Proceedings of the 7th WSEAS International Conference on Evolutionary Computing, Cavtat, Croatia, June 12-14, 2006 (pp33-38)

Bridging sensitivity information to improve the performance of Pareto Archived Evolution Strategy

APICHART SUPPAPITNARM
The National Metal and Materials Technology Center (MTEC)
National Science and Technology Development Agency (NSTDA)
114 Paholyothin Road, Tambol Klong 1, Klong Luang, Pathumthani
THAILAND
http://www.mtec.or.th

Abstract: - The Pareto Archived Evolution Strategy (PAES), one of the most successful evolutionary optimizers, has long been proven as an easy-to-implement algorithm due to its simple (1+1) search to solve for multiobjective problems. However, a recent comparative study with Multiobjective Simulated Annealing (MOSA), another potential heuristic search technique, showed that when the problems are constrained or becoming more complex, e.g. with a large number of control variables, PAES seemed not to explore the trade-off surface satisfactorily. By examining the nature of MOSA, this paper attempts to improve the performance of PAES by adding the sensitivity adjustment, one of the key characteristics of MOSA implementation.

Based on 4 standard test problems with either a large number of control variables or with three or more objectives, comparative results indicate that the performance of the PAES algorithm with the addition of sensitivity adjustment has been improved significantly. In one test problem, the performance of PAES even outperforms that of MOSA. On-going research is on progress to extend the test covering a wide range of different complex optimisation problems.

Key-Words: - Pareto Archived Evolution Strategy (PAES), Multiobjective Simulated Annealing (MOSA), heuristic search, constrained multiobjective optimisation

1 Introduction

Evolution Strategies (ES's) [1] are one of probabilistic search heuristics that mimic principles of natural evolution. Over forty years, ES's have been implemented and are now in a state where a variety of different variants have been proposed and research continues into many different directions, including applications, extensions of algorithms, and cross-mix with other evolutionary algorithms. ES's can simulate natural evolution of livings existence by mutation and recombination of the selected population with mutation playing a more central role as the main driving mechanism of the algorithms.

The early variant of an evolution strategy, the so called (1+1)-ES [1], works on the basis of two individuals, i.e., one parent and one offspring per generation and the offspring is mutated from the parent with normally distributed variations. Ongoing research efforts have led to the development of a variety of special variants of ES's for solving more complex and multiobjective problems. One example is a Pareto Archived Evolution Strategy (PAES) [2] that, in its baseline form, employs local search for the generation of new candidate solutions but utilizes population information to aid in the calculation of solution quality. Performance comparison between PAES and other well-known and respected multiobjective GA's such as the Niched Pareto Genetic Algorithm (NPGA) [3] and the Nondominated Sorting Genetic Algorithm (NSGA) [4] indicated that PAES was one of the most successful search methods [2, 5]. Recently, however, a comparative study with Multiobjective Simulated Annealing (MOSA) algorithm [6], another potential heuristic search technique, showed that when the optimisation problems are constrained or becoming more complex, e.g. with a large number of control variables, PAES seemed not to explore the trade-off surface very well [7]. This paper therefore attempts to improve the performance of PAES further by adding the sensitivity adjustment, one of the key characteristics of MOSA implementation, and illustrates its performance based on 4 standard test problems [8, 9].

2 Pareto Archived Evolution Strategy (PAES)

The Pareto Archived Evolution Strategy (PAES), developed by Knowles and Corne [2], is one of the most popular evolutionary optimisers that uses a simple (1+1) search to solve multiobjective problems successfully. Being evolutionary, PAES progresses by mimicking the nature of livings' adaptation and survival, i.e., by mutating or adapting the representation of the problem itself (the control variables) in order to increase the chance of survival (the better objectives) and exist in the environment. The general structure of PAES algorithm can be summarized as follows [2]:

1. Initialize a parent, evaluate its all objective functions and put it as the first member of solutions in the archive.
2. Mutate a parent to produce an offspring and evaluate the objectives.
3. Compare the offspring with the parent;
   3.1 If the offspring is dominated by the parent then discard the offspring and go to step 2.
   3.2 If the parent is dominated by the offspring, then update the offspring to become the next parent.
4. Compare the offspring with members in the archive;
   4.1 If the offspring is dominated by any member in the archive then discard the offspring and go to step 2.
   4.2 If the offspring dominates any member in the archive, then add the offspring to the archive and remove all the dominated members.
5. Test crowding procedure (for the offspring, the parent, and the archive)
6. Go to step 2 and repeat until a predefined number of generations is reached.

The control variables are normally encoded as a *k*-bit binary string in PAES with mutation probability, $P_m = 1/k$.

3 Modification of Original PAES

The implementation of the modified PAES was based on the original (1+1)-PAES proposed by Knowles and Corne [2] and added with the followings:

3.1 Adding the Sensitivity Adjustment of Variables

The first obvious modification that could improve the performance of PAES is the addition of sensitivity adjustment of MOSA implementation so that the modified PAES can handle both discrete and continuous control variables. It is clear that with continuous control variable, the search space is wider allowing more exploration into the potential solutions of the problem. However, with wider search space, more runtime may be needed to reach the final solutions. Hence, the intelligent adjustment of continuous variables is particularly suitable for complex problems, particularly those with high sensitivity, i.e. those with a small perturbation of variables can lead to significant change in objectives. Automatic adjustment of variable perturbation at each run can lead to the optimal search pattern as found in MOSA [6].

3.2 Introduction of Random Selection for Control Variable Mutation

In original PAES, the control variables are encoded as a k-bit binary string with mutation probability, $P_m = 1/k$. At each iteration, only one bit of the control variables will be mutated and this will only occur if a generated random number at that iteration $\leq P_m$. It is clear that, as the number of binary string becomes larger (which is the case when dealing with complex problems with many variables), there is less chance that mutation will ever occur. This limits the possibility of searching for optimal solutions caused by inadequate exploration of the search space.

The modified version of PAES mimics the simulation of variable perturbation of general heuristic search methods. That is, at each iteration, every control variable should have an equal opportunity to be perturbed (mutated). This has been implemented by, first, randomly selecting the number of control variables to mutate (which can be in the range from 1 to the total number of variables – i.e. every variable will be mutated), and second, perturbing those selected control variables. The variables to be perturbed will be identified with Roulette-wheel selection at each iteration. Fig.1 summarizes the algorithm of the modified PAES.

4 The Test Functions

Four standard complex test functions for evolutionary multiobjective optimisation [8, 9] were chosen for comparative purposes between the original (1+1)-PAES and the modified PAES. These are:

F1 is two-objective minimisation with 30 control variables proposed by Zitzler *et al.* [8]. It is considered a complex problem because of the many control variables with *Pareto* optima lying on a specific set of these variables:

$$\begin{aligned} f_1(x_1) &= x_1 \\ f_2 &= g \cdot h \\ g(x_2,\ldots,x_m) &= 1 + 9 \cdot \left(\textstyle\sum_{i=2}^{m} x_i\right)/(m-1) \\ h(f_1, g) &= 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1) \end{aligned} \quad (1)$$

```
Initialize a parent, evaluate its all objective functions
Initialize this as the first member in the archive
Do
  Mutate a parent (with random selection) to produce an offspring:
     Random selection to identify a number of control variables (CV) to mutate
     For i = 1 to i = CV
        Roulette-wheel selection to identify the control variable
        Perturb that control variable to produce a new value
     Evaluate the objectives of a newly generated offspring
  Compare the offspring with the parent;
  If the offspring is dominated by the parent
     Discard the offspring
  Else if the parent is dominated by the offspring
     Update the offspring to become the next parent
     Compare the offspring with members in the archive
     If the offspring is dominated by any member in the archive
        Discard the offspring
     If the offspring dominates any member in the archive
        Add the offspring to the archive and remove all the dominated members
     Else
        Add the offspring to the archive
  Else
     Compare the offspring with members in the archive
     If the offspring is dominated by any member in the archive
        Discard the offspring
     If the offspring dominates any member in the archive
        Add the offspring to the archive and remove all the dominated members
     Else
        Add the offspring to the archive
  Test crowding procedure for all the members in the archive
While (a predefined number of generations is not reached)
Return an archive of unique non-dominated solutions
```

Fig.1. Pseudocode for Modified PAES

where m = 30 and $x_1 \in [0,1]$. The *Pareto* optima is formed when $g(x) = 1$ and $x_2, x_3, ... x_{30} = 0$.

F2 is the test problem DTLZ1 [9], constructed as an M-objective problem with linear *Pareto*-optimal front:

$$\left.\begin{array}{lll} \text{Minimise} & f_1(x) & = \frac{1}{2} x_1 x_2 ... x_{M-1}(1+g(x_M)), \\ \text{Minimise} & f_2(x) & = \frac{1}{2} x_1 x_2 ...(1-x_{M-1})(1+g(x_M)), \\ & \vdots \; \vdots & \\ \text{Minimise} & f_{M-1}(x) & = \frac{1}{2} x_1 (1-x_2)(1+g(x_M)), \\ \text{Minimise} & f_M(x) & = \frac{1}{2} (1-x_1)(1+g(x_M)), \\ \multicolumn{3}{l}{\text{subject to } \; 0 \le x_i \le 1, \text{ for } \; i = 1, 2, ..., n,} \end{array}\right\} \quad (2)$$

where

$$g(x_M) = 100\left(|x_M| + \sum_{x_i \in x_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right) \quad (3)$$

The *Pareto*-optimal solution corresponds to $x_i^* = 0.5 \quad (x_i^* \in x_M)$ and the objective function values lie on the linear hyper-plane: $\sum_{m=1}^{M} f_m^* = 0.5$. The difficulty in this problem is to converge to this hyper-plane.

F3 is the test problem DTLZ3 [9] that has a spherical *Pareto*-optimal front as shown in Fig.2:

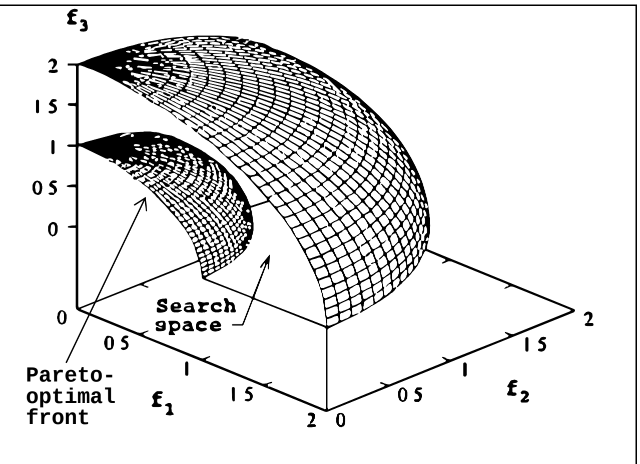

Fig.2. First quadrant of a unit sphere as a *Pareto*-optimal front for test problem F3

$$\left.\begin{array}{ll} \text{Minimise } f_1(x) & = (1+g(x_M))\cos(x_1\pi/2)..\cos(x_{M-1}\pi/2), \\ \text{Minimise } f_2(x) & = (1+g(x_M))\cos(x_1\pi/2)..\sin(x_{M-1}\pi/2), \\ \quad \vdots \; \vdots & \\ \text{Minimise } f_M(x) & = (1+g(x_M))\sin(x_1\pi/2), \\ \multicolumn{2}{l}{\text{subject to } \; 0 \le x \le 1, \;\; \text{for } i = 1, 2, ..., n,} \end{array}\right\} (4)$$

where $g(x_M)$ is given in Eq.(3). This introduces many local optimal fronts, on which any algorithm can easily get trapped.

F4 is the test problem DTLZ9 [9]. It is also a constrained problem, constructed using the constraint surface approach:

$$\left.\begin{array}{lll} \text{Min} & f_j(x) & = \sum_{i=[(j-1)\frac{n}{M}]}^{[j\frac{n}{M}]} x_i^{0.1}, \\ & & \quad j = 1, 2, \ldots, M, \\ \text{subject to } g_j(x) & & = f_M^2(x) + f_j^2(x) - 1 \geq 0, \\ & & \quad \text{for } j = 1, 2, \ldots, (M-1), \\ & 0 \leq x_i \leq 1, & \quad \text{for } i = 1, 2, \ldots, n. \end{array}\right\} \qquad (5)$$

The *Pareto*-optimal front is a curve with $f_1 = f_2 = \ldots = f_{M-1}$, lying on the intersection of all $(M - 1)$ constraints.

For test problem F2 to F4, they are indeed complex problems associated with many objectives from two, three to four and more. However, for the purpose of illustration and for clarification in the discussion of this paper, the results of these problems are shown only for three objectives.

5 Results and Discussion

Fig.3 shows the search pattern and the trade-off surface after 20000 iterations of the original PAES and modified PAES for the test problem F1 with 30 control variables. It is clear that the modified version of PAES could result in a significant improvement in exploring the trade-off surface compared with the original version. From the search pattern, it can be seen that while the original PAES spent most time finding the solutions on a large and scattered region that did not lead to identifying the optimal solutions, the modified PAES focused exclusively near the *Pareto* front and hence, succeeded in identifying the optimal solutions of the problem F1 much more easily.

Fig.4 shows the trade-off surface after 50000 iterations of the original PAES and modified PAES for problem F2. Again, it can be seen that the modified PAES succeeded in identifying the *Pareto* front of the problem as a linear hyper-plane while the original PAES identified another hyper-plane that was further away and not near optimal as a set of best solutions found after 50000 iterations (all the problems are minimisation, hence the closer to the origin of the graph, the better). This can be explained by the same reason already described for the problem F1 – the original PAES spent all its time exploring only the region well above the *Pareto* front. This resulted in a failure in finding the true optimal solutions for the problem. The introduction of random selection for control variable mutation in the modified version of PAES allows the algorithm to jump and cross the line and search for solutions in other regions giving itself a better chance in identifying the true optimal solutions for the problem.

Fig.5 shows the close-up on the trade-off surface after 50000 iterations of the original PAES and modified PAES for problem F3. This clarifies that the modified PAES could identify the trade-off surface for the problem as part of a unit sphere (see Figure 1) successfully, whereas, with the original PAES, not even the shape of the trade-off surface was formed. Dramatic improvement of algorithmic performance is clearly illustrated for this problem.

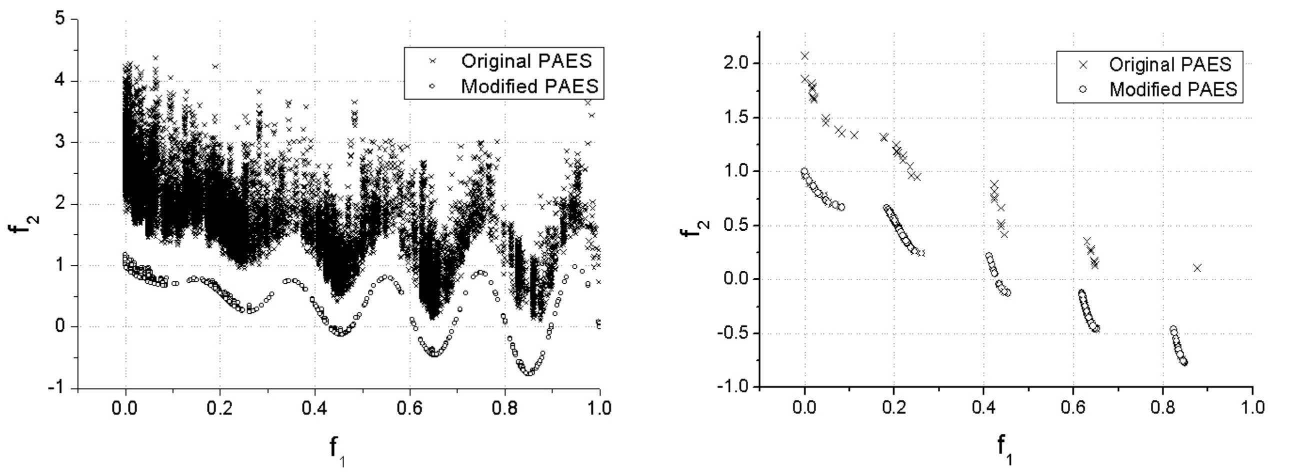

Fig.3. Comparison of the search pattern and the trade-off surface after 20000 iterations between the original PAES and modified PAES for the test problem F1

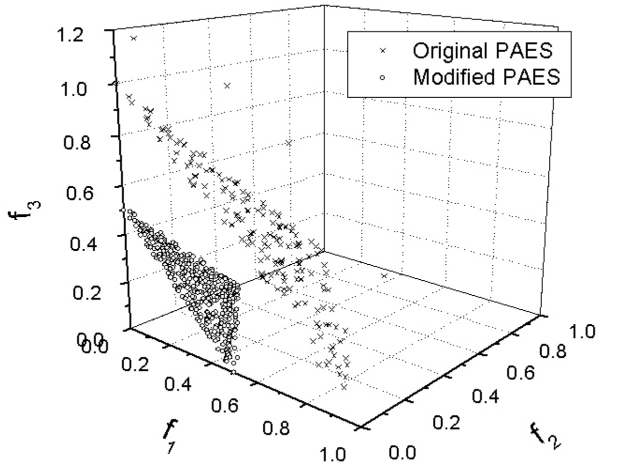

Fig.4. Comparison of the trade-off surface after 50000 iterations between the original PAES and modified PAES for the test problem F2

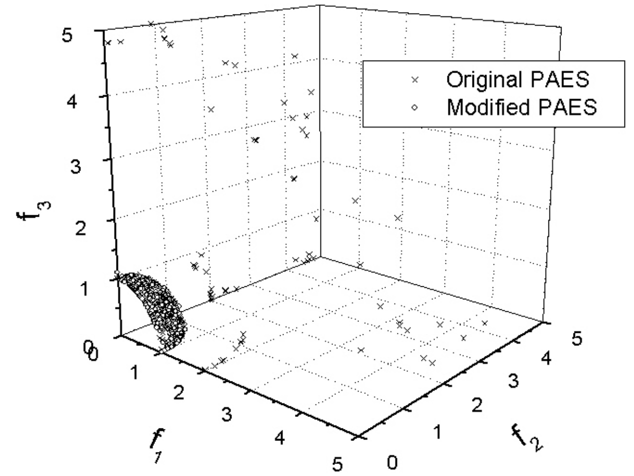

Fig.5. Close-up of the trade-off surface after 50000 iterations between the original PAES and modified PAES for the test problem F3

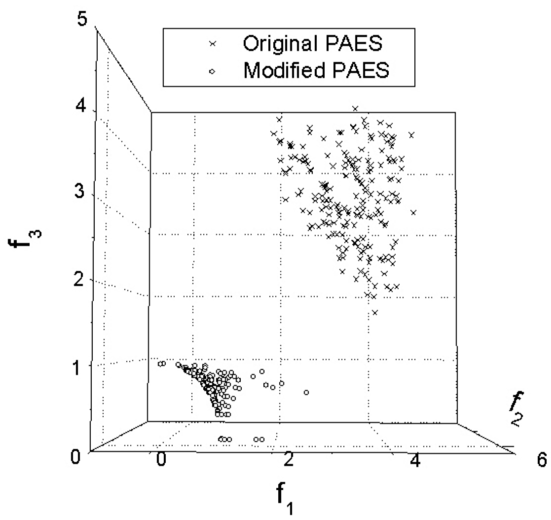

Fig.6. Comparison of the trade-off surface after 50000 iterations between the original PAES and modified PAES for the test problem F4

Figs.6-7 show the trade-off surfaces of the original PAES, the modified PAES, NSGA II and SPEA II (the two alternative, efficient evolutionary algorithms recently developed to handle highly complex problems) for the test problem F4. Again, it is clear that the modified PAES performed better than the original PAES in these two problems – similar results to F1, F2 and F3. Although the modified PAES could not identify the complete *Pareto* fronts in problem F4, neither the NSGA II or SPEA II, could. Fig.7 suggests that the performance of the modified PAES was comparable with that of NSGA II and SPEA II.

6 Conclusions

The comparison of algorithmic performance between the original PAES and the modified PAES on four standard test functions led us to the conclusion that the modified PAES that was implemented with the addition of sensitivity adjustment of continuous control variables and with the random selection for variable mutation, seemed to explore the trade-off surface better than the original PAES. Although the modified version of PAES did not explore the entire trade-off for many constrained problems presented here, its performance was comparable with that of other alternative, efficient evolutionary algorithms recently developed to handle highly complex problems, such as NSGA II and SPEA II.

While the original PAES could not explore the trade-off surface very well as it spent most time exploring on a scattered region, the modified PAES focused exclusively near the *Pareto* fronts and succeeded in identifying the optimal solutions for the problems. However, as the modified PAES did not explore the entire trade-off surface, this raises a question if such focus of search exploration of the modified PAES is a real advantage. On-going research has been conducted to investigate further on the performance of the modified PAES on a wide variety of complex constrained problems in comparison with that of other known, efficient optimisation algorithms.

References:

[1] H-P. Schwefel, *Evolution and Optimum Searching*, Wiley Interscience, John Wiley & Sons, New York, 1995.

[2] J.D. Knowles and D.W. Corne, “Approximating

the nondominated front using the Pareto Archived Evolution Strategy", *Evolutionary Computation*, 2000, 8(2), pp. 149–172.

[3] J. Horn, N. Nafpliotis, and D.E. Goldberg, "A Niched Pareto Genetic Algorithm for multiobjective optimization", In Proceedings of the *First IEEE Conference on Evolutionary Computation*, IEEE World Congress on Computational Intelligence, 1994, 1, pp. 67–72.

[4] Srinivas, N. and Deb, K., "Multiobjective optimization using Nondominated Sorting in Genetic Algorithms", *Evolutionary Computation*, 1994, 2(3), pp. 221–248.

[5] J.D. Knowles and D.W. Corne, "M-PAES: a memetic algorithm for multiobjective optimization", Proceedings of the *Congress on Evolutionary Computation* (CEC00), 2000, pp. 325–332.

[6] A. Suppapitnarm, K.A. Seffen, G.T. Parks and P.J. Clarkson, "A simulated annealing algorithm for multiobjective optimization", Engineering Optimization, 2000, 33(10), pp. 59–85.

[7] T. Wassanarpheernphong and A. Suppapitnarm, "Comparison of algorithmic performance between Pareto Archived Evolution Strategy (PAES) and Multiobjective Simulated Annealing (MOSA)", Proceedings of the *9th Annual National Symposium on Computational Science and Engineering (ANSCSE9)*, Mahidol University, Thailand, 2005, pp. 74–83

[8] E. Zitzler, K. Deb and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results", TIK-Report. 1999, pp. 70.

[9] K. Deb, L. Thiele, M. Laumanns and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization", TIK-Report. 2001, pp. 112.

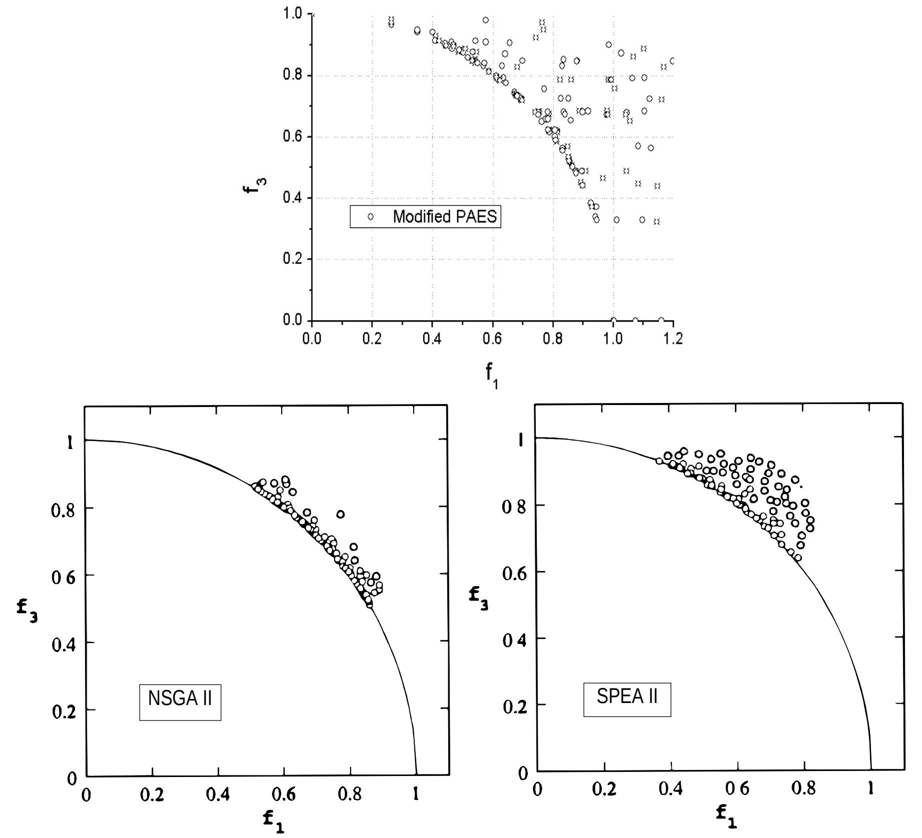

Fig.7. Close-up of the trade-off surface in two-objective projection for the test problem F4 in comparison with other multiobjective evolutionary algorithms (NSGA II and SPEA II, [9])