

Finding Sets of Non-Dominated Solutions with High Spread and Well-Balanced Distribution using Generalized Strength Pareto Evolutionary Algorithm

Filip Rudziński¹

¹Department of Electrical and Computer Engineering, Kielce University of Technology,
Al. 1000-lecia P.P. 7, 25-314 Kielce, Poland

Abstract

The paper presents a generalization of the Strength Pareto Evolutionary Algorithm 2 (SPEA2) and its application in selected well-known two- and three-objective optimization benchmark problems. The proposed solution is referred to as our SPEA3. The generalization consists in the exchange of the environmental selection procedure in SPEA2 for a new original algorithm which aims to determine the final non-dominated solutions with a high spread and well-balanced distribution in the objective space. During the evolutionary optimization process, the non-dominated solutions are gradually incorporated into the resulting set and placed in it in such a way that the distances between them and their nearest neighbors in the objective space are the greatest possible. A comparative analysis with alternative multi-objective optimization techniques shows that our approach is superior with regard to the spread and distribution of solutions while being still competitive with regard to their accuracy.

Keywords: multi-objective optimization, evolutionary computation, well spread and balanced distribution of non-dominated solutions

1. Introduction

The requirements for modern multi-objective optimization algorithms (MOEA) [1] that are to be applied in a multi-criteria decision-making process focus on finding a set of the so-called *non-dominated* solutions, which satisfy the following properties [2]. First, the *accuracy* of all solutions in the set, with respect to the optimal solutions, must be as high as possible. Then, a *spread* of the set (i.e., the distance between the extreme solutions) must be as wide as possible in order to achieve the solutions located on the boundaries of the objective space. Furthermore, the solutions must have a satisfactory *distribution* in the set (i.e., the distances between the nearest neighboring solutions in the objective space must be as similar as possible) in order to provide well-balanced levels of compromise between the values of particular optimization objectives. Finally, the

obtained set must contain a balanced number of solutions (neither too many nor too few) in order to give a flexible choice of solutions for a decision maker.

In this paper, a generalization of the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [3] addressing all of the above-mentioned requirements and its application in selected well-known two- and three-objective optimization benchmark problems are presented. The proposed solution is referred to as our SPEA3. The essence of generalization consists in the exchange of the environmental selection procedure in SPEA2 for a new original algorithm which aims to determine the final non-dominated solutions with a high spread and well-balanced distribution in the objective space. During the evolutionary optimization, the non-dominated solutions are gradually incorporated into the resulting set and placed in it in such a way that the distances between them and their nearest neighbors in the objective space are the greatest possible. The proposed algorithm still preserves the main advantages of the SPEA2 approach, i.e., it generates the relatively accurate final solutions and enables to determine (independently from the size of a genetic population) the number of those solutions according to the needs.

A comparative analysis of our approach with three well-known alternative multi-objective optimization techniques is carried out to demonstrate that our approach is superior with regard to the spread and distribution of solutions while being still competitive with regard to their accuracy.

2. Basic notions

An optimization of a multi-criteria decision-making problem aiming to minimize n objective functions $f_k(\mathbf{x})$ ($k = 1, 2, \dots, n$), where $\mathbf{x} = [x_1, x_2, \dots, x_m]^T$ is a vector of m decision variables $x_i \in \mathbb{R}$ ($i = 1, 2, \dots, m$) is considered. Any vector \mathbf{x} satisfying p inequality constraints $g_j(\mathbf{x}) \geq 0$ ($j = 1, 2, \dots, p$) and r equality constraints $h_j(\mathbf{x}) = 0$ ($j = 1, 2, \dots, r$) of the problem represents its *feasible solution* Ω . The set of all feasible solutions

defines the search space Ω .

All feasible solutions, which are not dominated (see Definition 2.1) by any other solution in the search space are called Pareto-optimal. They form the so-called Pareto-optimal set (or Pareto-optimal front) $\mathbf{F}_{OPT} \subseteq \Omega$. The task is to determine an approximation of the Pareto-optimal set in the form of the set \mathbf{A} with a limited number of non-dominated solutions.

Definition 2.1. *Solution S_1 dominates the other solution S_2 ($S_1 \succ S_2$), if and only if $\forall k : f_k(\mathbf{x}_{S_1}) \leq f_k(\mathbf{x}_{S_2})$ and $\exists k : f_k(\mathbf{x}_{S_1}) < f_k(\mathbf{x}_{S_2})$, where $f_k(\mathbf{x})$ are the objective functions ($k = 1, 2, \dots, n$) and \mathbf{x}_{S_1} , \mathbf{x}_{S_2} are vectors of m decision variables representing S_1 and S_2 , respectively.*

Definition 2.2. *Solution S_1 weakly dominates the other solution S_2 ($S_1 \succeq S_2$), if and only if $\forall k : f_k(\mathbf{x}_1) \leq f_k(\mathbf{x}_2)$, where $f_k(\mathbf{x})$ ($k = 1, 2, \dots, n$) as well as \mathbf{x}_1 and \mathbf{x}_2 are as in Definition 2.1.*

In general, the quality of the obtained set \mathbf{A} (thus, the performance of an optimization algorithm which determines the set \mathbf{A}) can be evaluated with respect to the Pareto-optimal set \mathbf{F}_{OPT} . However, in practical experiments, the Pareto-optimal set is usually unknown or its designation is not possible due to, e.g., infinite number of Pareto-optimal solutions. Thus, the evaluation of \mathbf{A} can be performed using a *reference set* \mathbf{F}_{REF} , which contains a finite number of Pareto-optimal solutions (or solutions that are as close to them as possible) obtained in an artificial way (e.g., using an external optimization technique or by mathematical analysis of the decision problem). Certain reference sets concerning well-known benchmark tests are available on the jMetal Web Site (<http://jmetal.sourceforge.net/problems.html>).

A broad review of the performance indices can be found in [2]. In this work, two most often applied indices are used in order to evaluate the accuracy, spread and distribution of the set \mathbf{A} . The accuracy measure is the generational distance (GD) [4]:

$$GD(\mathbf{A}, \mathbf{F}) = \frac{1}{|\mathbf{A}|} \left(\sum_{i=1}^{|\mathbf{A}|} d_{AF_i}^q \right)^{1/q}, \quad (1)$$

where $|\mathbf{A}|$ stands for the size of the set \mathbf{A} , the set \mathbf{F} denotes \mathbf{F}_{OPT} or \mathbf{F}_{REF} , and d_{AF_i} is the distance between solution $A_i \in \mathbf{A}$ and its nearest Pareto-optimal solution in the set \mathbf{F} :

$$d_{AF_i} = \min_{F_j \in \mathbf{F}} d(A_i, F_j). \quad (2)$$

In general case, $d(S_1, S_2)$ denotes the distance between solutions S_1 and S_2 in the objective space. For $q = 2$, d_{AF_i} is based on the Euclidean norm:

$$d(S_1, S_2) = \sqrt{\sum_{k=1}^n (f_k(\mathbf{x}_{S_1}) - f_k(\mathbf{x}_{S_2}))^2}, \quad (3)$$

where the vectors \mathbf{x}_{S_1} and \mathbf{x}_{S_2} represent the solutions S_1 and S_2 , respectively.

The spread and distribution of obtained solutions in the set \mathbf{A} are measured simultaneously using the generalization of Δ metric proposed in [1]:

$$\Delta(\mathbf{A}, \mathbf{F}) = \frac{\sum_{k=1}^n d_{EX_k} + \sum_{i=1}^{|\mathbf{A}|-1} (d_{NR_i} - \bar{d}_{NR})}{\sum_{k=1}^n d_{EX_k} + (|\mathbf{A}| - 1)\bar{d}_{NR}}. \quad (4)$$

\bar{d}_{NR} is the average of d_{NR_i} ($i = 1, 2, \dots, |\mathbf{A}| - 1$) whilst d_{NR_i} is the distance between solution $A_i \in \mathbf{A}$ and its nearest neighbor in the set \mathbf{A} :

$$d_{NR_i} = \min_{\substack{A_j \in \mathbf{A}, \\ j=1,2,\dots,|\mathbf{A}|; i \neq j}} d(A_i, A_j). \quad (5)$$

In turn, d_{EX_k} is the distance between the extreme solutions $A_{EX_k} \in \mathbf{A}$ and $F_{EX_k} \in \mathbf{F}$:

$$d_{EX_k} = d(A_{EX_k}, F_{EX_k}), \quad (6)$$

where $A_{EX_k} = \arg \max_{A_j \in \mathbf{A}} f_k(\mathbf{x}_{A_j})$ and analogously for F_{EX_k} .

3. The proposed SPEA3 approach

First, the SPEA2 algorithm (see Algorithm 1 below), which is the starting point of our SPEA3 approach, is outlined.

Algorithm 1: SPEA2 (Main loop) [3]

- Input: N (population size)
 \bar{N} (archive size)
 T (maximum number of generations)
- Output: \mathbf{A} (non-dominated set)
- Step 1: **Initialization:** Generate an initial population \mathbf{P}_0 and create the empty archive (external set) $\bar{\mathbf{P}}_0 = \emptyset$. Set $t = 0$.
- Step 2: **Fitness assignment:** Calculate fitness values of individuals in \mathbf{P}_t and $\bar{\mathbf{P}}_t$ (cf. Section 3.1 in [3]).
- Step 3: **Environmental selection:** Copy all non-dominated individuals in \mathbf{P}_t and $\bar{\mathbf{P}}_t$ to $\bar{\mathbf{P}}_{t+1}$. If size of $\bar{\mathbf{P}}_{t+1}$ exceeds N then reduce $\bar{\mathbf{P}}_{t+1}$ by means of the truncation operator, otherwise if size of $\bar{\mathbf{P}}_{t+1}$ is less than N then fill $\bar{\mathbf{P}}_{t+1}$ with dominated individuals in \mathbf{P}_t and $\bar{\mathbf{P}}_t$ (cf. Section 3.2 in [3]).
- Step 4: **Termination:** If $t \geq T$ or another stopping criterion is satisfied then set \mathbf{A} to the set of decision vectors represented by the non-dominated individuals in $\bar{\mathbf{P}}_{t+1}$. Stop.
- Step 5: **Mating selection:** Perform binary tournament selection with replacement on $\bar{\mathbf{P}}_{t+1}$ 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.

The SPEA2 (and thus also our SPEA3 approach) uses an archive (an external set) of final solutions whose size can be regulated depending on the needs and adjusted, in a flexible way, to the requirements of a decision maker. The proposed generalization of SPEA2 consists in the exchange of its environmental selection procedure - which is responsible for selecting non-dominated solutions from the population to the archive (see Step 3 in Algorithm 1) - for a new original algorithm (Step 3, and Steps 3a-e in Algorithm 2) which aims to determine the final non-dominated solutions with a high spread and well-balanced distribution in the objective space.

Algorithm 2: SPEA3 (Main loop)

Input: N (population size)
 \bar{N} (archive size)
 T (maximum number of generations)

Output: \mathbf{A} (non-dominated set)

Step 1: } As in Algorithm 1 (SPEA2).
Step 2: }

Step 3: **Environmental selection:** If \bar{P}_t is empty then copy three randomly selected individuals from P_t to \bar{P}_{t+1} and go to Step 4, otherwise follow the Steps 3a-e.

Step 3a: Make auxiliary archive \bar{Q}_{t+1} by copying all non-dominated individuals in P_t , which are also not weakly dominated by at least one individual in P_t .

Step 3b: Copy all individuals in \bar{P}_t to \bar{P}_{t+1} and if size of \bar{P}_{t+1} does not exceed \bar{N} then replenish \bar{P}_{t+1} with auxiliary individuals from \bar{Q}_{t+1} using Sub-algorithm 2a.

Step 3c: Replace all individuals in \bar{P}_{t+1} which are dominated by at least one individual in \bar{Q}_{t+1} with their nearest neighbors belonging to \bar{Q}_{t+1} . Every time when the nearest neighbor replaces an individual in \bar{P}_{t+1} , it must be immediately removed from \bar{Q}_{t+1} .

Step 3d: Minimize the distance differences between individuals in archive \bar{P}_{t+1} using Sub-algorithm 2b.

Step 3e: Clear all dominated individuals in archive \bar{P}_{t+1} and go to Step 4.

Step 4: }
Step 5: } As in Algorithm 1 (SPEA2).
Step 6: }

Two main activities of Algorithm 2 can be distinguished. The first activity (Sub-algorithm 2a) replenishes the archive by gradually adding auxiliary solutions selected from the population (starting with three randomly selected solutions and ending when the archive contains the desired number of solutions).

Sub-algorithm 2a (Replenishment of archive)

Input: \bar{N} (archive size)
 \bar{P}_{t+1} (the main archive)
 \bar{Q}_{t+1} (the auxiliary archive)

Output: \bar{P}_{t+1} (the main archive)
 \bar{Q}_{t+1} (the auxiliary archive)

Step 1: Make new archive \bar{Z}_{t+1} as a copy of \bar{P}_{t+1} .

Step 2: If \bar{Z}_{t+1} is empty or size of \bar{P}_{t+1} exceeds \bar{N} then stop the algorithm.

Step 3: Determine individual \bar{Z}_i in \bar{Z}_{t+1} with the greatest distance to its nearest neighbor \bar{Z}_j ($i \neq j$; $i, j \in \{1, 2, \dots, |\bar{Z}_{t+1}|\}$).

Step 4: Select an auxiliary individual \bar{Q}_{AUX} in \bar{Q}_{t+1} for which an absolute value of the difference between distances from \bar{Q}_{AUX} to \bar{Z}_i and to \bar{Z}_j is the smallest.

Step 5: If the distance between \bar{Z}_i and \bar{Q}_{AUX} is greater than the distance between \bar{Z}_i and \bar{Z}_j then remove \bar{Z}_i from \bar{Z}_{t+1} and go to Step 2.

Step 6: If the distance between \bar{Z}_j and \bar{Q}_{AUX} is greater than the distance between \bar{Z}_i and \bar{Z}_j then remove \bar{Z}_j from \bar{Z}_{t+1} and go to Step 2.

Step 7: Move auxiliary individual \bar{Q}_{AUX} from \bar{Q}_{t+1} to \bar{P}_{t+1} . Clear \bar{Z}_{t+1} (or alternatively, remove only \bar{Z}_i and \bar{Z}_j from \bar{Z}_{t+1} , in order to fill \bar{P}_{t+1} with more auxiliary individuals) and go to Step 2.

The second activity (Sub-algorithm 2b) gradually relocates the archived solutions in the objective space in such a way that the distances between them and their nearest neighbors are the greatest possible.

Sub-algorithm 2b (Minimization of distance differences between individuals in archive)

Input: \bar{P}_{t+1} (the main archive)
 \bar{Q}_{t+1} (the auxiliary archive)

Output: \bar{P}_{t+1} (the main archive)
 \bar{Q}_{t+1} (the auxiliary archive)

Step 1: If size of \bar{P}_{t+1} does not exceed 1 or \bar{Q}_{t+1} is empty then stop the algorithm, otherwise set individual counter $i = 1$.

- Step 2: Select i -th individual \bar{P}_i in archive $\bar{\mathbf{P}}_{t+1}$.
Make new archive $\bar{\mathbf{Z}}_{t+1}$ as a copy of $\bar{\mathbf{Q}}_{t+1}$.
- Step 3: If $\bar{\mathbf{Z}}_{t+1}$ is empty then go to Step 8.
- Step 4: Among members of $\bar{\mathbf{Z}}_{t+1}$ determine individual \bar{Z}_{NN} which is the nearest neighbor of \bar{P}_i .
- Step 5: Among members of $\bar{\mathbf{P}}_{t+1}$ determine individual \bar{P}_j which is the nearest neighbor of \bar{Z}_{NN} , such that $i \neq j$.
- Step 6: If the distance between \bar{Z}_{NN} and \bar{P}_j is greater than the distance between \bar{P}_i and \bar{P}_j then replace \bar{P}_i with \bar{Z}_{NN} in $\bar{\mathbf{P}}_{t+1}$.
- Step 7: Remove \bar{Z}_{NN} from $\bar{\mathbf{Z}}_{t+1}$ and go to Step 3.
- Step 8: Increment individual counter ($i = i + 1$).
If i exceeds the size of $\bar{\mathbf{P}}_{t+1}$ then stop the algorithm, otherwise go to Step 2.

It is important, that all distances between any solutions in the objective space are calculated using the metric (3) and the rescaled (into the range $[0, 1]$) objective functions $\bar{f}_k(\mathbf{x}_S)$, i.e.:

$$\bar{f}_k(\mathbf{x}_S) = \frac{f_k(\mathbf{x}_S) - f_{MIN_k}}{f_{MAX_k} - f_{MIN_k}}, \quad (7)$$

where f_{MIN_k} and f_{MAX_k} are the extreme values of the k -th objective function found from among evaluations of all solutions in the archive.

4. Experimental results

To show the operation of the proposed technique, 12 benchmark tests (see Table 1) representing different levels of complexity of decision problems are used. The first nine benchmarks (i.e., *SCH*, *FON*, *KUR*, *ConstrEx*, and a collection of *ZDT*s tests) are two-objective and the remaining ones (*DTLZ1*, *DTLZ2*, and *DTLZ3*) are three-objective problems.

The *SCH* benchmark is a basic test that should not provide any difficulties for the optimization techniques. The extreme solutions in the *FON* test are difficult to reach due to the irrational values of the objective functions of the boundary optimal solutions. The *KUR* test has a discontinuous Pareto-optimal front. The benchmark *ConstrEx* is a constrained test with unbalanced ranges of values of the objective functions. The *ZDT* and *DTLZ* benchmark collections represent optimization problems with complex, multimodal objective functions.

For the purpose of comparative analysis, the above benchmark tests are also used to test three well-known alternative multi-objective optimization techniques, i.e., the aforementioned SPEA2 [3], the Nondominated Sorting Genetic Algorithm II (NSGA-II) [5], and its modification ε -NSGA-II [6].

All experiments have been performed using genetic algorithms with populations of 200 binary-coded individuals (32-bit-strings) and the probabilities of crossover and mutation operations equal to

0.7 and 0.5, respectively. The archive size is equal to 50. The number of generations in all experiments is equal to 10000.

Figs. 1–4 show the exemplary final approximations of Pareto-optimal fronts of three selected two-objective problems, listed in Table 1. Analogously, Figs. 5 and 6 relate to the three-objective problems. The results are obtained with the use of our SPEA3 approach (part a) of Figs. 1–6) and the alternative techniques (parts b) and c) of Figs. 1–6).

It can clearly be seen that our approach determines the final solutions with the best spread and distribution in comparison to the remaining techniques, in all cases of the considered benchmark tests. For example, in the case of *SCH* and *DTLZ2* tests, the solutions from the final sets of ε -NSGA-II approach are arranged, in a more compact way, in the middle of the objective space and rarer in the extreme areas. In turn, in the case of *ConstrEx* test, the solutions of both approaches SPEA2 and ε -NSGA-II are distributed rarer in the lower part of the objective space than at the top.

A comparative analysis of our approach with the alternative techniques confirms the above findings. Table 2 contains the averaged results of 10-fold validations of each of the possible pairs of the considered benchmark tests and algorithms. In terms of determining the solutions with a high spread and well-balanced distribution, our approach is superior for 11 of 12 benchmark tests. In the case of the *KUR* problem only, our approach is worse than ε -NSGA-II and SPEA2. This may be due to the fact that our technique determines the solutions that are uniformly distributed within a single section along the discontinuous Pareto-optimal front. Concerning the accuracy of obtained solutions, our approach wins in 4 cases of the tests (i.e., *FON*, *ZDT2*, *ZDT4*, and *ZDT6*) whilst in the remaining 8 cases the results are comparable.

5. Conclusions

The generalization of the Strength Pareto Evolutionary Algorithm 2 (SPEA2) - referred to as SPEA3 - consisting in the exchange of the environmental selection procedure in SPEA2 for a new original algorithm which aims to determine the final non-dominated solutions with a high spread and well-balanced distribution in the objective space has been proposed.

The performance of our SPEA3 approach has been tested with the use of 12 well-known two- and three-objective optimization benchmark problems. The experimental results show that our approach generates better - in terms of the spread and distribution - non-dominated solutions than alternative techniques (SPEA2, NSGA-II, ε -NSGA-II), while remaining competitive in terms of the accuracy of those solutions.

Table 1: Formulation of the benchmark problems

Benchmark	m	n	Objective functions	Search space	Optimal solutions
<i>SCH</i>	1	2	$f_1(x) = x^2$ $f_2(x) = (x - 2)^2$	$x \in [-10^3, 10^3]$	$x \in [0, 2]$
<i>FON</i>	3	2	$f_1(\mathbf{x}) = 1 - \exp(-\sum_{i=1}^3 (x_i - \frac{1}{\sqrt{3}})^2)$ $f_2(\mathbf{x}) = 1 - \exp(-\sum_{i=1}^3 (x_i + \frac{1}{\sqrt{3}})^2)$	$x_i \in [-4, 4]$, $i = 1, 2, 3$	$x_1 = x_2 = x_3$ $\in [-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]$
<i>KUR</i>	3	2	$f_1(\mathbf{x}) = \sum_{i=1}^{m-1} (-10 \exp(-0.2 \sqrt{x_i^2 + x_{i+1}^2}))$ $f_2(\mathbf{x}) = \sum_{i=1}^m (x_i ^{0.8} + 5 \sin(x_i^3))$	$x_i \in [-5, 5]$, $i = 1, 2, \dots, m$	(refer [1])
<i>ConstrEx</i>	2	2	$f_1(\mathbf{x}) = x_1$, with constraints: $9x_1 - x_2 \geq 1$, $f_2(\mathbf{x}) = \frac{1+x_2}{x_1}$, $9x_1 + x_2 \geq 6$	$x_1 \in [0.1, 1]$ $x_2 \in [0, 5]$	
<i>ZDT1</i>	30	2	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - \sqrt{\frac{f_1(\mathbf{x})}{g(\mathbf{x})}}]$, where $g(\mathbf{x}) = 1 + 9(\sum_{i=2}^m x_i)/(m-1)$	$x_i \in [0, 1]$ $i = 1, 2, \dots, m$	$x_1 \in [0, 1]$ $x_i = 0$, $i = 2, 3, \dots, m$
<i>ZDT2</i>	30	2	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2]$, where $g(\mathbf{x}) = 1 + 9(\sum_{i=2}^m x_i)/(m-1)$	$x_i \in [0, 1]$ $i = 1, 2, \dots, m$	$x_1 \in [0, 1]$ $x_i = 0$, $i = 2, 3, \dots, m$
<i>ZDT3</i>	30	2	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})(1 - \sqrt{\frac{f_1(\mathbf{x})}{g(\mathbf{x})}} - \frac{f_1(\mathbf{x})}{g(\mathbf{x})} \sin(10\pi f_1(\mathbf{x})))$, where $g(\mathbf{x}) = 1 + 9(\sum_{i=2}^m x_i)/(m-1)$	$x_i \in [0, 1]$ $i = 1, 2, \dots, m$	$x_1 \in [0, 1]$ $x_i = 0$, $i = 2, 3, \dots, m$
<i>ZDT4</i>	10	2	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - \sqrt{\frac{f_1(\mathbf{x})}{g(\mathbf{x})}}]$, where $g(\mathbf{x}) = 1 + 10(m-1) + \sum_{i=2}^m (x_i^2 - 10 \cos(4\pi x_i))$	$x_1 \in [0, 1]$ $x_i \in [-1, 5]$ $i = 2, 3, \dots, m$	$x_1 \in [0, 1]$ $x_i = 0$, $i = 2, 3, \dots, m$
<i>ZDT6</i>	10	2	$f_1(\mathbf{x}) = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $f_2(\mathbf{x}) = g(\mathbf{x})[1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2]$, where $g(\mathbf{x}) = 1 + 9[(\sum_{i=2}^m x_i)/(m-1)]^{0.25}$	$x_i \in [0, 1]$ $i = 1, 2, \dots, m$	$x_1 \in [0, 1]$ $x_i = 0$, $i = 2, 3, \dots, m$
<i>DTLZ1</i>	12	3	$f_1(\mathbf{x}) = 0.5x_1x_2(1 + g(\mathbf{x}))$ $f_2(\mathbf{x}) = 0.5x_1(1 - x_2)(1 + g(\mathbf{x}))$ $f_3(\mathbf{x}) = 0.5(1 - x_1)(1 + g(\mathbf{x}))$, where $g(\mathbf{x}) = 100[10 + \sum_{i=2}^m ((x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)))]$	$x_i \in [0, 1]$ $i = 1, 2, \dots, m$	$x_1, x_2 \in [0, 1]$ $x_i = 0.5$, $i = 3, 4, \dots, m$
<i>DTLZ2</i>	12	3	$f_1(\mathbf{x}) = (1 + g(\mathbf{x})) \cos(x_1\pi/2) \cos(x_2\pi/2)$ $f_2(\mathbf{x}) = (1 + g(\mathbf{x})) \cos(x_1\pi/2) \sin(x_2\pi/2)$ $f_3(\mathbf{x}) = (1 + g(\mathbf{x})) \sin(x_1\pi/2)$, where $g(\mathbf{x}) = \sum_{i=2}^m (x_i - 0.5)^2$	$x_i \in [0, 1]$ $i = 1, 2, \dots, m$	$x_1, x_2 \in [0, 1]$ $x_i = 0.5$, $i = 3, 4, \dots, m$
<i>DTLZ3</i>	12	3	$f_1(\mathbf{x}) = (1 + g(\mathbf{x})) \cos(x_1\pi/2) \cos(x_2\pi/2)$ $f_2(\mathbf{x}) = (1 + g(\mathbf{x})) \cos(x_1\pi/2) \sin(x_2\pi/2)$ $f_3(\mathbf{x}) = (1 + g(\mathbf{x})) \sin(x_1\pi/2)$, where $g(\mathbf{x}) = 100[10 + \sum_{i=2}^m ((x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)))]$	$x_i \in [0, 1]$ $i = 1, 2, \dots, m$	$x_1, x_2 \in [0, 1]$ $x_i = 0.5$, $i = 3, 4, \dots, m$

References

- [1] K. Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.
- [2] T. Okabe, Y. Jin, and B. Sendhoff. A critical survey of performance indices for multi-objective optimisation. In *Proceeding of 2003 Congress on Evolutionary Computation*, pages 878–885. IEEE Press, 2003.
- [3] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Proceeding of the Evolutionary Methods for Design, Optimisation, and Control*, pages 95–100. CIMNE, Barcelona, Spain, 2002.
- [4] D. A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Wright Patterson AFB, OH, USA, 1999.
- [5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 63:182–197, Apr 2002.
- [6] J.B. Kollat and P.M. Reed. Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design. *Advances in Water Resources*, 29(6):792–807, Apr 2006.

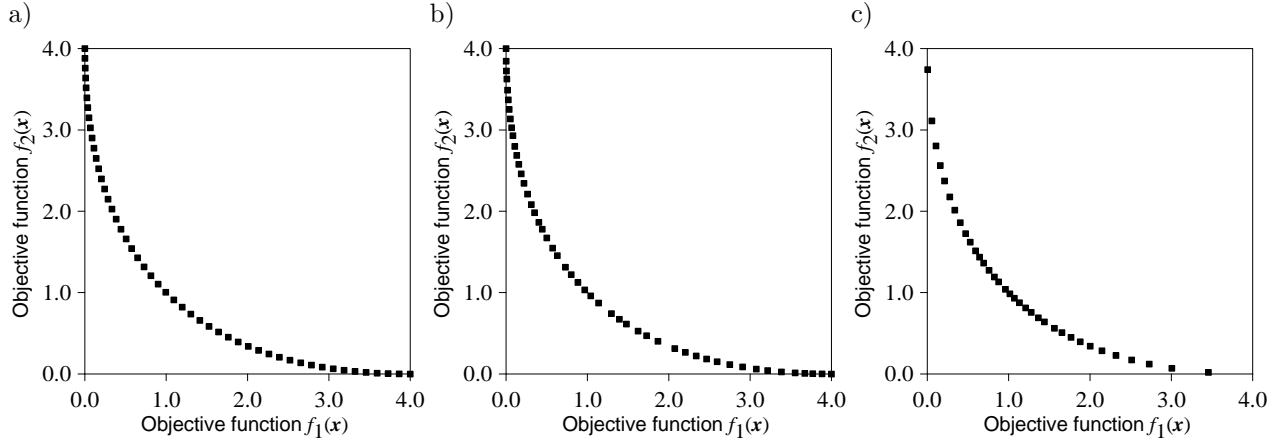


Figure 1: Final approximations of the Pareto-optimal solution sets for our SPEA3 (a), SPEA2 (b), and ε -NSGA-II (c) algorithms (*SCH* benchmark problem)

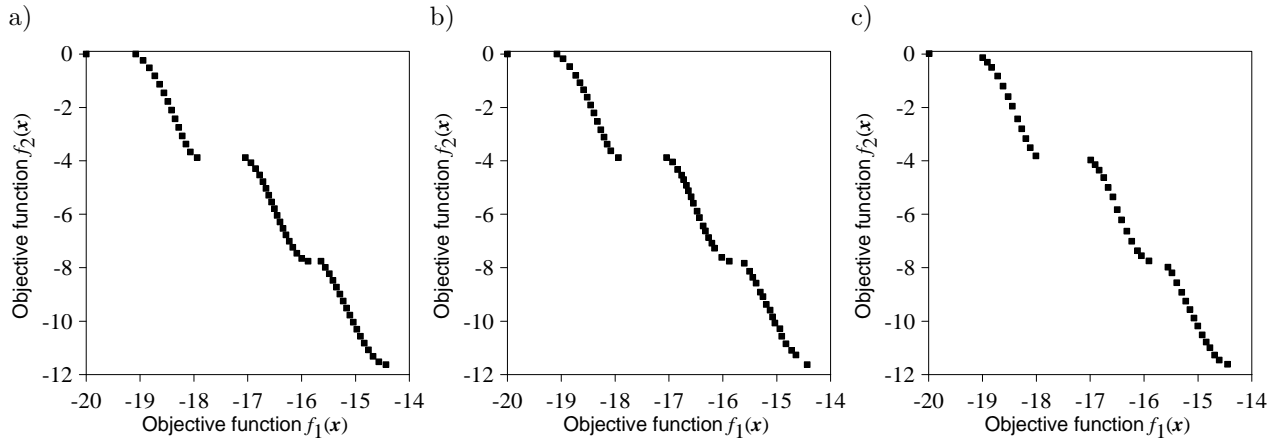


Figure 2: Final approximations of the Pareto-optimal solution sets for our SPEA3 (a), SPEA2 (b), and ε -NSGA-II (c) algorithms (*KUR* benchmark problem)

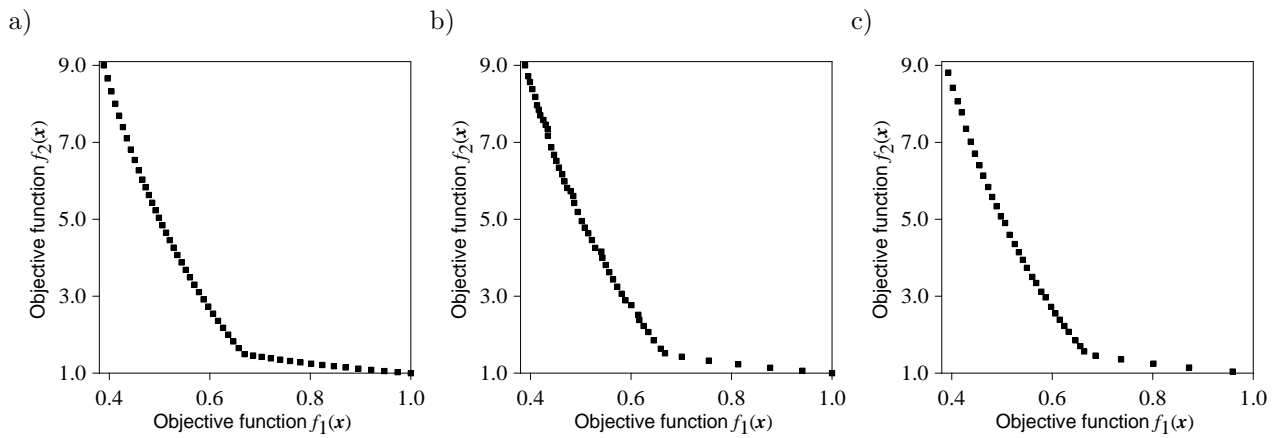


Figure 3: Final approximations of the Pareto-optimal solution sets for our SPEA3 (a), SPEA2 (b), and ε -NSGA-II (c) algorithms (*ConstrEx* benchmark problem)

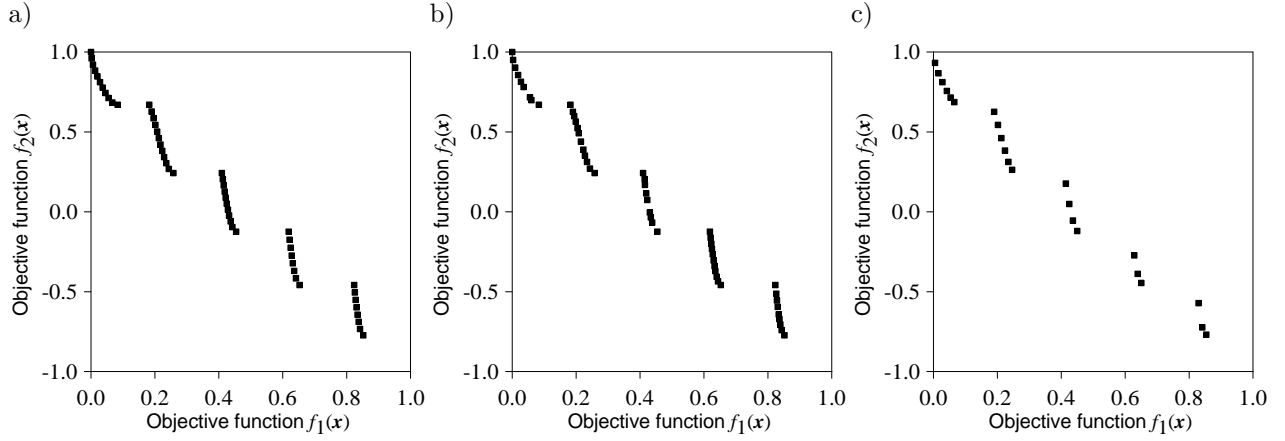


Figure 4: Final approximations of the Pareto-optimal solution sets for our SPEA3 (a), SPEA2 (b), ε -NSGA-II (c) algorithms (*ZDT3* benchmark problem)

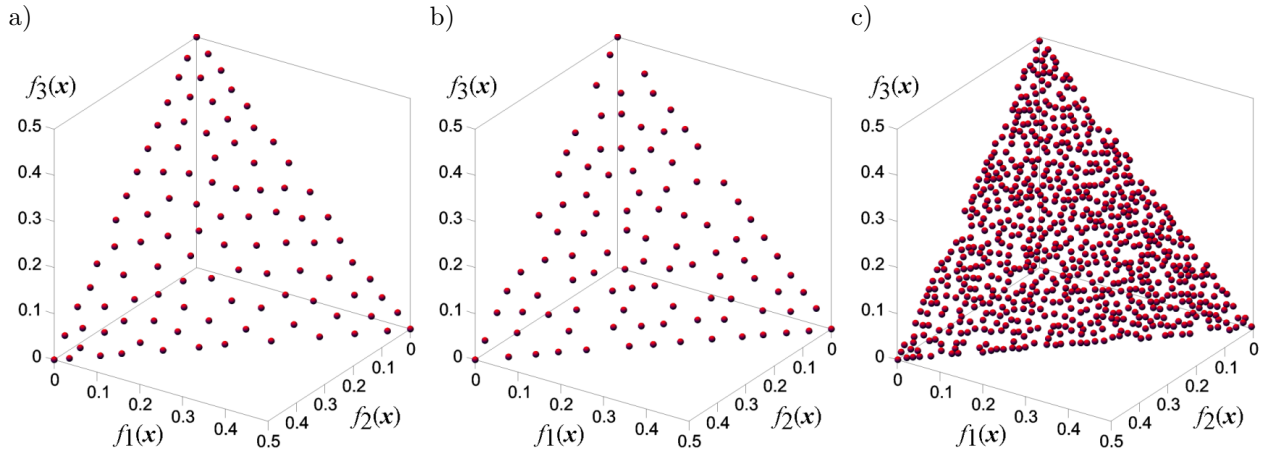


Figure 5: Final approximations of the Pareto-optimal solution sets for our SPEA3 (a), SPEA2 (b), ε -NSGA-II (c) algorithms (*DTLZ1* benchmark problem)

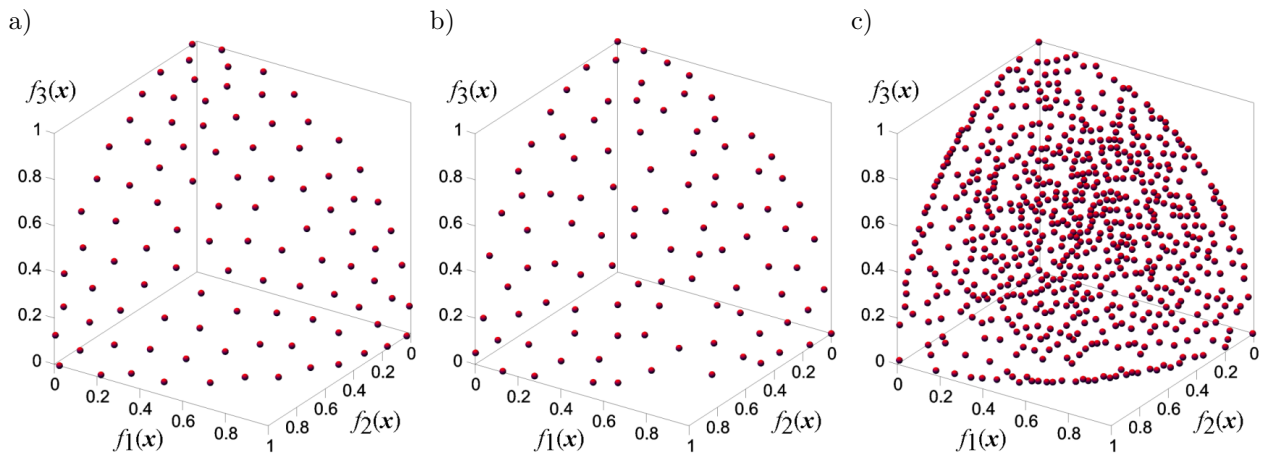


Figure 6: Final approximations of the Pareto-optimal solution sets for our SPEA3 (a), SPEA2 (b), ε -NSGA-II (c) algorithms (*DTLZ2* benchmark problem)

Table 2: A comparative analysis of our approach with alternative techniques

Benchmark	Algorithm	Generational distance (GD)		Spread and distribution measure (Δ)	
		Average	Std. deviation	Average	Std. deviation
<i>SCH</i>	Our SPEA3	1.3451E-3	6.1730E-5	3.9995E-2	1.2251E-2
	SPEA2	1.3380E-3	9.6155E-5	1.2570E-1	1.4079E-2
	NSGA-II	9.5733E-4	5.3899E-5	6.1462E-1	4.9235E-2
	ε -NSGA-II	1.5788E-3	1.2666E-4	5.0620E-1	2.3372E-2
<i>FON</i>	Our SPEA3	1.6315E-4	8.0883E-6	4.7014E-2	2.5965E-2
	SPEA2	5.3844E-4	6.3862E-5	1.3280E-1	1.9523E-2
	NSGA-II	2.8681E-4	2.6579E-5	5.4080E-1	8.6407E-2
	ε -NSGA-II	2.5421E-4	2.9344E-5	2.0861E-1	2.0671E-2
<i>KUR</i>	Our SPEA3	1.6140E-3	2.4740E-4	3.2292E-1	1.1618E-2
	SPEA2	2.7918E-3	7.7937E-4	2.7377E-1	3.5689E-2
	NSGA-II	1.5983E-3	1.5252E-4	6.4593E-1	5.2421E-2
	ε -NSGA-II	2.1452E-3	2.6530E-4	2.6397E-1	2.3781E-2
<i>ConstrX</i>	Our SPEA3	8.8700E-4	7.4204E-5	1.9946E-1	6.7897E-2
	SPEA2	1.3014E-3	3.2880E-4	3.8227E-1	3.8517E-2
	NSGA-II	6.9490E-4	5.1226E-5	7.3202E-1	4.7862E-2
	ε -NSGA-II	1.2806E-3	1.4710E-4	4.2724E-1	2.3001E-2
<i>ZDT1</i>	Our SPEA3	9.7817E-5	1.6439E-5	3.7962E-2	1.4228E-2
	SPEA2	2.8046E-4	9.2255E-5	1.3921E-1	1.8059E-2
	NSGA-II	1.5796E-4	3.1465E-5	5.6746E-1	3.9048E-2
	ε -NSGA-II	9.3581E-5	2.1614E-5	2.3392E-1	2.0649E-2
<i>ZDT2</i>	Our SPEA3	6.4035E-5	5.2462E-6	5.4382E-2	2.5916E-2
	SPEA2	1.8068E-4	7.7749E-5	1.3284E-1	2.0264E-2
	NSGA-II	9.6239E-5	2.2283E-5	5.9903E-1	5.0398E-2
	ε -NSGA-II	1.4531E-4	4.9443E-5	2.3692E-1	1.5422E-2
<i>ZDT3</i>	Our SPEA3	2.8630E-4	1.7898E-5	2.1027E-1	8.3822E-2
	SPEA2	3.3490E-4	7.9273E-5	1.6986E-1	2.9307E-2
	NSGA-II	1.8209E-4	1.2206E-5	5.8034E-1	5.7662E-2
	ε -NSGA-II	3.7932E-4	5.6220E-5	3.7572E-1	3.8395E-2
<i>ZDT4</i>	Our SPEA3	1.0192E-4	1.2383E-5	3.3304E-2	1.8313E-2
	SPEA2	1.0220E-4	1.6815E-5	1.3927E-1	2.4597E-2
	NSGA-II	5.1427E-3	3.3500E-3	8.0741E-1	1.5762E-1
	ε -NSGA-II	9.9188E-5	2.8830E-5	2.3583E-1	1.9457E-2
<i>ZDT6</i>	Our SPEA3	4.8811E-5	2.0380E-6	5.9647E-2	1.5067E-2
	SPEA2	4.9800E-5	3.2696E-6	1.1795E-1	2.0862E-2
	NSGA-II	3.6686E-3	2.8063E-3	8.0267E-1	1.1830E-1
	ε -NSGA-II	5.1863E-5	5.4961E-6	1.3321E-1	1.9420E-2
<i>DTLZ1</i>	Our SPEA3	3.47E-4	1.361E-4	2.446E-2	7.774E-2
	SPEA2	1.81E-1	3.605E-1	4.302E-1	6.79E-1
	NSGA-II	2.515E-2	2.017E-2	7.455E-1	1.153E-1
	ε -NSGA-II	1.804E-4	6.053E-5	2.722E-1	1.675E-1
<i>DTLZ2</i>	Our SPEA3	4.679E-3	2.031E-4	1.117E-1	6.527E-2
	SPEA2	1.521E-2	3.205E-3	1.33E-1	1.281E-2
	NSGA-II	5.144E-3	1.767E-3	5.729E-1	5.746E-2
	ε -NSGA-II	3.031E-4	1.451E-5	2.48E-1	6.733E-3
<i>DTLZ3</i>	Our SPEA3	1.141E-1	2.0380E-2	2.614E-2	1.506E-3
	SPEA2	1.715E+0	3.262E+0	5.952E-1	5.949E-1
	NSGA-II	3.012E-1	2.5E-1	1.2E+0	3.909E-1
	ε -NSGA-II	3.222E-3	2.322E-3	4.563E-1	1.481E-1