

# EXPERIMENTAL ANALYSIS OF $\epsilon$ -MULTIOBJECTIVE EVOLUTIONARY ALGORITHM

*In-Hee Lee, Soo-Yong Shin, and Byoung-Tak Zhang*

Biointelligence Laboratory  
School of Computer Science and Engineering  
Seoul National University, Seoul, Korea

## ABSTRACT

A multi-objective optimization problem (MOP) is often found in real-world optimization problem. Among various multi-objective optimization techniques, multi-objective evolutionary algorithm (MOEA) is highlighted as a good candidate due to its flexibility, feasibility, and its ability to handle multiple solutions. Among various MOEAs, we analyze  $\epsilon$ -MOEA which can achieve good convergence and diversity using  $\epsilon$ -dominance. We compare  $\epsilon$ -MOEA with controlled elitist non-dominated sorting genetic algorithm (NSGA-II) which is currently one of the most popular and widely used MOEAs and have showed reasonable performance in various applications. We compare two algorithms experimentally for popular and scalable test function DTLZ2 and its modified version. The experimental results show that  $\epsilon$ -MOEA is better than NSGA-II in convergence and is comparable to NSGA-II in diversity.

## 1. INTRODUCTION

A multi-objective optimization problem (MOP) is an optimization problem which has a number of heterogeneous and conflicting objective functions [4]. Many real-world problems including DNA sequence optimization [11] belong to multi-objective optimization problem. Recently, the multi-objective evolutionary algorithm (MOEA) has been highlighted as a good candidate for MOP due to its characteristics such as flexibility, feasibility, and handling diverse objectives [4, 17].

Most classical methods which deal the multi-objective optimization problem transformed multiple objectives to a single objective, for example, weighted sum approach. However, MOEA can deal multiple objectives directly. In addition, MOEA can handle the diverse population and produce multiple trade-off solutions by a single run. In con-

trast, the classical methods can produce only one solution at a time. Since the first real implementation of an MOEA (vector evaluated GA or VEGA) suggested by Schaffer in 1984, various MOEAs have been suggested. But most of these algorithms have two common design criteria: to find solution set converged to optimal as close as possible and to find diverse solution set [17]. In MOP, optimal solutions refer to Pareto-optimal solutions which is a solution one of whose objective value can be improved only by sacrificing one or more other objective values.

For the first criteria (convergence), solutions are evaluated based on the domination concept. An individual  $x$  dominates  $y$  if and only if  $x$  is better than or equal to  $y$  in all objectives and is strictly better than  $y$  at least in one objectives. Among various methods, domination ranking [7, 8] which counts the number of solutions a solution dominates, domination count [15] which counts the number of solution dominates a solution, and dominance depth [12] based on the front a solution belongs to are generally used.

To obtain diverse solution set, kernel-based method [7], nearest point method [16], and histogram method [9] are used to determine the solution distribution over feasible space. Kernel-based method and nearest point method estimate the solution distribution by summation of distance function (kernel) values and distance between a solution and its nearest solution respectively. Histogram method divides the solution space into boxes of constant size and utilizes the number of solutions in a box.

But, no single algorithm achieved the two criteria satisfactorily, because the design and evaluation of an MOEA is itself a kind of MOP. An MOEA which is good at finding diverse solutions tends to show poor convergence, and vice versa.  $\epsilon$ -multiobjective evolutionary algorithm ( $\epsilon$ -MOEA) [10] is proposed to combine both criteria with the use of  $\epsilon$ -domination concept and archive. By the introduction of  $\epsilon$ -dominance,  $\epsilon$ -MOEA can find diverse set of solutions effectively. And by keeping a separate archive, it can maintain optimal solutions found so far. We choose to analyze  $\epsilon$ -MOEA for these reasons.

In this paper, we compared  $\epsilon$ -MOEA with controlled elitist non-dominated sorting genetic algorithm (controlled eli-

---

This research was supported by the Molecular Evolutionary Computing (MEC) project of the Korean Ministry of Commerce, Industry and Energy, and the National Research Laboratory (NRL) Program from the Korean Ministry of Science and Technology, and the BK21-IT Program of the Korean Ministry of Education & Human Resources Development. The ICT at Seoul National University provided research facilities for this study.

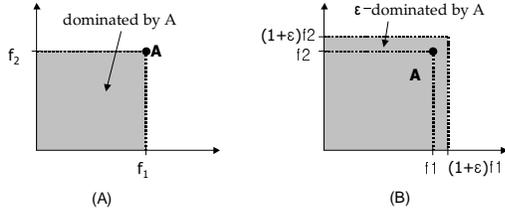


Figure 1: The concept of dominance and  $\epsilon$ -dominance. (a) The usual dominance concept. The shaded area denotes the area dominated by a solution  $A$ . (b) The  $\epsilon$ -dominance concept. The shaded area denotes the area  $\epsilon$ -dominated by  $A$ .

tist NSGA-II) [5]. This is a modified version of the previous algorithm NSGA-II [3]. NSGA-II is one of the most popular MOEAs, and has shown reasonable performance in various applications. And the comparison of controlled elitist NSGA-II and  $\epsilon$ -MOEA might give insight for better MOEA. In this paper,  $\epsilon$ -MOEA is compared to the controlled elitist NSGA-II on popular test function DTLZ2 [6] and its modified version.

Section 2 gives more detailed description of both algorithms: controlled elitist NSGA-II and  $\epsilon$ -MOEA. Section 3 shows the experimental results and conclusions are drawn in Section 4.

## 2. $\epsilon$ -MOEA

$\epsilon$ -MOEA is a steady-state genetic algorithm which maintains an archive of non-dominated individuals found so far. But unlike other MOEAs, it does not use dominance relation when updating the archive. Instead, it uses the  $\epsilon$ -dominance relation where its name comes from. In  $\epsilon$ -dominance relation,  $x$   $\epsilon$ -dominates  $y$  if the difference between  $x$  and  $y$  is greater than or equal to a certain amount  $\epsilon$  in all objectives and is strictly better than  $y$  by  $\epsilon$  in at least one objective (see Fig. 1).

The  $\epsilon$ -dominance is introduced to maintain a representative subset of non-dominated individuals. The  $\epsilon$ -non-dominated set is smaller than the usual non-dominated set, for the non-dominated solutions which can be  $\epsilon$ -dominated by others are removed in  $\epsilon$ -non-dominated set. And the minimum distance between nearest solutions is guaranteed. Thus, by maintaining the  $\epsilon$ -non-dominated set of the population as an archive, we may get a well-spread representative subset of Pareto-optimal set called  $\epsilon$ -approximate Pareto-optimal set [10] (see Fig. 2). We can adjust the density of the approximate set by controlling the value of  $\epsilon$ .

At generation  $t$ , parents are chosen from the population and the archive respectively. When selecting a parent from the population, the winner of tournament of size 2 is selected. The parent from the archive is selected ran-

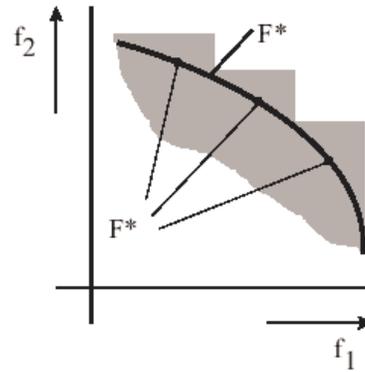


Figure 2: The  $\epsilon$ -approximate Pareto-optimal set. The points on the line are the Pareto-optimal non-dominated solutions. And the points with arrow  $\epsilon$ -dominate other points on the line. They makes the  $\epsilon$ -approximate Pareto-optimal set.

domly among the archive members. Then, an offspring is produced from these parents and evaluated. The offspring replaces an individual of the population dominated by it in usual sense. If no individual in the population is dominated by the offspring, it is simply discarded. If the offspring  $\epsilon$ -dominate one or more members of the archive, it replaces the  $\epsilon$ -dominated members. Or, it is added to the archive if no archive member  $\epsilon$ -dominates it and it  $\epsilon$ -dominates no archive member. Otherwise, the offspring is discarded. This process is repeated until termination (see Fig. 3).

By this way, the archive grows as the generation passes and is filled with  $\epsilon$ -approximate Pareto-optimal solutions. And the archive members at termination are declared to be solutions for the given problem.

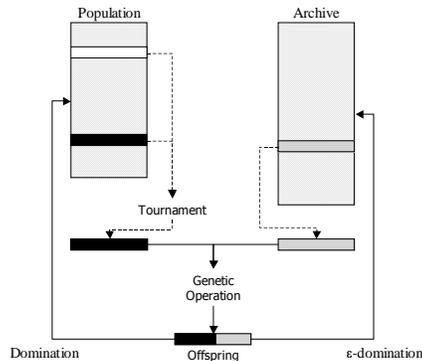


Figure 3: The schematic diagram of  $\epsilon$ -MOEA.

## 2.1. Controlled Elitist NSGA-II

The controlled elitist NSGA-II, introduced in [5] is based on a genetic algorithm and uses dominance relation when comparing two individuals. It was suggested as a modification of NSGA-II [3] to get better result by balancing exploration and exploitation. The main features of this algorithm are the controllable elitism and explicit diversity-preserving mechanism.

At generation  $t$ , after evaluation and recombination, the united population  $R_t$  of the parent population  $P_t$  and the offspring population  $Q_t$  is divided into several groups (fronts) by dominance relation. The individuals which are not dominated by any other individuals in  $R_t$  form the first front and those that are dominated only by those in the first front form the second front and so on. This process is called non-dominated sorting. After this, the new parent population  $P_{t+1}$  is filled by sorted individuals. The filling starts from the first (best) front: the elitism. But the numbers of individuals that are allowed to fill  $P_{t+1}$  from each front are restricted by a predefined distribution (see Fig. 4). In [5], a geometric distribution is used.

Under this scheme, various fronts always co-exist in  $P_{t+1}$ . The distribution of the size of each front is controlled by a reduction rate  $r$ .  $r$  is the ratio of the size of the next front over that of the previous front. The elitism in NSGA-II is controlled by adjusting  $r$ .

## 3. EXPERIMENTAL RESULTS

The algorithms introduced in the previous section is compared experimentally. The comparison result was analyzed in terms of diversity and convergence. As measures for diversity and convergence, we used maximum spread [14] and generational distance [13], respectively. The generational distance is a measure for average distance of non-dominated set found by an algorithm and Pareto-optimal set. The smaller generational distance value means the more converged set. And the maximum spread is the length of the diagonal of a hyperbox which surrounds the final non-dominated set. Thus, the larger maximum spread value de-

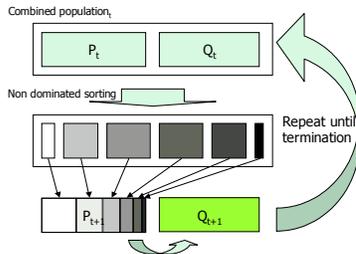


Figure 4: The schematic diagram of NSGA-II.

notes the more diverse set. Therefore, the algorithm which has small generational distance value and large maximum spread value will be better.

We used two versions of DTLZ2 [6] function as our test problem. We chose DTLZ2 because it is used in many experimental comparison studies of MOEA and is easy to increase or decrease the number of objectives. Its objective space lies outside of the first quadrant of a unit hyperball centered at 0. The objective functions are to be minimized, so its Pareto-optimal front is the surface of the hyperball and the objective space shrinks towards the surface of the hyperball from outside. Thus, the convergence characteristic of an algorithm is tested here and the algorithm with small generational distance and maximum spread values is good for this problem.

For the test of diversity characteristic of algorithms, we modified DTLZ2 to have search space lie between the first quadrant of two hyperballs centered at 0: one of radius 1 and the other of radius 2. The objective functions are to be maximized this time, therefore, Pareto-optimal front is the surface of the outer hyperball and the search space expands towards the surface of outer hyperball from that of the inner one. So the diversity characteristic is tested here and the algorithm with small generational distance value and large maximum spread value is good for this problem.

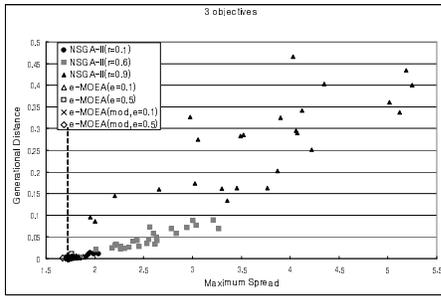
We used the simulated binary crossover [1] and the polynomial mutation [2] as genetic operators for both algorithm due to the domain of the benchmark function is real-valued. We tested both algorithm with various parameters of crossover and mutation operator. For  $\epsilon$ -MOEA, we used  $\epsilon \in \{0.1, 0.5, 0.9\}$  for all objectives. For controlled elitist NSGA-II, we used  $r \in \{0.1, 0.6, 0.9\}$ . We used 0.6, since it is the recommended value in [5]. For both algorithm, we used the usual value of crossover probability 0.9 and mutation probability 0.01. We used population size 100 for both algorithm. For the same number of function evaluation, the generation number was 1000 for controlled elitist NSGA-II and 100000 for  $\epsilon$ -MOEA.

### 3.1. DTLZ2 Test Function

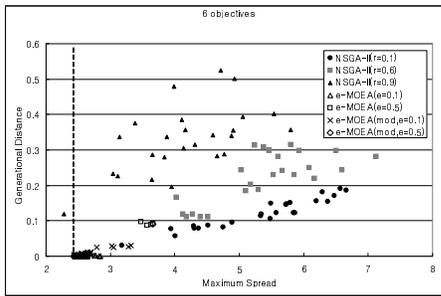
We compared two versions of  $\epsilon$ -MOEA to NSGA-II. One has an unlimited archive size, and the other has a limited archive size. For it might be unfair to compare  $\epsilon$ -MOEA with infinite archive and NSGA-II with finite population size. The archive of  $\epsilon$ -MOEA grows over generation but the first front of NSGA-II can not exceed the population size. For fair comparison, we modified  $\epsilon$ -MOEA to keep only a limited number of archive members. After the archive reaches the maximum size, newly found individual replaces the nearest one in the archive.

The generation distance and maximum spread of the first front (or archive) at final generation of the algorithms of different parameter settings is shown in Fig. 5. When

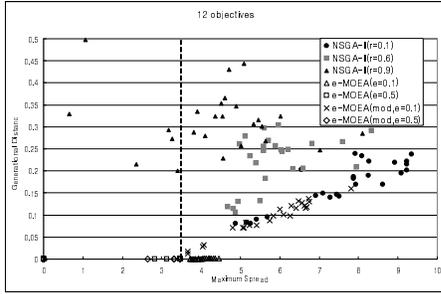
$\epsilon$  was set to 0.9,  $\epsilon$ -MOEA could keep only one member at the archive. Thus, we exclude that data from here and the next subsection. The dashed line in each graph denotes the maximum spread value of Pareto-optimal front.



(a) For 3-objectives.



(b) For 6-objectives.



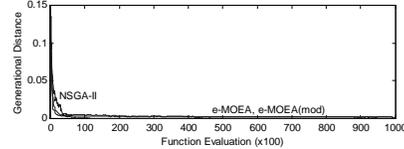
(b) For 12-objectives.

Figure 5: The results for DTLZ2 test function.

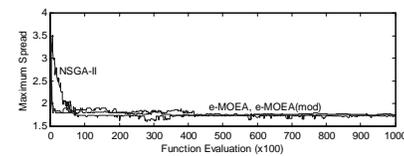
In Fig. 5, the controlled elitist NSGA-II shows large spread and  $\epsilon$ -MOEAs show better convergence. In controlled elitist NSGA-II, the diversity-preserving mechanism works as the front size approaches its limit due to convergence. Because the crowding distance metric used in NSGA-II favors the boundary solutions (the solutions which have best value for only very few objectives), they are hardly removed from the population until a solution dominates it is found and prevent other intermediate solutions in crowded region from survive. Therefore, the first front hardly shrinks. But, in DTLZ2 test function, the search space shrinks towards the surface of the unit hyperball. This is why the NSGA-II shows poor convergence in this problem. The

convergence of NSGA-II is improved as the  $r$  decreased. This is because, with small  $r$ , the first front can take large portion of population and the chance of solutions in more crowded region may survive is increased.

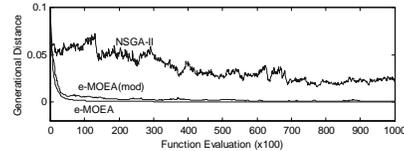
For both  $\epsilon$ -MOEAs, smaller  $\epsilon$  value gives better results. And as the number of objectives grows, the  $\epsilon$ -MOEA with finite archive size shows decreased performance than the case of infinite archive size. But, it still shows better performance than NSGA-II. That is, although the size of archive and population is similar,  $\epsilon$ -MOEA shows better performance.



(a) For 3-objectives.



(b) For 6-objectives.



(c) For 12-objectives.

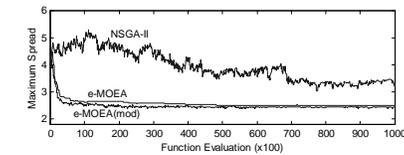


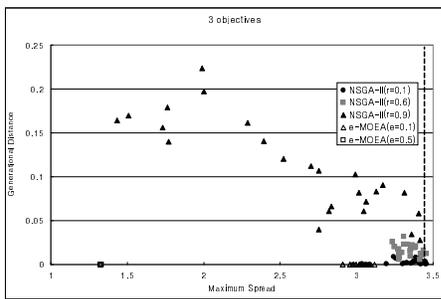
Figure 6: Example runs for DTLZ2 test function.

The sample runs of each algorithm are shown in Fig. 6. When the number of objectives is small, all algorithms show the similar results. But as the number of objectives grows, NSGA-II starts to fail to converge toward optimal front. In

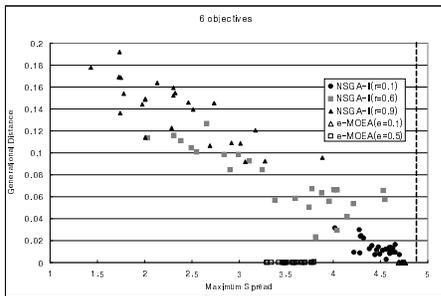
contrast, both  $\epsilon$ -MOEAs shows acceptable convergence and diversity even with increased number of objectives.

### 3.2. Modified DTLZ2 Test Function

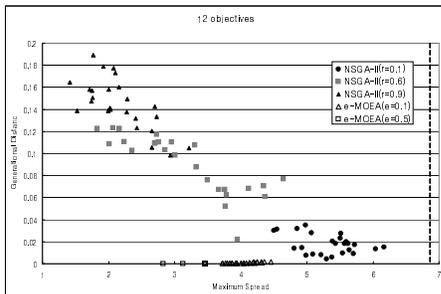
In contrast to the previous problem, the search space in this problem expands as it is close to the Pareto-optimal front. Therefore, the ability to spread across search space as well as to converge to optimal front is required. We modified DTLZ2 to compare this ability of each algorithm. The modified  $\epsilon$ -MOEA was not used here, because it has similar trend to  $\epsilon$ -MOEA. As in the previous test problem, generational distance and maximum spread values at the final generation are shown in Fig. 7. The dashed line denotes the optimal maximum spread value as before.



(a) For 3-objectives.



(b) For 6-objectives.

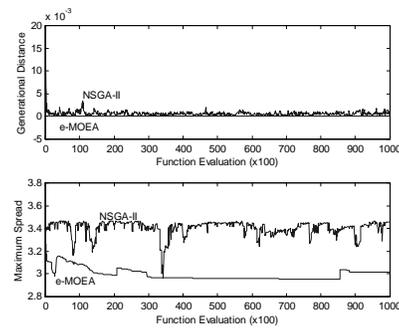


(c) For 12-objectives.

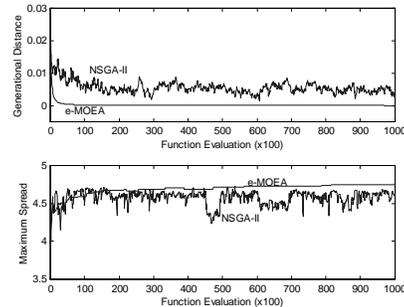
Figure 7: The results for modified DTLZ2 test function.

As can be seen in Fig. 7,  $\epsilon$ -MOEA shows comparable diversity to NSGA-II, except for the case of 3 objec-

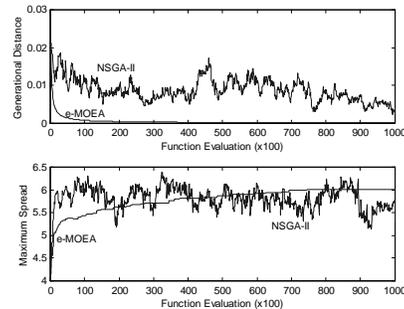
tives. And as the parameters  $\epsilon$  and  $r$  change, both algorithms shows similar trend as in the previous problem. The failure of NSGA-II in spite of its diversity-preserving mechanism is due to its slow convergence towards optimal front. As shown in Fig. 6 and Fig. 8, NSGA-II always shows slower convergence than  $\epsilon$ -MOEA. In this problem, the Pareto-optimal front has the widest area. Therefore, the algorithm which is not converged to the optimal front must have small maximum spread value. There is the chance that a algorithm shows poor diversity in spite of its converge to optimal front. But, Fig. 8 shows that the poor diversity of NSGA-II comes from its poor convergence. We guess the diversity-preserving mechanism slowed its convergence, but we need more analysis about this.



(a) For 3-objectives.



(b) For 6-objectives.



(c) For 12-objectives.

Figure 8: Example runs for DTLZ2 test function.

#### 4. CONCLUSIONS

We compared the  $\epsilon$ -MOEA with controlled elitist NSGA-II for various test functions under various parameter settings. The experimental results shows that the  $\epsilon$ -MOEA is comparable with NSGA-II in diversity. But,  $\epsilon$ -MOEA guarantees a minimum distance between solutions in archive and uniform distribution over the search space since the search space is divided into grid of width  $\epsilon$  and only one solution per grid is kept. There is no such guarantee in NSGA-II. And  $\epsilon$ -MOEA showed much better convergence behavior. In addition, although not shown here,  $\epsilon$ -MOEA needs smaller computational time. Thus, it can be concluded that  $\epsilon$ -MOEA is better from various point of views.

However, we can not always maintain infinite archive for  $\epsilon$ -MOEA for physical limitation. And when the archive size is restricted, a decrease in convergence was found. It would be worth to look for the finite archive updating policy without sacrificing convergence much.

#### 5. REFERENCES

- [1] K. Deb and R. B. Agrawal, "Simulated Binary Crossover for Continuous Search Space," *Complex Systems* vol. 9, no.2, pp. 115–148, 1995.
- [2] K. Deb and M. Goyal, "A Combined Genetic Adaptive Search (GeneAS) for Engineering Design," *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.
- [3] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II," *Proceedings of the Parallel Problem Solving from Nature VI*, pp. 849–858, 2000.
- [4] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Ltd., England, 2001.
- [5] K. Deb and T. Goel, "Controlled Elitist Non-dominated Sorting Genetic Algorithm for Better Convergence," *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pp. 42–50, 2001.
- [6] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Test Problems for Evolutionary Multi-Objective Optimization," KanGAL Report No. 200101, 2001.
- [7] C. M. Fonseca and P. J. Fleming, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization," *Proceedings of the Fifth International Conference on Genetic Algorithm*, pp. 416–423, 1993.
- [8] J. Horn and N. Nafpliotis, "Multiobjective optimization using the niched Pareto genetic algorithm," IlliGAL Technical Report 93005, 1993.
- [9] J. D. Knowles and D. W. Corne, "Approximating the Nondominated Front using the Pareto Archived Evolution Strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.
- [10] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining Convergence and Diversity in Evolutionary Multi-objective Optimizatin," *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, 2002.
- [11] I.-H. Lee, S.-Y. Shin, and B.-T. Zhang, "DNA Sequence Optimization Using Constrained Multi-Objective Evolutionary Algorithm," *Proceedings of the 2003 Congress on Evolutionary Computation*, vol. 4, pp. 2270–2276, 2003.
- [12] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithm," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.
- [13] D. V. Veldhuizen, "Multiobjective Evolutionary Algorithms: Classifications, Analysis, and New Innovations," Ph. D. Thesis, Dayton, OH:Air Force Institute of Technology, Technical Report No. AFIT/DS/ENG/99-01, 1999.
- [14] E. Zitzler, "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications," Ph. D. Thesis, Zürich, Switzerland: Swiss Federal Institute of Technology (ETH) (Dissertation ETH No. 13398), 1998.
- [15] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transaction on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [16] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," Technical Report 103, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, 2001.
- [17] E. Zitzler, M. Laumanns, and S. Bleuler, "A Tutorial on Evolutionary Multiobjective Optimization," *Metaheuristics for Multiobjective Optimization, Lecture Notes in Economics and Mathematical Systems*, Springer, vol. 535, pp. 3–37, 2004.