good solution fronts. We find that overall in both 2 dimensions and 3 dimensions using a high value for the parameter C in the multimodality transformation makes the test problem more difficult for the IBEA algorithm.

One way to explain this phenomenon with IBEA is that by moving the global optima towards the extrema, less solutions are likely to “move” there and fully fill up the entire “volume” of the global optima, where as with the optimal near the origin, there are more solutions around which can find the global optima. As discussed in 4.5 2), it could be harder in IBEA for solutions to “jump” between hills due to its diversity preservation based purely on hypervolume.

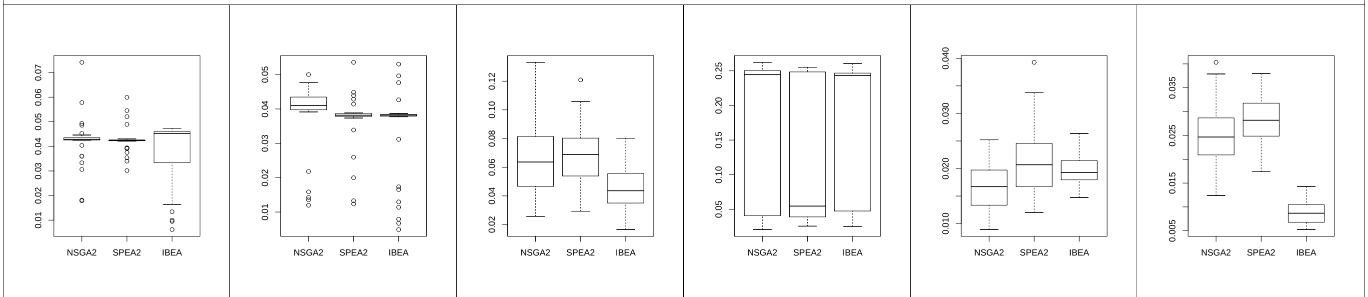
4.6 Experiment 6, Investigate shape

1) Aim and Experimental Setup: We recall that in the baseline experiments, WFG2, WFG3, WFG4 and WFG9 are test problems in which performance between the three MOEAs are more even and are more difficult for IBEA. Apart from multimodality there is no common transformation amongst these test problems, instead in this experiment we examine how the shape functions from these test problems will impact upon the performance of IBEA. This is easily accomplished in the WFG toolkit by plugging shape functions from WFG2, WFG3, and WFG4 into different test problems. The WFG2 shape function is a convex and discontinuous, WFG3 uses a

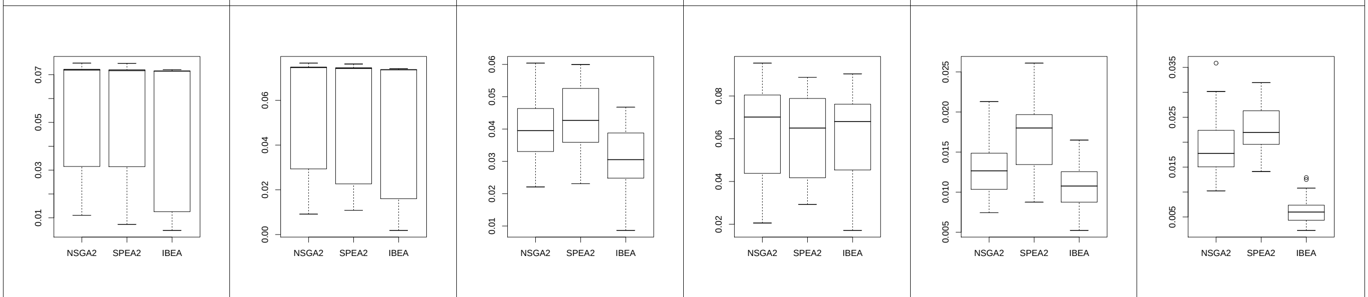
linear shape function, and the WFG4 shape function is a concave one. Experiments of Test 1-3 in Table 7 are carried out in both two and three dimensions, also note the shape function from WFG5 – WFG9 uses the concave shape function from WFG4 and will not be re-tested.

Table 7. Experiment six tests.	
Test 1	WFG9, use shape function from WFG3
Test 2	WFG4, use shape function from WFG2
Test 3	WFG2, use shape function from WFG4

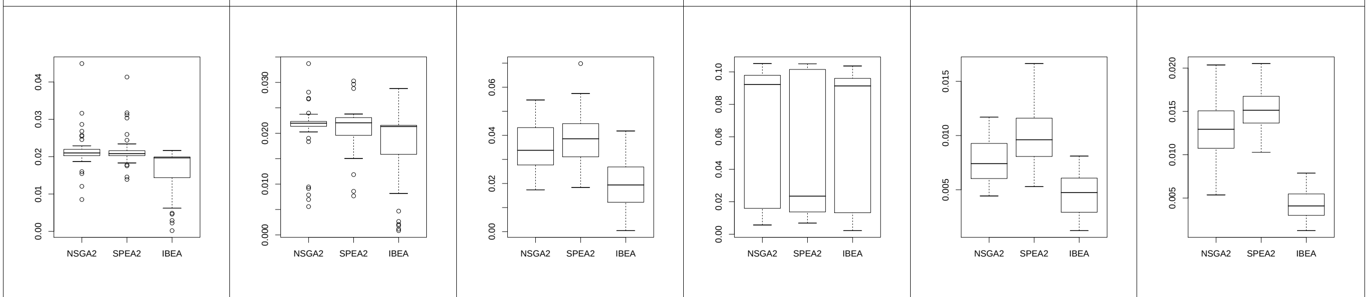
Figure 23: Experiment six, two dimensions, performance plots.



a) WFG9 baseline epsilon | b) Test 1 epsilon | c) WFG4 baseline epsilon | d) Test 2 epsilon | e) WFG2 baseline epsilon | f) Test 3 epsilon

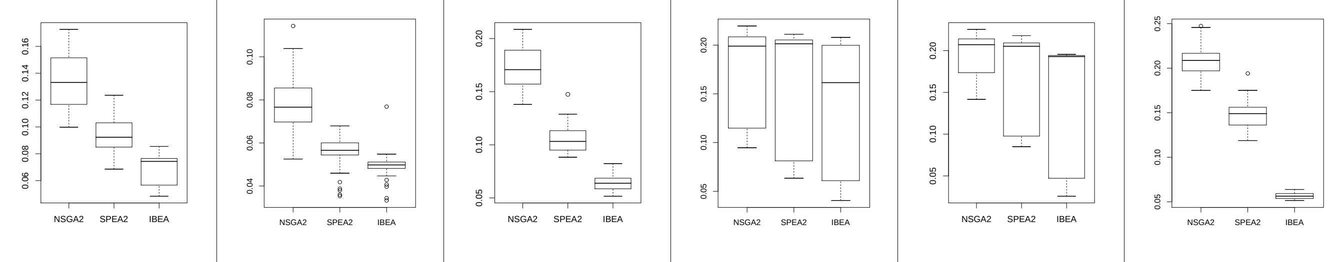


g) WFG9 baseline hypervolume | h) Test 1 hypervolume | i) WFG4 hypervolume | j) Test 2 hypervolume | k) WFG2 hypervolume | l) Test 3 hypervolume

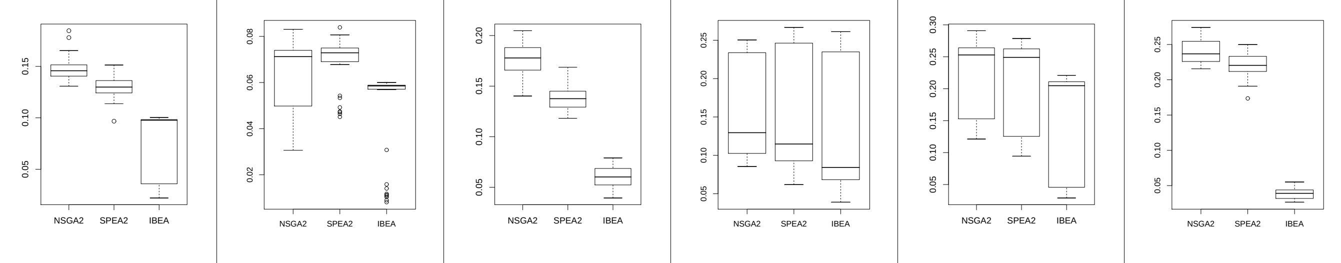


m) WFG9 baseline R | n) Test 1 R | o) WFG4 R | p) Test 2 R | q) WFG2 R | r) Test 3 R

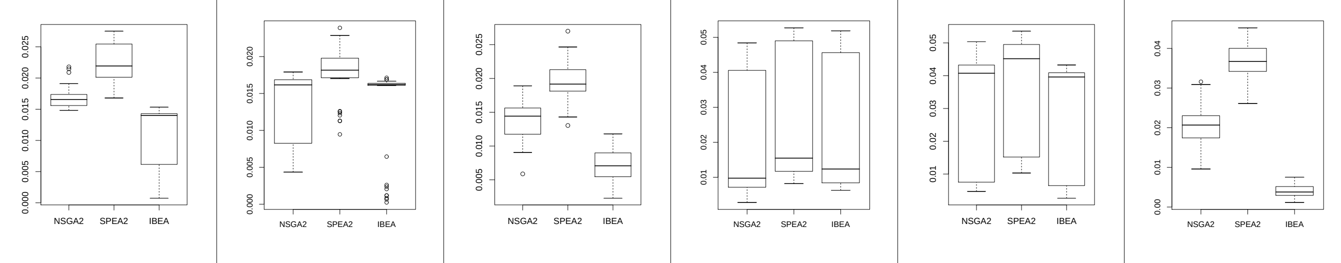
Figure 24: Experiment six, three dimensions, performance plots.



a) WFG9 baseline epsilon	b) Test 1 epsilon	c) WFG4 baseline epsilon	d) Test 2 epsilon	e) WFG2 baseline epsilon	f) Test 3 epsilon



g) WFG9 baseline hypervolume	h) Test 1 hypervolume	i) WFG4 hypervolume	j) Test 2 hypervolume	k) WFG2 hypervolume	l) Test 3 hypervolume



m) WFG9 baseline R	n) Test 1 R	o) WFG4 R	p) Test 2 R	q) WFG2 R	r) Test 3 R

2) Results: Test 1 uses the WFG3 shape function in the WFG9 test problem, we see that in Figure 23-24 for both the 2 dimensional and 3 dimensional case, the box-and-whisker-plots display a slight decrease in the performance lead of IBEA compared to baseline WFG9 performance results. This is more evident in the 3 dimensional case and suggests that the WFG3 shape function is tougher on IBEA than the WFG4 shape function used by WFG9. One way to explain this is that hypervolume is not well adjusted to the degenerate front from WFG3, which is in a lower dimension then that of the problem. Another explanation is that the convex part of the front favours solutions towards the origin as it is easier to obtain increases in hypervolume there.

In Test 2 the results show that the convex and discontinuous WFG2 shape function also presents more difficulty to IBEA compared to the WFG4 shape function, this is particularly evident in two dimensions where the performance of NSGA-II is performing slightly ahead of IBEA on all three performance indicators. However, the lead of NSGA-II in the box-whisker-plot for hypervolume is deceiving, the Mann-Whitely U statistic gives a p-value of 0.67 for NSGA-II beating IBEA and conversely a p-value of 0.33 for IBEA beating NSGA-II, so we cannot reject the either null hypothesis, and the result is inconclusive. In three dimensions the performance lead of IBEA also is drastically reduced by switching to the WFG2 shape function. The discontinuous convex shape function in WFG2 can pose problems for optimisation with selection using the hypervolume metric, where there is less incentive to find solutions near extrema, and a discontinuous shape can increase difficulty as when regions of easier hypervolume gain has been populated, it is then harder for solutions to “jump” across discontinuities to reach extrema.

In Test 3 when the concave WFG4 shape function is used in the WFG2 problem, the performance of IBEA increases drastically compared to baseline results of WFG2 in both two and three dimensions, this shows that the WFG4 shape function is easier on IBEA and is consistent with Test 2 results. In this experiment we have shown that both WFG2 and WFG3 shape functions present more difficulties to IBEA than the WFG4 shape function. The concave shape of WFG4 may favour hypervolume as initially there is incentive to find extrema, then solutions can approach the origin from “two directions”, as opposed to extrema where there is only “one direction”.

4.7 Experiment 7, beat IBEA

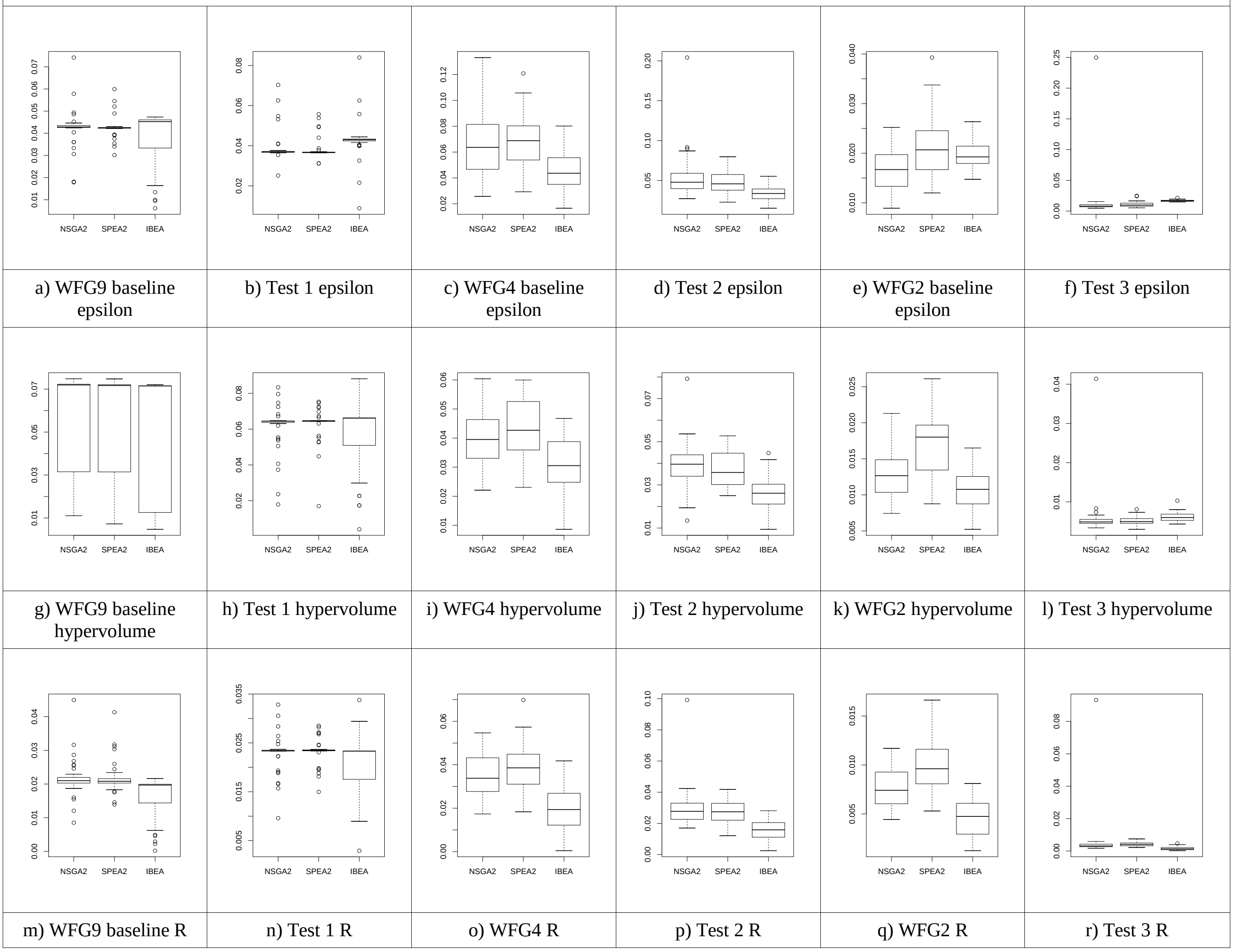
1) Aim and Experimental Setup: In this experiment we hope to use knowledge of IBEA's performance with various features built up from past experiments, to construct new test problems in which IBEA loses, and through doing so demonstrate the “no free lunch theorem”. In Test 1 we change the WFG4 shape function in WFG9 to the WFG2 shape function which is more difficult for IBEA (Experiment 6), we also adjust multimodality transformation parameters to ones harder on IBEA (Experiment 5). In Test 2 we change the shape function of WFG4 to the WFG3 shape function

which is a more difficult shape function on IBEA (Experiment 6). Finally in Test 3 we add transformation two from WFG9 to WFG2, which contains a multimodality transformation as well as a deceptiveness transformation, this again should make it harder for IBEA (Experiment 4). Test 1-3 in Table 8 are run in both two and three dimensions.

Table 8: Experiment seven tests.

Test 1	WFG9, use shape function from WFG2, change transformation 2 to multi(30,95,0.8)
Test 2	WFG4, use shape function from WFG3
Test 3	WFG2, add transformation 2 from WFG9 (multimodality and deceptiveness)

Figure 25: Experiment seven, two dimensions, performance plots.

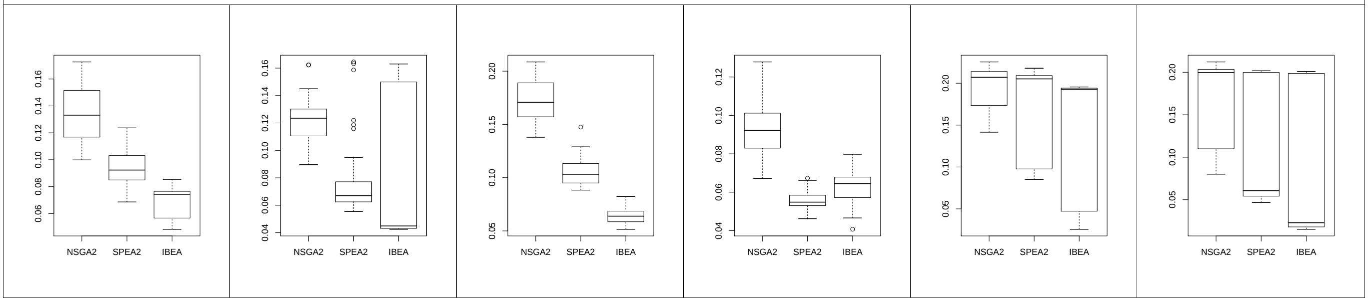


a) WFG9 baseline epsilon; b) Test 1 epsilon; c) WFG4 baseline epsilon; d) Test 2 epsilon; e) WFG2 baseline epsilon; f) Test 3 epsilon

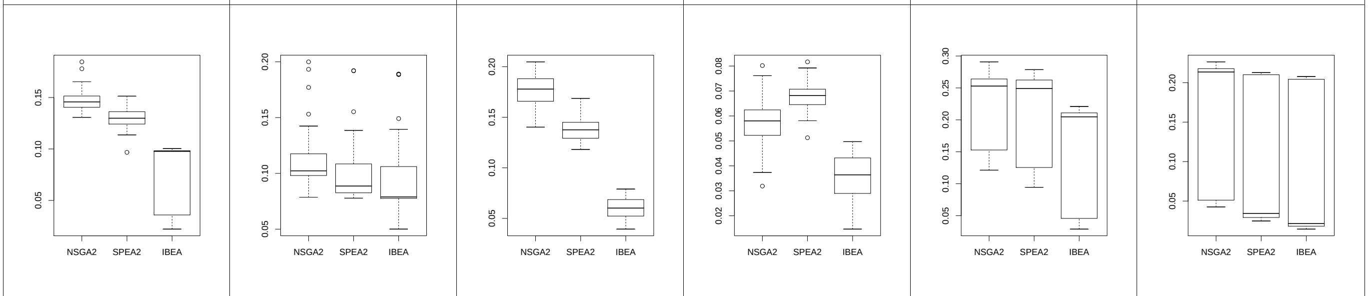
g) WFG9 baseline hypervolume; h) Test 1 hypervolume; i) WFG4 hypervolume; j) Test 2 hypervolume; k) WFG2 hypervolume; l) Test 3 hypervolume

m) WFG9 baseline R; n) Test 1 R; o) WFG4 R; p) Test 2 R; q) WFG2 R; r) Test 3 R

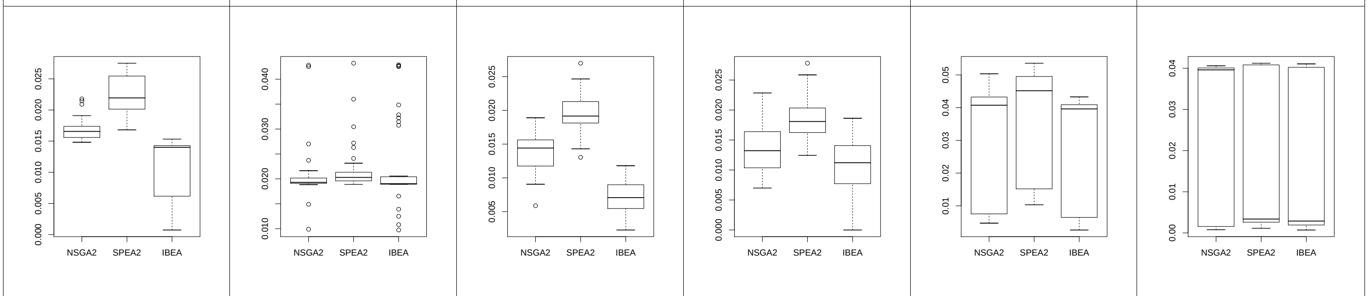
Figure 26: Experiment seven, three dimensions, performance plots.



a) WFG9 baseline epsilon	b) Test 1 epsilon	c) WFG4 baseline epsilon	d) Test 2 epsilon	e) WFG2 baseline epsilon	f) Test 3 epsilon



g) WFG9 baseline hypervolume	h) Test 1 hypervolume	i) WFG4 hypervolume	j) Test 2 hypervolume	k) WFG2 hypervolume	l) Test 3 hypervolume



m) WFG9 baseline R	n) Test 1 R	o) WFG4 R	p) Test 2 R	q) WFG2 R	r) Test 3 R

2) Results: Firstly as shown in Figure 25, in two dimensions we obtain positive results from our newly constructed test problems, in Test 1 the hypervolume performance indicator show NSGA-II and SPEA2 performing very close to IBEA, both with lower medians. The Mann-Whitney U test shows that NSGA-II is better than IBEA with a p-value of 0.14, which is insignificant and we cannot reject the null hypothesis that they beat each other. For the epsilon indicator SPEA2 is able to beat both IBEA and NSGA-II in this problem, which is a positive result. Whilst the R performance indicator shows that IBEA is slightly ahead. Overall it is clear that Test 1 in 2 dimensions is a problem in which NSGA-II and SPEA2 are able to perform at least as well as IBEA. In Test 2 IBEA is again dominant performing slightly ahead of both NSGA-II and SPEA2 on all indicators. Test 3 in two dimensions is a very good result as both NSGA-II and SPEA2 are able to comfortably beat IBEA in both the

hypervolume and epsilon metrics, this may be explained by the better performance of diversity preservation of both NSGA-II and SPEA2 in the presence of multimodality.

Figure 26 shows results for three dimensions, and the newly constructed problems have a tougher time. Although the performance of NSGA-II and SPEA2 come very close to IBEA, IBEA is able to comfortably win on all tests, for the exception of Test 2, where surprisingly SPEA2 wins on the epsilon performance metric.

Overall the results from this experiment are very exciting, as we have constructed new test problems with features which we observe make it hard for the IBEA algorithm, and been able to demonstrate the “no free lunch theorem” by beating IBEA which was the superior algorithm in all the baseline experiments. Test 3 under two dimensions produced particularly surprising results with NSGA-II and SPEA2 beating IBEA on hypervolume. The Hypervolume metric is used internally by IBEA as part of the selection mechanism, which is not the case with NSGA-II and SPEA. We have found a case where IBEA seemingly loses in its area of comparative advantage.

5 Conclusion

In this dissertation we have tested three popular MOEAs IBEA, NSGA-II and SPEA2 with test problems with a wide variety of features and in many objectives. Fortunately using WFG toolkit allowed us to easily and systematically adjust test problem features by varying transitions, this controls the complexity and nature of the problem.

By implementing the WFG toolkit as a selector in PISA it is easy to conduct performance testing on a wide range of MOEAs and more specifically allow experimentation on how different test problem features affect MOEA performance.

In experiments 1-3 we find that IBEA is able to consistently outperform NSGA-II and SPEA2 on WFG2-WFG9 problems in 2, 3 and 5 objectives, for the exception of the r metric in five dimensions. We also find that although NSGA-II is able to beat SPEA2 in two dimensions, SPEA2 wins in higher dimensions where NSGA-II is less able to preserve diversity, this is due to crowding distance performing sub par on higher dimensions, this is consistent with past findings [12].

We attempt to find test problem characteristics which make it more difficult for IBEA, and in experiment 4 we find that multimodality increases difficulty for IBEA both in two and three objectives. Further in Experiment 5 we find that both increasing parameters *a* and *c* in the multimodality transition significantly increases difficulty for IBEA in two and three objectives, where parameter a controls the amount of multimodality 'hills' and parameter c controls global optima position.

Going down a different path we investigate in experiment 6 which Pareto optimal geometries increase difficulty for IBEA. By switching shape functions between different test problems we observe that both the disconnected and convex shape function from WFG2 and the linear and degenerate shape function of WFG3 are more difficult for IBEA in both two and three dimensions, compared to the concave shape function of WFG4.

Finally in experiment 7 we attempt to beat IBEA by combining the results from experiment 4-6. By constructing a new test problems with multimodality transformation and using shape functions from WFG2 and WFG3, we find that for two dimensions taking the test problem WFG2 and adding multimodality and deceptiveness transformations from WFG9, we construct a new test problem where NSGA-II and SPEA are able to outperform IBEA on both epsilon and hypervolume metrics. Further for three dimensions using the WFG4 test problem and replacing its shape function with the shape function of WFG3, SPEA2 is able to beat IBEA in the epsilon metric. In doing so we have successfully demonstrated a case of the "no free lunch" theorem, where the elevated performance of IBEA on the example WFG test problems are paid for in performance losses with test problems constructed in experiment 7.

Through our experiments we see that IBEA is able to perform very well against NSGA-II and SPEA2 on the default WFG2-WFG9 test problems, however test problem features such as multimodality, convex and disconnected geometries, and linear and degenerate geometries make the test problem harder for IBEA. This motivates further research on other features which increase problem difficulty for IBEA, the construction of test problems where NSGA-II and SPEA2 can beat IBEA on the hypervolume metric in higher dimensions, and further analysis of the R metric in higher dimensions. The WFG toolkit can be used to both study MOEAs and create better test problem suites in the future.

Bibliography

[1] T. Back, K. Hammel, and H. shwefel, 1997 ‘Evolutionary Computation: Comments on the History and Current State’, IEEE Transactions on Evolutionary Computation, Vol 1, No 1, pp. 3-17.

[2] K. Shaw, A. L. Nortcliffe, M. Thompson, J. Love, P. J. Fleming, and C. M. Fonseca, “Assessing the performance of multiobjective genetic algorithms for optimization of a batch process scheduling problem,” in Proc. Congr. Evol. Comput., vol. 1, P. J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, and A. Zalzala, Eds., July 1999, pp. 37–45.

[3] P. Engrand, “A multi-objective optimization approach based on simulated annealing and its application to nuclear fuel management,” in Proc. 5th Int. Conf. Nuclear Eng., 1997, pp. 416–423.

[4] S. Watanabe, T. Hiroyasu, and M. Miki, “Parallel evolutionary multicriterion optimization for mobile telecommunication networks optimization,” in EUROGEN 2001—Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, Eds. Barcelona, Spain: EdsInternational Center for Numerical Methods in Engineering (CIMNE), Sept. 2001, pp. 167–172.

[5] S. Huband, P. Hingston, L. While, and L. Barone, “An evolution strategy with probabilistic mutation for multiobjective optimization,” in Proc. Congr. Evol. Comput., vol. 4, R. R. Ruhul Sarker, H. Abbass, K. C. Tan, B. McKay, D. Essam, and T. Gedeon, Eds., Dec. 2003, pp. 2284–2291.

[6] E. Zitzler and S. Künzli. “Indicator-Based Selection in Multiobjective Search.” In *Conference on Parallel Problem Solving from Nature (PPSN VIII)*, pages 832--842, 2004. Springer.

[7] E. Zitzler, M. Laumanns, and L. Thiele. “SPEA2: Improving the Strength Pareto Evolutionary Algorithm.” Technical Report TIK-Report 103, Swiss Federal Institute of Technology Zurich (ETH), May 2001.

[8] E. Zitzler and L. Thiele. “An evolutionary algorithm for multiobjective optimization: The strength pareto approach.” Technical Report 43, Institute TIK, ETH Zurich, Switzerland, May 1998.

[9] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II,” In Marc Schoenauer et al., editor, Proceedings of the Parallel Problem Solving from Nature VI Conference, pp. 849-858, Paris, France, 2000.

[10] S. Huband, P. Hingston, L. Barone, and L. While, “A review of multiobjective test problems and a scalable test problem toolkit,” IEEE Transactions on Evolutionary Computation, vol. 10, no. 5, pp. 477–506, October 2006.

[11] D. H. Wolpert and W. G. MacReady. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, April 1996.

[12] L. Bradstreet, L. Barone, L. While, S. Huband, and P. Hingston. "Use of the WFG Toolkit and PISA for Comparison of Multi-objective Evolutionary Algorithms." *2007 IEEE Symposium on Computational Intelligence in Multi-criteria Decision-making. IEEE MCDM 2007*, pages 382-389. IEEE, April 2007.

[13] C. A. Coello, 2006 "Evolutionary Multi-Objective Optimisation: A historical View of the Field", IEEE Computational Intelligence Magazine, Vol 1, No 1, pp. 28-36.

[14] C. M. Fonseca, and P. J. Flemming, 1995 "An Overview of Evolutionary Algorithms in Multiobjective Optimisation", Evolutionary Computation, Vol 3, No 1, pp. 1-16.

[15] C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont: "Evolutionary Algorithms for Solving Multi-Objective Problems." Kluwer Academic Publishers, New York (2002)

[16] Laumanns, M., Fonseca, C., and Grunert da Fonseca, V (2003): Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation, 7(2), 117-132.

[17] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: Methods and applications," Ph.D. dissertation, Shaker Verlag, Aachen, Germany, 1999.

[18] M. P. Hansen and A. Jaskiewicz. "Evaluating the quality of approximations to the non-dominated set." Technical Report IMM-REP1998 -7, Technical University of Denmark, March 1998.

[19] R. L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume." IEEE Trans. Evolutionary Computation, vol. 10, no. 1, pp. 29–38, 2006.

[20] L. Bradstreet, L. While, and L. Barone. "A Fast Incremental Hypervolume Algorithm." Accepted for *IEEE Transactions on Evolutionary Computation*.

[21]N. Srinivas and K. Deb, "Multiobjective function optimization using non-dominated sorting genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Fall 1995.

[22] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp.173–195, Summer 2000.

[23] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler. *PISA -- A Platform and Programming Language Independent Interface for Search Algorithms*. In Proc. of EMO'03, pages 494--508, 2003.

[24] K. Deb, “Multi-objective genetic algorithms: Problem difficulties and construction of test problems,” Evol. Comput., vol. 7, no. 3, pp. 205–230, Fall 1999.

[25] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable Test Problems for Evolutionary Multi-Objective Optimization.” Kanpur, India: Kanpur Genetic Algorithms Lab. (KanGAL), Indian Inst. Technol., 2001. KanGAL Report 2 001 001.

[26] S. Huband, L. Barone, L. While, and P. Hingston, “A scalable multiobjective test problem toolkit,” in Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, Mar. 2005, vol. 3410, Proc. Evolutionary Multi-Criterion Optimization: 3rd Int. Conf., pp. 280–294.

[27] S. Huband, P. Hingston, L. While, and L. Barone, “An evolution strategy with probabilistic mutation for multi-objective optimization,” in Proc. Congr. Evol. Comput., vol. 4, 2003, pp. 2284–2291.

[28] R. Purshouse, “On the evolutionary optimization of many objectives,” Ph.D. dissertation, Univ. Sheffield, Sheffield, U.K., 2003.

[29] E.. Zitzler, L. and Thiele, (1998): Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. Parallel Problem Solving from Nature (PPSN-V)

Appendix A. Test problem formulations

A1 Deb's Toolkit [10]

FUNCTIONS PROVIDED BY DEB FOR f_1, g, AND h. IF A FUNCTION IS COMPATIBLE WITH f_1 OR g, THEN $\mathbf{p}$ CAN BE SUBSTITUTED WITH $\mathbf{y}$ OR $\mathbf{z}$, RESPECTIVELY. FOR BREVITY, F1 AND G1 HAVE BEEN OMITTED FROM THE TABLE, AS THEY ARE SUBSUMED BY F2 AND G2, RESPECTIVELY

Name	For	Function	Parameter Domains
F2	f_1	$\delta + \sum_{i=1}^{\lvert\mathbf{p}\rvert} c_i p_i$, where $\delta, c_1, \ldots, c_{\lvert\mathbf{p}\rvert} > 0$.	Arbitrary
F3	f_1	$1 - \exp(-4r(\mathbf{p}))\sin^4(5\pi r(\mathbf{p}))$, where $r(\mathbf{p}) = \sqrt{\sum_{i=1}^{\lvert\mathbf{p}\rvert} p_i^2}$.	$[0, 1]$
F4	f_1, g	$2.0 - \exp\left(-\left(\frac{p_1 - 0.2}{0.004}\right)^2\right) - 0.8\exp\left(-\left(\frac{p_1 - 0.6}{0.4}\right)^2\right)$	$[0, 1]$
G2	f_1, g	$g_{\min} + (g_{\max} - g_{\min})\left(\frac{\sum_{i=1}^{\lvert\mathbf{p}\rvert} p_i - p_{i,\min}}{\sum_{i=1}^{\lvert\mathbf{p}\rvert} p_{i,\max} - p_{i,\min}}\right)^{\gamma}$, where $\gamma \neq 0$.	Arbitrary
G3.i	f_1, g	Rastrigin: $1 + 10\lvert\mathbf{p}\rvert + \sum_{i=1}^{\lvert\mathbf{p}\rvert}(p_i^2 - 10\cos(2\pi p_i))$	$[-30, 30]$
G3.ii	f_1, g	Schwefel: $1 + (6.5\pi)^2\lvert\mathbf{p}\rvert - \sum_{i=1}^{\lvert\mathbf{p}\rvert} p_i \sin(\sqrt{\lvert p_i\rvert})$	$[-512, 511]$
G3.iii	f_1, g	Griewank: $2 + \sum_{i=1}^{\lvert\mathbf{p}\rvert} p_i^2/4000 - \prod_{i=1}^{\lvert\mathbf{p}\rvert}\cos(p_i/\sqrt{i})$	$[-512, 511]$
H1.i	h	$1/f_1$, where $f_1 > 0$.	NA
H1.ii	h	$\begin{cases} 1 - \left(\frac{f_1}{\beta g}\right)^{\alpha}, & f_1 \leq \beta g \\ 0, & f_1 > \beta g \end{cases}$, where $f_1 \geq 0$, $g > 0$, and $\alpha \leq 1$.	NA
H2	h	As H1.ii, except $\alpha > 1$.	NA
H3	h	As H1.ii, except $\alpha = 0.25 + 3.75\frac{g - g_{\min}}{g_{\min}^* - g_{\min}}$, where $g_{\min}^*$ is the weakest locally optimal fitness value of g.	NA
H4	h	$1 - \left(\frac{f_1}{g}\right)^{\alpha} - \frac{f_1}{g}\sin(2\pi q f_1)$, where q is the number of disconnected regions in an unit interval of f_1.	NA

A2 ZDT test problems [10]

THE FIVE REAL-VALUED ZDT TWO OBJECTIVE PROBLEMS. SIMILAR TO DEB'S TOOLKIT, THE SECOND OBJECTIVE IS $f_2(\mathbf{y}, \mathbf{z}) = g(\mathbf{z})h(f_1(\mathbf{y}), g(\mathbf{z}))$, WHERE BOTH OBJECTIVES ARE TO BE MINIMIZED

Name	Problem	Parameter Domains
ZDT1	$f_1 = y_1$ $g = 1 + 9\sum_{i=1}^{k} z_i/k$ $h = 1 - \sqrt{f_1/g}$	$[0, 1]$
ZDT2	as ZDT1, except $h = 1 - (f_1/g)^2$.	$[0, 1]$
ZDT3	as ZDT1, except $h = 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1)$.	$[0, 1]$
ZDT4	as ZDT1, except $g = 1 + 10k + \sum_{i=1}^{k}(z_i^2 - 10\cos(4\pi z_i))$	$y_1 \in [0, 1]$ $z_1, \ldots, z_k \in [-5, 5]$
ZDT6	$f_1 = 1 - \exp(-4y_1)\sin^6(6\pi y_1)$ $g = 1 + 9(\sum_{i=1}^{k} z_i/k)^{0.25}$ $h = 1 - (f_1/g)^2$	$[0, 1]$

A3 DTLZ test problems [10]

SEVEN OF THE NINE DTLZ MANY OBJECTIVE PROBLEMS. ALL OBJECTIVES ARE TO BE MINIMIZED

Name	Problem	Parameter Domains
DTLZ1	$f_1 = (1+g)0.5\prod_{i=1}^{M-1} y_i$ $f_{m=2:M-1} = (1+g)0.5\left(\prod_{i=1}^{M-m} y_i\right)(1-y_{M-m+1})$ $f_M = (1+g)0.5(1-y_1)$ $g = 100\left[k+\sum_{i=1}^{k}\left((z_i-0.5)^2-\cos(20\pi(z_i-0.5))\right)\right]$	$[0,1]$
DTLZ2	$f_1 = (1+g)\prod_{i=1}^{M-1}\cos(y_i\pi/2)$ $f_{m=2:M-1} = (1+g)\left(\prod_{i=1}^{M-m}\cos(y_i\pi/2)\right)\sin(y_{M-m+1}\pi/2)$ $f_M = (1+g)\sin(y_1\pi/2)$ $g = \sum_{i=1}^{k}(z_i-0.5)^2$	$[0,1]$
DTLZ3	As DTLZ2, except the equation for g is replaced by the one from DTLZ1.	$[0,1]$
DTLZ4	As DTLZ2, except all $y_i \in y$ are replaced by y_i^{α}, where $\alpha > 0$.	$[0,1]$
DTLZ5	As DTLZ2, except all $y_2,\ldots,y_{M-1} \in \mathbf{y}$ are replaced by $\frac{1+2gy_i}{2(1+g)}$.	$[0,1]$
DTLZ6	As DTLZ5, except the equation for g is replaced by $g=\sum_{i=1}^{k} z_i^{0.1}$.	$[0,1]$
DTLZ7	$f_{m=1:M-1} = y_m$ $f_M = (1+g)\left(M-\sum_{i=1}^{M-1}\left[\frac{f_i}{1+g}(1+\sin(3\pi f_i))\right]\right)$ $g = 1+9\sum_{i=1}^{k} z_i/k$	$[0,1]$

A4 WFG shape functions [10]

SHAPE FUNCTIONS. IN ALL CASES, $x_1, \ldots, x_{M-1} \in [0,1]$. A, α, AND β ARE CONSTANTS

Linear

$$\begin{aligned}
\text{linear}_1(x_1,\ldots,x_{M-1}) &= \textstyle\prod_{i=1}^{M-1} x_i \\
\text{linear}_{m=2:M-1}(x_1,\ldots,x_{M-1}) &= \left(\textstyle\prod_{i=1}^{M-m} x_i\right)(1 - x_{M-m+1}) \\
\text{linear}_M(x_1,\ldots,x_{M-1}) &= 1 - x_1
\end{aligned}$$

When $h_{m=1:M} = \text{linear}_m$, the Pareto optimal front is a linear hyperplane, where $\sum_{m=1}^{M} h_m = 1$.

Convex

$$\begin{aligned}
\text{convex}_1(x_1,\ldots,x_{M-1}) &= \textstyle\prod_{i=1}^{M-1}(1-\cos(x_i\pi/2)) \\
\text{convex}_{m=2:M-1}(x_1,\ldots,x_{M-1}) &= \left(\textstyle\prod_{i=1}^{M-m}(1-\cos(x_i\pi/2))\right)(1-\sin(x_{M-m+1}\pi/2)) \\
\text{convex}_M(x_1,\ldots,x_{M-1}) &= 1 - \sin(x_1\pi/2)
\end{aligned}$$

When $h_{m=1:M} = \text{convex}_m$, the Pareto optimal front is purely convex.

Concave

$$\begin{aligned}
\text{concave}_1(x_1,\ldots,x_{M-1}) &= \textstyle\prod_{i=1}^{M-1}\sin(x_i\pi/2) \\
\text{concave}_{m=2:M-1}(x_1,\ldots,x_{M-1}) &= \left(\textstyle\prod_{i=1}^{M-m}\sin(x_i\pi/2)\right)\cos(x_{M-m+1}\pi/2) \\
\text{concave}_M(x_1,\ldots,x_{M-1}) &= \cos(x_1\pi/2)
\end{aligned}$$

When $h_{m=1:M} = \text{concave}_m$, the Pareto optimal front is purely concave, and a region of the hyper-sphere of radius one centred at the origin, where $\sum_{m=1}^{M} h_m^2 = 1$.

Mixed convex/concave ($\alpha > 0$, $A \in \{1, 2, \ldots\}$)

$$\text{mixed}_M(x_1,\ldots,x_{M-1}) = \left(1 - x_1 - \frac{\cos(2A\pi x_1 + \pi/2)}{2A\pi}\right)^{\alpha}$$

Causes the Pareto optimal front to contain both convex and concave segments, the number of which is controlled by A. The overall shape is controlled by α: when $\alpha > 1$ or when $\alpha < 1$, the overall shape is convex or concave respectively. When $\alpha = 1$, the overall shape is linear.

Disconnected ($\alpha, \beta > 0$, $A \in \{1, 2, \ldots\}$)

$$\text{disc}_M(x_1,\ldots,x_{M-1}) = 1 - (x_1)^{\alpha}\cos^2(A(x_1)^{\beta}\pi)$$

Causes the Pareto optimal front to have disconnected regions, the number of which is controlled by A. The overall shape is controlled by α (when $\alpha > 1$ or when $\alpha < 1$, the overall shape is concave or convex respectively, and when $\alpha = 1$, the overall shape is linear), and β influences the location of the disconnected regions (larger values push the location of disconnected regions towards larger values of x_1, and vice versa).

A5 WFG feature transformations & restrictions [10]

TRANSFORMATION FUNCTIONS. THE PRIMARY PARAMETERS y AND $y_1, \dots, y_{|\mathbf{y}|}$ ALWAYS HAVE DOMAIN [0,1]. A, B, C, α, AND β ARE CONSTANTS. FOR b_param, $\mathbf{y}'$ IS A VECTOR OF SECONDARY PARAMETERS (OF DOMAIN [0,1]), AND u IS A REDUCTION FUNCTION

Bias: Polynomial ($\alpha > 0$, $\alpha \neq 1$)

$$\text{b\_poly}(y, \alpha) = y^{\alpha}$$

When $\alpha > 1$ or when $\alpha < 1$, y is biased towards zero or towards one respectively.

Bias: Flat Region ($A, B, C \in [0,1]$, $B < C$, $B = 0 \Rightarrow A = 0 \wedge C \neq 1$, $C = 1 \Rightarrow A = 1 \wedge B \neq 0$)

$$\text{b\_flat}(y, A, B, C) = A + \min(0, \lfloor y - B \rfloor)\frac{A(B-y)}{B} - \min(0, \lfloor C - y \rfloor)\frac{(1-A)(y-C)}{1-C}$$

Values of y between B and C (the area of the flat region) are all mapped to the value A.

Bias: Parameter Dependent ($A \in (0,1)$, $0 < B < C$)

$$\text{b\_param}(y, \mathbf{y}', A, B, C) = y^{B+(C-B)v(u(\mathbf{y}'))}$$

$$v(u(\mathbf{y}')) = A - (1 - 2u(\mathbf{y}'))\left|\lfloor 0.5 - u(\mathbf{y}') \rfloor + A\right|$$

A, B, C, and the secondary parameter vector $\mathbf{y}'$ together determine the degree to which y is biased by being raised to an associated power: values of $u(\mathbf{y}') \in [0, 0.5]$ are mapped linearly onto $[B, B + (C - B)A]$, and values of $u(\mathbf{y}') \in [0.5, 1]$ are mapped linearly onto $[B + (C - B)A, C]$.

Shift: Linear ($A \in (0,1)$)

$$\text{s\_linear}(y, A) = \frac{|y - A|}{|\lfloor A - y \rfloor + A|}$$

A is the value for which y is mapped to zero.

Shift: Deceptive ($A \in (0,1)$, $0 < B \ll 1$, $0 < C \ll 1$, $A - B > 0$, $A + B < 1$)

$$\text{s\_decept}(y, A, B, C) = 1 + (|y - A| - B) \times \left(\frac{\lfloor y - A + B \rfloor (1 - C + \frac{A-B}{B})}{A - B} + \frac{\lfloor A + B - y \rfloor (1 - C + \frac{1-A-B}{B})}{1 - A - B} + \frac{1}{B}\right)$$

A is the value at which y is mapped to zero, and the global minimum of the transformation. B is the "aperture" size of the well/basin leading to the global minimum at A, and C is the value of the deceptive minima (there are always two deceptive minima).

Shift: Multi-modal ($A \in \{1, 2, \dots\}$, $B \geq 0$, $(4A + 2)\pi \geq 4B$, $C \in (0,1)$)

$$\text{s\_multi}(y, A, B, C) = \frac{1 + \cos\left[(4A+2)\pi\left(0.5 - \frac{|y-C|}{2(\lfloor C-y \rfloor + C)}\right)\right] + 4B\left(\frac{|y-C|}{2(\lfloor C-y \rfloor + C)}\right)^2}{B+2}$$

A controls the number of minima, B controls the magnitude of the "hill sizes" of the multi-modality, and C is the value for which y is mapped to zero. When $B = 0$, $2A + 1$ values of y (one at C) are mapped to zero, and when $B \neq 0$, there are $2A$ local minima, and one global minimum at C. Larger values of A and smaller values of B create more difficult problems.

Reduction: Weighted Sum ($|\mathbf{w}| = |\mathbf{y}|$, $w_1, \dots, w_{|\mathbf{y}|} > 0$)

$$\text{r\_sum}(\mathbf{y}, \mathbf{w}) = \left(\sum_{i=1}^{|\mathbf{y}|} w_i y_i\right) / \sum_{i=1}^{|\mathbf{y}|} w_i$$

By varying the constants of the weight vector $\mathbf{w}$, EAs can be forced to treat parameters differently.

Reduction: Non-separable ($A \in \{1, \dots, |\mathbf{y}|\}$, $|\mathbf{y}| \bmod A = 0$)

$$\text{r\_nonsep}(\mathbf{y}, A) = \frac{\sum_{j=1}^{|\mathbf{y}|}\left(y_j + \sum_{k=0}^{A-2}\left|y_j - y_{1+(j+k) \bmod |\mathbf{y}|}\right|\right)}{\frac{|\mathbf{y}|}{A}\lceil A/2 \rceil (1 + 2A - 2\lceil A/2 \rceil)}$$

A controls the degree of non-separability (noting that $\text{r\_nonsep}(\mathbf{y}, 1) = \text{r\_sum}(\mathbf{y}, \{1, \dots, 1\})$).

TRANSFORMATION FUNCTION RESTRICTIONS

Restriction	Comment
Constants	Must be fixed (not tied to the value of any parameters).
Primary parameters	For any given transition vector, all parameters of the originating transition vector must be employed exactly once as a primary parameter (counting parameters that appear independently as primary parameters), and in the same order in which they appear in the originating transition vector.
Secondary parameters	Care must be taken to avoid cyclical dependencies in b_param. Consider the following terminology: if a is a primary parameter of b_param, and b is one of the secondary parameters, then we say that a *depends* on b. If b likewise depends on c, then a also (indirectly) depends on c. When a depends on some parameter b, then there is an associated dependency between the corresponding working parameters. To prevent cyclical dependencies, no two working parameters should be dependent on one another. In addition, a parameter should not depend on itself.
Shifts	Parameters should not be subjected to more than one shift transformation.
Reductions	Reduction transformations should belong to transition vectors that are closer to the underlying parameter vector than any shift transformation.
b_flat	When $A = 0$, b_flat should only belong to transition vectors that are further away from the underlying parameter vector than any shift or reduction transformation.

Appendix B. Revised Proposal

Research Proposal

Title: "Investigate the performance of multi-objective algorithms on varying test problem features using the WFG toolkit."

Author: Liang Zhao

Supervisor: Luigi Barone

Background

Optimisation aims at locating a good solution given a set of constraints [1]. Many real world problems can be modelled and optimised. However, optimisation of complex systems by checking the performance of all parameters is often impossible as a consequence of non-separability, where parameters are dependent upon each other and adjusting one parameter affects others. Evolutionary algorithms are particularly well suited in solving complex optimisation problems [2], due to their adaptability and non-assumption of the underlying fitness landscape.

Evolutionary algorithms imitate the process of natural evolution in computers [2]. In the context of optimisation, this involves the following:

1. Generation an initial random population of solutions.

2. Applying the fitness function to each solution.

3. Selection of parents from the best solutions based on a fitness measure and discarding the rest.

4. Generation of new solutions from parents using techniques such as mutation and crossover.

Iterate through 2-4 until a solution reaches desired level of fitness.

Optimisation problems which contain more than one objective are known as multi-objective optimisation problem (MOP), where optimisation aims to find the Pareto optimal set [4]. Consider a solution vector; it is referred to as a Pareto optimal vector if there does not exist another solution vector which increases performance in one objective without decreasing performance in at least one other objective. A set of such solution vectors is called a Pareto optimal front, which represent efficient solutions with the best trade-offs between objectives. It is difficult to extend conventional optimisation techniques to the true multi-objective case, as they were not designed with multiple objectives in mind [3]. However evolutionary algorithms are positively well-suited to multi-objective optimisation [3], and are applicable in many industries such as management, medicine, chemistry, physics and scheduling [1].

MOEAs applying Pareto optimisation progress by assigning ranks to its solutions [5]. Eventually, selection has to be made between solutions of the same rank; this is the problem of incomparability where there is no obvious criteria to determine the better solution. Modern MOEAs have unique ways of solving this problem; For example, the Nondominated Sorting Genetic Algorithm II (NSGA-II) [6] which compares equal-ranked solutions using the crowding distance metric aims to maximise the spread of the solution front. Another example is the improved Strength Pareto Evolutionary Algorithm (SPEA-II) [7] which compares solutions based on how many archived solutions it dominates.

Many MOEAs exist today and all have their own characteristics, and there is no generally superior algorithm as a consequence of the “no free lunch” theorem [8]. However there may exist a superior algorithm for a given problem, or more generally a given class of problems. The fitness landscape of a test problem is defined by the mapping of objective parameters to the fitness space. A fitness landscape can be described with features such as its modality (number of optima), bias (the degree of evenness in distribution mapping parameter vectors in search space to objective vectors fitness space), and separability (whether each parameter can be optimised in turn, independent of others).

Certain test problem features will impact upon how well a MOEA performs. For example a test problem with a fitness landscape which is continuous and unimodal (one optima) will benefit the "hill-climb" algorithm which factors heavily the gradient, whilst the same algorithm may struggle on a multimodal fitness landscape with flat regions, where relying on the gradient becomes unhelpful. The process for comparing MOEAs involves:

1. Choose MOEAs to compare
2. Choose test problems to use
3. Choose criteria to compare MOEAs
4. Execute problems using MOEAs and obtain results
5. Compare results
6. Draw conclusion

The second step involves the choice of test problems, and is important to the success of the comparison. Artificially constructed test problems are widely used as they can be implemented to be fast and easily understood. The WFG test problem toolkit [9] is flexible, and allows test problem features such as bias, multi-modality, and non-separability to be varied. The aim of this project is to investigate which MOEAs are better suited to solving test problems of a specific feature set; As a result the WFG toolkit is appropriate for the purposes of this project.

PISA (A Platform and Programming Language Independent Interface for Search Algorithms) [10] is a framework providing the separation and integration of optimisation problems and algorithms. It is useful because implementation of problem and algorithm are done separately like modules, which can be combined with a library of existing problems and algorithms such as the NSGA-II and SPEA-II; PISA also provides performance assessment with a set of statistical tools for analysis. This project will use an implementation of the WFG toolkit within the framework along with numerous existing MOEAs for its investigations.

Aims and Questions

- Investigate whether some MOEAs are better suited to solving problems with certain features.
- Investigate what features result in better performance for different algorithms.
- Aim to demonstrate the "no free lunch" theorem.
- Compare findings with existing work.
- Suggest ways to improve upon the WFG toolkit.

Method

Period	**Task**
2 weeks	Further research into: WFG toolkit PISA
3 weeks	Develop Test strategy
2 weeks	Set up WFG toolkit under PISA
1 weeks	Set up MOEAs to be tested
6 weeks	Conduct testing
3 weeks	Do comparison and analysis
5 weeks	Write up Dissertation

References

[1] C. A. Coello, 2006 ‘Evolutionary Multi-Objective Optimisation: A historical View of the Field’, IEEE Computational Intelligence Magazine, Vol 1, No 1, pp. 28-36.

[2] T. Back, K. Hammel, and H. Shwefel, 1997 ‘Evolutionary Computation: Comments on the History and Current State’, IEEE Transactions on Evolutionary Computation, Vol 1, No 1, pp. 3-17.

[3] C. M. Fonseca, and P. J. Flemming, 1995 ‘An Overview of Evolutionary Algorithms in Multiobjective Optimisation’, Evolutionary Computation, Vol 3, No 1, pp. 1-16.

[4] K. Deb: Multi-Objective Optimisation using Evolutionary Algorithms. John Wiley & Sons, Chichester, UK (2001)

[5] C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, New York (2002)

[6] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II,” In Marc Schoenauer et al., editor, Proceedings of the Parallel Problem Solving from Nature VI Conference, pp. 849-858, Paris, France, 2000.

[7] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the Strength Pareto Evolutionary Algorithm,” in EUROGEN 2001.Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, Eds., Athens, Greece, 2002, pp. 95–100.

[8] Wolpert, D.H.; Macready, W.G., “No free lunch theorems for optimization”, IEEE Transactions on Evolutionary Computation, Volume 1, Issue 1, April 1997 Page(s):67 – 82, Digital Object Identifier 10.1109/4235.585893.

[9] S. Huband, L. Barone, L. While, and P. Hingston, “A Scalable Multiobjective Test Problem Toolkit,” in Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005, C. A. Coello Coello, A. Hern´andez Aguirre, and E. Zitzler, Eds. Guanajuato, M´exico: Springer. Lecture Notes in Computer Science Vol. 3410, March 2005, pp. 280–295.

[10] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler, “PISA—A Platform and Programming Language Independent Interface for Search Algorithms,” in Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003, C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, Eds. Faro, Portugal: Springer. Lecture Notes in Computer Science. Volume 2632, April 2003, pp. 494–508.