PERFORMANCE SCALING
OF
MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Vineet Khare

School of Computer Science
The University of Birmingham
Edgbaston, Birmingham B15 2TT, U.K.
`msc39vxk@cs.bham.ac.uk`

Project Supervisors

Prof. Xin Yao
School of Computer Science,
The University of Birmingham, UK
&
Prof. Kalyanmoy Deb
Indian Institute of Technology, Kanpur
Kanpur, INDIA

September 21, 2002

A report submitted in partial fulfilment of the
requirements for the degree of
MSc in Natural Computation

THE UNIVERSITY
OF BIRMINGHAM

Abstract

In real world problems, one is often faced with the problem of multiple, possibly competing, goals, which should be optimized simultaneously. These competing goals give rise to a set of compromise solutions, generally denoted as Pareto-optimal. If none of the objectives have preference over the other, none of these trade-off solutions can be said to be better than any other solution in the set. Multi-objective Evolutionary Algorithms (MOEAs) can find these optimal trade-offs in order to get a set of solutions that are optimal in an overall sense.

MOEAs are getting immense popularity in the recent past, mainly because of their ability to find a wide spread of Pareto-optimal solutions in a single simulation run. Various evolutionary approaches to multi-objective optimization have been proposed since 1985. Some of fairly recent ones are NSGA-II, SPEA2, PESA (which are included in this study) and others. They all have been mainly applied to two to three objectives. In order to establish their superiority over classical methods and demonstrate their abilities for convergence and maintenance of diversity, they need to be tested on higher number of objectives.

This project mainly investigates two issues - (1) Scalability of these algorithms with respect to the number of objectives, (2) Comparing these algorithms on the basis of -

- How close do they get to Pareto-optimal front?
- How well do they maintain diversity and provide a good spread of solutions on the converged front?
- Their running time.

Experiments were done for 2, 3, 4, 6, and 8 objectives for all three algorithms on four scalable test problems [DTLZ01] namely - DLTZ1, DLTZ2, DLTZ3 and DLTZ6. These problems differ from each other in the type of Pareto-optimal front, number of local Pareto-optimal fronts and the degree of difficulty they provide, to an algorithm, in both converging to the true Pareto-optimal front and maintaining a widely distributed set of solutions.

Keywords: Multi-objective Evolutionary Algorithms (MOEAs), scalability, Pareto Optimality, Perato-optimal front, non-domination, scalable test problems, PESA, SPEA2, NSGA-II.

Acknowledgements

I would like to take this opportunity to thank all those who painstakingly guided me during the project. I acknowledge the efforts of the following people and thank them for their help and co-operation.

First of all I would like to thank Prof. Xin Yao for his constant encouragement and for his useful suggestions from time to time. I would also like to acknowledge the help and guidance that I recieved from Prof. Kalyanmoy Deb (IIT Kanpur, India).

I am also thankful to Dr. Joshua D. Knowles (Université Libre de Bruxelles, Brussels, Belgium) for sending me his PESA source code and Dr. Eckart Zitzler (Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Switzerland) for sending me his SPEA2 source code. Without these codes I would not have been able to complete this project in time. Last but not the least, thanks are also due to the School of Computer Science, University of Birmingham and to the European Commission for supporting this MSc. program through its grant (ASI/B7-301/97/0126-08) in its ASIA IT&C programme.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the world around us it is rare for any problem to concern only a single value or objective. Generally multiple objectives or parameters have to be met or optimized before any solution is considered adequate. In traditional multiobjective optimisation it is usual to simply aggregate (see figure 2.4) together (add in some way) all the various objectives to form a single (scalar) fitness function, which can then be treated by classical techniques such as simple GAs, multiple objective linear programming (MOLP), multiple attribute utility theory (MAUT), random search, simulated annealing etc.

A problem that arises however is how to normalise, prioritise and weight the contributions of the various objectives in arriving at a suitable measure, e.g. when choosing a car how do we compare incommensurable values like size and colour? Also these objectives can interact or conflict with each other, increasing one can reduce others in turn and this can happen in nonlinear ways.

This mapping stage is in itself problematical in that the set of solutions produced is highly dependent upon the value sharing function used and how the weights are assigned. Only if this is adequate will the solutions also be adequate and it is very easy to assign prejudicial or inappropriate sharing factors which can lead to apparently quantitatively exact solutions that are sub-optimal or misleading in practice. Thus a Multiobjective Optimisation Problem requires solving two problems, firstly to establish an objective function and secondly to somehow search this space to locate acceptable solutions (ideally global optima) in terms of the decision variables.

Most of the conventional Operational Research methods of obtaining solutions or approaching the Pareto front focus on the first stage of ranking the objectives, i.e. trying to reduce the design space to a more easily managed mathematical form (since most such problems are far too complex to enumerate and evaluate all the possible combinations in any reasonable time). Some of these techniques are:

- *Stochastic* - very general, but inefficient (e.g. random walk, simulated annealing, Monte Carlo & tabu)
- *Linear Programming* - fast, but restricted to linearised situations only
- *Gradient Based/Hill Climbing* - nonlinear, applicable to smooth(differentiable) functions
- *Simplex Based* - nonlinear for discontinuous functions
- *Sequential Optimisation* - ranks objectives by preference and optimises them in order (lexicographic)
- *Weighting Objectives* - creating a single scalar vector function to optimise, multiple runs needed
- *Constraint* - optimises preferred objective with others treated as constraints
- *Global Criterion* - minimises the distance to an ideal vector
- *Goal Programming* - minimises deviation from target constraints
- *Multiattribute Utility Theory (MAUT)* - maximises preferences or fitnesses

These conventional optimization techniques, which tend to generate elements of the Pareto optimal set one at a time, are difficult to extend to the true multiobjective case (discussed in section 2.6). However, evolutionary algorithms had been pointed out to be possibly well-suited to multiobjective optimization since early in their development [FF95]. This is based on their ability to search for multiple solutions in parallel and to handle complicated tasks such as discontinuities, multi-modality and noisy function evaluations. Also, evolutionary algorithms can find multiple optimal solutions in one single run due to their population based approach. Hence they are ideal candidates for solving multi-objective optimization problems which involve the task of finding more than one optimum solutions.

1.1. MOTIVATION

Evolutionary algorithms are attractive in the context of multi-objective optimization for various reasons listed in section 2.6, but most of the work that has been done in evolutionary multi-objective optimization is restricted to 2 and 3 objectives. The main motivation behind this work is to investigate how these algorithms behave when tested on higher dimensional problems. Scalability of some modern algorithms (section 3.2) is to be investigated using an experimental study. This study would involve experiments with chosen algorithms (section 3.2) on four scalable (in terms of objectives and variables) test problems (chapter 4) for 2 to 8 objectives. For comparison purposes, three performance metrics are to be used (chapter 5).

1.2. MSC. PROJECT PROPOSAL

1.2.1. Aim

To investigate the performance scaling of Multi-objective Evolutionary Algorithms (MOEAs), available in literature, with increase in number of objectives.

1.2.2. Objectives

1. To get familiarised with some of the recent MOEAs (e.g. SPEA2, NSGA-II, PESA) available in literature.
2. To choose 3 performance metrices (one for measuring convergence, second for measuring diversity of obtained solutions and third one for the time complexity of the algorithm)
3. To perform a study investigating the scalability of each algorithm for 2 to 10 objectives; experimentation to be done on four test problems (DLTZ1, DLTZ2, DLTZ3, DLTZ4 [DTLZ01]).
4. To compare the algorithms with respect to the chosen performance metrices.

1.3. ORGANISATION OF REPORT

The rest of the report is organised as follows. Chapter 2 describes, in detail, the multi-objective optimization problem and various concepts related to it. Concepts of domination and pareto-optimality are discussed. A discussion on the two possible approaches to multi-objective optimization is also presented. Chapter 3 gives a brief literature review of different multi-objective evolutionary algorithms, followed by detailed descriptions on the three algorithms used in this study. Common variational operators used for all the algorithms are also presented. Chapter 4 describes the four scalable test problems used for experimentation. A discussion on the performance metrics used is presented in chapter 5. Chapter 6 describes in detail the experiments conducted including the parameters and results. Chapters 7 and 8 present a discussion on results and conclusion, respectively.

Chapter 2

Multi-Objective Optimization

To understand MOEAs we need to understand the Multi-Objective Optimisation Problem (MOOP) that these algorithms are applied to. These problems are the most realistic optimization problems. They require the simultaneous optimization of more than one objective function. Some examples of such problems are:

- In bridge construction, a good design is characterized by low total mass and high stiffness.
- Aircraft design requires simultaneous optimization of fuel efficiency, payload, and weight.
- In chemical plant design, or in design of a groundwater remediation facility, objectives to be considered include total investment and net operating costs.
- A good sunroof design in a car could aim to minimize the noise the driver hears and maximize the ventilation.
- The traditional portfolio optimization problem attempts to simultaneously minimize the risk and maximize the fiscal return.

In the following sections we will look at the differences between single and multi-objective optimization, followed by some concepts related to multi-objective optimization and finally, evolutionary multi-objective optimization with some of the reasons why it should be preferred over classical methods.

2.1. SINGLE & MULTI-OBJECTIVE OPTIMIZATION

There is one essential difference between single objective optimization and multiobjective optimization. This difference is based on the fact, that there is no natural sort order of points of the $n-$dimensional Euclidian space, if $n \geq 2$. Hence, a solution of an optimization problem can not be directly compared with some other. There is a special subspace structure of optimal solutions in multiobjective optimization, referred to as the Pareto set (section 2.5. A solution is Pareto-optimal, if this solution is not dominated by some other solution, i.e. if no change in the optimization problems' domain variables gains increase in all fitness values at once. The set of all Pareto-optimal solutions is the Pareto set (or Pareto-front). The task of multiobjective optimization is generally considered as the search for the Pareto-front.

2.2. MULTI-OBJECTIVE OPTIMIZATION PROBLEM (MOOP)

The Multi-Objective Optimization Problem (MOOP) (also called multicriteria optimization, multiperformance or vector optimization problem) can be defined as the problem of [Osy85]: *Finding a vector of decision variables x, which optimizes a vector function*

$$f_m(\mathbf{x}), \qquad m = 1, 2, \ldots, M;$$

satisfies inequality

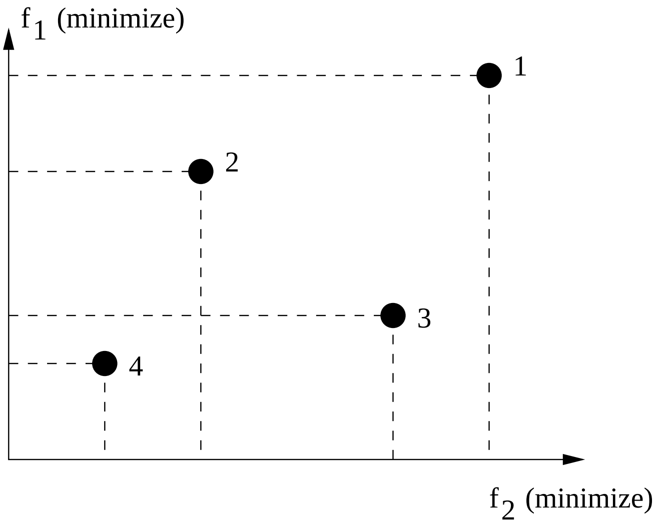


Figure 2.1: Concept of Domination

$$g_j(x) \geq 0, \qquad j = 1, 2, \ldots, J;$$

and equality constraints

$$h_k(x) = 0, \qquad k = 1, 2, \ldots, K;$$

and whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term optimize means finding such a solution which would give the values of all the objective functions acceptable to the decision maker.

These objective function constitute a multi-dimensional space in addition to the usual decision space. This additional space is called the *objective space,* $\mathcal{Z}$. For each solution $\mathbf{x}$ in the decision variable space, there exist a point in the objective space:

$$\mathbf{f}(\mathbf{x}) = \mathcal{Z} = (z_1, z_2, \ldots, z_M)^T$$

2.3. CONCEPT OF DOMINANATION

Most multi-objective optimization algorithm use the concept of domination. In these algorithms, two solutions are compared on the basis of whether one dominates the other solution or not.

Any solution $\mathbf{x}^{(1)}$ *is said to dominate* $\mathbf{x}^{(2)}$ *or* $\mathbf{x}^{(1)}$ *is said to be non-dominated by* $\mathbf{x}^{(2)}$ *if both the conditions 1 and 2 are true:*

1. $\mathbf{x}^{(1)}$ *is no worse than* $\mathbf{x}^{(2)}$ *in all objectives.*
2. $\mathbf{x}^{(1)}$ *is strictly better than* $\mathbf{x}^{(2)}$ *in at least one objective.*

Say, we have two objective functions f_1 and f_2, both to be minimized. In figure 2.1 we have 4 solutions and

- Solution 4 dominates solution 1, 2 and 3.
- Solution 2 and 3 dominate solution 1.

If any of the two coditions mentioned above is violated the solution $\mathbf{x}^{(1)}$ does not dominate $\mathbf{x}^{(2)}$. Hence in figure 2.1 neither of solutions 2 or 3 dominate each other.

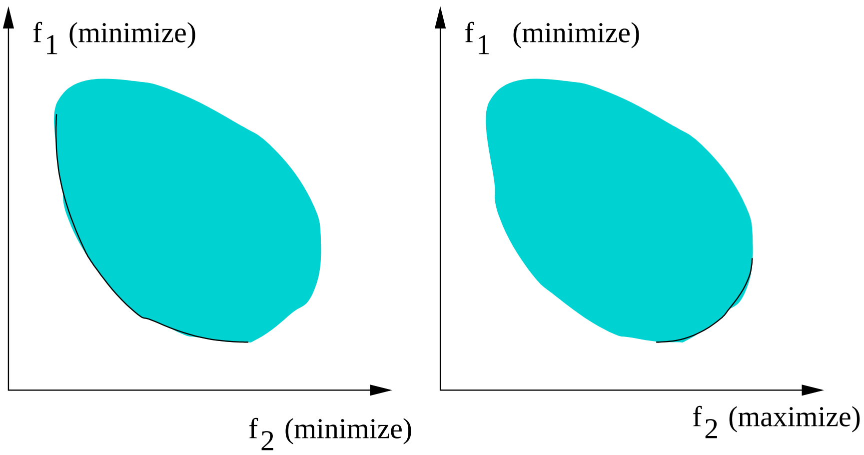


Figure 2.2: Different PO solution sets, for the same objective space, depending on maximization or minimization problem

2.4. NON-DOMINATED SET

For a given finite set of solutions, if we perform pairwise comparisons to find out which solutions dominate which and which are dominated by each other. We can find a subset of solutions such that, any two solutions of which do not dominate each other and all the other solutions of the finite set are dominated by one or more members of this subset. This subset is called the *non-dominated set* for the given set of solutions. Or in other words [Deb01]:

Among a set of solutions $\mathcal{P}$, the non-dominated set of solutions $\mathcal{P}'$ are those that are not dominated by any member of set $\mathcal{P}$.

2.5. PARETO OPTIMALITY

Pareto optimality can be defined as the *best that could be achieved without disadvantaging at least one group*. A single-objective optimization problem may have (and usually does have) a single-valued, unique solution. The solution to a MOOP is, as a rule, not a particular value, but a set of values of decision variables such that, for each element in this set, none of the objective functions can be further increased without a decrease of some of the remaining object functions (every such value of a decision variable is referred to as pareto-optimal (PO)).

When the set $\mathcal{P}$ (section 2.4) is the entire search space, or $\mathcal{P} = \mathcal{S}$, the resulting non-dominated set $\mathcal{P}'$ is called the *Pareto-optimal (PO) set*. Figure 2.2 the shaded area represents the entire objective space. In the left figure, the PO solution set is shown with a dark curve, if we choose to minimize both f_1 and f_2. But if we choose to minimize f_1 and maximize f_2, the resulting PO solution set is shown in the right figure.

Like global and local optimal solutions in the case of single objective optimization, there could be global and local PO sets in multi-objective optimization.

2.5.1. Globally Pareto-optimal set

Globally PO set is the non-dominated set of entire feasible search space $\mathcal{S}$. Since the solutions of this set are not domonated by any feasible member of the search space, they are the *optimal solutions for MOOP*.

2.5.2. Locally Pareto-optimal set

A locally Pareto-optimal set can be defined as [Deb01, pages 31–32]

If $\forall \mathbf{x} \in \underline{\mathcal{P}}$, $\exists$ no $\mathbf{y}$ (in the neighbourhood of $\mathbf{x}$, such that,

$$\| \mathbf{x} - \mathbf{y} \|_\infty \leq \epsilon,$$

where ϵ is a small positive number) dominating any member of the set $\underline{\mathcal{P}}$, then solutions belonging to set $\underline{\mathcal{P}}$ constitute a locally Pareto-optimal set.

In figure 2.3 the shorter curve represents a local PO front. None of its points have any neighbour which dominate any member of this set.

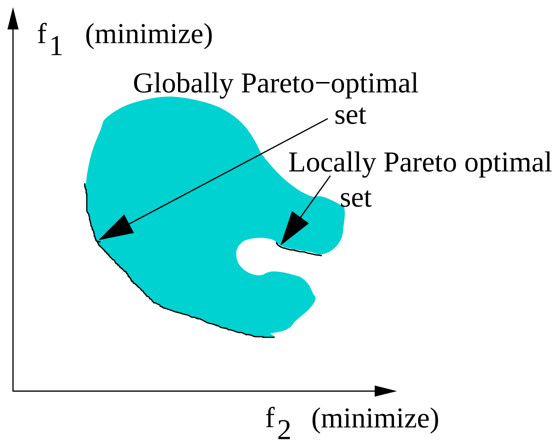


Figure 2.3: Locally Pareto-optimal set

2.6. EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION

In absence of any further information, none of the solutions on a pareto front can be said to be better than the others on the same front. All of the solutions on the PO front are optimal. This is the fundamental difference between a single and multi-objective optimization task. Even with this fundamental difference the user will finally require one solution for his/her practical purposes. The decision as to which solution to choose, requires some high level information. At some stage of the problem solving process the decision-maker is to articulate his/her preferences about the objectives. Depending on the stage of optimization where this high level information is used, there can be two possible approaches to MOOP.

1. *Preference-Based Approach OR a priori* Decision-maker combines the differing objectives into a scalar cost function (figure 2.4) and converts the MOOP to single objective optimization problem. Now any single objective optimizer can be used to obtain one single solution. This procedure can be repeated again and again to find multiple trade-off solutions by using a different cost function.

2. *Ideal Multi-Objective Optimization OR a posteriori* First a multi-objective optimizer is used to find multiple trade-off optimal solution with a wide range of values for the objectives, then one solution is chosen from them using the higher level information.

The trade-off solution obtained by using preference based strategy is largely sensitive to relative preference vector used in forming the composite cost function. A change in this preference vector will result in a different trade-off solution. Classical multi-objective optimization methods, which convert multiple objectives into single objective by using a relative preference vector of objectives, work according to this preference-based strategy. Unless a reliable and accurate preference vector is available, the optimal solution obtained by such methods is highly subjective to the particular user.

Evolutionary algorithms seem particularly suitable to solve MOOP problems, because they deal simultaneously with a set of possible solutions or population. This allows them to find several members of the Pareto optimal set in a single run of the algorithm, instead of having to perform a series of separate runs as in the case of the classical techniques. Additionally, evolutionary algorithms are less susceptible to the shape or continuity of the Pareto front (e.g., they can easily deal with discontinuous or concave Pareto fronts), whereas these two issues are a real concern classical techniques.

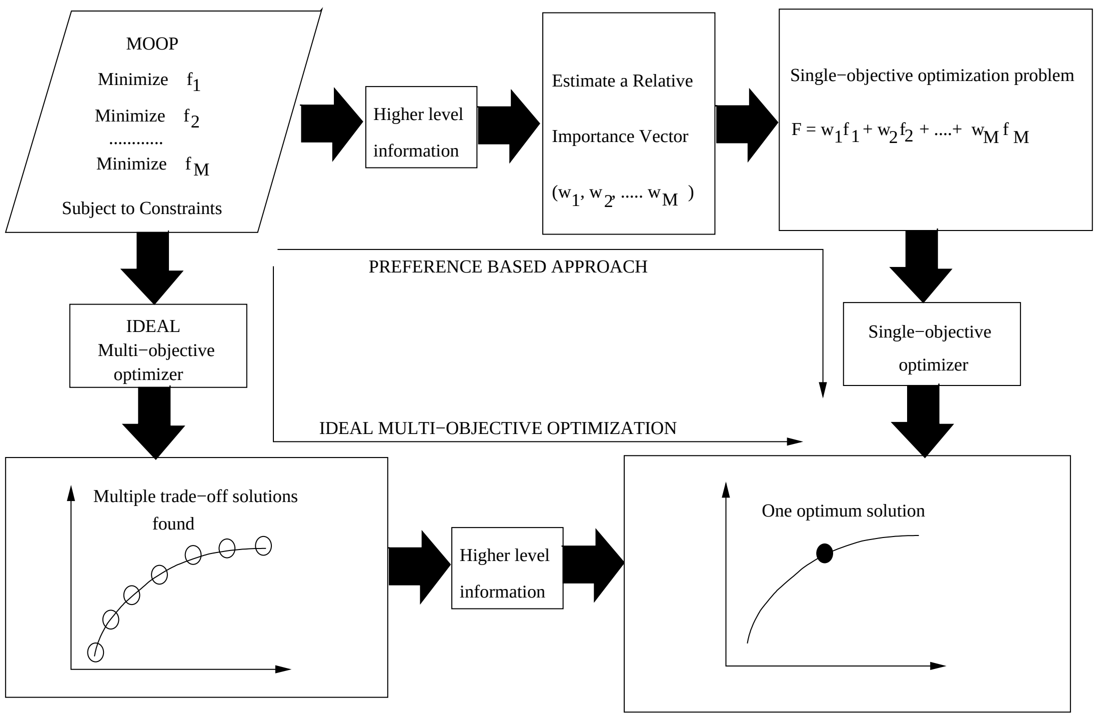


Figure 2.4: Two approaches to multi-objective optimization

Chapter 3

Description of Algorithms Used

This chapter presents a brief literature review on various MOEAs, which is by no means complete but can provide the reader with a better understanding of the algorithms (section 3.2) used in this study. Starting from the very first MOEA proposed, a description of different algorithms (section 3.1) which incorporate elitism is presented, followed by the detailed description of some of the state-of-the-art MOEAs (section 3.2) which are used in this comparative study.

In order to make the comparisons fair, common variational operators were used in all the three algorithms. Simulated Binary Crossover [DK95] and polynomial mutation [DG96] were used. Last section (section 3.3) describes these common operators.

Some evolutionary multi-objective optimization algorithms were developed in the early nineties, based on combining the ideas [Gol89] of

- *Pareto dominance* -to exploit the search space in the direction of the Pareto front
- *Niching techniques* -to explore the search space along the front to keep diversity

The very first one in them was - **MOGA** (Multi-Objective Genetic Algorithm) [FF93], in which each individual was ranked on the basis of the number of individuals by which it is dominated. The distribution of individuals over the Pareto region was performed by a fitness sharing procedure. Later, in 1994, two other MOEAs were proposed - **NSGA** and **NPGA**. In NSGA (Non-dominated Sorting Genetic Algorithm) [SD94],The rank of each individual was based on the rank of the front it belongs to. The distribution of individuals over the Pareto region was performed by a fitness sharing procedure. NPGA (Niched Pareto Genetic Algorithm) [HN93] was also based on the Pareto dominance concept but differed from NSGA and MOGA in selection. NPGA used tournament selection instead of proportionate selection like in NSGA and MOGA. The distribution of individuals over the Pareto region was performed by a fitness sharing procedure.

All the algorithms mentioned above had two shortcomings in general:

1. They use, as fitness sharing as a tool to keep diversity in the population through the whole Pareto front and hence they require a fitness sharing factor to be set.
2. They did not incorporate elitism, which has been demonstrated to improve significantly the performance of multi-objective algorithms [PM98, ZDT00].

3.1. ELITIST AND SHARING PARAMETERLESS MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Elitism maintains the knowledge acquired during the algorithm execution by conserving the individuals with best fitness in the population or in an auxiliary population. Some algorithms that make use of both improved concepts (elitism and no sharing factor) are given in the following sections.

3.1.1. PAES

PAES (Pareto Archived Evolution Strategy) [KC99] can be viewed as (1+1) ES but in addition to the parent and the offspring, an archive of best solutions found so far is also maintained at each generation. A new crowding method is introduced in this algorithm to promote diversity in the population. The objective space is divided into hypercubes by a grid which determines the density of individuals; the zones with lower density are favoured in detriment of the zones with higher density of points. This technique depends only on the parameter of number of grid divisions and is less computationally expensive than niching, avoiding the use of the fitness sharing factor. Starting with an initial random solution and an empty archive the following steps are performed per iteration (say t):

Parent $p(t)$ is mutated, by using a normally distributed probability function with zero mean and a fixed mutation strength, to obtain an offspring $c(t)$, which is then compared with the parent. Now there are three scenarios possible:

- *If* $p(t)$ *dominates* $c(t)$, $c(t)$ is discarded and we move to next iteration.
- $c(t)$ *dominates* $p(t)$ we accept $c(t)$ as the parent for next generation and a copy of it is kept in the archive.
- *If neither solution dominates* $c(t)$ is compared with the population of previously archived non-dominated solutions. Now again three cases are possible here:

  1. *If* $c(t)$ *is dominated by a member of archive* it is rejected and we move to next iteration.
  2. *If* $c(t)$ *dominates a member(s) of the archive,* dominated members are deleted and $c(t)$ is accepted as parent for next generation.
  3. *If neither* $c(t)$ *nor any archive member dominate the other,* $c(t)$ is added to archive only if the archive size hasn't already reached its maximum. If $c(t)$ is accepted and it belongs to a grid less crowded than that of the $p(t)$, it qualifies to become the parent for the next generation. If archive is full the above density based comparison between $p(t)$ and $c(t)$ decides who remains in the archive.

Initially conceived as a multi-objective local search method $(1 + 1)$-PAES, it has been extended later to the $(\mu + \lambda)$-PAES.

3.1.2. SPEA

Zitzler and Thiele [ZT99] proposed Strength Pareto Evolutionary Algorithm(SPEA), which stores the solutions of the best front found in an external auxiliary population. A clustering method (average linkage method) based on objective space was implemented to preserve diversity in the population, avoiding the use of any parameter such as the fitness sharing factor. Starting with an initial population and an empty archive the following steps are performed per iteration:

All non-dominated population members are copied to the archive; any dominated individuals or duplicates (regarding the objective values) are removed from the archive during this update operation. If the size of the updated archive exceeds a predefined limit, further archive members are deleted by a clustering technique which preserves the characteristics of the non-dominated front.

Step1: fitness values are assigned to both archive and population members:

- Each individual i in the archive is assigned a strength value $S(i) \in [0, 1)$, which at the same time represents its fitness value $F(i)$. $S(i)$ is the number of population members j that are dominated by or equal to i with respect to the objective values, divided by the population size plus one.
- The fitness $F(j)$ of an individual j in the population is calculated by summing the strength values $S(i)$ of all archive members i that dominate or are equal to j, and adding one at the end.

Step2: The next step represents the mating selection phase where individuals from the union of population and archive are selected by means of binary tournaments. Since the fitness is to be minimized here, each individual in the archive has a higher chance to be selected than any population member.

Step3: Finally, after recombination and mutation the old population is replaced by the resulting offspring population.

3.1.3. NSGA-II

NSGA-II [DPAM02] maintains the solutions of the best front found including them into the next generation. The rank of each individual is based on the concept of the level of non-domination corresponding to the NSGA. Moreover in this proposal is introduced a faster algorithm to sort the population that takes $O(mN^2)$ computations, instead of the previous one of $O(mN^3)$, where m is the number of objectives considered and N is the number of population members. A crowding distance is evaluated that considers the size of the largest cuboid enclosing each individual without including any other of the population. This parameter is used to keep diversity in the population and points belonging to the same front and with higher crowding distance are assigned a better fitness than those with lower crowding distance, avoiding the use of the fitness sharing factor. A detailed description of the algorithm is given in the next section.

3.2. MOST RECENT ALGORITHMS - USED IN THE STUDY

Following are the most recently proposed algorithms which have been used in this comparative study.

3.2.1. PESA

In PESA (Pareto Enveloped-based Selection Algorithm) [CKO00], not only the crowding mechanism is based on the hyper-cubes grid division as in PAES, but also the selection criterion is performed by this concept. Apart from standard parameters such as crossover and mutation rates, PESA has two parameters concerning population size (P_I&P_E), and one parameter concerning the hyper-grid crowding strategy. P_I is the size of internal population IP and P_E is the maximum size of the archive, or 'external population', EP. The third parameter is called the *Squeeze Factor* (figure 3.1) and is used in selection and archive update in PESA. The crowding strategy in PESA works by forming an implicit hyper-grid which divides (normalised) phenotype space into hyper-boxes.

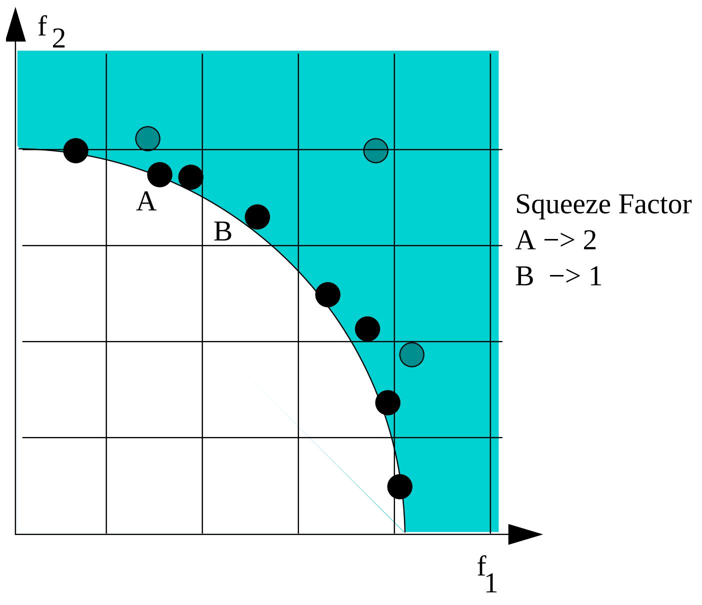


Figure 3.1: Squeeze factor and hyper-grid crowding strategy in PESA (for 2objective DLTZ2 or DLTZ3).

In Figure 3.1, hypergrids are illustrated by the thick horizontal and vertical lines; the problem is two-dimensional and hence these hyper-boxes are simply squares. Each chromosome in the archive is associated with a particular hyper-box in phenotype space, and has the squeeze factor equal to the total number of other chromosomes in the archive which inhabit the same box.

Step 1: Initialization Generate and evaluate each of an initial 'internal' population (IP) of P_I chromosomes, and initialise the 'external' population (EP) to the empty set.

Step 2: Archive Incorporation Incorporate the non-dominated members of IP into EP. A candidate may enter the archive if it is non-dominated within IP, and if is not dominated by any current member of the archive. Once a candidate has entered the archive, members of the archive which it dominated (if any) will be removed. If the addition of a candidate renders the archive over-full (its size temporarily becomes $P_E + 1$), then a current member of EP is removed. This choice is made by first finding the maximal squeeze factor in the population, and removing an arbitrary chromosome which has this squeeze factor.

Step 3: Termination If a termination criterion has been reached, then stop, returning the set of chromosomes in EP as the result. Otherwise, delete the current contents of IP, and repeat the following until P_I new candidate solutions have been generated: - With probability p_C, select two parents from EP, produce a single child via crossover, and mutate the child. With probability $(1 - p_C)$, select one parent and mutate it to produce a child.

Step 4: return to Step 2.

3.2.2. SPEA2

SPEA2 [ZLT01] was proposed as an improvement of SPEA. Authors in [ZLT01] identified some weaknesses of SPEA, e.g.

- If the archive contains only a single individual, all population members have the same rank independent of whether they dominate each other or not. As a consequence, the selection pressure is decreased substantially and in this particular case SPEA behaves like a random search algorithm.
- If many individuals of the current generation are indifferent, i.e., do not dominate each other, none or very little information can be obtained on the basis of the partial order defined by the dominance relation. In this situation, which is very likely to occur in the presence of more than two objectives, density information has to be used in order to guide the search more effectively. Clustering makes use of this information, but only with regard to the archive and not to the population.
- Clustering technique used in SPEA may lose outer solutions.

SPEA2 was designed to overcome the aforementioned problems. The overall algorithm Can be presented in the following steps:

Input:	N	(population size)
	$\overline{N}$	(archive size)
	T	(maximum number of generations)
Output:	A	(non-dominated set)

Step 1: Initialization: Generate an initial population P_0 and create the empty archive (external set) $\overline{P}_0 = \emptyset$. Set $t = 0$.

Step 2: Fitness assignment: Calculate fitness values of individuals in P_t and $\overline{P}_t$.Each individual i in the archive $\overline{P}_t$ and the population P_t is assigned a strength value $S(i)$, representing the number of solutions it dominates

$$S(i) = |\{j | j \in P_t + \overline{P}_t \wedge i \succ j\}| \tag{3.1}$$

where $|\cdot|$ denotes the cardinality of the set, $+$ stands for multiset union and the symbol $\succ$ corresponds to the pareto dominance relation. On the basis of S value, the raw fitness $R(i)$ of an individual i is calculated

$$R(i) = \sum_{j \in P_t + \overline{P}_t, j \succ i} S(j) \tag{3.2}$$

That is the raw fitness is determined by the strengths of its denominators in both archive and population, as opposed to SPEA where only archive members.In addition, density information is incorporated to discriminate between individuals having identical raw fitness values. The density estimation technique in SPEA2 is an adaptation of the $k-th$ nearest neighbor method, where the density at any point is a decreasing function of distance to the $k-th$ nearest data point. Here the inverse of distance to the $k-th$ neighbor is used as the density estimate. For each individual i the distances (in objective space) to all individuals j in archive and population are calculated and stored in a list. After sorting the list in increasing order , the $k-th$ element gives the distance sought, denoted as σ_i^k. $k = \sqrt{N + \overline{N}}$ is used as a common setting. Now the density

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{3.3}$$

In the denominator, two is added to ensure that its value is greater than zero and that $D(i) < 1$. Finally, adding $D(i)$ to the raw fitness value if an indiviual i gives its fitness $F(i)$

$$F(i) = R(i) + D(i) \tag{3.4}$$

Step 3: **Environmental selection:** Copy all non-dominated individuals in P_t and $\overline{P}_t$ to $\overline{P}_{t+1}$. Now there are three possible scenarios:

1. If $|\overline{P}_{t+1} = \overline{N}|$, the environmental selection step is complete.
2. If $|\overline{P}_{t+1} > \overline{N}|$, then reduce $\overline{P}_{t+1}$ by means of the *truncation operator*. This operator iteratively removes individuals from $\overline{P}_{t+1}$ until $|\overline{P}_{t+1} = \overline{N}|$. At each iteration an individual i is chosen for removal for which $i \leq_d j \; \forall j \in \overline{P}_{t+1}$ with

$$\begin{aligned} i \leq_d j \quad :\Leftrightarrow \quad & \forall 0 < k < |\overline{P}_{t+1}| : \sigma_i^k = \sigma_j^k \vee \\ & \exists 0 < k < |\overline{P}_{t+1}| : [(\forall 0 < l < k : \sigma_i^l = \sigma_j^l) \wedge \sigma_i^k < \sigma_j^k] \end{aligned}$$

   where σ_i^k denotes the distance of i to its $k-th$ neighbor in $\overline{P}_{t+1}$. In other words the individual which has the minimum distance to another individual is chosen at each stage; ties are broken by considering the second smalles distance and so on. Figure 3.2 illustrates the working of this truncation operator for 2 objective DLTZ2 or DLTZ3 problem with $\overline{N} = 5$.
3. if $|\overline{P}_{t+1} < \overline{N}|$, then fill $\overline{P}_{t+1}$ with dominated individuals in P_t and $\overline{P}_t$. This can be implemented by sorting the multiset $P_t + \overline{P}_t$ according to the fitness values and copy the first $\overline{N} - |\overline{P}_{t+1}|$ individuals i with $F(i) \geq 1$ from the resulting ordered list to $\overline{P}_{t+1}$.

Step 4: **Termination:** If $t \geq T$ or another stopping criterion is satisfied then set A to the set of decision vectors represented by the non-dominated individuals in $\overline{P}_{t+1}$ and stop.

Step 5: **Mating selection:** Perform binary tournament selection with replacement on $\overline{P}_{t+1}$ in order to fill the mating pool.

Step 6: **Variation:** Apply recombination and mutation operators to the mating pool and set P_{t+1} to the resulting population. Increment generation counter $(t = t + 1)$ and go to Step 2.

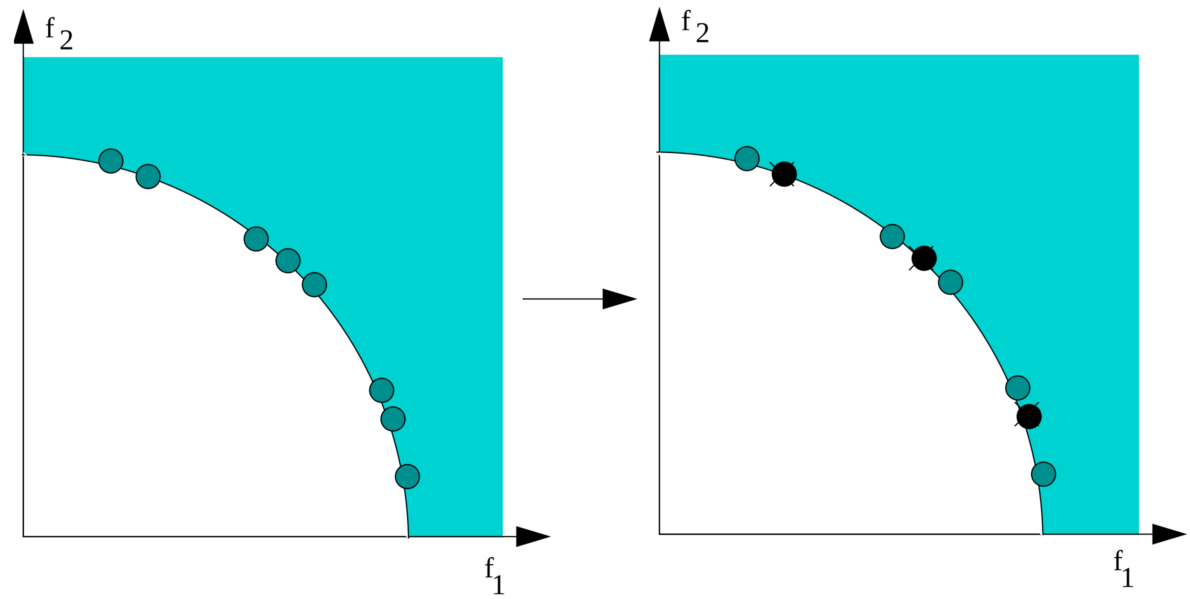


Figure 3.2: Archive truncation in SPEA2: black circles in the right figure are deleted by the truncation operator (Assuming 2objective DLTZ2 or DLTZ3 problem with $\overline{N} = 5$).

3.2.3. NSGA-II

NSGA-II [DAPM00] was proposed, as a modification of NSGA, to alleviate the three difficulties associated with NSGA. It incorporates elitism, doesn't require any sharing parameter to be set and uses a *fast non-dominated sorting approach* which makes it faster (section 3.1.3) than NSGA. Details of fast non-dominated sorting approach can be found in [DAPM00].

Algorithm can be described as follows. Initially, a random parent population P_0 is created. The population is sorted based on the non-domination. Each solution is assigned a fitness equal to its non-domination level (1 is the best level). Thus, minimization of fitness is assumed. Binary tournament selection, recombination, and mutation operators are used to create a child population Q_0 of size N. Further, the NSGA-II procedure can be outlined in the following steps:

Step 1: **Combine population** Combine parent and children population to create $R_t = P_t \cup Q_t$.

Step 2: **Non-domianted Sorting** Perform non-dominated sorting to R_t and identify different fronts $\mathcal{F}_i, i = 1, 2, \ldots, etc.$

Step 3: **Creating new population** Set new population $P_{t+1} = \emptyset$. Set counter $i = 1$.
Untill $|P_{t+1}| + |\mathcal{F}_i| < N$, perform $|P_{t+1}| = |P_{t+1}| \cup |\mathcal{F}_i|$ and $i = i + 1$.

Step 4: **Crowding Sort** Perform the crowding sort $(\mathcal{F}, <_c)$ procedure (described in detail below) and include the most widely spread $(N - |P_{t+1}|)$ solutions by using the crowding distance values in the sorted $\mathcal{F}_i$ to P_{t+1}.

Step 5: **Creating offspring population** Create offspring population Q_{t+1} from P_{t+1} by using the crowded tournament selection (described below), crossover and mutation operators.

A discussion on crowding sort involved in step 4 and crowded tournament selection in step 5 are given in the following sections.

3.2.3.1. Crowding-distance-assignment($\mathcal{L}$)

We need to calulate the density of solutions in $\mathcal{F}$ to do the crowding sort in step 4. To get an estimate of the density of solutions surrounding a particular point in the population, the average distance of the two points on either side of this point along each of the objectives is calculated. This quantity $i_{distance}$ serves as an estimate of the size of the largest cuboid enclosing the point i without including any other point in the population. This distance, $i_{distance}$, is called the crowding distance for the individual i and it is used for crowding sort.

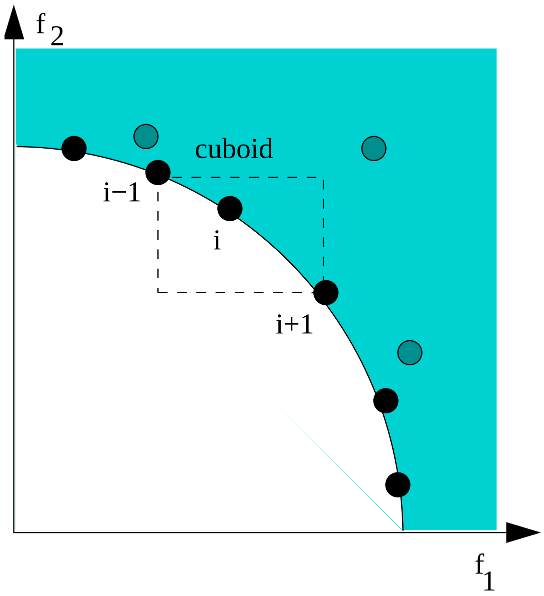


Figure 3.3: Crowding distance calculation (2objective DLTZ2 or DLTZ3 problem).

In Figure 3.3, the crowding distance of the $i-th$ solution in its front (marked with solid circles) is the average side-length of the cuboid (shown with a dashed box). The following algorithm is used to calculate the crowding distance of each point in a front.

$l = \|\mathcal{L}\|$	number of solutions in $\mathcal{L}$
for each i set $\mathcal{L}[i]_{distance}$	initialize distance
for each objective m	
$\mathcal{L} =$ sort$(\mathcal{L}, m)$	sort using each objective value
$\mathcal{L}[1]_{distance} = \mathcal{L}[l]_{distance} = \infty$	so that boundary points are always selected
for $i = 2$ to $(l-1)$	for all other points
$\mathcal{L}[i]_{distance} = \mathcal{L}[i]_{distance} + (\mathcal{L}[i+1].m - \mathcal{L}[i-1].m)$	

Here $\mathcal{L}[i].m$ refers to the $m-th$ objective function value of the $i-th$ individual in the set $\mathcal{L}$.

3.2.3.2. Crowded tournament selection operator

In crowded tournament selection, a solution i wins a tournament with another solution j if any of the following conditions are true:

1. i has a better non-domination rank than j, i.e. $r_i < r_j$.
2. If they have the same rank but solution i has a better crowding distance than solution j, i.e., $r_i = r_j$ and $i_{distance} > j_{distance}$.

3.3. VARIATIONAL OPERATORS USED

Since the test problems that we are dealing with have a continuous space, real encoding should be preferred to avoid problems related to hamming cliffs and to achieve arbitrary precision in the optimal solution. For this reason, in all the algorithms real-coded parameters were used and crossover and mutation operators were applied directly to real parameter values. The real parameter crossover and mutation operators used in this study are described in the following sections.

3.3.1. Simulated Binary Crossover

Simulated Binary Crossover (SBX) [DK95] creates two offspring solutions from two parents. SBX operator imitates the working principle of a single point crossover operator on binary strings. Authors in this paper showed that the common interval schemata between the parents are preserved in the offspring, thus SBX respects the interval schemata. The following steps are involved in creating two offspring ($x_i^{(1,t+1)}$ and $x_i^{(2,t+1)}$) solutions from two parents ($x_i^{(1,t)}$ and $x_i^{(2,t)}$).

Step 1: Choose a random number $u_i \in [0, 1)$.

Step 2: Calculate a spread factor, β_i as the ratio of the absolute difference in offspring values to that of parents:

$$\beta_i = \left| \frac{x_i^{(2,t+1)} - x_i^{(1,t+1)}}{x_i^{(2,t)} - x_i^{(1,t)}} \right| \tag{3.5}$$

From the following probability density function, the ordinate β_{qi} is found so that the area under the probability curve from 0 to β_{qi} is equal to u_i.

$$\mathcal{P}(\beta_i) = \begin{cases} 0.5(\eta_c + 1)\beta_i^{\eta_c}, & \text{if } \beta_i \leq 1; \\ 0.5(\eta_c + 1)\frac{1}{\beta_i^{\eta_c+2}}, & \text{otherwise} \end{cases} \tag{3.6}$$

where η_c is any non-negative real number. We will call this parameter *distribution index for crossover*. This parameter needs to be set by the user. A large value of η_c gives a higher probability for creating a near-parent solutions and a small value of η_c allows distant solutions to be selected as offspring. Using equation (3.6), we calculate β_{qi} by equating the area under the probability curve to u_i, as follows:

$$\beta_{qi} = \begin{cases} (2u_i)^{\frac{1}{\eta_c+1}}, & \text{if } u_i \leq 0.5; \\ (\frac{1}{2(1-u_i)})^{\frac{1}{\eta_c+1}}, & \text{otherwise} \end{cases} \tag{3.7}$$

Step 3: Compute the offspring using:

$$x_i^{(1,t+1)} = 0.5[(1 + \beta_{qi})x_i^{(1,t)} + (1 - \beta_{qi})x_i^{(2,t)}] \tag{3.8}$$

$$x_i^{(2,t+1)} = 0.5[(1 - \beta_{qi})x_i^{(1,t)} + (1 + \beta_{qi})x_i^{(2,t)}] \tag{3.9}$$

Here the offspring produced are symmetric about the parent solutions and for a fixed η_c the offspring have a spread which is proportional to that of the parent solutions. From equations (3.8) and (3.9) we can have:

$$(x_i^{(2,t+1)} - x_i^{(1,t+1)}) = \beta_{qi}(x_i^{(2,t)} - x_i^{(1,t)}) \tag{3.10}$$

which implies that if two parents are far away, say at the initial populations when the solutions are randomly placed, almost any value of offspring can be achieved (*Exploration*). However when the solutions tend to converge due to the actions of genetic operators (and hence the parents are close to each other) distant solutions are not allowed, thereby focussing the search to a narrow region (*Exploitation*). Thus SBX helps in exploring the search space at the initial generation, while it exploits the acquired knowledge at later stages.

3.3.2. Polynomial Mutation

In polynomial mutation [DG96], following steps are involved in mutating a solution $x_i^{(1,t+1)}$ to obtain the offspring $y_i^{(1,t+1)}$.

Step 1: Choose a random number $u_i \in [0, 1)$.

Step 2: Calculate the parameter $\overline{\delta}_i$ from the probability density function $\mathcal{P}(\delta) = 0.5(\eta_m + 1)(1 - |\delta|)^{\eta_m}$ as:

$$\overline{\delta}_i = \begin{cases} (2u_i)^{\frac{1}{\eta_m+1}} - 1, & \text{if } u_i < 0.5; \\ 1 - [2(1 - u_i)]^{\frac{1}{\eta_m+1}}, & \text{otherwise} \end{cases} \tag{3.11}$$

Step 3: Find the offspring using:

$$y_i^{(1,t+1)} = x_i^{(1,t+1)} + (x_i^{\mathcal{U}} - x_i^{\mathcal{L}})\overline{\delta}_i, \tag{3.12}$$

where $x_i^{\mathcal{U}}$ and $x_i^{\mathcal{L}}$ are the lower and upper bounds on the decision variable x. The shape of the probability distribution is directly controlled by the user defined parameter - *Distribution index for mutation* (η_m).

Chapter 4

Test Problems

Most earlier studies on MOEAs introduced test problems which were either simple or not scalable [Sch84, Kur91, FF95]. Few others were too complicated to visualize the exact shape and location of the resulting PO front [VFM96]. In this study four scalable test problems, namely DLTZ1, DLTZ2, DlTZ3 and DLTZ6, are used for the following reasons:

1. *Simplicity of construction* - These problems are quite easy to construct using the *Bottom-up approach* discussed in [DTLZ01]. In this approach the PO front is first assumed in the objective space and an overall objective search space is constructed from the front to define the test problem.

2. *Scalability* - These problems can be scaled to any number of decision variables and objectives.

3. *Knowledge of exact shape and location of the resulting PO front* - The resulting PO is very easy to comprehend and its exact shape and location is known. The corresponding decision variable values are also known.

4. *Ability to control difficulties in both converging to the true PO front and maintaining a widely distributed set of solutions* They can introduce controllable hindrance to converge to the true PO front and also to find a widely distributed set of PO solutions.

4.1. TEST PROBLEM DLTZ1

Equation (4.1) is an $M-$ objective test problem with linear PO front. The functional $g(\mathbf{x}_M)$ requires $|\mathbf{x}_M| = k$ variables and g can be any function with, $g \geq 0$. Equation (4.1) also has the suggested [DTLZ01] value of $g(\mathbf{x}_M)$.

$$\left.\begin{array}{rcl}
Minimize\ \ f_1(\mathbf{x}) & = & \frac{1}{2}x_1x_2\cdots x_{M-1}(1+g(\mathbf{x}_M)), \\
Minimize\ \ f_2(\mathbf{x}) & = & \frac{1}{2}x_1x_2\cdots(1-x_{M-1})(1+g(\mathbf{x}_M)), \\
\vdots & & \\
Minimize\ \ f_{M-1}(\mathbf{x}) & = & \frac{1}{2}x_1(1-x_2)(1+g(\mathbf{x}_M)), \\
Minimize\ \ f_M(\mathbf{x}) & = & \frac{1}{2}(1-x_1)(1+g(\mathbf{x}_M)), \\
 & 0 \leq x_i \leq 1, & for\ \ i = 1,2,\ldots,n, \\
where\ \ g(\mathbf{x}_M) & = & 100\left[|\mathbf{x}_M| + \sum_{x_i \in \mathbf{x}_M}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))\right].
\end{array}\right\} \quad (4.1)$$

The PO solution corresponds to $\mathbf{x}_M = \{0.5, 0.5, \ldots, 0.5\}^T$ and the objective function values lie on the linear hyper plane:

$$\sum_{m=1}^{M} f_m = 0.5$$

Figure 4.1 show the PO front for 3 objectives. The total number of variables in the above problem is $n = M + k - 1$. Where k can be set by the user giving him the freedom to scale to any number of variables. A value of $k = 5$ is used in this study. The difficulty in the problem is to converge to the global PO front avoiding $(11^k - 1)$ local PO fronts.

4.2. TEST PROBLEM DLTZ2

$$\left.\begin{array}{rcl} Minimize\ \ f_1(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\cos(x_1\frac{\pi}{2})\cos(x_2\frac{\pi}{2})\cdots\cos(x_{M-2}\frac{\pi}{2})\cos(x_{M-1}\frac{\pi}{2}), \\ Minimize\ \ f_2(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\cos(x_1\frac{\pi}{2})\cos(x_2\frac{\pi}{2})\cdots\cos(x_{M-2}\frac{\pi}{2})\sin(x_{M-1}\frac{\pi}{2}), \\ Minimize\ \ f_3(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\cos(x_1\frac{\pi}{2})\cos(x_2\frac{\pi}{2})\cdots\sin(x_{M-2}\frac{\pi}{2}), \\ \vdots & & \\ Minimize\ \ f_{M-1}(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\cos(x_1\frac{\pi}{2})\sin(x_2\frac{\pi}{2}), \\ Minimize\ \ f_M(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\sin(x_1\frac{\pi}{2}), \\ & 0 \le x_i \le 1, & for\ \ i = 1,2,\ldots,n, \\ where\ \ g(\mathbf{x}_M) & = & \sum_{x_i\in\mathbf{x}_M}(x_i-0.5)^2. \end{array}\right\} \quad (4.2)$$

The PO solution to this problem corresponds to $\mathbf{x}_M = \{0.5, 0.5, \ldots, 0.5\}^T$ and all objective values satisfy:

$$\sum_{m=1}^{M} f_i^2 = 1;$$

Figure 4.1 show the PO front for 3 objectives. The total number of variables in the above problem is $n = M + k - 1$. Where k can be set by the user. A value of $k = |\mathbf{x}_M| = 10$ is used.

4.3. TEST PROBLEM DLTZ3

$$\left.\begin{array}{rcl} Minimize\ \ f_1(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\cos(x_1\frac{\pi}{2})\cos(x_2\frac{\pi}{2})\cdots\cos(x_{M-2}\frac{\pi}{2})\cos(x_{M-1}\frac{\pi}{2}), \\ Minimize\ \ f_2(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\cos(x_1\frac{\pi}{2})\cos(x_2\frac{\pi}{2})\cdots\cos(x_{M-2}\frac{\pi}{2})\sin(x_{M-1}\frac{\pi}{2}), \\ Minimize\ \ f_3(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\cos(x_1\frac{\pi}{2})\cos(x_2\frac{\pi}{2})\cdots\sin(x_{M-2}\frac{\pi}{2}), \\ \vdots & & \\ Minimize\ \ f_{M-1}(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\cos(x_1\frac{\pi}{2})\sin(x_2\frac{\pi}{2}), \\ Minimize\ \ f_M(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\sin(x_1\frac{\pi}{2}), \\ & 0 \le x_i \le 1, & for\ \ i = 1,2,\ldots,n, \\ where\ \ g(\mathbf{x}_M) & = & 100\left[|\mathbf{x}_M| + \sum_{x_i\in\mathbf{x}_M}(x_i-0.5)^2 - \cos(20\pi(x_i-0.5))\right]. \end{array}\right\} \quad (4.3)$$

Figure 4.1 show the PO front for 3 objectives. The PO solution corresponds to $\mathbf{x}_M = \{0.5, 0.5, \ldots, 0.5\}^T$. The total number of variables in the above problem is $n = M + k - 1$. Where k can be set by the user. A value of $k = |\mathbf{x}_M| = 10$ is used. The g function in the problem introduces $(3^k - 1)$ local PO fronts and one global PO front. All local PO fronts are parallel to the global PO front and an MOEA can get stuck to any of these local PO fronts, before converging to the global PO front (at $g^* = 0$).

4.4. TEST PROBLEM DLTZ6

$$\left.\begin{array}{rcl}
Minimize\ \ f_1(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\cos(\theta_1\frac{\pi}{2})\cos(\theta_2\frac{\pi}{2})\cdots\cos(\theta_{M-2}\frac{\pi}{2})\cos(\theta_{M-1}\frac{\pi}{2}),\\
Minimize\ \ f_2(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\cos(\theta_1\frac{\pi}{2})\cos(\theta_2\frac{\pi}{2})\cdots\cos(\theta_{M-2}\frac{\pi}{2})\sin(\theta_{M-1}\frac{\pi}{2}),\\
Minimize\ \ f_3(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\cos(\theta_1\frac{\pi}{2})\cos(\theta_2\frac{\pi}{2})\cdots\sin(\theta_{M-2}\frac{\pi}{2}),\\
\vdots & & \\
Minimize\ \ f_{M-1}(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\cos(\theta_1\frac{\pi}{2})\sin(\theta_2\frac{\pi}{2}),\\
Minimize\ \ f_M(\mathbf{x}) & = & (1+g(\mathbf{x}_M))\sin(\theta_1\frac{\pi}{2}),\\
where\ \ \theta_i & = & \frac{\pi}{4(1+g(r))}(1+2g(r)x_i),\quad for\ \ i=2,3,\ldots,(M-1),\\
and\ \ g(\mathbf{x}_M) & = & \sum_{x_i\in\mathbf{x}_M} x_i^{0.1},\\
 & 0\le x_i\le 1, & for\ \ i=1,2,\ldots,n.
\end{array}\right\}\quad(4.4)$$

This problem will test an MOEA's ability to converge to a curve. Figure 4.1 show the PO front for 3 objectives. In this case there is only one independent variable (x_1) describing the PO front. The mapping between θ_i and x_i in equation (4.4) ensures that the curve is the only non-dominated region in the entire search space. Since $g = 0$ corresponds to PO front, $\theta_i = \pi/4$ for all but the first variable. As before, the total number of variables in the above problem is $n = M + k - 1$. Where k can be set by the user. A value of $k = |\mathbf{x}_M| = 10$ is used.

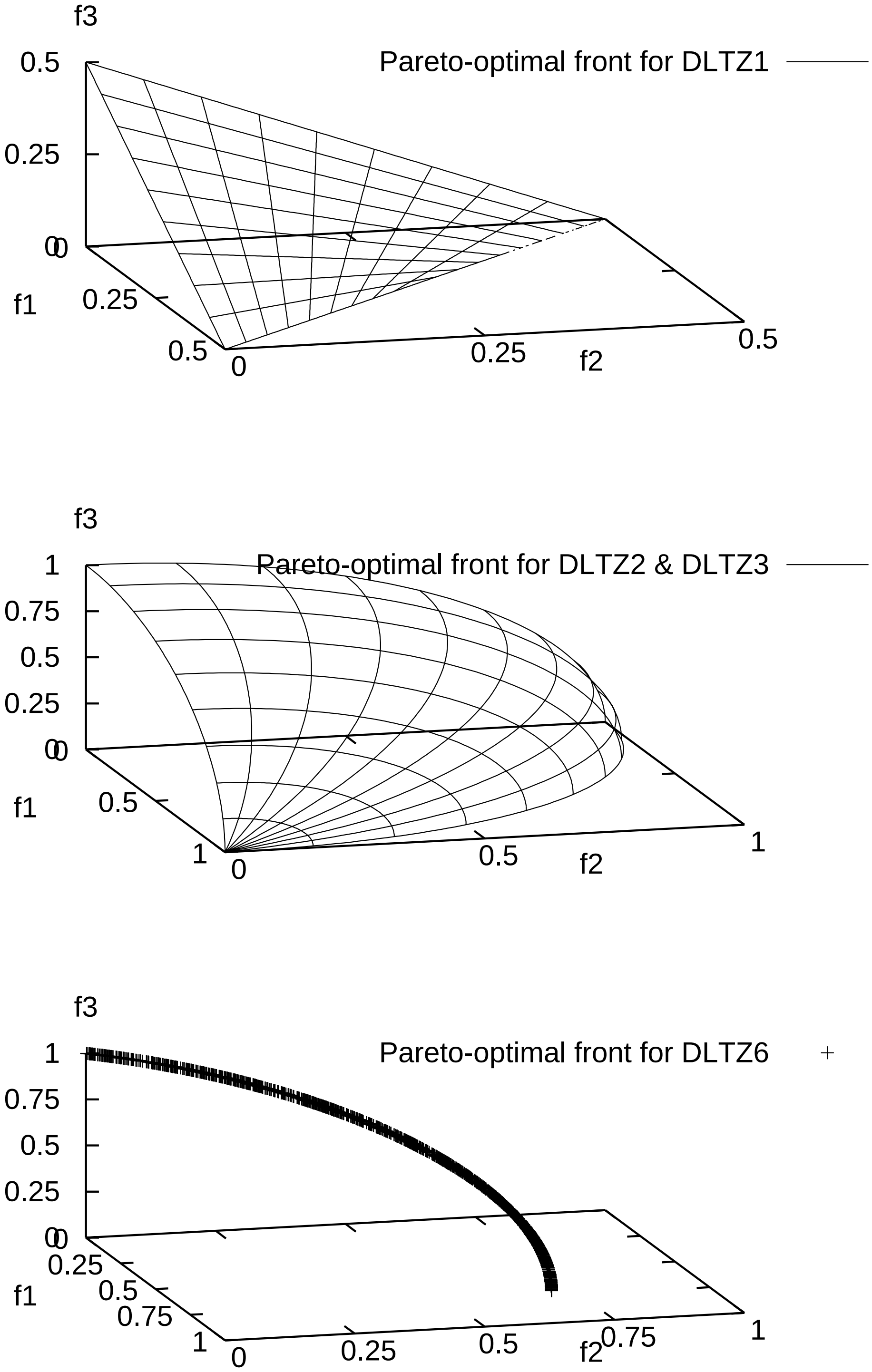


Figure 4.1: Pareto-optimal fronts for all 4 Test Problems with 3 Objectives

Chapter 5

Performance Metrics

Earlier MOEAs paid emphasis on getting more and more close to the true PO front in the objective space. Comparing the performance of different MOEAs is complicated by the fact that the result of an EMO run is not a single scalar value but a vector of objective values. This in turn requires more than one performance metric. Also as we know that there are two distinct goals (figure 5.1) in multi-objective optimization, (i) discover solutions as close to the PO solutions as possible (which requires search *towards* the PO front). (ii) find solutions as diverse as possible in the obtained non-dominated front (which requires search *along* the PO front). In some sense, these two goals are *orthogonal* to each other. Hence we require at least two metrics for these two goals.

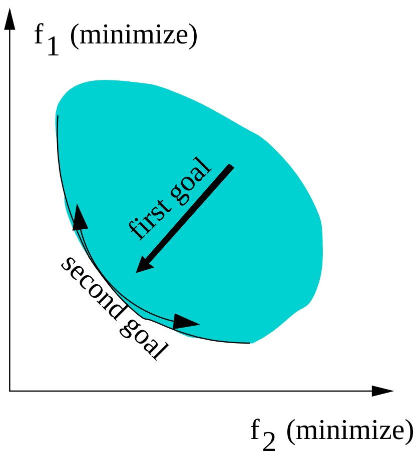


Figure 5.1: Two goals of multi-objective optimization

It has been widely accepted that one performance metric is not enough to judge the performance of an MOEA. A number of different performance metrics have been suggested in the literature. Hansen and Jaszkiewicz [HJ98] described three performance metrics namely - R1, R2 and R3. These metrics compare two non-dominated sets on the basis of some utility functions and determine the expected number of occasions the solutions of one set is better than the other. Detailed descriptions of these metrics can be found in [HJ98]. Knowles and Corne [KC02] analyzed different metrics available in literature on the basis of extent of outperformance relations between the two sets of non-dominated solutions, which are to be compared, and recommended the use of any of the above three metrics. However these metrics require a number of utility functions, their probabilities of occurrence and a numerical integration technique to evaluate them.

Zitzler [Zit99], in his $\mathcal{S}$ metric, calculates the hypervolume of the multi-dimensional region enclosed by the non-dominated set to be assessed and a 'reference point' to measure the diversity and convergence of the non-dominated set obtained. This hypervolume value represents the size of region the non-dominated set dominates. The relative value of $\mathcal{S}$ metric depends on the chosen reference point, hence same two non-dominated sets can have different relative $\mathcal{S}$ metric for different reference points.

Zitzler [ZLT+02], later showed that for an $M-$objective optimization problem, at least M performance metrics are needed to compare two or more sets of solutions. Use of any number less than M would result in inaccurate judgement because of dimensionality reduction. Deb [DJ02] suggested that we can overcome this dimensionality problem using a *functionally independent* set of variables, which would of course make it theoretically inaccurate but practically feasible. He also suggested two new *running* performance metrics - one for measuring the convergence to the reference set and other for measuring the diversity in population members at every generation of an MOEA run. In this study these two metrics (with slight variation) have been used in addition to a third one, which simply measures the running time of the MOEA. All three metrics, which were applied to only the final non-dominated set obtained by an MOEA to evaluate its performance, have been discussed in detail in the following sections.

5.1. METRIC FOR CONVERGENCE

The following metric, in some sense, represent the distance between the set of converged non-dominated solutions and the global PO front, hence lower values of convergence metric represent good convergence ability. Let P^* be the reference or target set of points on the Pareto-optimal front and let $\mathcal{F}$ be the final non-dominated set obtained by an MOEA. Then from each point i in $\mathcal{F}$ the smallest normalized Euclidean distance to P^* will be :

$$d_i = \min_{j=1}^{|P^*|} \sqrt{\sum_{k=1}^{M} \left(\frac{f_k(i) - f_k(j)}{f_k^{max} - f_k^{min}} \right)^2} \quad (5.1)$$

Here, f_k^{max} and f_k^{min} are the maximum and minimum function values of k-th objective function in P^*. However in this study, no target points were chosen because equations of PO fronts were known for all the four test problems. The orthogonal distance of a point A in the non-dominated set from the PO front was calculated directly from the equation of PO front. E.g. in DLTZ2 or DLTZ3

$$d_i = \|\vec{r}_A\| - 1 \quad (5.2)$$

Once these distances are known the convergence metric can be obtained by averaging the normalized distance for all points in $\mathcal{F}$:

$$C = \frac{\sum_{i=1}^{|\mathcal{F}|} d_i}{|\mathcal{F}|} \quad (5.3)$$

5.2. METRIC FOR DIVERSITY

Diversity metric is a number in the range $[0, 1]$, where 1 corresponds to best possible diversity and a 0 corresponds to worst possible diversity. In calculating the diversity metric, the obtained non-dominated points are projected on a hyper-plane, thereby losing a dimension of the points. The plane is divided into a number of small grids (or $M-1$ dimensional boxes). Depending on each grid contains a non-dominated point or not, a diversity metric is defined. If all grids are represented with at least one point, the best possible (with respect to the chosen number of grids) diversity measure is achieved. If some grids are not represented by a non-dominated point, the diversity is poor. Various parameters required to calculate this metric and their chosen values are given in table 5.1

Reference plane	$M-$th objective function $f_M = 0$
Number of grids (G_i)	Population size
Target (or reference) set of points (P^*)	One assumed solution in each grid

Table 5.1: Parameter settings for calculating the diversity metric

Calculating the diversity metric Following steps are involved in calculating the diversity metric:

1. For each grid indexed by $(i, j, \ldots)$ calculate following two arrays:

$$H(i,j,\ldots) = \begin{cases} 1, & \text{if the grid has a representative point in} P^*; \\ 0, & \text{otherwise.} \end{cases}$$
$$= \quad 1, \quad \text{for the chosen reference set} P^* \tag{5.4}$$
$$h(i,j,\ldots) = \begin{cases} 1, & \text{if H(i, j, ...) = 1 and the grid has a representative point in} \mathcal{F}; \\ 0, & \text{otherwise.} \end{cases}$$
$$= \begin{cases} 1, & \text{if the grid has a representative point in } \mathcal{F} \text{ for the chosen} P^*; \\ 0, & \text{otherwise.} \end{cases} \tag{5.5}$$

2. Assign a value $m(H(i, j, \ldots))$ to each grid depending on its and its neighbor's $h()$. Similarly, calculate $m(H(i, j, \ldots))$ using $H()$ for reference points.

3. Calculate the diversity metric (of the population P of non-dominated slution produced by an MOEA) by averaging the individual $m()$ values for $h()$ with respect to that for $H()$:

$$D(P) = \frac{\sum_{\substack{i,j,\ldots \\ H(i,j,\ldots)\neq 0}} m(h(i,j,\ldots))}{\sum_{\substack{i,j,\ldots \\ H(i,j,\ldots)\neq 0}} m(H(i,j,\ldots))} \tag{5.6}$$

Value function $m()$ for the grid was calculated by using its $h()$ and two neighboring $h()$ dimension-wise. With a set of three consecutive binary $h()$ values, there are a total of 8 possibilities. Any value function may be assigned by keeping in mind the following:

- A 111 is the best distribution and a 000 is the worst.
- A 010 or a 101 means a periodic pattern with a good spread and should be valued more than 110 or 011
- A 110 or 011 may be valued more than 001 or 100, because of more covered grids.

Based on the above observations, [DJ02] suggested the following (Table 5.2) $m()$ and $h()$ values. We have also used the same lookup table for calculating these parameters. Two or more dimensional hyper-planes are handled by calculating the above metric dimension-wise.

h(. . . j-1 . . .)	**h(. . . j . . .)**	**h(. . . j+1 . . .)**	**m(h(. . . j . . .))**
0	0	0	0.00
0	0	1	0.50
1	0	0	0.50
0	1	1	0.67
1	1	0	0.67
0	1	0	0.75
1	1	1	1.00

Table 5.2: Lookup table for calculating diversity metric

Figure 5.2 shows a sample calculation of diversity metric in case of 2-objective DLTZ2 or 2-objective DLTZ3 problem. Here circles represent the reference or target points, in every partition there is one such point and hence the number of partions is same as population size. Boxes represent the set of non-dominated points given by an MOEA. The $f_2 = 0$ is used as the reference plane here and the complete range on f_1 values are divided into $G_1 = 10$ grids. This complete range depends on the PO front the algorithm has converged to and the resulting divesrsity metric (see sections 5.2.1 and 5.2.2) will also be different. For boundary grids, an imaginary neighboring grid with a $h()$ and $H()$ value of one is always assumed. In figure 5.2 (a), these grids are shown with dashed lines. Even if we have more than one points in a grid, $h()$ still remains as one. Based on moving window containing three consecutive grids, the $m()$ values are computed in the figure. To avoid the boundary effects of using the imaginary grid, the metric value is normalized as follows:

$$\overline{D}(P) = \frac{\sum_{\substack{i,j,\ldots \\ H(i,j,\ldots)\neq 0}} m(h(i,j,\ldots)) - \sum_{\substack{i,j,\ldots \\ H(i,j,\ldots)\neq 0}} m(\mathbf{0})}{\sum_{\substack{i,j,\ldots \\ H(i,j,\ldots)\neq 0}} m(H(i,j,\ldots)) - \sum_{\substack{i,j,\ldots \\ H(i,j,\ldots)\neq 0}} m(\mathbf{0})} \tag{5.7}$$

where **0** is a zero-valued array. Though in our study $H(i,j,\ldots)$ is always equal to 1 (each grid contains one reference point), $H(i,j,\ldots) \neq 0$ consideration in computing the $\overline{D}(P)$ term and the boundary grid adjustment was suggested to allow a generic way to handle disconnected PO fronts in the original work [DJ02].

5.2.1. Metric based on Pareto-optimal front (Diversity Metric1)

Ideally we want our algorithm to give us non-dominated solutions on the global PO front and if we divide our objective space corresponding to this global PO front into grids equal to the population size, then one point in each grid would be the best possible diversity ($\overline{D}(P) = 1$). We will call this diversity metric (obtained by splitting the global PO region into grids) *diversity metric1*.

5.2.2. Metric based on Converged front (Diversity Metric2)

But if the algorithm isn't able to converge to the global PO front then the above metric will not be able to measure the diversity of non-dominated solutions produced by the MOEA. E.g. in figure 5.2 (b), if solid black squares represent the non-dominated points produced by an MOEA on a local PO front (circle with radius equal to 2 for 2-objective DLTZ2 or DLTZ3), the diversity metric based on global PO front gives us a diversity value of one, but (as clear from the figure) the spread of solutions isn't very good and it might end up converging to a very small region of global PO front. In such cases where an algorithm is struck in a local PO front, we should calculate our diversity metric based on the actual converged front instead of global PO front. We will call this diversity metric (obtained by splitting the converged PO region into grids) *diversity metric2*.

To compare two algorithms (for diversity) the use of diversity metric2 should be preferred because even if one of them has converged to true PO front and the other hasn't, the diversity metric2 of the former will be almost equal to its diversity metric1 value, which would not be the case with latter.

5.3. RUNNING TIME

This is simply the running time of an algorithm (in seconds) for the particular settings. It has been included in the performance metric set to evaluate how an MOEA scales in terms of time complexity with increase in number of objectives. A linear or polynomial increase in running time is acceptable but an exponential increase is undesirable.

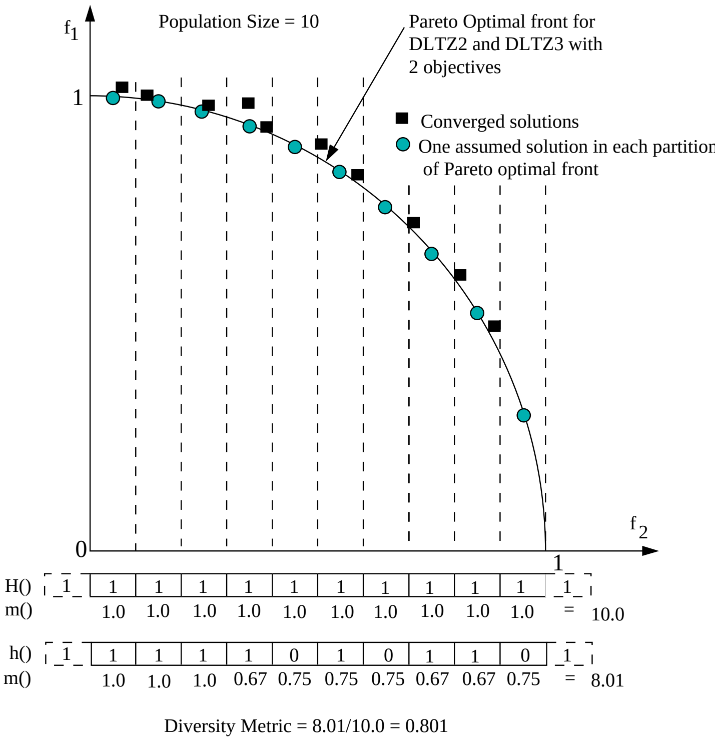
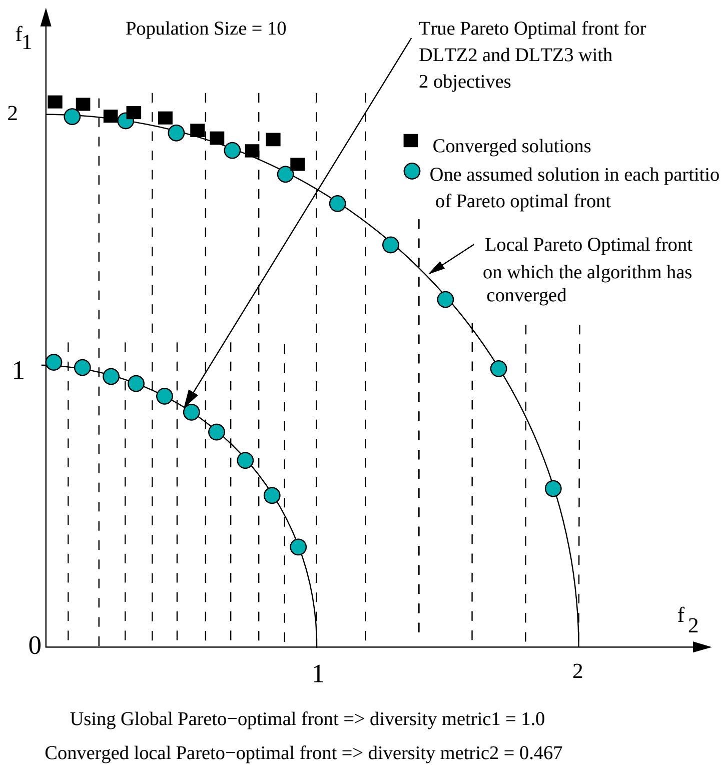


fig (a)



fig(b)

Figure 5.2: Fig (a) Calculating the diversity metric; Fig (b) Two different diversity metrices - obtained using true PO front and converged PO front

Chapter 6

Experimental Study

Extensive experimentation was done to compare the relative performance scaling of different algorithms. Experiments were conducted for the three algorithms (section 3.2) on the four test problems (sections 4.1 to section 4.4) for 2, 3, 4, 6 and 8 objectives. 30 runs were performed for the 2, 3, and 4 objective cases but because limited amount of time only 10 runs could be performed for higher number of objectives. That is, in total

3 algorithms *4 test problems * [30 runs* 3 different objectives + 10 runs *2 different objectives] = 1320

simulations runs were conducted for the study. (It is worth mentioning here that the running time for the slowest algorithm, PESA, increased sharply with the number of objectives, e.g. for 8-objective DLTZ2 problem a single run of PESA took on an average 93 hours for $0.72 * 10^6$ function evaluations). To make the comparisons fair the number of function evaluations (table 6.1) were kept constant for all three algorithms on a particular setting. E.g. for 8-objective DLTZ2 problem each algorithm was allowed $0.72 * 10^6$ function evaluations.

Algorithm	**Populations Involved**	**Function Evaluations**
PESA	Internal Population (IP) and External Population (EP)	($IP + EP$)* # of generations
SPEA2	Population Size (N) and Archive Size ($\overline{N}$)	($N + \overline{N}$) * # of generations
NSGA-II	Population Size(N)	2*N * # of generations

Table 6.1: Number of function evaluations

There were different parameters associated with the experimentation, some common to all the algorithms and some specific to a particular one. Before the actual experimentation some tuning of these parameters was required. Finding values for which an algorithm works best is, in itself, an optimization problem and if we are judging the performance on the three metrics (section 5.1 to section 5.3) it becomes a MOOP for each of the parameter involved. A very simplistic approach was adopted to tune (instead of optimizing) these parameters. Experiments with only 2-objectives were used and the purpose of this tuning was to find a set of values for which an MOEA performs well. Following is the description of these parameters and table 6.2 gives the tuned values for these parameters.

6.1. PARAMETER TUNING

Lets first look at the parameters that are common to all the algorithms. Other than population size, which is discussed separately in the next section because of its relative importance, the optimal values of different parameters for different algorithms found as the result of experiments with 2-objective are listed in table 6.2. These experiments were conducted on problems DLTZ2 and DLTZ3 and only the convergence and diversity metrics were used to evaluate the performance of the algorithm, running time was not considered.

Besides these, another parameter specific to PESA was the number of grids that each dimension of objective space is split into. Grid size $\mathcal{G} = 10$ was found to be optimal for 2-objective DLTZ2 and DLTZ3 problems.

Parameter	PESA	SPEA 2	NSGA-II
Crossover probability p_c	0.8	0.7	0.7
Distribution index for SBX η_c	15	15	15
Mutation probability p_m (if n = # of variables)	$\frac{1.0}{n}$	$\frac{1.0}{n}$	$\frac{1.0}{n}$
Distribution index for polynomial mutation η_m	15	15	20
Ratio of internal population size to archive size	1:1	1:1	1:1

Table 6.2: Tuned parameter values

6.2. POPULATION SIZE

Population size plays a crucial rule in the performance of an MOEA. As the number of objective functions (M) increases, more and more solutions tend to lie in the first non-dominated front. Most MOEAs (sections 3.1 and 3.2) assign the similar fitness to all solutions in the first non-dominated front. So as the number of objective functions increase, there is no (very little) selection advantage to any of these solutions. In absence of any selection pressure for better solutions, the task of recombination and mutation operators to find better solutions may be difficult in general.

It has been shown empirically [Deb01, pages 404–405] that for a particular M, the proportion of non-dominated solutions decreases with population size. If we require a population with a user-specified maximum proportion of non-dominated solutions, then these empirical results can be used to estimate what would be a reasonable population size. This requirement on population size increases exponentially with M. Ideally to investigate the scaling of an MOEA we should present it with a population having equal proportion of non-dominated solutions, for all M, to start with, but this is practically impossible because of exponential increase in population size. The population scheme used in this study is given in table 6.3.

M	Population Size	Maximum proportion of non-dominated solutions
2	20	0.2
3	50	0.22
4	100	0.28
5	150	0.36
6	250	0.45
7	400	0.52
8	600	0.60
9	850	0.68
10	1150	~ 0.75

Table 6.3: Population scheme - for Internal or main population (which is same as external population as the result of parameter tuning)

Ideally to maintain a constant value of maximum proportion of non-dominated solution we should have used an exponantial scheme in population size. But for practical reasons a polynomial scheme was chosen. The above scheme is quadratic with $R^2 = 0.9916$.

6.3. NUMBER OF GENERATIONS

As per the pervious discussions, each algorithm was allowed equal number of function evaluations for each simulation run. Population sizes and number of generations were kept constant across various algorithms. Population size was decided using the scheme presented in section 6.2. Number of generations used for different problems and different # of objectives (M) are listed in table 6.4.

More number of function evaluations were used for DLTZ3 and DLTZ6 because they can introduce more difficulties to a MOEA in converging to PO front and in finding a diverse set of solutions. DLTZ3 tests the ability

# of generations		
	For $M = 2, 3$ and 4	**For $M = 6$ and 8**
DLTZ1 & DLTZ2	300	600
DLTZ3 & DLTZ6	500	1000

Table 6.4: # of generations for different problems and # of objectives (M)

of an MOEA by introducing local PO fronts (section 2.5.2) and DLTZ6 tests them for their ability to converge to a curve (section 4.4.

Number of generations from 6-objectives onwards was doubled because none of the algorithms was able to converge to the global PO front in these many generations. Converging to a PO front means having a convergence metric (section 5.1) less than a threshold (ϵ). Any appropriate value of ϵ can be chosen. But here, instead of choosing some such threshold, the algorithms were compared (for convergence) solely on the basis of the convergence metric that they can achieve in given number of function evaluations.

6.4. RESULTS

All the experiments were conducted using these tuned parameters, though the parameters were tuned only for 2 objectives the same tuned parameter values were used for higher number of objectives. Following sections present the results of experimentation. To get a better understanding, the results presented here are split up according to performance on convergence to front, diversity of solutions obtained and running time. This classification will help us in analysing the performance of different algorithms on the basis of different performance metrics.

Results presented here are in concise form and only give the mean values of all the runs. For a detailed description of results, readers are referred to Appendix A. In Appendix A results for 2, 3 and 4 objectives are presented with the mean (MEAN), standard deviation (SD), minimum (MIN) and maximum (MAX) values obtained in 30 runs. Since only 10 runs could be performed for higher objectives all 10 runs are listed for 6 and 8 objectives.

6.4.1. Results for Convergence

A lower convergence metric value implies better convergence. Following sections present results for convergence on the four test problems. PESA gives the best performance in terms of convergence for all four problems. SPEA2 and NSGA-II have similar performances.

6.4.1.1. Results for Problem DLTZ1

Table 6.5 lists the convergence metric obtained for DTLZ1. Convergence metric results on DLTZ1 are also plotted in figure 6.1. Few observations can be made from these results:

- All three algorithms have similar performance for 2 to 4 objectives but as the number of objectives increase, PESA performs much better than the other two.
- Except for 6-objective case, convergence levels of SPEA2 and NSGA-II are more or less the same.

6.4.1.2. Results for Problem DLTZ2

Table 6.6 lists the convergence metric obtained for DTLZ2. Convergence metric results on DLTZ2 are also plotted in figure 6.3. Few observations can be made from these results:

- PESA performs much better than the other two.
- Convergence levels of SPEA2 and NSGA-II are more or less same.
- We should expect convergence metric to increase with the # of objectives but between 4 and 6 objective it has decreased, the reason being, the increased number of generations in 6 and 8 objectives.

Convergence Metric (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	300	2.86948	3.08825	2.27666
3	50	300	0.04419	0.04843	0.38360
4	100	300	0.02317	0.29925	3.10281
Convergence Metric (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	600	0.00117	5.99951	120.19162
8	600	600	0.00407	498.27151	465.30155

Table 6.5: Convergence Metric for DLTZ1 (see table A.1 for detailed results)

Convergence Metric (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	300	8.40378E-05	0.00026	0.00179
3	50	300	0.00035	0.006635	0.01003
4	100	300	3.60429	5.07137	6.32618
Convergence Metric (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	600	0.00301	2.00216	1.67564
8	600	600	0.00689	2.35258	2.30766

Table 6.6: Convergence Metric for DLTZ2 (see table A.2 for detailed results)

6.4.1.3. Results for Problem DLTZ3

Table 6.7 lists the convergence metric obtained for DTLZ3. Convergence metric results on DLTZ3 are also plotted in figure 6.5. Few observations can be made from these results:

- PESA performs much better than the other two.
- NSGA-II performs badly at the beginning for smaller number of objectives, but catches up with SPEA2.
- Both SPEA2 and NSGA-II have bad convergence levels for 6 and 8-objective cases, showing that they got stuck to one of the $(3^k - 1)$ local PO fronts (section 4.3).

Convergence Metric (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	500	22.52023	16.87313	21.32032
3	50	500	1.80296	2.39886	5.65577
4	100	500	1.16736	4.00596	66.94049
Convergence Metric (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	1000	0.15035	217.95360	1273.30601
8	600	1000	7.23062	1929.94832	1753.41364

Table 6.7: Convergence Metric for DLTZ3 (see table A.3 for detailed results)

6.4.1.4. Results for Problem DLTZ6

Table 6.8 lists the convergence metric obtained for DTLZ6. Convergence metric results on DLTZ3 are also plotted in figure 6.5. Here again SPEA2 and NSGA-II perform neck to neck but PESA once again appears as the overall winner except for the 2-objective case where it performs marginally worse than NSGA-II.

Convergence Metric (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	500	0.79397	0.77622	0.63697
3	50	500	0.20528	0.29271	0.24515
4	100	500	3.60430	5.07137	6.32619
Convergence Metric (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	1000	5.30454	10.53682	9.48750
8	600	1000	6.32247	10.62932	10.27306

Table 6.8: Convergence Metric for DLTZ6 (see table A.4 for detailed results)

6.4.2. Results for Diversity

As discussed in sections 5.2.1 and 5.2.2, two metrics were used to measure the diversity of solutions obtained in the non-dominated front. *Diversity metric1* measures the diversity of solutions on the basis of true PO front and *diversity metric2* uses the converged front for this purpose. If the algorithm hasn't converged to true PO front then the use of diversity metric2 is suggested. I have calculated both of these metrics for all the simulation runs. As anticipated (section 5.2.2) the two diversity metrics differ in value if the converged front is far from the true PO front and diversity metric2 is better performance indicator for reasons stated in the same section. Following sections discuss performance of different MOEAs on the basis of both of these metrics.

6.4.2.1. Results for Problem DLTZ1

Tables 6.9 and 6.10 list diversity metric1 and diversity metric2 respectively. Diversity metric (both 1 and 2) results on DLTZ1 are also plotted in figure 6.2. Few observations can be made form these results.

- SPEA2 performs better than the other two.
- NSGA-II is better than PESA in terms of diversity metric2 and for smaller # of objectives (2, 3, 4) in diversity metric1.

Diversity Metric1 (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	300	0.25093	0.55656	0.447843
3	50	300	0.42116	0.63186	0.57752
4	100	300	0.37605	0.54176	0.38676
Diversity Metric1 (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	600	0.33643	0.35645	0.19343
8	600	600	0.25245	0.26107	0.19744

Table 6.9: Diversity Metric1 for DLTZ1 (see table A.5 for detailed results)

Diversity Metric2 (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	300	0.50019	0.86516	0.72559
3	50	300	0.52274	0.78292	0.76968
4	100	300	0.48239	0.67835	0.72558
Diversity Metric2 (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	600	0.39297	0.48162	0.38293
8	600	600	0.29631	0.35056	0.37425

Table 6.10: Diversity Metric2 for DLTZ1 (see table A.6 for detailed results)

6.4.2.2. Results for Problem DLTZ2

Tables 6.11 and 6.12 list diversity metric1 and diversity metric2 respectively. Diversity metric (both 1 and 2) results on DLTZ2 are also plotted in figure 6.4. Few observations can be made form these results.

- NSGA-II outperforms the other two algorithms in terms of diversity metric2.
- Performance in terms of diversity across true PO front (diversity metric1) is more or less the same.

Diversity Metric1 (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	300	0.57396	0.81866	0.75177
3	50	300	0.57163	0.67260	0.74996
4	100	300	0.22557	0.22916	0.12825
Diversity Metric1 (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	600	0.47099	0.29675	0.48248
8	600	600	0.43230	0.30944	0.52913

Table 6.11: Diversity Metric1 for DLTZ2 (see table A.7 for detailed results)

Diversity Metric2 (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	300	0.58508	0.82185	0.75863
3	50	300	0.57993	0.71680	0.81106
4	100	300	0.48906	0.47385	0.59749
Diversity Metric2 (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	600	0.52335	0.64928	0.73253
8	600	600	0.57082	0.64134	0.75665

Table 6.12: Diversity Metric2 for DLTZ2 (see table A.8 for detailed results)

6.4.2.3. Results for Problem DLTZ3

Tables 6.13 and 6.14 list diversity metric1 and diversity metric2 respectively. Diversity metric (both 1 and 2) results on DLTZ3 are also plotted in figure 6.6. Few observations can be made form these results.

- SPEA2 and NSGA-II have somewhat similar performances in terms of diversity metric2.
- Performance of PESA is much worse than other two on diversity metric2, especially for higher objectives.

Diversity Metric1 (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	500	0.14023	0.17338	0.08845
3	50	500	0.38964	0.62793	0.26243
4	100	500	0.31659	0.58861	0.15869
Diversity Metric1 (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	1000	0.18812	0.08588	0.12068
8	600	1000	0.02463	0.15186	0.06456

Table 6.13: Diversity Metric1 for DLTZ3 (see table A.9 for detailed results)

Diversity Metric2 (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	500	0.49708	0.71668	0.57169
3	50	500	0.58655	0.78539	0.60255
4	100	500	0.51138	0.72006	0.50373
Diversity Metric2 (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	1000	0.32959	0.36687	0.57644
8	600	1000	0.11972	0.55771	0.58456

Table 6.14: Diversity Metric2 for DLTZ3 (see table A.10 for detailed results)

6.4.2.4. Results for Problem DLTZ6

Tables 6.15 and 6.16 list diversity metric1 and diversity metric2 respectively. Diversity metric (both 1 and 2) results on DLTZ6 are also plotted in figure 6.8. Few observations can be made form these results.

- SPEA2 and NSGA-II have somewhat similar performance in terms of diversity metric1, but NSGA-II has a marginal better performance on diversity metric2.
- PESA once again is the loser in terms of diversity metric2.

6.4.3. Results for Running Time

Tables 6.17 to 6.20 list the running time of different algorithms on different problems with different # of objectives. Running time results are also plotted in figures 6.1 for DLTZ1, 6.3 for DLTZ2, 6.5 for DLTZ3 and 6.7 for DLTZ6. All the simulations were performed on Solaris workstations. 10 different machines were used for 10 runs of a particular algorithm to save time. Results for all four problems are pretty much similar. NSGA-II was the fastest and PESA was the slowest of all three for all the # of objectives over all problemss. PESA on 8-objective DLTZ2 took on an average approximately 93 hours to run. Just for the sake of trying, I ran PESA on 10-objective DLTZ2 and at the time of writing the report, after 9 days of running the code, it hasn't produced the output.

Diversity Metric1 (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	500	0.20191	0.44403	0.40845
3	50	500	0.41961	0.64654	0.66156
4	100	500	0.22557	0.22916	0.12825
Diversity Metric1 (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	1000	0.27631	0.07593	0.07544
8	600	1000	0.27801	0.05583	0.04766

Table 6.15: Diversity Metric1 for DLTZ6 (see table A.11 for detailed results)

Diversity Metric2 (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	500	0.55794	0.85028	0.78385
3	50	500	0.56628	0.83747	0.81710
4	100	500	0.48906	0.47385	0.59749
Diversity Metric2 (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	1000	0.48725	0.60226	0.65365
8	600	1000	0.44885	0.55396	0.62897

Table 6.16: Diversity Metric2 for DLTZ6 (see table A.12 for detailed results)

Running Time (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	300	2.53	3.7	2.9
3	50	300	9.125	17.375	4.53
4	100	300	70.525	91.45	17
Running Time (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	600	4229.2	1839	213.9
8	600	600	95465.67	34735.33	1552.72

Table 6.17: Running Time for DLTZ1 (see table A.13 for detailed results)

Running Time (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	300	4.6	4.067	3.267
3	50	300	19.825	20.425	4.925
4	100	300	193.975	107.45	18.7
Running Time (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	600	8082.8	2408	235.9
8	600	600	334641.5	33722	1570.8

Table 6.18: Running Time for DLTZ2 (see table A.14 for detailed results)

Running Time (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	500	5.567	6.633	5.733
3	50	500	11.833	29.333	7.972
4	100	500	80.8	131.925	29.75
Running Time (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	1000	3402.4	3103.1	384.4
8	600	1000	85559.4	49382.2	2605.5

Table 6.19: Running Time for DLTZ3 (see table A.15 for detailed results)

Running Time (Averaged over 30 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
2	20	500	3.767	4.233	3.2
3	50	500	22.233	32.466	7.733
4	100	500	295.15	178.175	31
Running Time (Averaged over 10 runs)					
# of Objectives (M)	**Population Size**	**# of Generations**	**PESA**	**SPEA2**	**NSGA-II**
6	250	1000	13528.2	5640.1	386.9
8	600	1000	262802.9	53809.8	2741.1

Table 6.20: Running Time for DLTZ6 (see table A.16 for detailed results)

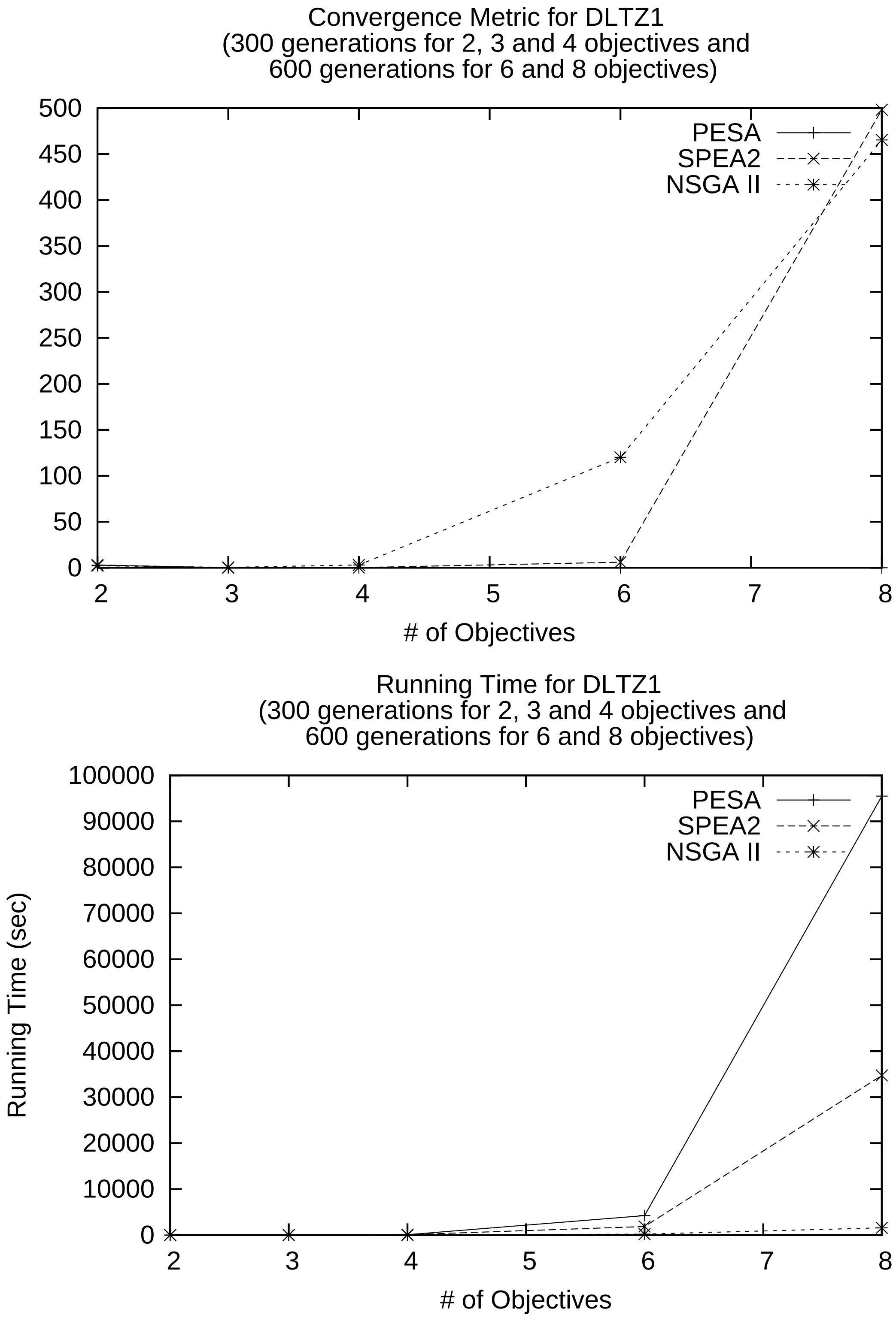
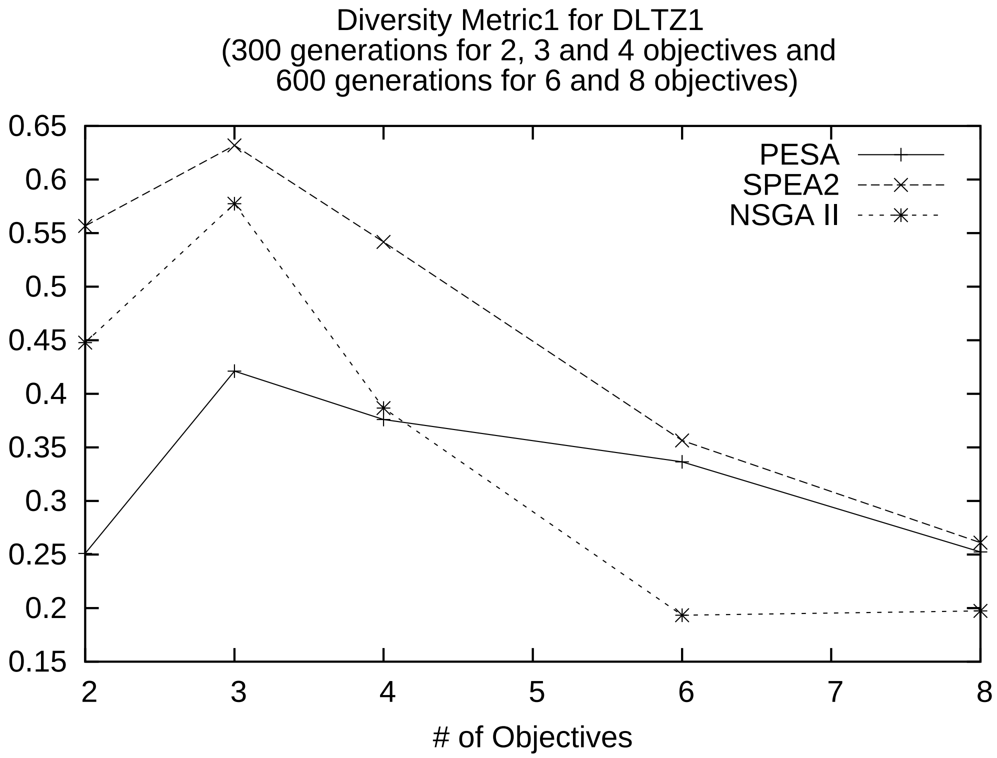


Figure 6.1: Convergence Metric & Running Time for DLTZ1



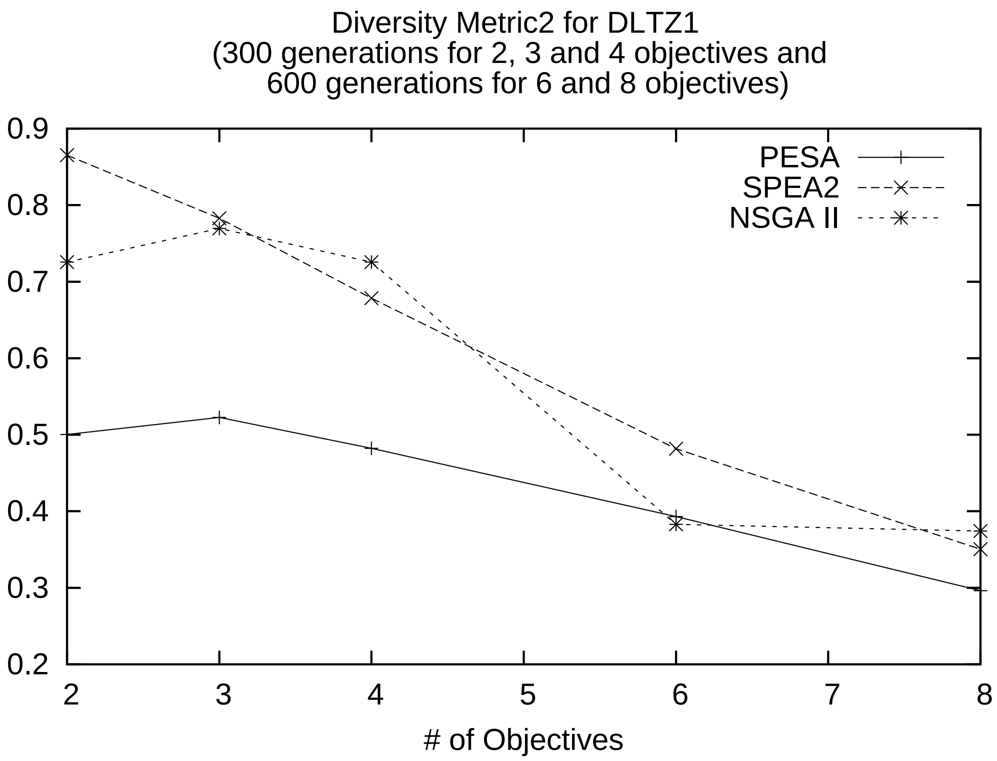
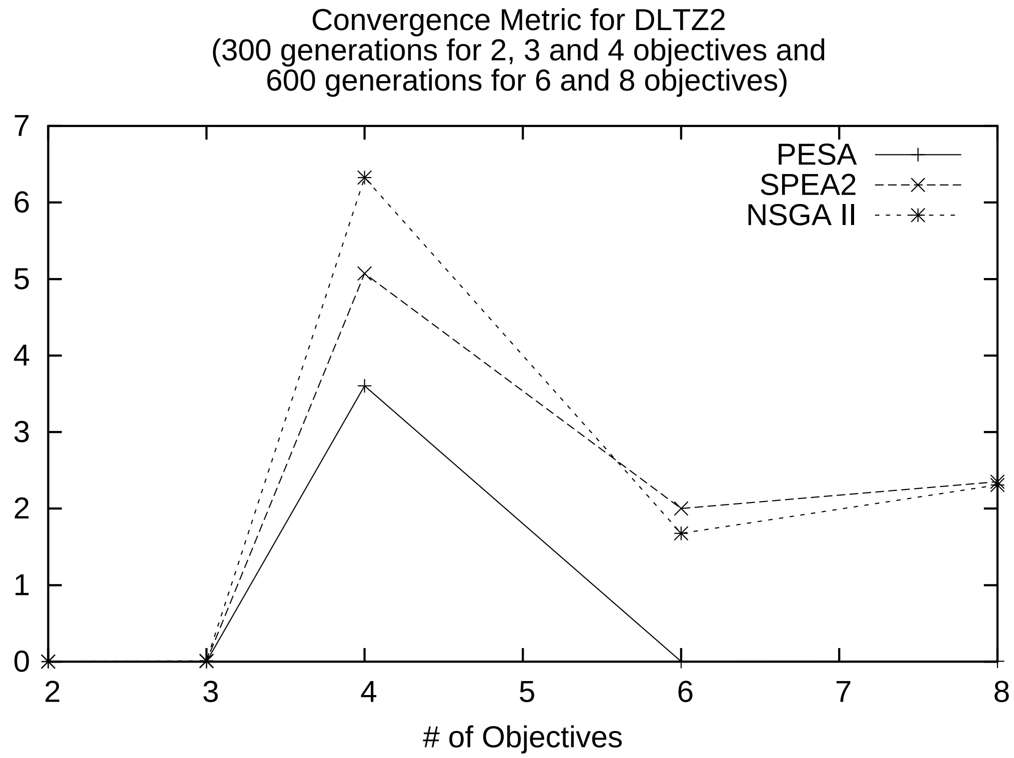


Figure 6.2: Diversity Metric (1 & 2) for DLTZ1



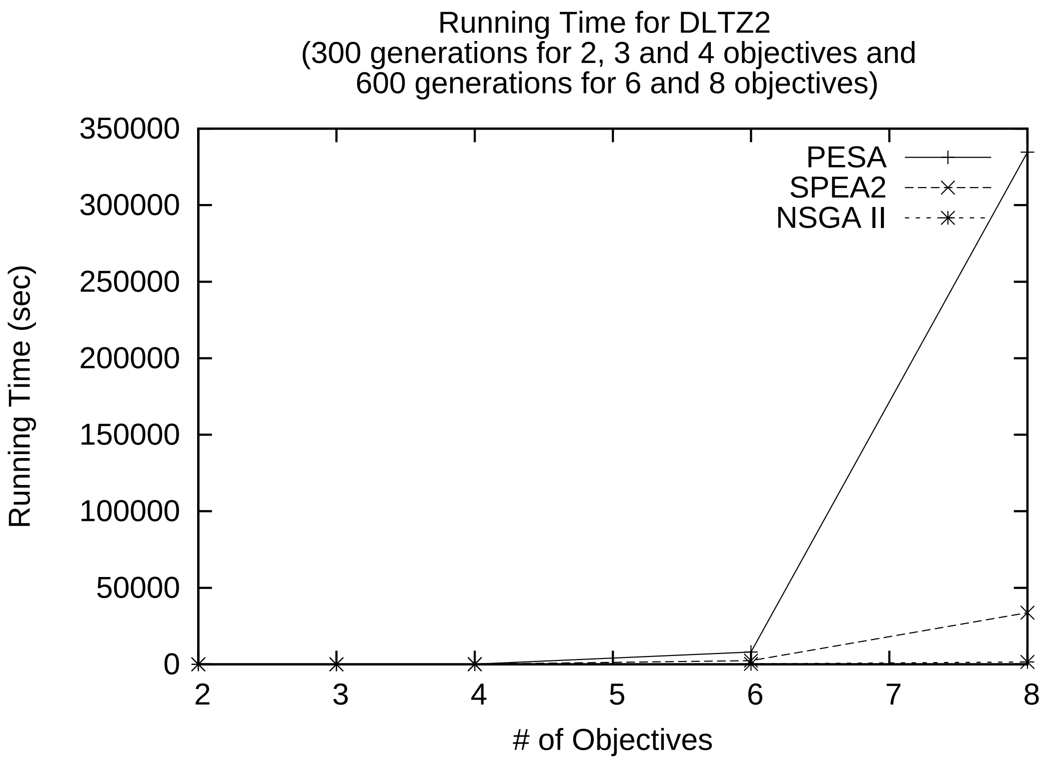


Figure 6.3: Convergence Metric & Running Time for DLTZ2

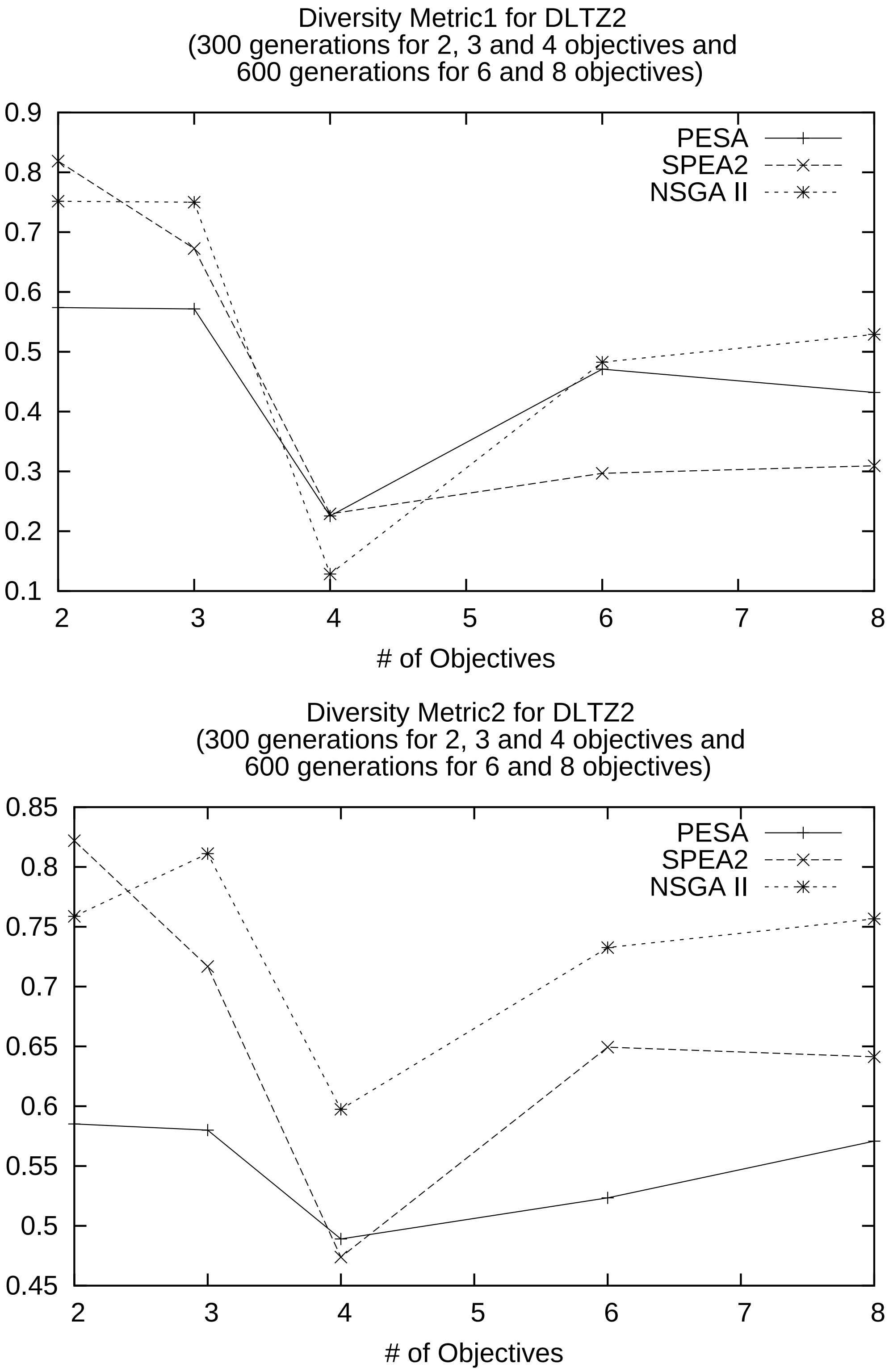
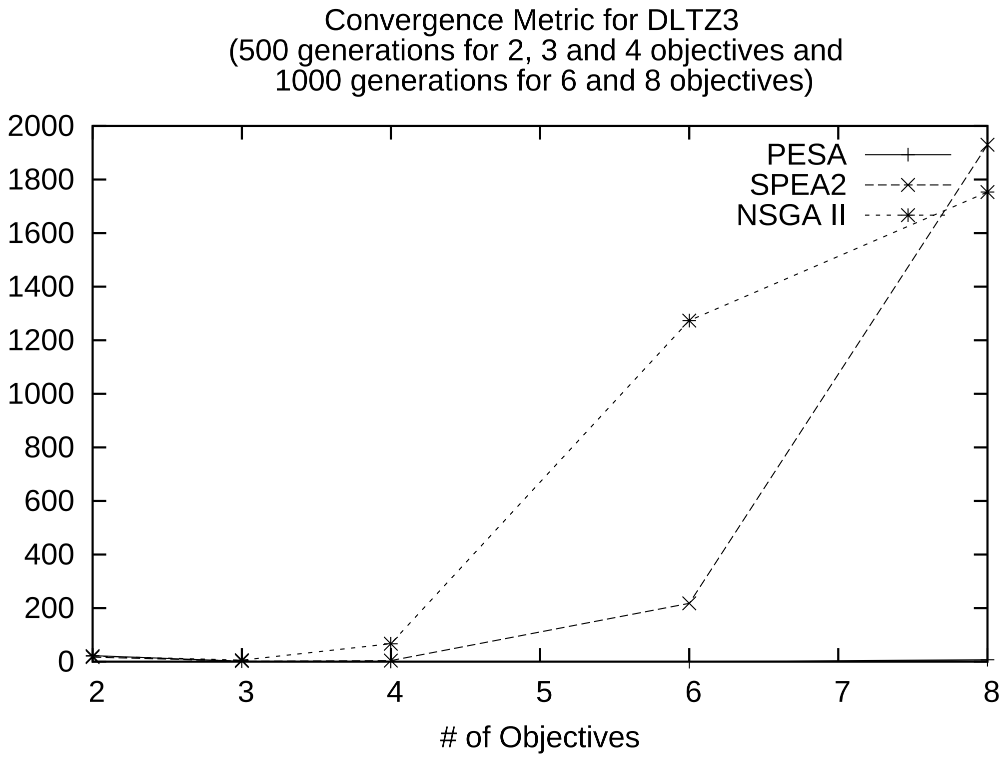


Figure 6.4: Diversity Metric (1 & 2) for DLTZ2



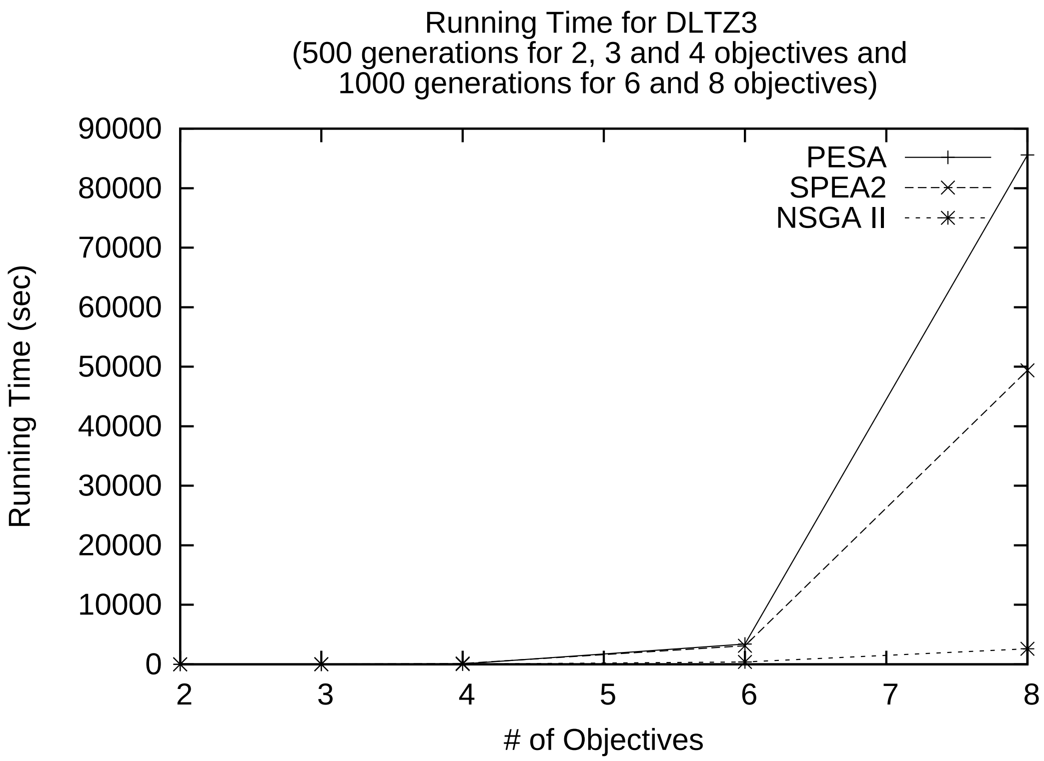


Figure 6.5: Convergence Metric & Running Time for DLTZ3

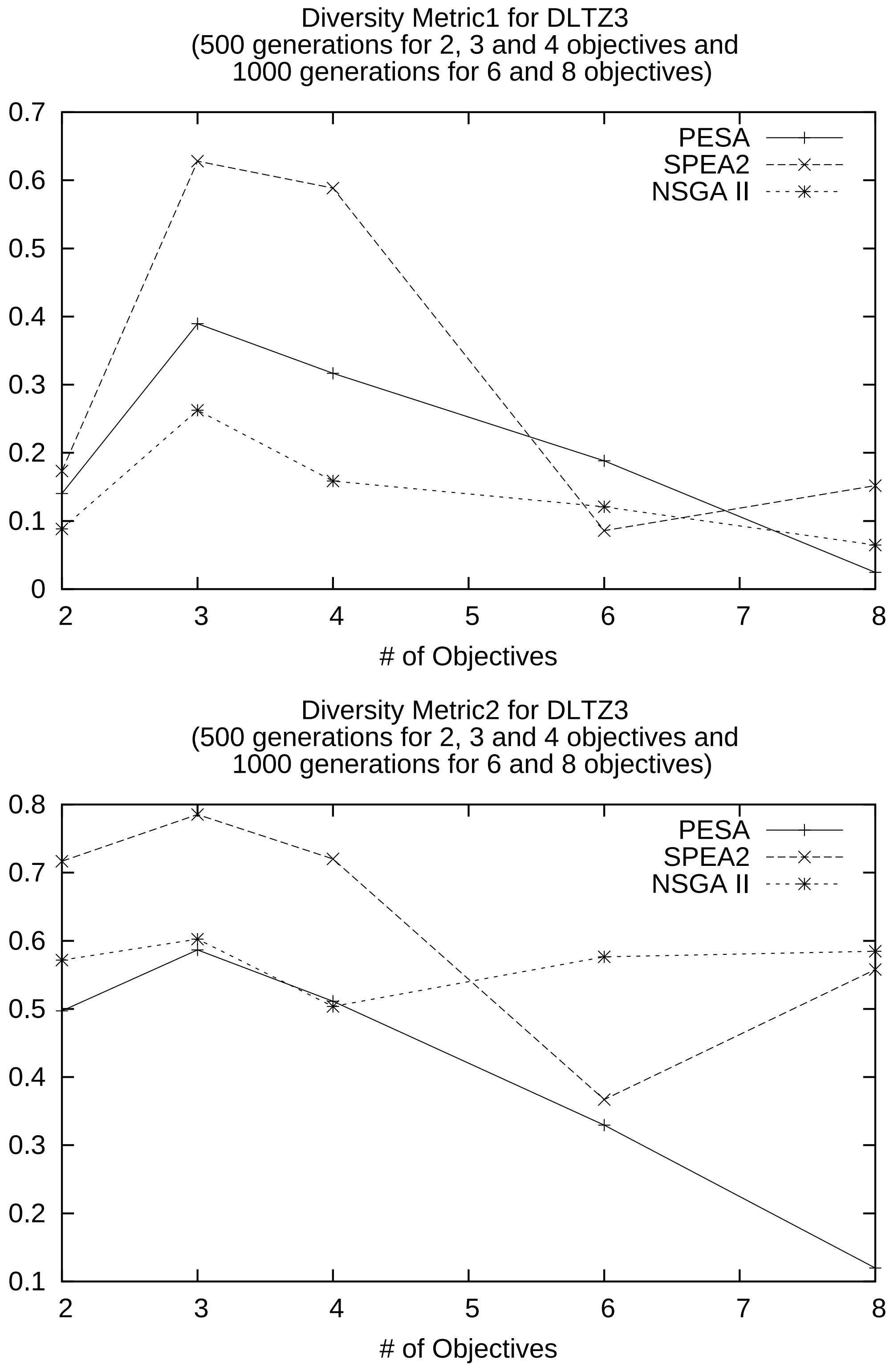
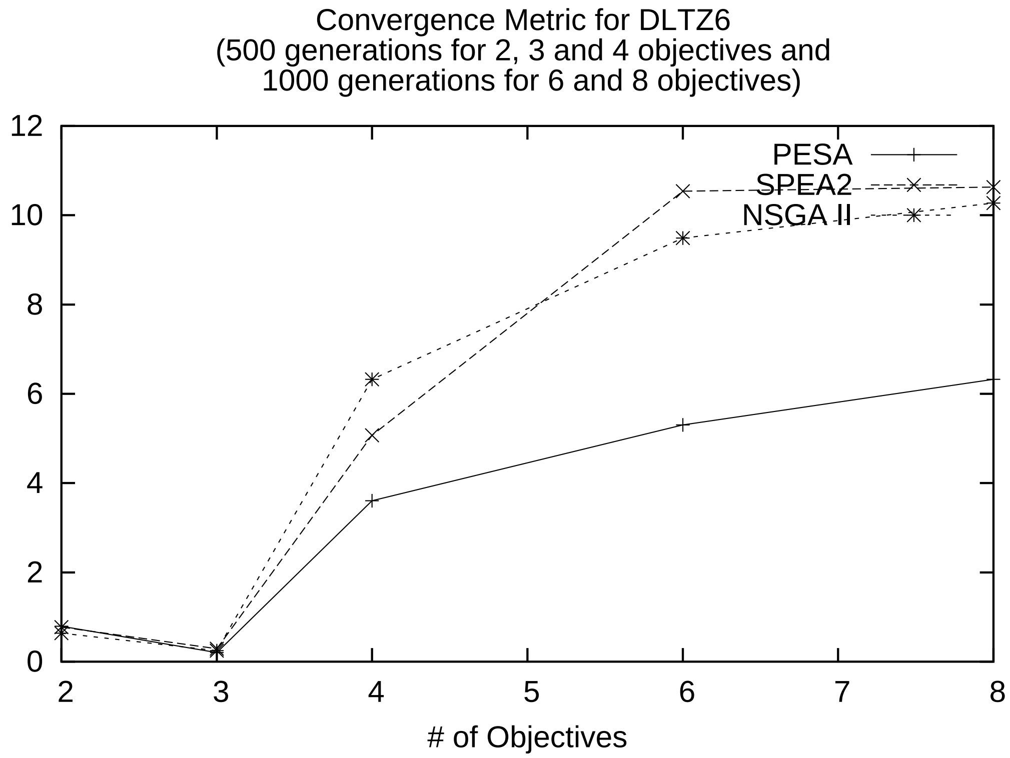


Figure 6.6: Diversity Metric (1 & 2) for DLTZ3



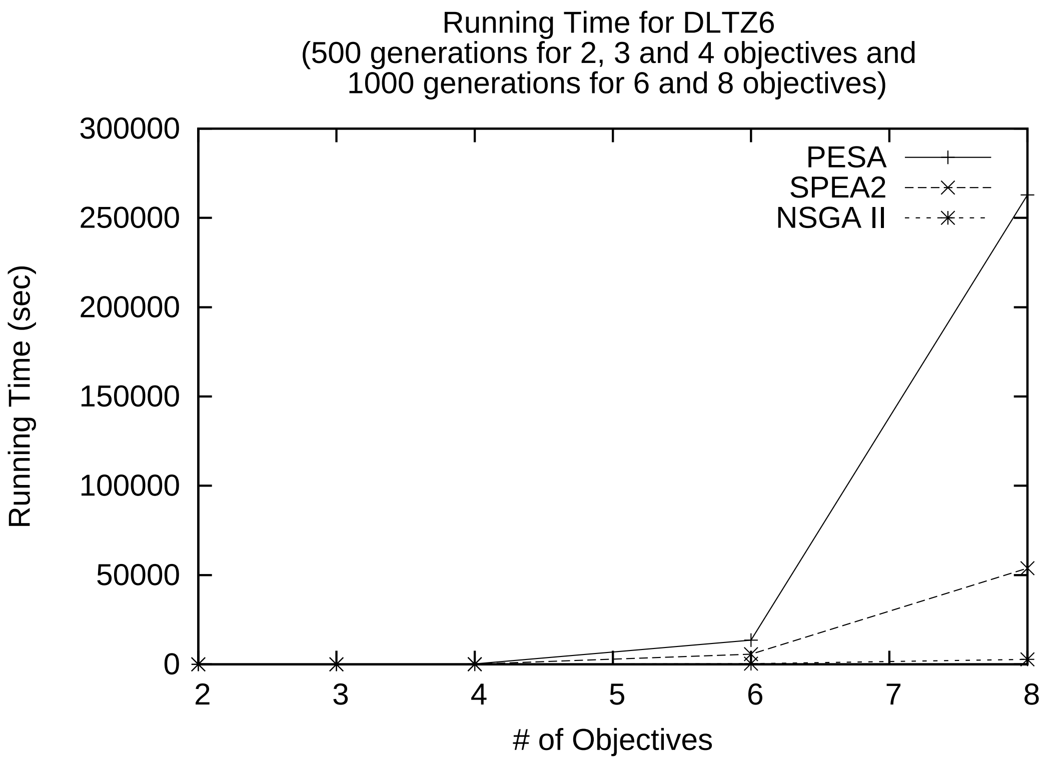
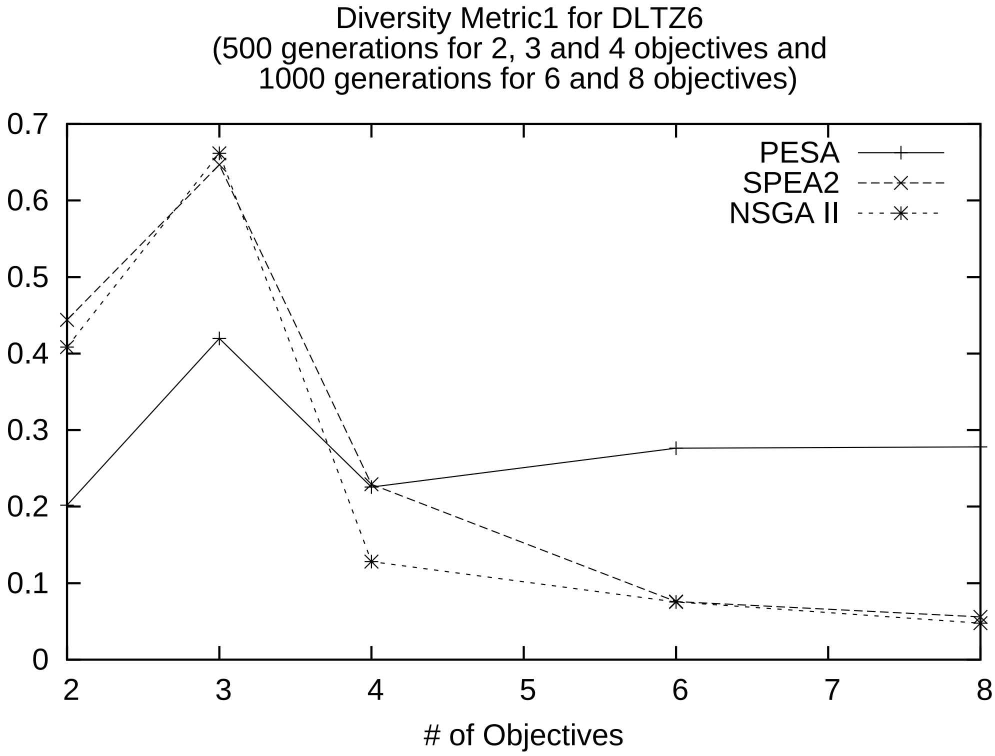


Figure 6.7: Convergence Metric & Running Time for DLTZ6



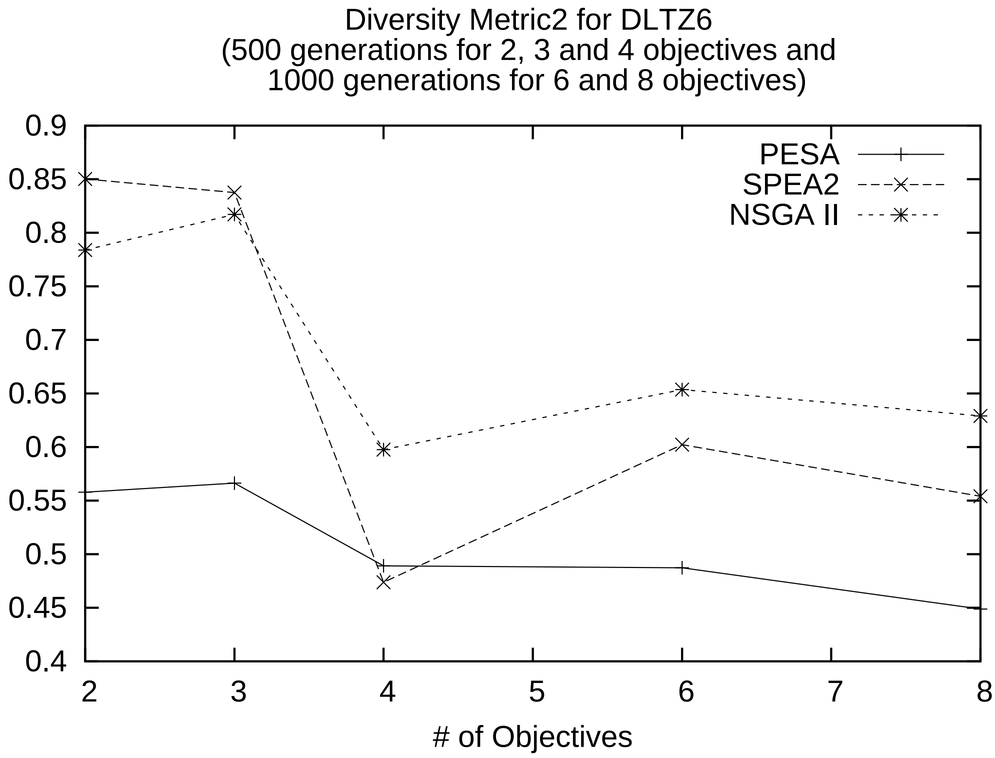


Figure 6.8: Diversity Metric (1 & 2) for DLTZ6

Chapter 7

Discussion on Results

Following is the discussion over the results presented in section 6.4. Few other points that have been observed during this extensive comparative study are also discussed.

1. *Scalability* Each algorithm scale differently in terms of the performance metrics chosen.

   - PESA scale very well in terms of convergence but poorly in terms of diversity maintenance and running time.
   - SPEA2 scales well in terms of diversity maintenance but suffers in converging to the global PO front and in running time.
   - NSGA-II scales well in terms of running time and diversity maintenance but suffers in converging to the global PO front.

2. *Convergence to PO front* Ability to converge to the PO front was found best in PESA, though it cannot produce a very diverse solution on the converge front. Also it has another difficulty attached to it - it can not scale in terms of time with increasing # of objectives.

   SPEA2 and NSGA-II have comparable performance in terms of convergence, which is inferior to PESA. Both SPEA2 and NSGA-II had difficulties in dealing with local PO fronts especially for higher # of objectives.

3. *Diversity in obtained solutions* Even in terms of diversity of solutions in the converged front SPEA2 and NSGA-II have similar performances, which is much better than PESA. Though, PESA was able to converge to global PO front many a times, when others couldn't, still its performance in terms of diversity of solutions in the PO front is only comparable with others.

4. *Running Time* NSGA-II was the fastest of all three algorithms, primarily because it doesn't involve expensive calculations related to clustering (as in SPEA2) or grid based calculations (as in PESA). Exponential increase in running time for PESA makes it impractical for higher objectives.

5. *Diversity metric* Diversity metric2, which uses actual converged front instead of global PO front should be used to compare two algorithms (for diversity) because even if one of them has converged to true PO front and the other hasn't, the diversity metric2 of the former will be almost equal to its diversity metric1 value, which would not be the case with latter.

6. *Grid size in PESA* Grid size in PESA is a crucial factor. If we choose very fine grids we can hope to get a good performance in terms of diversity but that would make the algorithm even more expensive.

7. *Effect of population size* As discussed in section 6.2, to investigate the scaling of an MOEA we should present it with a population having equal proportion of non-dominated solutions, for all M, to start with, but this is practically impossible because of the required exponential increase in population size.

8. *Swapping offspring in SBX* Swapping the offspring in SBX helped improving the performance of all three algorithms. This is because the way SBX has been formulated (see equation 3.10) - either offspring1 always gets a variable value greater than offspring2 or it always gets a smaller value. PESA always takes one offspring generated each time and rejects the other one, hence improved performance on PESA can be explained but the improvement in performance of SPEA2 and NSGA-II is something that is to be investigated.

9. *Limited function evaluations* There can be two possible approaches to compare two algoritms. We observe the function evaluations needed for an algorithm to achieve a required threshold of performance. Another approach, which was used in this study, is to fix the number of function evaluations and then compare them on the basis of their performance. Latter was followed of obvious practical reasons.

Chapter 8

Conclusion

All of the work that has been done in MOEAs is mostly limited to 2 and 3 objectives. In this project scalability issues related to three of the state-of-art algorithms (PESA, SPEA2 and NSGA-II) were explored. These algorithms were tested for their scalability with respect to number of objectives. Experiments were done starting from 2 to 8 objectives.

These algorithms were tested for their performance, on specially designed scalable test problems. In this study four scalable test problems, namely DLTZ1, DLTZ2, DlTZ3 and DLTZ6, were used because of their ease of construction, scalability to any number of objectives and variables, ability to provide controllable hindrance both in converging to front and in maintaining diversity and because the PO front for these problems were known apriori.

To compare two or more sets of non-dominated solutions of an $M-$objective problem require at least M performance metrics, otherwise this would result in an inaccurate judgement caused by reduction in dimensionality. However, having M performance metrics would make the comparison practically infeasible. In this study three performance metrics were used, in terms of which the scalability of these algorithms were assessed. First metric measures closeness of obtained non-dominated solution to the global PO front, thus it indicates the convergence ability of an MOEA. The second metric indicates the diversity of solutions in the obtained non-dominated set. Running time was also included as one of the metric to evaluate how an MOEA scales in terms of time complexity with increase in number of objectives. Use of 3 metrics has made this approach practically feasible at the cost of accuracy.

As the result of the study on four test problems, PESA was found to be best in terms of converging to the PO front, but it lacks good diversity maintenance. Also the algorithm was found to be slow because of expensive grid based calculations. Exponential increase in running time makes the algorithm infeasible for higher # of objectives. SPEA2 and NSGA-II performed equally well on convergence and diversity maintenance. Their convergence level was inferior to that of PESA but diversity maintenance was better. NSGA-II was found to be much faster than SPEA2 because of the expensive clustering of solutions. Running times for NSGA-II were an order of magnitude less than that of SPEA2 for higher objectives.

Though the results obtained establish different algorithms as 'winners' in terms of different performance metrics, however, they belong to a limited set of test problems and must always be regarded as tentative. Hence much further work is needed on further problems. E.g. problems which can test an MOEA for its ability to handle constraints can be used. Also, comparison of these algorithms with other classical methods on higher # of objectives would be useful.

Chapter 9

References

[CKO00] David W. Corne, Joshua D. Knowles, and Martin J. Oates. The Pareto Envelope-based Selection Algorithm for Multiobjective Optimization. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 839–848, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

[DAPM00] Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

[Deb01] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.

[DG96] Kalyanmoy Deb and Mayank Goyal. A Combined Genetic Adaptive Search (geneAS) for Engineering Design. *Computer Science and Informatics*, 26(4):30–45, 1996.

[DJ02] Kalyanmoy Deb and Sachin Jain. Running Performance Metrics for Evolutionary Multi-objective Optimization. Technical Report 2002004, KanGAL, Indian Institute of Technology, Kanpur 208016, India, 2002.

[DK95] Kalyanmoy Deb and Amarendra Kumar. Real-coded Genetic Algorithms with Simulated Binary Crossover: Studies on Multimodal and Multiobjective Problems. *Complex Systems*, 9:431–454, 1995.

[DPAM02] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[DTLZ01] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical Report 112, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.

[FF93] Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[FF95] Carlos M. Fonseca and Peter J. Fleming. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3(1):1–16, Spring 1995.

[Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[HJ98] Michael Pilegaard Hansen and Andrzej Jaszkiewicz. Evaluating the quality of approximations to the non-dominated set. Technical Report IMM-REP-1998-7, Technical University of Denmark, March 1998.

[HN93] Jeffrey Horn and Nicholas Nafpliotis. Multiobjective Optimization using the Niched Pareto Genetic Algorithm. Technical Report IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA, 1993.

[KC99] Joshua D. Knowles and David W. Corne. The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation. In *1999 Congress on Evolutionary Computation*, pages 98–105, Washington, D.C., July 1999. IEEE Service Center.

[KC02] Joshua Knowles and David Corne. On Metrics for Comparing Nondominated Sets. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 711–716, Piscataway, New Jersey, May 2002. IEEE Service Center.

[Kur91] Frank Kursawe. A Variant of Evolution Strategies for Vector Optimization. In H. P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature. 1st Workshop, PPSN I*, volume 496 of *Lecture Notes in Computer Science*, pages 193–197, Berlin, Germany, oct 1991. Springer-Verlag.

[Osy85] Andrzej Osyczka. Multicriteria optimization for engineering design. *Design Optimization*, pages 193–227, 1985.

[PM98] Geoffrey T. Parks and I. Miller. Selective Breeding in a Multiobjective Genetic Algorithm. In A. E. Eiben, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving From Nature — PPSN V*, pages 250–259, Amsterdam, Holland, 1998. Springer-Verlag.

[Sch84] J. David Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, 1984.

[SD94] N. Srinivas and Kalyanmoy Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.

[VFM96] Rémy Viennet, Christian Fontiex, and Ivan Marc. Multicriteria Optimization Using a Genetic Algorithm for Determining a Pareto Set. *Journal of Systems Science*, 27(2):255–260, 1996.

[ZDT00] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.

[Zit99] Eckart Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, November 1999.

[ZLT01] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001.

[ZLT+02] Eckart Zitzler, Marco Laumanns, Lothar Thiele, Carlos M. Fonseca, and Viviane Grunert da Fonseca. Why Quality Assessment of Multiobjective Optimizers Is Difficult. In W.B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M.A. Potter, A.C. Schultz, J.F. Miller, E. Burke, and N. Jonoska, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002)*, pages 666–673, San Francisco, California, July 2002. Morgan Kaufmann Publishers.

[ZT99] Eckart Zitzler and Lothar Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.

Appendix A

Detailed Results

Detailed results for Convergence Metric (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	2.86948	3.08825	2.27666
	SD	5.93164	5.35433	5.43593
	MIN	0.00591	0.01144	0.00823
	MAX	24.99170	13.08870	24.99000
3	MEAN	0.04419	0.04843	0.38360
	SD	0.12320	0.05331	0.50094
	MIN	0.00069	0.00446	0.01192
	MAX	0.53870	0.26143	2.70094
4	MEAN	0.02317	0.29925	3.10281
	SD	0.09059	0.66360	4.08272
	MIN	0.00112	0.00475	0.29167
	MAX	0.58075	2.96980	17.17910
Detailed results for Convergence Metric(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	0.00084	8.68173	252.33200
	Run 2	0.00165	0.77197	294.02400
	Run 3	0.00032	19.67100	107.66400
	Run 4	0.00035	3.92065	40.62650
	Run 5	0.00105	1.30696	29.76040
	Run 6	0.00153	1.45626	120.24400
	Run 7	0.00066	0.61886	85.20650
	Run 8	0.00167	0.80701	7.57305
	Run 9	0.00043	2.79386	43.84610
	Run 10	0.00322	19.96680	220.63900
8	Run 1	0.00047	499.81300	470.64300
	Run 2	0.00058	498.06100	454.66500
	Run 3	0.00016	496.47800	455.69400
	Run 4	0.00063	500.76200	464.76900
	Run 5	0.00049	497.52400	477.83300
	Run 6	0.00047	496.99100	481.24200
	Run 7	0.00049	467.98760	442.62800
	Run 8	0.00027	506.39978	455.53400
	Run 9	0.00059	495.73289	491.24534
	Run 10	0.00053	522.96570	458.76600

Table A.1: Full results of Convergence Metric for DLTZ1

Detailed results for Convergence Metric (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	0.00008	0.00026	0.00180
	SD	0.00019	0.00029	0.00082
	MIN	0.00001	0.00001	0.00056
	MAX	0.00092	0.00105	0.00337
3	MEAN	0.00035	0.00663	0.01003
	SD	0.00013	0.00224	0.00234
	MIN	0.00017	0.00316	0.00650
	MAX	0.00074	0.01241	0.01549
4	MEAN	0.00170	0.03369	0.04529
	SD	0.00039	0.00846	0.01373
	MIN	0.00109	0.01945	0.02392
	MAX	0.00266	0.05643	0.07656
Detailed results for Convergence Metric(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	0.00273	2.03153	1.67075
	Run 2	0.00303	1.95179	1.75410
	Run 3	0.00399	1.87814	1.63739
	Run 4	0.00281	2.09604	1.90039
	Run 5	0.00292	2.05034	1.70630
	Run 6	0.00302	1.99977	1.52191
	Run 7	0.00331	2.03722	1.66861
	Run 8	0.00259	1.94358	1.62651
	Run 9	0.00289	2.11748	1.71067
	Run 10	0.00273	1.91568	1.55977
8	Run 1	0.00775	2.36491	2.28753
	Run 2	0.00605	2.35616	2.36133
	Run 3	0.00693	2.36460	2.31716
	Run 4	0.00687	2.34644	2.33837
	Run 5	0.00709	2.33081	2.33621
	Run 6	0.00798	2.39876	2.30470
	Run 7	0.00577	2.39032	2.25913
	Run 8	0.00898	2.30543	2.22045
	Run 9	0.00575	2.29008	2.31631
	Run 10	0.00582	2.37833	2.33953

Table A.2: Full results of Convergence Metric for DLTZ2

Detailed results for Convergence Metric (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	22.52023	16.87313	21.32032
	SD	22.90480	16.72477	11.15397
	MIN	0.03424	0.14699	1.50539
	MAX	76.00240	51.41630	54.20570
3	MEAN	1.80296	2.39884	5.65577
	SD	5.78546	4.72212	6.26729
	MIN	0.01761	0.02129	0.05576
	MAX	25.37300	25.17790	31.81220
4	MEAN	1.16736	4.00596	66.94049
	SD	3.50522	4.00594	39.06815
	MIN	0.04186	0.06679	19.89080
	MAX	21.33010	16.95700	178.73900
Detailed results for Convergence Metric(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	0.08938	290.33400	1261.72000
	Run 2	0.46341	127.49400	1288.15000
	Run 3	0.04154	187.74300	1259.34000
	Run 4	0.17326	146.80700	1239.06000
	Run 5	0.05782	287.27000	1321.65000
	Run 6	0.14190	216.79500	1336.58000
	Run 7	0.24349	205.11700	1288.31000
	Run 8	0.12831	235.18500	1167.36000
	Run 9	0.12586	357.76200	1192.43000
	Run 10	0.03852	125.02900	1378.46000
8	Run 1	0.08938	290.33400	1261.72000
	Run 2	0.46341	127.49400	1288.15000
	Run 3	0.04154	187.74300	1259.34000
	Run 4	0.17326	146.80700	1239.06000
	Run 5	0.05782	287.27000	1321.65000
	Run 6	0.14190	216.79500	1336.58000
	Run 7	0.24349	205.11700	1288.31000
	Run 8	0.12831	235.18500	1167.36000
	Run 9	0.12586	357.76200	1192.43000
	Run 10	0.03852	125.02900	1378.46000

Table A.3: Full results of Convergence Metric for DLTZ3

Detailed results for Convergence Metric (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	0.79397	0.77622	0.63697
	SD	0.32237	0.23794	0.29986
	MIN	0.22178	0.24088	0.19279
	MAX	1.46463	1.24145	1.40166
3	MEAN	0.20528	0.29271	0.24515
	SD	0.21199	0.23631	0.22849
	MIN	0.00000	0.00000	0.00000
	MAX	0.69444	0.93898	0.76565
4	MEAN	3.60430	5.07137	6.32619
	SD	0.38084	0.22360	0.36229
	MIN	2.91572	4.66647	5.57813
	MAX	4.75811	5.61403	7.03207
Detailed results for Convergence Metric(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	4.96938	10.53360	9.51737
	Run 2	5.62785	10.53230	9.22393
	Run 3	5.73656	10.53390	9.71738
	Run 4	5.81924	10.54870	9.16755
	Run 5	5.18632	10.53320	9.44906
	Run 6	5.05015	10.54720	9.91849
	Run 7	5.13960	10.53210	9.54346
	Run 8	5.20524	10.52880	9.06623
	Run 9	4.99894	10.54930	9.49485
	Run 10	5.31216	10.52910	9.77670
8	Run 1	6.27629	10.64700	10.36310
	Run 2	6.38993	10.65060	10.23710
	Run 3	6.30120	10.64660	10.17330
	Run 4	6.25879	10.55080	10.33280
	Run 5	6.39879	10.65160	10.30590
	Run 6	6.22593	10.65873	10.30820
	Run 7	6.34523	10.68943	10.23730
	Run 8	6.56539	10.59546	10.28680
	Run 9	6.22674	10.64632	10.21300
	Run 10	6.23645	10.55666	10.27306

Table A.4: Full results of Convergence Metric for DLTZ6

Detailed results for Diversity Metric1 (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	0.25093	0.55656	0.44784
	SD	0.14059	0.34527	0.23329
	MIN	0.03526	0.03526	0.03526
	MAX	0.58000	0.95211	0.85632
3	MEAN	0.42116	0.63186	0.57752
	SD	0.07563	0.04565	0.15660
	MIN	0.25622	0.50867	0.05622
	MAX	0.58439	0.73898	0.70592
4	MEAN	0.37605	0.54176	0.38676
	SD	0.07125	0.02007	0.11091
	MIN	0.24966	0.50037	0.13818
	MAX	0.55178	0.59364	0.55519
Detailed results for Diversity Metric1(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	0.35471	0.18365	0.15051
	Run 2	0.33839	0.46945	0.14663
	Run 3	0.29958	0.15026	0.15727
	Run 4	0.36868	0.23337	0.20690
	Run 5	0.36333	0.44243	0.21597
	Run 6	0.38956	0.48729	0.15900
	Run 7	0.35909	0.47166	0.21035
	Run 8	0.32980	0.50668	0.28714
	Run 9	0.25179	0.46480	0.21995
	Run 10	0.30937	0.15493	0.18056
8	Run 1	0.24988	0.27383	0.19846
	Run 2	0.24294	0.27129	0.18922
	Run 3	0.26454	0.24343	0.20120
	Run 4	0.25093	0.25458	0.22297
	Run 5	0.23966	0.26912	0.20076
	Run 6	0.25094	0.25417	0.20597
	Run 7	0.25893	0.26946	0.19517
	Run 8	0.25294	0.24983	0.18966
	Run 9	0.26092	0.26092	0.17527
	Run 10	0.25286	0.26408	0.19560

Table A.5: Full results of Diversity Metric1 for DLTZ1

Detailed results for Diversity Metric2 (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	0.50019	0.86516	0.72559
	SD	0.17016	0.11721	0.13887
	MIN	0.19790	0.59737	0.45118
	MAX	0.87790	1.00000	0.90421
3	MEAN	0.52274	0.78292	0.76969
	SD	0.10693	0.06054	0.09305
	MIN	0.30306	0.58378	0.55571
	MAX	0.70490	0.91469	0.88745
4	MEAN	0.48240	0.67836	0.58683
	SD	0.09175	0.04362	0.09545
	MIN	0.27714	0.54522	0.28791
	MAX	0.66879	0.74485	0.71646
Detailed results for Diversity Metric2(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	0.41028	0.35522	0.38812
	Run 2	0.41537	0.59946	0.36625
	Run 3	0.31735	0.31547	0.37029
	Run 4	0.39060	0.37129	0.38116
	Run 5	0.43120	0.56978	0.38113
	Run 6	0.48527	0.56011	0.34819
	Run 7	0.39950	0.57194	0.38048
	Run 8	0.39711	0.60378	0.41261
	Run 9	0.27004	0.51754	0.39144
	Run 10	0.41303	0.35161	0.40964
8	Run 1	0.29465	0.35800	0.37856
	Run 2	0.30750	0.36084	0.37844
	Run 3	0.28679	0.34015	0.37905
	Run 4	0.28846	0.34662	0.39439
	Run 5	0.28453	0.33413	0.37017
	Run 6	0.28060	0.34216	0.37194
	Run 7	0.30667	0.37172	0.36276
	Run 8	0.27274	0.33919	0.37647
	Run 9	0.26881	0.39666	0.35262
	Run 10	0.36488	0.40559	0.38661

Table A.6: Full results of Diversity Metric2 for DLTZ1

Detailed results for Diversity Metric1 (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	0.57396	0.81867	0.75177
	SD	0.09135	0.01766	0.03891
	MIN	0.33368	0.77737	0.67684
	MAX	0.72895	0.85158	0.85158
3	MEAN	0.57163	0.67260	0.74996
	SD	0.04344	0.03255	0.02064
	MIN	0.48194	0.58796	0.69776
	MAX	0.69725	0.73755	0.79000
4	MEAN	0.52708	0.62136	0.71360
	SD	0.03692	0.01773	0.01881
	MIN	0.43993	0.58529	0.67242
	MAX	0.60801	0.65145	0.74650
Detailed results for Diversity Metric1(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	0.47917	0.28409	0.49031
	Run 2	0.42277	0.31398	0.46535
	Run 3	0.50914	0.29491	0.45811
	Run 4	0.46548	0.28364	0.49944
	Run 5	0.48906	0.26691	0.49252
	Run 6	0.50623	0.31535	0.49198
	Run 7	0.46243	0.29619	0.48530
	Run 8	0.47268	0.31475	0.47939
	Run 9	0.44485	0.27458	0.46009
	Run 10	0.45818	0.32317	0.50235
8	Run 1	0.43976	0.31882	0.53132
	Run 2	0.42424	0.30597	0.53893
	Run 3	0.43089	0.31192	0.53054
	Run 4	0.44983	0.30257	0.53052
	Run 5	0.45093	0.30796	0.54932
	Run 6	0.49832	0.30963	0.52897
	Run 7	0.47821	0.31796	0.52000
	Run 8	0.43215	0.30819	0.51625
	Run 9	0.39875	0.29875	0.52018
	Run 10	0.31692	0.31271	0.53985

Table A.7: Full results of Diversity Metric1 for DLTZ2

Detailed results for Diversity Metric2 (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	0.58509	0.82186	0.75863
	SD	0.09700	0.01992	0.04302
	MIN	0.33368	0.77737	0.68526
	MAX	0.75947	0.85632	0.87790
3	MEAN	0.57993	0.71680	0.81107
	SD	0.04528	0.03629	0.02452
	MIN	0.48367	0.64061	0.76888
	MAX	0.70408	0.79347	0.86286
4	MEAN	0.57993	0.71680	0.81107
	SD	0.04528	0.03629	0.02452
	MIN	0.48367	0.64061	0.76888
	MAX	0.70408	0.79347	0.86286
Detailed results for Diversity Metric2(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	0.52718	0.63810	0.73353
	Run 2	0.46229	0.65994	0.73879
	Run 3	0.57885	0.62894	0.72403
	Run 4	0.51590	0.64356	0.74832
	Run 5	0.55398	0.63955	0.75395
	Run 6	0.56575	0.67661	0.72128
	Run 7	0.51917	0.65037	0.72152
	Run 8	0.51545	0.64318	0.72565
	Run 9	0.49500	0.64741	0.72531
	Run 10	0.49994	0.66520	0.73300
8	Run 1	0.58532	0.64035	0.76048
	Run 2	0.55634	0.64166	0.75855
	Run 3	0.57983	0.64480	0.75159
	Run 4	0.56852	0.63780	0.76216
	Run 5	0.55638	0.64660	0.77560
	Run 6	0.57557	0.64394	0.76009
	Run 7	0.59100	0.64032	0.74527
	Run 8	0.60644	0.64566	0.74251
	Run 9	0.54187	0.64653	0.75080
	Run 10	0.54701	0.64739	0.76850

Table A.8: Full results of Diversity Metric2 for DLTZ2

Detailed results for Diversity Metric1 (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	0.14023	0.17339	0.08846
	SD	0.14497	0.19218	0.06994
	MIN	0.00000	0.03526	0.03526
	MAX	0.57947	0.72947	0.24632
3	MEAN	0.38965	0.62793	0.26244
	SD	0.13220	0.14088	0.19989
	MIN	0.01367	0.05541	0.02051
	MAX	0.57480	0.71541	0.71694
4	MEAN	0.31659	0.58861	0.15869
	SD	0.09393	0.02637	0.03146
	MIN	0.10138	0.55407	0.10054
	MAX	0.45323	0.65209	0.22212
Detailed results for Diversity Metric1(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	0.26200	0.07294	0.13776
	Run 2	0.09638	0.07929	0.10926
	Run 3	0.23940	0.10269	0.10434
	Run 4	0.16981	0.10936	0.13500
	Run 5	0.27371	0.07737	0.11782
	Run 6	0.17187	0.09992	0.10068
	Run 7	0.07937	0.08291	0.11834
	Run 8	0.16581	0.09068	0.12998
	Run 9	0.19127	0.07354	0.12310
	Run 10	0.23168	0.07018	0.13060
8	Run 1	0.02315	0.15694	0.07371
	Run 2	0.02671	0.15600	0.06647
	Run 3	0.02252	0.16160	0.07088
	Run 4	0.02255	0.13582	0.06314
	Run 5	0.02825	0.14899	0.09328
	Run 6	0.02645	0.14104	0.07017
	Run 7	0.02705	0.13744	0.07166
	Run 8	0.02766	0.13383	0.05779
	Run 9	0.02826	0.13022	0.05164
	Run 10	0.02886	0.12661	0.05042

Table A.9: Full results of Diversity Metric1 for DLTZ3

Detailed results for Diversity Metric2 (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	0.49708	0.71668	0.57169
	SD	0.14339	0.15166	0.15533
	MIN	0.19500	0.32053	0.27800
	MAX	0.78579	0.95211	0.78579
3	MEAN	0.58655	0.78540	0.60255
	SD	0.09515	0.11143	0.18633
	MIN	0.29265	0.23735	0.15061
	MAX	0.75480	0.87327	0.87490
4	MEAN	0.51138	0.72007	0.50374
	SD	0.07063	0.05674	0.04957
	MIN	0.39182	0.58024	0.38377
	MAX	0.65263	0.84118	0.60764
Detailed results for Diversity Metric2(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	0.42831	0.38503	0.55709
	Run 2	0.25926	0.30965	0.56666
	Run 3	0.36541	0.39917	0.59316
	Run 4	0.32823	0.34057	0.57686
	Run 5	0.41876	0.37533	0.57903
	Run 6	0.31181	0.37423	0.59618
	Run 7	0.20012	0.37125	0.58424
	Run 8	0.30659	0.38373	0.56925
	Run 9	0.33093	0.36958	0.55533
	Run 10	0.34650	0.36024	0.58666
8	Run 1	0.12233	0.55757	0.57234
	Run 2	0.10548	0.54439	0.59580
	Run 3	0.10192	0.54173	0.59897
	Run 4	0.10579	0.55591	0.55738
	Run 5	0.11641	0.55048	0.58668
	Run 6	0.10904	0.54972	0.59635
	Run 7	0.12954	0.54706	0.59111
	Run 8	0.12455	0.56837	0.58802
	Run 9	0.09836	0.55625	0.57749
	Run 10	0.09480	0.53907	0.58431

Table A.10: Full results of Diversity Metric2 for DLTZ3

Detailed results for Diversity Metric1 (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	0.20191	0.44404	0.40846
	SD	0.14198	0.25537	0.19784
	MIN	0.00000	0.00000	0.00000
	MAX	0.45684	0.72421	0.66737
3	MEAN	0.41962	0.64655	0.66157
	SD	0.06423	0.14591	0.13731
	MIN	0.26061	0.07153	0.23674
	MAX	0.52541	0.83102	0.81898
4	MEAN	0.22558	0.22917	0.12825
	SD	0.02790	0.01438	0.01845
	MIN	0.17758	0.19441	0.07926
	MAX	0.28256	0.25495	0.16205
Detailed results for Diversity Metric1(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	0.25504	0.08232	0.06243
	Run 2	0.25248	0.08953	0.10778
	Run 3	0.24496	0.07340	0.06055
	Run 4	0.28109	0.06469	0.08893
	Run 5	0.29282	0.06692	0.10227
	Run 6	0.30431	0.07883	0.07153
	Run 7	0.28642	0.07434	0.06268
	Run 8	0.29658	0.07761	0.06463
	Run 9	0.30113	0.07628	0.06542
	Run 10	0.24829	0.07541	0.06818
8	Run 1	0.28261	0.05203	0.04591
	Run 2	0.27154	0.04990	0.04963
	Run 3	0.27991	0.05054	0.05023
	Run 4	0.27531	0.07294	0.04546
	Run 5	0.27396	0.05378	0.04980
	Run 6	0.27261	0.06380	0.04348
	Run 7	0.27125	0.06645	0.04426
	Run 8	0.26990	0.06910	0.05619
	Run 9	0.26854	0.07175	0.04408
	Run 10	0.26719	0.07441	0.04754

Table A.11: Full results of Diversity Metric1 for DLTZ6

Detailed results for Diversity Metric2 (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	0.55795	0.85028	0.78386
	SD	0.12064	0.06017	0.08507
	MIN	0.29421	0.66790	0.61526
	MAX	0.81211	0.95211	0.92158
3	MEAN	0.56629	0.83748	0.81710
	SD	0.14311	0.03616	0.06102
	MIN	0.26061	0.70663	0.71510
	MAX	0.82857	0.90857	0.92061
4	MEAN	0.48907	0.47386	0.59749
	SD	0.03777	0.02237	0.02803
	MIN	0.39471	0.42633	0.52226
	MAX	0.56633	0.52303	0.67330
Detailed results for Diversity Metric2(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	0.49409	0.60792	0.64865
	Run 2	0.46802	0.62490	0.63540
	Run 3	0.44483	0.60759	0.67659
	Run 4	0.48154	0.60603	0.63410
	Run 5	0.49044	0.59878	0.66059
	Run 6	0.50453	0.58925	0.68078
	Run 7	0.51649	0.60843	0.65140
	Run 8	0.51663	0.58961	0.63621
	Run 9	0.48931	0.59200	0.64283
	Run 10	0.46671	0.59811	0.66998
8	Run 1	0.44999	0.53540	0.62642
	Run 2	0.43841	0.55041	0.62334
	Run 3	0.45818	0.54032	0.62398
	Run 4	0.44977	0.57866	0.63484
	Run 5	0.49856	0.58483	0.63804
	Run 6	0.49153	0.60447	0.63520
	Run 7	0.40238	0.53923	0.62373
	Run 8	0.41323	0.57249	0.63297
	Run 9	0.42408	0.59100	0.62226
	Run 10	0.46246	0.59717	0.62998

Table A.12: Full results of Diversity Metric2 for DLTZ6

Detailed results for Running Time in seconds(30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	2.53	3.70	2.90
	SD	0.51	0.47	0.55
	MIN	2.00	3.00	2.00
	MAX	3.00	4.00	4.00
3	MEAN	9.13	17.38	4.53
	SD	1.56	0.49	0.80
	MIN	7.00	17.00	4.00
	MAX	15.00	18.00	8.00
4	MEAN	70.53	91.45	17.00
	SD	5.53	1.04	0.39
	MIN	59.00	89.00	16.00
	MAX	81.00	93.00	18.00
Detailed results for Running Time in seconds(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	4218	1854	218
	Run 2	4335	1733	222
	Run 3	4300	1787	211
	Run 4	4229	2336	212
	Run 5	4064	1697	209
	Run 6	4105	1671	213
	Run 7	4063	2215	219
	Run 8	4222	1619	208
	Run 9	4673	1584	212
	Run 10	4083	1894	215
8	Run 1	88714	34323	1692
	Run 2	104299	34373	1545
	Run 3	93384	33125	1543
	Run 4	98534	38376	1564
	Run 5	95634	34314	1521
	Run 6	97462	33901	1520
	Run 7	94321	37898	1533
	Run 8	95467	35983	1546
	Run 9	99045	33578	1518
	Run 10	87796	31482	1545

Table A.13: Full results of Running Time for DLTZ1

Detailed results for Running Time in seconds (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	4.60	4.07	3.27
	SD	0.50	0.26	0.59
	MIN	4.00	4.00	3.00
	MAX	5.00	5.00	5.00
3	MEAN	19.83	20.43	4.93
	SD	3.69	0.59	0.83
	MIN	18.00	20.00	4.00
	MAX	36.00	22.00	8.00
4	MEAN	193.98	107.45	18.70
	SD	17.41	1.28	2.42
	MIN	180.00	104.00	18.00
	MAX	254.00	110.00	29.00
Detailed results for Running Time in seconds(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	8153	2304	227
	Run 2	8145	2233	224
	Run 3	8065	2225	234
	Run 4	8318	2961	323
	Run 5	8052	2248	223
	Run 6	7975	2295	226
	Run 7	7867	3045	222
	Run 8	8072	2246	227
	Run 9	8198	2288	228
	Run 10	7983	2235	225
8	Run 1	334600	32537	1743
	Run 2	334683	33196	1556
	Run 3	334983	37034	1571
	Run 4	334543	32994	1544
	Run 5	335894	32849	1548
	Run 6	336534	33848	1552
	Run 7	338763	32890	1544
	Run 8	332784	33903	1536
	Run 9	339764	33952	1563
	Run 10	323867	34017	1551

Table A.14: Full results of Running Time (seconds) for DLTZ2

Detailed results for Running Time in seconds (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	5.57	6.63	5.73
	SD	0.74	0.49	0.79
	MIN	4.00	6.00	5.00
	MAX	7.00	7.00	8.00
3	MEAN	11.83	29.33	7.97
	SD	1.50	0.63	1.13
	MIN	10.00	28.00	7.00
	MAX	15.00	31.00	14.00
4	MEAN	80.80	131.93	29.75
	SD	8.91	1.38	2.84
	MIN	68.00	129.00	29.00
	MAX	102.00	136.00	47.00
Detailed results for Running Time in seconds(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	3509	2956	373
	Run 2	3867	2774	365
	Run 3	3402	2904	371
	Run 4	3254	3778	366
	Run 5	3271	3021	372
	Run 6	3271	2964	512
	Run 7	3502	3755	370
	Run 8	3387	2989	370
	Run 9	3282	3011	375
	Run 10	3279	2879	370
8	Run 1	85976	49509	2928
	Run 2	86053	49758	2574
	Run 3	87437	49631	2589
	Run 4	83515	48684	2591
	Run 5	88929	49329	2566
	Run 6	82326	49952	2565
	Run 7	86748	49808	2573
	Run 8	81579	49665	2574
	Run 9	83133	49108	2534
	Run 10	89898	48378	2561

Table A.15: Full results of Running Time (seconds) for DLTZ3

Detailed results for Running Time in seconds (30 runs) Mean, Standard Deviation (SD), Minimum (MIN) and Maximum (Max) Values				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
2	MEAN	3.77	4.23	3.20
	SD	0.51	0.44	0.56
	MIN	3.00	4.00	2.00
	MAX	5.00	5.00	5.00
3	MEAN	22.23	32.47	7.73
	SD	4.38	0.78	0.45
	MIN	16.00	31.00	7.00
	MAX	39.00	34.00	8.00
4	MEAN	295.15	178.18	31.00
	SD	22.07	7.23	0.39
	MIN	272.00	173.00	30.00
	MAX	373.00	210.00	32.00
Detailed results for Running Time in seconds(10 runs) All 10 runs are listed				
# of Objectives (M)	**Metric Value**	**PESA**	**SPEA2**	**NSGA-II**
6	Run 1	13874	5418	387
	Run 2	13417	5428	385
	Run 3	13487	5418	389
	Run 4	13531	6947	385
	Run 5	13736	5261	391
	Run 6	13470	5252	387
	Run 7	13390	6904	387
	Run 8	13599	5256	384
	Run 9	13460	5267	384
	Run 10	13318	5250	390
8	Run 1	264680	63432	2990
	Run 2	259404	53475	3144
	Run 3	264316	63526	2646
	Run 4	265494	52565	2657
	Run 5	268758	52612	2691
	Run 6	297654	50357	2645
	Run 7	256354	48155	2650
	Run 8	227983	49047	2648
	Run 9	258945	53592	2646
	Run 10	264412	51337	2694

Table A.16: Full results of Running Time (seconds) for DLTZ6

Appendix B

Source Files

Running PESA	
Location	˜msc39vxk/sp/pesa/withSBX/
Files	new.c & metric.cpp
Compiling file1	gcc -lm new.c
Running file1	a.out
Output files	out (non-dominated solutions) & time (running time)
Compiling file2	g++ -lm metric.cpp -o met
Running file2	met
Output files	metrics (contains convergence and diversity metrics)
Note	for all problems: # of variable = cardinality(k) + # of objectives(M) -1

Table B.1: Running PESA

Running SPEA2	
Location	˜msc39vxk/sp/SPEA2/TEA/Ext_Multiobjective/Examples/
Files	teaSPEA2.cc & metric.cpp
Compiling file1	type ‘rem’ on the prompt
Running file1	type ‘teaSPEA2’ on the prompt
Output files	out (non-dominated solutions) & time (running time)
Compiling file2	g++ -lm metric.cpp -o met
Running file2	met
Output files	metrics (contains convergence and diversity metrics)
Note	for all problems: # of variable = cardinality(k) + # of objectives(M) -1

Table B.2: Running SPEA2

Running NSGA-II	
Location	~msc39vxk/sp/nsga2code/
Files	nsga2.c & metric.cpp
Compiling file1	gcc -lm nsga2.c
Running file1	a.out
Output files	out (non-dominated solutions) & time (running time)
Compiling file2	g++ -lm metric.cpp -o met
Running file2	met
Output files	metrics (contains convergence and diversity metrics)
Note	for all problems: # of variable = cardinality(k) + # of objectives(M) -1

Table B.3: Running NSGA-II

Appendix C

MSc. project Declaration

Appendix D

Statement of Information Search Strategy

1. **Parameters of literature search**

   Most of the literature search was carried out for the first part of the project in which various MOEAs, performance metrics and test problems were searched for.

   - Forms of literature

     The categories explored maximally were conference papers and Journal articles; Technical report, theses books and www pages were also searched for. My supervisor Prof. K. Deb also provided me with some references.

   - Age-Range of literature

     First ever MOEA was proposed in 1985. So the search was carried out from 1985 onwards.

   - Restrictions as to language

     I don't know any language other than English so my search will be limited to articles published in English.

2. **Appropriate search tools**

   - Engineering Index

     Ei Compendex (http://edina.ac.uk/compendex/login.shtml), can be used to search interdisciplinary engineering information database in the world. Various keywords can be used to search for conference papers, journal articles and some theses.

   - Science Citation Index (SCI)

     This was used for finding relevant journal papers. Cited reference search was also performed using SCI. In SCI keywords can be used to find some papers to start with, further cited reference search can be used. ResearchIndex (http://citeseer.nj.nec.com/cs) and Web of Science (http://wos.mimas.ac.uk) are two useful science citation indices.

   - Dissertations Abstracts International (DAI) and Index to Theses

     DAI (http://wwwlib.umi.com/dissertations/gateway) can be used to retrieve North American theses and Index to Theses (http://www.theses.com) can be used to retrieve UK theses.

3. **Search statements**

   The search statements will be based on following terms: Multi-Objective Evolutionary Algorithms (MOEAs), Evolutionary Multi-objective optimization, Pareto-optimality, Non-dominated solutions, etc - various combinations of these keywords were used.

4. **Brief evaluation of the search**

   For the first part of the project, using various keywords for searching in Science Citation Index, following 9 items were retrieved.

- 5 Conference proceedings
- 3 Technical Reports
- 1 PhD Thesis

Using these as starting points, cited reference search was carried out. For the second part of the project, my project supervisor suggested a few references. Other literature items were retrieved by the cited reference search.

Appendix E

List of Acronyms

DLTZ	Deb Laumanns Thele Zitzler (test problems)
EMO	Evolutionary Multi-objective optimization
MOEA	Multi-objective Evolutionary Algorithms
MOGA	Multi-objective Genetic Algorithm
MOOP	Multi-objective optimization problem
NPGA	Niched Pareto Genetic Algorithm
NSGA	Non-dominated Sorting Genetic Algorithm
PAES	Pareto Archived Evolution Strategy
PESA	Pareto Enveloped-based Selection Algorithm
PO	Pareto Optimal
SBX	Simulated Binary Crossover
SPEA	Strength Pareto Evolutionary Algorithm
VEGA	Vector Evaluated Genetic Algorithm