An New Evolutionary Multi-objective Optimization algorithm

Mu Sheng-jing, Su Hong-ye, Chu Jian
National Laboratory of Industrial Control Technology, Institute of Advanced Process Control, Zhejiang University, Yuquan Campus, Hangzhou, 310027,China
Corresponding Author email: **sjmu@iipc.zju.edu.cn**

Wang Yue-xuan
National CIMS Engineering Research Center
Institute of System Integration, Department of Automation, Tsinghua University, Beijing 100084,China

Abstract- This paper introduces a new, simple and efficient evolutionary algorithm to multi-objective optimization problem, which based on neighborhood and archived operation (NAGA). The innovations contain two main parts: neighborhood identify procedure to obtain Pareto optimal solutions from the population and neighborhood crowding procedure to maintain the diversity of Pareto optimal solutions previously found. The neighborhood identify procedure is composed of two steps, first to identify the locally non-dominated solutions from the population and then to obtain the global non-dominated solutions among the locally solutions. The neighborhood crowding is introduced to maintain a widely distributed set of Pareto solutions along the Pareto optimal front, which through implementing a comparison among the neighborhood bounds of new identified Pareto solutions and those of solutions in the archive. The winners, which are not in any ranges of the solutions in the archive, will be copied to the archive. A well-tuned fitness assignment method is structured to guide the population converging to the true Pareto optimal front. This method is pragmatic compromise between the computational simplicity and efficiency. Four nicely balanced test problems are provided to check the performance of the approach.

1 Introduction

Most real world engineering optimization problems normally have several (possible conflicting) objectives that must be satisfied at the same time. Typically, classical methods to MO problem are to scalarize multiple objectives into one objective by averaging the objectives with a weight vector or to combine multiple objectives to constraints with associated thresholds and penalty functions. But penalties and weights have proven to be problematic. The final solution is usually very sensitive to small changes in the penalty function coefficients and weighting factors.

Since 1985, a considerable amount of Evolutionary Multi-Objective Optimization (EMOO) approaches have been developed that are capable of searching for multiple solutions concurrently in a single run [Schaffer, 1985; Fonseca, et al., 1994, Srinivas, et al., 1994; Veldhuizen, 1996; Deb, et al., 2000; Zitzler, et al, 2000; Coello, 2001]. Fonseca, et al. (1994) concluded the methods reported in the literature into plain aggregating approaches, population-based non-Pareto approaches, Pareto-based approaches and Niche induction techniques, which the latter two are more popular. Coello (2001) discussed the most popular EMOO techniques currently in use, analyzed their advantages and disadvantages. Especially, he pointed out the two new EMOO approaches, Pareto Archived Evolution Strategy (PAES) [Knowles, et al., 2000] and Strength Pareto Evolutionary Algorithm (SPEA) [Zitzler & Thiele, 1999] are very promising. The common feature of the two approaches is to use a history archive that records all the non-dominated solutions previously found. But is seems that they are still use time-consuming methods to keep the diversity of the population. Furthermore, there are no algorithms addressed on how to identify the Pareto optimal solution from the population in detailed, which is an important task that can't be ignored.

In the present study, an efficient and simple multi-objective optimization genetic algorithm is proposed, which based on neighborhood and archived operation (named as NAGA for short). The neighborhood operation concludes two parts, identifying the Pareto optimal solution from the population by neighborhood comparison and spreading its population out along the Pareto optimal trade-off surface (Pareto optimal front) by neighborhood crowding. The algorithm applies neighborhood operations, compared with other multi-objective optimization evolutionary algorithms, such as niche based approaches or Pareto based approaches, requires low time computational complexity and storage. Four typical test problems used in previously literatures are selected to illustrate the performance of NAGA.

2 The Neighborhood and Archived Genetic Algorithm

The main works of general multi-objective optimization genetic algorithms contain two parts, identify the Pareto optimal solution from the population and convergence to the Pareto optimal front, and then distributed the solution on the Pareto optimal frontier uniformly. Our approach also focuses on the two tasks.

2.1 Identify Pareto optimal solution

The definition of Pareto optimal solution is available elsewhere [Srinivas, et al., 1994; Fonseca, et al., 1994; Zitzler & Thiele, 1999; Coello, 2001], the Pareto optimal identify procedure is based on these definitions.

It is assumed that x_i and x_j are Pareto optimal solution vectors whilst x_k is a non-Pareto solution vector of a minimization problem. The relation between x_i and x_j is that the criterion of x_i is partial worse than those of x_j whilst partial better than those of x_j. Namely, x_i and x_j are 'non-inferior' or non-domin -ated with respect to each other. The relation among Pareto optimal solutions x_i or x_j and x_k, however, is that there exists all criterions of at least one among x_i and x_j are better than those of x_k, which also called that x_k is 'dominated' . This concept always gives a set of solutions rather than a single solution, which titled as the Pareto optimal set.

According to the two level definitions of Pareto optimal solution, we design a two stages procedure based on neighborhood comparison to identify the Pareto optimal set from the population. The first step is the locally non-dominated identify procedure. It is based on the premise that the solution from the neighborhood of a Pareto solution is also a Pareto solution, while the solution from the neighborhood of a non-Pareto solution is also a non-Pareto one. Thus, compared with the original Pareto solution, the substitute solution from its neighborhood will better part objective function values and worsens others simultaneously. However, it is not the case for that of non-Pareto solution, i.e., its neighborhood solution will either improve all objective function values or degrade all objective function. Thereby, by comparing the monotonicities of all objective functions of a candidate solution and its neighborhood solution, a necessary condition to identify the Pareto optimal solution is obtained. That is, if all objective functions of a candidate solution are either monotone increasing or monotone decreasing in its neighborhood, the solution can be regard as an inferior solution and be discarded. Otherwise, if partial objective functions of a candidate solution are monotone increasing and others are monotone decreasing, it is identified as locally non-dominated solution and go to the second identify procedure.

The second identify procedure handles the locally non-dominated solution generated from the first identify step. It compares all objective values of the locally non-dominated solutions with those of all Pareto optimal solutions in the history archive. A locally non-dominated solution will be discarded if all of its objective values are worse than those of a certain Pareto optimal solution in the history archive. Finally, the rest ones after the comparison step are new Pareto optimal solutions and then performed neighborhood crowding proposed in the next subsection. As a matter of fact, this step is to determine whether the locally non-dominated solution is non-dominated within the entire search space, which is called as Pareto optimal solution or not. When the archive is empty in the initial population, the whole

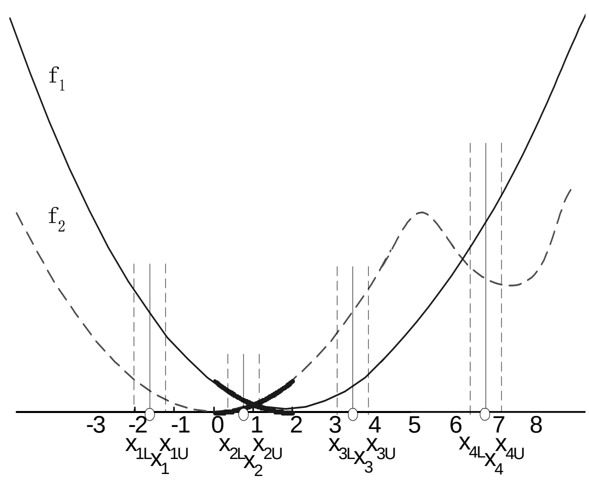

Figure 1. The diagrammatic presentation of the locally non-dominated solution identified procedure and Pareto optimal solution identify procedure.

initial population is instead.

A simple two-objective minimum problem of one variable is considered to illustrate the concept of the two step of Pareto optimal solution identified procedure The problem has two objectives f_1 and f_2, which is modified here for discriminate the locally non-dominated solution and the global non-dominated solution (Pareto optimal solution), which are shown in Figure 1. It is clear that the Pareto optimal solutions constitute all x values varying from 0 to 2 [Srinivas, et al., 1994]. x_1, x_2, x_3 and x_4 are four candidate solutions that should be identified. The dotted lines at the two sides of these solutions denote their neighborhoods. Considering x_1, which is between -2 and -1, the relations of the objectives of its left side neighborhood are $f_1(x_{1L}) > f_1(x_1)$ and $f_2(x_{1L}) > f_2(x_1)$, whilst those of its right side are $f_1(x_{1U}) < f_1(x_1)$ and $f_2(x_{1U}) > f_2(x_1)$. It is illustrated that x_{1U} is better than x_1, i.e., x_1 is dominated by x_{1U}. However, it is different case for solution x_2, in that $f_1(x_{2L}) > f_1(x_2)$ and $f_2(x_{2L}) < f_2(x_2)$ for the left side and $f_1(x_{2U}) < f_1(x_2)$ and $f_2(x_{2U}) > f_2(x_2)$ for the right side. Both x_2, x_{2L} and x_{2U} are non-dominated to each others, as a result of, x_2 is a non-dominated or locally non-dominated solution. When checking solution x_3, the same conclusion is drawn to that of solution x_1. Observing solution x_4, it also satisfy the relation $f_1(x_{4L}) < f_1(x_4)$ and $f_2(x_{4L}) > f_2(x_4)$ for the left side and the reverse relational operators for the right side. Thus, x_4 is also a locally non-dominated solution. However, when it is compared with x_2, both the two objective values are greater than those of x_2, which indicate that x_4 is dominated by x_2 and is not a Pareto solution. Consequently, after identified through the two steps, only x_2 from the four candidate solutions is a Pareto solution. In fact, only one side neighborhood objectives comparison is adequate to judge the objective functions in its neighborhood are monotonic or not. In the program code, the left side neighborhood objectives comparison alone is performed, which save the time to evaluation another side objectives vastly.

To simplified treatment, the neighborhood is the perturbation of a certain value of the solution vector. For example, the solution from x_i's neighborhood is the first value of vector x_i multiplies 1.0001.

2.2 Neighborhood crowding to keep diversity of the population

Maintain the diversity among the Pareto optimal solutions is another bottleneck in multi-objective optimization. It is usually the most time consuming part in the whole procedure. Zitzler, et al. (1999) discussed several popular and promising techniques and concluded them into niching techniques and non-niching techniques. Among the niching class, fitness sharing method, which first proposed by Goldberg, is the most popular. But it is complex and problem dependent for it need to calculate and compare the distance between each solution and others in the population and an important parameter, which so-called niche radius σ_{share} should be estimated in advance. The two shortages make fitness sharing method be hard to practical application, which is confirmed by many researchers [Deb et al., 2001; Coello, 2001] and our experience. In the previously proposed algorithms, NSGA and SPEA use niching method and its derivative, which the time complexity is $O(N^3)$ [Deb et al., 2001; Knowles, et al. 2000]. Among the non-niching techniques, restricted mating and crowding are the most two common methods used in multi-objective optimization. Unfortunately, similarity to the fitness sharing techniques, restricted mating method should be appropriately given a key parameter σ_{mate}, which denotes the threshold distance to allow individuals to mate. Furthermore, the calculation of distance among the population is also highly time consuming.

In this paper, a simple and efficient technique, neighborhood crowding is used to maintain the diversity of the population. Additionally, a history archive is introduced to store the Pareto optimal solution previously found. The crowding approach is implement by compared the new Pareto solution identified in the current population with those solutions in the archive. Each Pareto optimal solution in the archive is represented by a small interval $[x-\varepsilon, x+\varepsilon]$ that including the upper bound and lower bound of the solution. The two bounds are regard as the small interval of two sides of the solution, which can also be taken as its neighborhood. The definition of ε:

$$\varepsilon = d * (x^U - x^L) \quad (1)$$

where d is a adjustable coefficient of NAGA, and x^U and x^L are the upper bound and lower bound of the decision vector, respectively.

The neighborhood crowding method is performed as: If a new solution x_j is in the range of the interval or the neighborhood of a certain solution x_i of the archive, i.e., $x_j \in [x_i - \varepsilon_i, x_i + \varepsilon_i]$ for one-dimension problem, it will be discarded; Else if it is not in any ranges of the neighborhoods of all solutions of the archive, it will be copied to the archive and set its upper bound and lower bound as a fresh Pareto optimal solution.

The diagrammatic presentation of the neighborhood crowding with archived operation is shown in Figure 2. x_{i-1}, x_i and x_{i+1} are three successive solutions in the archive, and the size of the solution's neighborhood is set to ε, i.e., the neighborhood of solution x_i is $[x_i - \varepsilon_i, x_i + \varepsilon_i]$ as remarked by dotted line in Figure 2a. x_{j-1}, x_j and x_{j+1} are three candidate Pareto solutions identified by the neighborhood identify procedure. For the one-dimension case, as shown in Figure 2a, both x_{j-1} and x_{j+1} are not in any ranges of the solutions in the archive, whilst x_j is in the range of x_i 's neighborhood, i.e., $x_i - \varepsilon \le x_j \le x_i + \varepsilon$. It is assurable that x_j is a stale Pareto solution that should

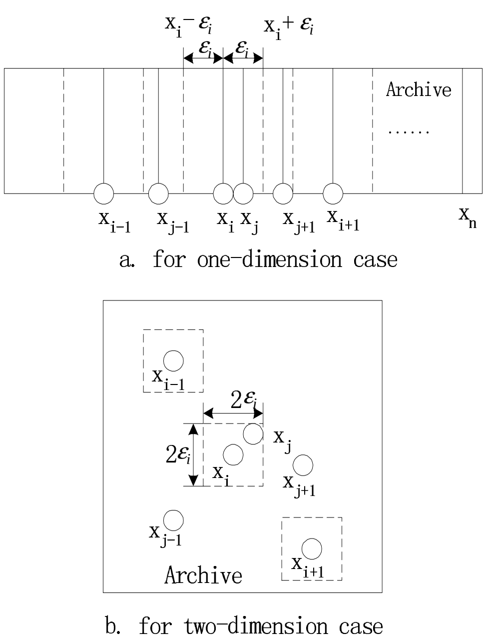

Figure 2 The neighborhood crowding with archived operation of NAGA.

be discarded and x_{j-1} and x_{j+1} are selected as the fresh Pareto solution to copy to the archive. In Figure 2b, the neighborhood crowding approach for two-dimension decision vector is illustrated. The dotted line square denotes the neighborhood of a solution in the archive. In Figure 2b, x_j is in the neighborhood of solution x_i, whilst x_{j-1} and x_{j+1} are stand-alone. Similar to the one-dimension case, the former is fail to be a fresh Pareto solution of the archive while the latter two are winners.

2.3 Fitness assignment

The fitness assignment procedure of our method is performed through assigning large value to the identified fresh Pareto optimal solution and low value to other discarded solution. The normalized Euclidean distance is introduced in the fitness assignment. For the fresh Pareto optimal solution identified in the current population, its fitness value is calculated as follows:

$$fitvalue(x_i) = fit(i) + dist(i) \quad (2)$$

where the normalized Euclidean distance $dist(i)$ is calculated as follows:

$$dist(i) = \frac{1}{n-1} \frac{\sum_{j=1, j\ne i}^{n} \left\| x_i - x_j \right\|}{\left\| x^u - x^l \right\|} \quad (3)$$

where n is the number of population, and x^u, x^l are the upper and lower bound values of the decision vector x, respectively.

The term $fit(i)$ is set be to same value among the fresh Pareto solution and another same small value among others. It is suitable to set the range of the two values is no less than the maximum value among $dist(i)$. In our applications, for the Euclidean distance $dist(i)$ is less than 0.5 in the most cases, $fit(i)$ is fixed to 1 for the fresh Pareto solution and 0.5 for the others. The fitness assignment will guide the evolutionary population converging to the Pareto optimal front.

2.4 The whole flow of NAGA

The whole flow of NAGA is outlined as follows,

1. *Initialization. Set the number of maximum generation,* $K\max$ *and the parameters of the general Genetic algorithms: Population size, Probability of Crossover and Probability of Mutation. Set the volume of the history archive* VA *and the neighborhood size* d. *Generating initial*

population with random distribution.

2. *Identify the Pareto optimal solutions from the population by Neighborhood identify method. Calculate and compare the objective functions of the initial population and their neighborhoods'.*
3. *Perform neighborhood crowding and add the Pareto optimal solution to the history archive. Assign fitness for the initial population.*
4. *If generation is greater than or equal to* $K\max$ *and the number of solution in the archive is greater than or equal to* VA*, stop.*
5. *Apply selection, crossover and mutation operation to generate new offspring. Preserve the best solution in the current population.*
6. *Identify the Pareto optimal solution from the offspring and assign the fitness of GA for the offspring.*
7. *Perform neighborhood crowding to judge whether the identified Pareto optimal solution is a fresh one or not. Add the fresh Pareto optimal solution to the archive.*
8. *Return to step 4*

In step 5, the best solution in the current offspring is the solution that its fitness value be the top. This approach is implement by Matlab 5.3 code. The other parts of the algorithm use general elitist genetic algorithm procedure. The selection operator, crossover operator and mutation operator are followed with those used or proposed by Mu [Mu, et al., 2002], which apply weighted roulette wheel procedure, arithmetical crossover and neighborhood mutation, respectively.

2.5 The analysis of computational complexity and the sensitive of specific parameters

In this subsection, we analyse the computational complexity: time complexity and storage complexity from the view of the procedure.

In order to identify a population of size N according to the neighborhood identify, each solution must be compared the monotonicity of its each objective functions with its neighborhood's in the first step. This requires $O(mN)$ comparisons for all solutions in the population, where m is the number of objective functions. The second identify procedure requires $O(amN)$ computational complexity for objectives comparisons, where a is the number of the locally non-dominated solution generated by the first stage identify procedure and usually significantly less than N. In the worst case, it requires $O(mN^2)$ time complexity to identify the Pareto optimal solutions. But in the average case, as proved repeat by the practical simulations, a is only one quarter or one third of the size of the population. The total storage of the identify procedure requires mN number of objectives and N number of fitness, which the storage complexity is $O(mN)$.

After the second procedure, the final number of Pareto optimal solutions identified in every generation is usually one third or half of a. So in the neighborhood crowding procedure, it requires to handle one tenth of N individuals compared with VA solutions if the archive is full, where VA is usually proportional to N. The time complexity of the second procedure is $O(mN^2)$ in the worst case. At the same time, this procedure requires store mVA number of objectives in the worst case. In summary, the overall time complexity of the algorithm is $O(mN^2)$ and the storage is $O(mN)+O(mVA)$, or $O(mN)$.

The specific parameters of the proposed algorithm are the size of the neighborhood, d in neighborhood crowding procedure and the volume of the history archive, VA. But they are less sensitive than σ_{share} in NSGA, NPGA, the number of subdivision levels, l in PAES intuitively.

The comparisons of the computational complexity and the sensitive of the specific parameters of the Neighborhood and Archived GA (NAGA) and the other popular algorithms are summarized in Table 1, which all assume in the worst case.

3 The test problems

In this section, we apply NAGA on four nicely balanced test problems, which are well known used by many other popular methods in [Zitzler, et al., 2000; Deb, et al.,

Table 1. Computational complexity and sensitive of the specific parameters of 6 algorithms

	NAGA	NSGA*	NSGA-II	PAES	NPGA	SPEA
Time complexity	$O(mN^2)$	$O(mN^3)$	$O(mN^2)$	$O(mN^2)$	$O(mN^2)$	$O(mN^3)$
Storage	$O(mN)$	$O(mN)$	$O(mN^2)$	$O(mN^2)$	$O(mN^2)$	$O(mN)$

* Partial results are borrowed from [Knowles, et al., 2000] and [Deb, et al., 2000]

2002]. For all test problems and with the NAGA, we use a population of size 100, a crossover probability of 0.9 and a mutation probability of 0.05. The maximum generation is 1000, the maximum number of solution in the history archive, or the volume of the archive VA is 500 and the size of the neighborhood of the solution is 1 ± 0.001 multiplies the decision vector. Each of the test problems defined below is structured in both minimize the objective functions, f_1, f_2 and f_3.

3.1 A non-convex problem: ZDT2

Figure 3 shows the distribution of the Pareto optimal solutions along the front. The results show that NAGA performed better in maintaining a widely distributed set of solutions on this problem than that of by SPEA in Figure 2 of ref [Zitzler, et al., 2000]. The true Pareto optimal front is formed with $g(x)=1$.

3.2 A convex problem with multiple noncontiguous parts: ZDT3

Figure 4 shows the distribution of the Pareto optimal solutions along the front. The Pareto optimal front consists of five noncontiguous convex parts, which is caused by the introduction of the sine function. The true Pareto optimal front is formed with $g(x)=1$. The obtained distribution of Pareto optimal solutions in Figure 4 performs well as the results by SPEA and better than NSGA, NPGA and other famous algorithms in Figure 3 of ref [Zitzler, et al., 2000].

3.3 A multi-modality problem with a mass of local Pareto optimal fronts: ZDT4

Figure 5 shows the distribution of the Pareto optimal solutions along the front. The global Pareto optimal front is formed with $g(x)=1$. Compared with the best results by SPEA in Figure 4 of ref [Zitzler, et al., 2000], the Pareto solutions obtained by NAGA behave better both in converging to the true Pareto front and maintaining a widely distributed set of solutions.

3.4 A scalable problem with three objective functions: DTLZ2

This test problem has a spherical Pareto optimal front as in Figure 6 (marked with line). The Pareto optimal solutions obtained by NAGA are covered the front with a density (marked with hollow square). This problem is used to investigate the algorithms' ability to scale up its performance in large number of objectives [Deb, et al., 2002]. The best result by NSGA-II in Figure 6 of ref [Deb, et al., 2002] is not better than the distribution of the marked with square in Figure 6, which denotes the Pareto optimal solutions of this problem by NAGA.

4 Conclusion

In this paper, we proposed a fast and efficient multi-objective optimization genetic algorithm based on neighborhood operations and archived operation (NAGA). It has been proved both on the views of the experiment effects and computational complexity analysis that our approach could find the Pareto optimal solutions and obtain well distribution on the Pareto optimal front fast. Four challenging examples are used to test our approach, which the final simulation results proved that NAGA is not bad or better than other methods in finding better distribution on the Pareto optimal front. It is foreseeable that NAGA should be promising and find increasing attention and applications

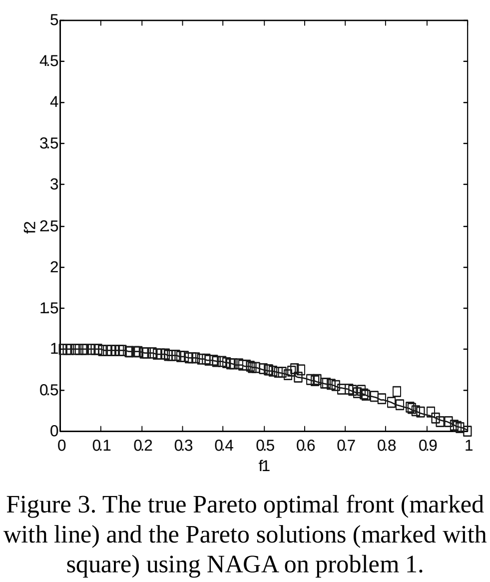

Figure 3. The true Pareto optimal front (marked with line) and the Pareto solutions (marked with square) using NAGA on problem 1.

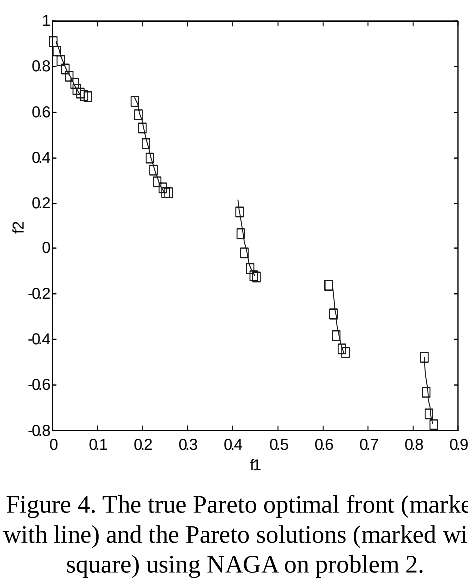

Figure 4. The true Pareto optimal front (marked with line) and the Pareto solutions (marked with square) using NAGA on problem 2.

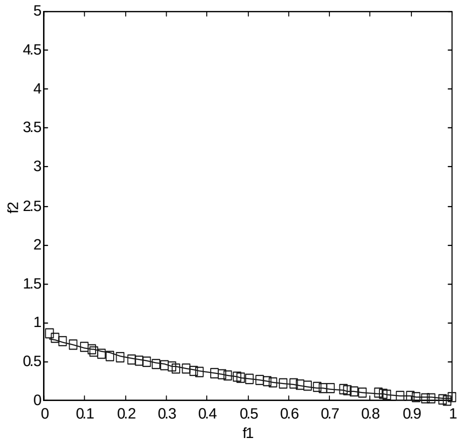

Figure 5. The true Pareto optimal front (marked with line) and the Pareto solutions (marked with square) using NAGA on problem 3.

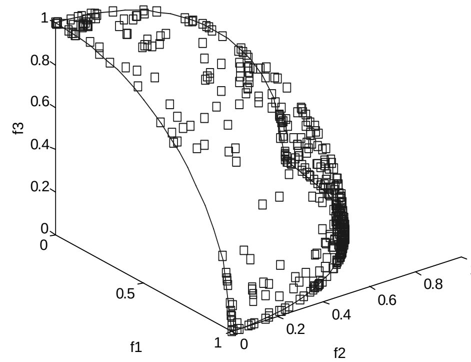

Figure 6. The true Pareto optimal front (marked with line) and the Pareto solutions (marked with square) using NAGA on problem 4.

in future.

Acknowledgements

This work is supported by The National Outstanding Youth Science Foundation of China (NSFC: 60025308) and National Key Technologies R&D Program in the 10th five-year plan （2001BA204B01）.

Bibliography

Schaffer, J.D. (1985). Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In Genetic Algorithms and their Applications. *Proceedings of the First International Conference on Genetic Algorithms*, pages 93-100, Lawrence Erlbaum.

Fonseca, C.M. & Fleming, P.J (1994). An Overview of Evolutionary Algorithms in Multiobjective Optimization. Technical report, Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U. K.

Coelle, C.A.C (2001). A Short Tutorial on Evolutionary Multiobjective Optimization. *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO-2001).*

Knowles, J.D. & Corne, D.W (2000). Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. Evolutionary Computation, 8(2): 149-172

Zitzler, E. and Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Trans. On Evolutionary Computation, 3(4), 257-271.

Srinivas, N. and Deb, K. (1994). Multiobjective optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation, 2(3): 221-248

Veldhuizen, D.A.V. (1999). Multiobjective Evolutionary Algorithms: Classification, Analyses, and New Innovations, PhD Dissertation of the Air Force Institute of Technology Air University, June.

Mu, S., Su, H., et al., (2002). A new Genetic Algorithm to handle the constrained optimization problem, *Proceedings of the 41st IEEE Conference of Decision and Control*, Las Vegas.

Deb, K. et al. (2000). A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, KanGAL report 200001, Indian Institute of Technology, Kanpur, India.

Zitzler, E., Deb, K. & Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, Evolutionary Computation, 8(2):173-195

Deb, K., Thiele, L., Laumanns, M. Zitzler, E. (2002). Scalable Multi-Objective Optimization Test Problems. *Proceedings of the Congress on Evolutionary Computation (CEC'2002)*, Vol. 1, pp. 825-830.