A Benchmark Study of Multi-Objective Optimization Methods

N. Chase, M. Rademacher, E. Goodman
Michigan State University, East Lansing, MI

R. Averill, R. Sidhu
Red Cedar Technology, East Lansing, MI

Abstract. A thorough study was conducted to benchmark the performance of several algorithms for multi-objective Pareto optimization. In particular, the hybrid adaptive method MO-SHERPA was compared to the NCGA and NSGA-II methods. These algorithms were tested on a set of standard benchmark problems, the so-called ZDT functions. Each of these functions has a different set of features representative of a different class of multi-objective optimization problem. It was concluded that the MOSHERPA algorithm is significantly more efficient and robust for these problems than the other methods in the study.

1. Introduction

Conventional parameter optimization methods seek to find a single optimized solution based on a weighted sum of all objectives. If all objectives get better or worse together, this conventional approach can effectively find the optimal solution. However, if the objectives conflict (as, for example, increasing performance and reducing cost typically do), then there is not a single optimal solution. In this case, a multi-objective optimization study should be performed that provides multiple solutions representing the tradeoffs among the objectives, denoted f_i (see Figure 1). This is commonly called Pareto optimization.

It is then up to the designer/engineer to select among these designs, with a better understanding of the true tradeoffs. The inset box contains a mathematical definition of the class of optimization problems being addressed here, which allows the possible inclusion of equality and/or inequality constraints.

1.1 Non-Dominated Sorting in Multi-Objective Optimization

A common technique for ranking designs in a multi-objective optimization study is to use the concept of non-dominated sorting, as developed by Deb [1-3].

Multi-objective Optimization Problem

Minimize (or maximize):

$$f_i(x_1,x_2,\ldots,x_n), \qquad i=1,2\ldots p$$

such that:

$$h_j(x_1,x_2,\ldots,x_n) < 0, \qquad j=1,2,\ldots q$$

where:

$(x_1,x_2,\ldots,x_n)$ are the n design variables
$f_i(x_1,x_2,\ldots,x_n)$ are the p objective functions
$h_j(x_1,x_2,\ldots,x_n)$ are the q inequality constraints

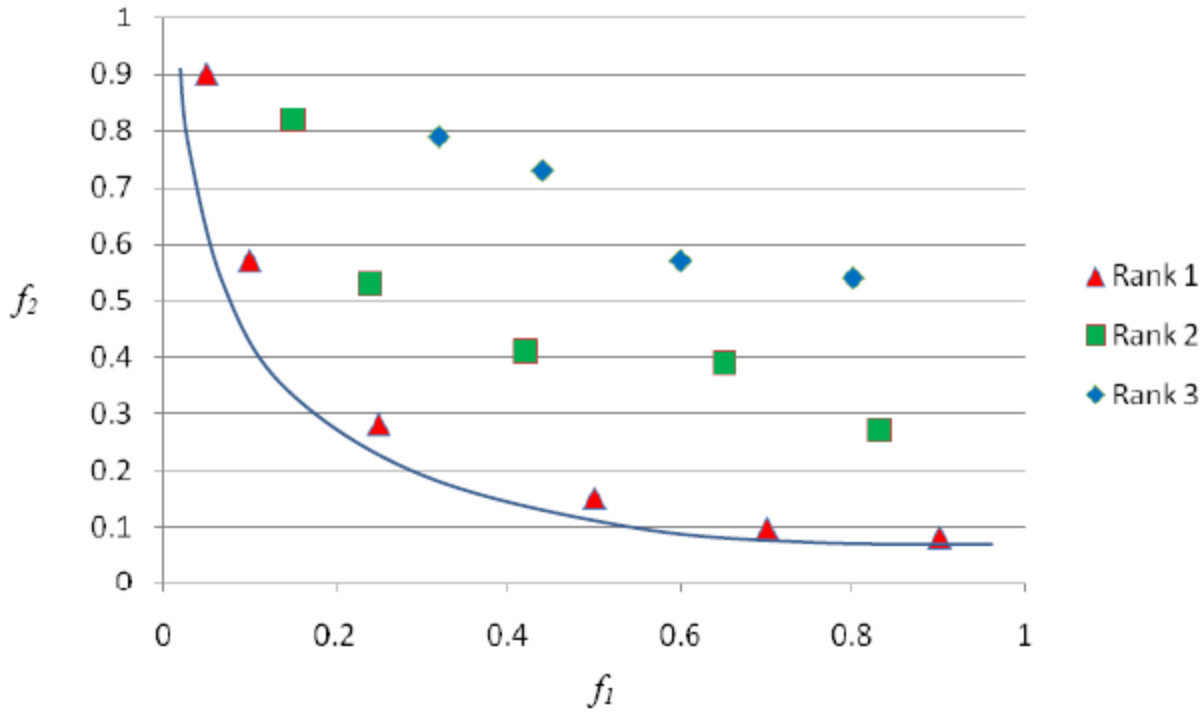

Figure 1. Example results of a Pareto optimization study, in which the tradeoff between the two objectives f_1 and f_2 is explored. None of the rank 1 designs is dominated by any other feasible design. That is, for each rank 1 design no other design is better in one objective and not worse in the other objective(s). The rank 1 designs (red triangles) are a nondominated set, and after they are subtracted from the set of feasible designs, the rank 2 designs (green squares) are those that are not dominated by any of the remaining points. Design sets of rank 3 and higher are determined in a similar manner. The "true Pareto front" is indicated by the solid blue line that lies below and to the left of the rank 1 designs. The set of rank 1 designs denoted by triangles is an example of a "local Pareto front."

The concept of domination is easily defined: design A dominates design B if it is better in at least one objective and not worse in all other objectives. The process of sorting designs based on dominance is called non-dominated sorting (NDS). At any stage in an optimization run, a population or archive of "current" designs is maintained. At each step, all feasible designs that are not dominated by any other designs in the population (or archive) are given the rank of 1. These are the only non-dominated designs in the population. Then these designs are conceptually removed from the archive, and the remaining designs are judged for domination. Those that are not dominated by any other of the remaining designs are given the rank of 2. The procedure is repeated, reranking the remaining designs after removing non-dominated designs, to establish ranks 3, 4, etc. This constitutes the NDS process and is illustrated in Figure 1. The points denoted by triangles are the first non-dominated set identified, so they are rank 1. After their removal, in the set under consideration. The set containing designs that are not dominated by any feasible solution in the entire search space is called the *Pareto set*, and the plot of the corresponding values of the objectives is called the *Pareto front*.

Sometimes a plot reveals what appears to be the Pareto front until additional (dominating) solutions are found, in which case these fronts are sometimes called *local Pareto fronts*. The phrase *true Pareto front* is then used to denote the points that represent the global Pareto front in the entire search space, not the local front among the points searched to date. As the run progresses, new designs will dominate and replace other designs on a series of local Pareto fronts. The end result will typically be a set of designs that are not dominated by any other designs, and which approach, or converge towards, the true Pareto front. From this set of designs, one can select the design that best fits the current needs (contains the best combination of objectives) or those that inspire further exploration.

4572 South Hagadorn Rd., Suite 3A ▲ East Lansing, MI 48823 ▲ TEL 517.664.1137 ▲ FAX 517.664.1175 ▲ www.redcedartech.com

1.2 Efficiency and Robustness in Multi-Objective Optimization

Optimization algorithms use the results from numerical analyses and simulations, herein called "evaluations," to guide the search for an optimal design. For example, a finite element analysis of a particular design candidate would be called an evaluation. In conventional parameter optimization, an algorithm's ***efficiency*** is measured in terms of the total number of evaluations required to find the optimal design or a design of a specified performance level. In Pareto optimization, efficiency is similarly judged by the number of evaluations needed to find a suitably accurate approximation of the Pareto front. In this paper, as in common usage, we will refer to reaching the Pareto front when the solutions, plotted as in Figure 1, appear to lie on the solid line plotted to represent the Pareto front (though in practice the true Pareto front is not known *a priori*).

Using fewer evaluations to find the Pareto front is very important because often each evaluation can require a significant amount of CPU time. For example, a nonlinear finite element simulation may require from several hours to several days of CPU time. So reducing the total number of evaluations needed has a large impact on the time required to find an optimized design or Pareto front. The effect of algorithm efficiency may be even larger in a Pareto optimization study, which often requires significantly more evaluations than conventional optimization.

The search path taken by an optimization algorithm will generally be different in each run, depending on its starting conditions. This means that the number of evaluations required to achieve a given level of design performance can be quite different from run to run. More importantly, the final results of several runs using the same algorithm may not be the same – that is, each run may fall short in some way from finding the Pareto front. These differences depend upon the starting conditions of the search, including the initial set (or population) of designs. When comparing the performance of optimization methods in a benchmark study such as this one, it is necessary to perform multiple runs of each algorithm on each problem to more accurately assess the mean and variation of the performance.

1.3 Objectives of the Current Study

In this study, the efficiency and robustness of several multi-objective Pareto optimization algorithms were investigated on a set of standard benchmark problems. The algorithms under consideration were: NSGA II [1-3], NCGA [4], and MO-SHERPA [5].

2. Overview of Multi-Objective Optimization Algorithms

A brief description of the methods considered in this study is presented in this section. A detailed mathematical formulation of the methods is left to the references cited.

2.1 NSGA-II

NSGA-II [1-3] is a multi-objective genetic algorithm that uses the non-dominated sorting (NDS) scheme and a crowding measure to rank individual designs. The crowding measure is a secondary measure used to favor an even distribution of points along the Pareto front. A design with a lower-numbered rank is considered to have a higher performance (or fitness) than designs of higher rank. A rank 1 design thus has a higher probability of producing offspring in the next generation, or cycle. In this paper, numerical studies use an implementation of NSGA-II with the following default parameter values: Crossover Probability = 0.9; Crossover Distribution Index = 20.0; and Mutation Distribution Index = 100.0.

2.2 NCGA

The NCGA (Neighborhood Cultivation Genetic Algorithm) method [4] is similar in many ways to NSGA-II, but it adds the neighborhood crossover operator to enhance the degree of exploitation (rapid convergence) versus exploration during the search. In NCGA, the selection of a pair of individuals for crossover is not performed randomly. Instead, individuals who are closer (in the objective space) to each other have a higher chance of being selected. Hence, the children that result from the cross-over operation have a higher chance of being close to the parents in the objective space.

In this paper, numerical studies use an implementation of NCGA with the following default parameter values: Crossover Type = 1; Crossover Rate = 1.0; Mutation Rate = 0.01; and Gene Size = 20.

4572 South Hagadorn Rd., Suite 3A ▲ East Lansing, MI 48823 ▲ TEL 517.664.1137 ▲ FAX 517.664.1175 ▲ www.redcedartech.com

2.3 MO-SHERPA

SHERPA is a proprietary hybrid and adaptive search strategy available within the HEEDS software code [5]. During a single parametric optimization study, SHERPA uses the elements of multiple search methods simultaneously (not sequentially) in a unique blended manner. This approach attempts to take advantage of the best attributes of each method. Attributes from a combination of global and local search methods are used, and each participating approach contains internal tuning parameters that are modified automatically during the search according to knowledge gained about the nature of the design space. This evolving knowledge about the design space also determines when and to what extent each approach contributes to the search. In other words, SHERPA efficiently learns about the design space and adapts itself so as to effectively search all sorts of design spaces, even very complicated ones.

SHERPA is a *direct* optimization algorithm in which all function evaluations are performed using the actual model as opposed to an approximate response surface model.

MO-SHERPA (Multi-Objective SHERPA) is a modified version of the algorithm SHERPA for multi-objective Pareto search. It works fundamentally like SHERPA, but has the advantage of handling multiple objectives independently of each other to provide a set of solutions, each of which is optimal in some sense for one of the objectives. MO-SHERPA uses a non-dominated sorting scheme to rank designs, but is quite different from NSGA-II and NCGA in other aspects.

2.4 Discretization of Variables

All three of the methods considered in this study can accommodate continuous as well as discrete variables. In addition, MO-SHERPA within HEEDS allows continuous variables to be discretized by specifying a resolution for each design variable. In this way, the size of the design space (number of possible solutions) can be effectively reduced, which in some cases may lead to a more efficient solution of the problem. This approach is also an effective way to control the resolution of values assigned to design variables, since it is not useful in many engineering designs to specify a variable to greater than a few significant figures.

However, because the implementation of NCGA and NSGA-II utilized does not allow for discretized variables, in the current study the resolution of all variables within MO-SHERPA was set to 1,000,001 (i.e., there were 1,000,001 equally distributed values of each design variable within the specified range). This setting was used to approximate a purely continuous variable, and to ensure that MO-SHERPA would not benefit unfairly in any way due to the resolution of the variables for this benchmark study.

3. Results of the Benchmark Studies

For this study, the ZDT family of functions [6] was selected, because it is a broad and popular set of test functions for benchmarking the performance of multi-objective Pareto optimization methods. Each of these test functions contains a particular feature that is representative of a real world optimization problem that could cause difficulty in converging to the Pareto front. In the following sections, each of these functions is described and the performance of the optimization methods on these test problems is investigated.

All of the ZDT functions contain two objectives, which is the most common usage of Pareto optimization, especially in engineering applications. However, none of the optimization methods in this study are limited to problems with only two objectives.

For each of the ZDT functions in this study, five runs were conducted using each algorithm (with the starting designs varying over a wide range of the design space). This was done to assess the robustness of the results obtained by each algorithm, as well as to ensure the results were not biased based upon the set of starting designs.

Each run was performed for three different population sizes (sometimes called “archive sizes”): 12, 40 and 100. In a stochastic search method for Pareto optimization, speed and robustness of search are often traded off through such a population size variable, which affects the number of solutions used for generating further solutions at each step. Results are presented here for each of the population sizes.

Except for the population size values, default values were used for all other parameters in the NSGA-II and NCGA methods. MO-SHERPA is self-adaptive, and has no tuning parameters.

4572 South Hagadorn Rd., Suite 3A ▲ East Lansing, MI 48823 ▲ TEL 517.664.1137 ▲ FAX 517.664.1175 ▲ www.redcedartech.com

To simplify the interpretation of results, only the non-dominated (rank 1) designs for each run are included in the plots. The rank 1 designs are determined in the same way for all methods and all runs by using the non-dominated sorting (NDS) routine of Deb [1-3]. This method was described briefly in Section 1.1. The data points are plotted one run at a time for each algorithm, then the next run of each algorithm, etc. The order of plotting among algorithms was changed systematically from graph to graph, for clarity and fairness.

Because the result of a two-objective Pareto optimization study is a set of points on a curve (the Pareto front), there is no easy or universally accepted way to compare the performance of multi-objective algorithms. In the current study, the results of all five runs for each method are included on each plot. In this way, it is possible to see the variation of the results among the different runs of each algorithm, and in some cases it is possible to discern the results from separate runs as local Pareto fronts. Hence, a subjective or visual comparison among the different algorithms is performed based on the "cloud" of rank 1 designs for all five runs at various stages of the search.

The performance of each algorithm is judged primarily by the placement of its data points relative to the optimal Pareto front, which is also presented in the plots. The closer the algorithm is to the optimal front, the better performing the algorithm. The robustness of an algorithm in the vicinity of the Pareto front can also be measured using the breadth of the "band" formed from its five runs. The narrower the band of a specific algorithm, the more robust it is in finding a particular front, and the less dependent upon the starting design given and the stochastic variation induced by the random number sequence.

Finding all points on the true Pareto front is a task that cannot be accomplished by any algorithm that produces sampled points, as a Pareto front is typically composed of one or multiple piecewise continuous function(s). We shall use the phrase "found the Pareto front" to mean that when viewed on a plot displaying the entire front, the points plotted are at least nearly all apparently in the vicinity of the Pareto front and distributed along it. "Did not find the Pareto front" will be used when the points found are visually widely separated from the optimal front, and it is apparent that they are not on the front.

3.1 ZDT1 Function

The ZDT1 function has a convex Pareto-optimal front. The objective functions are:

$$f_1(x) = x_1$$

$$f_2(x) = g(x)\left[1 - \sqrt{x_1/g(x)}\right]$$

Where g(x) is defined as:

$$g(x) = 1 + 9\left(\sum_{i=2}^{n} x_i\right)/(n-1)$$

In this ZDT1 function, thirty design variables x_i were chosen (n=30). Each design variable ranged in value from 0 to 1. The Pareto-optimal front appears when g = 1.0.

For each run, up to 10,000 evaluations were performed. Results for this problem are presented in Figures 2 through 4 at four stages of the search process – after 500, 1000, 5000, and 10,000 evaluations. For the function ZDT1, at all population (archive) sizes, MO-SHERPA dramatically outperformed the NCGA and NSGA-II algorithms. At a population size of 12, NCGA generally outperformed NSGA-II at 5,000 evaluations and beyond, although NSGA-II outperformed NCGA in shorter runs. At larger population sizes, NSGA-II outperformed NCGA in all runs, but never approached the performance of MO-SHERPA.

4572 South Hagadorn Rd., Suite 3A ▲ East Lansing, MI 48823 ▲ TEL 517.664.1137 ▲ FAX 517.664.1175 ▲ www.redcedartech.com

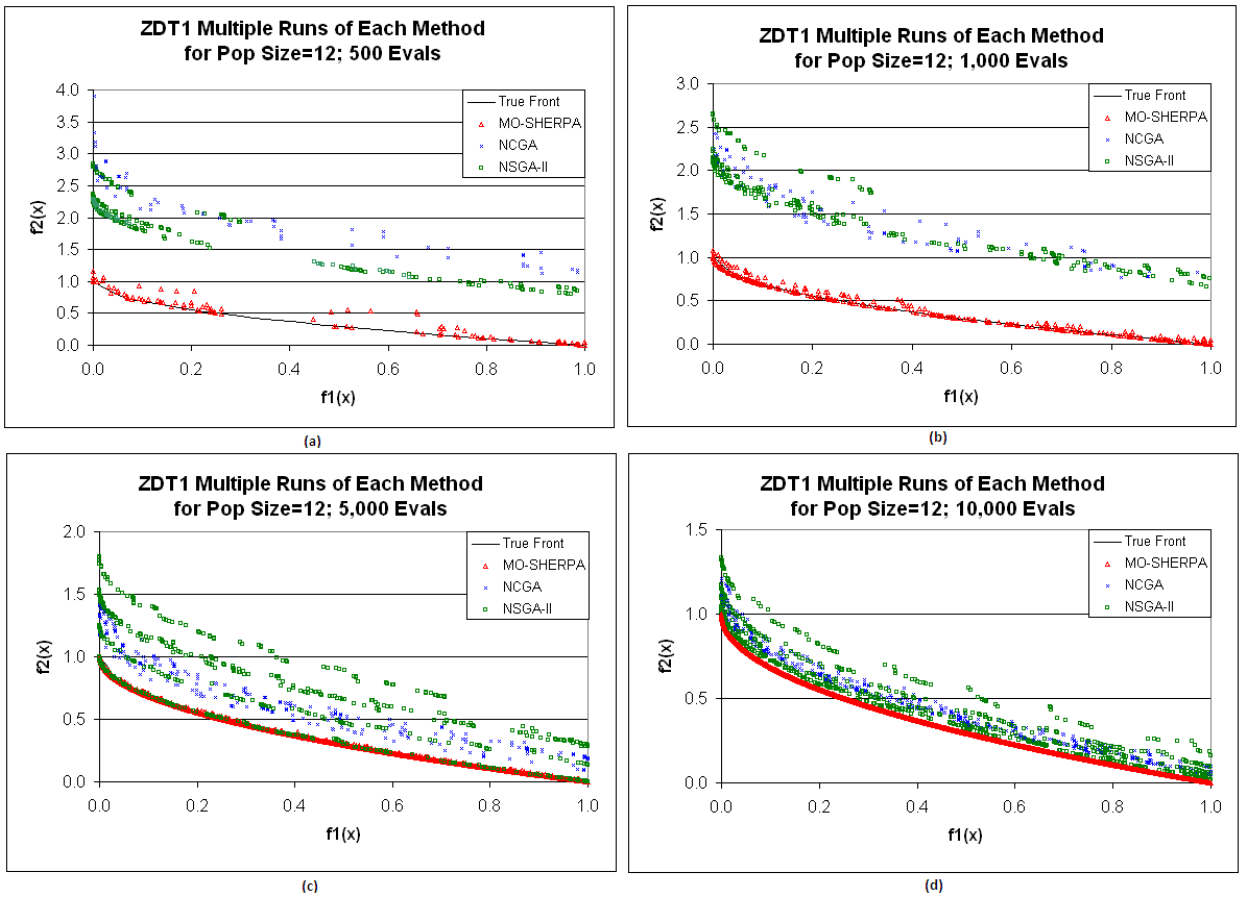

Figure 2 (a) – (d). Superimposed results of 5 runs each of MO-SHERPA, NCGA, and NSGA-II, on benchmark ZDT1 with population size 12. In (a), MO-SHERPA approaches the true Pareto front within 500 evaluations, while both other algorithms are far away. After 1,000 evaluations (b), MO-SHERPA is nearly converged to the Pareto front, while neither of the other algorithms is near the front. After 5,000 evaluations (c), all MO-SHERPA runs are essentially on the front, while only one of the five NSGA-II runs is near the front. Most NCGA runs are closer to the front than most NSGA-II runs. After 10,000 evaluations (d), NSGA-II is close to the true front for three of the runs, and NCGA is approaching the front.

4572 South Hagadorn Rd., Suite 3A ▲ East Lansing, MI 48823 ▲ TEL 517.664.1137 ▲ FAX 517.664.1175 ▲ www.redcedartech.com

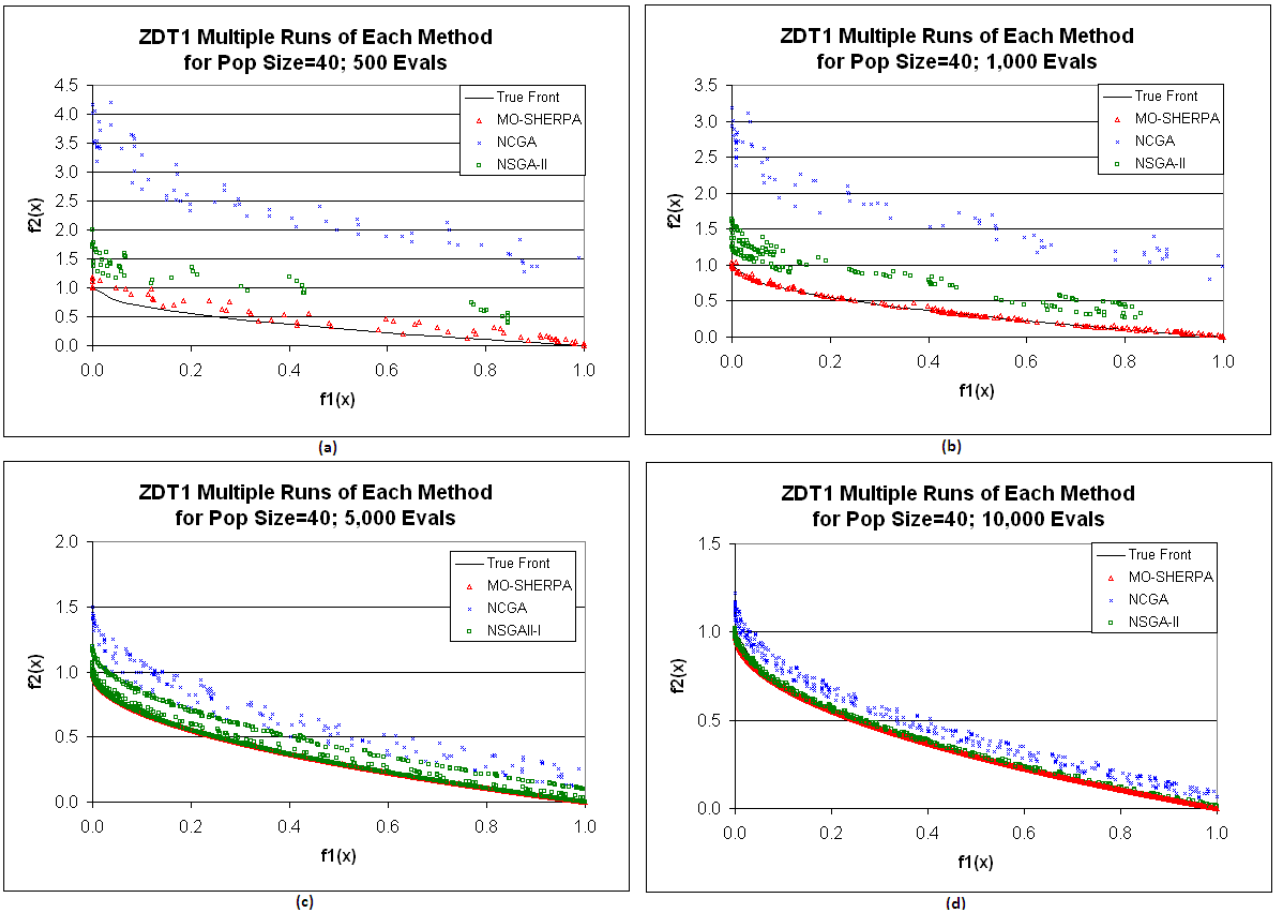

Figure 3 (a) – (d). Superimposed results of 5 runs each of MO-SHERPA, NCGA, and NSGA-II, on benchmark ZDT1 with population size 40. In (a), MO-SHERPA approaches the true Pareto front within 500 evaluations, while NCGA is far away and NSGA-II is close behind. After 1,000 evaluations (b), MO-SHERPA is nearly converged to the true front, while NCGA is still far away and NSGA-II has not made significant progress toward the true front. After 5,000 evaluations (c), all MO-SHERPA runs are essentially on the front, while only two of the five NSGA-II runs are on the front. After 10,000 evaluations (d), NSGA-II is close to the true front for all of the runs, and NCGA is approaching the front.

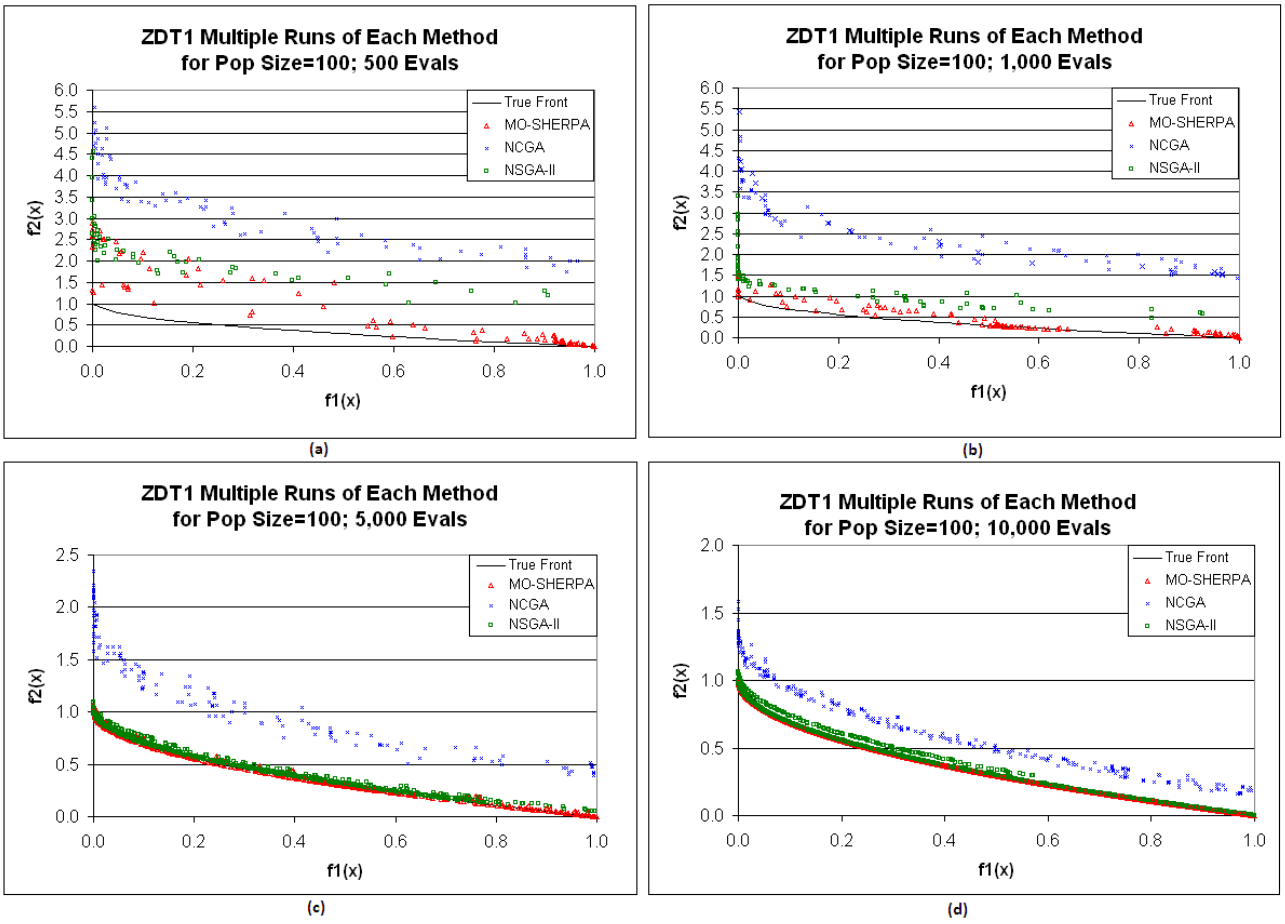

Figure 4 (a) – (d). Superimposed results of 5 runs each of MO-SHERPA, NCGA, and NSGA-II, on benchmark ZDT1 with population size 100. In (a), MO-SHERPA approaches the true Pareto front within 500 evaluations for three of the runs, while NSGA-II is close behind and NCGA is still far away from the true Pareto front. After 1,000 evaluations (b), MO-SHERPA is near the true front for all runs, NSGA-II is approaching the true front in some runs, while NCGA is still far away. After 5,000 evaluations (c), all MO-SHERPA runs are essentially on the front, as are four of the five NSGA-II runs. After 10,000 evaluations (d), NSGA-II is essentially on the front for four of the five runs, and NCGA is approaching the front.

3.2 ZDT2 Function

The ZDT2 function has a non-convex Pareto-optimal front. The objective functions are:

$$f_1(x) = x_1$$

$$f_2(x) = g(x)\left[1 - \left(\frac{x_1}{g(x)}\right)^2\right]$$

where g(x) is defined as:

$$g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$$

In this ZDT2 function, thirty design variables x_i were chosen (n=30). Each design variable ranged in value from 0 to 1. The Pareto-optimal front appears when g = 1.0.

For each run, up to 10,000 evaluations were performed. Results for this problem are presented in Figures 5 through 7 at four stages of the search process – after 500, 1000, 5000, and 10,000 evaluations.

For the ZDT2 function, MO-SHERPA dramatically outperformed the two other algorithms at all population sizes and run durations. It provided many Pareto-optimal or near-Pareto-optimal solution points within 500 to 1,000 evaluations, especially at smaller population sizes. Between NSGA-II and

NCGA, there was not a clear winner across all population sizes and run lengths, but NSGA-II performed better as runs grew longer. Neither of these algorithms provided a clear indication of the front at 1,000 evaluations, while MO-SHERPA had many points at or near it.

4572 South Hagadorn Rd., Suite 3A ▲ East Lansing, MI 48823 ▲ TEL 517.664.1137 ▲ FAX 517.664.1175 ▲ www.redcedartech.com

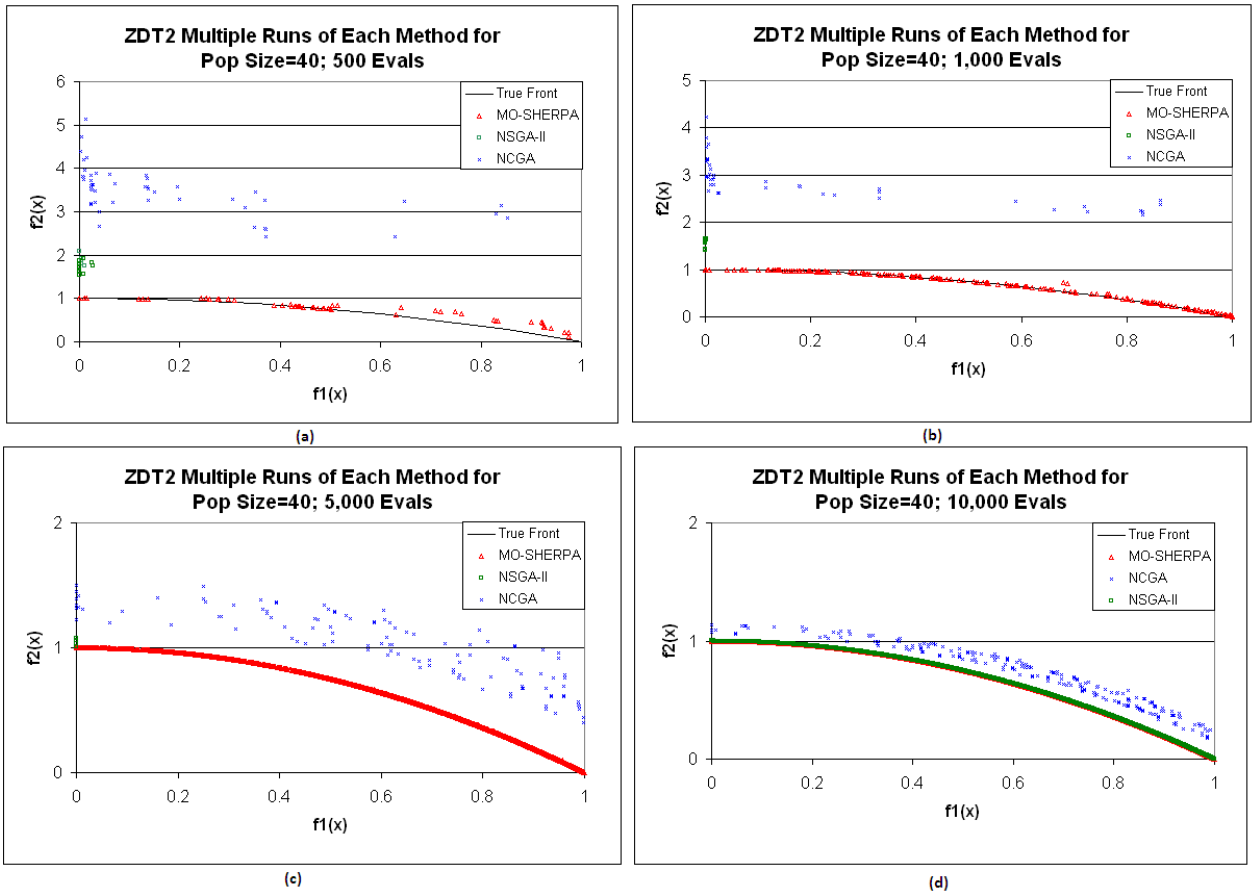

Figure 6 (a) – (d). Superimposed results of 5 runs each of MO-SHERPA, NCGA, and NSGA-II, on benchmark ZDT2 with population size 40. In (a), MO-SHERPA is closing in on the true Pareto front within 500 evaluations, while both other algorithms are far away. After 1,000 evaluations (b), MO-SHERPA is nearly converged to the Pareto front, while neither of the other algorithms is near the front. After 5,000 evaluations (c), all MO-SHERPA runs are essentially on the front, while all NSGA-II runs are near or on the front but not well distributed (and partially obscured by the MO-SHERPA points plotted later on this graph), and none of the NCGA runs is near the front. After 10,000 evaluations (d), NSGA-II is on the true front for all of the runs, obscuring the MO-SHERPA points plotted first, and NCGA is closing in on the front.

4572 South Hagadorn Rd., Suite 3A ▲ East Lansing, MI 48823 ▲ TEL 517.664.1137 ▲ FAX 517.664.1175 ▲ www.redcedartech.com

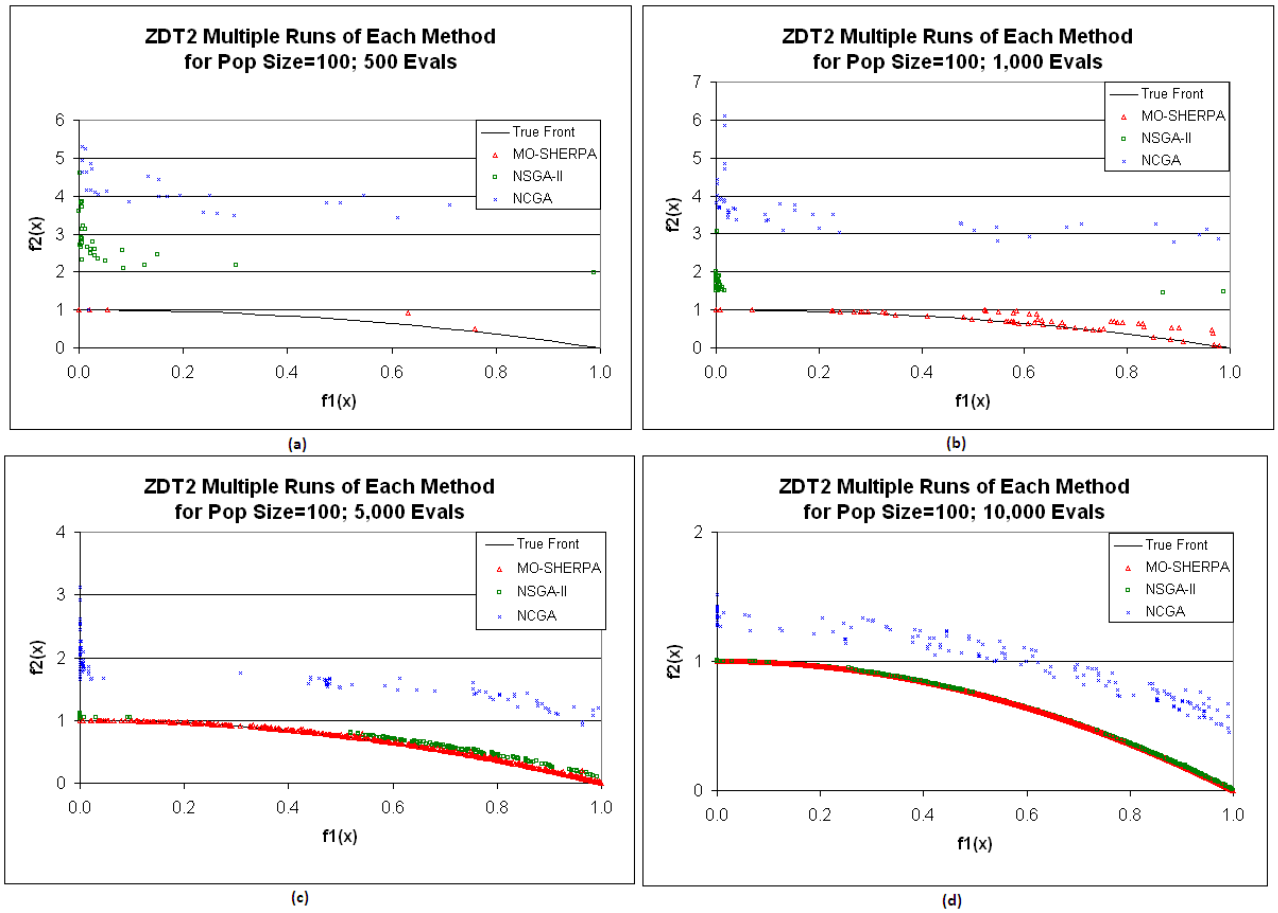

Figure 7 (a) – (d). Superimposed results of 5 runs each of MO-SHERPA, NCGA, and NSGA-II, on benchmark ZDT2 with population size 100. In (a), MO-SHERPA is closing in on the true Pareto front within 500 evaluations (but the front is very sparsely populated), while both other algorithms are far away. After 1,000 evaluations (b), MO-SHERPA is nearly converged to the Pareto front and much better distributed along it, while neither of the other algorithms is near the front. After 5,000 evaluations (c), all MO-SHERPA runs are essentially on the front, while all NSGA-II runs are near the front but not well distributed, and none of the NCGA runs is near the front. After 10,000 evaluations (d), NSGA-II is on the true front for all of the runs, and NCGA is closing in on the front.

4572 South Hagadorn Rd., Suite 3A ▲ East Lansing, MI 48823 ▲ TEL 517.664.1137 ▲ FAX 517.664.1175 ▲ www.redcedartech.com

3.3 ZDT3 Function

The ZDT3 function adds a discreteness feature to the front. Its Pareto-optimal front consists of several noncontiguous convex parts. The introduction of a sine function in this objective function causes discontinuities in the Pareto-optimal front, but not in the parameter space. The objective functions are:

$$f_1(x) = x_1$$

$$f_2(x) = g(x)\left[1 - \sqrt{x_1/g(x)} \quad - \frac{x_1}{g(x)} \sin(10\pi x_1)\right]$$

where g(x) is defined as:

$$g(x) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1)$$

In this ZDT3 function, thirty design variables x_i were chosen (n=30). Each design variable ranged in value from 0 to 1. The Pareto-optimal front appears when g = 1.0.

For each run, up to 10,000 evaluations were performed. Results for this problem are presented in Figures 8 through 10 at four stages of the search process – after 1000, 2000, 5000 and 10,000 evaluations.

On the ZDT3 function, with its piecewise continuous front, MO-SHERPA dramatically outperformed the two algorithms NCGA and NSGA-II. At all population sizes, MO-SHERPA had identified and densely populated all segments of the front within a few thousand evaluations. NCGA did not converge on the front completely even in 10,000 evaluations, but it did always find all segments of the front (as did MO-SHERPA). NSGA-II, while generally getting closer to the front in fewer evaluations than did NCGA, missed some of the segments of the front in some of its runs, producing an incorrect shape for the front (including points extending vertically beyond the limits of the front's segments, particularly on the right-hand side of each graph). Thus, while NCGA and NSGA-II search each had some advantages over the other, neither approached the performance of MO-SHERPA.

4572 South Hagadorn Rd., Suite 3A ▲ East Lansing, MI 48823 ▲ TEL 517.664.1137 ▲ FAX 517.664.1175 ▲ www.redcedartech.com

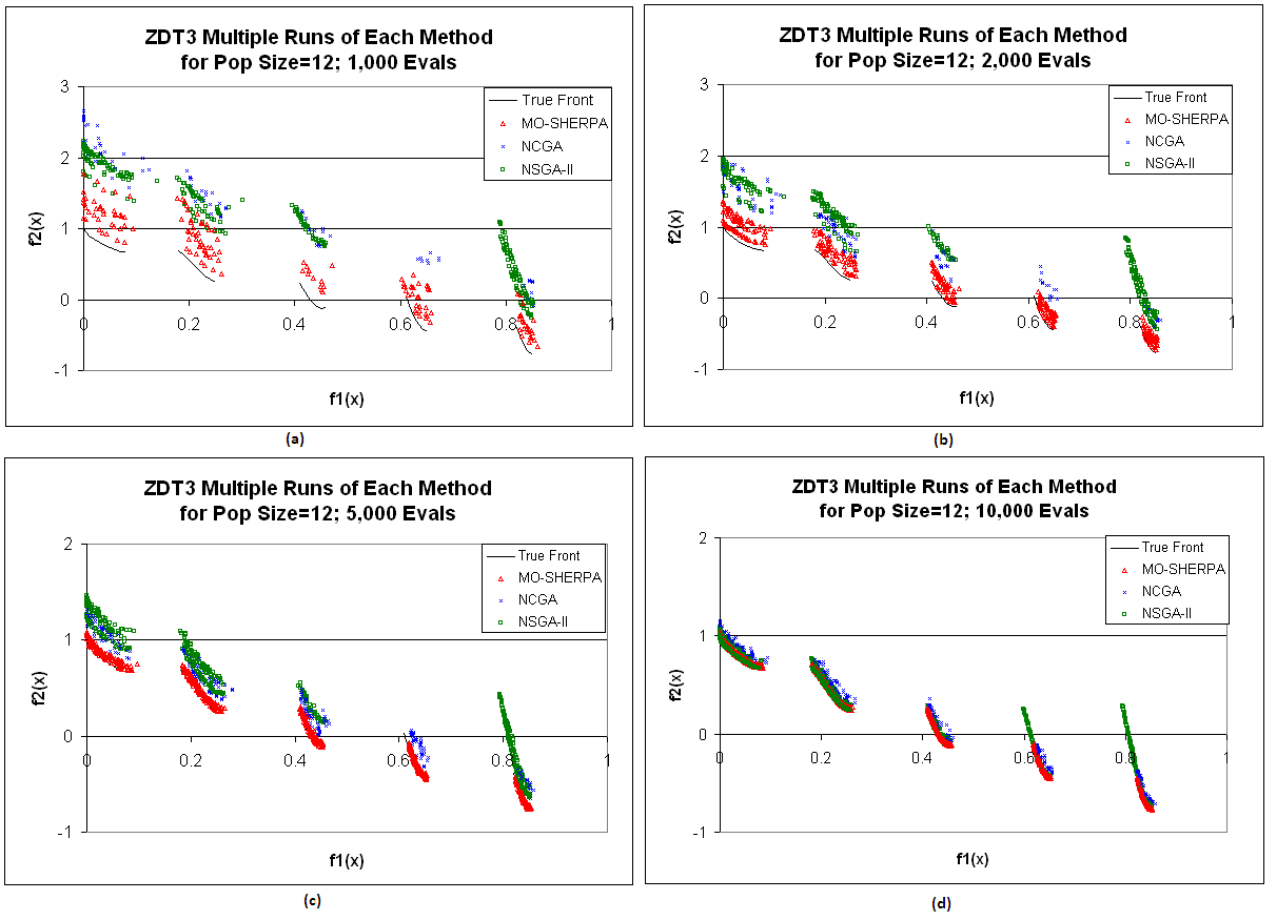

Figure 8 (a) – (d). Superimposed results of 5 runs each of MO-SHERPA, NCGA, and NSGA-II, on benchmark ZDT3 with population size 12. In (a), MO-SHERPA is closing in on the true Pareto front within 1,000 evaluations while the other two algorithms are far away. After 2,000 evaluations (b), MO-SHERPA is continuing to close in on the Pareto front while NCGA closes in on the front as well. NSGA-II is further from the front. After 5,000 evaluations (c), all MO-SHERPA runs are essentially on the front, while all NCGA and NSGA-II runs are near the front but not on it. After 10,000 evaluations (d), NSGA-II is on the true front for all of the runs, and NCGA is on it for all but two runs, which are very near the front. However, NSGA-II is still displaying many points that are beyond the vertical extent of the true front. That is because in some runs, it failed to find the points on the middle segment of the front, and then failed to find the points in the second-from-right segment, as well.

4572 South Hagadorn Rd., Suite 3A ▲ East Lansing, MI 48823 ▲ TEL 517.664.1137 ▲ FAX 517.664.1175 ▲ www.redcedartech.com

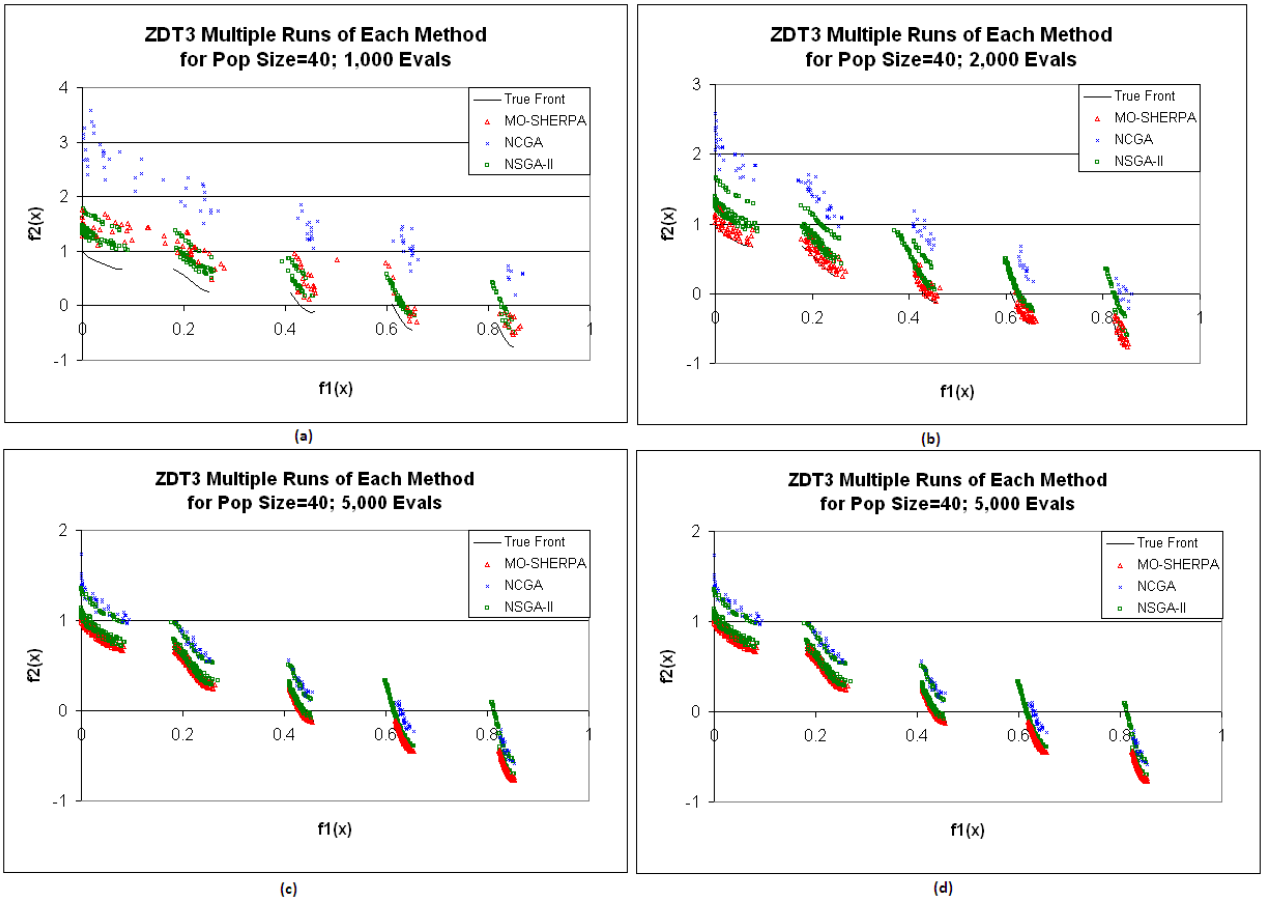

Figure 9 (a) – (d). Superimposed results of 5 runs each of MO-SHERPA, NCGA, and NSGA-II, on benchmark ZDT3 with population size 40. In (a), MO-SHERPA and NSGA-II are closing in on the true Pareto front within 1,000 evaluations while NCGA is far away. After 2,000 evaluations (b), MO-SHERPA has nearly converged on the Pareto front while NCGA and NSGA-II are closing in on the front. After 5,000 evaluations (c), all MO-SHERPA runs are essentially on the front, while all NCGA runs are near the front but not on it. Four of the five NSGA-II runs have many points near the front, but many NSGA-II points also extend beyond the limits of the segments of the true front. After 10,000 evaluations (d), NCGA is very near the front for all runs. NSGA-II has many points on the front, but still has many points extending beyond the vertical limits of the rightmost two segments (those points are actually dominated.

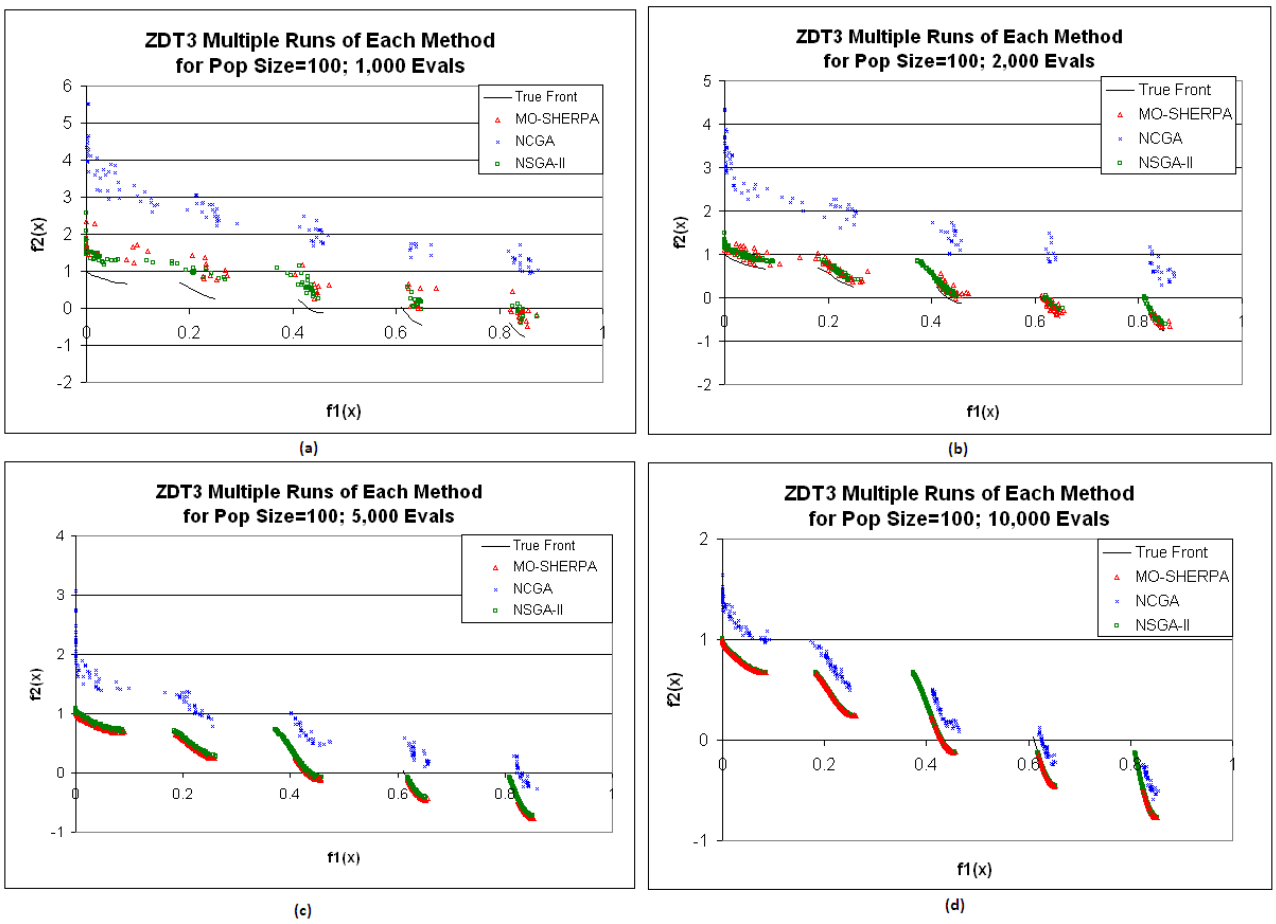

Figure 10 (a) – (d). Superimposed results of 5 runs each of MO-SHERPA, NCGA, and NSGA-II, on benchmark ZDT3 with population size 100. In (a), MO-SHERPA and NSGA-II are closing in on the true Pareto front within 1,000 evaluations while NCGA is far away. After 2,000 evaluations (b), both MO-SHERPA and NSGA-II have nearly converged upon the Pareto front, while NCGA makes some progress. After 5,000 evaluations (c), all MO-SHERPA runs are essentially on the front. NSGA-II runs have points on the front, but also many beyond the true extent of the front's segments. After 10,000 evaluations (d), NCGA still has not converged upon the front for any of its runs. NSGA continues to plot points beyond the segments, because in some runs, the points dominating them on the true Pareto front (in the segment to the immediate left) have not been found.

3.4 ZDT4 Function

The ZDT4 function has 21 local Pareto-optimal fronts and therefore is highly multi-modal. The objective functions are:

$$f_1(x) = x_1$$

$$f_2(x) = g(x)\left[1 - \sqrt{x_1/g(x)}\ \right]$$

where g(x) is defined as:

$$g(x) = 1 + 10(n-1) + \sum_{i=2}^{n} \left[x_i^2 - 10\cos(4\pi x_i)\right]$$

In this ZDT4 function, ten design variables x_i were chosen (n=10). The design variable ranges are from -5 to 5 for the last nine design variables and 0 to 1 for x_1. The global Pareto-optimal front appears when g= 1.0.

For each run, up to 25,000 evaluations were performed. Results for this problem are presented in Figures 11 through 13 at four stages of the search process – after 7500, 12500, 20000 and 25000 evaluations.

On the ZDT4 problem, which is made difficult by the presence of many local Pareto fronts, MOSHERPA clearly outperforms both NCGA and NSGA-II. Between NCGA and NSGA-II, performance is variable – typically, some runs of one are better than some runs of the other, but neither ever approaches the performance of MO-SHERPA, regardless of population size or length of run (through 25,000 evaluations).

4572 South Hagadorn Rd., Suite 3A ▲ East Lansing, MI 48823 ▲ TEL 517.664.1137 ▲ FAX 517.664.1175 ▲ www.redcedartech.com

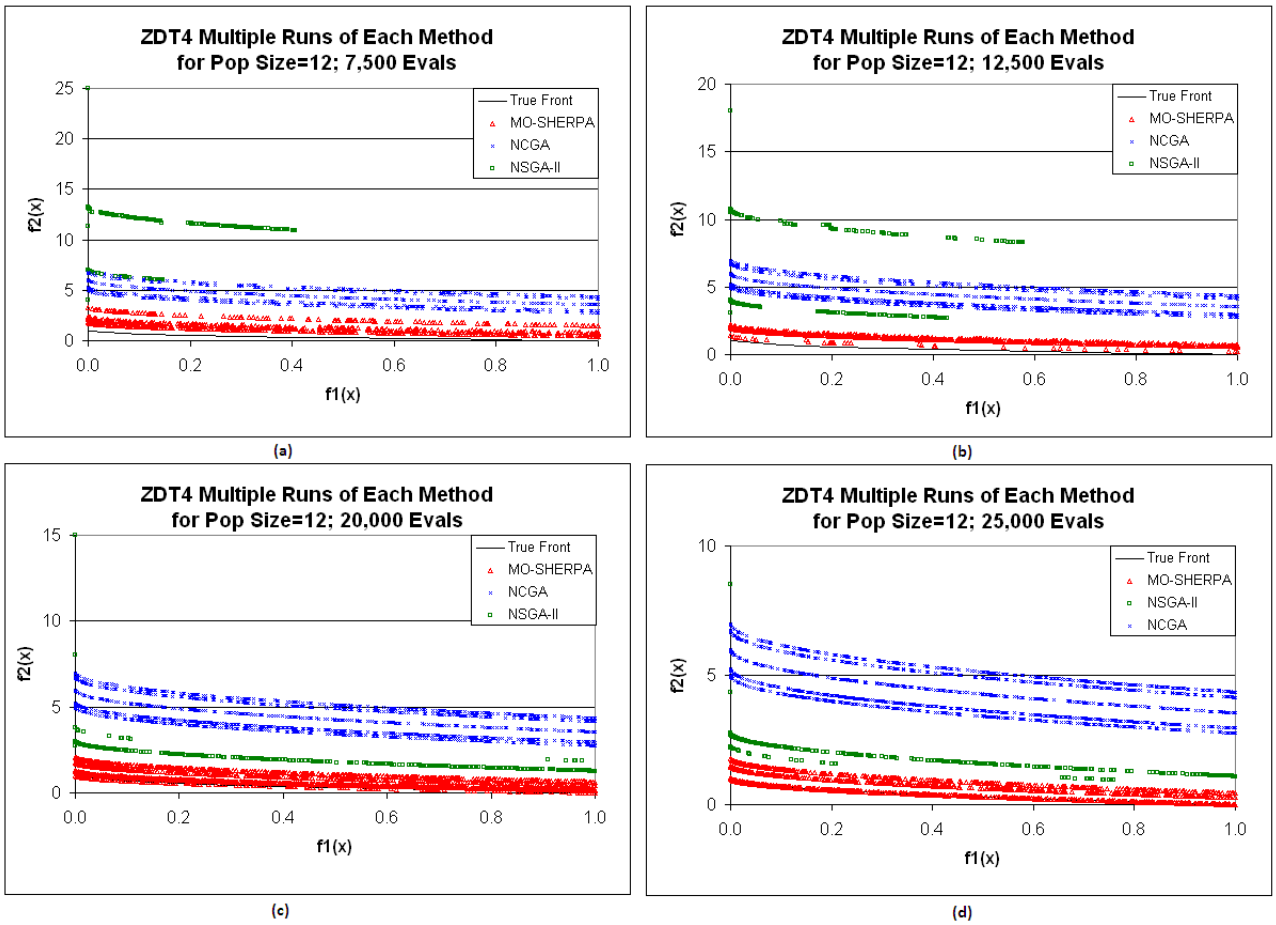

Figure 11 (a) – (d). Superimposed results of 5 runs each of MO-SHERPA, NCGA, and NSGA-II, on benchmark ZDT4 with population size 12. In (a), MO-SHERPA approaches the true Pareto front within 7,500 evaluations for three of the runs, while NCGA is further away and NSGA-II sometimes even further. After 10,000 evaluations (b), MO-SHERPA is close to the true front for one run, but the others are a small distance away. NCGA and NSGA-II are still far from the front. After 20,000 evaluations (c), MO-SHERPA is on the front for three of the five runs, while NCGA is far from the front. NSGA-II only has one fully formed front, and is far from the true front. After 25,000 evaluations (d), NCGA has made very little progress and NSGA-II has made some progress, but still only has two runs where fronts appear, and both are still far from the true front.

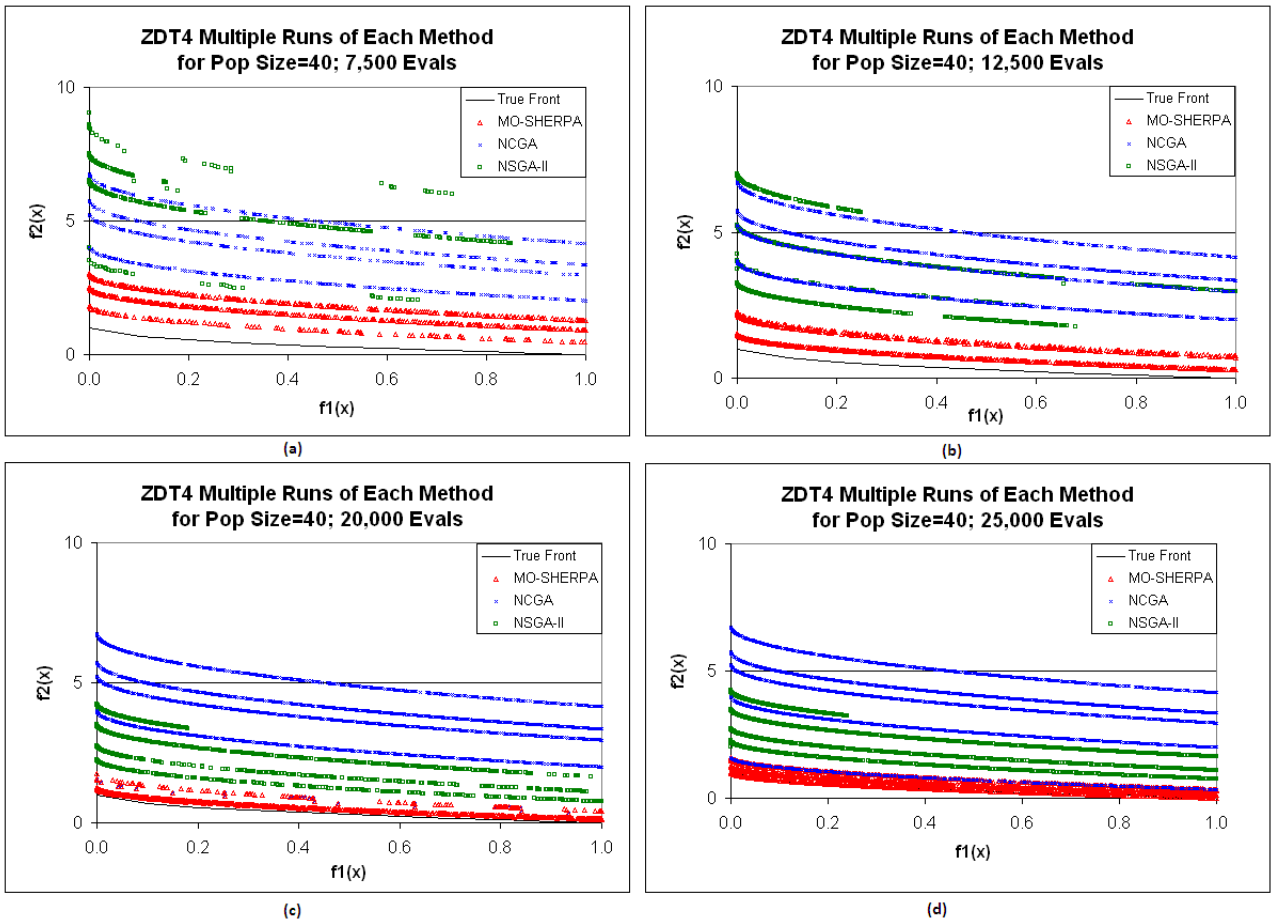

Figure 12 (a) – (d). Superimposed results of 5 runs each of MO-SHERPA, NCGA, and NSGA-II, on benchmark ZDT4 with population size 40. In (a), MO-SHERPA is approaching the front, but has not yet reached it, after 7,500 evaluations. NSGA-II and NCGA trail behind, with some runs very far from the true front. After 12,500 evaluations (b), all algorithms make progress toward the true front, with NCGA making the smallest amount of progress. After 20,000 evaluations (c), MO-SHERPA is very close to the front for three of the runs, with both the other algorithms still lagging behind. After 25,000 evaluations (d), MO-SHERPA is converged on the true front for three of the five runs. The closest competitor is NCGA, whose best run is very close to the worst for MO-SHERPA.

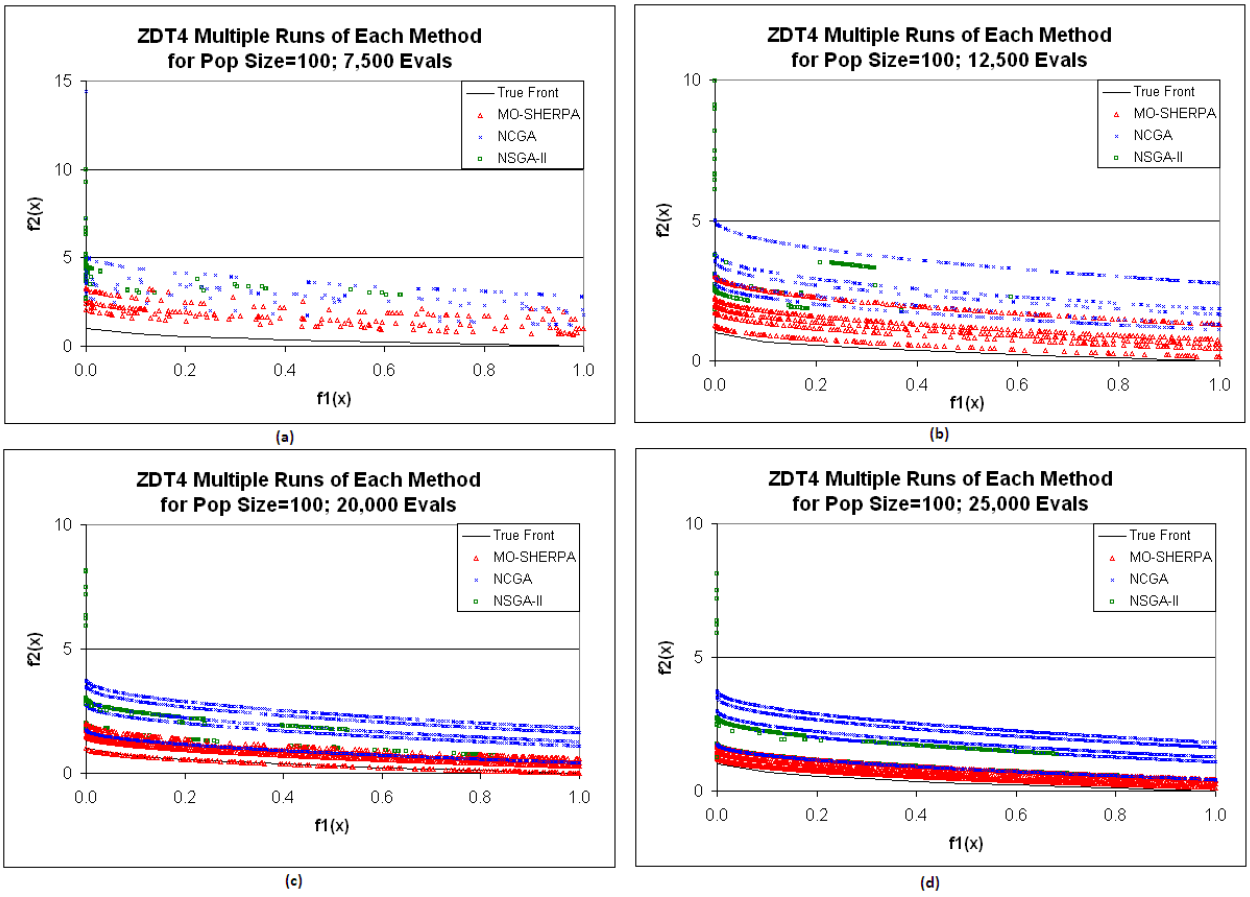

Figure 13 (a) – (d). Superimposed results of 5 runs each of MO-SHERPA, NCGA, and NSGA-II, on benchmark ZDT4 with population size 100. In (a), none of the algorithms are on the front, though MO-SHERPA is leading the pack. Several runs for NSGA-II have only a few points, all close to zero for f1(x). After 12,500 evaluations (b), MO-SHERPA is approaching close to the true front with one run, and is competitive with the others. NSGA-II does not have good fronts developed at this point. After 20,000 evaluations (c), MO-SHERPA is converged on the true front for one run, and the others are not far away, relative to all of the NCGA and NSGA-II runs. NCGA has one run that is better than the worst from MO-SHERPA, and NSGA-II has no points near the front. After 25,000 evaluations (d), all runs are close to the front for MO-SHERPA, while NCGA is only able to get one front within the same range as MO-SHERPA. NSGA-II is able to find one point near the front, but the points in all other runs are far from the true front.

3.5 ZDT6 Function

The ZDT6 function has a non-uniform search space: the Pareto-optimal solutions are non-uniformly distributed along the global Pareto front, and also the density of the solutions is lowest near the Pareto optimal front and highest away from the front. The objective functions are defined as:

$$f_1(x) = 1 - e^{(-4x_1)} sin^6(6\pi x_1)$$

$$f_2(x) = g(x)[1 - \left(\frac{f_1(x)}{g(x)}\right)^2]$$

where g(x) is defined as:

$$g(x) = 1 + 9\left[\frac{\sum_{i=2}^{n} x_i}{(n-1)}\right]^{\frac{1}{4}}$$

In this ZDT6 function, ten design variables x_i were chosen (n=10). The design variable ranges are from 0 to 1. The global Pareto-optimal front appears when g = 1.0.

For each run, up to 10,000 evaluations were performed. Results for this problem are presented in Figures 14 through 16 at four stages of the search process – after 500, 2000, 5000 and 10,000 evaluations.

On the ZDT6 benchmark, MO-SHERPA dramatically outperformed the NCGA and NSGA-II algorithms for all population sizes and all lengths of run (through 10,000 evaluations). NSGA-II was the next best on this problem, outperforming NCGA to an increasing extent as the run progressed. NCGA never reached the vicinity of the front in many of its runs, regardless of population size, within the 10,000-evaluation runs. At 5,000 evaluations, MO-SHERPA's results on all runs were still superior to NSGA-II's results on even its best run.

4572 South Hagadorn Rd., Suite 3A ▲ East Lansing, MI 48823 ▲ TEL 517.664.1137 ▲ FAX 517.664.1175 ▲ www.redcedartech.com

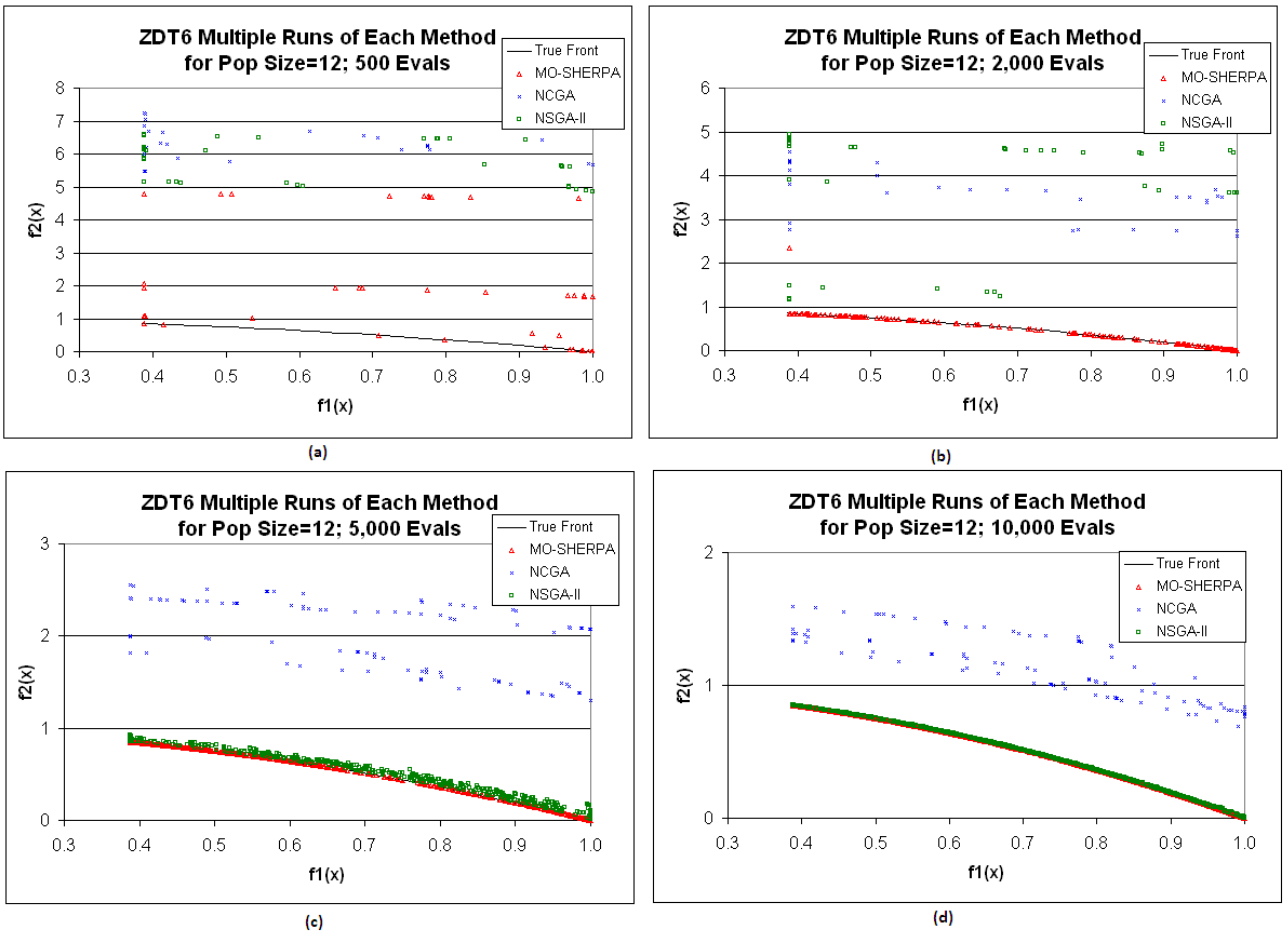

Figure 14 (a) – (d), Superimposed results of 5 runs each of MO-SHERPA, NCGA, and NSGA-II, on benchmark ZDT6 with population size 12. After 500 evaluations (a), MO-SHERPA points are close to or on the front on two of the five runs, while NCGA and NSGA-II points are both very far away from the front on all runs. After 2,000 evaluations (b), MO-SHERPA is converged on the front for all but one run, while NCGA and NSGA-II are generally still far from the front. After 5,000 evaluations (c), MO-SHERPA is converged on the true front. NSGA-II is very close to the front for all its runs, and NCGA is still far from the front. After 10,000 evaluations (d), NSGA-II has converged on the true front, while NCGA continues to make progress but is not near the true front.

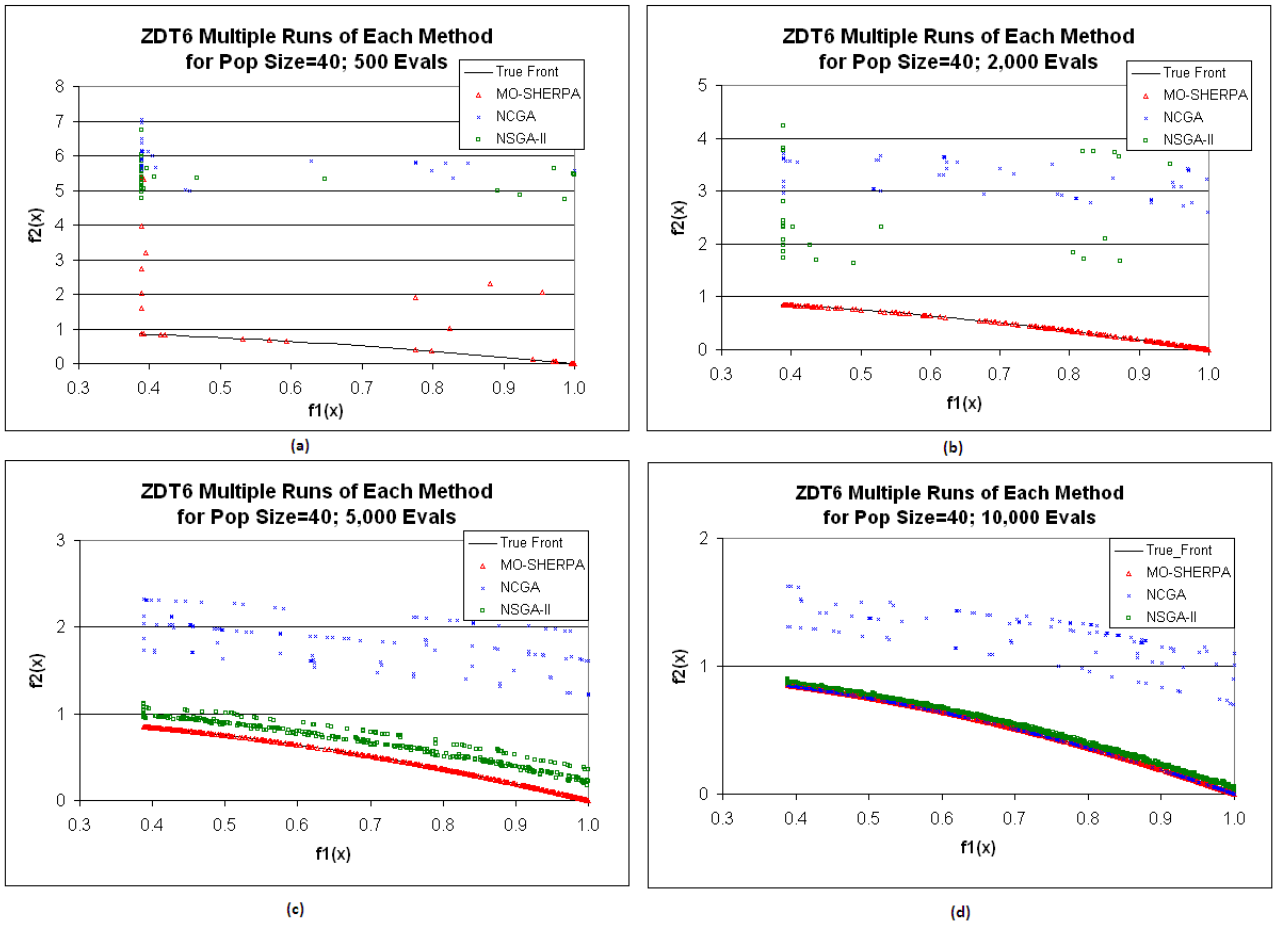

Figure 15 (a) – (d), Superimposed results of 5 runs each of MO-SHERPA, NCGA, and NSGA-II, on benchmark ZDT6 with population size 40. In (a), after 500 evaluations, MO-SHERPA is on the front for one run, and has better points than NCGA and NSGA-II on most of its other runs. After 2,000 evaluations (b), MO-SHERPA is converged on the true front for all of its runs. NCGA and NSGA-II are both far from the front. After 5,000 evaluations (c), NSGA-II is near the true front, while NCGA is far away. After 10,000 evaluations, NSGA-II is converged on or near the true front, while NCGA is still far away for most of its runs.

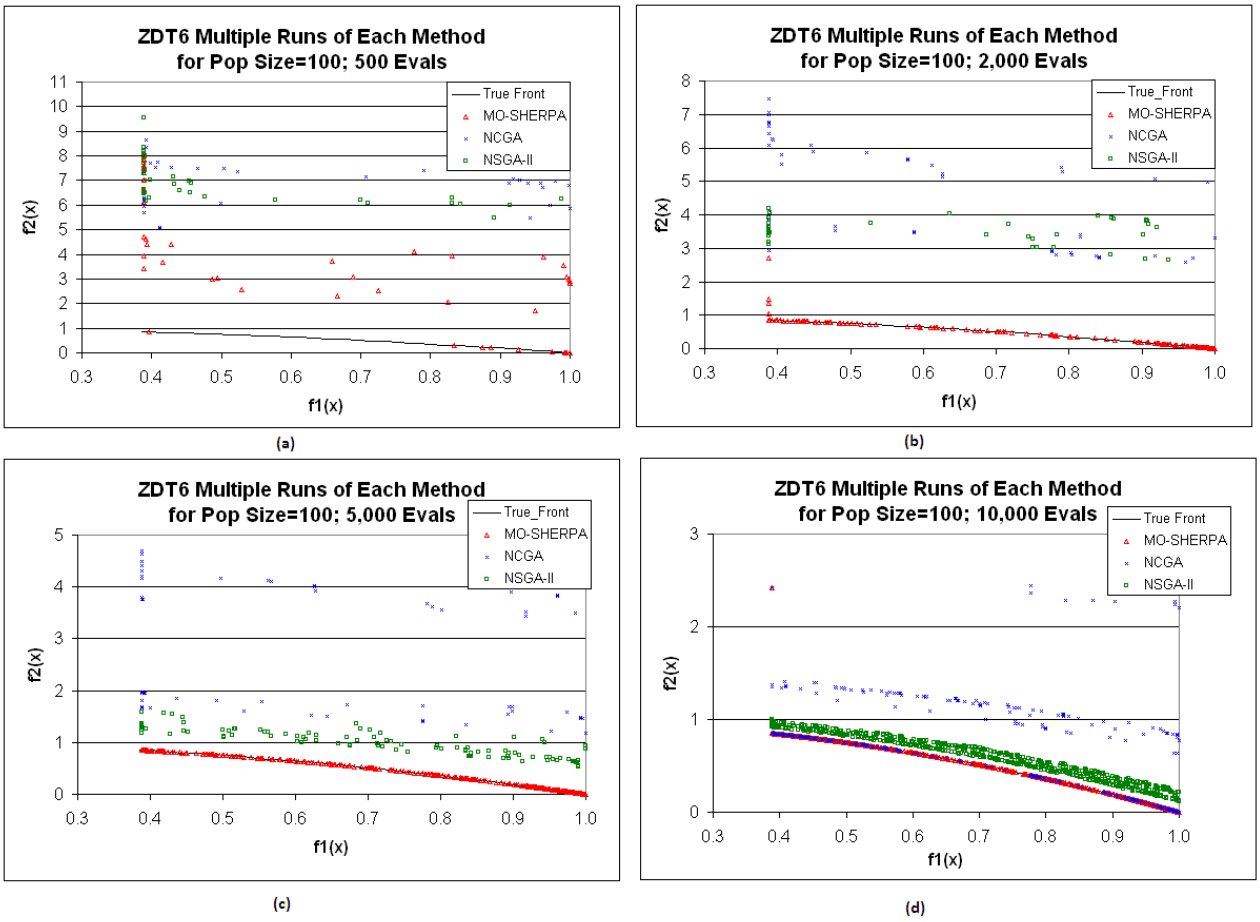

Figure 16 (a) – (d), Superimposed results of 5 runs each of MO-SHERPA, NCGA, and NSGA-II, on benchmark ZDT6 with population size 100. After 500 evaluations (a), MO-SHERPA has found some points on the true front in one run, and has found better points than NCGA or NSGA-II have found, in most of its other runs. After 2,000 evaluations (b), MO-SHERPA is converged on the front for most of its runs, but still has a few points on the left-hand side which will eventually be dominated (i.e., are not on the Pareto front). NCGA and NSGA-II are far away from the true front. After 5,000 evaluations (c), MO-SHERPA is fully converged on the true front for all runs, while NCGA and NSGA-II have still not reached the true front. After 10,000 evaluations (d), NSGA-II is approaching the front, but has not yet reached it, while NCGA remains far from the front in at least three of its five runs.

4572 South Hagadorn Rd., Suite 3A ▲ East Lansing, MI 48823 ▲ TEL 517.664.1137 ▲ FAX 517.664.1175 ▲ www.redcedartech.com

4. Summary and Conclusions

A benchmark study was conducted that compared the performance of three multi-objective optimization algorithms on a set of standard test problems, called the ZDT functions. All of the algorithms studied are direct methods and have some common characteristics, but other aspects of these methods are significantly different. On each of the test problems and for all population sizes considered, MOSHERPA dramatically outperformed both NSGA-II and NCGA in terms of efficiency and robustness.

The superior behavior of MO-SHERPA is attributed to its hybrid and adaptive formulation, which makes it effective over a wide range of problems.

5. References

1. Pratap, A., Deb, K., Agarwal, S. and Meyarivan, T., "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *KanGAL report 200001, Indian Institute of Technology*, Kanpur, India 2000.

2. Deb, K., *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, UK: Wiley, 2001.

3. Deb, K., Pratap, A., Agarwal, S.; Meyarivan, T., "A fast and elitist multi-objective genetic algorithm: NSGA-II,"*IEEE Transactions on Evolutionary Computation*, Volume 6, Issue 2, pp.182 –197, April, 2002.

4. Shinya Watanabe, Tomoyuki Hiroyasu, Mitsunori Miki, "NCGA: Neighborhood Cultivation Genetic Algorithm for Multi-Objective Optimization Problems," *GECCO Late Breaking Papers*, pp. 458-465, 2002.

5. *HEEDS Version 5.2 User's Manual*, 2008.

6. Zitzler, E., Deb, K., and Thiele, L., "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, Vol. 8, No. 2, pp. 173-195, 2000.

4572 South Hagadorn Rd., Suite 3A ▲ East Lansing, MI 48823 ▲ TEL 517.664.1137 ▲ FAX 517.664.1175 ▲ www.redcedartech.com