INVESTIGATION OF STRENGTH PARETO EVOLUTIONARY ALGORITHM

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Madhumitha Kakarlapudi

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

November 2018

Fargo, North Dakota

North Dakota State University
Graduate School

Title

INVESTIGATION OF STRENGTH PARETO EVOLUTIONARY ALGORITHM

By

Madhumitha Kakarlapudi

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Simone Ludwig
Chair

Dr. Pratap Kotala

Dr. María de los Ángeles Alfonseca-Cubero

Approved:

November 6, 2018
Date

Dr. Kendall Nygard
Department Chair

ABSTRACT

The Strength Pareto Evaluation Algorithm (SPEA) (Zitzler and Thiele 1999) is one of the prominent technique for approximating the pareto-optimal set for the Multiple Objective Optimization (MOO) algorithm. The Strength Pareto Evaluation Algorithm 2 (SPEA2) is an improved version of SPEA that was introduced in the year 2001. SPEA2 in contrast to SPEA incorporates a fine-grained fitness assignment strategy, an improved archive truncation technique, and a density assessment procedure. In this paper, we studied the influence of the optimization ability of SPEA2 on different benchmark functions by evaluating different performance metrics. The benchmark functions used in the paper include 10 constrained functions (CF's) and 10 unconstrained functions (UF's), through which, by varying parameters such as number of iterations, variable size, population and archives, we performed our experiments.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Dr. Simone Ludwig for her continuous support and guidance provided throughout my research. I truly could not have found a better advisor and mentor for my graduate studies.

I would also like to thank Dr. María de los Ángeles Alfonseca-Cubero and Dr. Pratap Kotala for accepting to be in my committee and providing their valuable inputs and encouragement.

DEDICATION

To my parents, in-Laws and husband

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Multiobjective Evolutionary Optimization (MEO) was first introduced in 1984 by Schaffer [1], after which several Pareto-based evolutionary algorithms had been proposed. Among them are Multi-Objective Genetic Algorithm (MOGA) [2], Niched Pareto Genetic Algorithm (NGPA) [3,4], and Non-Dominated Sorting Genetic Algorithm (NSGA) [5] which proved the ability of MEO algorithms to estimate the pair of optimal adjustments in a single optimization run. But the main drawback with these algorithms is that they did not include elitism explicitly. Elitism's importance was recognized when Strength Pareto Evaluation Algorithm (SPEA) [6] and Pareto Archived Evolution Strategy (PAES) [7] were presented. SPEA clearly outperformed the then existing alternative methods under consideration. Later NSGA-II and the Pareto Envelop-based Selection Algorithm (PESA) had proven to outperform SPEA on certain test cases. Later, SPEA2, a modified version of SPEA was introduced to eliminate the potential weaknesses of its antecedent to design a dominant and improved MEO algorithm.

1.1. Differences between SPEA and SPEA2

The main differences between SPEA [7] and SPEA2 [8] are:

1) Improved fitness assignment scheme is used.
2) Search process has become more precise as the nearest neighbor density estimation technique is incorporated in SPEA2.
3) New archive truncation method is presented that assures the protection of boundary solutions.

1.2. Objectives and Outline

The main goal of this paper is to study the optimization ability of SPEA2 on different benchmark functions using a performance metric. The benchmark functions used in the paper include 10 constrained functions (CF’s) [10,11] and 10 unconstrained functions (UF’s) [10], through which, by varying parameters such as number of iterations, variable size, population and archives, we performed our experiments.

The outline of my paper is organized as follows. In Chapter 2, a detailed background on Multiobjective Evolutionary Optimization (MEO) is given. In Chapter 3, the SPEA2 algorithm is described in detail. In Chapter 4, benchmark functions are listed. In Chapter 5, experimental methods and results are discussed. In Chapter 6, conclusions are made.

2. LITERATURE REVEW

Evolutionary Algorithms (EA) in general are stochastic optimization methods that mimic the natural evolution process. Several of the evolutionary algorithms have been proposed since the 1970's, among which are genetic algorithms, evolution strategies and evolutionary methodologies [1]. The main principle of all these methods is that they optimize a set of solutions. Using some approximations, the set of solutions is modified by mainly two principles: selection and variation. The selection mimics the reproduction and variation mimics the natural competence of producing new beings by means of mutation and recombination. Even though the fundamental principles are simple, these algorithms have proven themselves as a general, robust and powerful search mechanism. Moreover, EAs seem to be especially suited to multi-objective optimization because they can capture several pareto-optimal solutions in a single run and may escape likenesses of solutions by recombination.

2.1. Evolutionary Approaches to Multiobjective Optimization

A general MEO problem can be defined as a vector or a functional form *f* that maps a list of *n* parameters (decision variables) to a list of *k* objectives. A mathematical form can be represented as follows:

min/max $$y = f(x) = (f_1(x), f_2(x), \dots, f_k(x))$$

subject to $$x = (x_1, x_2, \dots, x_n) \in X$$
$$y = (y_1, y_2, \dots, y_k) \in Y$$

where x is called the *decision vector* and y is called the *objective vector*.

The set of solutions of a MEO problem contains all the decision vectors that cannot be improved in any objective without degradation in other objectives, such a vector is called as Pareto-optimal. Mathematically it can be represented as follows:

$$\forall_i \in \{1,2,\ldots,n\}: \ f_i(a) \geq f_i(b) \ \cap \ \exists_j \in \{1,2,\ldots,n\}: \ f_j(a) > f_j(b)$$

where, a and b are the two decision vectors belonging to X. We can say that a dominated b if and only if above equation is valid. All those decision vectors that are not dominated are called as nondominated or Pareto-optimal and the set of solutions that are formed is denoted as Pareto-optimal set or front. It denotes the tradeoff surface with respect to the n objectives.

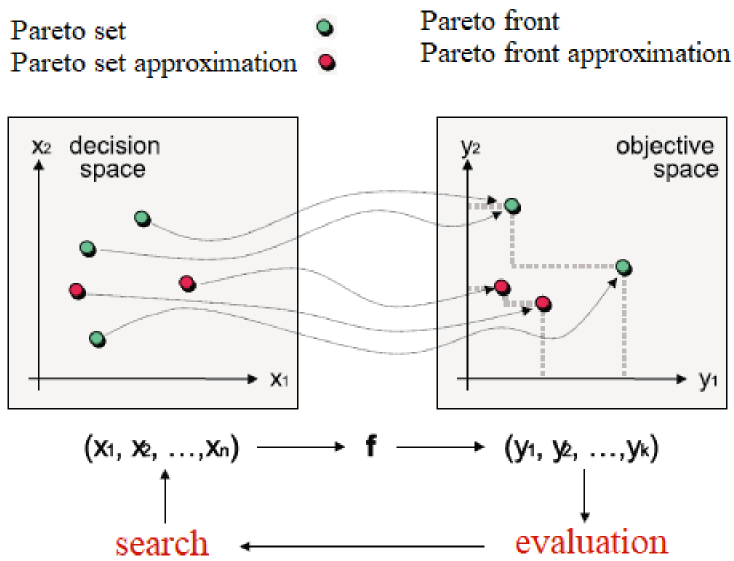

Fig 2.1. Illustration of a general Multiobjective optimization problem [9]

Most of the MEOs have concentrated on this approximation of the Pareto set. Accordingly, the outcome of these algorithms is considered to be a set of mutually nondominated solutions, in short, called as Pareto set approximation.

2.1.1. Plain Aggregating Approach

EA can be applied to problems where individual objectives are combined or aggregated to form a scalar function. Moreover, combining the objectives has the advantage of producing a single optimized solution, requiring no further interaction with the decision maker. Several applications of such approaches have been reported in literature. For example, the use of the popular weighted sum approach by Jakob et. al., 1992 [12]. Likewise handling constraints with penalty functions is another example where the functions are problem-dependent, the method developed by Richardson et. al., 1989 [13].

2.1.2. Population Based Non-Pareto Approach

In a single EA run, the possibility of exploiting populations of multiple non-dominated solutions concurrently was recognized for the first time by Schaffer and Grefenstette (1985) [14]. Their approach was named as Vector Evaluated Genetic Algorithm (VEGA), in which sub-populations of next generations were selected from the pool of old generations separately, based on the objectives. After shuffling these sub-populations, crossover and mutation were applied, the non-dominated individuals were then identified by monitoring the population.

Fourman (1985) also introduced non-aggregating population based MEO [15]. In his method, selection was performed by comparing pairs of individuals, each pair according to one of the objectives. This is also practiced by Ben-Tal (1980) by the name lexicographic ordering [16]. Objectives were assigned different priorities at first by the user and then individuals were compared according to the objective with high priority. If this results in a tie, the objective with the second highest priority was used and so on.

Similarly, Hajela and Lin (1992) also exploited population based non-pareto approach based on the weighted sum method by explicitly including the weights in the chromosome and promoting their diversity in the population through fitness sharing [17].

2.1.3. Pareto Based Approach

All the methods discussed in the previous section like Schaffer, Fourman, Ben and Hajela and Lin [14-17] promote the generation of multiple non-dominated solutions. But, none of them use the actual definition of pareto-optimality. The first proposed pareto based fitness assignment was made by Goldberg (1989) [18]. It involved assigning rank 1 to the non-dominated individuals and removing them from contention, then finding a new set of non-dominated individuals by ranking them 2 and so forth.

Later in 1993, Fonsea and Fleming [19] have proposed a slightly different approach, when an individual's rank is corresponding to the number of individual in the current population by which it is dominated. Therefore, the non-dominated individuals are all ranked the same, while the dominated ones are penalized according to the population density in the corresponding region of the trade-off surface.

Similarly, Sinivas and Deb (1994) [5] have proposed a pareto based approach identical to the Goldenberg's version of ranking population. Additionally, they have provided a means of evolving only a given region of the trade-off surface. During the search, goal values were changed that alter the fitness landscape accordingly and allows the decision maker to direct the population.

Several other pareto-based approaches have been proposed similar to the ones discussed above. However, none of these methods did incorporate the concept of elitism explicitly. Few years later, the importance of elitism in multiobjective search was recognized and experimentally supported by Parks and Miller (1998) [20] and Zitzler, Deb and Thiele (2000) [22]. Among them are the SPEA that was introduced by Zitzler and Thiele (1998) [21] and Zitzler, Deb and Thiele (2000) [22] and PAES by Knowles and Corne (1999) [7].

2.2. The Strength Pareto Evolution Algorithm

As SPEA is the basis for SPEA2, in this section, a brief overview of SPEA is described. SPEA uses a mixture of established and new techniques to approximate Pareto-optimal solution set. The following are the basic techniques, on which SPEA is based:

- It stores the individuals externally that represent a non-dominated front among the other solutions that are under consideration.
- It uses the principle of Pareto dominance to assign a scalar fitness value to individual.
- To reduce the number of individuals that are externally stored, clustering is performed without destroying the characteristics of the trade-off surface.
- The above three techniques are combined in a single algorithm.
- Although the populations dominate each other, the fitness assignment of a population number is determined only from the external set of individuals.
- All the external set of individuals participate in selection.
- To preserve the diversity in the population, a new Pareto based niching method is introduced.

The following steps are the flowchart of the SPEA algorithm.

Input: *N (population size)*

$\bar{N}$ *(maximum size of external set)*

T (maximum number of generations)

p_c *(crossover probability)*

p_m *(mutation rate)*

Output: ***A*** *(nondominated set)*

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

Step 2: ***Update of external set:*** *Set the temporary external set* $\bar{P}' = \bar{P}_t$

a) *Copy individuals whose decision vectors are nondominated regarding* $m(P_t)$ *to* $\bar{P}'$: $\bar{P}' = \bar{P}' + \{i \mid i \in P_t \wedge m(i) \in p(m(P_t))\}$

b) *Remove the individuals from $\bar{P}'$ whose corresponding decision vectors are weakly dominated regarding m($\bar{P}'$), i.e., as long as there exists a pair (i , j) with i,j ∈ $\bar{P}'$ and m(i) ≥ m(j) do $\bar{P}' = \bar{P}' - \{j\}$.*

c) *Reduce the number of individuals externally stored by means of clustering, i.e., call Clustering Algorithm with parameters $\bar{P}'$ and $\bar{N}$, and assign the resulting reduced set to $\bar{P}_{t+1}$.*

Step 3: ***Fitness assignment:*** *Calculate fitness values of individuals in P_t and $\bar{P}_t$ by invoking Fitness Assignment Algorithm (see below).*

Step 4: ***Selection:*** *Set $P' = Ø$. For $i = 1, \ldots, N$ do*

a) *Select two individuals i,j ∈ $P_t + \bar{P}_t$ at random.*

b) *If F(i) < F(j) then $P' = P' + \{i\}$ else $P' = P' + \{j\}$. Note that fitness is to be minimized here.*

Step 5: ***Recombination:*** *Set $P'' = Ø$. For $i = 1, \ldots, \frac{N}{2}$ do*

a) *Choose two individuals i,j ∈ P' and remove them from P'.*

b) *Recombine i and j. The resulting children are k, l ∈ I*

c) *Add k, l to P'' with probability p_c. Otherwise add i, j to P''*

Step 6: ***Mutation:*** *Set $P'' = Ø$. For each individual i ∈ P'' do*

a) *Mutate i with mutation rate p_m. The resulting individual is j ∈ I*

b) *Set $P''' = P''' + \{j\}$*

Step 7: ***Termination:*** *Set $P_{t+1} = P'''$ and t = t+1. If t ≥ T or another stopping criterion is satisfied then set **A** = p(m($\bar{P}_t$)) else go to Step 2.*

In Step 2, the external set $\bar{P}$ is updated and reduced if its maximum size $\bar{N}$ is overstepped. In Step 3, the individuals in $\bar{P}$ and P are evaluated interdependently from each other and assigned fitness values. In Step 4, the selection phase is made where individuals from $\bar{P}$ + P (union of population and external set) are selected in order to fill the mating pool, in which binary tournament selection with replacement is used. Finally, the recombination and mutation processes are applied as usual. Fitness assignment and clustering are described next:

2.2.1. Fitness Assignment

The fitness assignment is performed as follows:

Input: P_t *(population)*

$\bar{P}_t$ *(external set)*

Output: *F (fitness values)*

Step 1: *Each individual i ∈* $\bar{P}_t$ *is assigned a real value S(i) ∈ [0,1), called strength; s(i) is proportional to the number of population members*

$$S(i) = \frac{|\{j \mid j \in P_t \cap m(i) \geq m(j)\}|}{N+1}$$

The fitness of i is equal to its strength: F(i) = S(i)

Step 2: *The fitness of an individual j ∈* P_t *is calculated by summing the strengths of all externally stored individuals i ∈* $\bar{P}_t$ *whose decision vectors weakly dominate m(j). We add one to the total in order to guarantee that members of* $\bar{P}_t$ *have better fitness than members of* P_t *(note that fitness is to be minimized here, i.e., small fitness values correspond to high reproduction probabilities):*

$$F(j) = 1 + \sum_{i \in \overline{P_t}, m(i) \geq m(j)} S(i) \; where \; F(j) \in [1, N)$$

This mechanism intuitively reflects the idea of preferring individuals near the Pareto-optimal front and distributing them at the same time along the trade-off surface. The main difference with fitness sharing is that niches are not defined in terms of distance but Pareto dominance. This renders the setting of a distance parameter superfluous, although the parameter $\bar{N}$ influences the niching capability, as will be discussed in the next section.

2.2.2. Clustering Procedure

In certain problems, the Pareto-optimal set can be extremely large or even contain an infinite number of solutions. However, from the DM's point of view, presenting all nondominated solutions found is useless when their number exceeds reasonable bounds. Moreover, the size of the external set influences the behavior of SPEA.

A method that has been applied to this problem successfully and studied extensively in the same context is cluster analysis, and was initially introduced by Morse (1980). In general, cluster analysis partitions a collection of p elements into q groups of relatively homogeneous elements, where q < p. The average linkage method, a clustering approach that has proven to perform well on this problem (Morse 1980), has been chosen here.

Input: $\bar{P}'$ *(external set)*

$\bar{N}$ *(maximum size of external set)*

Output: $\bar{P}_{t+1}$ *(updated external set)*

Step 1: *Initialize cluster set C; each individual i* ∈ $\bar{P}'$ *constitutes a distinct cluster:*

$C = \bigcup_{i \in \bar{P}'} \{\{i\}\}$

Step 2: *If* $|C| \leq \bar{N}$*, go to step 5, else go to step 3*

Step 3: *Calculate the distance of all possible pairs of clusters. The distance d_c of two clusters C_1 and $C_2 \in C$ is given as the average distance between pairs of individuals across the two clusters*

$$d_c = \frac{1}{|C_1||C_2|} . \sum_{i_1 \in C_1, i_2 \in C_2} d(i_1, i_2)$$

where the function d reflects the distance between two individuals i_1 and i_2 (here the distance in objective space is used).

Step 4: *Calculate Determine two clusters C_1 and C_2 with minimal distance dc; the chosen clusters merge into larger cluster: C = C \ $\{C_1, C_2\} \cup \{C_1 \cup C_2\}$. Go to Step 2.*

Step 5: *Per cluster, select a representative individual and remove all other individuals from the cluster. We consider the centroid (the point with minimal average distance to all other points in the cluster) as the representative individual. Compute the reduced nondominated set by uniting the representatives of the clusters: $\bar{P}_{t+1} = \bigcup_{c \in C} c$.*

2.2.3. Elitist Multiobjective Evolutionary Algorithm

The elitism mechanism used in SPEA can be generalized for incorporation in arbitrary multiobjective evolution algorithm (MOEA) implementations. The only difference is that the population and the external set are already united before (and not after) the fitness assignment phase. This guarantees that any fitness assignment scheme can be used in combination with this elitism variant.

Input: *N (population size)*

$\bar{N}$ (maximum size of external set)

T (maximum number of generations)

p_c (crossover probability)

p_m (mutation rate)

Output: ***A*** *(nondominated set)*

Step 1: ***Initialization:*** *Set $\bar{P}_0$ = Ø. Set t = 0. Initialize P_0*

Step 2: ***Update of external set:*** *Set the temporary external set $\bar{P}' = \bar{P}_t$*

a) *Copy individuals whose decision vectors are nondominated regarding $m(P_t)$ to $\bar{P}'$: $\bar{P}' = \bar{P}' + \{i \mid i \in P_t \wedge m(i) \in p(m(P_t))\}$*

b) *Remove the individuals from $\bar{P}'$ whose corresponding decision vectors are weakly dominated regarding $m(\bar{P}')$, i.e., as long as there exists a pair (i , j) with $i,j \in \bar{P}'$ and $m(i) \geq m(j)$ do $\bar{P}' = \bar{P}' - \{j\}$.*

c) *Reduce the number of individuals externally stored by means of clustering, i.e., call Clustering Algorithm with parameters $\bar{P}'$ and $\bar{N}$, and assign the resulting reduced set to $\bar{P}_{t+1}$.*

Step 3: ***Elitism:*** *Set $\bar{P}_t = P_t + \bar{P}_t$*

Step 4: ***Fitness assignment:*** *...*

Step 5: ***Selection:*** *...*

Step 6: ***Recombination:*** *. . .*

Step 7: ***Mutation:*** *. . .*

Step 8: ***Termination:*** *Set $P_{t+1} = P'''$ and t = t+1. If $t \geq T$ or another stopping criterion is satisfied then set **A** = $p(m(\bar{P}_t))$ else go to Step 2.*

2.2.4. Potential Weaknesses of SPEA

Although, SPEA performed well in different comparative studies (Zitzler and Thiele 1999; Zitzler, Deb, and Thiele 2000), there is still room for improvement as recent studies (Corne, Knowles, and Oates 2000; Deb, Agrawal, Pratap, and Meyarivan 2000) have shown. Particularly, the following issues are identified as the potential weaknesses of SPEA:

- **Fitness Assignment:** The dominating individuals of the same archive have matching fitness values. In other words, in the case when the archive contains only a single individual, all population members have the same rank irrespective of whether they dominate each other or not. Due to this, the selection pressure is reduced substantially and in this particular case SPEA behaves almost like some random search algorithm.
- **Density Estimation:** If most of the individuals of the current generation are identical, i.e., do not dominate each other, none or very little information can be obtained based on the partial order defined by the dominance relation. Particularly, in this situation, that is very likely to occur in presence of more than two objectives, density information must be used to guide the search more efficiently. In such a case, clustering will be of great use, but not to the population and only with regard to the archive.
- **Archive Truncation:** The clustering technique used in SPEA can be used to reduce the nondominated set without terminating its features, but it may miss external solutions. Nevertheless, these solutions should be kept in the archive in order to obtain a decent spread of nondominated solutions.

In the next chapter, the SPEA2 algorithm, which is an improved version of SPEA is detailed, which was designed to overcome the aforementioned issues.

3. THE SPEA2 ALGORITHM

In distinction to SPEA, SPEA2 uses a fine-grained fitness assignment approach which implements density information that will be described in the following sections. Additionally, the archive size is fixed, that means, whenever the number of nondominated individuals is fewer than the predefined archive size, the archive is filled up by dominated individuals; with SPEA, the archive size may vary over time. Moreover, the clustering technique, which is raised when the nondominated front surpasses the archive limit, has been replaced by an alternative truncation method which has similar characteristics but does not loose boundary points. Lastly, another difference between SPEA and SPEA2 is that in SPEA2, only archive members contribute in the mating selection method.

The flow of the SPEA2 algorithm is as follows:

Input: *N (population size)*

$\bar{N}$ (maximum size of external set)

T (maximum number of generations)

Output: ***A*** *(nondominated set)*

Step 1: ***Initialization:*** *Generate an initial population P_0 and create the empty external set $\bar{P}_0$ = Ø. Set $t = 0$.*

Step 2: ***Fitness assignment:*** *Calculate fitness values of individuals in P_t and $\bar{P}_t$ by invoking Fitness Assignment Algorithm.*

Step 3: ***Environmental Selection:*** *Copy all non-dominated individuals in P_t and $\bar{P}_t$ to $\bar{P}_{t+1}$. If size of $\bar{P}_{t+1}$ exceed $\bar{N}$ then reduce $\bar{P}_{t+1}$ by mean of the truncation operator; otherwise if size of $\bar{P}_{t+1}$ is less than $\bar{N}$ then fill $\bar{P}_{t+1}$ with dominated individuals in P_t and $\bar{P}_t$.*

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 $\bar{P}_{t+1}$. Stop.*

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

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

3.1. SPEA2 Fitness Assignment

In contrast to SPEA, in SPEA2 both the dominating and dominated solutions are taken into consideration to avoid the situation where the same archive members can have identical fitness values. Here, each individual *i* in the archive A_t and Population P_t is assigned a strength value S(i). Also at each S value, the raw fitness R(i) was determined. Additionally, the density information is implemented to discriminate between individuals having identical raw fitness values. This technique of incorporating density information was adapted from the k-th nearest neighbor method.

The run time of the fitness assignment method is dominated by density estimator ($O(L^2 \log L)$). Moreover, the S and R values are calculated at $O(L^2)$ complexity, where L = M+N.

3.2. SPEA2 Environment Selection

Update of the archive is operated differently in SPEA2 when compared to SPEA. In case of SPEA2, over time, the numbers of individuals contained in the archive become constant and also the truncation method prevents the boundary solutions being removed.

4. BENCHMARK FUNCTIONS

In this paper, a set of 10 unconstrained (bound constrained) and 10 constrained multiobjective optimization test instances were used to perform optimization.

4.1. Unconstrained Multiobjective Test Problems

4.1.1. Unconstrained Problem 1

The two objectives to be minimized in this problem are:

$$f_1 = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left[x_j - \sin(6\pi x_1 + \frac{j\pi}{n})\right]^2$$

$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left[x_j - \sin(6\pi x_1 + \frac{j\pi}{n})\right]^2$$

where $J_1 = \{j | j \text{ is odd and } 2 \leq j \leq n\}$ and $J_2 = \{j | j \text{ is even and } 2 \leq j \leq n\}$.

The search space is $[0,1] \times [-1,1]^{n-1}$

Its Pareto Front (PF) is

$$f_2 = 1 - \sqrt{f_1}, \quad 0 \leq f_1 \leq 1$$

Its Pareto Set (PS) is

$$x_j = \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), \; j = 2, \dots, n, \quad 0 \leq x_1 \leq 1$$

4.1.2. Unconstrained Problem 2

The two objectives to be minimized in this problem are:

$$f_1 = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left[y_j\right]^2$$

$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left[y_j\right]^2$$

where $J_1 = \{j|j$ is odd and $2 \le j \le n\}$ and $J_2 = \{j|j$ is even and $2 \le j \le n\}$ and

$$y_j = \begin{cases} x_j - \left[0.3\, x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right] \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_1 \\ x_j - \left[0.3\, x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right] \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_2 \end{cases}$$

The search space is $[0,1] \times [-1,1]^{n-1}$

Its PF is

$$f_2 = 1 - \sqrt{f_1}, \quad 0 \le f_1 \le 1$$

Its PS is

$$x_j = \begin{cases} \left[0.3\, x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right] \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_1 \\ \left[0.3\, x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right] \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_2 \end{cases}$$

$$0 \le x_1 \le 1$$

4.1.3. Unconstrained Problem 3

The two objectives to be minimized in this problem are:

$$f_1 = x_1 + \frac{2}{|J_1|} \left(4 \sum_{j \in J_1} [y_j]^2 - 2 \prod_{j \in J_1} \cos\left(\frac{20 y_j \pi}{\sqrt{y}}\right) + 2\right)$$

$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|} \left(4 \sum_{j \in J_2} [y_j]^2 - 2 \prod_{j \in J_2} \cos\left(\frac{20 y_j \pi}{\sqrt{y}}\right) + 2\right)$$

where $J_1 = \{j|j$ is odd and $2 \le j \le n\}$ and $J_2 = \{j|j$ is even and $2 \le j \le n\}$ and

$$y_j = \begin{cases} x_j - \left[0.3\, x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right] \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_1 \\ x_j - \left[0.3\, x_1^2 \cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right] \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), & j \in J_2 \end{cases}$$

$$y_j = x_j - x_1^{0.5\left(1.0 + \frac{3(j-2)}{n-2}\right)}, \quad j = 2, \ldots, n,$$

The search space is $[0,1]^n$

Its PF is

$$f_2 = 1 - \sqrt{f_1}, \quad 0 \le f_1 \le 1$$

Its PS is

$$x_j = x_1^{0.5\left(1.0 + \frac{3(j-2)}{n-2}\right)}, \; j = 2, \ldots, n, \quad 0 \le x_1 \le 1$$

4.1.4. Unconstrained Problem 4

The two objectives to be minimized in this problem are:

$$f_1 = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} h(y_j)$$

$$f_2 = 1 - x_1^2 + \frac{2}{|J_2|} \sum_{j \in J_2} h(y_j)$$

where $J_1 = \{j | j \text{ is odd and } 2 \le j \le n\}$ and $J_2 = \{j | j \text{ is even and } 2 \le j \le n\}$ and

$$y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), \; j = 2, \ldots, n,$$

and

$$h(t) = \frac{|t|}{1 + e^{2|t|}}$$

The search space is $[0,1] \times [-2,2]^{n-1}$

Its PF is

$$f_2 = 1 - f_1^2, \quad 0 \le f_1 \le 1$$

Its PS is

$$x_j = \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), \; j = 2, \ldots, n.$$

$$0 \le x_1 \le 1$$

4.1.5. Unconstrained Problem 5

The two objectives to be minimized in this problem are:

$$f_1 = x_1 + \left(\frac{1}{2N} + \varepsilon\right)|\sin(2N\pi x_1)| + \frac{2}{|J_1|}\sum_{j\in J_1} h(y_j)$$

$$f_2 = 1 - x_1 + \left(\frac{1}{2N} + \varepsilon\right)|\sin(2N\pi x_1)| + \frac{2}{|J_2|}\sum_{j\in J_2} h(y_j)$$

where $J_1 = \{j|j \text{ is odd and } 2 \le j \le n\}$ and $J_2 = \{j|j \text{ is even and } 2 \le j \le n\}$. N is an integer, $\varepsilon > 0$,

$$y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right),\ j = 2, \dots, n,$$

and

$$h(t) = 2t^2 - \cos(4\pi t) + 1$$

The search space is $[0,1] \times [-1,1]^{n-1}$

Its PF has 2N+1 Pareto Optimal solutions: $\left(\frac{i}{2N}, 1 - \frac{i}{2N}\right)$, for $i = 0,1, \dots, 2N$.

4.1.6. Unconstrained Problem 6

The two objectives to be minimized in this problem are:

$$f_1 = x_1 + \max\{0, 2\left(\frac{1}{2N} + \varepsilon\right)\sin(2N\pi x_1)\} + \frac{2}{|J_1|}\left(4\sum_{j\in J_1}[y_j]^2 - 2\prod_{j\in J_1}\cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right)$$

$$f_2 = 1 - x_1 + \max\{0, 2\left(\frac{1}{2N} + \varepsilon\right)\sin(2N\pi x_1)\} + \frac{2}{|J_2|}\left(4\sum_{j\in J_2}[y_j]^2 - 2\prod_{j\in J_2}\cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2\right)$$

where $J_1 = \{j|j \text{ is odd and } 2 \le j \le n\}$ and $J_2 = \{j|j \text{ is even and } 2 \le j \le n\}$. N is an integer, $\varepsilon > 0$,

$$y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right),\ j = 2, \dots, n,$$

The search space is $[0,1] \times [-1,1]^{n-1}$

Its PF consists of

- One isolated point, (0,1), and
- N disconnected pairs:

$$f_2 = 1 - f_1, f_1 \in \bigcup_{i=1}^{N} \left[\frac{2i-1}{2N}, \frac{2i}{2N}\right].$$

4.1.7. Unconstrained Problem 7

The two objectives to be minimized in this problem are:

$$f_1 = \sqrt[5]{x_1} + \frac{2}{|J_1|} \sum_{j \in J_1} {y_j}^2$$

$$f_2 = 1 - \sqrt[5]{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} {y_j}^2$$

where $J_1 = \{j | j$ is odd and $2 \le j \le n\}$ and $J_2 = \{j | j$ is even and $2 \le j \le n\}$ and

$$y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, \dots, n$$

The search space is $[0,1] \times [-1,1]^{n-1}$

Its PF is

$$f_2 = 1 - f_1, \quad 0 \le f_1 \le 1$$

Its PS is

$$x_j = \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = 2, \dots, n. \quad 0 \le x_1 \le 1$$

4.1.8. Unconstrained Problem 8

The three objectives to be minimized in this problem are:

$$f_1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{|J_1|} \sum_{j \in J_1} \left[x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n})\right]^2$$

$$f_2 = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + \frac{2}{|J_2|} \sum_{j \in J_2} \left[x_j - 2x_2 \sin(2\pi x_1 + \frac{j\pi}{n})\right]^2$$

$$f_3 = \sin(0.5x_1\pi) + \frac{2}{|J_3|}\sum_{j\,\in\,J_3}\left[x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n})\right]^2$$

where

$J_1 = \{j|3 \leq j \leq n$, and j-1 is a multiplication of 3}

$J_2 = \{j|3 \leq j \leq n$, and j-2 is a multiplication of 3}

$J_3 = \{j|3 \leq j \leq n$, and j is a multiplication of 3}

The search space is $[0,1]^2 \times [-2,2]^{n-2}$

Its PF is

$$f_1^2 + f_2^2 + f_3^2 = 1, \quad 0 \leq f_1, f_2, f_3 \leq 1$$

Its PS is

$$x_j = 2x_2\sin\left(2\pi x_1 + \frac{j\pi}{n}\right), \; j = 3, \dots, n.$$

4.1.9. Unconstrained Problem 9

The three objectives to be minimized in this problem are:

$$f_1 = 0.5[\max\{0, (1+\varepsilon)(1-4(2x_1-1)^2)\} + 2x_1]x_2 + \frac{2}{|J_1|}\sum_{j\,\in\,J_1}\left[x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n})\right]^2$$

$$f_2 = 0.5[\max\{0, (1+\varepsilon)(1-4(2x_1-1)^2)\} + 2x_1]x_2 + \frac{2}{|J_2|}\sum_{j\,\in\,J_2}\left[x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n})\right]^2$$

$$f_3 = 1 - x_2 + \frac{2}{|J_3|}\sum_{j\,\in\,J_3}\left[x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n})\right]^2$$

where

$J_1 = \{j|3 \leq j \leq n$, and j-1 is a multiplication of 3}

$J_2 = \{j|3 \leq j \leq n$, and j-2 is a multiplication of 3}

$J_3 = \{j|3 \leq j \leq n$, and j is a multiplication of 3} and $\varepsilon = 0.1$, also can take any positive value

The search space is $[0,1]^2 \times [-2,2]^{n-2}$

Its PF has two parts, the first part is:

$$0 \le f_3 \le 1,$$

$$0 \le f_1 \le \frac{1}{4}(1-f_3),$$

$$f_2 = 1 - f_1 - f_3;$$

and the second part is:

$$0 \le f_3 \le 1,$$

$$\frac{3}{4}(1-f_3) \le f_1 \le 1,$$

$$f_2 = 1 - f_1 - f_3;$$

Its PS also has two parts that are disconnected:

$$x_1 \in [0, 0.25] \cup [0.75, 1], 0 \le x_2 \le 1$$

$$x_j = 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right), \quad j = 3, \dots, n.$$

4.1.10. Unconstrained Problem 10

The three objectives to be minimized in this problem are:

$$f_1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{|J_1|}\sum_{j \in J_1}[4y_j^2 - \cos(8\pi y_j) + 1]$$

$$f_2 = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + \frac{2}{|J_2|}\sum_{j \in J_2}[4y_j^2 - \cos(8\pi y_j) + 1]$$

$$f_3 = \sin(0.5x_1\pi) + \frac{2}{|J_3|}\sum_{j \in J_3}[4y_j^2 - \cos(8\pi y_j) + 1]$$

where

$J_1 = \{j | 3 \leq j \leq n$, and j-1 is a multiplication of 3$\}$

$J_2 = \{j | 3 \leq j \leq n$, and j-2 is a multiplication of 3$\}$

$J_3 = \{j | 3 \leq j \leq n$, and j is a multiplication of 3$\}$

and

$$y_j = x_j - 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right), \; j = 3, \dots, n$$

The search space is $[0,1]^2 \times [-2,2]^{n-2}$

Its PF is

$$f_1^2 + f_2^2 + f_3^2 = 1, \quad 0 \leq f_1, f_2, f_3 \leq 1$$

Its PS is

$$x_j = 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right), \; j = 3, \dots, n.$$

4.2. Constrained Multiobjective Test Problems

4.2.1. Constrained Problem 1

The two objectives to be minimized in this problem are:

$$f_1(x) = x_1 + \frac{2}{|J_1|} \sum_{j \in J_1} \left[x_j - x_1^{0.5(1.0 + \frac{3(j-2)}{n-2})} \right]^2$$

$$f_2(x) = 1 - \sqrt{x_1} + \frac{2}{|J_2|} \sum_{j \in J_2} \left[x_j - x_1^{0.5(1.0 + \frac{3(j-2)}{n-2})} \right]^2$$

where $J_1 = \{j | j$ is odd and $2 \leq j \leq n\}$ and $J_2 = \{j | j$ is even and $2 \leq j \leq n\}$.

The constraint is

$$f_1 + f_2 - a|\sin[N\pi(f_1 - f_2 + 1)]| - 1 \geq 0$$

where N is an integer and $a \geq \frac{1}{2N}$.

The search space is $[0,1]^n$

Its PF in the objective space consists of 2N+1 points:

$$\left(\frac{i}{2N}, 1-\frac{i}{2N}\right), i = 0,1,\ldots,2N.$$

4.2.2. Constrained Problem 2

The two objectives to be minimized in this problem are:

$$f_1 = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}\left[x_j - \sin(6\pi x_1 + \frac{j\pi}{n})\right]^2$$

$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}\left[x_j - \cos(6\pi x_1 + \frac{j\pi}{n})\right]^2$$

where $J_1 = \{j|j \text{ is odd and } 2 \leq j \leq n\}$ and $J_2 = \{j|j \text{ is even and } 2 \leq j \leq n\}$.

The constraint is

$$\frac{t}{1+e^{4|t|}} \geq 0$$

where

$$t = f_2 + \sqrt{f_1} - \mathrm{a}\sin\left[N\pi\left(\sqrt{f_1} - f_2 + 1\right)\right] - 1$$

The search space is $[0,1] \times [-1,1]^{n-1}$

Its PF in the objective space consists of

- An isolated Pareto Optimal solution (0,1) in the objective space and
- N disconnected parts, the *i-th* part is:

$$f_2 = 1 - \sqrt{f_1}, \qquad \left(\frac{2i-1}{2N}\right)^2 \leq f_1 \leq \left(\frac{2i}{2N}\right)^2, \qquad i = 1,\ldots,N.$$

4.2.3. Constrained Problem 3

The two objectives to be minimized in this problem are:

$$f_1 = x_1 + \frac{2}{|J_1|} \left(4 \sum_{j \in J_1} [y_j]^2 - 2 \prod_{j \in J_1} \cos\left(\frac{20 y_j \pi}{\sqrt{j}}\right) + 2\right)$$

$$f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|} \left(4 \sum_{j \in J_2} [y_j]^2 - 2 \prod_{j \in J_2} \cos\left(\frac{20 y_j \pi}{\sqrt{j}}\right) + 2\right)$$

where $J_1 = \{j | j \text{ is odd and } 2 \le j \le n\}$ and $J_2 = \{j | j \text{ is even and } 2 \le j \le n\}$, and

$$y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right),\ j = 2, \dots, n.$$

The constraint is

$$f_2 + f_1^2 - a\sin[N\pi(f_1^2 - f_2 + 1)] - 1 \ge 0$$

The search space is $[0,1] \times [-2,2]^{n-1}$

Its PF in the objective space consists of

- An isolated Pareto Optimal solution (0,1) in the objective space and
- N disconnected parts, the *i-th* part is:

$$f_2 = 1 - f_1^2, \qquad \sqrt{\left(\frac{2i-1}{2N}\right)} \le f_1 \le \sqrt{\frac{2i}{2N}}, \qquad i = 1, \dots, N.$$

4.2.4. Constrained Problem 4

The two objectives to be minimized in this problem are:

$$f_1 = x_1 + \sum_{j \in J_1} h_j(y_j)$$

$$f_2 = 1 - x_1 + \sum_{j \in J_2} h_j(y_j)$$

where $J_1 = \{j | j \text{ is odd and } 2 \le j \le n\}$ and $J_2 = \{j | j \text{ is even and } 2 \le j \le n\}$.

$$y_j = x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right),\ j = 2, \dots, n,$$

The search space is $[0,1] \times [-2,2]^{n-1}$

$$h_2(t) = \begin{cases} |t| & if\ t < \frac{3}{2}(1 - \frac{\sqrt{2}}{2}) \\ 0.125 + (t-1)^2 & otherwise \end{cases}$$

and

$$h_j(t) = t^2$$

for $j = 3,4, \dots, n$.

The constraint is:

$$\frac{t}{1 + e^{4|t|}} \geq 0$$

where

$$t = x_2 - \sin\left(6\pi x_1 + \frac{2\pi}{n}\right) - 0.5x_1 + 0.25.$$

The PF in the objective space is:

$$f_2 = \begin{cases} 1 - f_1 & if\ 0 \leq f_1 \leq\ 0.5 \\ -0.5f_1 + \frac{3}{4} & if\ 0.5 \leq f_1 \leq\ 0.75 \\ 1 - f_1 + 0.125 & if\ 0.75 \leq f_1 \leq\ 1 \end{cases}$$

4.2.5. Constrained Problem 5

The two objectives to be minimized in this problem are:

$$f_1 = x_1 + \sum_{j \in J_1} h_j(y_j)$$

$$f_2 = 1 - x_1 + \sum_{j \in J_2} h_j(y_j)$$

where $J_1 = \{j|j \text{ is odd and } 2 \leq j \leq n\}$ and $J_2 = \{j|j \text{ is even and } 2 \leq j \leq n\}$.

$$y_j = \begin{cases} x_j - 0.8x_1 \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), & if\ j \in J_1 \\ x_j - 0.8x_1 \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), & if\ j \in J_2, \end{cases}$$

$$h_2(t) = \begin{cases} |t| & if\ t < \frac{3}{2}(1 - \frac{\sqrt{2}}{2}) \\ 0.125 + (t-1)^2 & otherwise \end{cases}$$

and

$$h_2(t) = 2t^2 - \cos(4\pi t) + 1$$

for $j = 3,4,\ldots,n$.

The search space is $[0,1] \times [-2,2]^{n-1}$

The constraint is:

$$x_2 - 0.8x_1 \sin\left(6\pi x_1 + \frac{2\pi}{n}\right) - 0.5x_1 + 0.25 \geq 0$$

The PF in the objective space is:

$$f_2 = \begin{cases} 1 - f_1 & if\ 0 \leq f_1 \leq 0.5 \\ -0.5f_1 + \frac{3}{4} & if\ 0.5 \leq f_1 \leq 0.75 \\ 1 - f_1 + 0.125 & if\ 0.75 \leq f_1 \leq 1 \end{cases}$$

4.2.6. Constrained Problem 6

The two objectives to be minimized in this problem are:

$$f_1 = x_1 + \sum_{j \in J_1} (y_j)^2$$

$$f_2 = (1 - x_1)^2 + \sum_{j \in J_2} (y_j)^2$$

where $J_1 = \{j|j \text{ is odd and } 2 \leq j \leq n\}$ and $J_2 = \{j|j \text{ is even and } 2 \leq j \leq n\}$, and

$$y_j = \begin{cases} x_j - 0.8x_1 \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), & if\ j \in J_1 \\ x_j - 0.8x_1 \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), & if\ j \in J_2, \end{cases}$$

The search space is $[0,1] \times [-2,2]^{n-1}$

The constraint is:

$$x_2 - 0.8x_1 \sin\left(6\pi x_1 + \frac{2\pi}{n}\right) - \sin(0.5(1-x_1)) - (1-x_1)^2\sqrt{|0.5(1-x_1) - (1-x_1)^2|} \geq 0$$

and

$$x_4 - 0.8x_1 \sin\left(6\pi x_1 + \frac{4\pi}{n}\right) - \sin(0.25\sqrt{(1-x_1)}$$
$$- 0.5(1-x_1))\sqrt{|0.25\sqrt{(1-x_1)} - 0.5(1-x_1)|} \geq 0$$

The PF in the objective space is:

$$f_2 = \begin{cases} (1-f_1)^2 & if\ 0 \leq f_1 \leq 0.5 \\ 0.5(1-f_1) & if\ 0.5 \leq f_1 \leq 0.75 \\ 0.25\sqrt{1-f_1} & if\ 0.75 \leq f_1 \leq 1 \end{cases}$$

4.2.7. Constrained Problem 7

The two objectives to be minimized in this problem are:

$$f_1 = x_1 + \sum_{j \in J_1} h_j(y_j)$$

$$f_2 = (1-x_1)^2 + \sum_{j \in J_2} h_j(y_j)$$

where $J_1 = \{j|j$ is odd and $2 \leq j \leq n\}$ and $J_2 = \{j|j$ is even and $2 \leq j \leq n\}$, and

$$y_j = \begin{cases} x_j - \cos\left(6\pi x_1 + \frac{j\pi}{n}\right), & if\ j \in J_1 \\ x_j - \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), & if\ j \in J_2, \end{cases}$$

$$h_2(t) = h_4(t) = t^2$$

and

$$h_j(t) = 2t^2 - \cos(4\pi t) + 1$$

for $j = 3,5,6, \ldots, n$.

The search space is $[0,1] \times [-2,2]^{n-1}$

The constraints are:

$$x_2 - \sin\left(6\pi x_1 + \frac{2\pi}{n}\right) - \sin(0.5(1-x_1)) - (1-x_1)^2\sqrt{|0.5(1-x_1) - (1-x_1)^2|} \geq 0$$

and

$$x_4 - \sin\left(6\pi x_1 + \frac{4\pi}{n}\right) - \sin(.25\sqrt{(1-x_1)} - .5(1-x_1))\sqrt{|.25\sqrt{(1-x_1)} - .5(1-x_1)|} \geq 0$$

The PF in the objective space is:

$$f_2 = \begin{cases} (1-f_1)^2 & if\ 0 \leq f_1 \leq 0.5 \\ 0.5(1-f_1) & if\ 0.5 \leq f_1 \leq 0.75 \\ 0.25\sqrt{1-f_1} & if\ 0.75 \leq f_1 \leq 1 \end{cases}$$

4.2.8. Constrained Problem 8

The three objectives to be minimized in this problem are:

$$f_1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{|J_1|}\sum_{j\,\in\,J_1}\left[x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n})\right]^2$$

$$f_2 = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + \frac{2}{|J_2|}\sum_{j\,\in\,J_2}\left[x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n})\right]^2$$

$$f_3 = \sin(0.5x_1\pi) + \frac{2}{|J_3|}\sum_{j\,\in\,J_3}\left[x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n})\right]^2$$

where

$J_1 = \{j|3 \le j \le n$, and j-1 is a multiplication of 3\}

$J_2 = \{j|3 \le j \le n$, and j-2 is a multiplication of 3\}

$J_3 = \{j|3 \le j \le n$, and j is a multiplication of 3\}

The search space is $[0,1]^2 \times [-4,4]^{n-2}$

The constraint is

$$\frac{f_1^2 + f_2^2}{1 - f_3^2} - a\,|\sin[N\pi\left(\frac{f_1^2 + f_2^2}{1 - f_3^2} + 1\right)]| - 1 \ge 0$$

Its PF will have 2N+1 disconnected parts:

$$f_1 = \left[\frac{i}{2N}(1 - f_3^2)\right]^{\frac{1}{2}}$$

$$f_2 = [1 - f_1^2 - f_3^2]^{\frac{1}{2}}$$

$$0 \le f_3 \le 1$$

4.2.9. Constrained Problem 9

The three objectives to be minimized in this problem are:

$$f_1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{|J_1|}\sum_{j \in J_1}\left[x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n})\right]^2$$

$$f_2 = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + \frac{2}{|J_2|}\sum_{j \in J_2}\left[x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n})\right]^2$$

$$f_3 = \sin(0.5x_1\pi) + \frac{2}{|J_3|}\sum_{j \in J_3}\left[x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n})\right]^2$$

where

$J_1 = \{j|3 \le j \le n$, and j-1 is a multiplication of 3\}

$J_2 = \{j|3 \le j \le n$, and j-2 is a multiplication of 3\}

$J_3 = \{j|3 \le j \le n$, and j is a multiplication of 3\}

The search space is $[0,1]^2 \times [-2,2]^{n-2}$

The constraint is

$$\frac{f_1^2 + f_2^2}{1 - f_3^2} - a\,|\sin[N\pi\left(\frac{f_1^2 - f_2^2}{1 - f_3^2} + 1\right)]| - 1 \geq 0$$

Its PF consists of:

- A curve:

$$f_1 = 0$$

$$0 \leq f_2 \leq 1$$

$$f_3 = [1 - f_2^2]^{\frac{1}{2}}$$

- N disconnected nonlinear 2-D surfaces, the i-th one it:

$$0 \leq f_3 \leq 1$$

$$\{\frac{2i-1}{2N}(1 - f_3^2)\}^{\frac{1}{2}} \leq f_1 \leq \{\frac{2i}{2N}(1 - f_3^2)\}^{\frac{1}{2}}$$

$$f_1 = [1 - f_1^2 - f_2^2]^{\frac{1}{2}}.$$

4.2.10. Constrained Problem 10

The three objectives to be minimized in this problem are:

$$f_1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{|J_1|}\sum_{j \in J_1}[4y_j^2 - \cos(8\pi y_j) + 1]$$

$$f_2 = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + \frac{2}{|J_2|}\sum_{j \in J_2}[4y_j^2 - \cos(8\pi y_j) + 1]$$

$$f_3 = \sin(0.5x_1\pi) + \frac{2}{|J_3|}\sum_{j \in J_3}[4y_j^2 - \cos(8\pi y_j) + 1]$$

where

$J_1 = \{j | 3 \leq j \leq n$, and j-1 is a multiplication of 3}

$J_2 = \{j|3 \leq j \leq n$, and j-2 is a multiplication of 3}

$J_3 = \{j|3 \leq j \leq n$, and j is a multiplication of 3}

and

$$y_j = x_j - 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right), \quad j = 3, \dots, n$$

The search space is $[0,1]^2 \times [-2,2]^{n-2}$

The constraint is

$$\frac{f_1^2 + f_2^2}{1 - f_3^2} - a\,|\sin[N\pi\left(\frac{f_1^2 - f_2^2}{1 - f_3^2} + 1\right)]| - 1 \geq 0$$

Its PF consists of:

- A curve:

$$f_1 = 0$$

$$0 \leq f_2 \leq 1$$

$$f_3 = [1 - f_2^2]^{\frac{1}{2}}$$

- N disconnected nonlinear 2-D surfaces, the i-th one it:

$$0 \leq f_3 \leq 1$$

$$\{\frac{2i-1}{2N}(1 - f_3^2)\}^{\frac{1}{2}} \leq f_1 \leq \{\frac{2i}{2N}(1 - f_3^2)\}^{\frac{1}{2}}$$

$$f_1 = [1 - f_1^2 - f_2^2]^{\frac{1}{2}}.$$

5. RESULTS AND DISCUSSION

Performance metrics of all the functions that are mentioned in Chapter 4 are tested and the results were tabulated by running the code each time by varying either a) number of iterations, b) number of variables, c) population size, and d) archive size. In each of these runs, the rest of the parameters were kept constant to understand the role of each of these factors.

In the first study, as shown in Table 5.1, the number of iterations (nIter) were set to 50, 100, 150 and 200 in each run by keeping the number of variables, population size and archive size constant. It was observed that the standard deviation decreases as the iterations increase, suggesting that more iterations are needed to reduce the error in the standard deviation.

In the second study, as shown in Table 5.2, the variable size was varied keeping the rest of the factors constant. Even here, it was observed that with the increase in variable size, there was a substantial decrease in the standard deviation.

In the third set of studies, the population size was changed by keeping variable size, iterations and archive size constant. In this case, with the increase in the size of population, there was a significant increase in the standard deviation.

In the fourth set, the archive size was varied keeping the remaining factors constant. In this case, as the archive size was increased, there was a decrease in the standard deviation.

Table 5.1. Effect of nIter with constant variable size of nVar 30, nPop 50, nArchive 50

		Objective #1					Objective #2					Objective #3				
No. of Iterations	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
50	**UF1**	0.032	1.130	1.098	0.425	0.537	0.050	1.252	1.201	0.444	0.562					
	UF2	0.062	1.526	1.465	0.456	0.572	0.063	1.399	1.336	0.392	0.599					
	UF3	0.816	0.882	0.067	0.020	0.850	0.567	0.652	0.086	0.023	0.600					
	UF4	0.743	0.919	0.175	0.048	0.794	0.571	0.910	0.338	0.092	0.688					
	UF5	1.122	3.542	2.420	0.612	1.972	0.793	2.704	1.912	0.622	1.619					
	UF6	0.671	1.546	0.875	0.360	1.028	0.490	1.037	0.547	0.177	0.691					
	UF7	0.079	1.048	0.968	0.203	0.305	0.044	1.032	0.988	0.199	0.785					
	UF8	0.028	1.340	1.312	0.275	0.949	0.048	0.983	0.934	0.246	0.320	0.056	1.148	1.092	0.275	0.342
	UF9	0.027	1.087	1.060	0.279	0.297	0.026	1.634	1.608	0.321	0.416	0.477	1.318	0.841	0.232	0.755
	UF10	1.017	2.580	1.563	0.381	1.629	0.396	2.607	2.211	0.538	0.967	1.636	3.329	1.693	0.432	2.397
	CF1	0.031	1.079	1.048	0.453	0.509	0.049	1.048	0.999	0.436	0.540					
	CF2	0.045	1.095	1.050	0.317	0.722	0.029	1.000	0.971	0.269	0.286					
	CF3	0.726	1.228	0.502	0.119	0.841	0.497	0.947	0.450	0.130	0.662					
	CF4	0.194	1.810	1.616	0.549	0.852	0.646	2.813	2.167	0.756	1.650					
	CF5	7.063	17.057	9.994	3.480	10.934	1.137	19.800	18.663	6.422	5.959					
	CF6	0.697	6.056	5.359	1.371	2.491	0.315	5.004	4.689	1.064	1.442					
	CF7	5.737	12.457	6.720	1.702	8.374	3.252	11.899	8.647	2.000	4.612					
	CF8	0.026	1.596	1.570	0.410	0.893	0.026	1.270	1.244	0.344	0.520	0.031	1.083	1.053	0.330	0.318
	CF9	0.025	1.760	1.735	0.465	0.877	0.026	1.346	1.320	0.361	0.365	0.053	1.197	1.144	0.399	0.419
	CF10	0.478	3.188	2.710	0.671	1.488	0.623	2.098	1.475	0.323	1.321	0.512	3.143	2.631	0.841	2.005
100	**UF1**	0.021	1.049	1.027	0.361	0.306	0.041	1.041	1.000	0.380	0.684					
	UF2	0.052	1.414	1.362	0.414	0.545	0.044	1.064	1.021	0.306	0.487					

Table 5.1. Effect of nIter with constant variable size of nVar 30, nPop 50, nArchive 50 (continued)

		Objective #1					Objective #2					Objective #3				
No. of Iterations	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
100	**UF3**	0.747	0.931	0.183	0.051	0.815	0.560	1.750	1.190	0.190	0.707					
	UF4	0.704	0.901	0.197	0.061	0.812	0.579	0.667	0.089	0.025	0.622					
	UF5	1.394	1.420	0.026	0.006	1.399	1.237	1.323	0.085	0.031	1.259					
	UF6	0.701	0.809	0.108	0.031	0.738	0.633	0.889	0.255	0.078	0.739					
	UF7	0.014	0.941	0.927	0.165	0.273	0.112	1.038	0.925	0.159	0.786					
	UF8	0.055	1.261	1.207	0.317	0.795	0.009	1.115	1.106	0.193	0.147	0.080	1.112	1.033	0.324	0.689
	UF9	0.042	1.255	1.213	0.308	0.453	0.040	1.065	1.025	0.277	0.471	0.429	1.179	0.750	0.152	0.586
	UF10	0.534	2.345	1.812	0.456	1.373	0.646	2.533	1.887	0.492	1.292	0.580	3.701	3.121	0.744	1.831
	CF1	0.011	1.036	1.025	0.480	0.464	0.030	1.034	1.004	0.467	0.557					
	CF2	0.014	1.024	1.010	0.367	0.229	0.036	1.051	1.015	0.380	0.785					
	CF3	0.745	0.890	0.145	0.042	0.795	0.558	0.713	0.155	0.050	0.627					
	CF4	0.189	1.201	1.012	0.294	0.680	0.385	1.398	1.013	0.284	0.928					
	CF5	7.369	20.936	13.566	3.697	15.140	0.561	3.998	3.437	0.604	1.552					
	CF6	0.706	3.591	2.886	0.879	1.617	0.254	2.378	2.125	0.597	1.146					
	CF7	7.378	8.643	1.265	0.303	7.609	4.328	4.845	0.517	0.146	4.478					
	CF8	0.024	1.473	1.449	0.319	0.960	0.001	1.199	1.197	0.297	0.248	0.017	1.134	1.118	0.349	0.374
	CF9	0.047	1.712	1.665	0.386	1.127	0.004	1.233	1.229	0.355	0.290	0.012	1.160	1.147	0.322	0.301
	CF10	0.620	2.712	2.092	0.575	1.628	0.725	2.035	1.309	0.307	1.422	0.745	3.151	2.406	0.622	1.573
150	**UF1**	0.016	1.038	1.021	0.358	0.304	0.020	1.038	1.017	0.384	0.663					
	UF2	0.022	1.621	1.599	0.473	0.595	0.026	0.958	0.932	0.297	0.438					
	UF3	0.725	0.878	0.152	0.044	0.803	0.501	0.625	0.124	0.038	0.577					
	UF4	0.732	0.892	0.160	0.045	0.802	0.531	0.794	0.263	0.064	0.616					
	UF5	1.492	1.944	0.452	0.124	1.632	1.383	1.774	0.390	0.121	1.577					
	UF6	0.719	0.749	0.031	0.007	0.727	0.656	0.787	0.131	0.040	0.724					
	UF7	0.017	0.844	0.826	0.184	0.262	0.217	1.015	0.799	0.176	0.777					

Table 5.1. Effect of nIter with constant variable size of nVar 30, nPop 50, nArchive 50 (continued)

		Objective #1					**Objective #2**					**Objective #3**				
No. of Iterations	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
	UF8	0.193	1.507	1.314	0.269	0.942	0.013	2.305	2.292	0.667	0.673	0.028	1.102	1.074	0.264	0.336
	UF9	0.009	0.776	0.767	0.235	0.257	0.014	0.938	0.923	0.275	0.401	0.486	1.112	0.626	0.184	0.672
	UF10	0.919	1.588	0.669	0.160	1.083	0.777	1.427	0.650	0.127	0.920	1.396	2.842	1.445	0.299	2.311
	CF1	0.011	1.019	1.008	0.314	0.207	0.033	1.041	1.008	0.342	0.766					
	CF2	0.021	1.073	1.052	0.433	0.601	0.016	1.011	0.994	0.409	0.392					
	CF3	0.770	0.885	0.115	0.032	0.821	0.492	0.698	0.206	0.059	0.558					
	CF4	0.083	1.274	1.191	0.415	0.611	0.071	1.212	1.141	0.414	0.677					
	CF5	4.845	14.600	9.754	3.764	7.945	1.046	9.450	8.405	2.823	6.395					
	CF6	0.744	4.171	3.427	1.053	1.966	0.267	1.976	1.710	0.411	0.839					
	CF7	7.656	19.441	11.785	2.588	10.189	5.578	11.756	6.178	1.676	8.835					
	CF8	0.085	1.274	1.189	0.299	0.817	0.015	1.123	1.108	0.262	0.232	0.025	1.216	1.191	0.392	0.548
	CF9	0.047	1.216	1.170	0.293	0.793	0.006	0.770	0.765	0.141	0.098	0.064	1.156	1.092	0.346	0.644
	CF10	0.571	2.000	1.429	0.539	1.087	1.094	1.993	0.899	0.264	1.489	0.503	2.571	2.068	0.794	1.645
200	**UF1**	0.005	1.008	1.004	0.264	0.125	0.019	1.022	1.003	0.291	0.840					
	UF2	0.014	1.198	1.183	0.385	0.422	0.101	1.035	0.934	0.283	0.554					
	UF3	0.777	0.921	0.144	0.045	0.834	0.545	0.689	0.145	0.045	0.603					
	UF4	0.693	0.892	0.199	0.048	0.791	0.556	0.670	0.114	0.034	0.616					
	UF5	1.082	1.563	0.482	0.166	1.357	1.045	1.745	0.700	0.256	1.390					
	UF6	0.725	0.814	0.089	0.026	0.758	0.688	0.915	0.228	0.069	0.782					
	UF7	0.042	0.875	0.833	0.137	0.283	0.232	1.017	0.785	0.128	0.777					
	UF8	0.013	1.152	1.139	0.287	0.796	0.001	0.562	0.560	0.113	0.084	0.013	1.104	1.090	0.365	0.533
	UF9	0.015	1.531	1.516	0.327	0.334	0.013	0.816	0.803	0.218	0.328	0.441	1.274	0.832	0.225	0.704
	UF10	0.670	2.224	1.554	0.529	1.374	0.384	1.065	0.681	0.204	0.687	0.466	1.609	1.143	0.502	1.029
	CF1	0.020	1.024	1.005	0.315	0.181	0.013	1.031	1.018	0.332	0.803					

Table 5.1. Effect of nIter with constant variable size of nVar 30, nPop 50, nArchive 50 (continued)

		Objective #1					Objective #2					Objective #3				
No. of Iterations	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
	CF2	0.017	1.054	1.037	0.414	0.312	0.004	1.029	1.025	0.424	0.677					
	CF3	0.715	0.918	0.203	0.060	0.809	0.549	0.676	0.127	0.041	0.598					
	CF4	0.162	1.508	1.346	0.377	0.616	0.346	1.526	1.180	0.380	0.981					
	CF5	5.168	21.897	16.729	6.496	15.004	1.452	12.519	11.067	4.551	4.333					
	CF6	0.672	2.049	1.376	0.427	1.179	0.663	2.123	1.460	0.386	1.119					
	CF7	4.659	12.491	7.831	2.630	9.517	6.213	10.303	4.090	1.359	7.689					
	CF8	0.099	1.114	1.014	0.265	0.806	0.006	0.771	0.765	0.161	0.150	0.045	1.088	1.042	0.375	0.535
	CF9	0.050	1.142	1.092	0.249	0.805	0.002	1.855	1.852	0.555	0.405	0.038	1.188	1.150	0.332	0.516
	CF10	0.797	2.137	1.340	0.486	1.241	1.143	1.839	0.696	0.195	1.376	0.618	1.800	1.181	0.485	1.372

Table 5.2. Effect of nVar with constant Number of iterations 100, nPop 50, nArchive 50

		Objective #1					Objective #2					Objective #3				
nVar	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
10	**UF1**	0.000	1.025	1.025	0.358	0.491	0.003	1.085	1.083	0.343	0.480					
	UF2	0.001	1.618	1.617	0.338	0.271	0.072	1.155	1.083	0.255	0.627					
	UF3	0.600	2.226	1.626	0.580	1.134	0.407	0.519	0.112	0.038	0.470					
	UF4	0.322	0.605	0.282	0.073	0.563	0.408	2.380	1.972	0.647	0.675					
	UF5	0.666	1.324	0.658	0.180	0.868	0.570	1.006	0.436	0.116	0.931					
	UF6	0.681	1.515	0.834	0.117	0.705	0.441	0.717	0.275	0.038	0.697					
	UF7	0.017	0.939	0.922	0.186	0.225	0.110	0.989	0.878	0.180	0.782					

Table 5.2. Effect of nVar with constant Number of iterations 100, nPop 50, nArchive 50 (continued)

		Objective #1					Objective #2					Objective #3				
nVar	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
10	**UF8**	0.068	1.480	1.412	0.305	0.783	0.000	1.513	1.513	0.358	0.317	0.001	1.080	1.079	0.346	0.472
	UF9	0.001	2.088	2.087	0.495	0.506	0.000	0.692	0.692	0.214	0.207	0.385	1.851	1.466	0.400	0.743
	UF10	0.298	3.985	3.687	1.110	1.247	0.079	2.343	2.264	0.622	0.845	0.172	4.113	3.942	1.052	1.639
	CF1	0.001	1.009	1.008	0.350	0.310	0.001	1.030	1.029	0.364	0.598					
	CF2	0.004	1.048	1.044	0.386	0.499	0.007	1.091	1.084	0.387	0.459					
	CF3	0.591	0.831	0.240	0.066	0.670	0.395	2.605	2.210	1.021	1.902					
	CF4	0.000	1.000	1.000	0.336	0.435	0.011	1.065	1.054	0.304	0.660					
	CF5	0.596	6.926	6.330	2.123	3.916	0.000	1.933	1.933	0.524	0.369					
	CF6	0.000	2.168	2.168	0.605	1.109	0.013	1.181	1.169	0.391	0.384					
	CF7	1.134	2.144	1.010	0.292	1.509	0.701	0.838	0.137	0.039	0.743					
	CF8	0.000	1.462	1.462	0.322	0.667	0.000	0.991	0.991	0.302	0.394	0.002	1.099	1.097	0.327	0.677
	CF9	0.000	1.690	1.690	0.361	0.754	0.000	1.429	1.429	0.420	0.388	0.005	1.122	1.117	0.372	0.495
	CF10	0.161	4.043	3.882	0.819	1.041	0.000	4.204	4.204	1.430	1.228	1.139	7.463	6.324	1.159	1.810
20	**UF1**	0.009	1.082	1.072	0.453	0.361	0.031	1.090	1.059	0.466	0.673					
	UF2	0.017	1.323	1.306	0.362	0.480	0.045	1.252	1.207	0.283	0.503					
	UF3	0.703	1.018	0.315	0.090	0.933	0.533	0.749	0.216	0.067	0.654					
	UF4	0.641	0.768	0.127	0.041	0.683	0.573	0.719	0.146	0.043	0.634					
	UF5	0.637	0.978	0.341	0.099	0.735	1.169	2.276	1.107	0.365	1.865					
	UF6	0.774	1.223	0.449	0.166	0.871	0.569	0.719	0.150	0.060	0.681					
	UF7	0.033	0.937	0.905	0.178	0.229	0.096	0.982	0.886	0.171	0.791					
	UF8	0.059	1.242	1.183	0.295	0.751	0.000	1.014	1.014	0.212	0.191	0.031	1.402	1.371	0.364	0.734
	UF9	0.041	1.744	1.703	0.465	0.615	0.016	1.377	1.360	0.359	0.439	0.404	1.644	1.240	0.259	0.627
	UF10	0.357	2.016	1.659	0.538	0.693	0.677	1.350	0.673	0.194	0.936	0.564	2.050	1.486	0.470	1.576
	CF1	0.008	1.016	1.008	0.348	0.437	0.028	1.096	1.068	0.381	0.506					
	CF2	0.012	1.026	1.014	0.354	0.226	0.013	1.029	1.016	0.368	0.744					
	CF3	0.815	1.257	0.442	0.133	1.111	0.377	0.689	0.312	0.103	0.569					

Table 5.2. Effect of nVar with constant Number of iterations 100, nPop 50, nArchive 50 (continued)

		Objective #1					Objective #2					Objective #3				
nVar	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
	CF4	0.048	1.264	1.216	0.377	0.939	0.333	1.321	0.988	0.372	0.657					
	CF5	4.047	12.408	8.361	2.419	8.290	0.418	0.870	0.452	0.084	0.489					
	CF6	0.617	2.502	1.884	0.590	1.305	0.216	1.681	1.464	0.407	0.743					
	CF7	4.108	5.875	1.767	0.526	4.809	2.726	4.266	1.540	0.457	3.245					
	CF8	0.009	1.339	1.330	0.296	0.899	0.003	1.599	1.596	0.420	0.421	0.004	1.143	1.138	0.313	0.365
	CF9	0.031	1.185	1.154	0.283	0.792	0.012	1.607	1.595	0.461	0.427	0.013	1.075	1.062	0.381	0.450
	CF10	0.458	2.682	2.224	0.772	1.430	0.367	2.341	1.974	0.558	1.130	0.318	3.323	3.004	0.932	1.833
	UF1	0.054	1.144	1.090	0.322	0.306	0.070	1.107	1.037	0.362	0.745					
	UF2	0.093	1.185	1.091	0.331	0.596	0.092	1.066	0.973	0.291	0.479					
	UF3	0.687	0.844	0.157	0.046	0.744	0.457	0.741	0.285	0.077	0.556					
	UF4	0.735	0.800	0.065	0.018	0.759	0.489	0.532	0.043	0.013	0.507					
	UF5	1.199	2.180	0.981	0.326	1.415	1.190	2.392	1.201	0.334	2.163					
	UF6	0.798	0.807	0.009	0.003	0.801	0.608	0.619	0.011	0.003	0.611					
	UF7	0.065	1.145	1.080	0.298	0.415	0.085	1.177	1.092	0.290	0.770					
	UF8	0.099	1.323	1.224	0.257	1.062	0.017	0.595	0.578	0.142	0.176	0.047	1.243	1.196	0.269	0.306
	UF9	0.053	0.650	0.597	0.152	0.214	0.050	0.569	0.519	0.136	0.221	0.622	1.231	0.609	0.161	0.857
50	**UF10**	1.689	2.680	0.991	0.260	2.095	1.196	2.391	1.195	0.341	1.720	1.179	2.897	1.718	0.488	2.199
	CF1	0.036	1.058	1.023	0.414	0.360	0.072	1.085	1.013	0.410	0.691					
	CF2	0.037	1.055	1.018	0.373	0.473	0.056	1.067	1.012	0.388	0.545					
	CF3	0.762	0.809	0.047	0.014	0.782	0.481	0.517	0.036	0.011	0.495					
	CF4	0.703	1.832	1.129	0.369	1.161	0.761	1.968	1.207	0.366	1.495					
	CF5	13.548	32.210	18.662	5.161	24.973	2.438	13.365	10.927	2.449	4.202					
	CF6	1.164	5.585	4.421	1.506	2.763	1.272	6.072	4.800	1.514	3.164					
	CF7	12.384	34.445	22.060	6.227	17.688	12.000	22.235	10.235	2.359	17.019					
	CF8	0.087	1.199	1.113	0.213	0.989	0.028	1.231	1.203	0.265	0.202	0.059	1.044	0.986	0.301	0.471
	CF9	0.019	1.625	1.606	0.356	0.963	0.025	1.561	1.536	0.400	0.577	0.058	1.156	1.098	0.278	0.310

Table 5.2. Effect of nVar with constant Number of iterations 100, nPop 50, nArchive 50 (continued)

		Objective #1					Objective #2					Objective #3				
nVar	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
	CF10	1.070	2.954	1.884	0.493	1.939	1.004	2.678	1.675	0.479	1.807	0.849	3.047	2.198	0.642	1.714

Table 5.3. Effect of nPop with constant Number of iterations 100, nVar 50, nArchive 50

		Objective #1					Objective #2					Objective #3				
nPop	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
10	**UF1**	0.170	1.242	1.072	0.385	0.660	0.097	1.203	1.106	0.388	0.561					
	UF2	0.209	1.494	1.285	0.377	0.797	0.055	1.397	1.342	0.395	0.490					
	UF3	0.977	1.235	0.258	0.081	1.082	0.573	0.900	0.327	0.090	0.708					
	UF4	0.889	0.918	0.029	0.009	0.899	0.581	0.615	0.034	0.008	0.586					
	UF5	1.587	2.240	0.653	0.240	1.908	1.797	2.537	0.740	0.232	2.029					
	UF6	0.863	0.903	0.040	0.013	0.875	0.736	0.845	0.109	0.031	0.774					
	UF7	0.122	1.289	1.167	0.319	0.551	0.136	1.328	1.191	0.322	0.856					
	UF8	0.084	2.057	1.973	0.528	1.038	0.063	0.974	0.911	0.231	0.386	0.088	1.251	1.163	0.378	0.416
	UF9	0.017	0.730	0.713	0.190	0.224	0.035	0.666	0.631	0.161	0.274	0.620	1.118	0.498	0.099	0.797
	UF10	1.157	1.755	0.598	0.145	1.343	0.761	1.213	0.452	0.151	1.004	2.353	3.169	0.816	0.220	2.705
	CF1	0.129	1.229	1.100	0.362	0.598	0.124	1.220	1.096	0.391	0.641					
	CF2	0.130	1.204	1.074	0.394	0.744	0.084	1.208	1.124	0.368	0.464					
	CF3	0.860	0.886	0.026	0.008	0.871	0.635	0.699	0.064	0.021	0.658					
	CF4	0.955	3.145	2.190	0.736	1.894	0.291	2.398	2.107	0.709	1.033					
	CF5	7.675	22.153	14.478	5.681	12.316	2.698	8.383	5.686	1.796	5.016					
	CF6	0.901	6.379	5.478	1.404	2.263	0.566	3.381	2.815	0.801	1.818					
	CF7	6.489	17.969	11.480	3.032	11.392	7.525	24.439	16.914	5.384	13.004					
	CF8	0.047	1.179	1.132	0.454	0.652	0.061	0.567	0.505	0.207	0.338	0.022	1.064	1.042	0.480	0.420

Table 5.3. Effect of nPop with constant Number of iterations 100, nVar 50, nArchive 50 (continued)

		Objective #1					Objective #2					Objective #3				
nVar	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
	CF9	0.071	1.528	1.457	0.336	0.962	0.062	1.169	1.107	0.261	0.500	0.084	1.237	1.152	0.307	0.298
	CF10	1.037	2.580	1.543	0.385	1.556	0.609	1.754	1.146	0.332	1.088	1.890	3.044	1.153	0.304	2.524
20	**UF1**	0.051	1.053	1.002	0.373	0.291	0.039	1.077	1.038	0.387	0.778					
	UF2	0.099	1.302	1.203	0.369	0.587	0.079	1.462	1.383	0.345	0.525					
	UF3	0.858	0.928	0.070	0.019	0.885	0.607	0.672	0.064	0.016	0.626					
	UF4	0.881	0.930	0.050	0.015	0.898	0.600	0.716	0.116	0.036	0.644					
	UF5	0.929	2.088	1.159	0.410	1.535	1.427	2.216	0.789	0.217	1.707					
	UF6	0.696	0.765	0.070	0.020	0.727	0.662	0.932	0.270	0.072	0.728					
	UF7	0.034	1.131	1.097	0.247	0.309	0.056	1.090	1.034	0.235	0.811					
	UF8	0.024	1.417	1.393	0.336	0.975	0.031	0.723	0.692	0.201	0.274	0.025	1.130	1.105	0.294	0.280
	UF9	0.047	0.979	0.931	0.245	0.365	0.023	0.748	0.725	0.180	0.259	0.536	1.141	0.605	0.122	0.704
	UF10	1.025	3.181	2.156	0.624	2.108	0.853	3.778	2.925	0.494	1.583	0.891	2.257	1.366	0.398	1.529
	CF1	0.040	1.116	1.076	0.404	0.528	0.102	1.252	1.150	0.406	0.599					
	CF2	0.031	1.156	1.124	0.349	0.524	0.087	1.338	1.250	0.372	0.584					
	CF3	0.726	1.226	0.500	0.138	0.872	0.544	1.118	0.574	0.166	0.763					
	CF4	0.723	1.059	0.337	0.159	0.953	0.245	0.645	0.400	0.183	0.381					
	CF5	6.508	16.779	10.271	3.879	11.180	1.627	7.017	5.390	1.725	3.267					
	CF6	0.842	6.535	5.694	1.464	1.915	0.412	2.721	2.309	0.645	1.529					
	CF7	8.560	26.078	17.518	6.101	14.139	7.216	20.792	13.576	4.847	11.287					
	CF8	0.011	1.483	1.472	0.275	1.061	0.018	0.814	0.796	0.206	0.269	0.019	1.041	1.021	0.211	0.202
	CF9	0.020	1.757	1.737	0.353	1.063	0.038	1.496	1.458	0.338	0.558	0.038	1.066	1.028	0.172	0.180
	CF10	0.812	4.527	3.715	0.830	2.474	0.990	4.269	3.279	0.692	2.260	0.940	4.591	3.651	0.830	2.282
40	**UF1**	0.016	1.054	1.038	0.445	0.541	0.027	1.053	1.026	0.440	0.471					
	UF2	0.008	1.350	1.343	0.406	0.505	0.062	1.206	1.144	0.330	0.541					
	UF3	0.714	0.972	0.258	0.074	0.853	0.497	1.100	0.603	0.118	0.608					
	UF4	0.669	0.908	0.239	0.077	0.799	0.568	0.682	0.114	0.032	0.619					

Table 5.3. Effect of nPop with constant Number of iterations 100, nVar 50, nArchive 50 (continued)

		Objective #1					Objective #2					Objective #3				
nVar	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
	UF5	1.432	2.124	0.692	0.312	1.693	1.125	1.813	0.688	0.297	1.501					
	UF6	0.690	1.361	0.671	0.290	0.903	0.549	1.028	0.479	0.158	0.743					
	UF7	0.029	1.052	1.023	0.217	0.352	0.040	1.032	0.992	0.204	0.730					
	UF8	0.148	1.493	1.344	0.301	0.920	0.005	1.133	1.128	0.347	0.301	0.058	1.450	1.392	0.449	0.573
	UF9	0.034	0.808	0.774	0.222	0.303	0.028	0.828	0.800	0.241	0.372	0.527	1.175	0.648	0.161	0.693
	UF10	0.494	2.724	2.229	0.754	1.410	0.740	2.244	1.503	0.370	1.394	0.997	3.137	2.140	0.598	1.960
	CF1	0.034	1.128	1.093	0.409	0.518	0.027	1.098	1.071	0.410	0.509					
	CF2	0.022	1.067	1.045	0.394	0.446	0.018	1.105	1.087	0.394	0.530					
	CF3	0.783	0.968	0.185	0.049	0.846	0.530	0.702	0.172	0.051	0.593					
	CF4	0.633	1.380	0.747	0.138	0.943	0.347	1.037	0.689	0.138	0.792					
	CF5	6.077	20.485	14.408	3.078	9.061	1.290	6.930	5.641	1.086	4.787					
	CF6	0.651	5.595	4.943	1.418	2.189	0.237	3.354	3.117	0.790	1.301					
	CF7	3.072	13.023	9.951	4.084	7.353	6.051	15.792	9.740	3.645	10.528					
	CF8	0.485	1.511	1.025	0.215	0.989	0.022	1.927	1.905	0.461	0.409	0.040	1.144	1.103	0.295	0.342
	CF9	0.054	1.501	1.447	0.325	1.001	0.002	0.897	0.894	0.178	0.145	0.022	1.115	1.093	0.386	0.463
	CF10	0.537	3.446	2.910	0.923	1.833	0.283	2.901	2.618	0.478	1.021	0.608	3.570	2.962	0.732	2.054
50	**UF1**	0.018	1.095	1.077	0.403	0.375	0.026	1.158	1.132	0.419	0.648					
	UF2	0.023	1.288	1.265	0.322	0.441	0.049	1.041	0.993	0.244	0.509					
	UF3	0.719	0.953	0.235	0.071	0.821	0.529	0.947	0.418	0.118	0.630					
	UF4	0.677	0.905	0.228	0.068	0.791	0.560	0.690	0.131	0.036	0.613					
	UF5	1.191	1.270	0.079	0.024	1.218	1.366	1.477	0.111	0.030	1.416					
	UF6	0.743	0.848	0.105	0.031	0.800	0.540	0.871	0.331	0.084	0.619					
	UF7	0.043	0.905	0.862	0.167	0.272	0.257	1.000	0.743	0.146	0.776					
	UF8	0.026	1.245	1.219	0.319	0.794	0.007	0.448	0.441	0.099	0.097	0.030	1.164	1.135	0.368	0.671
	UF9	0.011	0.757	0.746	0.227	0.286	0.021	0.633	0.612	0.160	0.210	0.524	1.062	0.537	0.143	0.709
	UF10	0.628	1.711	1.084	0.317	1.053	0.601	1.772	1.171	0.325	0.980	1.295	1.996	0.702	0.199	1.619

Table 5.3. Effect of nPop with constant Number of iterations 100, nVar 50, nArchive 50 (continued)

		Objective #1					Objective #2					Objective #3				
nVar	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
	CF1	0.007	1.121	1.115	0.393	0.366	0.020	1.074	1.054	0.408	0.617					
	CF2	0.022	1.025	1.003	0.371	0.345	0.051	1.060	1.009	0.393	0.648					
	CF3	0.726	0.898	0.172	0.050	0.813	0.547	0.574	0.027	0.006	0.558					
	CF4	0.218	1.568	1.350	0.340	0.728	0.843	2.404	1.562	0.401	1.522					
	CF5	4.727	13.719	8.991	3.559	7.781	0.713	5.872	5.159	1.672	4.100					
	CF6	0.787	4.556	3.770	1.045	1.968	0.326	2.828	2.502	0.690	1.273					
	CF7	5.745	19.021	13.276	2.088	6.645	7.721	13.460	5.739	1.796	11.706					
	CF8	0.093	1.704	1.610	0.360	0.888	0.013	1.258	1.245	0.378	0.358	0.066	1.420	1.354	0.397	0.502
	CF9	0.031	1.409	1.378	0.279	0.899	0.022	2.098	2.076	0.465	0.294	0.029	1.216	1.186	0.340	0.469
	CF10	0.507	2.614	2.107	0.562	1.329	0.448	1.678	1.230	0.360	0.917	1.157	3.117	1.960	0.485	2.067

Table 5.4. Effect of nArchive with constant Number of iterations 50, nVar 30, nPop 50

		Objective #1					Objective #2					Objective #3				
nArchive	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
	UF1	0.058	1.115	1.057	0.421	0.609	0.029	1.048	1.020	0.402	0.459					
	UF2	0.030	1.132	1.102	0.289	0.334	0.095	1.184	1.089	0.309	0.693					
	UF3	0.754	0.972	0.218	0.060	0.860	0.537	0.696	0.159	0.049	0.599					
	UF4	0.739	0.878	0.139	0.044	0.825	0.580	1.080	0.500	0.138	0.680					
25	**UF5**	1.322	2.160	0.837	0.228	1.500	1.318	1.960	0.643	0.174	1.688					
	UF6	0.740	0.761	0.021	0.006	0.746	0.606	0.609	0.003	0.001	0.607					
	UF7	0.027	0.975	0.948	0.239	0.315	0.133	1.069	0.937	0.236	0.771					
	UF8	0.055	1.097	1.041	0.360	0.754	0.011	0.240	0.229	0.069	0.118	0.129	1.300	1.171	0.379	0.707
	UF9	0.034	0.975	0.941	0.277	0.290	0.061	1.212	1.151	0.247	0.298	0.533	1.324	0.792	0.248	0.854

Table 5.4. Effect of nPop with constant Number of iterations 100, nVar 50, nArchive 50 (continued)

		Objective #1					Objective #2					Objective #3				
nArchive	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
	UF10	1.244	2.284	1.040	0.216	1.462	1.100	2.970	1.870	0.364	1.243	1.357	2.196	0.839	0.158	1.932
	CF1	0.039	1.051	1.012	0.393	0.468	0.077	1.116	1.039	0.403	0.582					
	CF2	0.068	1.117	1.049	0.420	0.502	0.045	1.055	1.010	0.411	0.558					
	CF3	0.729	0.925	0.195	0.065	0.799	0.592	0.893	0.302	0.118	0.717					
	CF4	0.637	1.090	0.453	0.178	0.833	1.066	1.552	0.486	0.185	1.313					
	CF5	8.358	19.049	10.691	3.311	12.571	0.963	3.169	2.206	0.677	1.832					
	CF6	0.892	1.862	0.970	0.271	1.134	1.036	2.451	1.414	0.429	1.808					
	CF7	8.837	20.434	11.597	4.183	14.711	4.418	14.908	10.490	3.282	9.080					
	CF8	0.057	1.264	1.207	0.310	0.950	0.022	0.514	0.491	0.127	0.160	0.065	1.175	1.110	0.347	0.491
	CF9	0.112	1.483	1.371	0.275	1.150	0.059	1.256	1.197	0.263	0.301	0.069	1.076	1.006	0.262	0.319
	CF10	0.578	2.735	2.156	0.607	1.629	0.775	2.638	1.864	0.548	1.681	0.645	2.684	2.039	0.756	1.771
	UF1	0.018	1.084	1.066	0.374	0.517	0.056	1.111	1.055	0.380	0.511					
	UF2	0.025	1.307	1.282	0.327	0.382	0.073	1.196	1.123	0.322	0.651					
	UF3	0.803	1.162	0.359	0.099	0.923	0.457	0.828	0.371	0.105	0.603					
	UF4	0.828	0.887	0.059	0.021	0.857	0.509	0.551	0.041	0.013	0.531					
	UF5	1.089	2.135	1.046	0.270	1.904	0.934	2.579	1.644	0.509	1.436					
	UF6	0.746	0.806	0.061	0.015	0.760	0.640	0.799	0.158	0.055	0.722					
	UF7	0.041	1.042	1.001	0.270	0.360	0.060	1.070	1.010	0.257	0.767					
50	**UF8**	0.037	1.500	1.463	0.246	1.027	0.019	0.653	0.635	0.192	0.279	0.048	1.108	1.060	0.267	0.294
	UF9	0.022	0.773	0.751	0.198	0.312	0.010	0.687	0.677	0.189	0.197	0.557	1.100	0.543	0.117	0.698
	UF10	0.766	3.606	2.840	0.815	1.986	0.844	2.754	1.911	0.439	1.530	0.717	3.584	2.867	0.684	2.025
	CF1	0.051	1.127	1.076	0.365	0.553	0.069	1.098	1.029	0.340	0.496					
	CF2	0.038	1.066	1.029	0.430	0.549	0.063	1.083	1.021	0.414	0.506					
	CF3	0.771	1.007	0.236	0.067	0.859	0.502	0.863	0.361	0.100	0.643					
	CF4	0.166	0.690	0.524	0.233	0.452	1.260	1.957	0.697	0.260	1.569					
	CF5	6.436	14.902	8.467	2.647	9.896	1.887	10.458	8.571	2.727	3.980					

Table 5.4. Effect of nPop with constant Number of iterations 100, nVar 50, nArchive 50 (continued)

		Objective #1					Objective #2					Objective #3				
nArchive	**Function**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**	**Min**	**Max**	**Range**	**St.D.**	**Mean**
	CF6	0.756	4.406	3.650	1.249	1.898	0.612	3.462	2.850	0.778	1.506					
	CF7	7.273	15.142	7.869	2.040	10.865	6.191	19.823	13.632	4.944	9.871					
	CF8	0.683	1.469	0.786	0.197	1.062	0.052	1.176	1.125	0.285	0.375	0.025	0.526	0.501	0.125	0.178
	CF9	0.043	1.450	1.407	0.277	1.003	0.023	0.902	0.879	0.255	0.425	0.024	1.135	1.110	0.207	0.151
	CF10	0.715	2.131	1.416	0.435	1.375	0.437	1.602	1.165	0.337	0.932	0.678	2.808	2.130	0.511	1.766
75	**UF1**	0.036	1.118	1.082	0.411	0.510	0.025	1.073	1.048	0.392	0.496					
	UF2	0.059	1.534	1.475	0.406	0.573	0.043	1.291	1.248	0.351	0.574					
	UF3	0.749	0.918	0.169	0.052	0.817	0.528	0.857	0.329	0.096	0.642					
	UF4	0.830	0.959	0.129	0.036	0.899	0.520	0.573	0.053	0.012	0.537					
	UF5	1.069	2.387	1.317	0.374	1.632	0.854	3.117	2.263	0.780	1.623					
	UF6	0.696	0.815	0.119	0.033	0.748	0.619	0.868	0.249	0.062	0.705					
	UF7	0.017	1.159	1.142	0.272	0.384	0.049	1.062	1.013	0.253	0.722					
	UF8	0.035	1.431	1.396	0.327	1.006	0.016	0.695	0.679	0.158	0.238	0.024	1.077	1.053	0.316	0.269
	UF9	0.020	0.931	0.911	0.234	0.335	0.031	0.535	0.504	0.149	0.182	0.526	0.888	0.363	0.087	0.697
	UF10	0.753	3.610	2.857	0.731	1.484	0.473	2.260	1.788	0.492	1.243	0.724	3.634	2.911	0.879	2.224
	CF1	0.047	1.176	1.128	0.401	0.499	0.029	1.153	1.124	0.412	0.585					
	CF2	0.037	1.068	1.031	0.359	0.459	0.056	1.060	1.004	0.362	0.569					
	CF3	0.755	1.079	0.325	0.110	0.887	0.482	0.812	0.330	0.106	0.585					
	CF4	0.207	1.281	1.073	0.353	0.895	0.308	2.025	1.717	0.601	0.995					
	CF5	3.991	15.278	11.286	4.214	8.201	1.035	7.330	6.296	1.845	3.908					
	CF6	0.854	6.676	5.822	2.001	3.236	0.432	3.894	3.462	1.124	1.538					
	CF7	8.017	20.943	12.926	5.777	15.228	4.181	9.934	5.753	1.912	6.063					
	CF8	0.043	1.405	1.361	0.231	1.045	0.031	1.707	1.675	0.314	0.368	0.027	1.045	1.017	0.264	0.236
	CF9	0.014	1.402	1.387	0.362	0.922	0.022	1.001	0.980	0.187	0.302	0.029	1.226	1.197	0.333	0.271
	CF10	0.385	3.328	2.942	0.748	1.612	0.700	3.052	2.352	0.599	1.780	1.007	3.079	2.071	0.561	2.026

6. CONCLUSION

In conclusion, we have studied the influence of the optimization ability of SPEA2 on different benchmark functions by evaluating them using different performance metrics. The benchmark functions used include 10 constrained functions (CF's) and 10 unconstrained functions (UF's). The results of the performance metrics running several experiments was obtained by varying parameters such as number of iterations, variable size, population and archives. Inclusion of these benchmark functions were implemented successfully in the code and performance studies were conducted.

We observed that an increase in the number of iterations decreased the standard deviation error proving that Pareto optimal values are obtained at high numbers of iterations. An increase in the number of variable size was also able to reduce the standard deviation value dramatically. With the increase in the number of population size, we observed a significant increase in the standard deviation suggesting that an optimal value is obtained at lower population sizes or there should be more number of iterations needed at higher population sizes to have reduced standard deviation values. Additionally, an increase in the archive size also decreases the standard deviation suggesting that archive size is essential for obtaining optimal solutions.

To extend this work, our next step is to compare the performance of benchmark functions of SPEA2 to that of other evolution algorithms. This will allow us to comment on the overall performance of several algorithms. Also, applying the SPEA2 algorithm to real world problems such as financial time series will be a potential future aspect such as Niched Pareto Genetic Algorithm (NGPA), which has already been used to find patterns in this field.

REFERENCES

[1] Schaffer, J. D. (1984). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. Unpublished Ph.D. thesis, Vanderbilt University, Nashville, Tennessee.

[2] Deb, K. (1999). Multi-objective genetic algorithms: Problem difficulties and construction of test problems. Evolutionary Computation, 7(3):205–230

[3] Horn, J. and Nafpliotis, N. (1993). Multiobjective optimization using the niched pareto genetic algorithm. IlliGAL Technical Report 93005, Illinois Genetic Algorithms Laboratory, University of Illinois, Urbana, Illinois.

[4] Horn, J., Nafpliotis, N. and Goldberg, D. E. (1994). A niched pareto genetic algorithm for multiobjective optimization. In Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congres

[5] Srinivas, N. and Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation, 2(3):221–248

[6] Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation, 3(4):257–271

[7] Knowles, J. and Corne, D. (1999). The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation, Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, vol. 1. IEEE.

[8] Zitzler, E., Laumanns, M. and Thiele, L. (2002). SPEA2 : Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou, and T. Fogarty, editors, Evolutionary Methods for Design, Optimisation, and Control, pages 19–26, Barcelona, Spain. CIMNE.

[9] Zitzler, E., Laumanns, M. and Thiele, L. (2004). A tutorial on evolutionary multi-objective optimization. In Xavier Gandibleux, Marc Sevaux, Kenneth Sšrensen and T'kindt, Vincent, editors, Metaheuristics for Multiobjective Optimisation, chapter 1, pages 1–32-Springer Verlag.

[10] Huang, V. L., Qin, A. K., Deb, K., Zitzler, E., Suganthan, P. N., Liang, J. J., Preuss, M. and Huband, S. (2007). "Problem definitions for performance assessment of multi-objective optimization algorithms," Nanyang Technological University, Singapore, Tech. Rep.

[11] Deb, K., Pratap, A. and Meyarivan, T. (2000). Constrained test problems for multiobjective evolutionary computation, KanGAL, Tech. Rep.

[12] Jakob, W., Gorges-Schleuter, M., and Blume, C. (1992). Application of genetic algorithms to task planning and learning. In (Manner and Manderick, 1992), pages 291-300

[13] Richardson, J. T., Palmer, M. R., Liepins, G., and Hilliard, M. (1989). Some guidelines for genetic algorithms with penalty functions. In (Schaer,1989), pages 191-197

[14] Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. In (Grefenstette, 1985), pages 93-100

[15] Fourman, M. P. (1985). Compaction of symbolic layout using genetic algorithms. In (Grefenstette, 1985), pages 141-153

[16] Ben-Tal, A. (1980). Characterization of Pareto and lexicographic optimal solutions. In (Fandel and Gal, 1980), pages 1-11

[17] Hajela, P. and Lin, C.-Y. (1992). Genetic search strategies in multicriterion optimal design. Structural Optimization, 4:99-107

[18] Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading, Massachusetts.

[19] Fonseca, C. M. and Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In (Forrest, 1993), pages 416-423

[20] Parks, G.T & Miller, I. (1998). Selective breeding in a multi-objective genetic algorithm. Proceedings of the Parallel Problem Solving from Nature 5th Workshop (PPSN-V), Belgium, 250-259

[21] Zitzler, E. & Thiele, L. (1998). An evolutionary algorithm for multi-objective optimization: The strength Pareto approach, Technical report 43, Zurich, Switzerland: Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology. (ETH).

[22] Zitzler, E., K. Deb, and L. Thiele (2000).Comparison ofmultiobjective evolutionary algorithms: Empirical results. Evolutionary Computation 8(2), 173–195