OBJECTIVE REDUCTION IN MANY OBJECTIVE OPTIMISATION PROBLEMS USING EVOLUTIONARY ALGORITHM

A thesis submitted in partial fulfillment of the requirements for the award of the degree of

M. Tech.

in

Many Objective Optimization

by

Abhiranjan Kumar

2008IPG-02

विश्वजीवनामृतं ज्ञानम्

ABV INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT

GWALIOR-474015

2013

CANDIDATE’S DECLARATION

I hereby certify that I have properly checked and verified all the items as prescribed in the checklist and ensure that my thesis/report is in proper format as specified in the guideline for thesis preparation.

I also declare that the work containing in this report is my own work. I, understand that plagiarism is defined as anyone or combination of the following:

1. To steal and pass off (the ideas or words of another) as one’s own
2. To use (another’s production) without crediting the source
3. To commit literary theft
4. To present as new and original an idea or product derived from an existing source.

I understand that plagiarism involves an intentional act by the plagiarist of using someone else’s work/ideas completely/partially and claiming authorship/originality of the work/ideas. Verbatim copy as well as close resemblance to some else’s work constitute plagiarism.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programmes, experiments, results, websites, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report/dissertation/thesis are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable. My faculty supervisor(s) will not be responsible for the same.

Abhiranjan Kumar

Roll No.: 2008IPG-02

Date: July 23, 2013

Abstract

Evolutionary multiobjective optimization (EMO) is one of the most active research areas in the field of evolutionary computation. Whereas EMO algorithms have been successfully used in various application tasks, it has also been reported that they do not work well on many-objective problems. In this paper, first we examine the problem posed by the increase in objectives in multiobjective optimization problem. Next we briefly review recent proposals for the scalability improvement of EMO algorithms to many-objective problems. Then we will suggest a new method using inversion counts and disorder of a sequence to find all the set of redundant objectives and select a representative objective from each such set.

Key Words: Algorithm design, evolutionary computation, many-objective optimization, multiobjective optimization, objective reduction, inversion, disorder, rank.

ACKNOWLEDGEMENT

I am highly indebted to **ABV-Indian Institute of Information Technology & Management and Dr. Pramod Kumar Singh**, and obliged as they have given us the autonomy of functioning and experimenting with ideas. I would like to take this opportunity to express my profound gratitude to them not only for their academic guidance but also for their personal interest in my project and constant support coupled with confidence boosting and motivating sessions which proved very fruitful and were instrumental in infusing self-assurance and trust within me. The nurturing and blossoming of the present work was mainly due to their valuable guidance, suggestions, astute judgment, constructive criticism and an eye for perfection. My mentor always answered myriad of my doubts with smiling graciousness and prodigious patience, appreciating my work and improving it by giving us a free hand in my project. It‘s only because of their overwhelming interest and helpful attitude, the present work has attained the stage it has.

I am especially grateful for all the help provided and the resources that were made available to me without which the project would not have been satisfactorily completed. I would also like to thank our parents whose good wishes and silent blessings always remained with me throughout the course of the project.

Finally, I am grateful to all our friends and colleagues, whose constant encouragement served to renew my spirit, refocus my attention and energy and helped me in carrying out this work.

Abhiranjan Kumar
2008IPG-02

Dated: July 23, 2013

TABLE OF CONTENTS

List of Tables

List of Figures

1. INTRODUCTION

Evolutionary multiobjective optimization (EMO) is one of the most active research areas in the field of evolutionary computation. Whereas EMO algorithms have been successfully used in various application tasks, it has also been reported that they do not work well on many-objective problems.

Recently a large number of successful applications evolutionary multiobjective optimization (EMO) algorithms have been reported in the literature [1]-[2]. The main advantage of EMO algorithms over other optimization techniques is that multiple non-dominated solutions can be obtained by their single run. Pareto dominance-based algorithms such as NSGA-II [2] have been frequently used in recent studies.

<u>Requirements of a Multi-objective Optimizer for Engineering Design</u>

The globally optimal trade-off surface of a multi-objective optimization problem can contain a potentially infinite number of Pareto-optimal solutions. The task of a multi-objective optimizer is to provide an accurate and useful representation of the trade-off surface to the decision-maker. The set of solutions generated by the optimizer is known as an approximation set. Three aspects of solution set quality can be considered. These are listed below.

Proximity: The approximation set should contain solutions whose corresponding objective vectors are close to the true Pareto front.

Diversity: The approximation set should contain a good distribution of solutions, in terms of both extent and uniformity. Good diversity is commonly of interest in objective-space, but may also be required in decision-space. In objective-space, the approximation set should extend across the entire range of the true Pareto front with a parametrically uniform distribution across the surface.

Pertinency: The approximation set should only contain solutions in the decision maker's (DM's) region of interest (ROI). In practice, and especially as the number of objectives increases, the DM is interested only in a sub-region of objective space. Thus, there is little benefit in representing trade-off regions that lie outside the ROI. Focusing on pertinent areas of the search space helps to improve optimizer efficiency and reduces unnecessary information that the DM would otherwise have to consider.

Whereas such an EMO algorithm usually works very well on two-objective problems, their search ability is severely deteriorated by the increase in the number of objective functions. For example, it was demonstrated in the literature [3] that multiple runs of single-objective optimization methods outperformed EMO algorithms when they were applied to many-objective problems under the same computation load. This is because almost all solutions in each population become non-dominated with each other when EMO algorithms are applied to many-objective problems. As a result, a diversity maintenance mechanism has a dominant effect on many-objective evolution (because Pareto dominance cannot generate a large selection pressure toward the Pareto front).

Multiobjective problems with four or more objectives are often referred to as many-objective problems. Whenever a well-known and frequently-used Pareto dominance-based EMO algorithm is applied to such a many-objective problem, following three serious difficulties are faced.

1. Deterioration of the search ability of Pareto dominance-based EMO algorithms such as NSGA-II[2]. When the number of objectives increases, almost all solutions in each population become non-dominated. This severely weakens the Pareto dominance-based selection pressure toward the Pareto front. That is the convergence property of EMO algorithms is severely deteriorated.

2. Exponential increase in the number of solutions required for approximating the entire Pareto front. The goal of EMO algorithms is to find a set of non-dominated solutions that well approximate the entire Pareto front. Since the Pareto front is a hyper-surface in the objective space, the number of solutions required for its approximation exponentially increases with the dimensionality of the objective space (i.e., with the

number of objectives.) That is, we may need thousands of non-dominated solutions to approximate the entire Pareto front of a many-objective problem.

3. Difficulty of the visualization of solutions. It is usually assumed that the choice of a final solution from a set of obtained non-dominated solutions is done by a decision maker based on his/her preference. The increase in the number of objectives makes the visualization of obtained non-dominated solutions very difficult. This means that the choice of a final solution becomes very difficult in many-objective optimization.

The first difficulty (i.e., the deterioration of the search of the search ability of EMO algorithms by the increase in the number of objectives) has been pointed out in a number of studies. The deterioration of the search ability was clearly demonstrated through the comparison with multiple runs of single-objective optimizers. A straightforward idea for the scalability improvement of EMO algorithms to many-objective problems is to increase selection pressure toward Pareto dominance in order to decrease the number of non-dominated solutions in each population. Another idea for the scalability improvement is the use of different fitness evaluation mechanisms, instead of Pareto dominance. One approach based on this idea is the use of indicator-based evolutionary algorithms where hypervolume are used to evaluate each solution. Another approach is to use different scalarizing functions for fitness evaluation.

The second difficulty (i.e., the exponential increase in the number of non-dominated solutions that are necessary for the approximation of the Pareto front) has often been tackled by incorporating preference information in EMO algorithms. Preference information is used to concentrate on a small region of the Pareto front while EMO algorithms are used to find multiple non-dominated solutions in such a small region of the Pareto front.

A direct approach for the handling of the third difficulty (the difficulty of the visualization of solutions) is to decrease the number of objectives. Of course, dimensionality reduction (i.e., objective reduction) can remedy not only third difficulty but also the other difficulties. Visualisation techniques of non-dominated solutions with many objectives have been proposed in literature where objective vectors are mapped into a low-dimensional space for their visualization. A number of visualization techniques of high-dimensional objective vectors have also been proposed in the field of multiple criteria decision making.

2. MOTIVATION

Almost all the materialistic problem can be codified as a multi/many objective problem. We have to trade one objective to improve other. Until now we used human wisdom to select solutions from a given set of solutions. We can use genetic algorithms to make machines to learn from their mistakes and success and take wise decision to select the solutions from the given solution set.

Most of the real life problems have more than 2 conflicting objectives. This make the selection of a good solution from the solution set computationally difficult. So we will work to develop methods for removing redundant objectives in many objective optimisation problem so that multiobjective optimisation algorithms will also be computationally compatible with them.

3. LITERATURE REVIEW

The work of K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan in “A fast and elitist multiobjective genetic algorithm: NSGAII”[2] in 2002 suggested a nondominated sorting-based multiobjective evolutionary algorithm (NSGA-II) which has a better complexity than the previous algorithms proposed and also used the elitist approach. They tested this algorithm on various test problems and find that this algorithm has a better diversity and fast converging than other algorithms. NSGA-II was able to approach the true front than the other prevailing approaches. It attributed the NSGA-II with fast nondominated sorting procedure, approach with no parameters (parameterless approach), elitist strategy and efficient constraint-handling method.

In 2008 Hisao Ishibuchi, Noritaka Tsukamoto and Yusuke Nojima published a paper "Behavior of Evolutionary Many-Objective Optimization"[4] in which they discuss their experiments on the most well-known evolutionary multiobjective algorithm, NSGA-II. They applied this algorithm on multiobjective 0/1 knapsack problems with 2, 4, 6 and 8 objectives. They also experimented the recent proposals and examine their effects on the search ability of EMO algorithms. In this paper, they observed that all solutions in each population were non-dominated with each other in almost all generations (except for early generations). The number of non-dominated solutions in the merged population before the generation update in NSGA-II was increased by the increase in the number of objectives. As a result the convergence of solutions to the Pareto front was deteriorated by the increase in the number of objectives.

In the paper "Evolutionary Many-Objective Optimization: A short review"[5], the authors Hisao Ishibuchi, Norataka Tsukamoto and Yuske Nojima demonstrated the difficulties faced by evolutionary multiobjective optimization algorithms in their scalability to many-objective problems. They also reviewed some approaches proposed in the literature for the scalability improvement of EMO algorithms. The proposed some of the future work relating to hypervolume calculation, use of alternative indicators and use of a set of uniform scalarizing functions. They also proposed other approaches that focus on region-specific search in many objective problems in which along with Pareto dominance, decision maker's preference should be considered.

In their work "The plane with parallel coordinates: The Visual Computer"[6], A. Inselberg, in 1985, proposed a system which removes the limitation faced while using Cartesian system when dealing with higher dimensions by introducing the concept of parallel coordinates. The Cartesian system of having the axes orthogonal to each other has obvious limitations when trying to visualize geometry with higher than 3 dimensions or more than 3 variables in a set of data. Parallel coordinates give a systematic and rigorous way of representing the relationships between multiple variables. The approach of parallel coordinates places all the axes parallel to each other thus allowing any number of axes to be shown in a flat representation.

The work of P. J. Fleming, R. C. Purshouse, R. J. Lygoe in "Many-Objective Optimization: An Engineering Design Perspective"[7], in 2007, discusses the application of multiobjective optimization approaches in engineering problems. With this approach, new challenges for algorithm design, visualization and implementation arises. In this paper these three topics was addressed. The use of goals and a preferability operator in a progressive articulation of preferences setting have been demonstrated to be effective in selectively reducing the region of interest in a many-objective search, mitigating the prevailing lack of selective pressure. Through this approach, methods of reducing the dimensionality of the problem have been introduced. In this paper, approaches for improving the computation power, like use of grid computing, use of PDA enabled for computational steering is also discussed.

One of the most well known and frequently used evolutionary multiobjective optimisation algorithm, NSGA-II, was proposed in [2] by K. Deb et. al. It has the ($\mu + \lambda$)-ES style generation update mechanism as follows (usually $\mu = \lambda$ in NSGA-II):

Outline of NSGA-II

Step 1:	P := Initialize (*P*)
Step 2:	while a termination condition is not satisfied, do
Step 3:	*P′* := Selection (*P*)
Step 4:	*P’’* := Genetic Operations (*P’*)
Step 5:	*P* := Generation Update(*P*∪*P’’*)
Step 6:	end while
Step 7:	return (Non-dominated solutions (*P*))

In Step 3, each solution in the current population is evaluated using Pareto sorting (as a primary criterion) and a crowding distance (as a secondary criterion) in the following manner for parent selection. First the best rank (i.e., Rank 1) is assigned to all the nondominated solutions in the current population. Those solutions with Rank 1 are tentatively removed from the current population. Next the second best rank (i.e., Rank 2) is assigned to all the non-dominated solutions in the remaining population. Solutions with Rank 2 are tentatively removed from the remaining population. In this manner, ranks are assigned to all solutions in the current population. To compare solutions with the same rank, a crowding distance is used as a secondary criterion. Roughly and informally speaking for two objective problems, the crowding distance of a solution is the Manhattan distance between its two adjacent solutions in the objective space. When two solutions have the same rank, one solution with a larger crowding distance is viewed as being better than the other with a smaller distance. A prespecified number of pairs (i.e., λ pairs) of parents are selected from the current population *P* by binary tournament selection with replacement to form a parent population *P*' in Step 3. An offspring solution is generated from each pair of parents by crossover and mutation to form an offspring population *P*'' in Step 4. The current and offspring populations are merged to form an enlarged population in Step 5. Each solution in the merged (i.e., enlarged) population is evaluated by Pareto sorting and the crowding distance in the same manner as Step 3. A prespecified number of the best solutions (μ solutions) are chosen from the merged population as the next population *P* in Step 5[4].

The solution set generated by any evolutionary algorithm should follow following characteristics:

1) Maximum sum of the objective values: MaxSum

$$\mathrm{MaxSum}(\Psi) = \max_{x \in \Psi} \sum_{i=1}^{k} f_i(x)$$

This evaluates the convergence of solutions toward the Pareto front around its centre region.

2) Sum of maximum objective values: SumMax

$$SumMax(\Psi) = \sum_{i=1}^{k} \max_{x \in \Psi} f_i(x)$$

This measure evaluates the convergence of solutions toward the Pareto front around its k edges.

3) <u>Sum of the range of the objective values: Range</u>

$$Range\ (\Psi) = \sum_{i=1}^{k} [\max_{x\, \in\, \Psi}\{f_i(x)\} - \min_{x\, \in\, \Psi}\{f_i(x)\}]$$

This measure evaluates the diversity of solutions in the objective space.

Whereas such an EMO algorithm usually works very well on two-objective problems, their search ability is severely deteriorated by the increase in the number of objective functions. For example, it was demonstrated in the literature [3] that multiple runs of single-objective optimization methods outperformed EMO algorithms when they were applied to many-objective problems under the same computation load.

Issues of Many Objective Optimizations

Interaction often arises between objectives and these have been classified as conflict or harmony. A relationship in which performance in one objective is seen to deteriorate as performance in another is improved is described as conflicting. A relationship in which enhancement of performance in an objective is witnessed as another objective is improved can be described as harmonious. The conflict that exists in a many-objective optimization task is a serious challenge for EMO researchers.

For N conflicting objectives, an (N-1)-dimensional trade-off hypersurface exists in objective space. Due to the 'curse of dimensionality' (the sparseness of data in high dimensions), the ability to fully explore surfaces in greater than five dimensions is regarded as highly limited. It is advisable to use dimensionality reduction techniques prior to application of the estimator. This assumes that the 'true' structure of the surface is of lower dimension, but the potential for reduction may be limited for a trade-off surface in which all objectives are in conflict with each other.

Preference Based Methods

The exploitation of Decision Maker preferences, either a priori, a posteriori, or progressively, is arguably the current best technique for handling large numbers of conflicting objectives. In the a priori and progressive cases, the aim of EMO is to achieve a good representation of trade-off regions of interest to the DM (essentially limiting the ambition of the optimizer by requiring it to represent only a sub-space of the trade-off hyper- surface).

In a priori schemes, decision maker preferences are incorporated before the search begins. In progressive methods, DM preferences are incorporated during the search. The key advantage of these techniques over a priori methods is that the DM may be unsure of his or her preferences at the beginning of the procedure and may be informed and influenced by information that becomes available during the search. The final class of methods is a posteriori, in which a solution is chosen from the approximation set returned by the optimizer.

Visualisation (Parallel Coordinates)

The approach of parallel coordinates places all the axes parallel to each other thus allowing any number of axes to be shown in a flat representation. Fig. 1 illustrates the mapping between the Cartesian system and the corresponding representation in parallel coordinates, where points A and B in the coordinate system are represented by lines in the parallel coordinates representation. Fig. 2 illustrates a representation that deals with more than two objectives (four objectives, in fact). Here, each line in the graph connects the performance objectives achieved by an individual member of the population and represents a potential solution to the design problem[7].

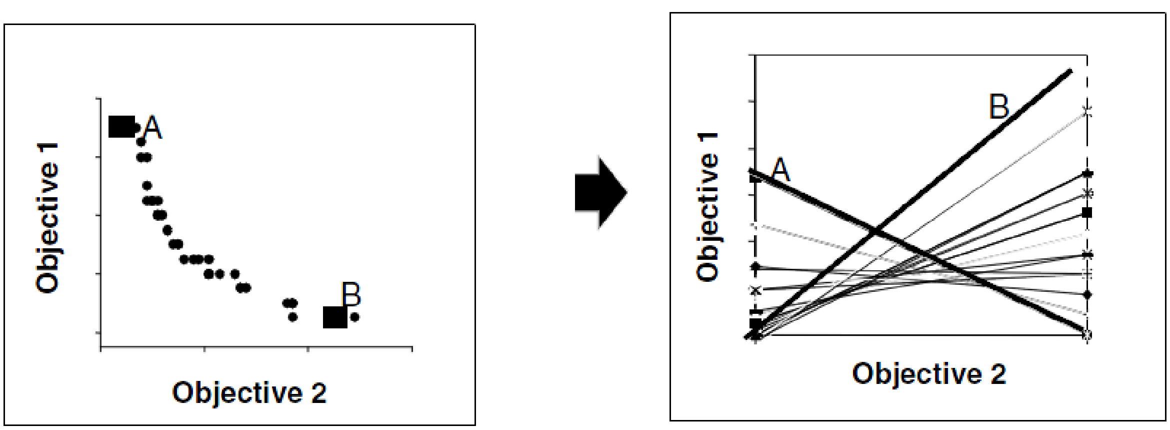

Figure 1. Mapping between Cartesian system and corresponding parallel coordinate

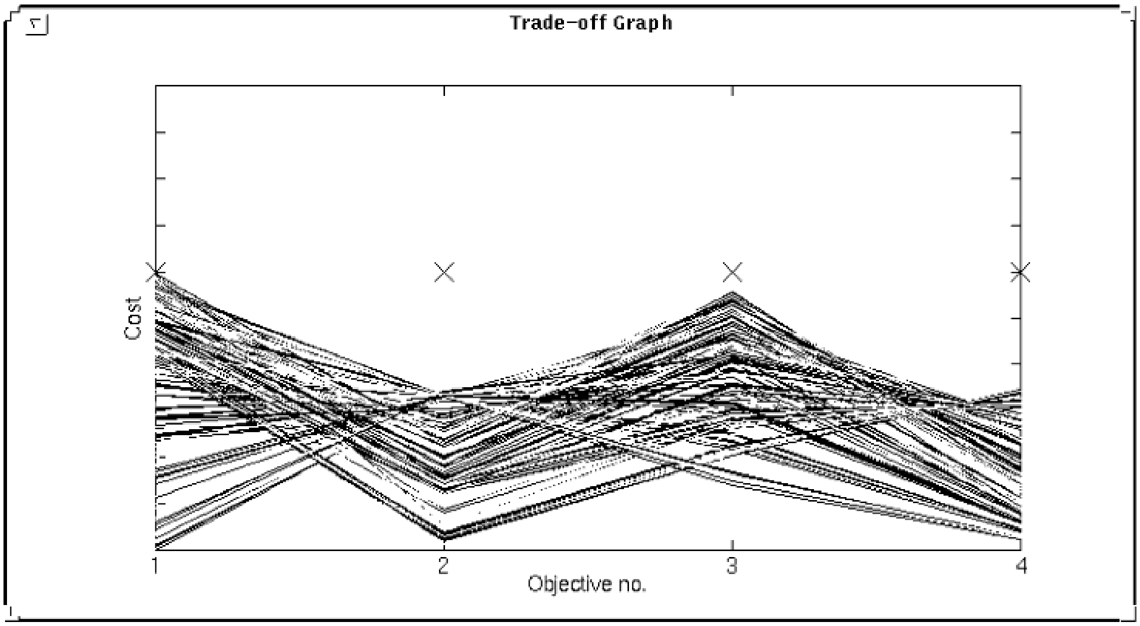

Figure 2. Parallel coordinates for four objectives

It is not sufficient just to be able to display multivariate data in a 2-dimensional representation. The key requirement is to be able to easily interpret the relationships between the variables. It can be shown that the geometrical features of a surface in n-dimensional space are preserved in the parallel coordinates system. This is important because it allows these features to be easily identifiable when represented in parallel coordinates and therefore the relationship between the variables that give rise to these features can be visualised

4. OBJECTIVES

The problem statement of Objective reduction in Many Objective Optimisation using Evolutionary Algorithms states that given a problem with many objectives, we had to find the subset of objectives whose removal doesn't affect the final solution. The solution set thus obtained should not deviate by the intended solution by a large factor.

5. METHODLOGY

5.1 Preliminaries

In this study, we assume that all objectives are equally important and, without loss of generality, we will refer only to minimization problems. Here, we are interested in solving many-objective optimization problems with the following form.

$$\textbf{Minimize } F(X_i) = [f_1(X_i), f_2(X_i), \dots, f_M(X_i)]^T$$
$$\textbf{Subject to } \boldsymbol{X_i} \in \boldsymbol{R}$$

Where $\boldsymbol{X_i}$ is a decision vector (containing decision variables). $\mathbf{F(X_i)}$ of the M-dimensional objective vector ($M > 3$), $\mathbf{f_m(X_i)}$ is the *m-th* objective function, and R is the feasible region delimited by the problem's constraints.

From now on, we will use the concept which is much similar to that of ranking dominance. The idea is to find ranks of solutions in terms of each objective, separately, and use these ranks to find the redundant objectives.

We denote the rank of i'th vector, $\mathbf{X_i}$, with respect to j'th objective, $\mathbf{f_j}$, as $\mathbf{Rank(f_j(X_i))}$ in a set of solutions. Vector with rank 1 is the best vector and vector with rank N is the worst vector with respect to corresponding objective.

$$Rank\big(f_j(X_i)\big) = 1 + \big|\{X_k \mid f_j(X_k) < f_j(X_i)\}\big|$$

Apart from above, we assume that all the vector X_i, $i \in [1, N]$, have unique ranking with respect to each objective.

5.2 Inversion Sequence and Disorder of a permutation

Let $\{x_1, x_2, \dots, x_n\}$ be a permutation of the set $\{1, 2, \dots, n\}$. The pair (x_i, x_j) is called an inversion if $i < j$ and $x_i > x_j$. Thus an inversion in a permutation corresponds to a pair of numbers that are out of their natural order. For example, the permutation $\{3, 1, 5, 2, 4\}$ has four inversions, namely $(3, 1)$, $(3, 2)$, $(5, 2)$ and $(5, 4)$. The only permutation of $\{1, 2, \dots, n\}$ with no inversions is $\{1, 2, \dots, n\}$.

For a permutation $\{x_1, x_2, \ldots, x_n\}$ we let a_i denote the number of inversions whose second component is i, In other words,

a_i equals the number of integers that precede i in the permutation but are greater than component. It measures how much i is out of order.

$$a_i = \left|\left\{x_j \;\middle|\; j < i \text{ and } x_j > x_i\right\}\right|$$

Inversion Sequence:

Inversion sequence for permutation is defined as the sequence of numbers $\{a_1, a_2, \ldots, a_n\}$.

$$I = \{a_1, a_2 \ldots a_n\}$$

Disorder of permutation

The disorder of a permutation $\{x_1, x_2 \ldots x_n\}$ *with inversion sequence* $\{a_1, a_2 \ldots a_n\}$ is defined as

$$\text{Disorder,} \qquad d = a_1 + a_2 + \ldots + a_n$$

$$d = \sum_{k=1}^{n} a_k$$

Given the ranking of solution vectors with respect to two different objectives, we will use the above theory to find the redundancy between two objectives.

Below are few examples, where we will use the above methods to find the degree of disorder between *i*'th and *j*'th objective, F_i and F_j, respectively.

Example 1: Completely redundant objectives

	Rank(F_i)	*Rank(F_j)*
X_1	1	1
X_2	2	2
X_3	4	4
X_4	3	3
X_5	6	6
X_6	5	5

Sorting with respect to *Rank(F_i)* and calculating inversion sequence.

	Rank(F_i)	*Rank(F_j)*	*Inversion*
X_1	1	1	0
X_2	2	2	0
X_4	3	3	0
X_3	4	4	0
X_6	5	5	0
X_5	6	6	0
Disorder			0

$$Normalised\ disorder, d = \frac{0}{\left(\frac{6*5}{2}\right)} = 0$$

Example 2: Highly redundant objectives

	Rank(F_i)	*Rank(F_j)*
X_1	1	1
X_2	2	2
X_3	4	5
X_4	6	6
X_5	3	3
X_6	5	4

Sorting with respect to $Rank(f_i)$ and calculating inversion sequence.

	Rank(f_i)	*Rank(f_j)*	*Inversion*
X_1	1	1	0
X_2	2	2	0
X_5	3	3	0
X_3	4	5	0
X_6	5	4	1
X_4	6	6	0
Disorder			1

$$Normalised\ disorder, d = \frac{1}{\left(\frac{6*5}{2}\right)} = 0.067$$

Example 3: Loosely redundant objectives

	Rank(F_i)	*Rank*(F_j)
X_1	1	1
X_2	2	4
X_3	4	2
X_4	6	6
X_5	3	3
X_6	5	5

Sorting with respect to *Rank*(F_i) and calculating inversion sequence.

	Rank(F_i)	*Rank*(F_j)	*Inversion*
X_1	1	1	0
X_2	2	4	0
X_5	3	3	1
X_3	4	2	2
X_6	5	5	0
X_4	6	6	0
Disorder			3

$$Normalised\ disorder, d = \frac{3}{\left(\frac{6 * 5}{2}\right)} = 0.20$$

Example 4: Another loosely redundant objectives

	Rank(F_i)	*Rank(F_j)*
X_1	1	2
X_2	2	1
X_3	4	3
X_4	6	5
X_5	3	4
X_6	5	6

Sorting with respect to *Rank(F_i)* and calculating inversion sequence.

	Rank(F_i)	*Rank(F_j)*	*Inversion*
X_1	1	2	0
X_2	2	1	1
X_5	3	4	0
X_3	4	3	1
X_6	5	6	0
X_5	6	5	1
Disorder			3

$$Normalised\ disorder, d = \frac{3}{\left(\frac{6*5}{2}\right)} = 0.20$$

Example 5: Non-redundant objectives

	Rank(F_i)	*Rank*(F_j)
X_1	1	4
X_2	2	5
X_3	4	6
X_4	6	1
X_5	3	3
X_6	5	2

Sorting with respect to *Rank*(F_i) and calculating inversion sequence.

	Rank(F_i)	*Rank*(F_j)	*Inversion*
X_1	1	4	0
X_2	2	5	0
X_5	3	3	2
X_3	4	6	0
X_6	5	2	4
X_4	6	1	5
Disorder			11

$$Normalised\ disorder, d = \frac{11}{\left(\frac{6*5}{2}\right)} = 0.73$$

Theorem: If disorder between two objectives is *0* then these two objectives are redundant on each other.

Proof: Let objective f_i and f_j be two objectives whose disorder, d, is *0*. This means the inversion sequence for them consists of all zero.

$$Disorder, d = 0$$
$$Inversion\ Sequence, I = \{0, 0 \dots\ 0\}$$
$$a_k = 0, \qquad \forall\ k\ \in |R|$$

This leads to the following formulation

$$Rank(F_i(x_k)) = Rank(F_j(x_k)) \quad , \qquad \forall\ x_k \in R$$

Therefore each solution vector, x_k, has same ranking in both objectives, making one of these objectives as redundant.

Axiom: If two objectives are redundant then the disorder, *d*, between these two are much closer to *0* than $N * (N-1)/2$.

Why: If two objectives are redundant then the number of inversions for each component vector tends to be near zero.

$$Number\ of\ inversions, \qquad a_k \approx 0, \qquad \forall\ k\ \in |R|$$
$$i.e., \qquad 0 \le a_k \ll\ k$$
$$Disorder, \qquad d = \sum_{k=1}^{N} a_k$$

Therefore for redundant objectives

$$|d - 0| \ll \left| N * \frac{(N-1)}{2} -\ d \right|$$
$$0\ \le d\ \ll N * \frac{(N-1)}{2}$$

5.3 Algorithm

Now we present algorithm to calculate the normalized disorder matrix between each pair of objective.

create_disorder_matrix (F, X)

1. *let objectives be represented as* $F = [f_1, f_2 \dots f_M]$.

2. *let solution vectors be represented as* $X = [x_1, x_2 \dots x_N]^T$

where, $x_k \in R, \quad \forall\, k \in [1, N]$

and R is the feasible region for solution vector delimited by the problem's constraints.

3. $set\ M := |F|$

4. $set\ N := |X|$

5. *create a M x M disorder matrix,* $D[1,2 \dots M][1,2 \dots M]$.

6. *create a N x M ranking matrix, R, such that*

$$R[i][j] = Rank\left(f_j\,(x_i)\right)$$

7. $for\ each\ pair\ (i, j)\ , \quad 1 \le i, j \le M$

8. let D_{ij} be a *N x 2 matrix, where first and second column represents i'th and j'th objective respectively, i.e.,*

$$D_{ij}\,[k][1] = R[k][i], \quad \forall\, k \in [1, N]$$

$$D_{ij}\,[k][2] = R[k][j], \quad \forall\, k \in [1, N]$$

9. *sort the matrix with respect to first column, that is i'th objective.*

10. *for each row, k, calculate the inversion number,*a_k*, considering j'th objective/second column.*

$$a_k = \left|\left\{ r \,\middle|\, D_{ij}\,[r][2] > D_{ij}\,[k][2], \quad \forall\, r \in [1, k] \right\}\right|$$

11. *add all these inversion numbers to calculate the disorder between i'th and j'th objective, d.*

$$d = \sum_{k=1}^{N} a_k$$

12. *set D[i][j] = d / (N*(N-1)/2).*

13. *end for*

14. *return D*

Now we present the outline of our algorithm

objective_reduction_by_inversions (F)

1. *let* $F = [f_1, f_2 \dots f_M]$ *be initial set of objectives.*

2. *set a threshold value,* ϵ.

3. *while termination condition is not reached*

4. *initialize a random population, P.*

5. $set\ P' := nsga\_2(F, P)$ *// run an evolutionary algorithm (like nsga-II) to create a pareto optimal solution, P', from P.*

6. $set\ P := P'$

7. $set\ D := create\_disorder\_matrix(F, P)$

8. $set\ M := |F|$

9. *set* $group[1..M] := \{1, 2 \dots M\}$

10. *for each pair* $(i,\ j), where\ 1 \le i, j \le M$

11. $if(\ i\ \neq j\ \&\ D[i][j] \le\ \epsilon\ )$

12. $connect\_group(i, j)$

13. *end if*

14. *end for*

15. $set\ F' = \emptyset$

16. *for each connected group* $F'' = [F_{i_1,}F_{i_2} \dots, F_{i_k}]$ *in F*

17. $for\ r = 1\ to\ k$

18. $set\ w_{i_r} := \sum_{p=1}^{k} D[i_r][i_p]$

19. *end for*

20. $select\ r\ where\ w_{i_r} = minimum\ \{w_{i_p} | p\ \in [1, k]\}$

21. $set\ F' := F' \cup r$

22. *end for*

23. $set\ F := F'$

24. *end while*

25. *return F*

5.4 Example for objective reduction from connected objectives

Given a set of 10 objectives

$$F = [f_1, f_2 \dots f_{10}]$$

If we get following disorder matrix, then we will construct the following graph, where we connect all those nodes whose weight is less than or equal to the threshold, ϵ.

Disorder matrix :

Disorder	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
f_1	0	0.01	0.10	0.41	0.86	0.07	0.28	0.91	0.03	0.45
f_2	0.01	0	0.56	0.24	0.45	0.01	0.95	0.65	0.04	0.32
f_3	0.10	0.56	0	0.42	0.77	0.87	0.46	0.045	0.25	0.03
f_4	0.41	0.24	0.42	0	0.32	0.41	0.68	0.28	0.41	0.41
f_5	0.86	0.45	0.77	0.32	0	0.12	0.06	0.71	0.36	0.65
f_6	0.07	0.01	0.87	0.41	0.12	0	0.32	0.85	0.005	0.14
f_7	0.28	0.95	0.46	0.68	0.06	0.32	0	0.68	0.65	0.21
f_8	0.91	0.65	0.045	0.28	0.71	0.85	0.68	0	0.47	0.08
f_9	0.03	0.04	0.25	0.41	0.36	0.005	0.65	0.47	0	0.12
f_{10}	0.45	0.32	0.03	0.41	0.65	0.14	0.21	0.08	0.12	0

Table 1. Disorder Matrix

$Creating\ connected\ group\ of\ objectives\ using\ above\ algorithm, with\ \epsilon = 0.09$

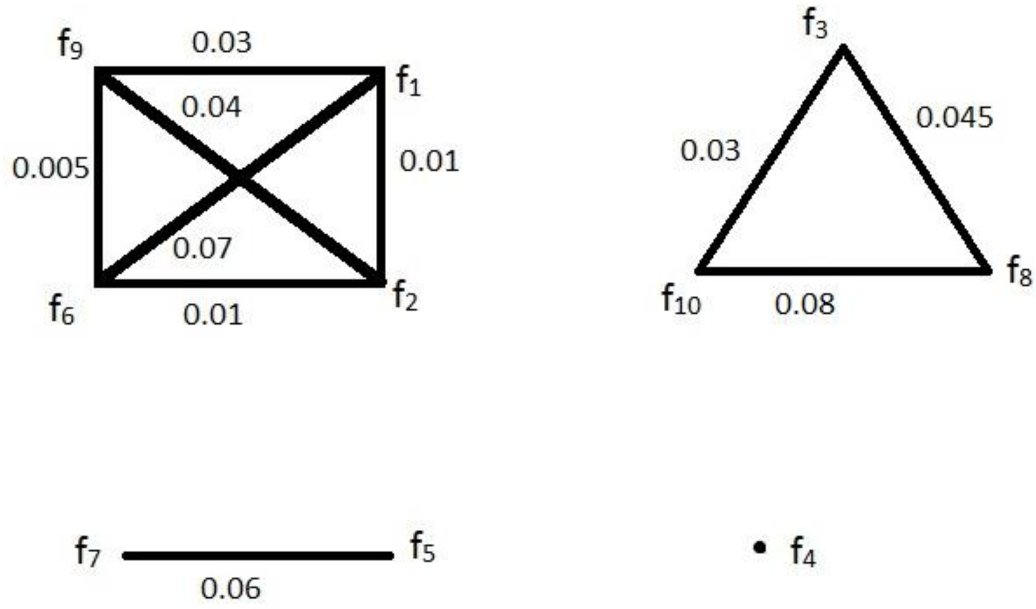

Figure 3. Connected group of objectives

<u>Finding weight of each node in the connected components</u>

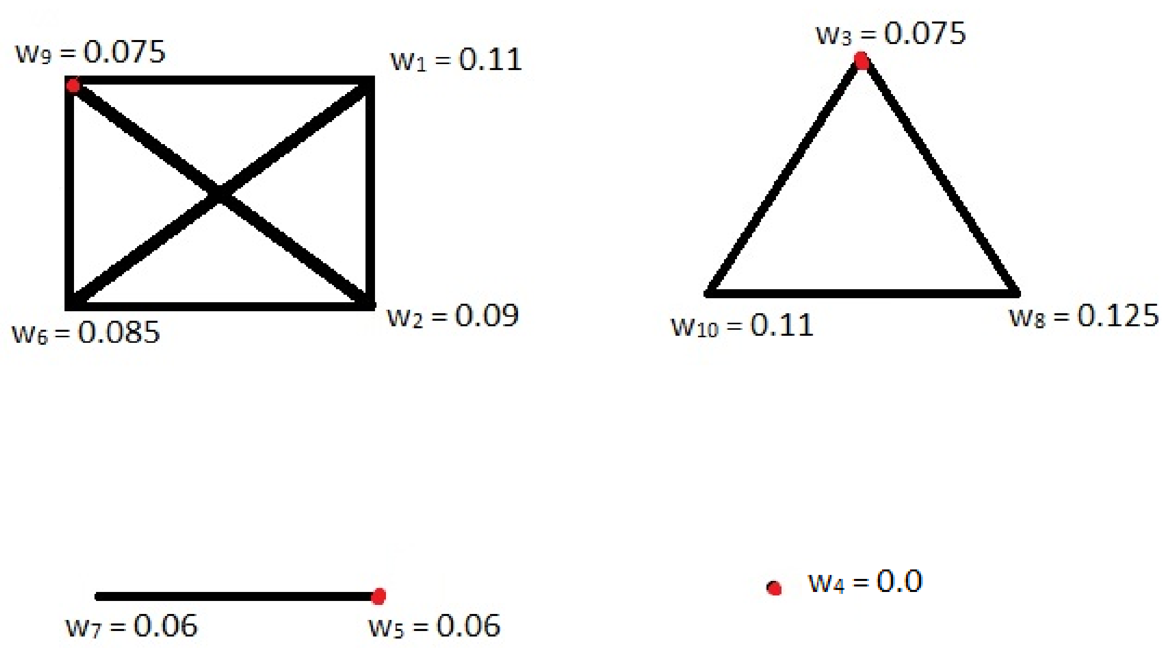

Figure 4. Assigning weights to objectives

Thus from given set of original objectives

$$F = [f_1, f_2 \dots f_{10}]$$

we get these connected components

$$\{f_1, f_2, f_6, f_9\}, \{f_3, f_8, f_{10}\}, \{f_5, f_7\}, \{f_4\}$$

Where each objective in a objective set is redundant to others.

Selecting objectives from each group with minimum weight, w_i, we can find the representative objective from each objective set.

Thus our algorithm will return this reduced objective set

$$F' = [f_3, f_4, f_5, f_9]$$

6. RESULTS AND DISSCUSSION

We executed our algorithm on a variation of DLTZ5 problem having M objectives with a small change in its formulation.

$$
\begin{aligned}
&Min \quad f_1(x) = \big(1 + 100g(x_M)\big)cos(\theta_1)cos(\theta_2) \dots cos(\theta_{M-2})cos(\theta_{M-1}) \\
&Min \quad f_2(x) = \big(1 + 100g(x_M)\big)cos(\theta_1)cos(\theta_2) \dots cos(\theta_{M-2})sin(\theta_{M-1}) \\
&Min \quad f_3(x) = \big(1 + 100g(x_M)\big)cos(\theta_1)cos(\theta_2) \dots sin(\theta_{M-2}) \\
&\qquad\qquad . \qquad\qquad\qquad\qquad . \\
&\qquad\qquad . \qquad\qquad\qquad\qquad . \\
&Min \quad f_{M-1}(x) = \big(1 + 100g(x_M)\big)cos(\theta_1)sin(\theta_2) \\
&Min \quad f_M(x) = \big(1 + 100g(x_M)\big)sin(\theta_1) \\
&where \quad g(x_M) = \sum_{x_i \in x_M} (x_i - 0.5)^2 \\
&\qquad \theta_i = \frac{\pi}{2}(x_i), \qquad\qquad\qquad for\ i = 1, \dots, (I-1) \\
&\qquad \theta_i = \frac{\pi}{4\big(1+g(x_M)\big)}(1 + 2g(x_M)x_i), \quad for\ i = I, \dots (M-1) \\
&\qquad 0 \le x_i \le 1, \qquad\qquad\qquad for\ i = 1,2, \dots, n
\end{aligned}
$$

DTLZ(I, M) Problem Set

Here we choose $k = |x_M| = 10$, so that total number of variables is $n = M + k - 1$. This problem is referred as DTLZ5(I, M), where I denotes the dimensionality of the Pareto-optimal surface and M is the number of objectives in the problem. The variables x_1 till x_I take independent values and are responsible for the dimensionality of the front, whereas other variables (x_I till x_{M-1}) take a fixed value of ${}^{\pi}/_{4}$ for the Pareto-optimal points. A nice aspect this test problem is that by simply setting I to an integer between two and M , the dimensionality (I) of the Pareto-optimal front can be changed. For I = 2, a minimum of two objectives (f_M and any other objective) will be enough to represent the correct Pareto-optimal front.

At each iteration in *objective_reduction_by_inversions* method we use non-dominated sorting genetic algorithm (NSGA-II) to find the pareto optimal solutions which we take into consideration for calculating the disorder matrix in *create_disorder_matrix* method. To provide a optimal computational effect, at each iteration, we have chosen 500 as population size and NSGA-II is run for 1000 generations. We used SBX crossover where a crossover probability is 0.9 and index 5. For mutation we chose polynomial mutation with a probability of 0.1 and index 50.

Our algorithm can be broken down into two separate sub-algorithms. In one part, we assign the disorder between objectives in *F*, and in second we find the redundant group of objectives and select the objectives with least sum of disorder among the group and reject others in subsequent generation. We execute our algorithm until there's no objective reduction is possible.

6.1 Problem DLTZ(2, 5)

This problem has 5 objectives out of which 3 are redundant. The pareto optimal solution can be found with f_5 and any other objective function.
Following are the results of iteration in *objective_reduction_by_inversions*.

Iteration #1

	F1	F2	F3	F4	F5
F1	0	0.2496	0.3283	0.3542	0.7896
F2	0.2496	0	0.3241	0.3581	0.7947
F3	0.3283	0.3241	0	0.3731	0.698
F4	0.3542	0.3581	0.3731	0	0.6945
F5	0.7896	0.7947	0.698	0.6945	0

Table 2. Disorder Matrix, DTLZ(2, 5). Iteration #1

Initial objective sets: 1 2 3 4 5

Redundant Objective Sets:

[F(1), F(2), F(3), F(4)]

[F(5)]

Reduced Objective Set: [F(2), F(5)]

Iteration #2

	F2	F5
F2	0	0.9999
F5	0.9999	0

Table 3. Disorder Matrix, DTLZ(2, 5). Iteration #2

Initial objective sets: 2 5

Redundant Objective Sets:

[F(2)]

[F(5)]

Reduced Objective Set: [F(2), F(5)]

At the end of *objective_reduction_by_inversions* procedure objective F_2 and F_5 are discovered to form the correct Pareto-optimal frontier. Solving this problem with the traditional NSGA-II procedure is very computational intensive and even then the results are not even close to pareto-front.

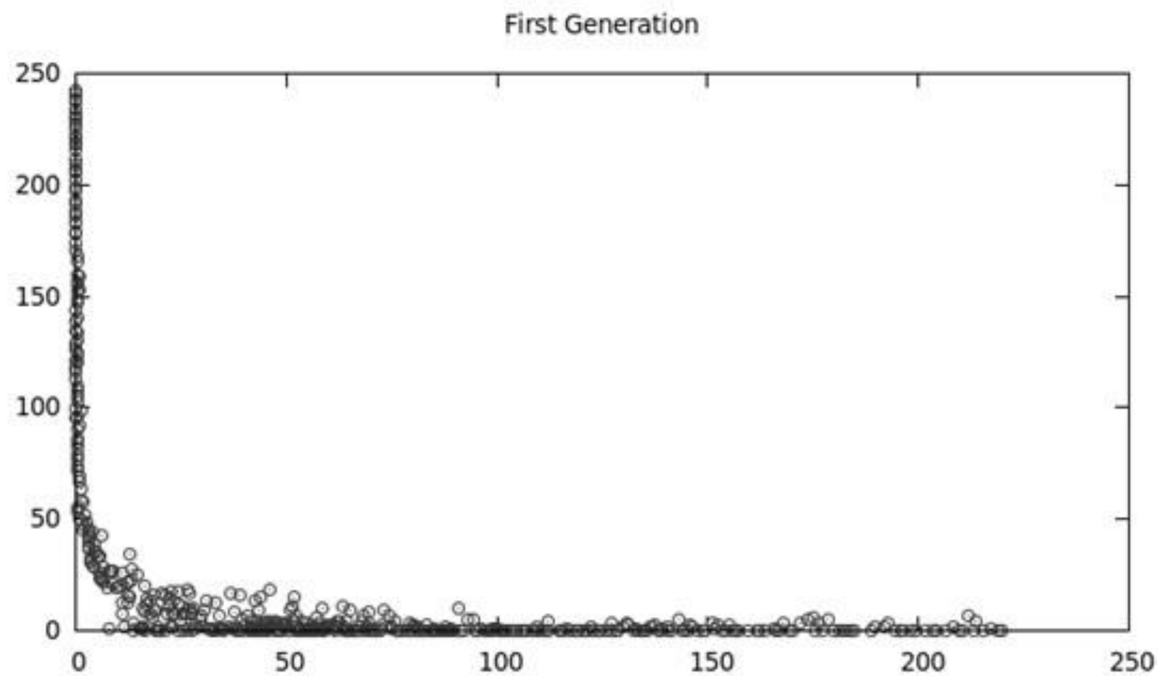

Figure 5. DTLZ(2, 5) at first generation

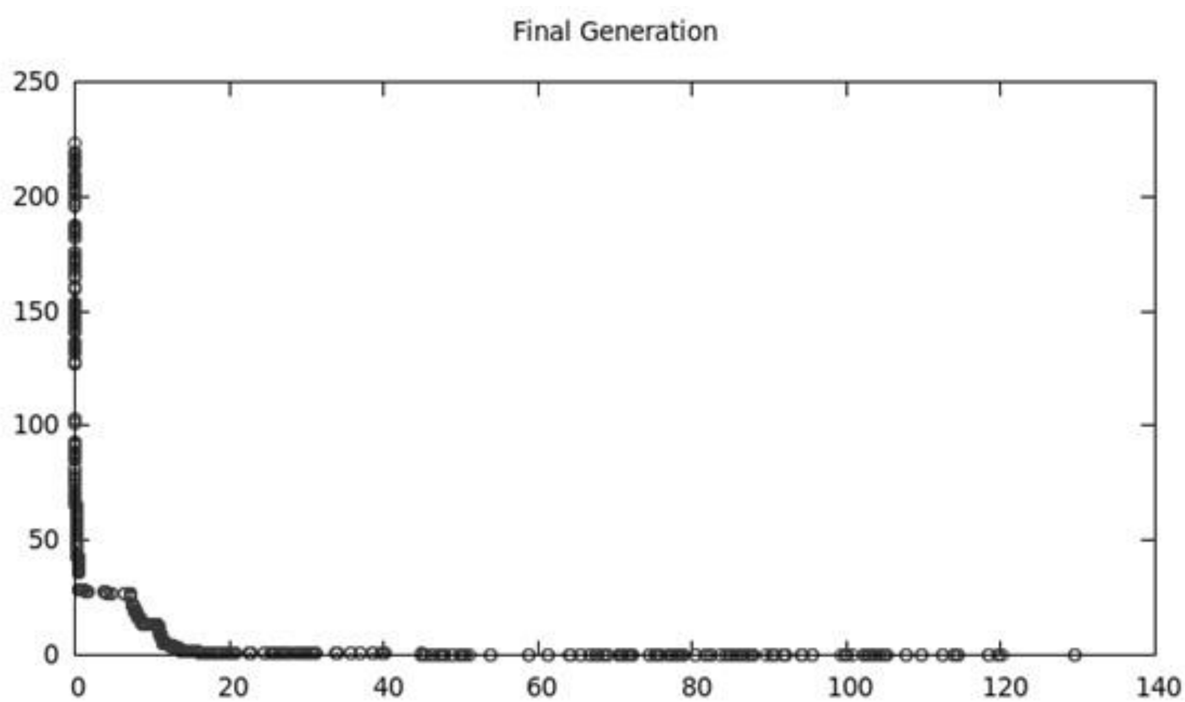

Figure 6. DTLZ(2, 5) at final generation

6.2 Problem DLTZ(3, 5)

This problem has 5 objectives out of which 3 are redundant. The pareto optimal solution can be found with f_5 and any other objective function.

Following are the results of iteration in *objective_reduction_by_inversions*.

Iteration #1

	F1	F2	F3	F4	F5
F1	0	0.3862	0.4284	0.5938	0.5792
F2	0.3862	0	0.4278	0.5936	0.6033
F3	0.4284	0.4278	0	0.5938	0.6127
F4	0.5938	0.5936	0.5938	0	0.5567
F5	0.5792	0.6033	0.6127	0.5567	0

Table 4. Disorder Matrix, DTLZ(3, 5). Iteration #1

Objective sets: 1 2 3 4 5

Redundant Objective Sets:

[F(1), F(2)]

[F(3)]

[F(4)]

[F(5)]

Reduced Objective Set: [F(2), F(3), F(4), F(5)]

Iteration #2

	F2	F3	F4	F5
F2	0	0.3696	0.6063	0.6901
F3	0.3696	0	0.5843	0.6469
F4	0.6063	0.5843	0	0.5978
F5	0.6901	0.6469	0.5978	0

Table 5. Disorder Matrix, DTLZ(3, 5). Iteration #2

Objective sets: 2 3 4 5

Redundant Objective Sets:

[F(2), F(3)]

[F(4)]

[F(5)]

Reduced Objective Set: [F(3), F(4), F(5)]

Iteration #3

	F3	F4	F5
F3	0	0.7149	0.6224
F4	0.7149	0	0.6628
F5	0.6224	0.6628	0

Table 6. Disorder Matrix, DTLZ(3, 5). Iteration #3

Objective sets: 3 4 5

Redundant Objective Sets:

[F(3)]

[F(4)]

[F(5)]

Reduced Objective Set: [F(3), F(4), F(5)]

At the end of *objective_reduction_by_inversions* procedure objective F_3, F_4 and F_5 are discovered to form the correct Pareto-optimal frontier. Solving this problem with the traditional NSGA-II procedure is very computational intensive and even then the results are not even close to pareto-front.

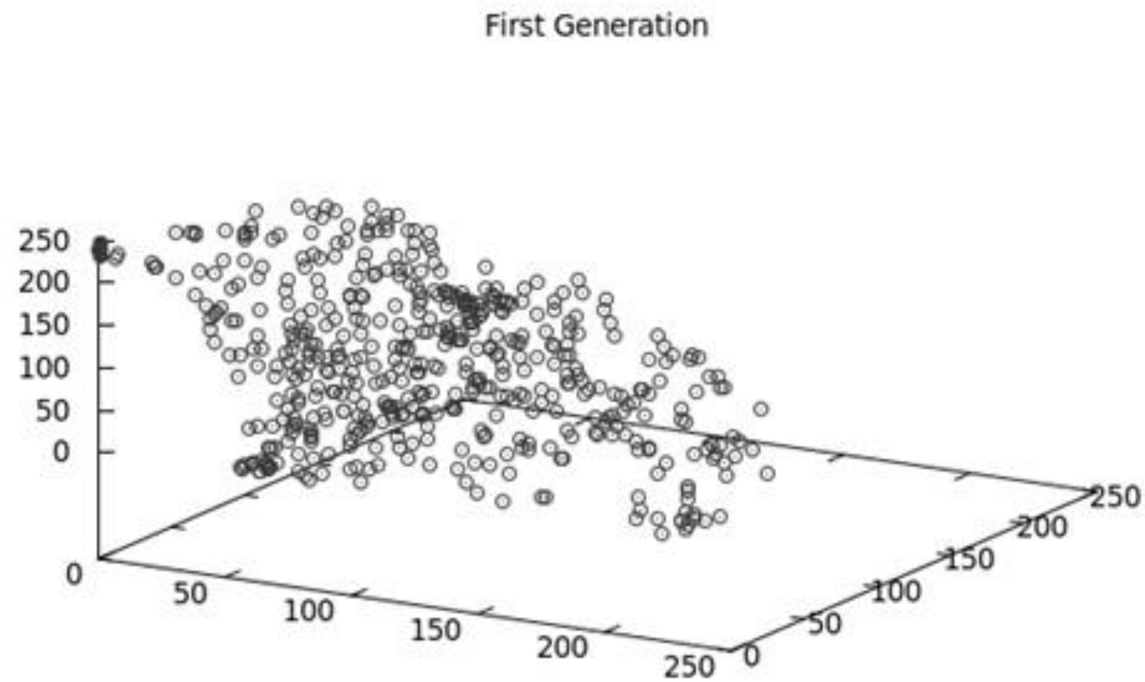

Figure 7. DTLZ(3, 5) at first generation

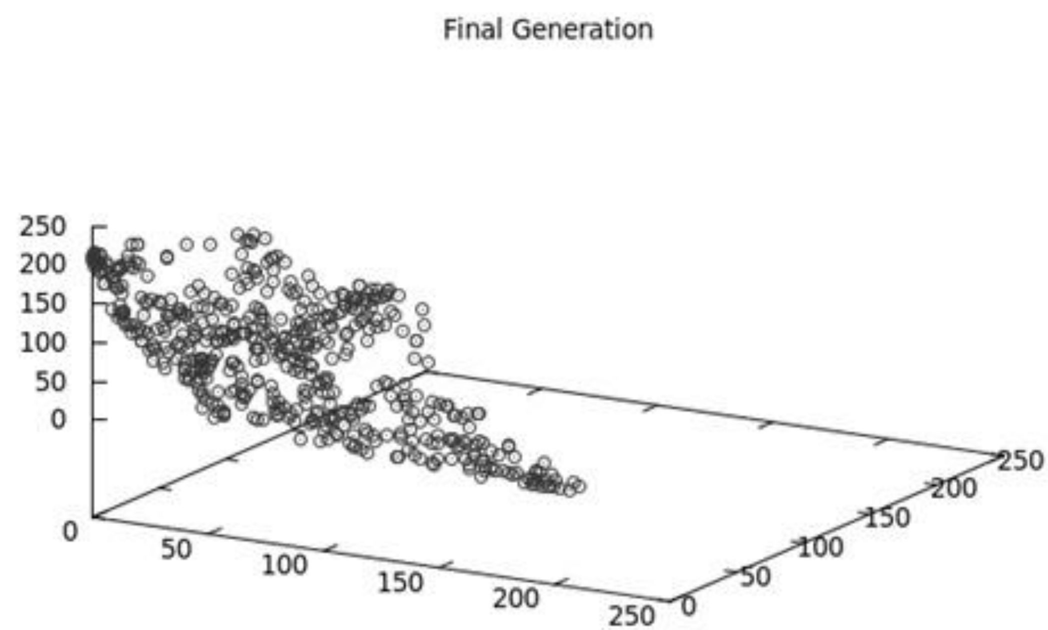

Figure 8. DTLZ(3, 5) at final generation

6.3 Problem DLTZ(2, 10)

This problem has 5 objectives out of which 3 are redundant. The pareto optimal solution can be found with f_5 and any other objective function.
Following are the results of iteration in *objective_reduction_by_inversions*.

Iteration #1

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
F1	0	0.252	0.2705	0.3162	0.3343	0.3827	0.3736	0.4043	0.431	0.6438
F2	0.252	0	0.2963	0.3114	0.3469	0.365	0.3878	0.4313	0.4128	0.6364
F3	0.2705	0.2963	0	0.3003	0.3336	0.3653	0.3813	0.3982	0.4499	0.6479
F4	0.3162	0.3114	0.3003	0	0.3256	0.3698	0.3621	0.4171	0.4334	0.6547
F5	0.3343	0.3469	0.3336	0.3256	0	0.3578	0.3319	0.4092	0.4803	0.6398
F6	0.3827	0.365	0.3653	0.3698	0.3578	0	0.3723	0.4303	0.4285	0.6278
F7	0.3736	0.3878	0.3813	0.3621	0.3319	0.3723	0	0.4057	0.4813	0.6492
F8	0.4043	0.4313	0.3982	0.4171	0.4092	0.4303	0.4057	0	0.4703	0.6
F9	0.431	0.4128	0.4499	0.4334	0.4803	0.4285	0.4813	0.4703	0	0.5997
F10	0.6438	0.6364	0.6479	0.6547	0.6398	0.6278	0.6492	0.6	0.5997	0

Table 7. Disorder Matrix, DTLZ(2, 10). Iteration #1

Objective sets: 1 2 3 4 5 6 7 8 9 10

Redundant Objective Sets:
[F(1), F(2), F(3), F(4), F(5), F(6), F(7), F(8)]
[F(9)]
[F(10)]

Reduced Objective Set: [F(1), F(9), F(10)]

Iteration #2

	F1	F9	F10
F1	0	0.3594	0.892
F9	0.3594	0	0.7486
F10	0.892	0.7486	0

Table 8. Disorder Matrix, DTLZ(2, 10). Iteration #2

Objective sets: 1 9 10

Redundant Objective Sets:

[F(1), F(9)]

[F(10)]

Reduced Objective Set: [F(9), F(10)]

Iteration #3

	F9	F10
F9	0	0.9999
F10	0.9999	0

Table 9. Disorder Matrix, DTLZ(2, 10). Iteration #3

Objective sets: 9 10

Redundant Objective Sets:

[F(9)]

[F(10)]

Reduced Objective Set: [F(9), F(10)]

At the end of *objective_reduction_by_inversions* procedure objective F_9 and F_{10} are discovered to form the correct Pareto-optimal frontier. Solving this problem with the traditional NSGA-II procedure is very computational intensive and even then the results are not even close to pareto-front.

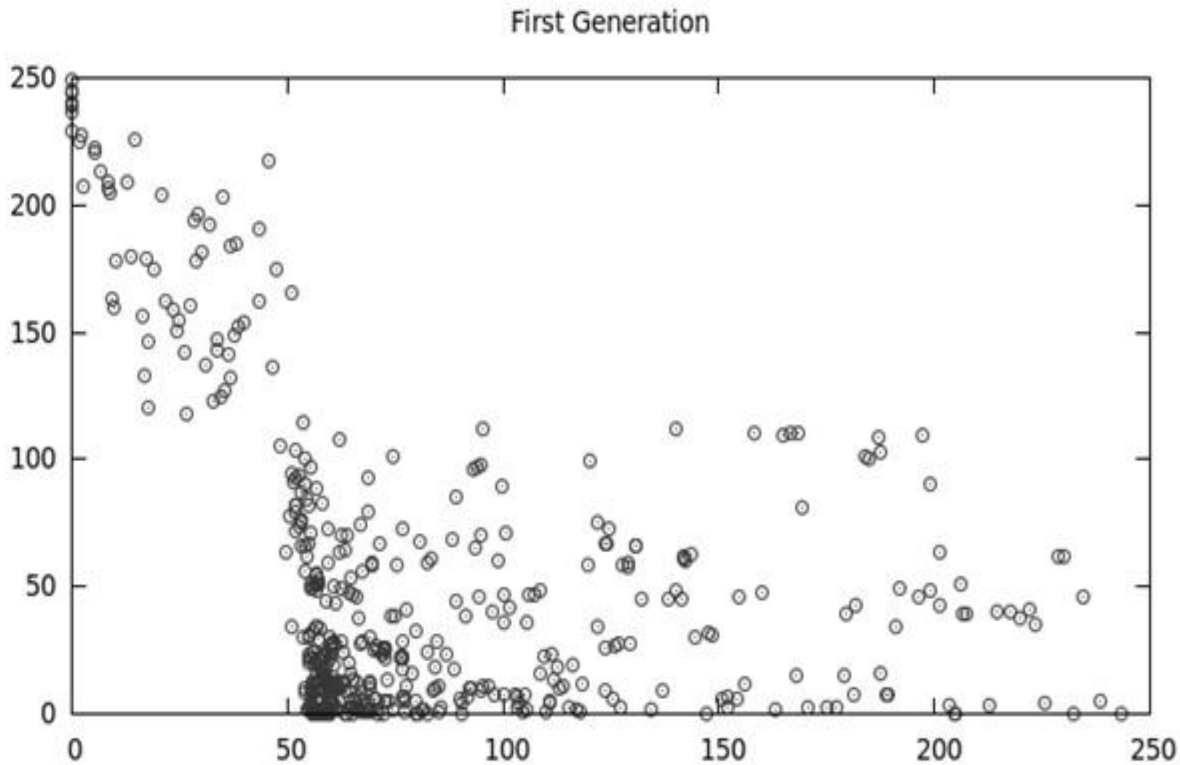

Figure 9. DTLZ(2, 10) at first generation

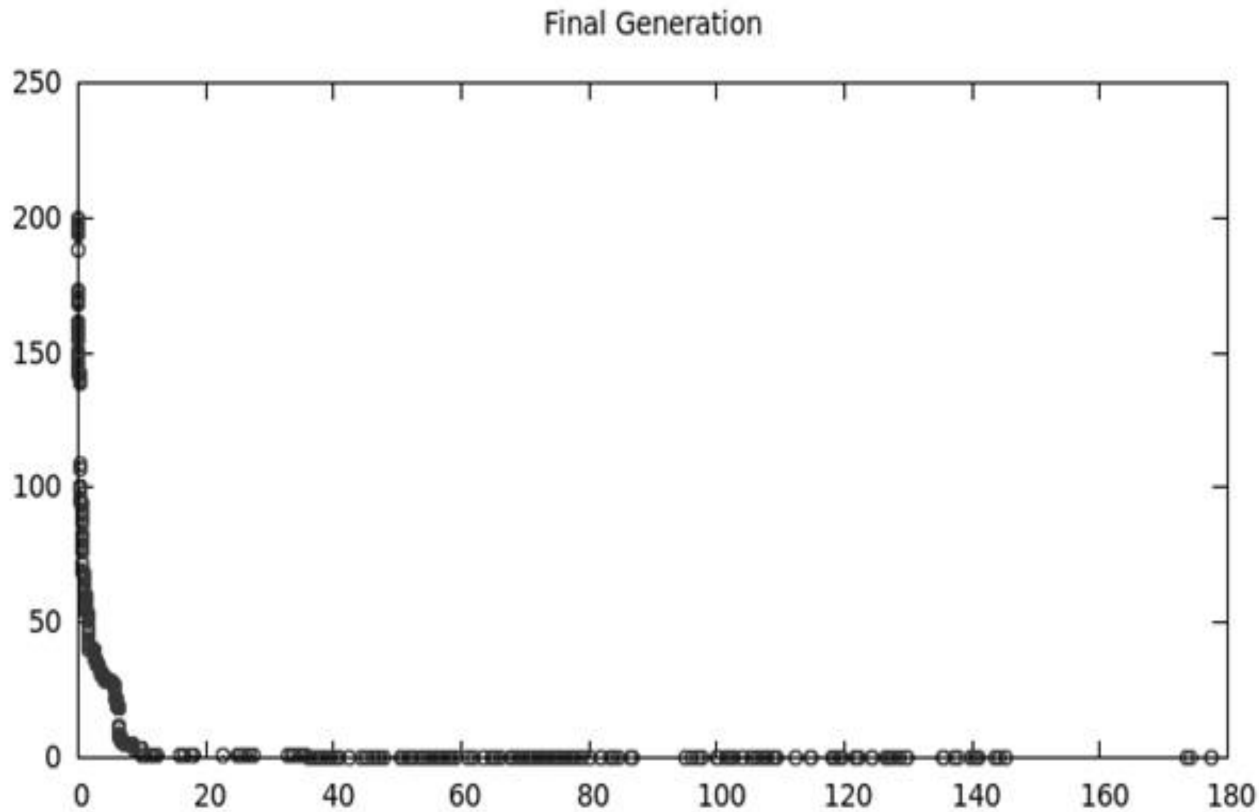

Figure 10. DTLZ(2, 10) at final generation

6.4 Problem DLTZ(3, 10)

This problem has 5 objectives out of which 3 are redundant. The pareto optimal solution can be found with f_5 and any other objective function.

Following are the results of iteration in *objective_reduction_by_inversions*.

Iteration #1

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
F1	0	0.2366	0.3201	0.3563	0.3511	0.3791	0.3896	0.4152	0.5773	0.5865
F2	0.2366	0	0.3226	0.3602	0.352	0.3851	0.3746	0.4155	0.5775	0.6206
F3	0.3201	0.3226	0	0.3614	0.3761	0.3712	0.3913	0.4199	0.5804	0.5903
F4	0.3563	0.3602	0.3614	0	0.3385	0.357	0.3933	0.445	0.5669	0.5909
F5	0.3511	0.352	0.3761	0.3385	0	0.3669	0.3748	0.3999	0.578	0.629
F6	0.3791	0.3851	0.3712	0.357	0.3669	0	0.3738	0.4292	0.5832	0.6019
F7	0.3896	0.3746	0.3913	0.3933	0.3748	0.3738	0	0.4064	0.5775	0.6269
F8	0.4152	0.4155	0.4199	0.445	0.3999	0.4292	0.4064	0	0.6005	0.5903
F9	0.5773	0.5775	0.5804	0.5669	0.578	0.5832	0.5775	0.6005	0	0.4597
F10	0.5865	0.6206	0.5903	0.5909	0.629	0.6019	0.6269	0.5903	0.4597	0

Table 10. Disorder Matrix, DTLZ(3, 10). Iteration #1

Objective sets: 1 2 3 4 5 6 7 8 9 10

Redundant Objective Sets:

[F(1), F(2), F(3), F(4), F(5), F(6), F(7), F(8)]

[F(9)]

[F(10)]

Reduced Objective Set: [F(2), F(9), F(10)]

Iteration #2

	F2	F9	F10
F2	0	0.7468	0.7106
F9	0.7468	0	0.5426
F10	0.7106	0.5426	0

Table 11. Disorder Matrix, DTLZ(3, 10). Iteration #2

Objective sets: 2 9 10

Redundant Objective Sets:

[F(2)]

[F(9)]

[F(10)]

Reduced Objective Set: [F(2), F(9), F(10)]

At the end of *objective_reduction_by_inversions* procedure objective F_2, F_9 and F_{10} are discovered to form the correct Pareto-optimal frontier. Solving this problem with the traditional NSGA-II procedure is very computational intensive and even then the results are not even close to pareto-front.

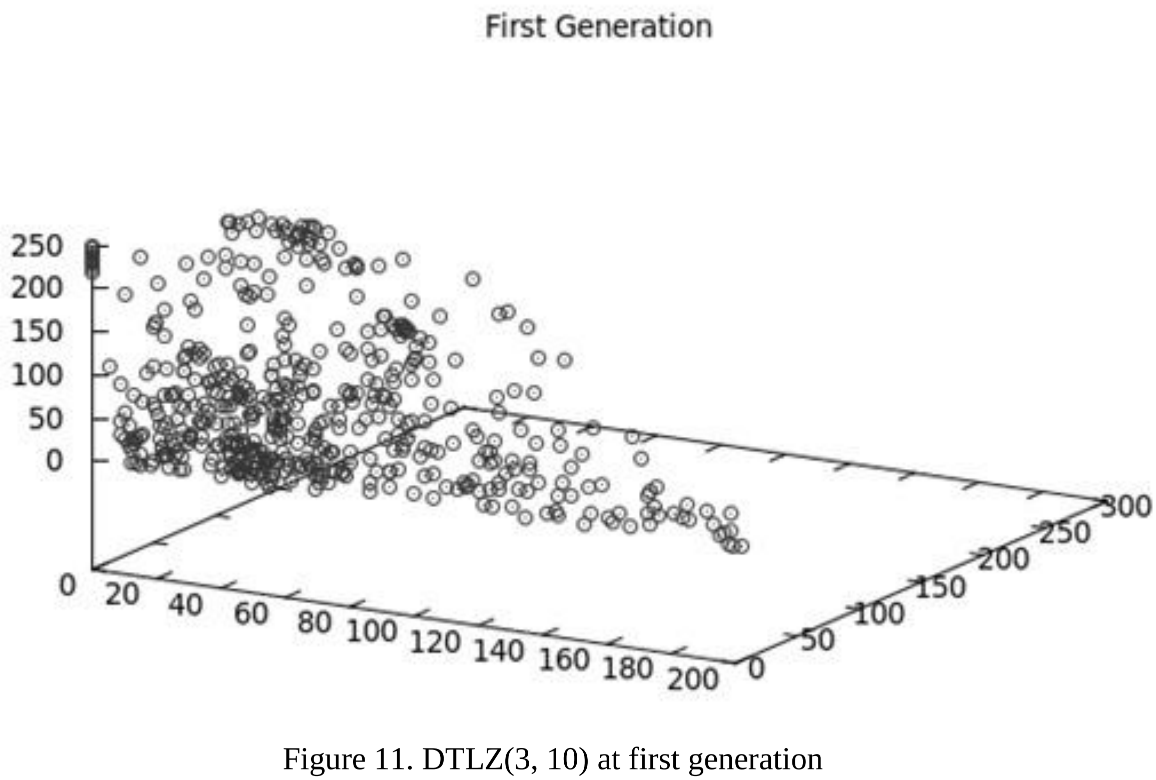

Figure 11. DTLZ(3, 10) at first generation

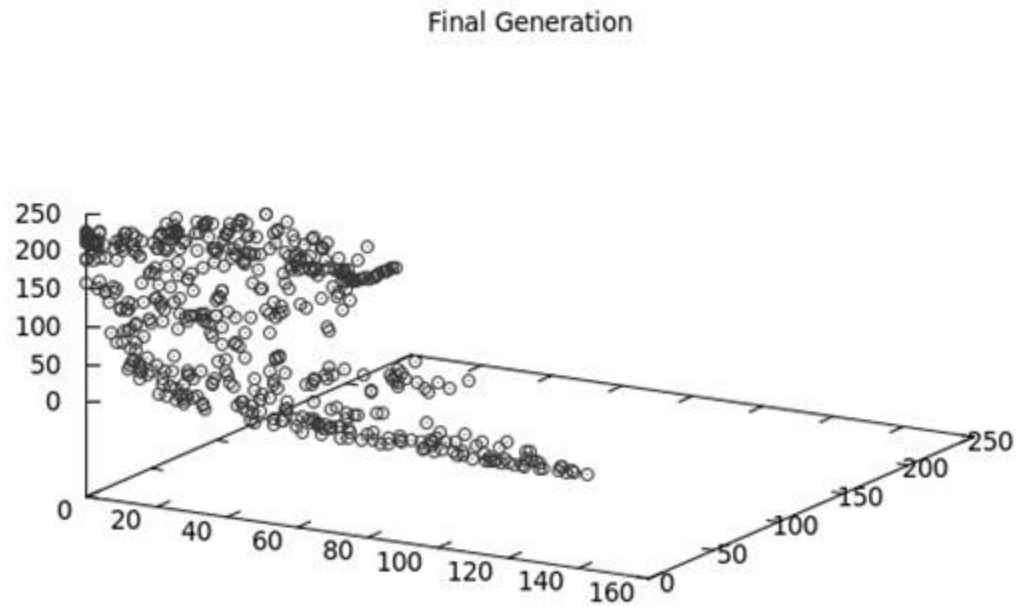

Figure 12. DTLZ(3, 10) at final generation

6.5 Problem DLTZ(5, 10)

This problem has 5 objectives out of which 3 are redundant. The pareto optimal solution can be found with f_5 and any other objective function.

Following are the results of iteration in *objective_reduction_by_inversions*.

Iteration #1

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
F1	0	0.2662	0.3214	0.3116	0.359	0.3968	0.53	0.5364	0.5489	0.543
F2	0.2662	0	0.3595	0.3424	0.3705	0.4057	0.5202	0.5099	0.5248	0.5255
F3	0.3214	0.3595	0	0.3424	0.3457	0.3924	0.518	0.5379	0.5169	0.5482
F4	0.3116	0.3424	0.3424	0	0.3417	0.3789	0.5541	0.562	0.5521	0.5571
F5	0.359	0.3705	0.3457	0.3417	0	0.3589	0.523	0.5436	0.5532	0.5807
F6	0.3968	0.4057	0.3924	0.3789	0.3589	0	0.5314	0.5593	0.5554	0.61
F7	0.53	0.5202	0.518	0.5541	0.523	0.5314	0	0.4446	0.4863	0.4741
F8	0.5364	0.5099	0.5379	0.562	0.5436	0.5593	0.4446	0	0.4756	0.4636
F9	0.5489	0.5248	0.5169	0.5521	0.5532	0.5554	0.4863	0.4756	0	0.4857
F10	0.543	0.5255	0.5482	0.5571	0.5807	0.61	0.4741	0.4636	0.4857	0

Table 12. Disorder Matrix, DTLZ(5, 10). Iteration #1

Objective sets: 1 2 3 4 5 6 7 8 9 10

Redundant Objective Sets:

[F(1), F(2), F(3), F(4), F(5), F(6)]

[F(7)]

[F(8)]

[F(9)]

[F(10)]

Reduced Objective Set: [F(1), F(7), F(8), F(9), F(10)]

Iteration #2

	F1	F7	F8	F9	F10
F1	0	0.4839	0.5234	0.5315	0.5827
F7	0.4839	0	0.5353	0.5379	0.5221
F8	0.5234	0.5353	0	0.5458	0.5536
F9	0.5315	0.5379	0.5458	0	0.5309
F10	0.5827	0.5221	0.5536	0.5309	0

Table 13. Disorder Matrix, DTLZ(5, 10). Iteration #2

Objective sets: 1 7 8 9 10

Redundant Objective Sets:

[F(1)]

[F(7)]

[F(8)]

[F(9)]

[F(10)]

Reduced Objective Set: [F(1), F(7), F(8), F(9), F(10)]

At the end of *objective_reduction_by_inversions* procedure objective F_1, F_7, F_8, F_9 and F_{10} are discovered to form the correct Pareto-optimal frontier. Solving this problem with the traditional NSGA-II procedure is very computational intensive and even then the results are not even close to pareto-front.

6.6 Some other DLTZ(I, M) problems

Apart from the above mentioned test problems we have tested our algorithms on the following problem set, whose results are as follows.

6.6.1 DTLZ(2, 20)

Reduced objective set: [F(7), F(20)]

6.6.2 DTLZ(3, 20)

Reduced objective set: [F(6), F(19), F(20)]

6.6.3 DTLZ(5, 20)

Reduced objective set: [F(6), F(17), F(18), F(19), F(20)]

6.7 Comparative Analysis

We compared the result of the algorithm *objective_reduction_by_inversions* with that of L-PCA[16] discussed by Saxena et al. For each problem set DTLZ(I, M), we are successfully able to group objectives [F(1), F(2)…F(M-I+1)] as redundant objective set. Left over objective functions are responsible for the dimensionality of the pareto front, along with one representative element among the first M-I+1 objective set. The part where the output of *objective_reduction_by_inversions* algorithm differs from L-PCA algorithm is how they choose the representative element from redundant objective set. In *objective_reduction_by_inversions* we considered the distance (sum of disorder from each objective function in redundant objective set) while considering the representative element. While in L-PCA, authors find the objectives which have least contribution in some principal components and reduce them from the original set of objectives.

A. Computational Complexity

While considering the computational complexity of these algorithms, we ignore the computational complexity of obtaining non-dominated solution set like Saxena et al.[16].

For the purpose of calculating the computational complexity, *objective_reduction_by_inversion* algorithm can be evaluated in three following subparts.

Objective_reduction_by_inversion = calculating disorder matrix + grouping redundant objectives + finding representative element from each redundant objective set

The computational analysis will help in finding the overall running time complexity of the algorithm.

- <u>Calculating Disorder Matrix</u>

  For each pair of objectives among M^2 pairs, radix sort can be implemented to sort the objectives in linear time for the evaluation of disorder between them.

  $$O(NM^2)$$

- <u>Grouping Redundant Objectives</u>

  For each pair of redundant objectives among M^2 possible pairs, we merge them in the same group of redundant objectives. Each merge operation, using disjoint set data structure, will have amortized complexity of $\log M$.

  $$O(M^2 \log M)$$

- <u>Representative element from redundant objective set</u>

  For each objective in a redundant objective set of size *M,* we need *M* operations to calculate its evaluate its weight. Therefore its overall running time complexity is:

  $$O(M^2)$$

Therefore overall running time complexity of this algorithm is

$$O(NM^2 + M^2 \log M + M^2)$$

which degenerates to

$$O(NM^2 + M^2 \log M)$$

L-PCA	objective_reduction_by_inversions
$O(NM^2 + M^3)$	$O(NM^2 + M^2 \log N)$

Table 14. Complexity Analysis

Here we used counting sort for sorting the solution set w.r.t objective functions (F_i), disjoint-set data-structure for merging all redundant objectives in a common set, and binary index tree for counting inversions.

B. Number of Iterations Required

In the table below, we provide comparison between the number of iterations required by algorithm proposed here and L-PCA to find the non-redundant objective set for a sample run.

Problem Set	Objective_reduction_by_inversion	L-PCA
DTLZ(2, 5)	2	2
DTLZ(2, 10)	2	4
DTLZ(2, 20)	2	5
DTLZ(3, 5)	2	2
DTLZ(3, 10)	3	3
DTLZ(5, 10)	2	3
DTLZ(5, 20)	2	2

Table 15. Iterations Required

So from the sample run, we can figure out that the algorithm mentioned in this paper requires equal or less number of iterations to reduce the redundant objective set.

We can use three conditions as terminating condition.

- We will set a relatively higher threshold value, ϵ, and run only one iteration of while loop.

- We will run the iterations of while loop until we can't get at least K number of non-redundant objectives.

- We will run the iterations of while loop until there is no further reduction of objectives possible.

While the run time complexity for first case is relatively less than the second one, but it suffers from the drawback that there is a high probability of error propagation in first method.

Second and third method has a relatively higher running time complexity but it will result into a more accurate result as compared to the first method.

Second method suffers from the drawback if the number of non-redundant objectives is more than *K*, then it will also consider even non-redundant objectives as a redundant objective.

We used the concept of dominance relation which is much similar to the concept of ranking dominance. Ranking dominance provides a promising alternative to pareto dominance. However more research is needed to improve the ranking dominance relation.

Also further research is required to deal with the many objectives optimization problems where more than one vector can have same rank with respect to any objective.

7. CONCLUSION

For each problem set DTLZ(I, M), we are successfully able to group objectives [F(1), F(2)…F(M-I+1)] as redundant objective set. Left over objective functions are responsible for the dimensionality of the pareto front, along with one representative element among the first M-I+1 objective set. Thus we can infer that our algorithm is correct in finding the dimensionality of true pareto-optimal surface. As this algorithm has a better execution time complexity as compared to L-PCA algorithm, this can serve as an alternative to it.

We can use three conditions as terminating condition.

- We will set a relatively higher threshold value, ϵ, and run only one iteration of while loop.
- We will run the iterations of while loop until we can't get at least *K* number of non-redundant objectives.
- We will run the iterations of while loop until there is no further reduction of objectives possible.

While the run time complexity for first case is relatively less than the second one, but it suffers from the drawback that there is a high probability of error propagation in first method.

Second and third method has a relatively higher running time complexity but it will result into a more accurate result as compared to the first method.

Second method suffers from the drawback if the number of non-redundant objectives is more than K, then it will also consider even non-redundant objectives as a redundant objective.

We used the concept of dominance relation which is much similar to the concept of ranking dominance. Ranking dominance is a good alternative to pareto dominance. We can further research to improve the ranking dominance relation.

Also further research is required to deal with the many objectives optimization problems where more than one vector can have same rank with respect to any objective.

REFERENCES

[1]. K. Deb, "Multi-Objective Optimization Using Evolutionary Algorithms," John Wiley & Sons, Chichester, 2001.

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGAII,” IEEE Trans. on Evolutionary Computation 6 (2), pp. 182-197, April 2002.

[3] E. J. Hughes, “Evolutionary many-objective optimization: Many once or one many?,” Proc. of 2005 IEEE Congress on Evolutionary Computation, pp. 222-227, Edinburgh, UK, September 2-5, 2005.

[4] H. Ishibuchi, N. Tsukamoto, Y. Nojima, "Behavior of Evolutionary Many-Objective Optimization," Tenth International Conference on Computer Modeling and Simulation, pp. 266-271, April 1-3, 2008.

[5] H. Ishibuchi, N. Tsukamoto, Y. Nojima, "Evolutionary many-objective optimization: A short review," Proc. of 2008 IEEE Congress on Evolutionary Computation, pp. 2419-2426, June 1-6, 2008.

[6] A. Inselberg, "The plane with parallel coordinates," The Visual Computer 1: Number 2, pp 69-91, 1985.

[7] P. J. Fleming, R. C. Purshouse, R. J. Lygoe, "Many-Objective Optimization: An Engineering Design Perspective," Lecture Notes in Computer Science 3410: Evolutionary Multi-Criterion Optimization, pp. 14-32, Springer, Japan, 2007.

[8] L.S. Batista, F. Campelo, F. G. Guimaraes, J. A. Ramirez, "A comparison of dominance criteria in many-objective optimization problems," Evolutionary Computation (CEC), 2011 IEEE Congress on, pp.2359-2366, June 5-8 2011

[9] D. Corne, J Knowles, “Techniques for highly multiobjective optimization: Some non-dominated points are better than others,” Proc. of 2007 Genetic and Evolutionary Computation Conference, pp. 773-780, London, July 7-11 2007.

[10] D. Brockhoff, E. Zitzler, “Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization,” Conference on Parallel Problem Solving from nature (PPSN IX), Volume 4193 of Lecture Notes in Computer Science, pp 533-542, Germany 2006.

[11] Y. J. SHIN, C. H. PARK, "Analysis of Correlation Based Dimention Reduction Methods," International Journal of Applied Mathematics and Computer Science, Volume 21, pp 549-558, Poland, 2011.

[12] C. J. C. Burges, “Dimension Reduction: A Guided Tour,” Foundations and Trends in Machine Learning, Volume 2, No. 4, pp 275-365, USA, 2009.

[13] M. Steyvers, “Multidimensional Scaling,” Encyclopedia of Cognitive Science, USA, 2002.

[14] D. Brockhoff, E. Zitzler, “Objective reduction in evolutionary multiobjective optimization: Theory and applications,” Evolunary Computation, Volume 17 Issue 2, pp 135-165, USA, 2009.

[15] S. Balakrishnama, A. Ganapathiraju, “Linear Discrimant Analysis - A Brief Tutorial,” International Symposium on Information Processing, USA, 1998.

[16]. D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, Q. Zhang, “Objective Reduction in Many-Objective Optimization: Linear and Nonlinear Algorithms,” Evolutionary Computation, IEEE Transactions on Evolutionary Computation, Volume 17, No. 1, pp 77-99, 2013.

[17] S. Kukkonen, J. Lampinen, "Ranking-Dominance and Many-Objective Optimization," Proc. of 2007 IEEE Congress on Evolutionary Computation, pp 3983-3990, Singapore, 2007.

[18] C. Zhou, J. Zheng, K. Li, H. Lv, “Objective Reduction based on the Least Square Method for Large-dimensional Multi-objective Optimization Problem,” Proc. of Fifth International Conference on Natural Computation, Volume 4, pp 350-354, USA, 2009.

[19] A. L. Jaimes, C. A. C. Coello, D. Chakraborty, “Objective Reduction Using a Feature Selection Technique,” Proc. of 10^{th} annual conference on Genetic and Evolutionary Computation, pp 673-680, USA, 2008.

[20] O. Schutze, A. Lara, C. A. C. Coello, “On the Influence of the Number of Objectives on the Hardness of a Multiobjective Optimization Problem,” IEEE Transactions on Evolutionary Computation, Volume 15, No. 4, pp 444-455, 2011.

[21] R. A. Brualdi, “Introductory Combinatorics,” 5th edition, Pearson, 2010.

[22] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, “Scalable test problems for evolutionary multiobjective optimization,” Evolutionary Multiobjective Optimization. Theoretical Advances and Applications, A. Abraham, L. Jain, and R. Goldberg, Springer, pp 105–145, London, 2005.

[23] J. D. Knowles, D. W. Corne, “Approximating the nondominated front using Pareto archived evolution strategy,” Evolutionary Computation, Volume 8, No. 2, pp. 149–172, 2000.

[24] M. Garza-Fabre, G. Toscano-Pulido, C. A. C. Coello, E. Rodriguez-Tello, “Effective Ranking + Speciation = Many-Objective Optimization,” 2011 IEEE Congress on Evolutionary Computation, pp 2115-2122, Los Angeles, 2011.

[25] E. Talbi, S. Mostaghim, T. Okabe, H. Ishibuchi, G. Rudolph, C. A. C. Coello, “Parallel Approaches for Multi-objective Optimization,” Multiobjective Optimization. Interactive and Evolutionary Approaches, J. Branke, K. Deb, K. Miettinen, and R. Slowinski, Eds. Berlin, Germany: Springer. Lecture Notes in Computer Science Vol. 5252, 2008, pp. 349–372, Germany, 2008.

[26] K. Deb, D. K. Saxena, "On finding Pareto-optirnal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," KanGAL Report, No. 2005011, Kanpur Genetic Algorithms Laboratory (KanGAL), Indian Institute of Technology Kanpur, 2005.