J Glob Optim (2006) 36:597–608
DOI 10.1007/s10898-006-9029-9

ORIGINAL ARTICLE

Additive scaling and the DIRECT algorithm

D. E. Finkel · C. T. Kelley

Received: 20 March 2006 / Accepted: 21 March 2006 /
Published online: 27 June 2006

Abstract In this paper we show that the convergence behavior of the DIviding RECTangles (DIRECT) algorithm is sensitive to additive scaling of the objective function. We illustrate this problem with a computation and show how the algorithm can be modified to eliminate this sensitivity.

Keywords DIRECT · Global optimization · Additive scaling

1 Introduction

DIviding RECTangles (DIRECT) [16] is an optimization algorithm designed to 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 this sensitivity may lead to slow asymptotic convergence. Examples, which illustrate this effect can be generated easily, as we do in Sect. 3.2, by adding a large positive constant to a function that DIRECT would otherwise optimize very well. We suggest a modification to the algorithm, and present test results that illustrate the effectiveness of our modification.

D. E. Finkel
Advanced Concepts and Technologies, MIT Lincoln Laboratory, 244 Wood St, Lexington, MA 02420-9108, USA
e-mail: dfinkel@ll.mit.edu

C. T. Kelley (✉)
Center for Research in Scientific Computation and Department of Mathematics, North Carolina State University, Box 8205, Raleigh, NC 27695-8205, USA
e-mail: tim_Kelley@ncsu.edu

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

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

with $\Omega = \{x \in \mathbb{R}^n : l_i \le x_i \le u_i, \quad i = 1 \ldots n\}$, 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 [4, 17]. The simulators can have nonsmooth functions built into them (e.g. **IF** statements, **max** functions and table lookups), or may 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.

The `DIRECT` algorithm can be effective [2, 3, 9, 16, 18, 20] in finding the basin of convergence for a global solution on low-dimensional problems. Unlike other deterministic sampling methods [1, 5, 12, 14, 15, 19], `DIRECT` does not use a pattern or stencil in its search, nor does it approximate gradient information, even indirectly.

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. A parameter is used to protect the iteration against excessive local bias in the search. In this paper we show that one effect of the use of this parameter is a degradation in the performance of the algorithm if the function to be minimized is poorly scaled in the additive sense, i.e., the maximum and minimum values are both large relative to the size of the bounding hypercube.

The `DIRECT` algorithm 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 [2, 18, 20]. `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 Sect. 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 $\mathcal{T}\colon\mathbb{R}^n \to \mathbb{R}^n$

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

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 this paper that

$$\Omega = \left\{x \in \mathbb{R}^n\colon 0 \le x_i \le 1, \quad i = 1 \ldots n\right\}.$$

2.1 The first iteration

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

The DIRECT algorithm begins the division process by evaluating f at neighbors of c in every dimension. The neighbors are determined to be the points two thirds of the way 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 shows 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].

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

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

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

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

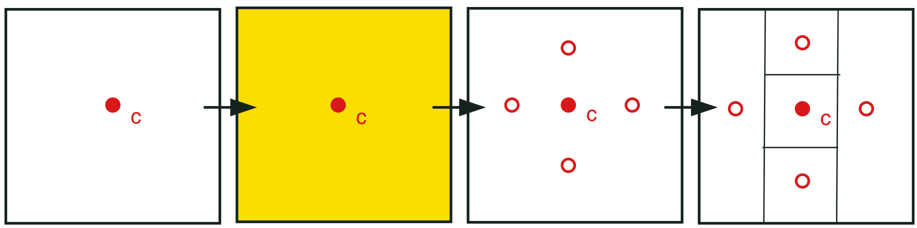

Fig. 1 The first iteration of DIRECT

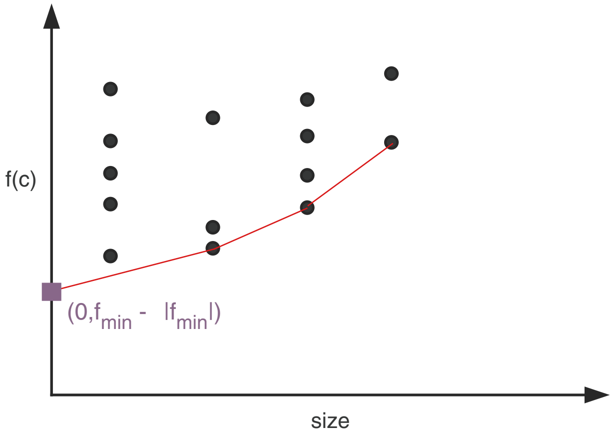

Fig. 2 Geometric interpretation of (2) and (3)

convex hull of the cloud of points. These hyperrectangles are subdivided in the next phase of the iteration.

The purpose of the parameter ϵ is to guard the `DIRECT` iteration against excessive emphasis on local search [16]. In Fig. 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 this parameter, and it is 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 1 [16].

The `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 parameter ϵ

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

Table 1 Division of a hyperrectangle R with center c

1	Let R be a potentially optimal hyperrectangle with center c.
2	Let ξ be the maximal side length of R.
3	Let I be the set of coordinate directions corresponding to sides of R with length ξ.
4	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
5	Let $w_i = \min \left\{ f(c \pm \frac{1}{3}\xi e_i) \right\}$
6	Divide the hyperrectangle containing c into thirds along the dimensions in I, starting with the dimension with lowest w_i and continuing to the dimension with the highest w_i

3.1 The role of ϵ

In [16], the parameter ϵ was introduced as a way to guard against excessive local search.

Different values for ϵ were examined in [16] on a set of popular global optimization test problems [6, 21]. On most problems, convergence was not affected by ϵ. In a few cases, the performance of DIRECT improved for large values of the 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 Fig. 3, we see the benefits of the 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 Additive scaling and its interaction with the parameter ϵ

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

The first example shows that DIRECT is affected by 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 parameter ϵ. The figures indicate that ϵ is affecting the ability of DIRECT to find a global solution. In Fig. 4, sample points cluster around the three global optima when $\epsilon = 0$. When the parameter is set to the recommended value of 10^{-4} from [16], the sample points do not cluster. Note that if $\epsilon = 10^{-4}$, the distance between the best point found by DIRECT and the true optimum is 0.34; when $\epsilon = 0$, that distance is 1.12e-5.

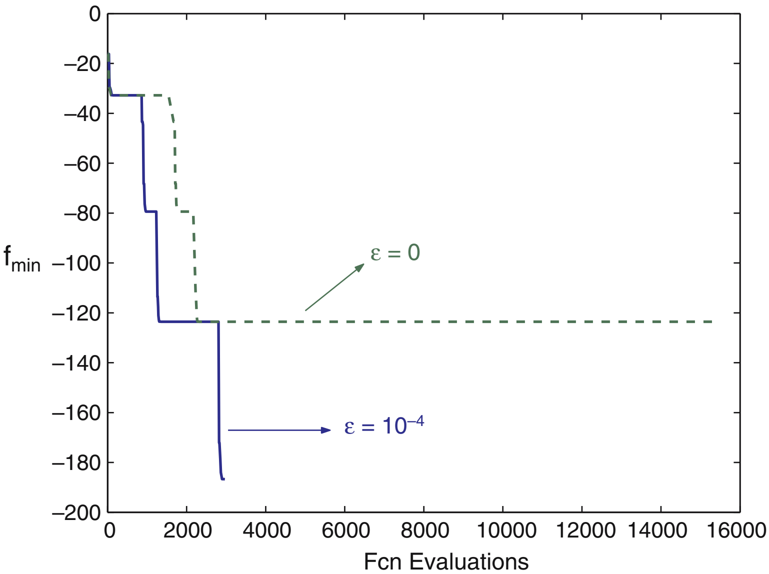

Fig. 3 Results for different values of ϵ on the Shubert test problem

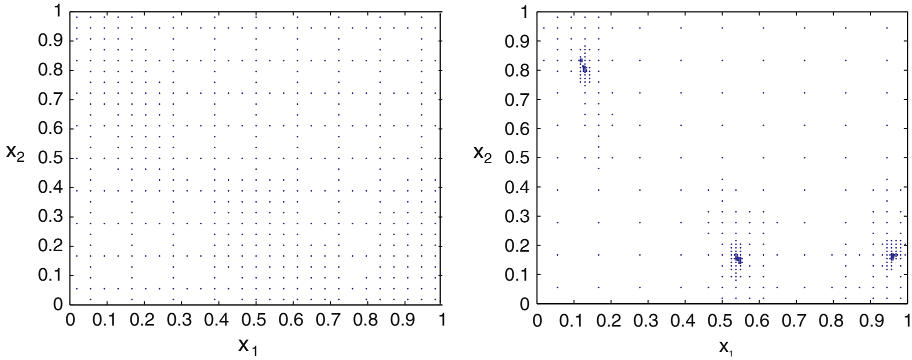

Fig. 4 Results for different values of ϵ on the perturbed Branin test problem. The picture on the left shows the sampled points when $\epsilon = 10^{-4}$. The right figure is the points DIRECT samples when $\epsilon = 0$

As a second example, consider the simple problem

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

where $\Omega = \{x \in \mathbb{R}^4 : -2 \leq x_i \leq 3\}$. We will see that the standard value $\epsilon = 10^{-4}$ can slow the asymptotic convergence of DIRECT. 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 $\epsilon = 10^{-4}$. Keep in mind that simply setting $\epsilon = 0$ is not a solution to this problem. While there is no sensitivity to additive scaling when $\epsilon = 0$, the problem of excessive local search will resurface.

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, Fig. 2). This candidate is rejected for subdivision if it does not satisfy (3), the condition controlled by the parameter ϵ. Rejecting the smallest hypercube for subdivision can produce poor convergence, as seen in Figs. 4 and 5. In Theorems 3.1 and 3.2, we describe how rejecting the small hypercube leads to poor performance by DIRECT.

Theorem 3.1 shows how poor additive scaling can cause the algorithm to perform poorly as the size of the hypercube which contains the optimal point becomes small. Theorem 3.2, which was suggested by a clever referee of this paper, uses similar reasoning to show that if the additive scaling is particularly bad, that a hypercube, which contains the current minimal point will not become potentially optimal until all larger hyperrectangles are divided, i.e. DIRECT becomes an exhaustive grid search.

Theorem 3.1 *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* DIRECT*, and let* R *be a hypercube with a center* c *and side length* 3^{-l}*. Suppose that*

(i) $\alpha(R) \leq \alpha(T)$, *for all* $T \in \mathcal{S}$ *(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).*

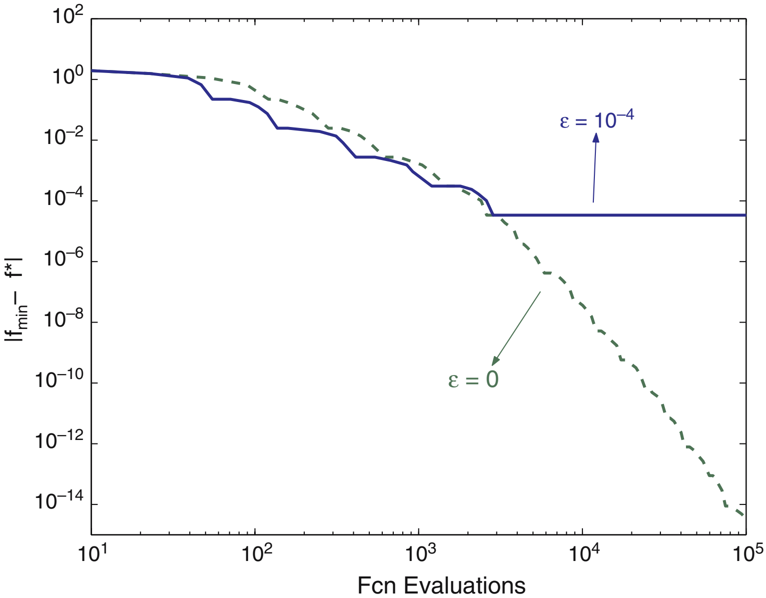

Fig. 5 Slow asymptotic convergence on the convex test problem $f(x) = \|x\|_1 + 1$

If

$$\alpha(R) < \frac{\epsilon|f(c)|}{2K}\left(\sqrt{n+8} - \sqrt{n}\right) \tag{5}$$

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 $i = 1, \ldots, N$ are the same size as R.

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

From (2), it is clear that

$$\tilde{K} \leq \frac{f(c_T) - f(c)}{\alpha(T) - \alpha(R)}$$

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

$$\tilde{K} = \min_{T \in \mathcal{S}} \frac{f(c_T) - f(c)}{\alpha(T) - \alpha(R)} = \frac{f(c_{\tilde{T}}) - f(c)}{\alpha(\tilde{T}) - \alpha(R)} \tag{6}$$

and show that (3) cannot be satisfied.

Since $\alpha(R) = 3^{-l}\sqrt{n}/2$, inequality (5) is equivalent to

$$\frac{2K}{\sqrt{n+8} - \sqrt{n}} < \frac{\epsilon|f(c)|}{\frac{3^{-l}}{2}\sqrt{n}}. \tag{7}$$

The smallest $T \in \mathcal{S}$ with $\alpha(T) > \alpha(R)$ will have one side of length 3^{-l+1} and $n-1$ sides of length 3^{-l}, such a hyperrectangle will have size

$$\alpha(T) = \sqrt{(n-1)\left(\frac{3^{-l}}{2}\right)^2 + \left(\frac{3^{-l+1}}{2}\right)^2} = \frac{3^{-l}}{2}\sqrt{n+8}.$$

Hence,

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

The Lipschitz continuity of f implies that

$$f(c_{\tilde{T}}) - f(c) = f(c \pm 3^{-l} e_{\tilde{i}}) - f(c) \leq K3^{-l}. \tag{9}$$

Therefore,

$$\tilde{K} = \frac{f(c_{\tilde{T}}) - f(c)}{\alpha(\tilde{T}) - \alpha(R)} \leq \frac{K3^{-l}}{\frac{3^{-l}}{2}\left(\sqrt{n+8} - \sqrt{n}\right)} = \frac{2K}{\sqrt{n+8} - \sqrt{n}}. \tag{10}$$

Combining (10) and (7), we see that

$$\tilde{K} < \frac{\epsilon |f(c)|}{\frac{3^{-l}}{2}\sqrt{n}}. \tag{11}$$

From $f(c) = f_{\min}$, it follows that

$$f(c) - \tilde{K}\alpha(R) > f_{\min} - \epsilon |f_{\min}|. \tag{12}$$

Recall that in (6) we chose $\tilde{K}$ as large as possible. Therefore, hypercube R cannot be potentially optimal.

If $\epsilon > 0$ and the global minimum of f is nonzero, then inequality (5) will hold when the search has progressed to the point of centering the global minimizer in a sufficiently small hyperrectangle. With poor additive scaling, $|f(c)|$ in Eq. (5) will be higher, thereby increasing the right side of the inequality (5). Hence the condition for the theorem will be met earlier in the search, which will prevent `DIRECT` from dividing the rectangle with the best point, and thereby prevent clustering near optimal solutions. We observed this effect in Fig. 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 Fig. 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 $\epsilon \neq 0$.

Theorem 3.2 describes the worst case of additive scaling. If (13) holds, then R will not be potentially optimal if there are any larger hyperrectangles at all. One can clearly add a constant to f to make (13) hold, so the potential for this worst case is always present.

Theorem 3.2 *Let f have Lipschitz constant K. Let*

$$f^* = \min_{x \in \Omega} f(x).$$

Let $\mathcal{S}$, R, and c be as in Theorem 3.1. *If*

$$f^* > \frac{K\sqrt{n}}{\epsilon(\sqrt{1+8/n}-1)} \tag{13}$$

then R is not potentially optimal if any $T \in \mathcal{S}$ is larger than R.

Proof We assume that (13) holds. Hence $f > 0$ and $|f| = f$. Assume that there is at least one $T \in \mathcal{S}$ with $\alpha(T) > \alpha(R)$. Let $\tilde{T} \in \mathcal{S}$ be such that

$$f(c_{\tilde{T}}) = \min_{T\in\mathcal{S},\alpha(T)>\alpha(R)} f(c_T).$$

R will not be potentially optimal if

$$f(c) - \frac{\alpha(R)}{\alpha(\tilde{T})}(f(c_{\tilde{T}} - (1-\epsilon)f(c)) > (1-\epsilon)f(c), \tag{14}$$

which is equivalent to

$$f(c) > \frac{f(c_{\tilde{T}} - f(c))}{\epsilon(\alpha(\tilde{T})/\alpha(R) - 1)}. \tag{15}$$

Clearly $f(c) \geq f^*$. We can estimate the right side of (15) from above by noting that the diameter of Ω is $\sqrt{n}$ and so

$$f(c_{\tilde{T}}) - f(c) \leq K\sqrt{n}.$$

As in the proof of Theorem 3.1,

$$\alpha(\tilde{T})/\alpha(R) \geq \frac{\sqrt{n+8}}{\sqrt{n}} = \sqrt{1+8/n}.$$

Hence (13) implies (15), which completes the proof.

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

4 Modified DIRECT

We seek a modification to `DIRECT` that is easy to implement, improves performance on poorly scaled (additively) problems, and adds no new parameters.

Similar to [8], our modification is a simple update to the definition of potentially optimal hyperrectangles. After each iteration, we scale the function values by subtracting the median of the collected function values. The result is an update to (3):

$$f(c_R) - \hat{K}\alpha(R) \leq f_{\min} - \epsilon|f_{\min} - f_{\text{median}}|. \tag{16}$$

In [8], (3) was updated to introduce a lower bound for the influence of the balance parameter. We have modified `DIRECT` so that the balance parameter's influence is reduced. We experimented with several different scaling values (e.g. the average of collected function values, the maximum, and the first function value), but encountered performance problems with each. For example, additively scaling all function values by $f_{\max}$ lead to poor performance on problems whose function values spanned multiple orders of magnitude. Similar problems occured when we tried other scaling values.

Table 2 Comparison of DIRECT and modified DIRECT: unperturbed problems

Problem	S5	S7	S10	H3	H6	BR	GP	C6	SH
DIRECT: $\epsilon = 10^{-4}$	155	145	145	199	571	195	191	285	2,967
DIRECT: $\epsilon = 0$	179	145	145	199	571	195	191	285	Fail
Modified DIRECT - median	155	145	145	199	571	259	191	285	3,663

Table 3 Comparison of DIRECT and modified DIRECT on test problems additively perturbed by 100,000

Problem	S5	S7	S10	H3	H6	BR	GP	C6	SH
DIRECT: $\epsilon = 10^{-4}$	Fail	Fail	Fail	Fail	Fail	Fail	16,135	135,969	57,093
DIRECT: $\epsilon = 0$	187	145	145	199	Fail	195	16,135	Fail	Fail
Modified DIRECT - median	155	145	145	199	571	259	191	285	3,663

Our test results indicate that (16) does not impact performance on well-behaved problems, and can improve convergence speed on problems with poor additive scaling.

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

Our first test is to compare DIRECT and the modification on the nine original on the nine original test problems from [16]. 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, 9, 16, 21].

We perform the same test as done in [16]; that is, each algorithm is terminated when

$$\frac{f_{\min} - f^*}{|f^*|} \leq 10^{-4},$$

where $f_{\min}$ is function value at the best point found by the algorithm. Our results are summarized in Table 2. We tabulate the number of function evaluations needed for the two algorithms to terminate, and included results for the simple modification of updating $\epsilon = 0$. We report the test as a failure if it took more than $200,000$ function evaluations.

On seven of nine original test problems, our modification does not affect convergence. On the Shubert test problem our modification weakened the good effects of the parameter ϵ, but did much better than simply setting $\epsilon = 0$. The benefits of the modified approach are seen clearly if the problems are poorly additively scaled.

In the next test, we additively perturb the problems in the test set by 100,000, and again compare the performance of our modification to the original DIRECT. We use the same termination criteria used in Table 2; that is, we terminate when

$$\frac{f_{\min} - f^*}{|f^* - 100,000|} \leq 10^{-4}. \tag{17}$$

Our results are shown in Table 3. Once again, a failure occurs when the problems takes more than 200,000 functions to satisfy (17).

The results indicate that our modification can improve the performance of DIRECT on additively perturbed problems. Points sampled by our modification cluster near

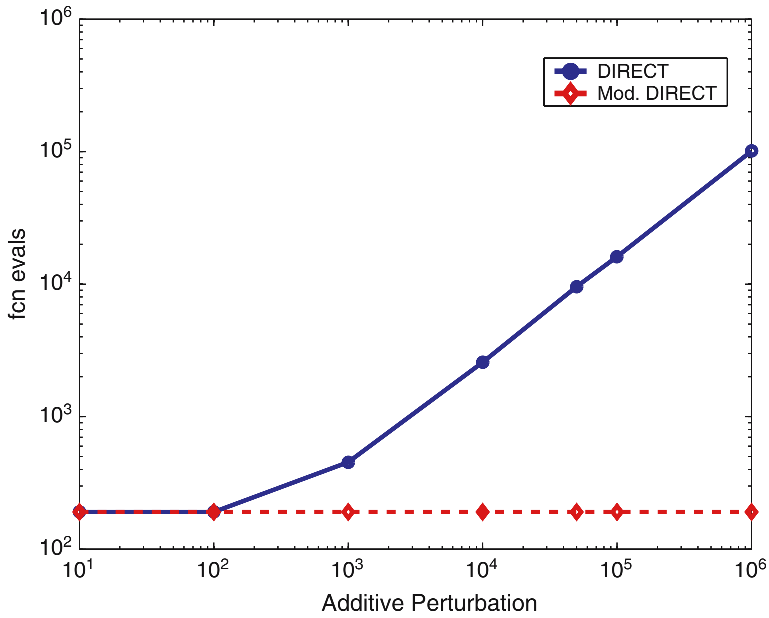

Fig. 6 Function evaluations required for convergence on the Goldstein-Price problem vs additive perturbation

optima independently of the objective function scale. Once again, our modification weakens DIRECT on the Shubert problem, but is better than simply setting $\epsilon = 0$.

Lastly, we present the impact of additive scaling on one test problem, the Goldstein-Price problem. In Fig. 6, we compare DIRECT and our modified implementation as the additive perturbation is increased. We use (17) to terminate both algorithms. Our modification is not affected by the perturbations, while convergence of the original implementation of DIRECT requires more function evaluations as the perturbation grows.

5 Conclusion

The DIRECT algorithm uses a parameter ϵ to guard against excessive local bias in the search. We show the wat this parameter is used makes DIRECT sensitive to additive scaling and, as a result DIRECT may have slow asymptotic convergence on poorly scaled problems. We quantify the sensitivity of DIRECT in Theorem 3.1.

We propose a modification to DIRECT [16] that addresses this problem. Test results show that our modification can remove sensitivity to additive scaling and improve clustering of sample points near optimal solutions.

Acknowledgements The authors would like to thank Evin Kramer for her insight into DIRECT, and Joerg Gablonsky for his suggestions in developing the software used in this paper. The authors are also grateful to an anonymous referee for suggesting that we could simplify the original version of our modified algorithm, and for suggesting Theorem 3.2. This research was supported in part by National Science Foundation grants DMS-0070641, DMS-0112542, and DMS-0209695. Version of March 17, 2006.

References

1. Audet, C., Dennis, J.E.: Mesh adaptive direct search algorithms for constrained optimization. Technical Report TR04-02, Department of Computational and Applied Mathematics, Rice Univeristy (2004)
2. Audet, C., Dennis, Jr., J.E.: Analysis of generalized pattern searches. SIAM J. Optim. **13**, 889–903 (2003)
3. Bartholomew-Biggs, M., Parkhurst, S., Wilson, S.: Using DIRECT to solve an aircraft routing problem. Comput. Optim. Appl. **21**, 311–323 (2002)
4. Carter, R., Gablonsky, J., Patrick, A., Kelley, C., Eslinger, O.J.: Algorithms for nosiy problems in gas transmission pipeline optimization. Optim. Eng. **2**(2), 139–157 (2001)
5. Cramer, E., Carter, M.: 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. pp. 5–39 Kissimmee, FL (1997)
6. Dixon, L., Szego, G.: Towards global optimisation 2, 1 edn. North-Holland, New York, NY, (1978)
7. Finkel, D., Kelley, C.: Nonsmooth convergence analysis of DIRECT. Technical Report CRSC-TR04-28, Center for Research in Scientific Computation, North Carolina State University (2004)
8. Floudas, C., Pardalos, P. (eds.): Encyclopedia of Optimization, Chapter The DIRECT Global Optimization Algorithm, pp. 431–440. The Encyclopedia of Optimization. Kluwer Academic Dordrecht (2001)
9. Gablonsky, J.: Modifications of the direct algorithm. Ph.D. thesis, North Carolina State University (2001a)
10. Gablonsky, J., Kelley, C.: A locally-biased form of the DIRECT algorithm. J. Global Optim. **21**, 27–37 (2001)
11. Gablonsky, J.M.: DIRECT version 2.0 user guide. Technical Report CRSC-TR01-08, Center for Research in Scientific Computation, North Carolina State University (2001b)
12. Gilmore, P., Kelley, C.T.: An implicit filtering algorithm for optimization of functions with many local minima. SIAM J. Optim. **5**, 269–285 (1995)
13. He, J., Watson, L., Ramakrishnan, N., Shaffer, C., Verstak, A., Jian, J., Bae, K., Bae, K., Tranter, W.: Dynamic data structures for a direct search algorithm. Comput. Optim. Appl. **23**(1), 5–25 (2002)
14. Hooke, R., Jeeves, T.: Direct search solution of numerical and statistical problems. J. Assoc. Comput. Mach. **8**, 212–229 (1961)
15. Hough, P.D., Kolda, T.G., Torczon, V.J.: Asynchronous parallel pattern search for nonlinear optimization. SIAM J. Sci. Comput. **23**, 134–156 (2001)
16. Jones, D., Perttunen, C., Stuckman, B.: Lipschitzian optimization without the Lipschitz constant. J. Optim. Theor. Appl. **79**(1), 157–181 (1993)
17. Kolda, T., Lewis, R., Torczon, V.: Optimization by direct search: new perspective on some classical and modern methods. SIAM Rev. **45**(3), 385–482 (2003)
18. Ljunberg, K., Holmgren, S.: Simultaneous search for multiple QTL using the global optimization algorithm DIRECT. Bioinformatics **20**(12), 1887–1895 (2004)
19. Nelder, J., Mead, R.: A simplex method for function minimization. Comput. J. **7**, 308–313 (1965)
20. Sasena, M., Papalambros, P., Goovaerts, P.: Global optimization of problems with disconnected feasible Regions via Surrogate Modeling. In: 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization. Atlanta, GA (2002)
21. Yao, Y.: Dynamic Tunneling algorithm for global optimization. IEEE Trans. Syst. Man Cybern. **19**, 1222–1230 (1989)