An Adaptive Restart Implementation of DIRECT *

D.E. Finkel and C.T. Kelley †
North Carolina State University, Center for Research in Scientific Computation and Department of Mathematics, Box 8205, Raleigh, N. C. 27695-8205, USA
*(*definkel@ncsu.edu*), (*tim_kelley@ncsu.edu*)*

August 24, 2004

Abstract. This paper is concerned with the algorithmic behavior of the DIRECT (DIviding RECTangles) algorithm. We show that DIRECT is sensitive to additive scaling, and this sensitivity can affect convergence. We present a modified version of the algorithm, and illustrate the effectiveness of our modification with numerical results.

Keywords: DIRECT, global optimization, additive scaling

1. Introduction

DIRECT (DIviding RECTangles) [16] is an optimization algorithm designed to aggressively search for global minima of a real valued objective function over a bound-constrained domain. The algorithm does not use derivative information in its search; instead, it relies on the iteration history to determine future sample locations.

The strength of DIRECT is the balanced global and local search it performs. This article describes conditions under which the balance of the algorithm becomes skewed. We observe that DIRECT is sensitive to additive scaling, and has slow asymptotic convergence. We suggest a modification to the algorithm, and present test results that illustrate the effectiveness of our modification.

For this paper, we are concerned with a bound constrained optimization problem:

$$(P) \qquad \min_{x \in \Omega} f(x), \qquad \text{where } f : \mathbb{R}^n \to \mathbb{R}, \tag{1}$$

with $\Omega = \{x \in \mathbb{R}^n : l_i \leq x_i \leq u_i, \;\; i = 1...n\}$, and $l, u \in \mathbb{R}^n$ given. We assume that f is Lipschitz continuous on Ω. In many applications, f is nonsmooth, or no derivative information is available. For example, an evaluation of f may require several different simulations to be performed [5, 17]. The simulators can have nonsmooth functions built into them (e.g. **IF** statements, **max** functions and table lookups), or may

* This research was supported in part by National Science Foundation grants DMS-0070641, DMS-0112542, and DMS-0209695.
† Version as of August 24, 2004.

add noise to the problem via truncation error. Finite differences may fail to accurately approximate the gradient of f. Sampling methods, such as DIRECT, can solve such problems when gradient-based methods fail.

DIRECT can be effective [16, 9, 4, 20, 18, 3] in finding the basin of convergence for a global solution on low dimensional problems. Unlike other derivative-free methods ([19, 14, 15, 2, 1, 12], DIRECT carries on an aggressive global search.

The algorithm operates by systematically dividing the box domain, Ω, into hyperrectangles, and evaluating the objective function in their centers. There are two phases to an iteration of DIRECT; first, hyperrectangles are identified as *potentially optimal*, i.e., they have potential to contain a global solution. The second phase of an iteration is to divide potentially optimal hyperrectangles into smaller hyperrectangles. The objective function is evaluated in the centers of new hyperrectangles. An balance parameter is used to bias the search towards unexplored regions of the domain. In this paper we explore the effects of using this parameter, and proposes a modification to the algorithm based on the analysis.

DIRECT performs a global search in that the algorithm continues to search for global solutions after local minima have been detected. When given no termination criteria, DIRECT will exhaustively sample the domain [16]; an observation that has been used to describe theoretical nonsmooth convergence of the algorithm [7]. In practice, DIRECT typically clusters sample points around local and global optima after a few iterations [20, 18, 3]. DIRECT can be implemented so that many evaluations of the objective function are done simultaneously on a parallel machine [11].

In the next section, we describe DIRECT. In Section 3, we examine some vulnerabilities of the algorithm. Section 4 is a description of our modification to DIRECT. We close with test results.

2. DIRECT

The DIRECT algorithm begins by scaling the domain, Ω, to the unit hypercube with the linear transformation $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$

$$\mathcal{T}(x) = X(u - l) + l, \quad \text{for all } x \in \Omega$$

where X is a $n \times n$ matrix with elements of x along its diagonal, and zeros elsewhere.

This mapping does not change the optimization process, and simplifies analysis of the algorithm. Therefore, we assume for the rest of

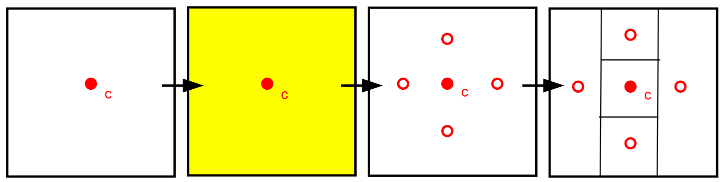

Figure 1. The first iteration of DIRECT.

this paper that

$$\Omega = \{x \in \mathbb{R}^n : 0 \leq x_i \leq 1, \quad i = 1...n\}.$$

2.1. The first iteration

DIRECT begins by evaluating f at the center of Ω, $c = (1/2, ..., 1/2)$. Determining potentially optimal hyperrectangles in the first iteration is trivial; Ω itself is the first potentially optimal hyperrectangle. DIRECT moves to the next phase of the iteration, and divides the potentially optimal hyperrectangle.

DIRECT begins the division process by evaluating f at neighbors of c in every dimension. The neighbors are determined to be the midpoints between c and the boundary. Thus, f is evaluated at

$$c \pm \frac{1}{3} e_i, \quad \text{for all } i \in [1, n],$$

where e_i is the ith unit vector. The $2n$ points sampled become centers of their own hyperrectangles, and the algorithm continues to the next iteration. Figure 1 illustrates this process on a two-dimensional example. We provide more details on this procedure in the next section.

2.2. General Iterations of DIRECT

After the first iteration, the algorithm selects potentially optimal hyperrectangles sparingly. Rules for division are also developed so that all dimensions are sampled equally.

The definition of a potentially optimal hyperrectangle is given below, and is originally from [16].

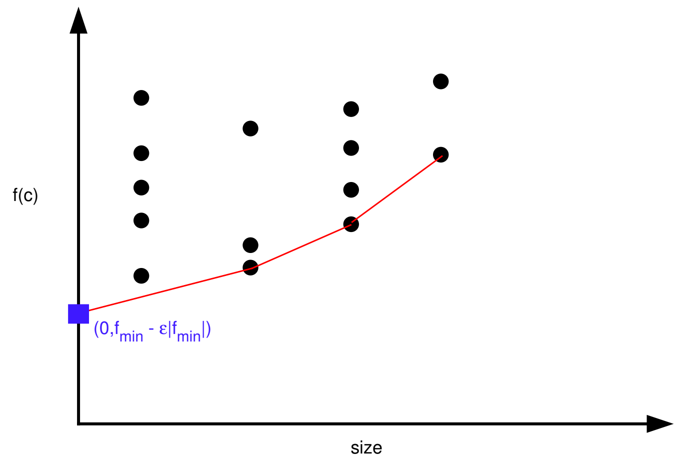

Figure 2. Geometric interpretation of (2) and (3).

DEFINITION 1. *Let $\mathcal{S}$ be the set of hyperrectangles created by DIRECT after k iterations, and let $f_{\min}$ be the best value of the objective function found so far. A hyperrectangle $R \in \mathcal{S}$ with center c_R and size $\alpha(R)$ is said to be potentially optimal if there exists $\hat{K}$ such that*

$$f(c_R) - \hat{K}\alpha(R) \;\leq\; f(c_T) - \hat{K}\alpha(T), \quad \textit{for all } T \in \mathcal{S} \tag{2}$$

$$f(c_R) - \hat{K}\alpha(R) \;\leq\; f_{\min} - \epsilon|f_{\min}|. \tag{3}$$

In [16], hyperrectangle size is measured by the distance from its center to a vertex. In [10], the authors modified `DIRECT` and measure hyperrectangles by their longest side. In practice, this modification biases the algorithm toward local solutions [10].

Figure 2 is a geometric interpretation of Definition 1. Each point on the graph represents a hyperrectangle in $\mathcal{S}$, with an additional square dot added at $(0, f_{\min} - \epsilon|f_{\min}|)$. Equations (2) and (3) define the set of hyperrectangles that correspond to the lower convex hull of the cloud of points. These hyperrectangles are subdivided in the next phase of the iteration.

The purpose of the balance parameter is to bias `DIRECT` towards unexplored space in the domain. [16]. In Figure 2, the square dot alters the lower convex hull, and the small hyperrectangle which contains the low function value is not potentially optimal. This paper is concerned with the balance parameter, and it's role in the convergence of DIRECT.

Potentially optimal hyperrectangles are subdivided along their long coordinate directions. This strategy ensures equal sampling in every dimension, and is outlined in Table I [16].

Table I. Division of a hyperrectangle R with center c

1:	Let R be a potentially optimal hyperrectangle with center c.
1:	Let ξ be the maximal side length of R.
2:	Let I be the set of coordinate directions corresponding to sides of R with length ξ.
3:	Evaluate the objective function at the points $c \pm \frac{1}{3}\xi e_i$, for all $i \in I$, where e_i is the ith unit vector
4:	Let $w_i = \min \left\{ f(c \pm \frac{1}{3}\xi e_i) \right\}$
5:	Divide the hyperrectangle containing c into thirds along the dimensions in I, starting with the dimension with smallest w_i and continuing to the dimension with the largest w_i.

DIRECT typically terminates when a user-supplied budget of function evaluations is exhausted. Alternative stopping criteria have been used [9]. In [13], an implementation of DIRECT is introduced that relaxes the definition of potentially optimal hyperrectangles. This modification was designed for large parallel computers.

3. The Balance Parameter

In this section we carefully examine the role of the balance parameter. We show that ϵ can affect the convergence of DIRECT with examples and analysis.

3.1. The positive effects of the balance parameter

In [16], the balance parameter was introduced as a way to bias DIRECT globally. The intent of ϵ is to discourage the algorithm from sampling in well-searched areas of the domain.

Different values for ϵ were examined in [16] on a set of popular global optimization test problems [6, 21]. On some problems, convergence was not affected by ϵ. On other problems, the performance of DIRECT

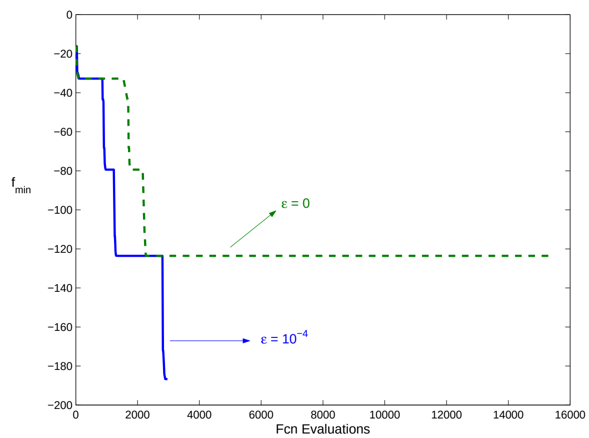

Figure 3. Results for different values of ϵ on the Shubert test problem.

improved for large values of the balance parameter. The recommended value of $\epsilon = 10^{-4}$ was chosen because it produced the most robust results for `DIRECT`. In [8], this recommendation was revised to include a lower bound for the right-hand-side of (3).

In Figure 3, we see the benefits of the balance parameter when `DIRECT` tries to find the global minima of the Shubert test function [16, 21]. When ϵ is set to zero, `DIRECT` fails to find the solution in a reasonable amount of function evaluations.

3.2. The negative effects of the balance parameter

The rest of this paper is concerned with the consequences of using balance parameter. We begin with two examples that illustrate `DIRECT`'s sensitivity to ϵ.

The first example shows that `DIRECT` is affected to additive scaling. For this example, we added 10^6 to the Branin function [6]. In [16], 195 function evaluations were needed by `DIRECT`. For this experiment, a budget of 500 function evaluations is used.

Figure 4 compares `DIRECT` with and without the balance parameter. The figures indicate that ϵ is affecting the ability of `DIRECT` to find a global solution. In Figure 4, sample points cluster around the three global optima when $\epsilon = 0$. When the balance parameter is set to the

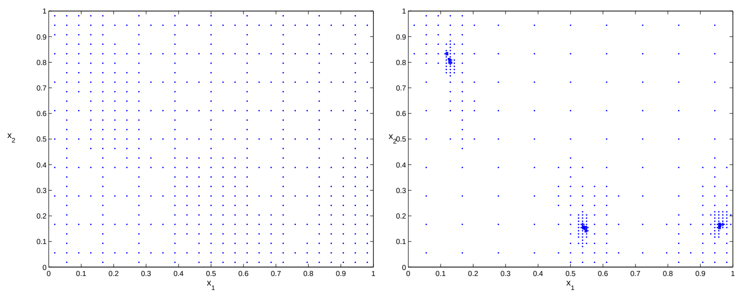

Figure 4. Results for different values of ϵ on the perturbed Branin test problem. The picture on the left shows the sampled points when $\epsilon = 10^{-4}$. The right figure is the points DIRECT samples when $\epsilon = 0$.

recommended value of 10^{-4} from [16], the sample points do not cluster. Note that if $\epsilon = 10^{-4}$, $\|x_{\min} - x^*\| = 0.34$ where $x_{\min}$ is the location of the low function value. When $\epsilon = 0$, $\|x_{\min} - x^*\| = 1.12\text{e-}5$.

The balance parameter also slows the asymptotic convergence of DIRECT. Consider the simple problem

$$\min_{\Omega} \sum_{i=1}^{4} |x_i| + 1, \tag{4}$$

where $\Omega = \left\{x \in \mathbb{R}^4 : -2 \leq x_i \leq 3\right\}$. Figure 5 describes the results when DIRECT is given an exorbitant budget of 100,000 function evaluations. When $\epsilon = 0$, the relative error drops to machine precision, compared to a much larger relative error when $\epsilon = 10^{-4}$.

The behavior of DIRECT in these examples is explained by observing the smallest hyperrectangles. The smallest hyperrectangle after k iterations of DIRECT is always a hypercube, and is a candidate to be potentially optimal whenever the value at the center is $f_{\min}$ (see Figure 2). This candidate is rejected for subdivision if it does not satisfy (3), the condition controlled by the balance parameter. Rejecting the smallest hypercube for subdivision can produce poor convergence, as seen in Figures 4 and 5 In Lemma 1, we describe when rejecting the small hypercube leads to poor performance by DIRECT.

LEMMA 1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a Lipschitz continuous function with Lipschitz constant K, Let $\mathcal{S}$ be the set of hyperrectangles created by*

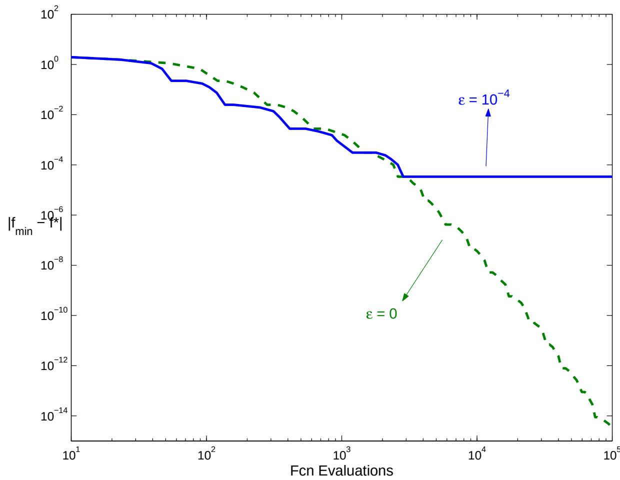

Figure 5. Slow asymptotic convergence on the convex test problem $f(x) = \|x\|_1 + 1$.

DIRECT, and let R be a hypercube with a center c at side length 3^{-l}. Suppose that

(i) $\alpha(R) \leq \alpha(S)$, *for all $S \in \mathcal{S}$ (i.e. R is in the set of smallest hypercubes).*

(ii) $f(c) = f_{\min} \neq 0$ *(i.e. $f(c)$ is the low value found).*

If

$$\alpha(R) < \frac{\epsilon|f(c)|}{6K}\left(\sqrt{n+8} - \sqrt{n}\right), \tag{5}$$

then R will not be potentially optimal until all hyperrectangles in the "neighborhood" of R, i.e. all hyperrectangles whose centers are on the stencil $c \pm 3^{-l}e_i$ for $i = 1, ..., N$ are the same size as R.

Proof. Assume (5) holds. In [9], it was shown that the size of a hyperrectangle, S, with long side of length 3^{-l} is given by

$$\alpha(S) = \frac{3^{-l}}{2}\sqrt{n - 8p/9}, \tag{6}$$

where $0 \leq p \leq n-1$. When S is a hypercube, $p = 0$.

For hypercube R to be potentially optimal there must exist $\tilde{K}$ such that (2) and (3) hold.

From (2), it is clear that

$$\tilde{K} \leq \frac{f(c_S) - f(c)}{\alpha(S) - \alpha(R)}$$

for all $S \in \mathcal{S}$, $\alpha(S) > \alpha(R)$. We define $\tilde{K}$ to be as large as possible; that is, we let

$$\tilde{K} = \min_{S \in \mathcal{S}} \frac{f(c_S) - f(c)}{\alpha(S) - \alpha(R)} = \frac{f(c_{\tilde{S}}) - f(c)}{\alpha(\tilde{S}) - \alpha(R)}, \tag{7}$$

and show that (3) cannot be satisfied.

Inequality (5) is equivalent to

$$\frac{3 \cdot 2K}{\sqrt{n+8} - \sqrt{n}} \leq \frac{\epsilon |f(c)|}{\frac{3^{-l}}{2}\sqrt{n}}. \tag{8}$$

From (6), we see that

$$\alpha(\tilde{S}) - \alpha(R) \geq \frac{3^{-l}}{2}\left(\sqrt{n - 8\tilde{p}/9} - \sqrt{n}\right) \geq \frac{3^{-(l+1)}}{2}\left(\sqrt{n+8} - \sqrt{n}\right). \tag{9}$$

The last inequality is found by setting $\tilde{p} = n - 1$. Therefore,

$$\frac{1}{\alpha(\tilde{S}) - \alpha(R)} \leq \frac{1}{\frac{3^{-(l+1)}}{2}\left(\sqrt{n+8} - \sqrt{n}\right)}. \tag{10}$$

Also note that

$$f(c_{\tilde{S}}) - f(c) = f(c \pm 3^{-l} e_{\tilde{i}}) - f(c) \leq K 3^{-l}. \tag{11}$$

due to the Lipschitz continuity of f. Therefore,

$$\tilde{K} = \frac{f(c_{\tilde{i}}) - f(c)}{d_{\tilde{i}} - d} < \frac{K 3^{-l}}{\frac{3^{-(l+1)}}{2}\left(\sqrt{n+8} - \sqrt{n}\right)} = \frac{3 \cdot 2K}{\sqrt{n+8} - \sqrt{n}}. \tag{12}$$

Combining (12) and (8), we see that

$$\tilde{K} < \frac{\epsilon |f(c)|}{\frac{3^{-l}}{2}\sqrt{n}}. \tag{13}$$

Since $\alpha(R) = \frac{3^{-l}}{2}\sqrt{n}$ and $f(c) = f_{\min}$, it follows that

$$f(c) - \tilde{K} d > f_{\min} - \epsilon |f_{\min}|. \tag{14}$$

Recall that in (7) we chose $\tilde{K}$ as large as possible. Therefore, hypercube R cannot be potentially optimal.

Inequality (5) describes conditions when `DIRECT` will prioritize hyperrectangles near $f_{\min}$ over the hyperrectangle that contains $f_{\min}$. When $\epsilon \neq 0$, additive scaling can make the right hand side of (5) large, which slows the convergence of `DIRECT`, and prevents clustering near optimal solutions. We observed poor behavior due to additive scaling in Figure 4.

When $\frac{|f(c)|}{K} \approx 1$, the right-hand side of (5) is much smaller, and the poor behavior occurs in later iterations, as seen in Figure 5 for problem (4). For the problem described in (4), the smallest hyperrectangle begins being ignored when it reaches a size of 1.7e-5, which is precisely what (5) predicts. Inequality (5) illustrates why `DIRECT` has a slow rate of asymptotic convergence when $\epsilon \neq 0$.

In the next section, we propose a modified version of `DIRECT`, and present numerical results for several different test problems.

4. `DIRECT`-restart

We propose a modified version of `DIRECT` that adaptively updates the value of the balance parameter. Our modification is not sensitive to additive scaling, and is more aggressive with its global search than the original implementation of `DIRECT`.

Our strategy is to start `DIRECT` with $\epsilon = 0$, and let the algorithm search unimpeded by the balance parameter. At the end of each iteration, we check to to see if the progress of the optimization has stagnated. If $f_{\min}$ has not decreased for several iterations, then we shift to the global phase of the algorithm by updating ϵ to $\epsilon_{\max}$. For our results, we set $\epsilon_{\max} = 10^{-2}$. `DIRECT` continues to subdivide, now with the balance skewed towards unexplored regions. If the optimization stagnates once again, we set the balance parameter back to zero, and continue. We repeat until the termination criteria is satisfied. We call our modification `DIRECT`-restart, and summarize it in Algorithm 1.

When $\epsilon = 0$ we define stagnation as five iterations without an improvement of 10^{-4} in $f_{\min}$. In our experience, we found that leaving $\epsilon = \epsilon_{\max}$ produced the best results, so we only reset $\epsilon = 0$ if fifty iterations go by without an improvement of 10^{-4} in $f_{\min}$.

We believe that `DIRECT`-restart improves on the original implementation in two ways. First, it is not sensitive to additive scaling, since the right hand side of (5) zero until ϵ is updated to $\epsilon_{\max}$.

Algorithm 1 DIRECT-restart($f_{\max}$)

Transform Ω to unit hypercube, initialize parameters.
Evaluate f at $c = (1/2, ..., 1/2)$.
Subdivide Ω into $2n$ hyperrectangles and sample f in new centers.
while $f_{\text{count}} < f_{\max}$ **do**
  {*Determine* ϵ *value*}
  if $\epsilon = 0$ **then**
    if stagnation **then**
      Set $\epsilon = \epsilon_{\max}$.
    end if
  else if $\epsilon > 0$ **then**
    if stagnation **then**
      Set $\epsilon = 0$.
    end if
  end if
  Find potentially optimal hyperrectangles.
  Divide potentially optimal hyperrectangles.
  Update f_{count}.
end while

Second, since we separate the local and global phases, we can use a larger value for the balance parameter than recommended in [16, 8]. An aggressive global search should improve convergence on problems with many local minima.

5. Numerical Results

We present a small set of test results that illustrate the robustness of our modification.

Our first test is to compare DIRECT and DIRECT-restart on the nine original test problems from [16], commonly referred to as the *Jones* test set. The nine problems, S5, S7, S10, H3, H6, BR, GP, C6, and SH, are low dimensional ($n \in [2, 6]$), have multiple local and global minima, and cannot be reliably solved by gradient-based methods. More information about these problems is found in [6, 21, 16, 9].

We provide the two algorithms with a budget of function evaluations based on the convergence results in [16]. We use the number of function evaluations reported for convergence in [16] as our budgets. Our results are summarized in Table II.

In Table II, an "X" implies that algorithm found a better solution than the other for a particular test problem. The "Difference" row indi-

Table II. Comparison of `DIRECT` and `DIRECT`-restart on the original Jones test set.

Problem	S5	S7	S10	H3	H6	BR	GP	C6	SH
Budget	154	144	144	198	570	194	190	284	2966
`DIRECT`	X	X	X	X	X	X	X	X	
`DIRECT-restart`		X	X	X	X	X	X	X	X
Difference	0.03	0	0	0	0	0	0	0	0.01

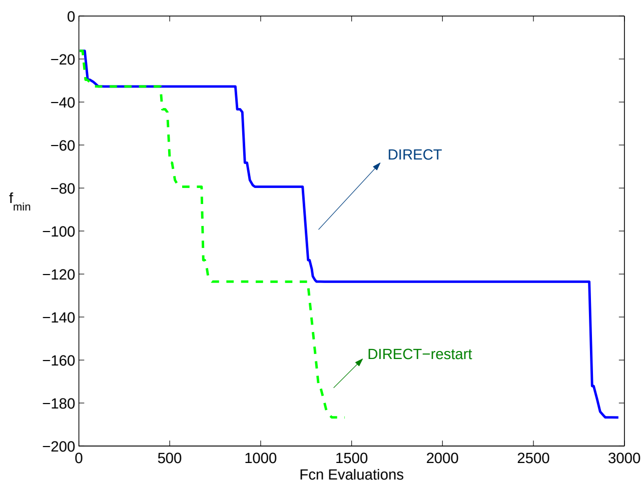

Figure 6. A comparison of `DIRECT` and `DIRECT`-restart on the Shubert test problem.

cates the difference in low function values produced by each algorithm. On most of the problems the algorithms find identical solutions. Thus, our modification does not impede the performance of `DIRECT` on well behaved problems.

On one of the problems, `DIRECT`-restart converges faster than `DIRECT`. Figure 6 compares `DIRECT` and `DIRECT`-restart on the "SH" (Shubert) problem. Since `DIRECT`-restart uses a large $\epsilon_{\max}$, it does not get stuck in local solutions, and finds the global solution in significantly fewer function evaluations than `DIRECT` with $\epsilon = 10^{-4}$.

In the next test, we additively perturb the Jones set by 100,000. We report comparison results in Table III.

Table III. Comparison of DIRECT and DIRECT-restart on the additively perturbed Jones test set.

Problem	S5	S7	S10	H3	H6	BR	GP	C6	SH
Budget	154	144	144	198	570	194	190	284	2966
DIRECT									
DIRECT-restart	X	X	X	X	X	X	X	X	X
Difference	8.52	7.56	7.59	0.14	1.28	0.05	0.01	0.04	0.04

The results indicate that adaptively updating ϵ can improve the performance of DIRECT on additively perturbed problems. Table IV compares the solution accuracy for both implementations on the additively perturbed Jones problems.

Table IV. Solution accuracy on the additively perturbed Jones test set.

Problem	S5	S7	S10	H3	H6	BR	GP	C6	SH
Budget	154	144	144	198	570	194	190	284	2966
					$\|x_{\min} - x^*\|$				
DIRECT	8.66	8.66	8.66	0.09	1.00	0.34	4.11e-3	0.09	4.36e-3
DIRECT-restart	0.02	2.7e-3	2.7e-3	0.02	3.7e-3	1.6e-3	4.57e-4	9.5e-4	2.49e-6

The figures in Table III illustrate the lack of clustering caused by a fixed balance parameter. DIRECT-restart clusters near optima, and finds better solution with the same budget.

6. Conclusion

The DIRECT algorithm aggressively searches for global minima by using a balance parameter to bias the search towards unexplored regions of the domain. We show the algorithm is sensitive to additive scaling, and has slow asymptotic convergence. We quantify the sensitivity of DIRECT in Lemma 1.

DIRECT-restart is a modification to DIRECT [16] that tunes the balance parameter during the optimization process. Test results show that our modification can improve convergence speed, remove sensitivity to additive scaling, and improve clustering of sample points near optimal solutions.

7. Acknowledgments

The authors would like to thank Evin Cramer for her insight into DIRECT, and Joerg Gablonsky for his suggestions in developing the software used in this paper.

References

1. Audet, C. and J. E. Dennis: 2004, 'Mesh adaptive direct search algorithms for constrained optimization'. Technical Report TR04-02, Department of Computational and Applied Mathematics, Rice Univeristy.
2. Audet, C. and J. D. Jr.: 2003, 'Analysis of Generalized Pattern Searches'. *SIAM J. Optim.* **13**, 889–903.
3. Bartholomew-Biggs, M., S. Parkhurst, and S. Wilson: 2002, 'Using DIRECT to Solve an Aircraft Routing Problem'. *Computational Optimization and Applications* **21**, 311 – 323.
4. Carter, R., J. Gablonsky, A. Patrick, C. Kelley, and O.J.Eslinger: 2001, 'Algorithms for Nosiy Problems in Gas Transmission Pipeline Optimization'. *Optimization and Engineering* **2**(2), 139–157.
5. Cramer, E. and M. Carter: 1997, 'Global Optimization of a Multi-element Airfoil'. In: *Collection of Technical Papers, Pt. 2, AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit, and AIAA/ASME/AHS Adaptive Structures Forum*. Kissimmee, FL, pp. 5 – 39.
6. Dixon, L. and G. Szego: 1978, *Towards Global Optimisation 2*. New York, NY: North-Holland, first edition.
7. Finkel, D. and C. Kelley: 2004, 'Nonsmooth Convergence Analysis of DIRECT'. Technical Report CRSC-TR04-28, Center for Research in Scientific Computation, North Carolina State University.
8. Floudas, C. and P. Pardalos (eds.): 2001, *Encyclopedia of Optimization*, Chapt. The DIRECT Global Optimization Algorithm, pp. 431–440, The Encyclopedia of Optimization. Kluwer Academic.
9. Gablonsky, J.: 2001a, 'Modifications of the Direct Algorithm'. Ph.D. thesis, North Carolina State University.
10. Gablonsky, J. and C. Kelley: 2001, 'A Locally-Biased form of the DIRECT Algorithm'. *Journal of Global Optimization* **21**, 27–37.
11. Gablonsky, J. M.: 2001b, 'DIRECT Version 2.0 User Guide'. Technical Report CRSC-TR01-08, Center for Research in Scientific Computation, North Carolina State University.
12. Gilmore, P. and C. T. Kelley: 1995, 'An implicit filtering algorithm for optimization of functions with many local minima'. *SIAM J. Optim.* **5**, 269–285.
13. He, J., L. Watson, N. Ramakrishnan, C. Shaffer, A. Verstak, J. Jian, K. Bae, K. Bae, and W. Tranter: 2002, 'Dynamic Data Structures for a Direct Search Algorithm'. *Computational Optimization and Applications* **23**(1), 5–25.
14. Hooke, R. and T. Jeeves: 1961, 'Direct search solution of numerical and statistical problems'. *Journal of the Association for Computing Machinery* **8**, 212–229.

15. Hough, P. D., T. G. Kolda, and V. J. Torczon: 2001, 'Asynchronous Parallel Pattern Search for Nonlinear Optimization'. *SIAM J. Sci. Comput.* **23**, 134–156.
16. Jones, D., C. Perttunen, and B. Stuckman: 1993, 'Lipschitzian Optimization Without the Lipschitz Constant'. *Journal of Optimization Theory and Application* **79**(1), 157–181.
17. Kolda, T., R. Lewis, and V. Torczon: 2003, 'Optimization by Direct Search: New Perspective on some classical and modern methods'. *SIAM Review* **45**(3), 385–482.
18. Ljunberg, K. and S. Holmgren: 2004, 'Simultaneous Search for multiple QTL using the global optimization algorithm DIRECT'. *Bioinformatics* **20**(12), 1887–1895.
19. Nelder, J. and R. Mead: 1965, 'A simplex method for function minimization'. *Comput. J.* **7**, 308–313.
20. Sasena, M., P. Papalambros, and P. Goovaerts: 2002, 'Global Optimization of Problems with Disconnected Feasible Regions via Surrogate Modeling'. In: *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*. Atlanta, GA.
21. Yao, Y.: 1989, 'Dynamic Tunneling Algorithm for Global Optimization'. *IEEE Transactions on System, Man and Cybernetics* **19**, 1222 – 1230.