Project number IST-25582

CGL

Computational Geometric Learning

Convergence Rate of the DIRECT Algorithm

STREP

Information Society Technologies

Period covered:	November 1, 2011–October 31, 2012
Date of preparation:	October 22, 2012
Date of revision:	October 22, 2012
Start date of project:	November 1, 2010
Duration:	3 years
Project coordinator name:	Joachim Giesen (FSU)
Project coordinator organisation:	Friedrich-Schiller-Universität Jena Jena, Germany

Convergence Rate of the DIRECT Algorithm

Yves Brise Bernd Gärtner

October 22, 2012

Abstract

The DIRECT algorithm is a deterministic sampling method for Lipschitz continuous function optimization. We give the first quantitative analysis of the convergence rate of DIRECT. We show that, in $\mathbb{R}^d$, $O(d^{2d+2}\frac{1}{\epsilon^{2d}})$ function evaluations are sufficient to guarantee an absolute error in function value not larger than ϵ. Dropping Lipschitz continuity the same bound applies to the sampling distance from the location of the global minimum, with an even better hidden constant.

1 Introduction

The DIRECT algorithm was introduced in 1993 by Jones, Perttunen, and Stuckman [11]. It is a global optimization algorithm for the setup described above. Its aim is to provide an efficient optimization framework for problems where very little might be known about the objective function. In particular, being an enhancement of Lipschitzian optimization (e.g. see Shuberts algorithm [12]) we do not need derivative information. The only thing that we have to be able to do is to evaluate the function at a given point.

2 Basics & DIRECT Algorithm

Let us look at the problem we are confronted with. We are considering a bound constrained function optimization which means that the objective domain has simple bounds, i.e. every variable of the function has an upper and a lower bound. function only needs to be Lipschitz continuous in the vicinity of the minimum, we will assume this kind of continuity over the whole domain, except where mentioned otherwise.

The problem can be formalized as follows,

$$\min_{x \in \Delta_d} f(x) \tag{1}$$

where $\Delta_d = \{x \in \mathbb{R}^d \mid x_i \in [0,1] \; \forall i \in [d]\}$ is the unit cube and $f : \Delta_d \to \mathbb{R}$ is a Lipschitz continuous function.

A Lipschitz continuous function f with Liptschitz constant K satisfies the following condition,

$$|f(x) - f(y)| \leq K|x - y|, \; \forall x, y \in \Delta_d.$$

Note that any simple bound constraints can be normalized to meet the form above, with a possible change of the Lipschitz constant though. For convenience we would like to deal with the unit cube only.

We define

$$\begin{aligned} f_{\min} &:= \min_{x \in \Delta_d} f(x), \\ X_{\min} &:= \{x \in \Delta_d \,|\, f(x) = f_{\min}\}. \end{aligned}$$

$X_{\min}$ is the set of points at which the function f attains its minimum. Obviously, $f(x_{\min}) = f_{\min}$, for all $x_{\min} \in X_{\min}$.

In the following we will describe how the DIRECT algorithm works. A more comprehensive description can be found in the original paper [11].

The DIRECT algorithm runs in rounds and maintains a set of evaluation points, which are located at the centers of a set of hyperrectangles. We denote the set of hyperrectanlges by $\mathcal{R}$, or more specifically $\mathcal{R}^{(k)}$ if we refer to the set after round k. The hyperrectangles in $\mathcal{R}$ partition Δ_d at all times. The algorithm begins with $\mathcal{R}^{(0)} = \{\Delta_d\}$. In every round, it subdivides at least one of the rectangles.

Any hyperrectangle $R \in \mathcal{R}$ will have at most two different side lengths, namely $3^{-\ell}$ and $3^{-\ell-1}$ for some number $\ell \in \mathbb{N}_0$ (or just $3^{-\ell}$ if it is a cube). Let c_R be the center of R. When R is subdivided, this is done only along the long sides. Let $e_1, \ldots, e_s$ be the unit vectors parallel to the long sides, for some $1 \leq s \leq d$. The next step is to evaluate f at every point $c_R \pm 3^{-\ell-1} e_i$, for every $1 \leq i \leq s$. These are $2s$ new function evaluations in total. We determine the index i for which the values $c_R \pm 3^{-\ell-1} e_i$ contain the smallest value among all the newly evaluated points. Ties are broken arbitrarily. Then we slice R along that direction into three congruent rectangles. That means cutting it by hyperplanes normal to e_i through the points $c_R + \frac{1}{2} 3^{-\ell-1} e_i$ and $c_R - \frac{1}{2} 3^{-\ell-1} e_i$. Note that, conveniently, the two points $c_R \pm 3^{-\ell-1} e_i$ are the centers of the two outer rectangles, while c_R is still the center of the middle rectangle R'. Furthermore, we can continue in the same fashion and cut R' along the direction that contains the smallest function

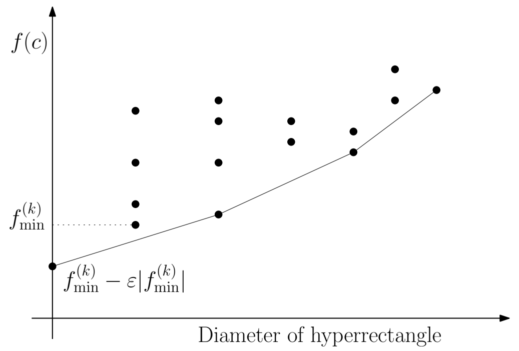

Figure 1: Diagram to determine potentially optimal hyperrectangles. Every rectangle corresponds to a point. One coordinate indicates the size of the rectangle, i.e. half the diameter, and the other coordinate indicates the function value at its center, i.e. $f(c)$.

value among all the remaining directions. This way we cut R into $2s + 1$ smaller rectangles of various sizes. However, all rectangles still maintain the property that the have at most two different side lengths.

The second important part about DIRECT is, how, in each round, to find the hyperrectangles that we should subdivide by the method just described. This is done according to the following definition.

Definition 2.1 ([11]). *A hyperrectangle $R \in \mathcal{R}^{(k)}$ is called* potentially optimal *in round $k + 1$ if there exists some constant $K > 0$ such that the following two conditions hold,*

(i) $f(c_R) - K\frac{diam(R)}{2} \leq f(c_{R'}) - K\frac{diam(R')}{2},\ \forall R' \in \mathcal{R}^{(k)},$

(ii) $f(c_R) - K\frac{diam(R)}{2} \leq f_{\min}^{(k)} - \varepsilon|f_{\min}^{(k)}|,$

where $diam(R)$ is the Euclidian diameter of R, and c_R is the center of R.

In every round we subdivide all potentially optimal rectangles. Condition (i) says that we should choose hyperrectangles that have the greatest potential to find a large decrease in function value. This is done by requiring that there be a (hypothetical) Lipschitz constant K, for which some rectangle R may contain smaller function values than all the rectangles that are larger than R (K must not be too large). A similar thing should hold for all rectangles smaller than R (K must not be too small). Obviously, it is also important that $f(c_R)$ is small too. We can conveniently identify those rectangles by plotting them in a simple two dimensional graph (see Figure 1). Every rectangle $R \in \mathcal{R}$ is plotted as the point $(\frac{diam(R)}{2}, f(c_R))$. Then, identifying the points conforming to condition (i) simply consists of identifying the points that lie on the lower convex hull of the set of all points.

Condition (ii) states that the potential improvement in R should be at least some ε-fraction of the current minimum $f_{\min}^{(k)}$. Excluding the hyperrectangles that do not comply with this condition can easily be achieved by additionally considering the point $(0, f_{\min}^{(k)} - \varepsilon|f_{\min}^{(k)}|)$ in the plot of Figure 1. This condition was introduced to guard against too much local search (subdivision of small rectangles) in the algorithm.

That concludes the description of the algorithm. Let us now have a closer look at some of the properties of the hyperrectangles. Recall that some rectangle $R \in \mathcal{R}$ has at most two different side lengths, say $3^{-\ell}$ and $3^{-\ell-1}$. We define ℓ to be the *level* of R. Furthermore, we say that a hyperrectangle R of level ℓ is at *stage* $p \in \{0, \ldots, d-1\}$ if $d-p$ sides have length $3^{-\ell}$ and p sides have length $3^{-\ell-1}$. Note that a hyperrectangle of level ℓ and stage p is the result of p subdivisions of a cube of side length $3^{-\ell}$. Let us denote the set of all hyperrectangles of level ℓ and stage p after some round k by $\mathcal{R}_{\ell,p}^{(k)}$, and a hyperrectangle of level ℓ and stage p by $R_{\ell,p}$.

Now, let us consider the Euclidean diameter of rectangles, which is used to group the rectangles while determining potentially optimal hyperrectangles.

$$diam(\ell, p) = \sqrt{\sum_{i=1}^{p} 3^{-2\ell-2} + \sum_{i=p+1}^{d} 3^{-2\ell}} = 3^{-\ell}\sqrt{d - \frac{8p}{9}} \qquad (2)$$

That is, with increasing level and stage the diameter gets smaller.

3 Related Work

As we have already mentioned, the DIRECT algorithm was introduced in [11]. Jones, Perttunen, and Stuckman point out in their paper that the algorithm will converge arbitrarily close to the global optimum eventually, but they do not state any bounds on the rate.

Another work concerned with the convergence behavior of DIRECT is a technical report by Finkel and Kelley [7]. What they show is that the generalized directional derivative of f is positive for all directions if one is located at a so called cluster point of the evaluation sequence of DIRECT. This means that, if we find a cluster point, then we also found a local or global minimum of f.

There are also several variations of the algorithm. For example, Finkel and Kelley [8] introduce a variant that is robust against additive scaling of the objective function. Gablonsky and Kelley [9, 10] provide an implementation of DIRECT using a different scheme for grouping the hyperrectangles, i.e. not by the l^2 but by the l^∞ diameter. Björkman and Holmström [2] provide an implementation in

Matlab. Deng and Ferris [6] adapt the algorithm to accommodate noisy functions by repeated evaluation. Finally, Chiter [4, 5] proposes a modified subdivision and sampling scheme to remedy the problem of slow convergence should the minimum lie at the boundary of the domain.

Our analysis does only apply to the original algorithm and the adaption by Finkel and Kelley [8]. Other variants have to be checked in more detail, to obtain similar results to ours.

Finally, recent years have brought up several reports on practical applications using the DIRECT algorithm. Bartholomew-Biggs, Parkhurst, and Wilson [1] use it to tackle the aircraft routing problem. Zhu and Bogy [13, 14] apply it to the component design of hard drive heads. Chang, Hung, and Lee [3] use a combination of neural networks and the DIRECT algorithm to recognize human faces.

4 Convergence Result

First we introduce an upper bound on the size of $\mathcal{R}^{(k)}$. Since this equals the number of function evaluations it is what we are interested in ultimately. So, we define $n(k) := |\mathcal{R}^{(k)}|$ to be the number of hyperrectangles after round k, e.g. $n(0) = 1$ and $n(1) = 2d + 1$.

Observation 4.1. *In the DIRECT algorithm, after round k there can be at most dk groups of hyperrectangles of different l^2 diameter.*

This is a crucial observation, which is also mentioned in [11]. The subdivision of a hyperrectangle of arbitrary level ℓ and stage p is going to produce rectangles of the same level ℓ and higher stage, and three rectangles of level $\ell + 1$ and stage 0. Therefore, level i can only be populated after round i. Since we produce exactly two new hyperrectangles in every direction when we subdivide, and recurse on the third, we also established that the subdivision of a hyperrectangle can be responsible for at most $2d$ new function evaluations.

Lemma 4.2. *In the DIRECT algorithm, the number of function evaluations after round k is at most $2d^2k^2 + O(d)$, i.e.*

$$n(k) \leq 2d^2k^2 + O(d).$$

Proof. Using Observation 4.1 we can write down the following recurrence relation for $n(k)$,

$$\begin{aligned} n(1) &= 1 + 2d \\ n(i+1) &\leq n(i) + 2d^2 i, \ i \geq 2. \end{aligned}$$

Solving this we get

$$n(k) \leq n(1) + 2d^2 \sum_{i=1}^{k-1} i = 1 + 2d + 2d^2(k-1)k, \ k \in \mathbb{N}_0. \tag{3}$$

This concludes the proof, because obviously $n(k) \leq 2d^2k^2 + O(d)$. □

The second pillar we need is an upper bound on the number of rounds until a particular level/stage pair is depleted. Let us make this formal. First of all, let $n(\ell, p, k)$ be the number of hyperrectangles of level ℓ and stage p before round k. Then we can define the following,

$$\gamma(\ell, p) := \max\{k \in \mathbb{N} \mid \ n(\ell, p, k) \neq 0\}.$$

Note that this definition implies that after round $\gamma(\ell, p)$ the levels and stages of larger hyperrectangles will be depleted too, because otherwise for some $k \geq \gamma(\ell, p)$ level ℓ stage p would get populated again.

Jones, Perttunen, and Stuckman argue in [11] that the DIRECT algorithm must converge to the global optimum, because of the fact that the largest hyperrectangle is subdivided in every round. Using the same observation we are able to determine $\gamma(\ell, p)$.

Lemma 4.3. *In the DIRECT algorithm, when minimizing over Δ_d, we find that $\gamma(\ell, p) \leq 3^{d\ell+p}$.*

Proof. Let us first determine the dependence on p. Any rectangle present at some level ℓ and stage $p > 0$ is derived by a sequence of subdivisions of some rectangle at stage 0 of the same level ℓ. In this first step, we are interested in the maximum number of rounds it takes until a single rectangle $R_{\ell,0}$ on level ℓ and stage 0 and all of its children have left stages smaller or equal than some i, for $0 \leq i \leq d-1$. Note that, here we only consider $R_{\ell,0}$ and its children, so in every round one rectangle of this family is actually subdivided. Since all the levels have the same number of stages, this is the same as considering $\gamma(0, i)$. For this we have the following recursive equation,

$$\begin{aligned} \gamma(0,0) &= 1 \\ \gamma(0,i) &\leq 3\gamma(0,i-1), \quad 1 \leq i \leq d-1. \end{aligned}$$

We can upper bound $\gamma(0, i)$ by $\gamma(0, i-1)$ plus the maximum number of rectangles that will have accumulated in stage i, because every subsequent round kills exactly one of them. The maximum number of rectangles accumulating on stage i, however, is $2\gamma(0, i-1)$. To see this, we have assume the worst case, namely that every round kills at most one rectangle in the stages smaller than i. On the other hand, every subdivision of a rectangle on stages smaller than i is responsible for exactly 2 new rectangles on stage i. We get the claimed recursion. Solving we get,

$$\gamma(0,p) \leq 3^p,\ 0 \leq p \leq d-1. \tag{4}$$

Finally, we can write down the following recursive inequality for a general level $j \geq 0$,

$$\gamma(j+1,p) \leq 3\gamma(j,d-1)\gamma(0,p). \tag{5}$$

This can be explained as follows. Once a hyperrectangle is born at level $j+1$ and stage 0 we already argued that it will be responsible for at most $\gamma(0,p)$ additional rounds. The number of hyperrectangles that will actually appear on level $j+1$ and stage 0 is upper bounded by three times the number of rounds necessary to deplete everything up to level j and stage $d-1$. Recall that any subdivision of a rectangle on level j generates three new rectangles on level $j+1$ and stage 0. In fact, only rectangles that live in level j will be responsible for accumulation in level $j+1$, which means that we are counting too much, hence the inequality. If we solve equation (5) we get,

$$\gamma(\ell,p) \leq 3^{d\ell+p},\ \ell \geq 0, 0 \leq p \leq d-1. \tag{6}$$

□

Now we are ready to put everything together. We can denote the optimal function value after round $k \in \mathbb{N}_0$ by $f_{\min}^{(k)}$, and let $S^{(k)}$ be the set of sampled points after round k. Recall that $f_{\min}$ is the optimal function value and $X_{\min}$ is the set of points at which this value is attained. Now, let $X_{\min}^{(k)} \subseteq S^{(k)}$ denote the set of points for which $f(x) = f_{\min}^{(k)}$, where $x \in X_{\min}^{(k)}$. We will also need

$$y_x^{(k)} := \operatorname*{arg\,min}_{s \in S^{(k)}} ||x-s||_2,\ \forall x \in X_{\min}.$$

This is the sampled point closest to x, at the end of round k. Note that we do not explicitly know $y_x^{(k)}$ for any x, because we do not know $X_{\min}$ of course. However, we will make statements about the maximal distance between x $y_x^{(k)}$. In every round of DIRECT at least one of the largest rectangles is subdivided. Therefore, it is not hard to see that $y_x^{(k)}$ converges to x for $k \to \infty$. The question is how fast.

As before, assume that $x \in X_{\min}$ in the following. The trivial observation on which we base our analysis is that

$$||x - y_x^{(k)}||_2 \leq \max_{R_{\ell,p} \in \mathcal{R}^{(k)}} \frac{diam(\ell,p)}{2} = \max_{R_{\ell,p} \in \mathcal{R}^{(k)}} \frac{1}{2} 3^{-\ell} \sqrt{(d - \frac{8}{9}p)}. \tag{7}$$

It simply means that a location of the global optimum, x, cannot be farther away from its nearest evaluation point than half the diameter of the largest rectangle in round k. The equality simply follows from (2).

Using equation (7) we can pinpoint the number of rounds required by the DIRECT algorithm until we get a satisfactorily precise location of the minimum value.

Lemma 4.4. *Let $\epsilon > 0$, and f an arbitrary function, being minimized over Δ_d, using the DIRECT algorithm. In this setting $O((\frac{\sqrt{d}}{2\epsilon})^d)$ rounds are sufficient to guarantee an evaluation point closer than ϵ to the location(s) of the minimum of f. Formally, for some $k = c\,(\frac{\sqrt{d}}{2\epsilon})^d$, with some suitably chosen constant c, after k rounds we have,*

$$\forall x \in X_{\min}\ \exists x' \in X_{\min}^{(k)} \ \ \textit{such that} \ \ |x - x'| < \epsilon.$$

Proof. We use equation (7) and require

$$\max_{R_{\ell,p}\in\mathcal{R}^{(k)}} \frac{diam(\ell,p)}{2} < \epsilon. \tag{8}$$

By setting $p = 0$ in (7), we can derive

$$\ell > \frac{1/2\ln d - \ln\epsilon - \ln 2}{\ln 3}, \ \forall R_{\ell,p} \in \mathcal{R}^{(k)}. \tag{9}$$

In plain words, this states that k has to be such that the level of all rectangles in round k is larger than the right hand side term in equation (9). In Lemma 4.3 we already computed the number of rounds necessary to reduce the largest hyperrectangle sufficiently. In fact, we need

$$k > \gamma(\frac{1/2\ln d - \ln\epsilon - \ln 2}{\ln 3}, 0) = e^{1/2\,d\,\ln d}\ e^{-d\,\ln\epsilon}\ e^{-d\,\ln 2}\ \in O((\frac{\sqrt{d}}{2\epsilon})^d) \tag{10}$$

rounds. □

Because we are ultimately interested in the number of function evaluations, this naturally leads us to the following theorem.

Theorem 4.5. *Let $\epsilon > 0$, and f an arbitrary function, being minimized over Δ_d, using the DIRECT algorithm. We need $\Theta(d^{d+2}\frac{1}{(2\epsilon)^{2d}})$ function evaluations to guarantee an evaluation point closer than ϵ to the location(s) of the minimum. Formally, after $c\,d^{d+2}\frac{1}{(2\epsilon)^{2d}}$ function evaluations, using some suitably chosen positive constant c, for any $x \in X_{\min}$, we already have evaluated a point x' such that $|x - x'| < \epsilon$.*

Proof. Given Theorem 4.4 and Lemma 4.2 the proof of the upper bound is easy. We know that we need $k \in O((\frac{\sqrt{d}}{2\epsilon})^d)$ rounds, and that this results in at most $2d^2k^2 + O(d)$ function evaluations. The upper bound of the theorem follows immediately.

To see the lower bound, we have to argue that there is a function such that the DIRECT algorithm actually uses that many function evaluations. Consider the following function, $f : \Delta_d \to \mathbb{R}$, and f is constant "almost everywhere". The minimum at some corner of Δ_d, and f rises steeply from the minimum at this corner.

□

The DIRECT algorithm was designed to handle optimization problems where the evaluation of the objective function is only possible at considerable cost. Therefore, in the next theorem, we relate the number of function evaluations necessary to achieve some small absolute error.

Remark 4.1. Let us reconsider what we have showed. We have bounded the complexity of DIRECT by relating it to the number of function evaluations. This might seem somewhat naive, because we are forgetting about all the other steps. However, it is justified. First, because of the design principle of DIRECT, and second because of the following.

During the execution of the algorithm there are only a few steps that are imposing an additional computational burden. There is the updating of the function domain Δ_d when we subdivide. However, this can be completely amortized in the number of function evaluations. Every time we have to create a new cell, we also have to evaluate its center. Therefore, there will not be more of such constant time operations as there will be evaluations.

The second suspicious operation is the computation of the convex hull in every round. Luckily, this also fits into the bound we claimed. In [11] Jones, Perttunen, and Stuckman point out that we can considerably shorten the number of steps necessary for the computation of the convex hull. In fact, by keeping the points sorted along the abscissa and ordinate, we only need $O(m)$ steps, where m is the number of distinct diameters occurring. As mentioned before, there can only be dk different diameters after any round k. Therefore, for the computation of convex hulls during k rounds, we only need $O(\sum_{i=1}^{k} di)$ steps. Plugging in $k \in O(d^d \frac{1}{\epsilon^d})$, we see that we are still on the safe side by using $O(d^{2d+1} \frac{1}{\epsilon^{2d}})$ steps only.

Remark 4.2. In the abstract we have mentioned that we can make a similar statement about the sampling distance from the global minimum, which still holds even if we drop the Lipschitz continuity of f. In fact, the analysis goes through exactly the same way. We simply replace $|f_{\min} - f_{\min}^{(k)}|$ by $||x_{\min} - x_{close}^{(k)}||_2$

in Theorem 4.4 and in Theorem 4.5. Consequently, we do not need the factor L in equation (8), which means that the hidden constants of both theorems get better.

On the other hand, this result is not as useful as the other one, because we do not gain a lot of knowledge about the location of the optimal value, as it might be implied. Obviously, $x_{close}^{(k)}$ and $x_{\min}$ can be arbitrarily far apart.

5 Lower Bound

Since our result does not look very promising we also give a lower bound for the general model in which we are only allowed to evaluate the function. That is, how many evaluations do we need at least to achieve a given absolute error ϵ.

Theorem 5.1. *Let $\epsilon > 0$. If f is a Lipschitz continuous function with Lipschitz constant L, being minimized over Δ_d, any algorithm that is only allowed to make function evaluations might need $\Omega(L^d d^d \frac{1}{\epsilon^d})$ evaluations to guarantee*

$$|f_{\min} - f_{\min}^{(k)}| < \epsilon.$$

Proof. Look at k evaluations. Construct polynomial that agrees with those evaluations, but has its minimum still far apart. □

6 Conclusion

We have showed the first derivation of a convergence rate for the DIRECT algorithm. We can bound the absolute error in function value and also the maximum sampling distance from the global minimum, but we cannot guarantee to find its location for a given prescribed error ϵ.

The estimations we have made are still very crude. In particular, the assumption that we have to let the largest rectangle shrink sufficiently. In practice, experiments show that DIRECT usually shows a much nicer convergence behavior, namely clustering of small hyperrectangles around the minima of the function. However, it seems reasonable that for a particular choice of ε in the definition of potential optimality, we can always find a function which shows the worst case convergence that we assume here. In particular, if we do not have Lipschitz continuity, because then we can prescribe the function value with arbitrary granularity and rate of change.

In view of the increasing popularity of the DIRECT algorithm we hope to have opened a door for further investigation. It might be possible to specify conditions on f which are easily checked and which exhibit a much faster convergence.

References

[1] M. C. Bartholomew-Biggs, S. C. Parkhurst, and S. P. Wilson. Using direct to solve an aircraft routing problem. *Computational Optimization and Applications*, 21(3):311–323, 2002.

[2] Mattias Björkman and Kenneth Holmström. Global optimization using the DIRECT algorithm in matlab, 1999.

[3] Yau-Zen Chang, Kao-Ting Hung, and Shih-Tseng Lee. Human face detection with neural networks and the DIRECT algorithm. *Artificial Life and Robotics*, 12(1):112–115, 2008.

[4] Lakhdar Chiter. Towards a new DIRECT algorithm. A two-points based sampling method, 2005.

[5] Lakhdar Chiter. A new sampling method in the DIRECT algorithm. *Applied Mathematics and Computation*, 175(1):297–306, 2006.

[6] Geng Deng and Michael C. Ferris. Extension of the DIRECT algorithm for noisy functions. In *Proceedings of the Winter Simulation Conference WSC 2007*, pages 312–319. WSC, 2007.

[7] D. E. Finkel and C. T. Kelley. Convergence analysis of the DIRECT algorithm, 2004.

[8] D. E. Finkel and C. T. Kelley. Additive scaling and the DIRECT algorithm. *Journal of Global Optimization*, 36(4):597–608, 2006.

[9] J. Gablonsky. An implementation of the direct algorithm, 1998.

[10] J. M. Gablonsky and C. T. Kelley. A locally-biased form of the DIRECT algorithm. *Journal Global Optimization*, 21(1):27–37, 2001.

[11] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[12] Bruno O. Shubert. A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis*, 9(3):379–388, 1972.

[13] Hong Zhu and David B. Bogy. Direct algorithm and its application to slider air bearing surface optimization. *Tribology International*, 38(2):2168–2170, 2002.

[14] Hong Zhu and David B. Bogy. Hard disc drive air bearing design: modified direct algorithm and its application to slider air bearing surface optimization. *Tribology International*, 37(2):193–201, 2004.