Locally Biased Variations of the DIRECT Algorithm

Hong Zhu and D. B. Bogy

Computer Mechanics Laboratory
Department of Mechanical Engineering
University of California
Berkeley, CA 94720

ABSTRACT

In this report, we investigate three locally biased forms of the DIRECT algorithm. The first form, referred to as DIRECT-I, was developed by Gablonsky and Kelley, and uses fewer groups than does DIRECT. The other two variations are proposed here, one of which uses a double partition for the hyper-rectangle containing the best sample point and is referred to as DIRECT-II; the other, referred to as DIRECT-III, combines the features of DIRECT-I and DIRECT-II. We tested all three variations and compared the results with those obtained by using the standard DIRECT algorithm. Results show that for some cases, locally biased variations obtain the global minimum point quicker. We also tested the locally biased variations of the DIRECT algorithm in the context of a slider air bearing surface (ABS) optimization problem. It is found that the three locally biased variations of the DIRECT algorithm generally have higher convergence rates than does the standard DIRECT algorithm. The variations perform especially well in some situations and they may dramatically reduce the time needed to find the global minimum points.

1. INTRODUCTION

Optimization is the process of minimizing a function subject to conditions on the variables. This function is generally called the objective function or cost function. The conditions set on the variables are referred to as constraints.

We can state the optimization problem as:
Minimize $\{f(\boldsymbol{x}) \mid \boldsymbol{x} \in S\}$, where $f(\boldsymbol{x})$ is the objective function, and S is the search space.

In this report, we consider the bounded constrained optimization problem

Minimize $\{f(\boldsymbol{x}) \mid \boldsymbol{x} \in [\boldsymbol{u},\boldsymbol{v}]\}$, where $\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}$ are n-dimensional vectors.

There are many global optimization algorithms, and they can be divided into two fundamentally different categories, i.e. deterministic algorithms and stochastic algorithms. For the deterministic algorithms, every new search point is chosen in a definite way so no random processes are involved. For the stochastic algorithms, random elements are introduced to generate the new search points.

The DIRECT algorithm is a global deterministic algorithm developed by Jones et al. in 1993 [4] . The DIRECT algorithm is guaranteed to converge [4] and it has a very fast convergence rate. Thus it can find the global minimum very quickly compared with other algorithms [4][5] .

We have presented the details of the DIRECT algorithm and the results of numerical experiments as well as its application to the slider Air Bearing Surface (ABS) optimization in a CML technical report [1].

In this report, we report on three locally biased variations of the standard DIRECT algorithm. We first introduce the standard DIRECT algorithm and then we present the three variations. Subsequently, we discuss our results from experimentation using combinations of all forms of these variations. Finally we present results for a test case of slider ABS optimization.

2. NUMERICAL METHOD

2.1 Standard DIRECT algorithm

DIRECT is an acronym for **DI**viding **RECT**angles, a key step in the algorithm. It is a global deterministic algorithm based on the classical one-dimensional Lipschitzian optimization algorithm known as the Shubert algorithm. It is a multi-dimensional Lipschitzian optimization method which can be used without knowing the Lipschitz constant. DIRECT is designed to solve problems subjected to bounded constraints.

Without loss of generality, in the DIRECT algorithm we always assume that every variable has a lower bound of 0 and an upper bound of 1, since we can always normalize the variables to this interval. Thus, the search space is an n-dimensional unit hyper-cube. There are two main components in the DIRECT algorithm: one is the dividing strategy for the hyper-cubes and the hyper-rectangles (they are referred to as "boxes" in our reports); the other is the selection of the potentially optimal boxes. We briefly introduce them in sections 2.1.1 and 2.1.2. For more details, please refer to Ref. [1].

2.1.1 Dividing strategy

The dividing strategy of the DIRECT algorithm for the hyper-cubes and the hyper-rectangles is as follows:

A. Partition of a hyper-cube
Assume m is the center point a hyper-cube. We sample the points $m \pm \delta e_i$, where δ equals 1/3 of the side length of the cube and e_i is the i-th Euclidean base-vector. We define $s_i = \min \{ f(m-\delta e_i), f(m+\delta e_i) \}$, then the partition will be in the order given by s_i, starting with the lowest s_i. This means the hyper-cube is first partitioned along the direction with the lowest s_i, then the remaining field is partitioned along the direction of the second lowest s_i, and so on until the hyper-cube is partitioned in all directions.

B. Partition of a hyper-rectangle
Hyper-rectangles are only partitioned along their longest sides. This partition strategy ensures a reduction in the maximal side length of a hyper-rectangle.

2.1.2 Selection of potentially optimal boxes

Let m_i denote the center point of the i-th hyper-rectangle, and d_i the distance from the center point to the vertices. Then the potentially optimal boxes are defined as follows:

Definition 2.1 *Let $\varepsilon > 0$ be a positive constant and f_{min} be the current lowest function value. A hyper-rectangle (box) j is said to be potentially optimal if there exists some rate-of-change constant $\widetilde{K} > 0$ such that*

$$f(m_j) - \widetilde{K} d_j \le f(m_i) - \widetilde{K} d_i \quad \text{for any } i \tag{2.1}$$

$$f(m_j) - \widetilde{K} d_j \le f_{min} - \varepsilon |f_{min}| \tag{2.2}$$

2.2 Locally biased variations of the standard DIRECT algorithm

2.2.1 DIRECT algorithm with fewer groups

The first variation we discuss was developed by Gablonsky and Kelley[6], and uses fewer groups. In this report, we refer to it as DIRECT-I.

The only difference between the standard DIRECT algorithm and this variation is the definition of the measure of the groups. For the standard DIRECT algorithm, the group measure is defined as the distance from the center point of a box to its vertices, which is illustrated in Fig. 1. For DIRECT-I, the group measure is defined as the length of the longest side of a box, which is illustrated in Fig. 2.

From Figs. 1 and 2 it is clear that by changing the definition of the group measure, DIRECT-I has relatively fewer groups than the standard DIRECT does. Since only the point with the lowest value in a group is eligible to be potentially optimal, DIRECT-I will bias the search toward the local minima.

2.2.2 DIRECT algorithm with double partitions

We developed a second locally biased variation of the standard DIRECT algorithm, referred to here as DIRECT-II. The purpose of this variation is to partition the box containing the point of the lowest function value twice

during each iteration. By doing this, we double the weight on the search around the point with the lowest function value. Thus, the algorithm searches more intensively around the point with the lowest function value.

2.2.3 DIRECT algorithm with both features

Based on the above two variations, we propose a third variation which combines their features, i.e. fewer groups and double partitions. This third variation is referred as DIRECT-III.

Since the DIRECT-III algorithm combines the two locally biased features, it is expected that its search will be heavily biased toward the local minima.

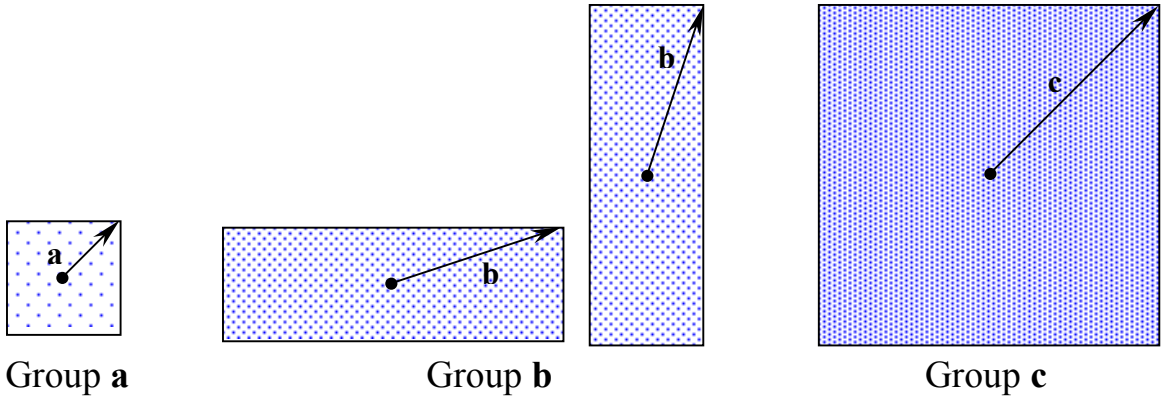

Fig. 1 Example of different groups in standard DIRECT

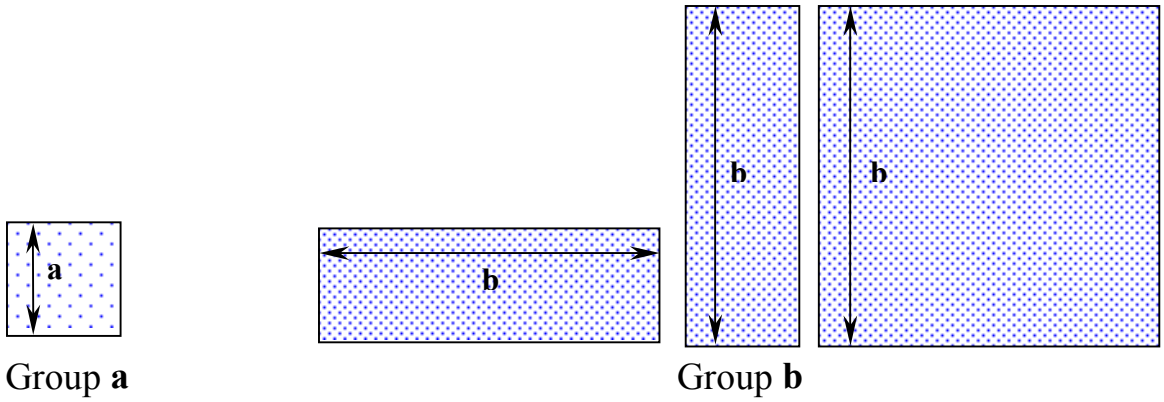

Fig. 2 Example of different groups in DIRECT-I

2.3 Process demonstration of the algorithms

Here we use a 2-D example to demonstrate the process of the DIRECT algorithm and its three locally biased variations. The function used is:

$$F(x_1, x_2) = 10\,(\,|x_1 - 0.4|^{1/2}\,) + 50\,(\,|x_2 - 0.2|^{3/2}\,) \quad \text{where } x_1, x_2 \in [0,1]$$

Figures 3A ~ 3F, Figs. 4A ~ 4F, Figs. 5A ~ 5F and Figs. 6A ~ 6F show the first 5 iterations for the standard DIRECT algorithm, DIRECT-I, DIRECT-II and DIRECT-III, respectively.

For those pictures on the left-hand side, the x-axis stands for variable x_1 and the y-axis stands for variable x_2, so the unit square is the search space. The shadowed areas are the potentially optimal boxes (can be squares or rectangles) just partitioned. The dots represent the center points of the boxes. The circular dot shows the sample point with the lowest function value. The numbers under those dots are the function values at those center points.

For those pictures on the right-hand side, the horizontal axis stands for the group measure and the vertical axis stands for the function value. The circular points represent the center points of the non-optimal boxes. The solid round points represent the center points of the potentially optimal boxes. The lines connecting these potentially optimal points form the convex hull of all the data points.

Figure 3A shows the initial state of the standard DIRECT algorithm. In this state only one central point is evaluated and it is designated as the potentially optimal point. The box located in this way is partitioned as shown in Fig. 3B.

$s_1 = \min \{13, 14.8\} = 13$
$s_2 = \min \{3.47, 28.4\} = 3.47$

The x_2 direction (y) gets partitioned first, followed by the x_1 direction (x). Because only one potentially optimal point is chosen in Fig. 3B, only one box containing that point is partitioned in Fig. 3C. The rectangle is only partitioned along its longest side.

For the DIRECT-I algorithm, which defines the longest side length of a box as its group measure, fewer groups will be used. For the standard

DIRECT algorithm, the numbers of groups from the initial state to the iteration five are: 1, 2, 2, 3, 3, 5; for DIRECT-I, the numbers are: 1, 2, 2, 2, 2, 3.

Since DIRECT-I uses fewer groups, it also has fewer potentially optimal points at each iteration and the search will be more locally focused.

For the DIRECT-II algorithm, the box containing the lowest function value is partitioned twice during each iteration. Because its definition of the group measure is the same as that of the standard DIRECT algorithm, and because more boxes will be partitioned during each iteration, the result should be more groups in each iteration compared with the standard DIRECT algorithm. From Figs. 5A ~ 5F, we see that the numbers of groups from the initial state to the fifth iteration for DIRECT-II are: 1, 2, 3, 5, 7, 8.

Because the DIRECT-III algorithm combines the above two locally biased measures, it has a strong locally orientated search strategy.

If we compare Figs. 3F, 4F, 5F and 6F, which are the results at iteration five for DIRECT, DIRECT-I, DIRECT-II and DIRECT-III respectively, it is clear that DIRECT-II and DIRECT-III generate more sample points around the local minimum points for a given number of iterations than DIRECT does. Since DIRECT-I generates fewer sample points in each iteration as compared with the standard DIRECT algorithm, it follows that, if the number of function evaluations is fixed, all three locally biased variations of the standard DIRECT algorithm will have more sample points around the local minima than would the standard DIRECT algorithm.

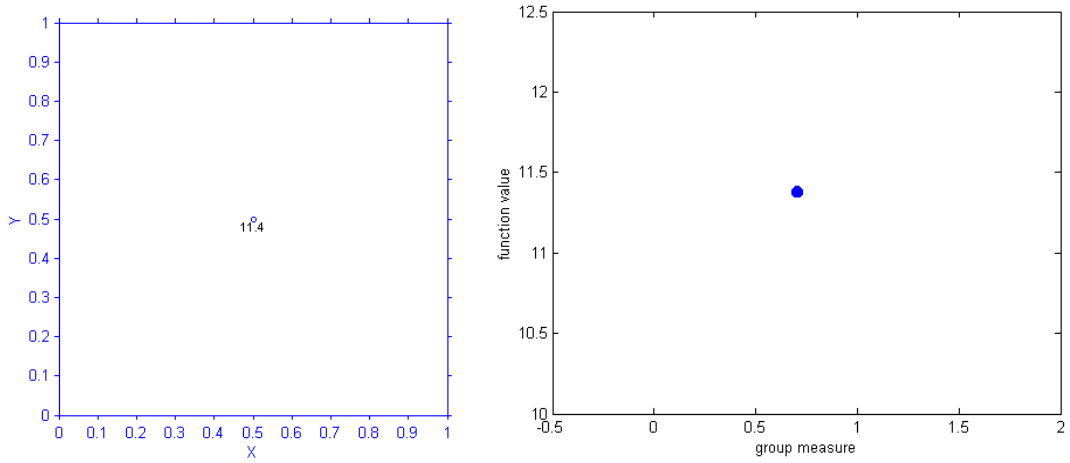

Fig. 3A DIRECT initial state

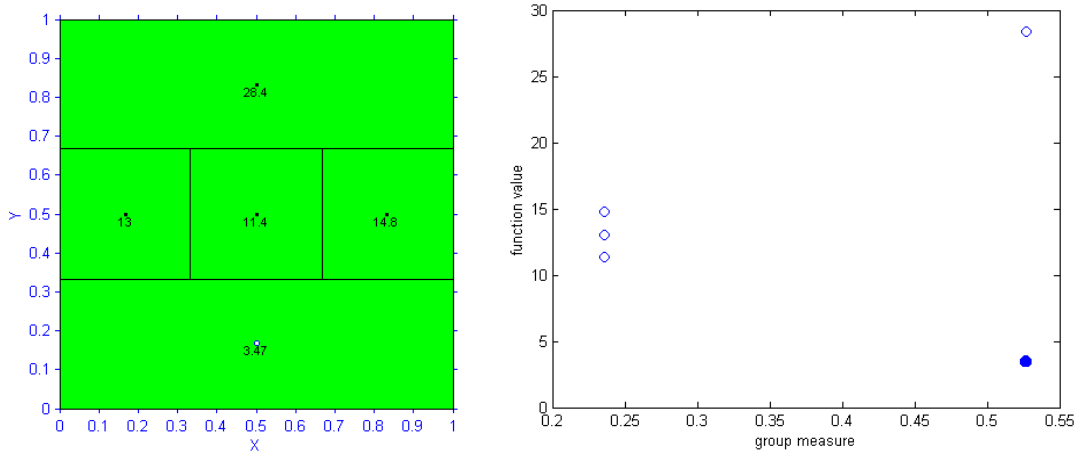

Fig. 3B DIRECT iteration 1

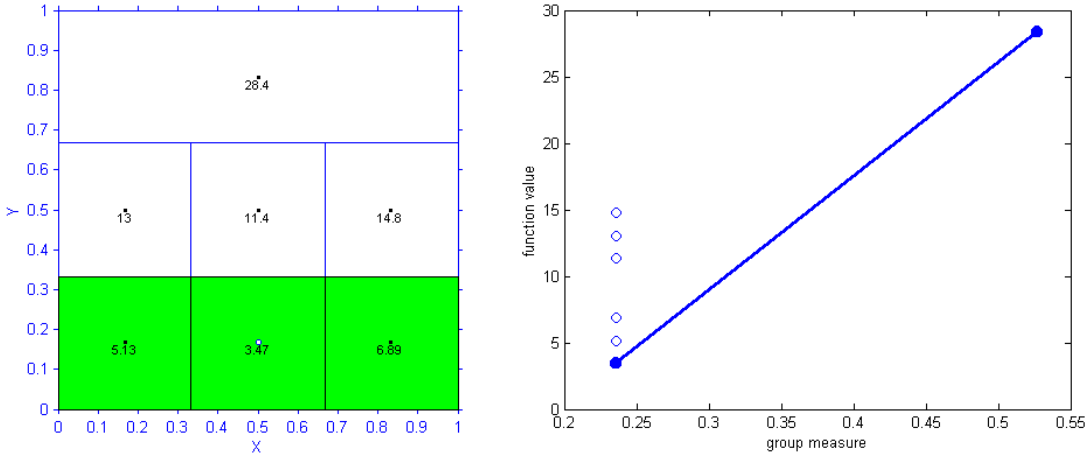

Fig. 3C DIRECT iteration 2

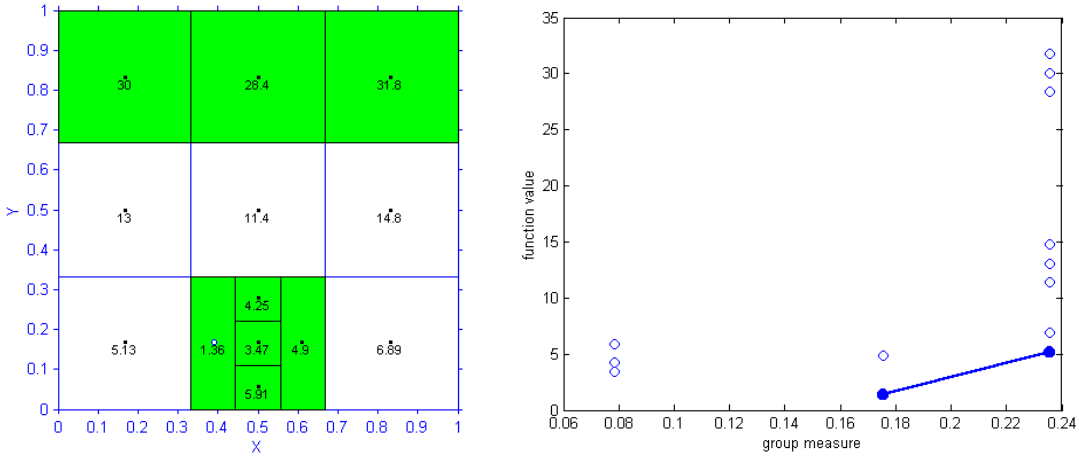

Fig. 3D DIRECT iteration 3

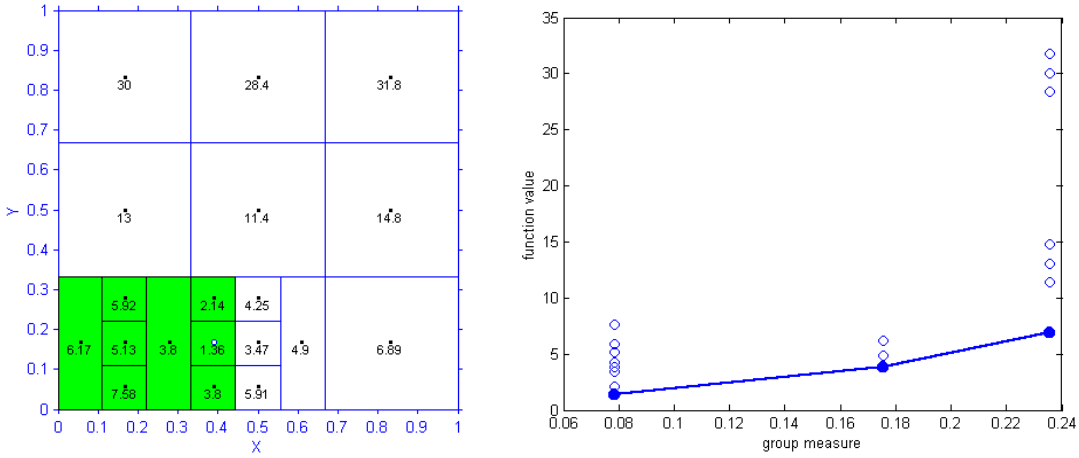

Fig. 3E DIRECT iteration 4

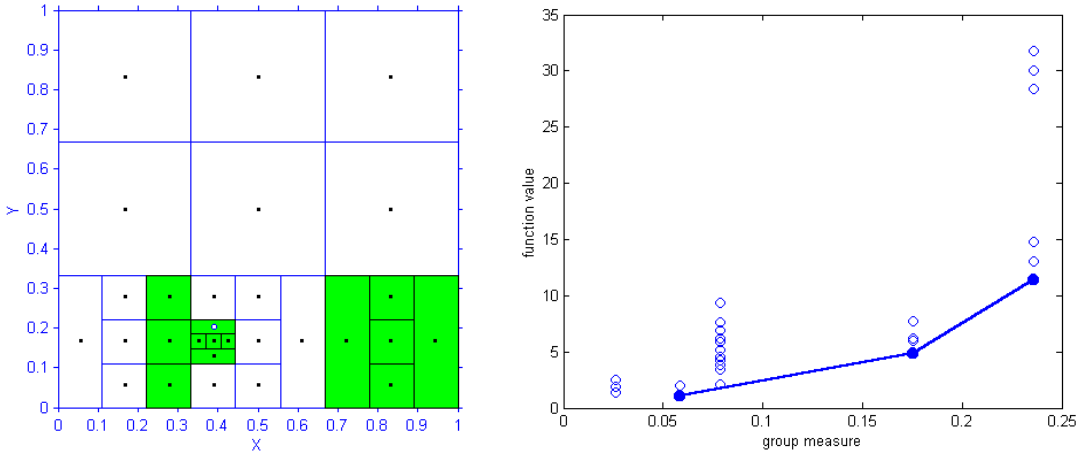

Fig. 3F DIRECT iteration 5

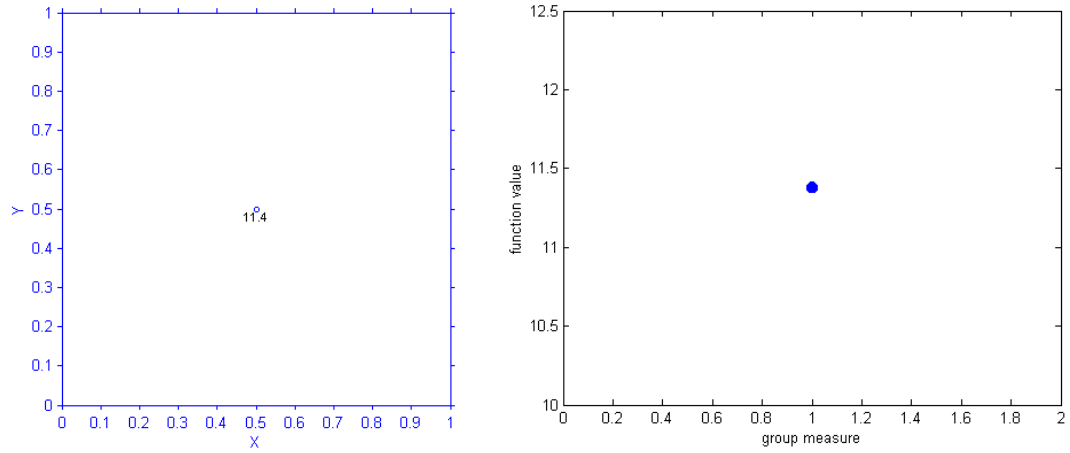

Fig. 4A DIRECT-I initial state

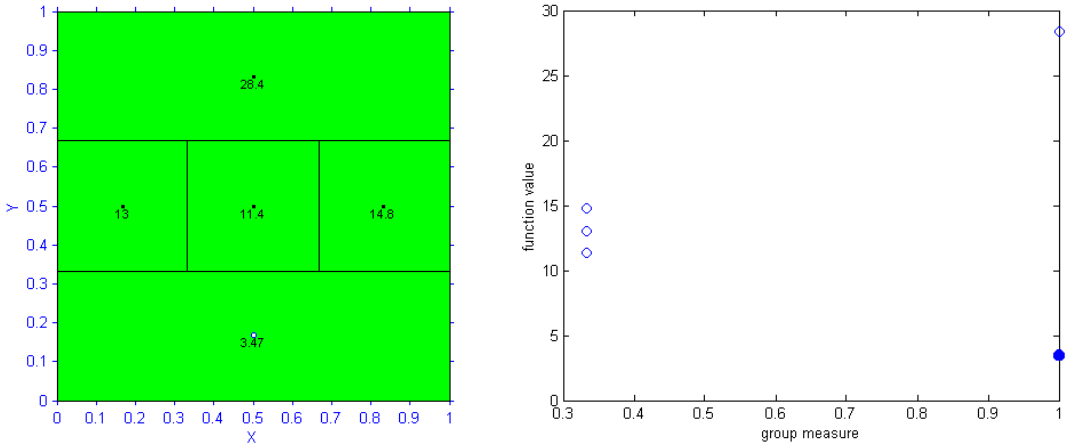

Fig. 4B DIRECT-I iteration 1

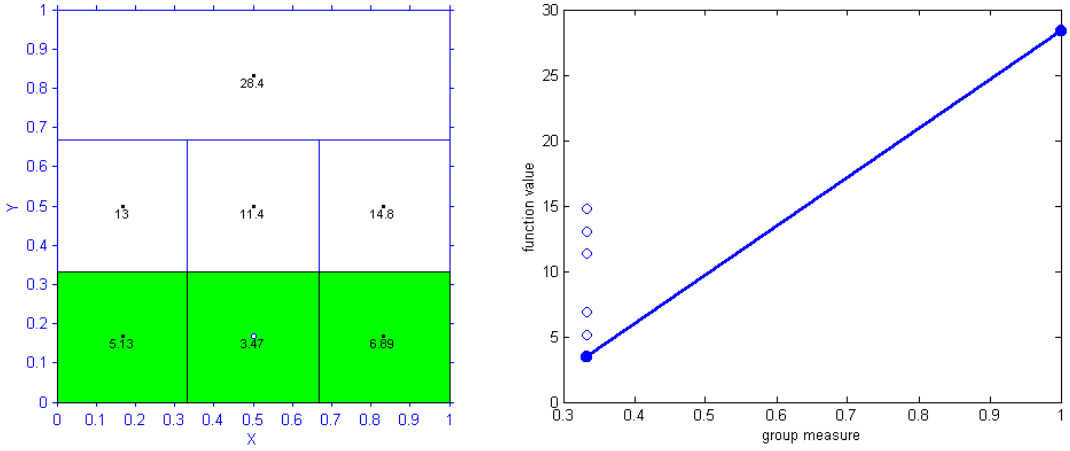

Fig. 4C DIRECT-I iteration 2

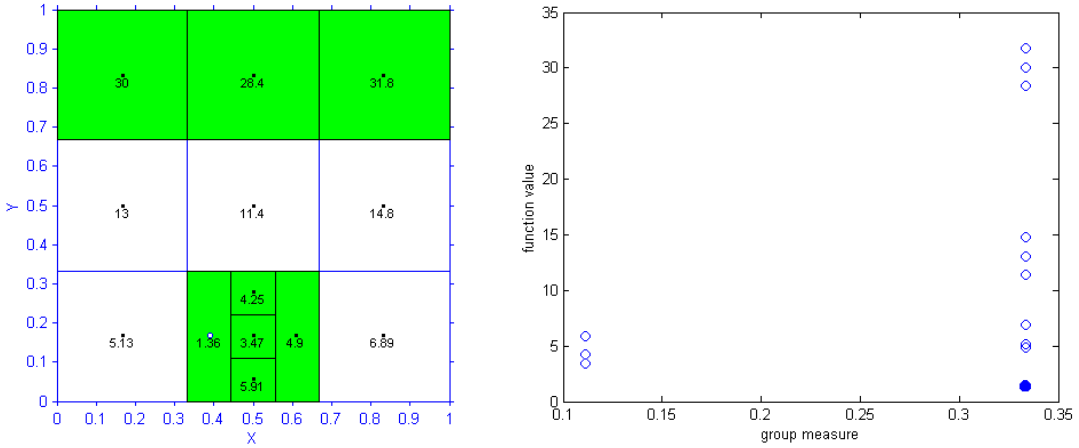

Fig. 4D DIRECT-I iteration 3

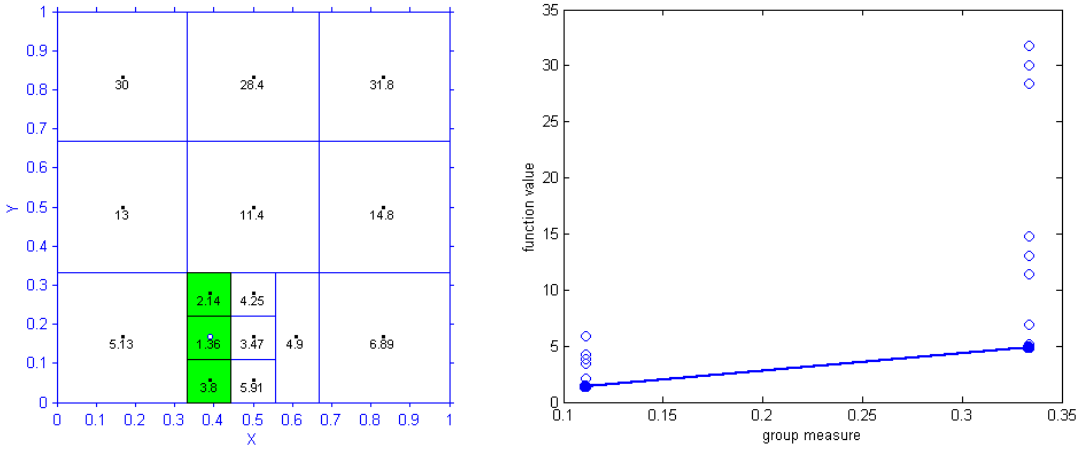

Fig. 4E DIRECT-I iteration 4

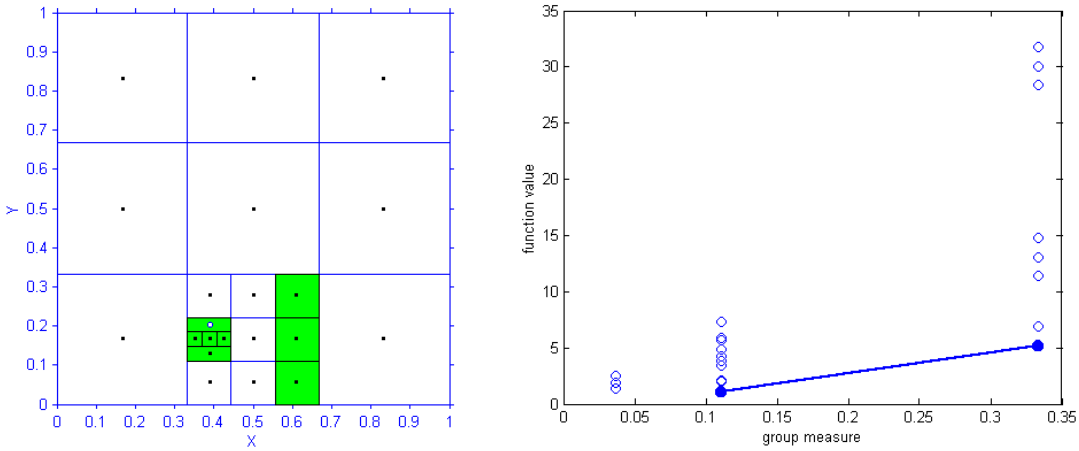

Fig. 4F DIRECT-I iteration 5

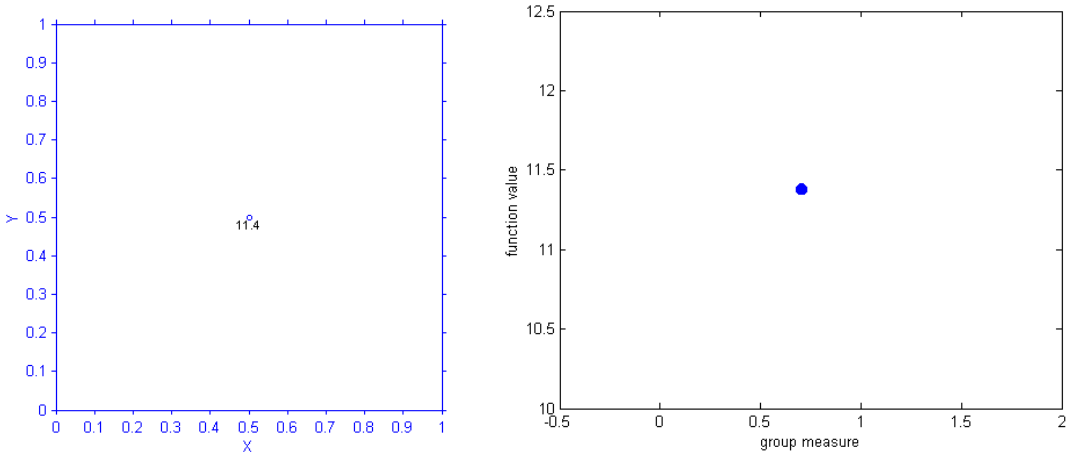

Fig. 5A DIRECT-II initial state

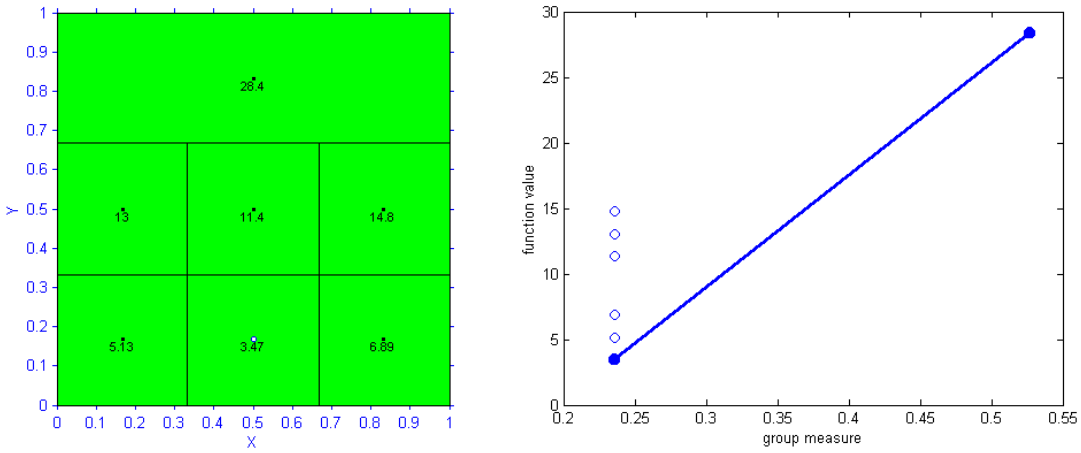

Fig. 5B DIRECT-II iteration 1

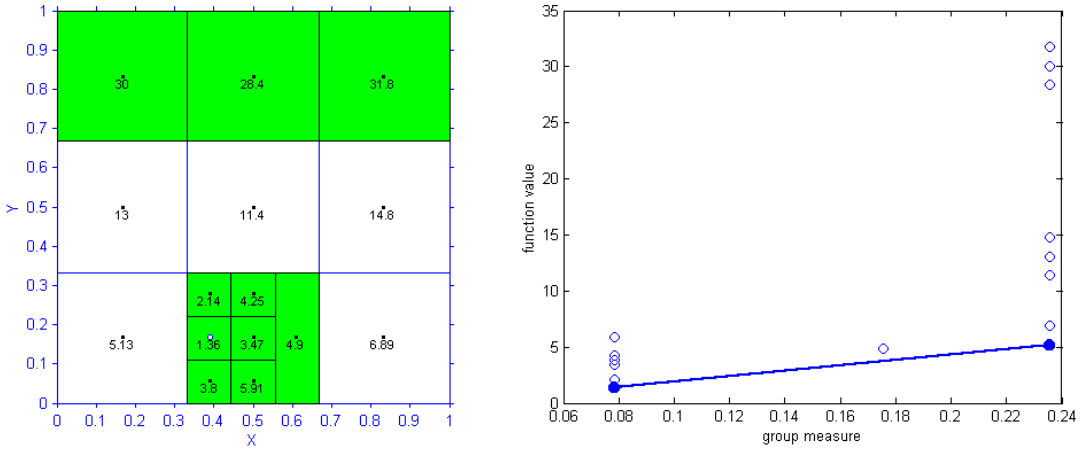

Fig. 5C DIRECT-II iteration 2

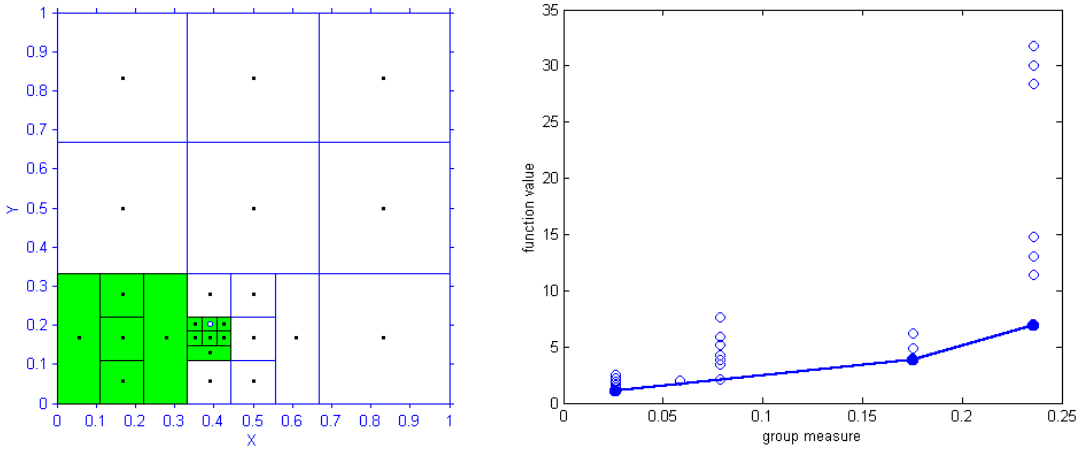

Fig. 5D DIRECT-II iteration 3

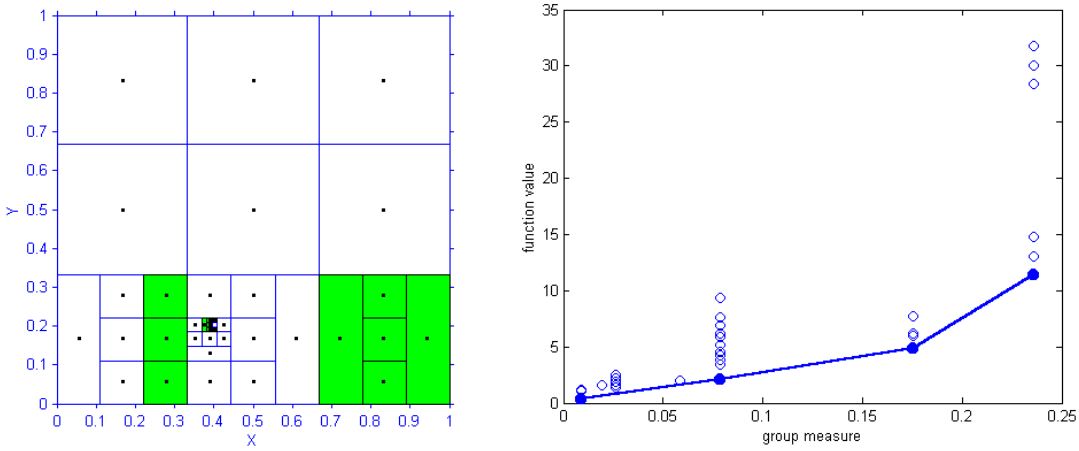

Fig. 5E DIRECT-II iteration 4

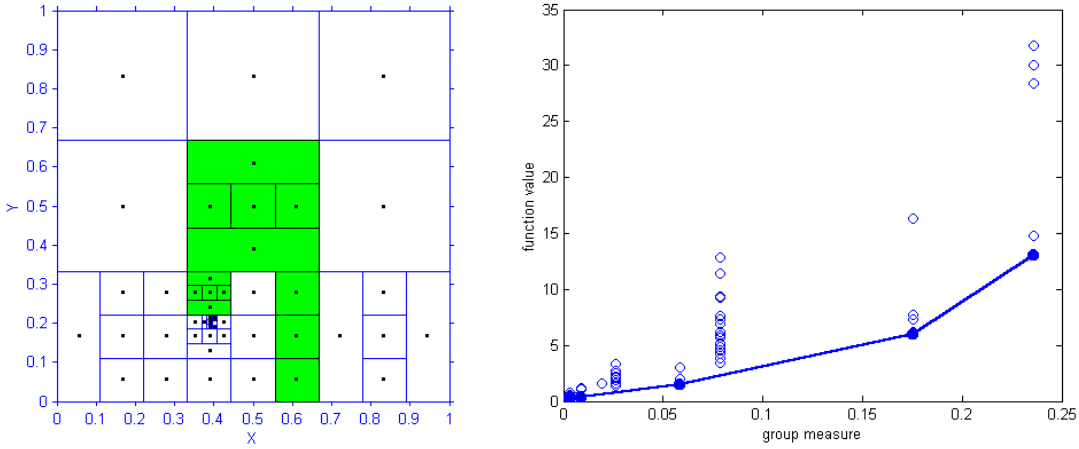

Fig. 5F DIRECT-II iteration 5

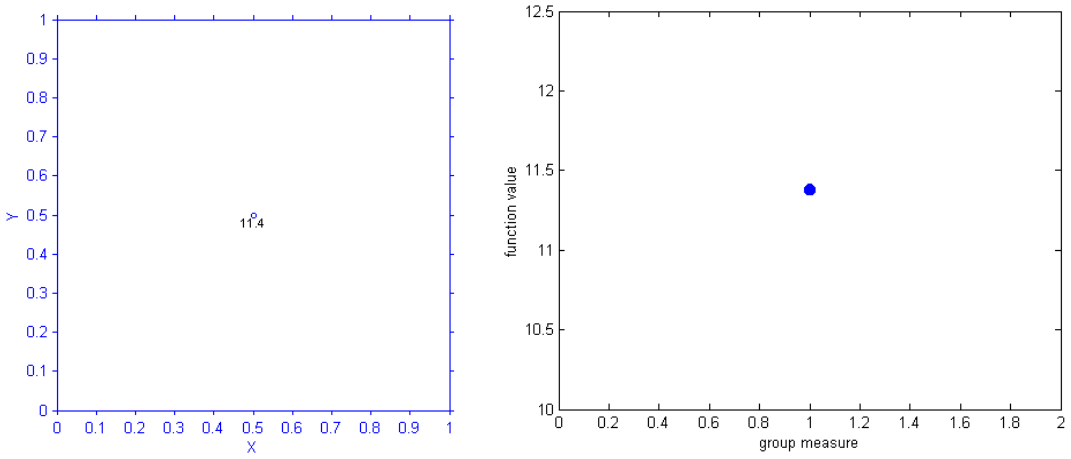

Fig. 6A DIRECT-III initial state

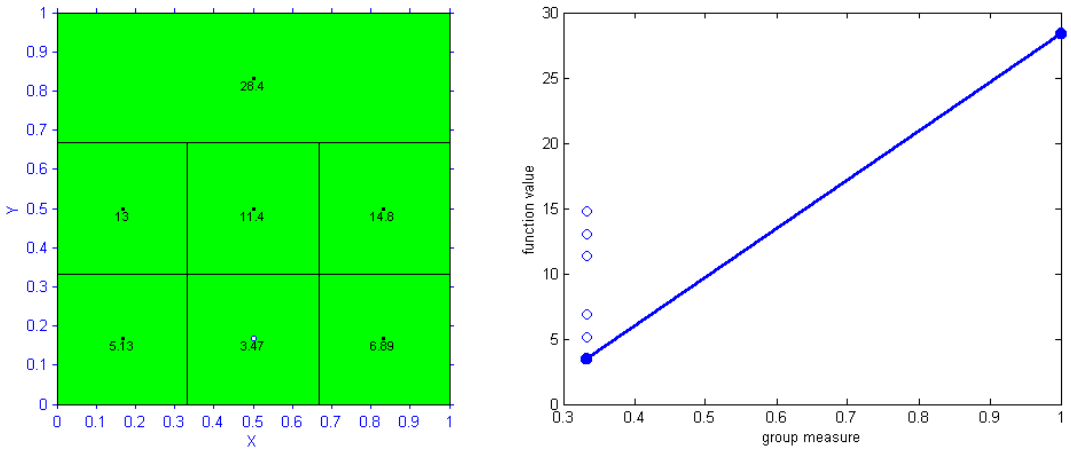

Fig. 6B DIRECT-III iteration 1

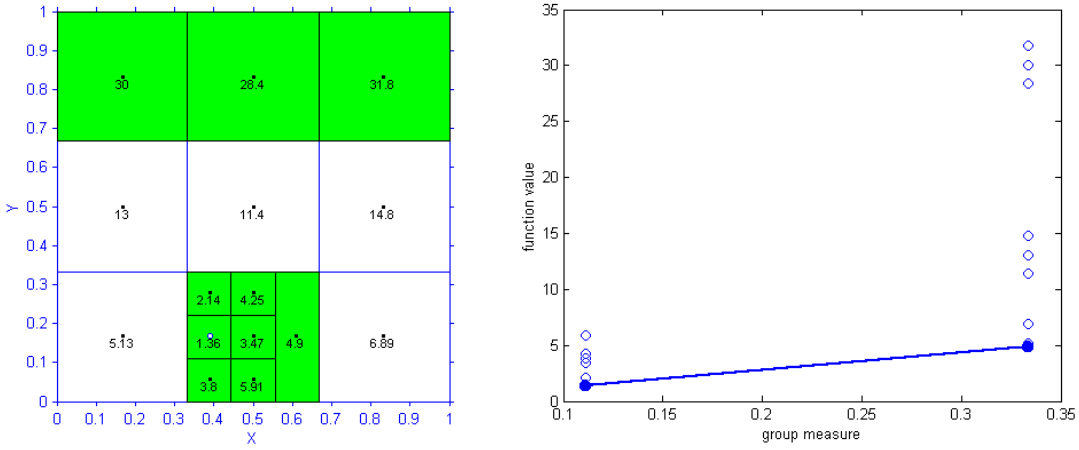

Fig. 6C DIRECT-III iteration 2

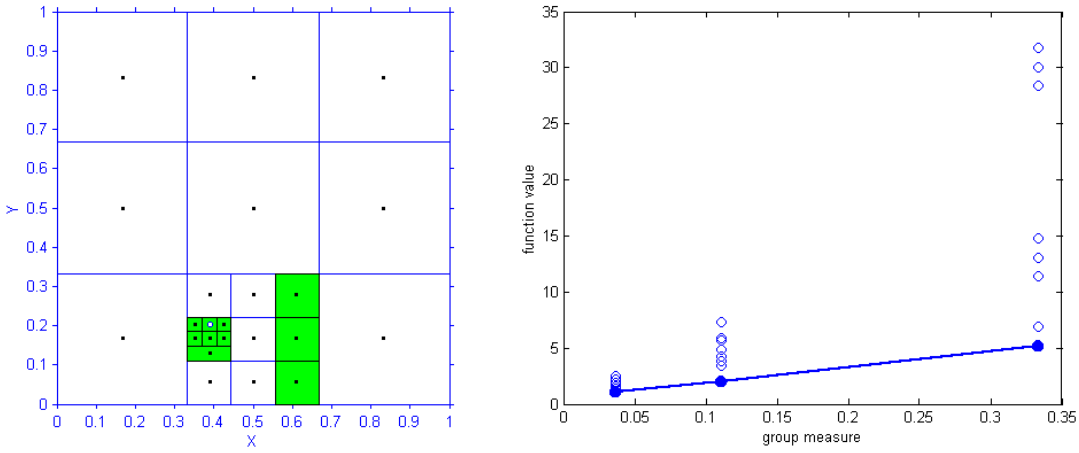

Fig. 6D DIRECT-III iteration 3

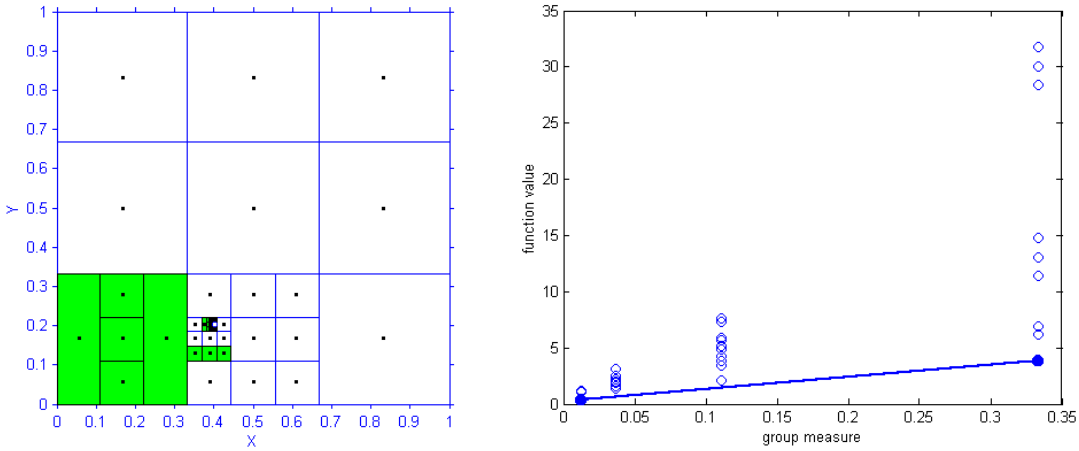

Fig. 6E DIRECT-III iteration 4

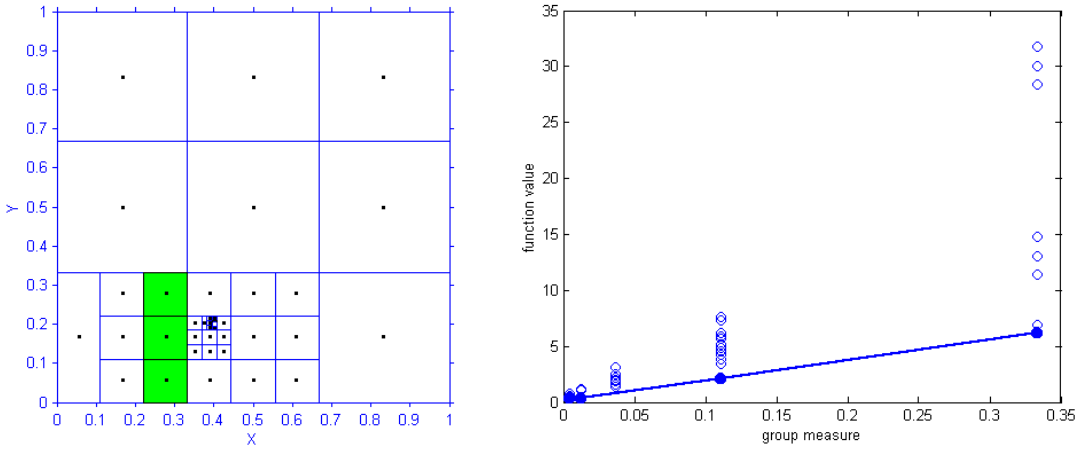

Fig. 6F DIRECT-III iteration 5

3. NUMERICAL EXPERIMENTS

3.1 Testing functions with one global and local minimum point

The testing functions used here include 2-D, 3-D, 5-D, 10-D and 20-D functions. These functions have only one global and local minimum point, and the minimum values of these functions are zero. They are defined as follows:

2-D: $F(x_1, x_2) = (x_1 - 0.4)^2 + (x_2 - 0.2)^2$.

3-D: $F(x_1, x_2, x_3) = (x_1 - 0.2)^2 + (x_2 - 0.3)^2 + (x_3 - 0.4)^2$.

5-D: $F(x_1, x_2, x_3, x_4, x_5) = (x_1 - 0.1)^2 + (x_2 - 0.3)^2 + (x_3 - 0.5)^2 + (x_4 - 0.7)^2 + (x_5 - 0.9)^2$.

10-D: $F(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}) = (x_1 - 0.1)^2 + (x_2 - 0.2)^2 + (x_3 - 0.3)^2 + (x_4 - 0.4)^2 + (x_5 - 0.5)^2 + (x_6 - 0.6)^2 + (x_7 - 0.7)^2 + (x_8 - 0.8)^2 + (x_9 - 0.9)^2 + (x_{10} - 1.0)^2$.

20-D: $F(x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8,x_9,x_{10},x_{11},x_{12},x_{13},x_{14},x_{15},x_{16},x_{17},x_{18},x_{19},x_{20}) = (x_1 - 0.05)^2 + (x_2 - 0.1)^2 + (x_3 - 0.15)^2 + (x_4 - 0.2)^2 + (x_5 - 0.25)^2 + (x_6 - 0.3)^2 + (x_7 - 0.35)^2 + (x_8 - 0.4)^2 + (x_9 - 0.45)^2 + (x_{10} - 0.5)^2 + (x_{11} - 0.55)^2 + (x_{12} - 0.6)^2 + (x_{13} - 0.65)^2 + (x_{14} - 0.7)^2 + (x_{15} - 0.75)^2 + (x_{16} - 0.8)^2 + (x_{17} - 0.85)^2 + (x_{18} - 0.9)^2 + (x_{19} - 0.95)^2 + (x_{20} - 1.0)^2$.

For all these cases, $x_i \in [0,1]$, $i = 1,...20$.

The results for the 2-D case are shown in Figs. 7 ~ 13.

Figure 7 shows the surface shape of the 2-D function. Figure 8 shows its contour lines.

Figures 9 ~ 12 show the optimization results obtained by using DIRECT, DIRECT-I, DIRECT-II and DIRECT-III respectively.

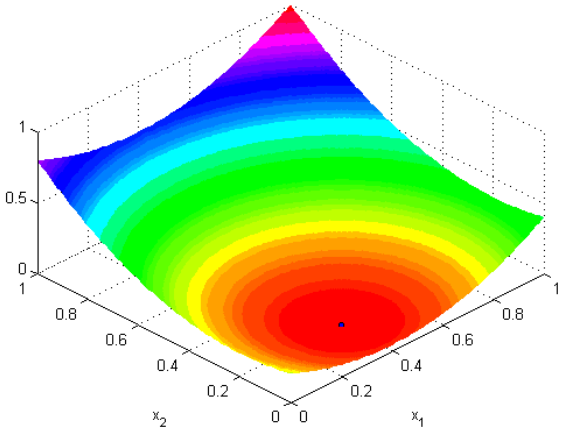

Fig. 7 Surface shape

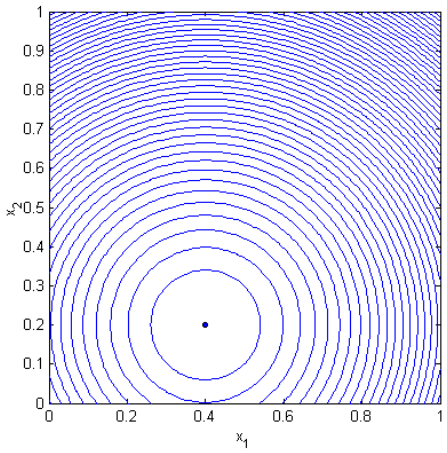

Fig. 8 Contour lines

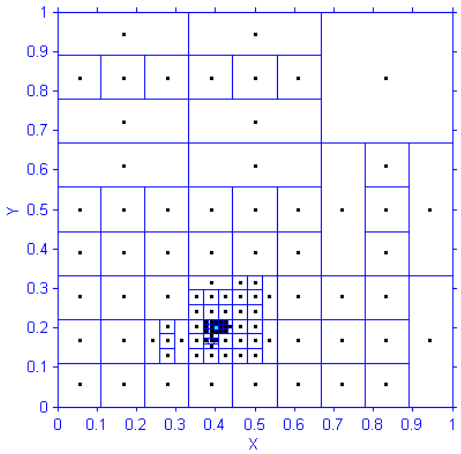

Fig. 9 Results of DIRECT

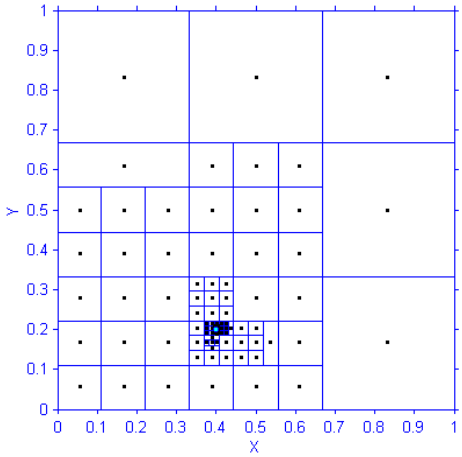

Fig. 10 Results of DIRECT-I

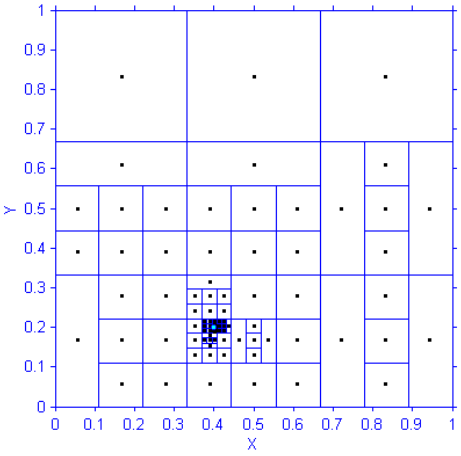

Fig. 11 Results of DIRECT-II

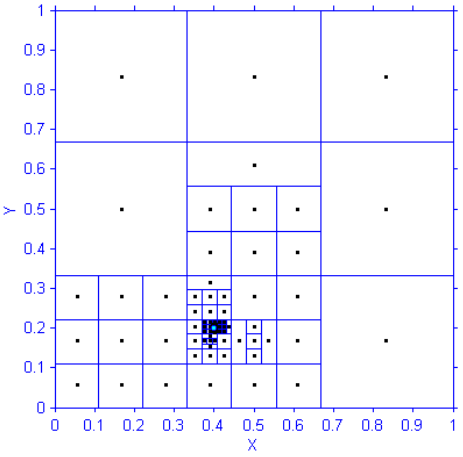

Fig. 12 Results of DIRECT-III

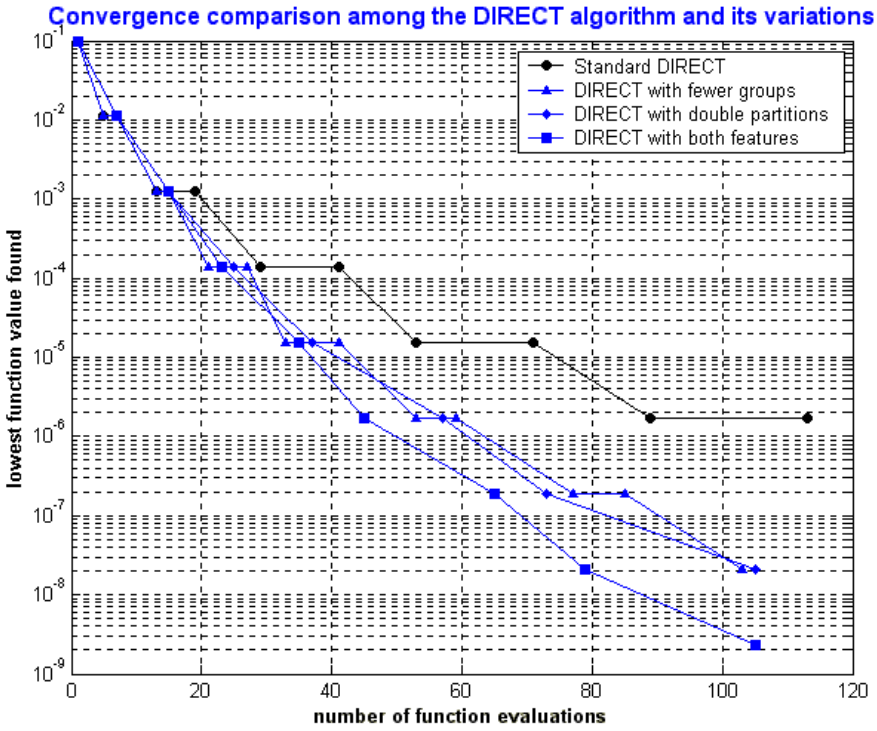

Fig. 13 Convergence comparison for the 2-D case

From Figs. 9 ~ 12 it's clear that all four algorithms converge to the global minimum point. However, the locally biased variations focus more on a local search, thus leaving a relatively larger unexplored area. Using DIRECT, we have only one unexplored box (the largest one, with side length of 1/3). However, we have 5, 3 and 6 such boxes for DIRECT-I, DIRECT-II and DIRECT-III, respectively. Figure 13 shows the convergence comparison among these four algorithms.

Figures 14 ~ 17 show the convergence comparison for the 3-D, 5-D, 10-D and 20-D cases, respectively. In these figures the locally biased variations generally have a faster convergence rate than does the standard DIRECT algorithm. DIRECT-I and DIRECT-II have similar convergence rates. DIRECT-III, which combines those two locally biased measures, has the fastest convergence rate. Since higher dimension problems require a larger search space, a fairly fast convergence rate is of great importance in obtaining the global minimum within the limited number of function evaluations. Since DIRECT-III has the fastest convergence rate, it shows superior performance for higher dimension problems such as the 10-D and the 20-D problems.

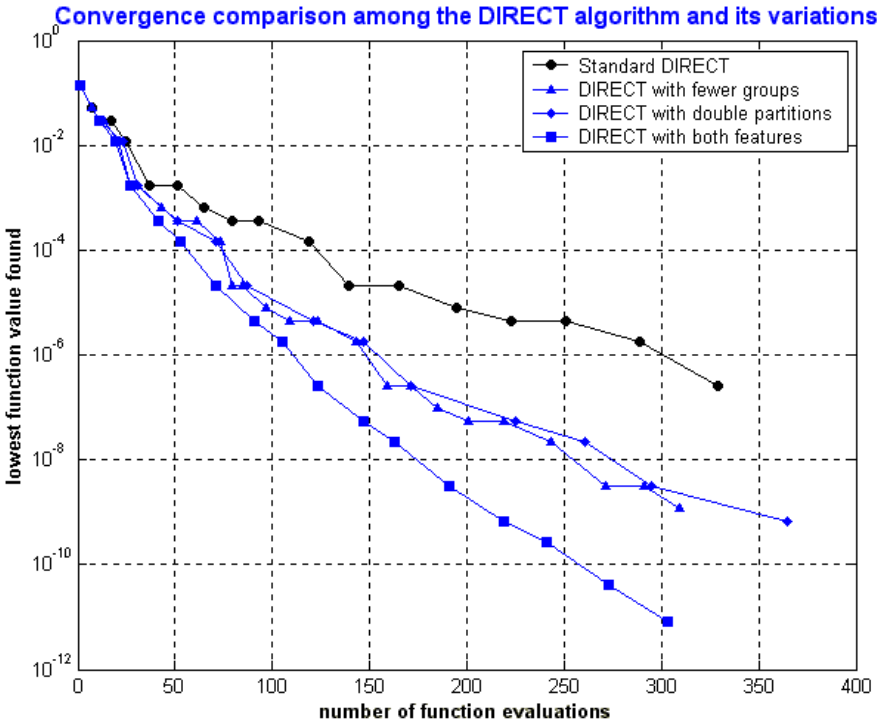

Fig. 14 Convergence comparison for the 3-D case

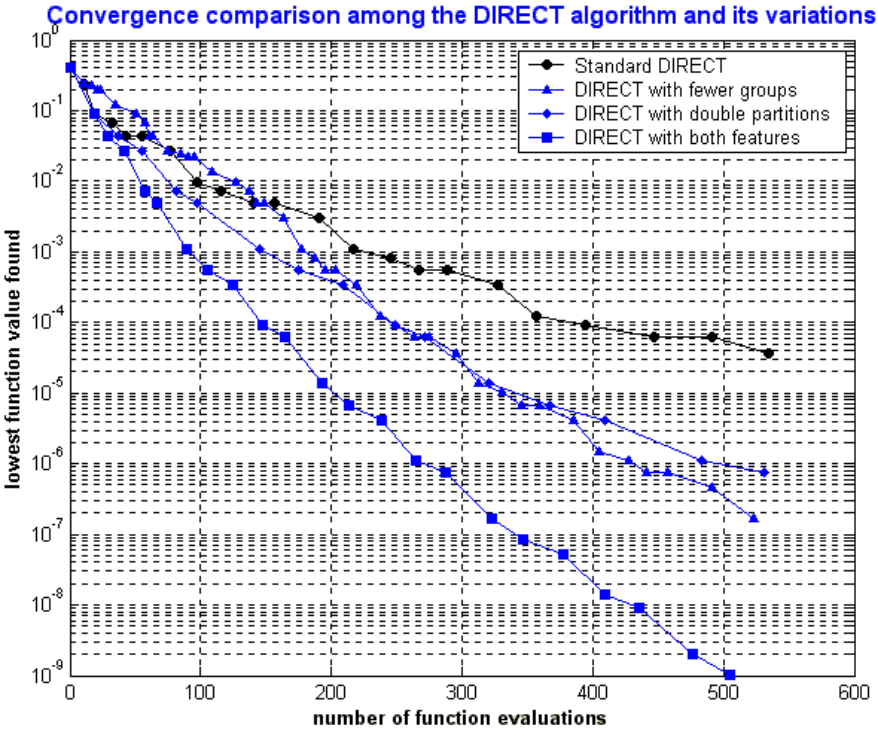

Fig. 15 Convergence comparison for the 5-D case

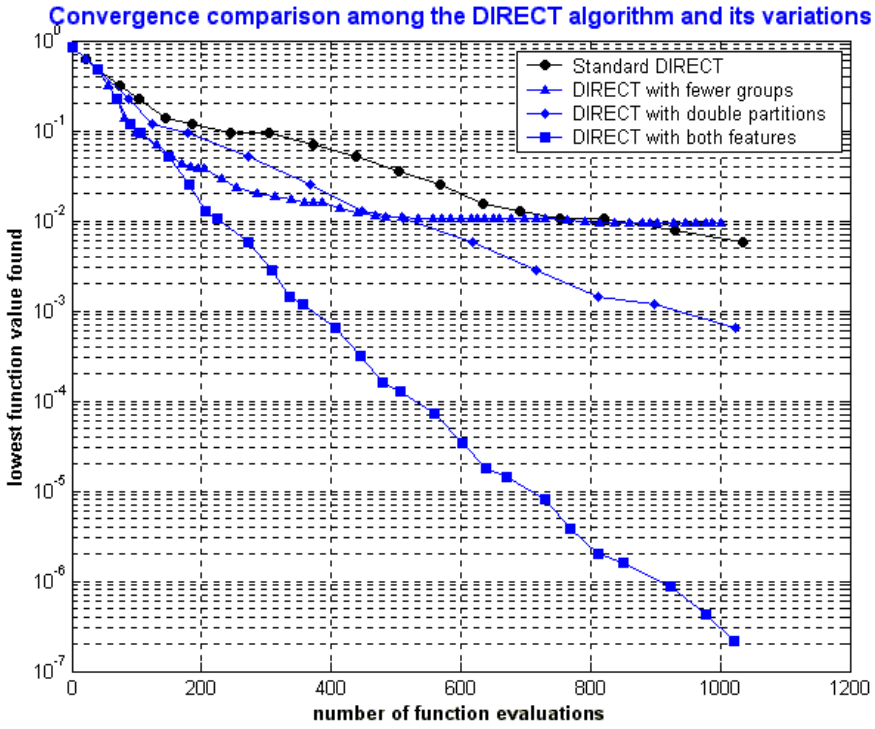

Fig. 16 Convergence comparison for the 10-D case

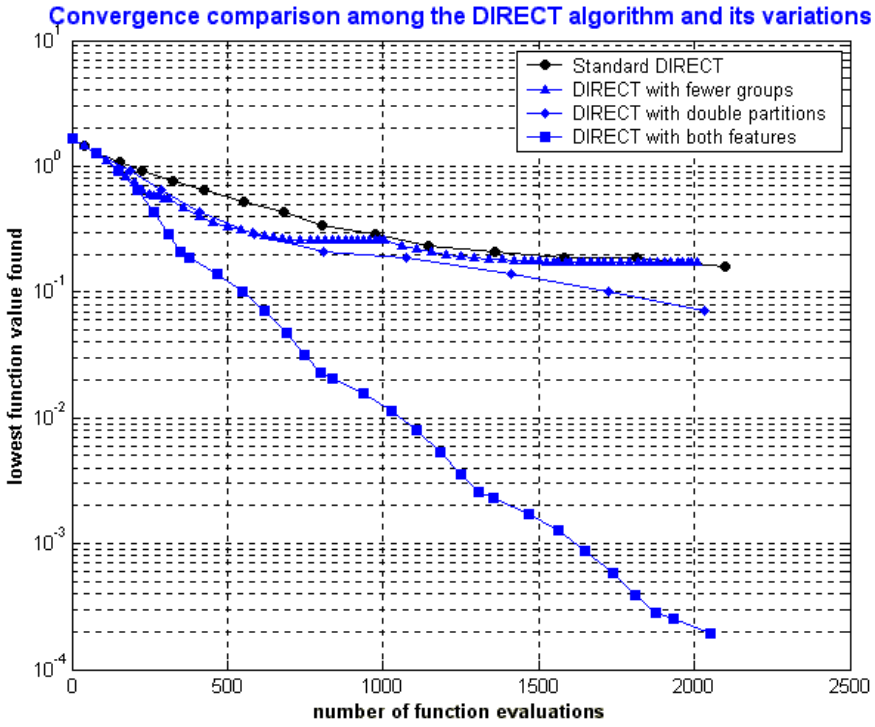

Fig. 17 Convergence comparison for the 20-D case

As an example, let's take a look at the 20-D case. Figure 18 shows the optimization results for the 20-D case.

Table 1 shows the values of the best points found by the three locally biased variations within 2000 function evaluations and the exact solution. Table 1 also shows the relative errors. The results obtained by using DIRECT-III have the smallest relative errors among the standard DIRECT algorithm and its three variations.

From Fig. 18 and Table 1, it is clear that, after 2000 function evaluations, the results obtained by using DIRECT-III almost match the exact solution precisely, far better than the results obtained by using DIRECT, DIRECT-I and DIRECT-II.

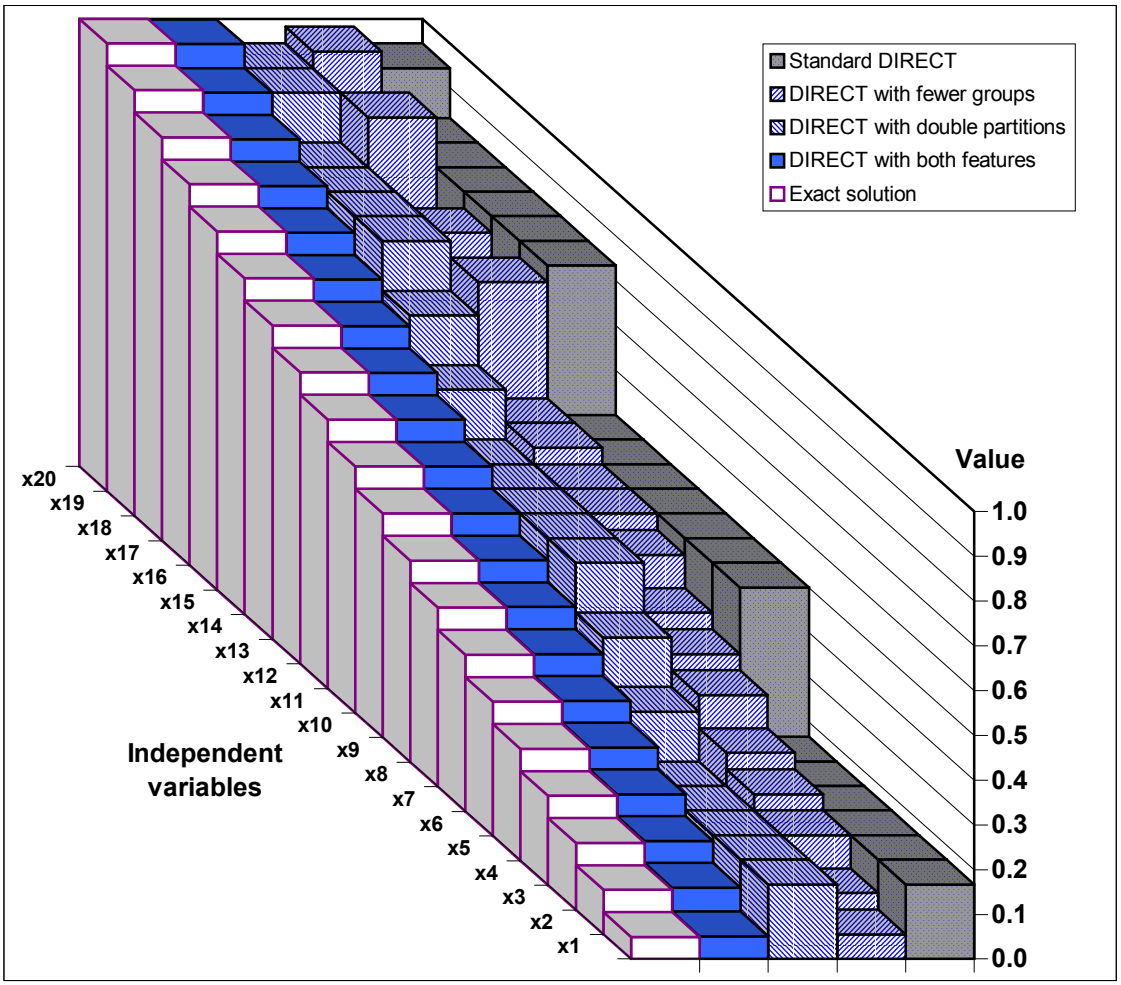

Fig. 18 Comparison of the final optimization results for the 20-D case

	Best Point					Relative Error (%)			
	DIRECT	DIRECT-I	DIRECT-II	DIRECT-III	Exact Solution	DIRECT	DIRECT-I	DIRECT-II	DIRECT-III
x_1	0.167	0.056	0.167	0.051	0.05	234.00	12.00	234.00	2.00
x_2	0.167	0.093	0.167	0.105	0.10	67.00	-7.00	67.00	5.00
x_3	0.167	0.167	0.167	0.154	0.15	11.33	11.33	11.33	2.67
x_4	0.167	0.204	0.167	0.204	0.20	-16.50	2.00	-16.50	2.00
x_5	0.167	0.241	0.167	0.253	0.25	-33.20	-3.60	-33.20	1.20
x_6	0.167	0.315	0.278	0.302	0.30	-44.33	5.00	-7.33	0.67
x_7	0.500	0.352	0.389	0.352	0.35	42.86	0.57	11.14	0.57
x_8	0.500	0.389	0.500	0.401	0.40	25.00	-2.75	25.00	0.25
x_9	0.500	0.463	0.500	0.451	0.45	11.11	2.89	11.11	0.22
x_{10}	0.500	0.500	0.500	0.500	0.50	0.00	0.00	0.00	0.00
x_{11}	0.500	0.500	0.500	0.549	0.55	-9.09	-9.09	-9.09	-0.18
x_{12}	0.500	0.537	0.500	0.599	0.60	-16.67	-10.50	-16.67	-0.17
x_{13}	0.500	0.537	0.611	0.648	0.65	-23.08	-17.38	-6.00	-0.31
x_{14}	0.833	0.796	0.722	0.698	0.70	19.00	13.71	3.14	-0.29
x_{15}	0.833	0.648	0.833	0.747	0.75	11.07	-13.60	11.07	-0.40
x_{16}	0.833	0.796	0.833	0.796	0.80	4.13	-0.50	4.13	-0.50
x_{17}	0.833	0.648	0.833	0.846	0.85	-2.00	-23.76	-2.00	-0.47
x_{18}	0.833	0.944	0.833	0.895	0.90	-7.44	4.89	-7.44	-0.56
x_{19}	0.833	0.648	0.944	0.944	0.95	-12.32	-31.79	-0.63	-0.63
x_{20}	0.944	0.944	0.944	0.998	1.00	-5.60	-5.60	-5.60	-0.20

Table 1 Summary of the optimization results for the 20-D case

3.2 Testing functions with one global minimum point and multiple local minimum points

Here we consider two functions, the Rosenbrock function and the local Shubert function.

The Rosenbrock function is a standard test function in optimization theory, and is defined as: $F(x_1, x_2) = 100\,(x_1 - x_2^2)^2 + (1 - x_1)^2$, where $x_1, x_2 \in [-2.048, 2.048]$. If we normalize the range of variables x_1 and x_2 into $[0,1]$, then the global minimum point is (0.74414, 0.74414) and the global minimum is 0. The global minimum point is located in a long narrow flat valley with lots of local minima.

The surface shape and the contour lines of the Rosenbrock function are shown in Figs. 19 and 20. The round dot in Fig. 19 represents the global minimum point.

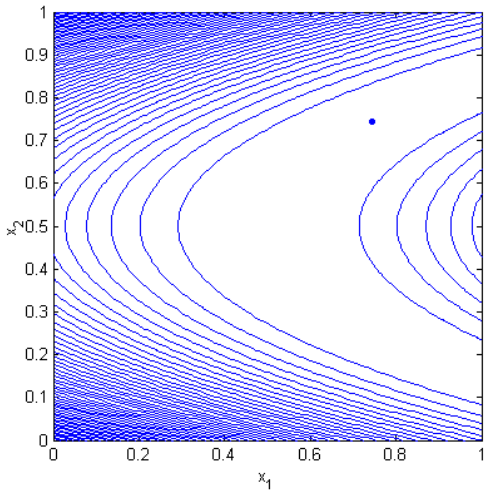

Fig. 19 Contour lines

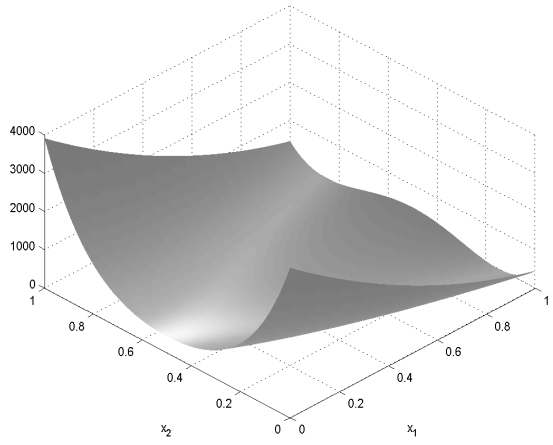

Fig. 20 Surface shape

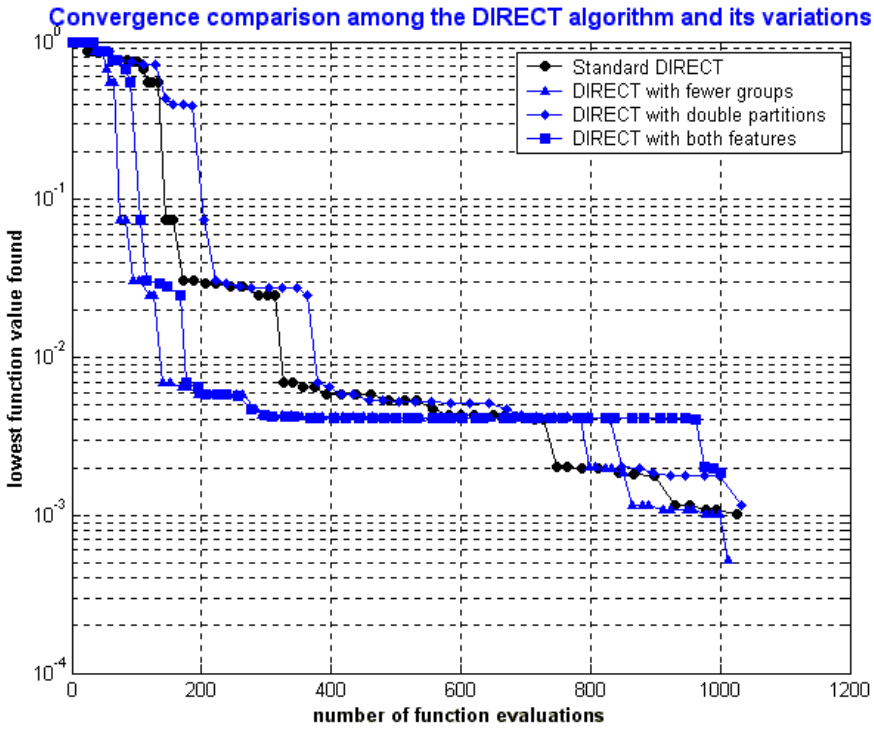

Fig. 21 Convergence comparison for the Rosenbrock function

The convergence comparison in Fig. 21 shows that DIRECT-I and DIRECT-III have faster convergence rates in the early stage. However, all four algorithms show similar convergence rates in the final phase.

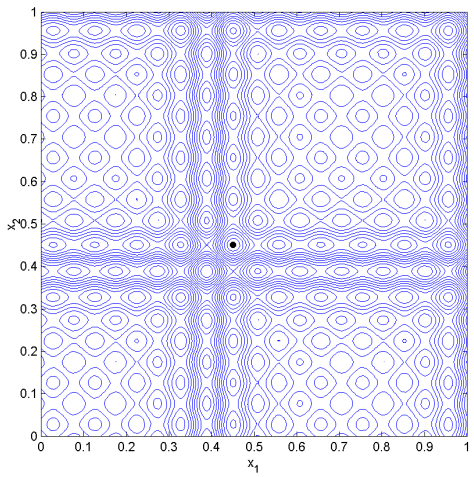

Fig. 22 Contour lines

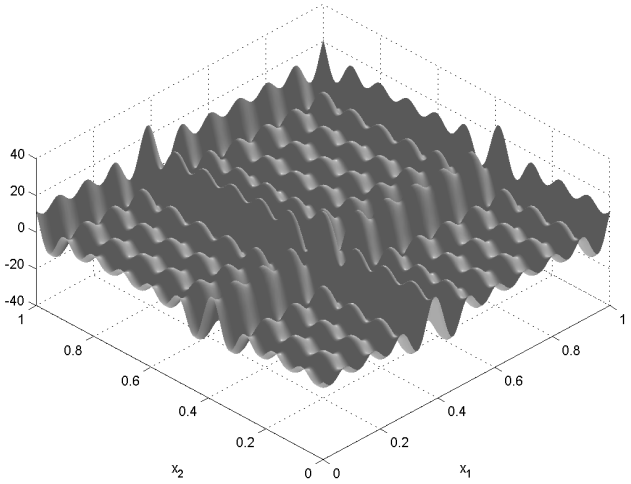

Fig. 23 Surface shape

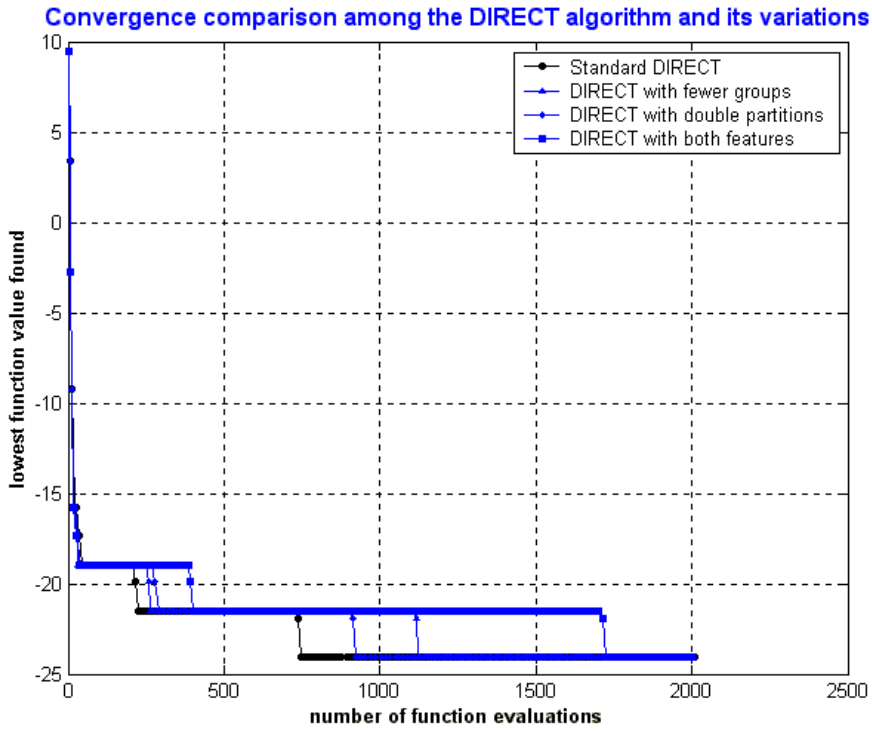

Fig. 24 Convergence comparison for the local Shubert function

The second function is the local Shubert function, which is defined as follows:

$$F(x_1, x_2) = -(\sum_{i=1}^{5} i \sin((i+1)x_1 + i) + \sum_{j=1}^{5} j \sin((j+1)x_2 + j)) ,$$

where $x_1, x_2 \in [-5, 5]$. It has one global minimum point and 100 local minimum points. If we normalize the range of variables x_1 and x_2 into $[0,1]$, then its global minimum point is (0.4508609, 0.4508609) and the global minimum is -24.062499. The 3-D surface and 2-D contour of the local Shubert function are shown in Figs. 22 and 23, respectively. The round dot in Fig. 22 denotes the global minimum point.

The convergence comparison given in Fig. 24 shows that, for this case, DIRECT has the fastest convergence rate and DIRECT-III has the slowest. However, all the algorithms converge to the same global minimum point. It is easy to imagine that, because the local Shubert function has many local minimum points, it would be likely for the locally biased variations of the DIRECT algorithm to spend too much time on the local searches. These locally biased variations get to the global minimum point slower than does the standard DIRECT algorithm.

3.3 Testing functions with multiple global and local minimum points

The first testing function considered here is called the "six-hump" function, defined as:

$$F(x_1, x_2) = 4x_1^2 - 2.1x_1^4 + (1/3)x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4,$$

where $x_1 \in [-2, 2]$, $x_2 \in [-1, 1]$. This function has two global minimum points and 4 other local minimum points. If we normalize the range of variables x_1 and x_2 into $[0,1]$, then its global minimum points are (0.52246, 0.14367) and (0.47754, 0.85633) and its global minimum is -1.03163.

The surface shape and the contour lines of the six-hump function are shown in Figs. 25 and 26. The round dots in Fig. 25 represent the global minimum points. We can clearly discern the six "humps" from these two figures. Figures 27 ~ 30 show the optimization results obtained by using DIRECT, DIRECT-I, DIRECT-II and DIRECT-III, respectively. The tiny dots represent the sample points in the center of the boxes. The centers of the circles represent the position of the global minimum points. We can observe the strongly biased property of the DIRECT-III by looking at its large unexplored area.

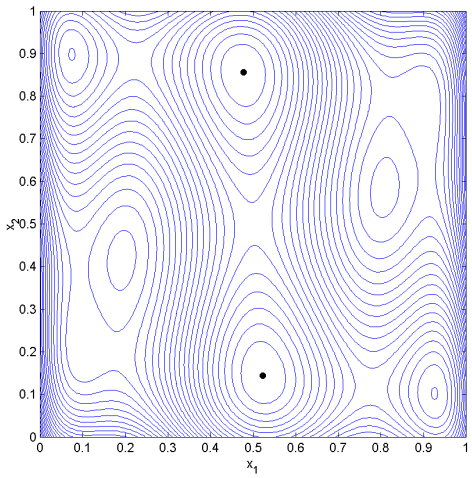

Fig. 25 Contour lines

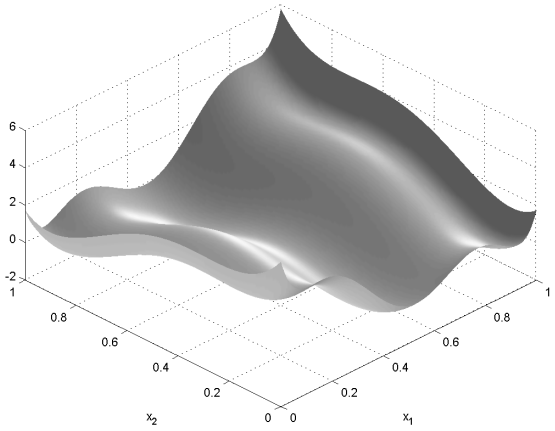

Fig. 26 Surface shape

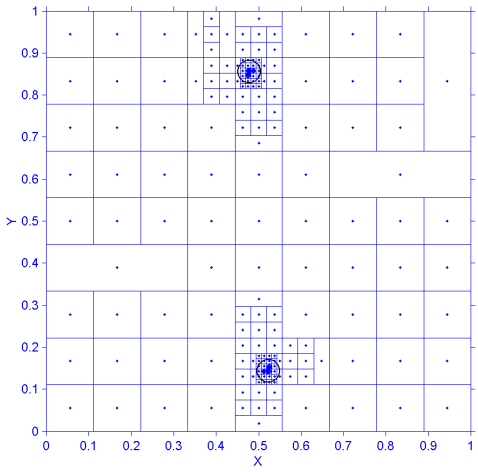

Fig. 27 Results of DIRECT

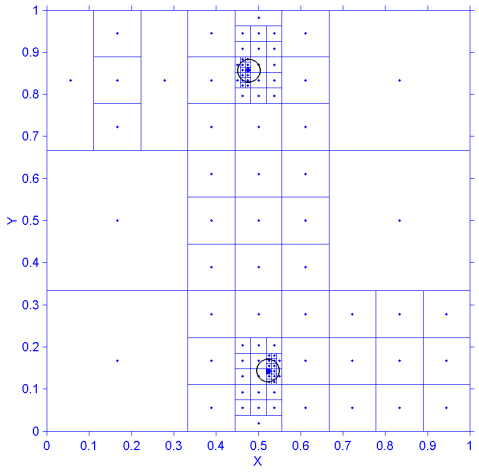

Fig. 28 Results of DIRECT-I

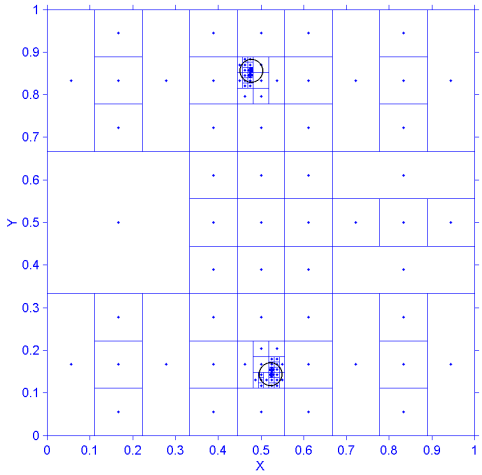

Fig. 29 Results of DIRECT-II

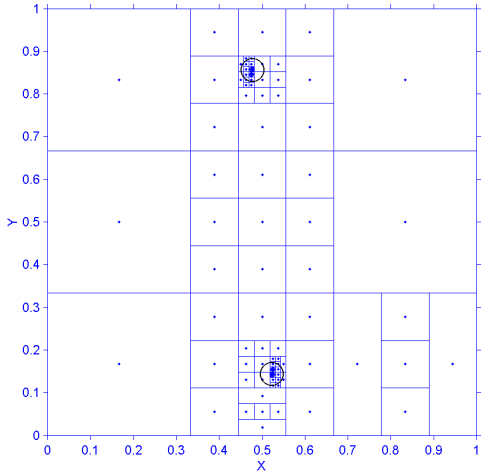

Fig. 30 Results of DIRECT-III

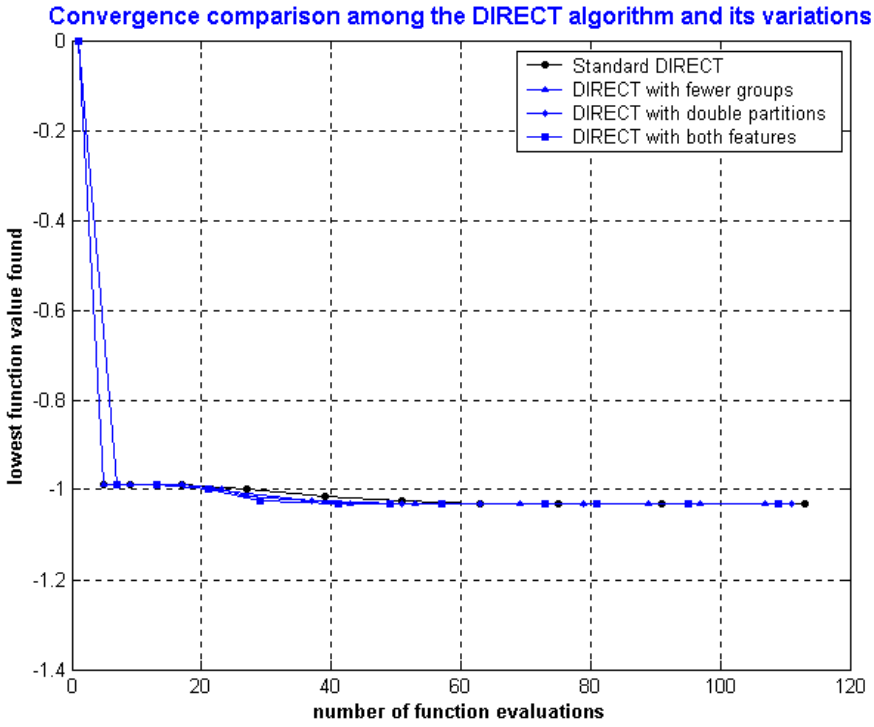

Fig. 31 Convergence comparison for the six-hump function

It's clear from Figs. 27 ~ 30 that sample points cluster around the two global minimum points for both the standard DIRECT algorithm and its variations. Thus, all algorithms found the two global minimum points. Figure 31 shows the convergence comparison. The DIRECT algorithm and its variations show very similar fast convergence rates.

The second testing function we considered is the Branin function, defined as:

$$F(x_1, x_2) = [1 - 2x_2 + (1/20)\sin(4\pi x_2) - x_1]^2 + [x_2 - (1/2)\sin(2\pi x_1)]^2 ,$$

where $x_1, x_2 \in [-10,10]$. This function has five global minimum points and the global minimum is 0. If we normalize the range of variables x_1 and x_2 into $[0,1]$, then the five global minimal points are (0.55, 0.5), (0.50743, 0.52010), (0.52013, 0.51437), (0.57987, 0.48563) and (0.59257, 0.47990). The surface shape and the contour lines of the Branin function are shown in Figs. 32 and 33. The five round dots in Fig. 32 represent the global minimum points.

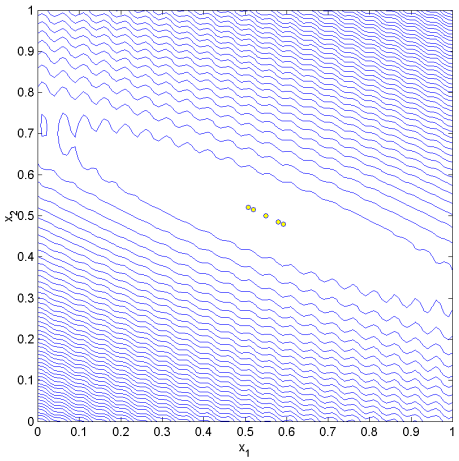

Fig. 32 Contour lines

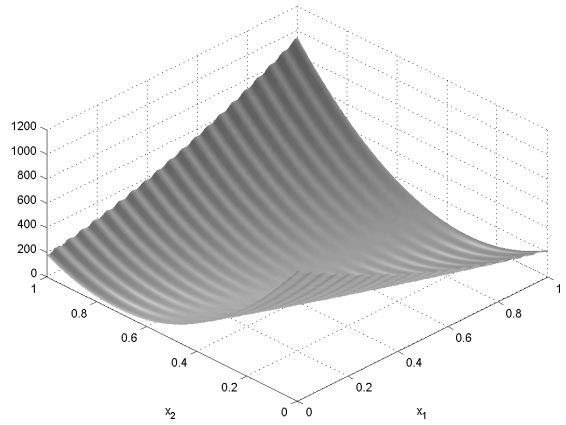

Fig. 33 Surface shape

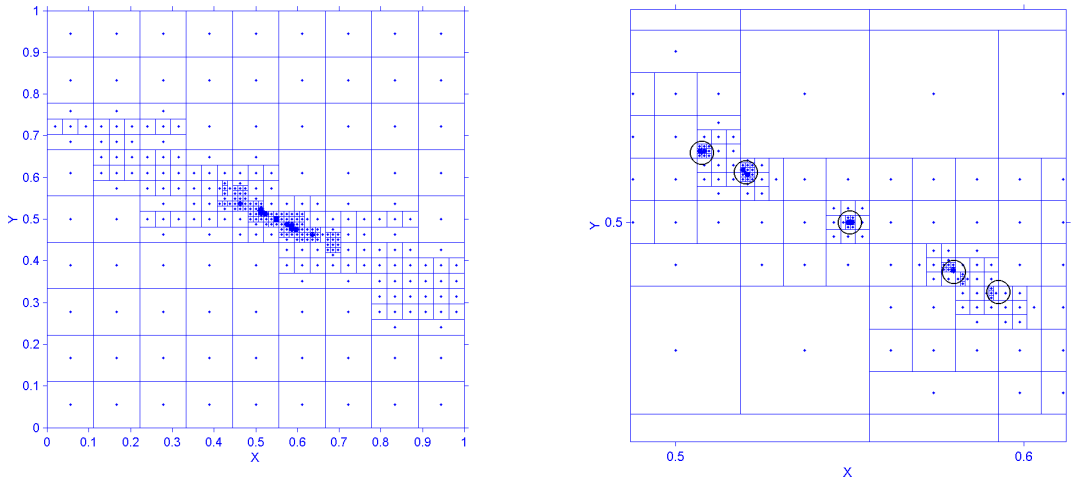

Fig. 34 Results of DIRECT

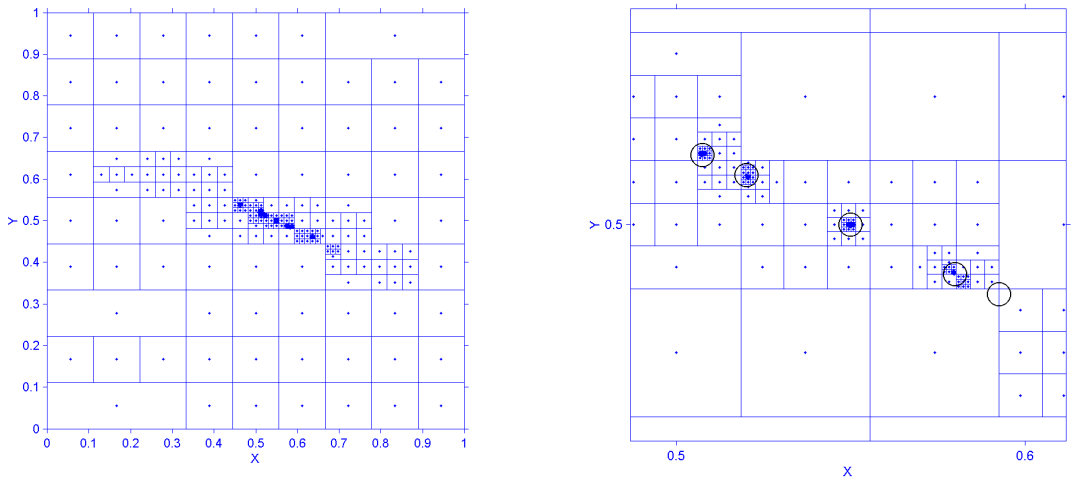

Fig. 35 Results of DIRECT-I

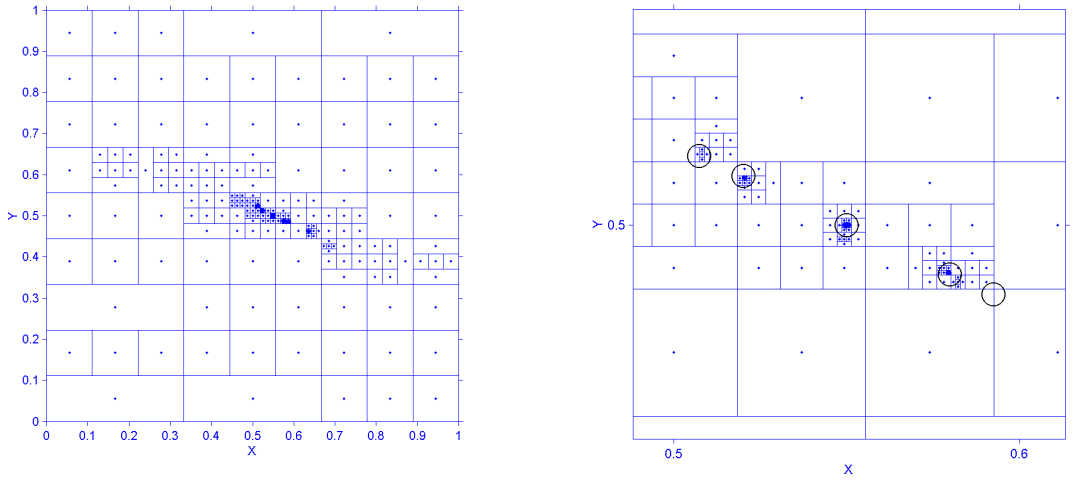

Fig. 36 Results of DIRECT-II

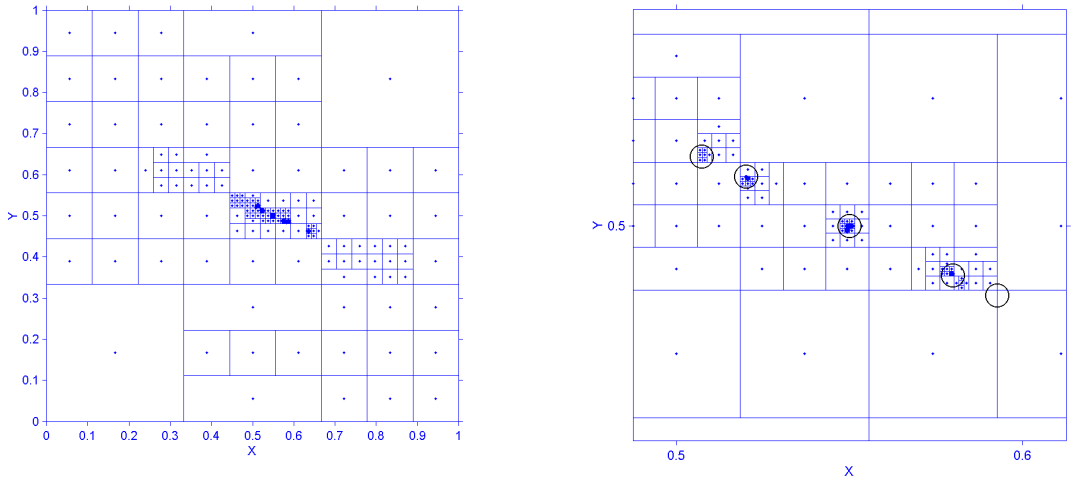

Fig. 37 Results of DIRECT-III

Figures 34 ~ 37 show the optimization results after 500 function evaluations using DIRECT, DIRECT-I, DIRECT-II and DIRECT-III, respectively. The tiny dots in the figures on the left represent the sample points. The figures on the right are the local zoom-ins of the ones on the left, around the global minimum points. The centers of the circles denote the locations of the global minimum points.

Figure 34 shows that DIRECT found all of the five global minimum points at this stage, while Figs. 35, 36 and 37 show that only four global minimum points were found by the three locally biased variations of DIRECT.

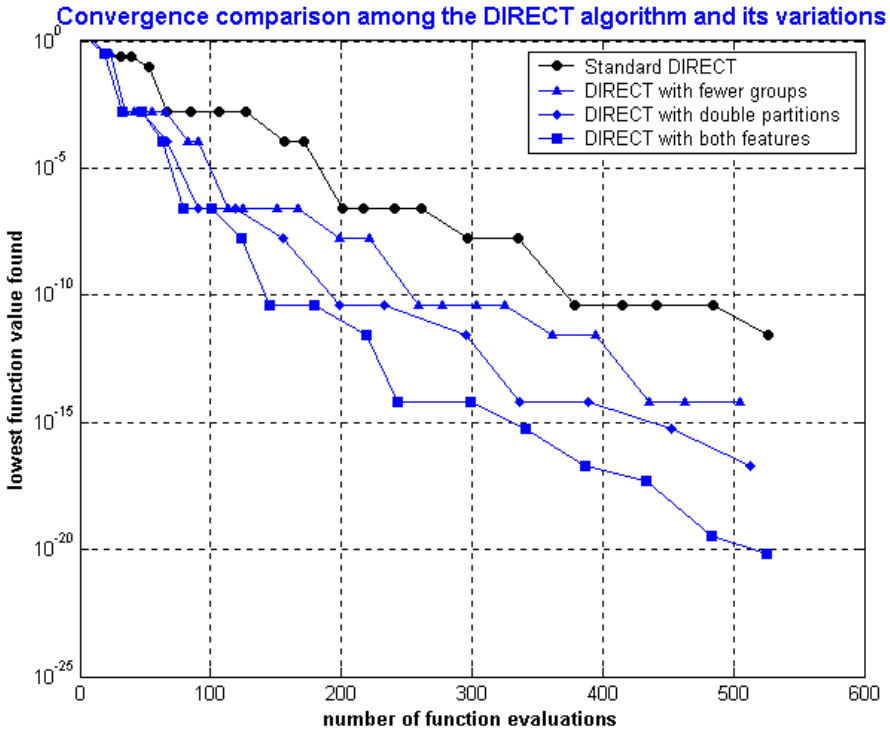

Fig. 38 Convergence comparison for the Branin function

Figure 38 shows the convergence comparison. The locally biased DIRECT algorithms clearly have higher convergence rates than does the standard DIRECT algorithm with DIRECT-III having the fastest convergence rate among them.

Therefore, for this testing case the locally biased DIRECT algorithms have faster convergence rate, but the standard DIRECT algorithm found all the global minimum points more quickly.

Next we will discuss the performance of the standard DIRECT algorithm and its three locally biased variations in the slider Air Bearing Surface (ABS) optimization.

4. SLIDER AIR BEARING DESIGN OPTIMIZATION CASE

4.1 Air bearing design optimization problem

Given a prototype slider ABS design, we wish to optimize it to get uniform flying heights near the target flying height with a flat roll profile, and to increase its air bearing stiffness if possible.

In this case we used the NSIC 7nm flying height slider as the prototype. The rail shape and the 3-dimensional rail geometry are shown in Figs. 39 and 40, respectively.

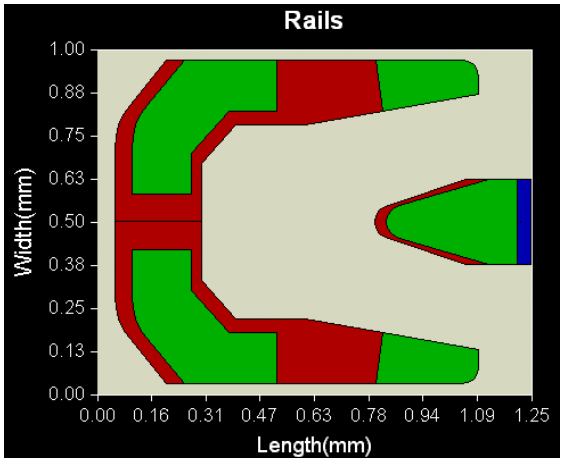

Fig. 39 Rail shape of the initial ABS design

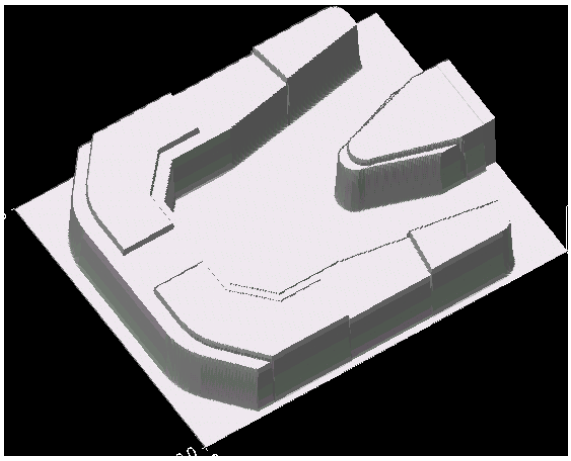

Fig. 40 3-D rail shape of the initial ABS design

The slider is a Pico slider (1.25×1.0mm) that flies over a disk rotating at 7200 RPM. Its flying heights are all around 7nm from OD to ID. For this case we want to lower all its flying heights to the target flying height, i.e. 5nm, and at the same time maintain a flat roll profile at the three different radial positions OD, MD and ID. The objective function or cost function is defined as:

$$
\begin{aligned}
&\boldsymbol{1}\times(\textit{FH Max Difference}) + \\
&\boldsymbol{9}\times(\textit{FH}) + \\
&\boldsymbol{1}\times(\textit{Roll}) + \\
&\boldsymbol{1}\times(\textit{Roll Cutoff}) + \\
&\boldsymbol{1}\times(\textit{Pitch Cutoff}) + \\
&\boldsymbol{1}\times(\textit{Vertical Sensitivity}) + \\
&\boldsymbol{1}\times(\textit{Pitch Sensitivity}) + \\
&\boldsymbol{1}\times(\textit{Roll Sensitivity}) + \\
&\boldsymbol{1}\times(\textit{Negative Force}).
\end{aligned}
$$

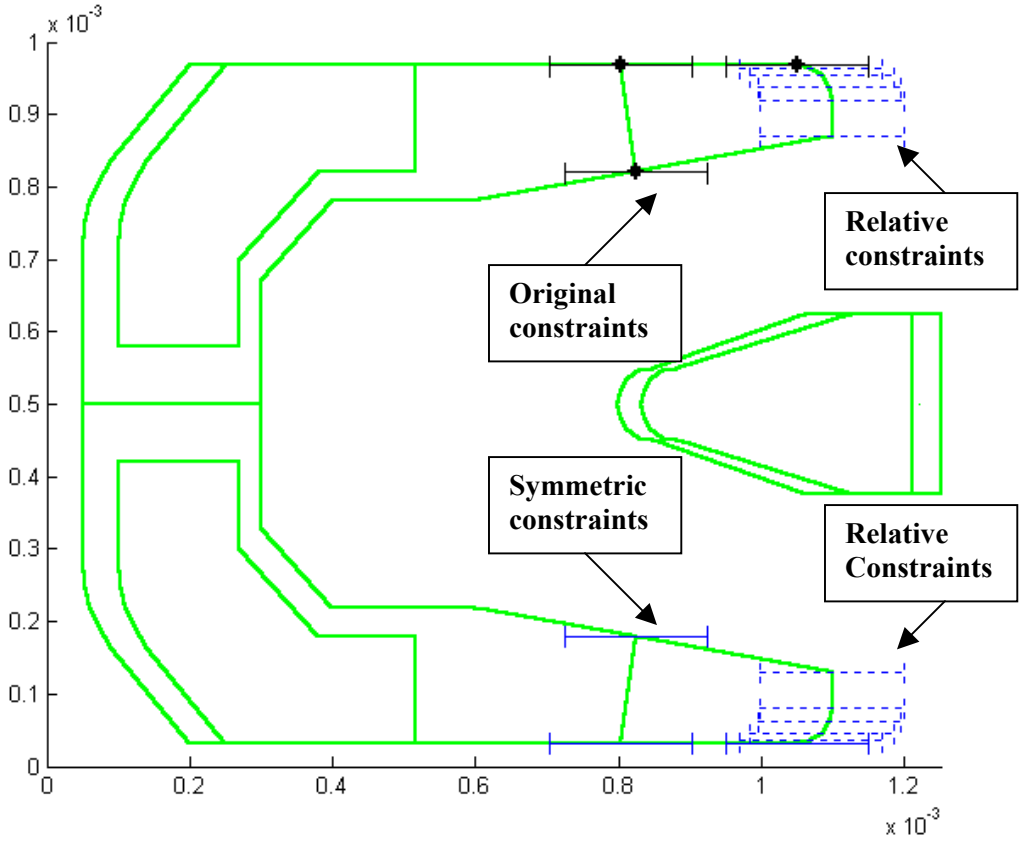

Fig. 41 Constraints defined on the initial design

The goal of the optimization is to minimize this multi-objective function under the given constraints (note that since we are primarily concerned with the flying heights, we put a heavier weight (9) on that term). All the objective terms are normalized and their definitions can be found in the "CML optimization program version 2.0 user's manual"[3]. The constraints we defined are shown in Fig. 41, and the definition of those constraints can also be found in the user's manual.

4.2 Simulation results

Using the initial design, constraints and objective function, we carried out the optimization using the DIRECT algorithm and its three locally biased variations.

Figure 42 shows the convergence comparison. For this testing case, all four algorithms show a similarly fast convergence rate. The best objective function values obtained by using DIRECT, DIRECT-I, DIRECT-II and DIRECT-III are 4.46, 4.46, 4.43 and 4.43.

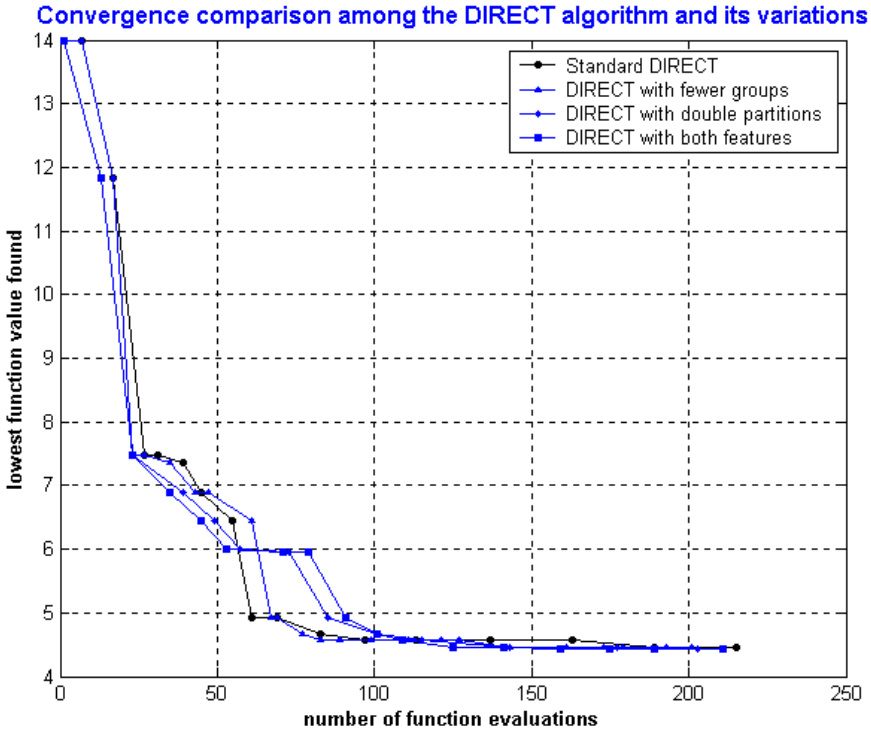

Fig. 42 Convergence comparison for the ABS optimization case

Figures 43 ~ 46 show the optimized ABS designs obtained after 200 function evaluations by using DIRECT, DIRECT-I, DIRECT-II and DIRECT-III, respectively. In these figures, the green lines (light-colored) show the rail shape of the initial design and the blue lines (dark-colored) show the rail shape of the optimized design. The four optimized ABS designs are almost the same.

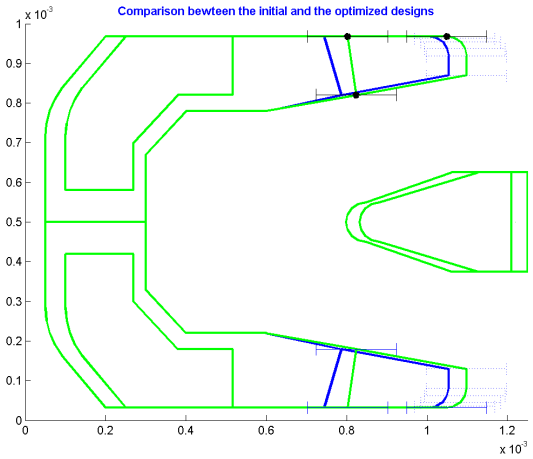

Fig. 43 Results of DIRECT

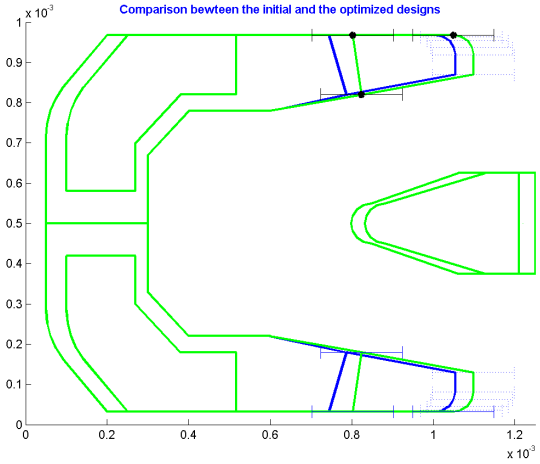

Fig. 44 Results of DIRECT-I

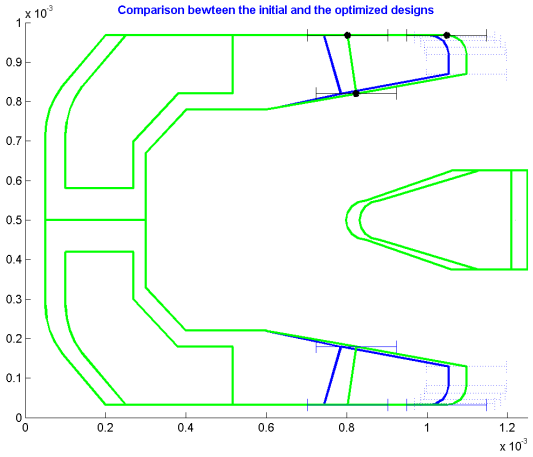

Fig. 45 Results of DIRECT-II

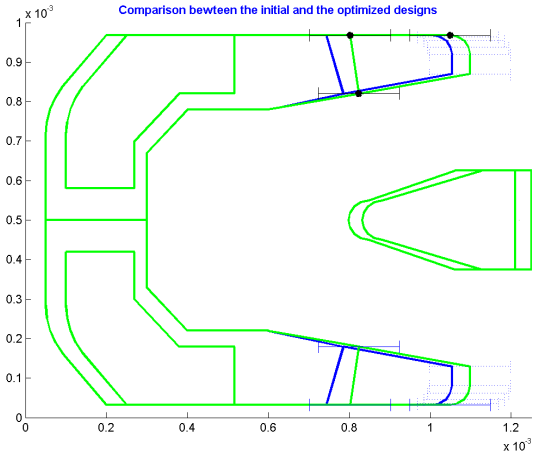

Fig. 46 Results of DIRECT-III

Figures 47 ~ 50 show the variations of the slider performance parameters for all the best-so-far designs using the standard DIRECT algorithm and its three variations respectively. All four optimized ABS designs show very uniform flying heights around the target 5nm FH, and a reasonably flat roll profile.

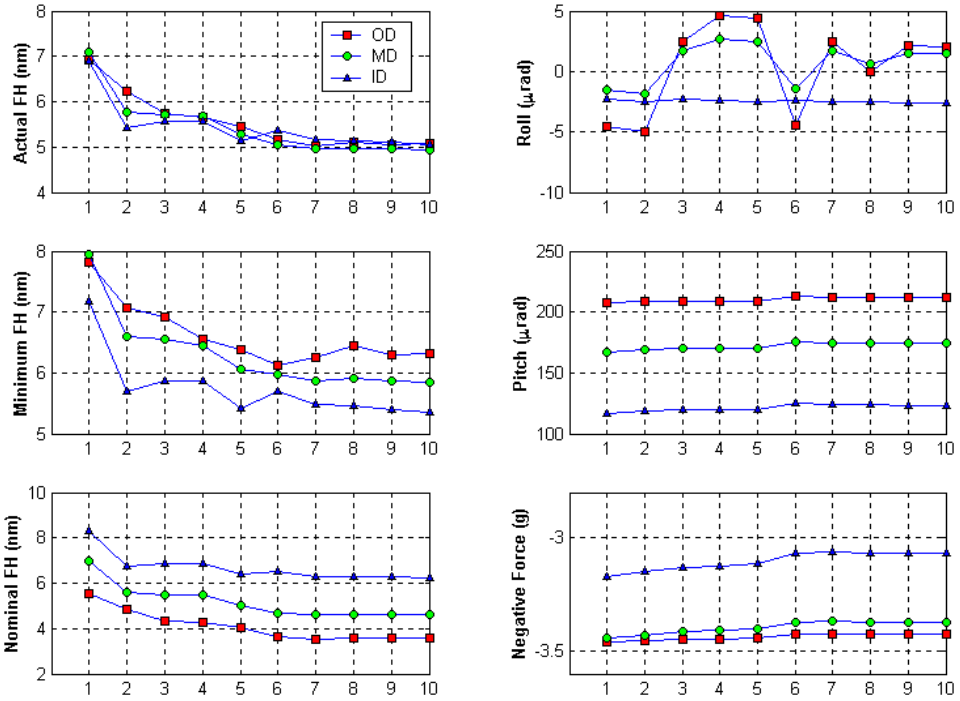

Fig. 47 Variations of the performance parameters using DIRECT

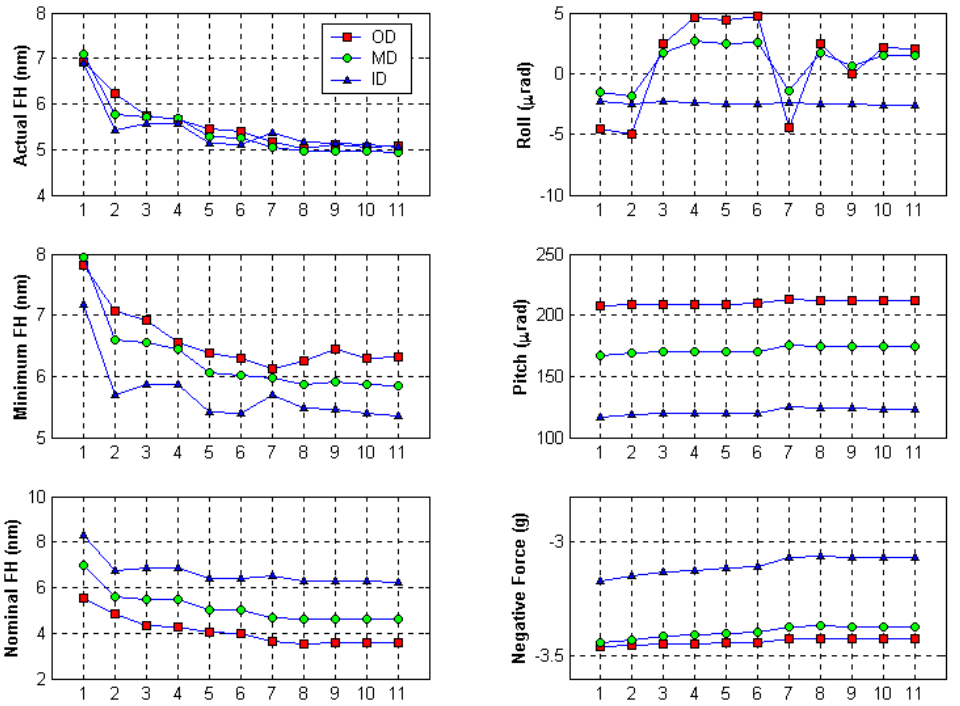

Fig. 48 Variations of the performance parameters using DIRECT-I

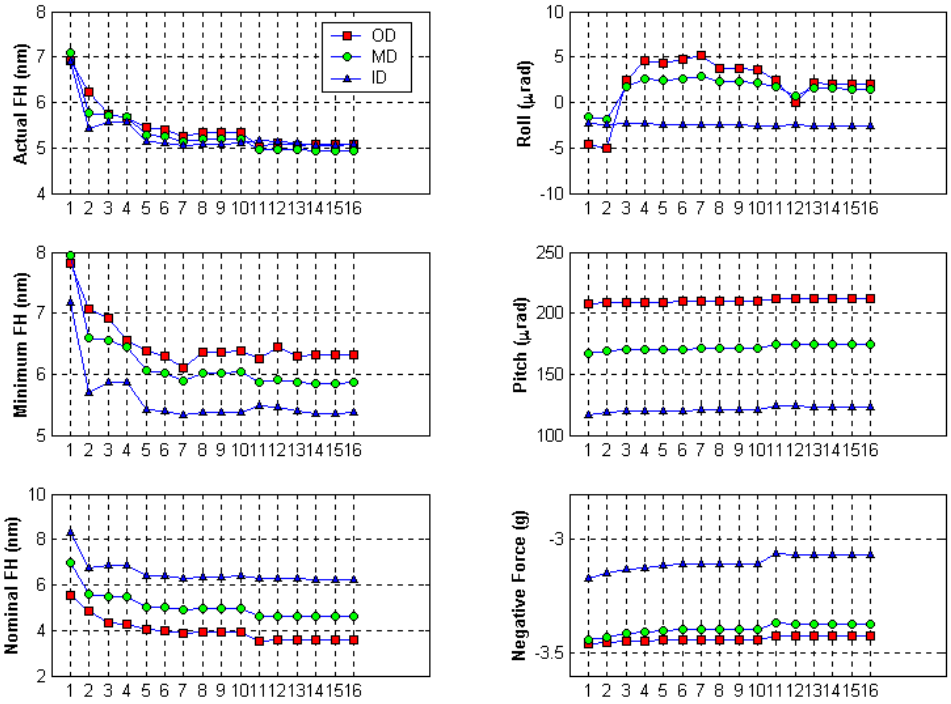

Fig. 49 Variations of the performance parameters using DIRECT-II

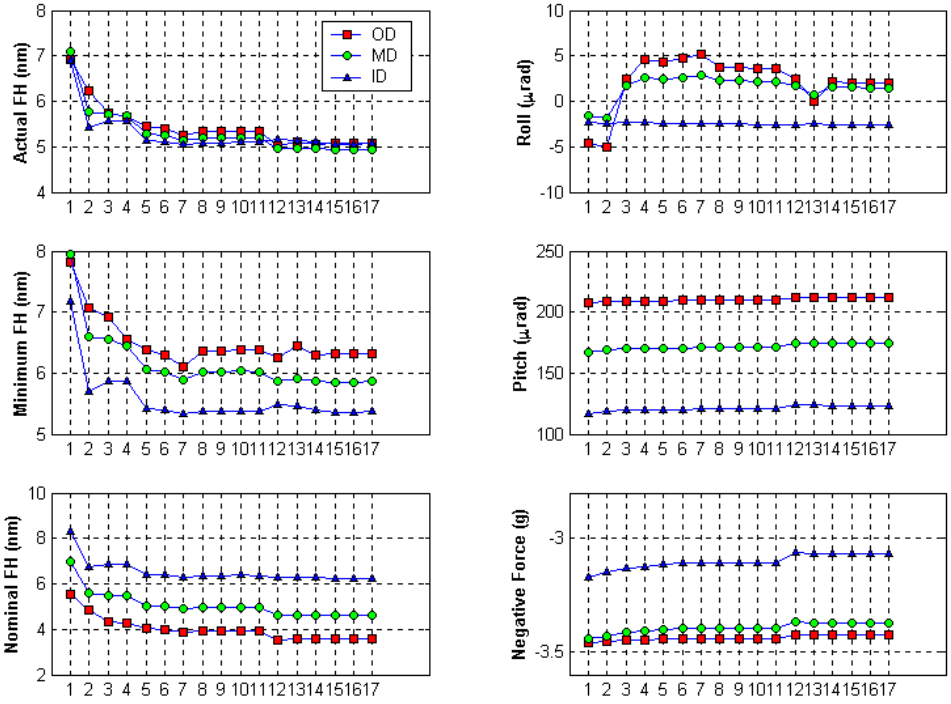

Fig. 50 Variations of the performance parameters using DIRECT-III

7. CONCLUSION

The DIRECT algorithm is a deterministic global optimization technique used to find the minimum of a Lipschitz continuous function without knowing the Lipschitz constant.

We carried out extensive numerical experiments using the DIRECT algorithm and its three locally biased variations, i.e., DIRECT-I (having fewer groups), DIRECT-II (having double partitions for the box containing the point with the lowest function value), and DIRECT-III (which combines these two measures).

For testing functions with only one global and local minimum point, all the locally biased variations have faster convergence rates than does the standard DIRECT algorithm. DIRECT-I and DIRECT-II have similar convergence rates, whereas DIRECT-III has a faster convergence rate. For higher dimension problems, DIRECT-III is superior to the other three and can find the global minimum point far more quickly.

For testing functions with one global minimum point and multiple local minimum points, it's hard to tell which algorithm is best. Though they might show different convergence properties at some stages, they all show a similar convergence trend in the long run.

For testing functions with multiple global and local minimum points, the locally biased variations have a similar or higher convergence rate than does the standard DIRECT algorithm. However, the standard DIRECT will find all the global minimum points earlier.

The slider ABS optimization problem is a strongly nonlinear problem. The results of the test case show very similar performance for DIRECT and its three variations.

In summary, the three locally biased variations of the DIRECT algorithm generally have higher convergence rates than does the standard DIRECT algorithm. The variations perform especially well in some situations and they may dramatically reduce the time needed to find the global minimum points.

ACKNOWLEDGEMENTS

This study is supported by the Computer Mechanics Laboratory (CML) at the University of California at Berkeley and partially supported by the Extremely High Density Recordings (EHDR) project of the National Storage Industry Consortium (NSIC).

REFERENCES

1. Zhu, H. and Bogy, D., 2001, "*DIRECT Algorithm and its Application to Slider Air Bearing Surfaces Optimization*", Technical Report 2001-003, Computer Mechanics Laboratory, University of California at Berkeley.

2. Zhu, H. and Bogy, D., 2000, "*Optimization of Slider Air Bearing Shapes using Variations of Simulated Annealing*", Technical Report 2000-010, Computer Mechanics Laboratory, University of California at Berkeley.

3. Zhu, H. and Bogy, D., 2001, "*The CML Air Bearing Optimization Program Version 2.0*", Technical Report, Computer Mechanics Laboratory, University of California at Berkeley.

4. Jones, D. R., Perttunen, C. D. and Stuckman, B. E., 1993, "*Lipschitzian Optimization Without the Lipschitz Constant* ", Journal of Optimization Theory and Application, Vol. 79, No. 1, pp 157-181.

5. Gablonsky, J. M., 1998, "*An Implementation of the DIRECT algorithm*", Technical Report CRSC-TR98-29, Center for Research in Scientific Computation, North Carolina State University.

6. Gablonsky, J. M. and C. T. Kelley, 2000, "*A locally-biased form of the DIRECT algorithm*", Technical Report CRSC-TR00-31, Center for Research in Scientific Computation, North Carolina State University.