

Directed Voronoi Search: a method for bound constrained global optimization

B. L. Robertson^a, C. J. Price^a, M. Reale^a, J. A. Brown^a

^a*Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch, New Zealand*
Email: M.Reale@math.canterbury.ac.nz

Abstract: Global optimization problems occur in many fields including mathematics, computer science, engineering, and economics. The purpose of a global optimization algorithm is to efficiently find an objective function's global minimum, usually in some bounded region. In this article we consider bound constrained global optimization, where the search is performed in a box. The global optimization problem is deceptively simple and it is usually difficult to find the global minimum on general functions. One of the difficulties is that there is often no way to verify that a local minimum is indeed a global minimum. If the objective function is convex, for example, the local minimum is also the global minimum. However, many optimization problems are not convex. Of particular interest in this article are objective functions that lack any special properties such as continuity, smoothness, or a Lipschitz constant.

This article presents a new algorithm for bound constrained global optimization, called Directed Voronoi Search (DVS). At each iteration the objective function is evaluated at a batch of points selected from the optimization region Ω . Initially, the points are selected from a uniform distribution over Ω . These points form a set called the training data and are used to construct a Voronoi diagram on Ω . The Voronoi diagram partitions Ω into a collection of convex polyhedral sub-regions with one point in each sub-region. The objective function is then evaluated at vertices determined by sub-regions containing points with low function values, and large sub-regions. The former direct the search to areas where the objective function is presumed to be low, and the latter shrink large sub-regions to reduce the risk of missing the global minimum. These new points are then added to the training data, the Voronoi diagram is recalculated, and the method repeats.

In order to establish convergence we require that the objective function is lower semi-continuous and bounded below. With these relatively mild conditions the DVS algorithm is shown to converge to an essential global minimizer almost surely. It is also possible to implement a deterministic DVS algorithm. The deterministic instance is shown to converge to an open set essential global minimizer.

Numerical results show that the DVS algorithm is effective at solving low dimensional global optimization problems. Comparisons with the DIRECT algorithm and Accelerated Random Search (ARS) are also made. The DVS algorithm is superior to ARS on the test problems considered and performed similarly to DIRECT.

Keywords: *DIRECT, global optimization, Halton sequence, numerical results, Voronoi diagram*

1 INTRODUCTION

The bound constrained global optimization problem is of the form

$$\min f(x) \text{ subject to } x \in \Omega, \quad (1)$$

where the search region Ω is defined by an n -dimensional box of the form

$$\Omega = \{x \in \mathbb{R}^n : l_i \leq x_i \leq u_i \text{ for all } i = 1, \dots, n\}.$$

The objective function f maps Ω into $\mathbb{R} \cup \{+\infty\}$ and is assumed to be lower semi-continuous. The inclusion of $\{+\infty\}$ means the objective can be assigned the value $+\infty$ at points or regions where it cannot be evaluated. This means certain constrained optimization problems can be handled using the extreme barrier approach of Audet and Dennis (2003), which assigns the value $+\infty$ to infeasible points.

Despite the deceptively simple form of (1), finding a global minimum is usually very difficult because there is often no way to verify that a local minimum is indeed a global minimum. Nevertheless a smorgasbord of methods have been proposed in the literature to handle a variety of optimization problems. These methods can be sorted into two main groups — deterministic methods and stochastic methods (also known as random search or Monte Carlo methods). Stochastic methods (such as genetic algorithms, simulated annealing, multi-start, and clustering algorithms, see for example Zabinsky (2003); Zhigljavsky (1991)) include an element of randomness or probability in their design and deterministic methods (such as interval analysis, branch and bound, and tunnelling methods, see for example Horst and Hoang (1996); Horst and Pardalos (1995)) do not. Stochastic methods ensure convergence to a global minimum in probability, and deterministic methods tend to guarantee asymptotic convergence.

The simplest method for solving (1) is to evaluate f at a number of randomly generated points over Ω and take the best function value as an estimate of the global minimum. This method is called Pure Random Search (PRS). Unsurprisingly, PRS often performs poorly in practice and a practitioner would usually choose another method to solve their particular problem. In this section we briefly describe two popular methods that are used in the numerical results section of this article.

The Accelerated Random Search (ARS) algorithm of Appel et al. (2003) is a stochastic method for solving (1). ARS evaluates the objective function at points selected from a finite sequence of contracting sub-regions (initially Ω itself) centered on the point with the best function value. If a point with a lower function value is found, or if the sequence is exhausted, the search returns back to Ω . This approach may appear counter intuitive, but can be very effective at refining the current estimate of the global minimum. However, if the current estimate, x_k , is only a local minimizer, ARS can perform poorly Price et al. (2012). Let x_* be the closest global minimizer to x_k . To reduce f below $f(x_k)$, ARS needs to find a point in the set $L(x_k) = \{x \in \Omega : f(x) \leq f(x_k)\}$. If $\|x_k - x_*\|$ is large ARS might only be able to sample $L(x_k)$ by randomly sampling Ω . Thus sampling $L(x_k)$ can be a low probability event.

DIRECT is a deterministic algorithm that Jones et al. (1993) presented to solve (1). DIRECT generates a sequence of nested partitions on Ω . Each partition consists of a finite number of boxes (initially Ω itself) with the objective function evaluated at the center of each box. At each iteration one or more boxes are divided into three smaller, equally sized boxes using two hyperplanes orthogonal to one of the coordinate axes. Dividing a box in this way means that the center point of the old box will be the center point for one of the new boxes. Hence, only two function evaluations are required for each box that is divided. The method continues dividing boxes in this way, generating finer and finer partitions on Ω , until stopping conditions are satisfied.

In this article a new optimization method is presented, called Directed Voronoi Search (DVS). The method evaluates the objective function at points determined by a Voronoi diagram. We begin by introducing the Voronoi partition on Ω . The DVS algorithm is described in Section 3 and convergence results are given in Section 4. The algorithm is tested on a selection of bound constrained global optimization problems in Section 5. Concluding remarks are given in Section 6.

2 VORONOI PARTITION

The Voronoi diagram of a set of points is a partition on \mathbb{R}^n that assigns surrounding regions to nearby points. A Voronoi partition on Ω using a set of points

$$X = \{x_i \in \Omega : i = 1, 2, \dots, N\}$$

is defined by the set of N convex polyhedral sub-regions

$$A_i = \{x \in \Omega : \|x - x_i\|_2 \leq \|x - x_j\|_2 \text{ for all } j = 1, 2, \dots, N\}$$

for $1 \leq i \leq N$ (see Figure 2). The Voronoi cell for x_i is A_i , and the notation V_i is used to denote the set of vertices for x_i 's Voronoi cell. The set of all vertices, $\mathcal{V} = \{v \in V_i : i = 1, 2, \dots, N\}$, is called the Voronoi vertex set (see Figure 2).

3 THE DVS ALGORITHM

The Directed Voronoi Search (DVS) algorithm consists of a single loop. At each iteration a set of points with known objective function values, called the training data set T , is used to determine the Voronoi vertex set, \mathcal{V} . The objective function is then evaluated at points selected from both \mathcal{V} and Ω . Selection strategies are described in the following two subsections. The new points are then added to T , \mathcal{V} is recalculated, and the method repeats. To stop T becoming arbitrarily large, points with relatively high function values are removed from the training data set if $|T| > T_{\max}$. The algorithm generates a sequence of points $\{x_k\}$ with monotonically decreasing objective function values $\{f(x_k)\}$. A statement of the DVS algorithm is given in Figure 1.

-
- (1) Choose $N > 0$, $N_\Omega \geq 1$, and $T_{\max} \gg N$. Evaluate the objective function at N points in Ω and call this set of points T_0 . Let x_0 be the best point found and set $k = 0$.
 - (2) Form a Voronoi partition on Ω using T_k and calculate the Voronoi vertex set, \mathcal{V}_k .
 - (3) Select a finite number of points from \mathcal{V}_k and N_Ω point(s) from a uniform distribution over Ω . Call this set of points X and evaluate the objective function at each $x \in X$.
 - (4) Set $T_{k+1} = T_k \cup X$ and let x_{k+1} be the best point found. If $|T_{k+1}| > T_{\max}$, the $|T_{k+1}| - T_{\max}$ points with the largest f values are discarded. If stopping conditions are not met, increment k and go to Step (2).
-

Figure 1. The Directed Voronoi Search algorithm.

3.1 Selection Strategies

There is much freedom in selecting points at Step (3) of the DVS algorithm. Spreading the points evenly over Ω can reduce the risk of missing the global minimum, but will tend to have a slow rate of convergence. Directing the search where the objective is presumed to be low can increase the rate of convergence to a local minimum, but may miss the global minimum altogether. In this article we try to strike a balance between both approaches by evaluating the objective at vertices selected from Pareto optimal Voronoi cells (see Figure 2).

Definition 1. (Pareto Optimal Voronoi Cell). A Voronoi cell containing the sample point x is Pareto optimal unless there exists a cell containing a point z satisfying both

$$f(z) \leq f(x) \quad \text{and} \quad m(z) \geq m(x)$$

with at least one equality holding strictly, where $m(x)$ is the Lebesgue measure of the cell containing x .

Evaluating the objective at Pareto optimal Voronoi cells means that the cells in Voronoi partitions will not remain large, and that the point density will increase where the objective function is known to be relatively low. The former reduces the risk of missing the global minimum and the latter attempts to increase the rate of convergence.

The stochastic DVS algorithm selects an initial batch of $N > 0$ points, T_0 , from a uniform distribution over Ω . Step (3) selects at least one point ($N_\Omega \geq 1$) from a uniform distribution over Ω , and a finite number of points from the k^{th} Voronoi vertex set \mathcal{V}_k . We consider two approaches:

- (I). For each Pareto optimal Voronoi cell V_i , evaluate the objective function at the vertex which has the greatest 2-norm distance from x_i . If $f(x_i) \leq f(x_j)$ for all $x_j \in T$, also evaluate the objective at the vertex which has the least 2-norm distance from x_i .

(II). Same as approach (I), but if $f(x_i) \leq f(x_j)$ for all $x_j \in T$, evaluate the objective at all the vertices in V_i .

Evaluating the objective at the farthest vertex gives the greatest reduction in cell size when the Voronoi partition is recalculated and keeps the search global. Evaluating the closest vertex on the cell with the best function value aims to reduce f locally. Approach (II) has a stronger local search component, allowing the algorithm to focus its search in the neighborhood of a local (potentially global) minimizer.

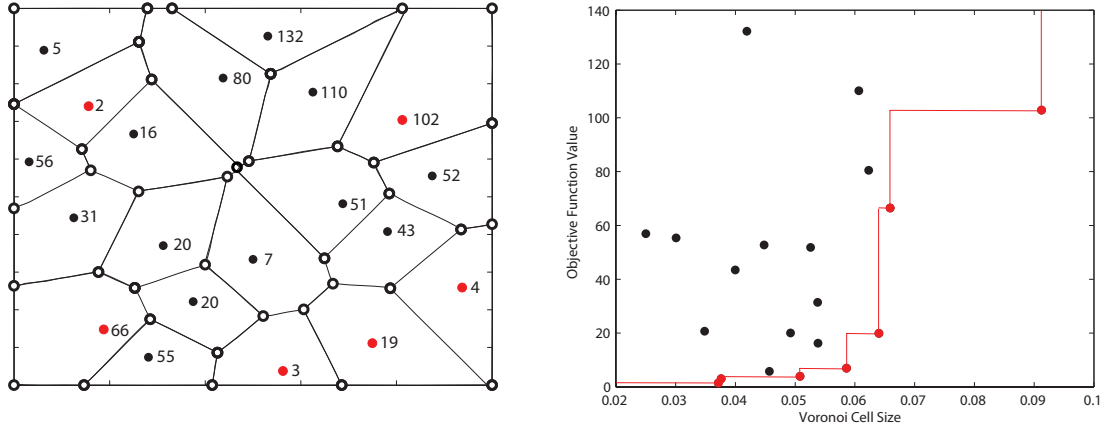


Figure 2. 20 points distributed over Ω for the Branin problem. The number next to each point states the objective function value at that point. (Left) The Voronoi partition on Ω , where points in the Voronoi vertex set are denoted \circ . (Right) A plot showing which cells are Pareto optimal. Points corresponding to Pareto optimal Voronoi cells are shown in red. Note that the cell with best function value and the cell with the largest area are both Pareto optimal.

3.2 Deterministic DVS

A deterministic DVS algorithm is also presented. The deterministic instance selects points from the Halton sequence, Halton (1960). The Halton sequence is a quasi-random sequence of points that is known to quickly distribute points evenly over $[0, 1]^n$ in low dimensions. Furthermore, the Halton sequence generates a dense set of points in $[0, 1]^n$.

Proposition 1. *The Halton sequence $\{u_t\}_{t=1}^{\infty}$ is dense in $[0, 1]^n$.*

Proof. A simple proof is given by Abramson et al. (2009) and Halton (1960) provides a detailed proof. \square

The first N points in the Halton sequence are used to give the initial batch of points T_0 . Step (3) selects at least one point from a uniform distribution over Ω . These points are selected sequentially from the Halton sequence (scaled and translated to Ω) in the deterministic instance. Hence after k iterations, the objective function has been evaluated at the first $N + kN_{\Omega}$ points in the Halton sequence. The points from \mathcal{V}_k are selected from the Pareto optimal Voronoi cells using either approach (I) or (II) described above.

4 CONVERGENCE

The convergence properties of the Directed Voronoi Search algorithm are analyzed with the stopping conditions removed. This allows us to examine the asymptotic properties of the sequence of iterates generated by the algorithm. The first convergence result shows that every cluster point of the sequence $\{x_k\}$ generated by the stochastic DVS algorithm is an essential global minimizer of f with probability one.

Definition 2. (Essential global minimizer). *An essential global minimizer x_* is a point for which the set*

$$L(x_*) = \{x \in \Omega : f(x) < f(x_*)\}$$

has Lebesgue measure zero.

If the objective function is continuous, then $f(x_*)$ is also a global minimum in the classical sense.

Assumption 1. *The objective function f is bounded below and lower semi-continuous on Ω .*

The lower semi-continuous condition precludes the existence of a sequence $\{x_k\}$ converging to a point x_* for which $f(x_*) - \delta > f(x_k)$ for some $\delta > 0$ when $\|x_k - x_*\| < \epsilon$, where $\epsilon > 0$ and sufficiently small.

Theorem 1. *The sequence $\{x_k\}$ generated by the Directed Voronoi Search algorithm is an infinite sequence.*

Proof. It is sufficient to show that the main loop of the algorithm (Steps 2 - 4) is a finite process. The cardinality of the training data set T_k is bounded above by T_{\max} and so Steps 2 and 4 are finite processes. Step 3 evaluates the objective function at a finite number of points. \square

Theorem 2. *Each cluster point x_* of the sequence $\{x_k\}$ generated by the stochastic Directed Voronoi Search algorithm is an essential global minimizer of f with probability one.*

Proof. Assumption 1, Theorem 1, and the fact that Ω is bounded ensure the existence of cluster points in $\{x_k\}$.

Let x_* be any cluster point of $\{x_k\}$ and assume, by contradiction, x_* is not an essential global minimizer of f . The definition of an essential global minimizer ensures there exists a set $L(x_*) = \{z \in \Omega : f(z) < f(x_*)\}$ with positive Lebesgue measure. Step 3 selects at least one random point from Ω at each iteration and so Ω is sampled infinitely often. Therefore, the probability that L is sampled is at least

$$\lim_{k \rightarrow \infty} 1 - (1 - m(L)/m(\Omega))^k = 1.$$

Hence, $\lim_{k \rightarrow \infty} f(x_k) < f(x_*)$ almost surely, which contradicts the lower semi-continuous assumption on f . Thus, x_* must be an essential global minimizer of f , almost surely. \square

The following theorem shows that the deterministic Directed Voronoi Search algorithm converges to an open set essential global minimizer.

Definition 3. (Open set essential global minimizer). *An open set essential global minimizer $x_{\#}$ is a point whose function value $f_{\#}$ is not more than the supremum of the values $f_0 \in \mathbb{R}$ for which the level set*

$$L(f_0) = \{x \in \Omega : f(x) < f_0\}$$

contains no open ball of positive radius.

To show that the deterministic DVS algorithm converges to an open set essential global minimizer it is sufficient to show that the algorithm generates a dense set of points in Ω .

Theorem 3. *Each cluster point x_* of the sequence $\{x_k\}$ generated by the deterministic Directed Voronoi Search algorithm is an open set essential global minimizer of f .*

Proof. Assumption 1, Theorem 1, and the fact that Ω is bounded ensure the existence of cluster points in $\{x_k\}$.

Let x_* be any cluster point of $\{x_k\}$ and assume, by contradiction, x_* is not an open set essential global minimizer of f . The definition of an open set essential global minimizer ensures there exists an open ball $L \subset \Omega$ on which the objective function is less than $f(x_*)$. Proposition 1 shows that the Halton sequence is dense in Ω and Step (3) selects points sequentially from the Halton sequence. Hence, there exists a k for which $x_k \in L$, giving $f(x_k) < f(x_*)$. The sequence $\{f(x_k)\}$ is monotonically decreasing, so $f(x_k) < f(x_*)$ for all k sufficiently large, contradicting the lower semi-continuity of f . Thus, x_* must be an open set essential global minimizer of f . \square

5 NUMERICAL RESULTS

In this section we investigate the performance of the DVS algorithm on 20 two-dimensional problems, as listed in Table 1. Problems 1-10 are from Ali et al. (2005), 11-16 from Moré et al. (1981), and 17-19 from Price et al. (2012). The offset Rastrigin function is given by

$$f_{20} = (x_1 - 2/3)^2 + (x_2 - 1/3)^2 - \cos(18x_1 - 12) - \cos(18x_2 - 6).$$

The search regions are from Price et al. (2013), listed in column 3 of Table 1. With the exception of problem 3, the search regions are square and so the lower and upper bounds of each dimension are listed.

Numerical results were generated with $N = 20$, $N_\Omega = 1$, and $T_{\max} = 50000$. A lower bound $\tau = 1e-10$ was also placed on cell size. For a cell x to be Pareto optimal, it must satisfy Definition 1 and have $m(x) > \tau$. There was little difference between the stochastic and deterministic instances of the DVS algorithm. Hence, only the deterministic results are presented. The DVS algorithm was compared with ARS and DIRECT. The parameters of these algorithms were chosen as in Price et al. (2012).

	Problem	Ω	ARS	DIRECT	DVS I	DVS II
1	Ackley	$[-30, 30]$	2624	-	1468	5027
2	Becker-Lago	$[-10, 10]$	118	81	516	149
3	Branin	$[-5, 10] \times [0, 15]$	153	107	639	140
4	Camel Back - 3	$[-5, 5]$	120	-	254	115
5	Camel Back - 6	$[-5, 5]$	123	159	293	140
6	Cosine Mixture	$[-1, 1]$	192	-	510	92
7	Dekkers-Aarts	$[-20, 20]$	1068	291	3805	1219
8	Goldstein-Price	$[-2, 2]$	191	123	1249	166
9	mod. Rosenbrock	$[-5, 5]$	3 fails	405	666	281
10	Schubert	$[-10, 10]$	329	2285	524	322
11	Beale	$[0, 5]$	214	329	687	434
12	Brown almost linear	$[-5, 25]$	226	197	751	269
13	Chebyquad	$[-1, 1]$	44	25	40	48
14	Freudenstein-Roth	$[-15, 15]$	2067	5619	6137	590
15	Jennrich-Sampson	$[-10, 10]$	18833	527	4440	4162
16	Trigonometric	$[-0.5, 1]$	102	85	209	63
17	Weka 1	$[0, 1]$	114	15387	37	30
18	Weka 2	$[0, 1]$	1722	1691	879	341
19	Weka 3	$[0, 1]$	4 fails	1861	48	27
20	offset Rastrigin	$[-1, 1]$	2929	749	1031	3228

Table 1. Number of function evaluations required to satisfy $f(x_k) - f(x_*) \leq 1e-3$ (problem 16 uses $1e-5$ to eliminate a known local minimum). ARS is stochastic so the results are averages over ten runs. Runs using more than 50000 function evaluations were halted and listed as ‘fails’. If ARS failed on a problem, the number of fails is reported. DIRECT always selects Ω ’s center point as its first point. Hence, comparisons with DIRECT on problems with a global minimizer at Ω ’s center are meaningless. These problems are marked with ‘-’ in DIRECT’s column. Figures marked in bold indicate the fastest method for each problem.

The final two columns in Table 1 list the results for the deterministic DVS algorithm using selection approach (I), DVS I, and using selection approach (II), DVS II. Both approaches solved all the problems considered and DVS II was faster on 17 of the problems. This suggests that evaluating the objective at every vertex on the cell with the best function value is an effective strategy. This additional local search effort halved the required number of function evaluations to solve most problems.

DVS II performed better than ARS on this set of problems. ARS failed to solve problems 9 and 19 on three and four runs, respectively. On each fail ARS converged to a local, rather than global, minimum from which it could not escape within 50000 function evaluations. For the 18 problems that ARS solved, it was notably faster than DVS II on problems 1 and 11 and slightly faster on problems 2, 5, 7, 12, 13, and 20. However, DVS II is the superior method because it solved all the problems and was faster on 12 of them.

DIRECT and DVS II performed similarly on the 17 problems used for comparison — DVS II was faster on 8 problems and DIRECT was faster on 9. DIRECT required considerably more function evaluations than the other methods to solve problem 17. Any point of the form $x_* = (-1, x_2)$, where $x_2 \in [-1, 1]$, is a global minimizer for this problem. It does not take long for the DVS algorithm to solve this problem because some of the cell vertices will have $x_1 = -1$ (see Figure 2). DIRECT systematically divides up the search space as it approaches the $x_1 = -1$ boundary, requiring many function evaluations.

A final problem from Price et al. (2012) is used to test the robustness of ARS, DIRECT, and DVS. The function is given by

$$f_{\text{weka4}} = \min \left(2 + x_1 \cos(\theta) + x_2 \sin(\theta), 40\|x - c\|_2 \right) \quad \text{over} \quad [-1, 1]^2,$$

where $\theta \in [0, 2\pi]$ and $c \in [-1, 1]^2$ are chosen randomly. Most of this function is determined by an inclined plane, but the global minimizer is located at the base of the randomly positioned cone that cuts the plane. Hence, most of this function provides no information about where the global minimum is located. Ten different instances of this problem were generated and used to test the performance of ARS, DIRECT and DVS. DIRECT, DVS I, and DVS II took an average of 12104, 7215, and 4912, respectively, to reduce f_{weka4} below $1e-3$. On average, DVS II is twice as fast as DIRECT on this difficult problem. ARS failed to solve this problem. Each run converged to the local minimum determined by the inclined plane, which lies at a corner of Ω .

6 CONCLUSION

A new algorithm for bound constrained global optimization is presented, called Directed Voronoi Search. Initially the objective function is evaluated at a batch of points selected from the optimization region Ω . These points form a training data set which is used to construct a Voronoi partition on Ω . The objective function is then evaluated at certain vertices of the Voronoi partition. These vertices are chosen to potentially reduce the objective function or to reduce the risk of missing the global minimum. These new points are then added to the training data, the Voronoi partition is recalculated, and the method repeats. Convergence to an essential global minimizer with probability one is demonstrated under mild conditions. A deterministic instance is also shown to converge to an open set essential global minimizer. Numerical simulations show that the method is effective at solving low dimensional global optimization problems.

REFERENCES

- Abramson, M. A., C. Audet, J. E. Dennis, and S. L. Digabel (2009). Orthomads: a deterministic mads instance with orthogonal directions. *SIAM J. Optimization* (20), 948–966.
- Ali, M. M., C. Khompatporn, and Z. B. Zabinsky (2005). A numerical evaluation of several stochastic algorithms on selected continuous global optimization problems. *Journal of Global Optimization* (31), 635–672.
- Appel, M. J., R. Labarre, and D. Radulović (2003). On accelerated random search. *SIAM J. Optimization* (14), 708–731.
- Audet, C. and J. E. Dennis (2003). Analysis of generalized pattern searches. *SIAM J. Optimization* (13), 889–903.
- Halton, J. H. (1960). On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik* (2), 84–90.
- Horst, R. and T. Hoang (1996). *Global Optimization: Deterministic Approaches*. Berlin: Springer-Verlag.
- Horst, R. and P. M. Pardalos (1995). *Handbook of Global Optimization*. Netherlands: Kluwer Academic Publishers.
- Jones, D., C. D. Perttunen, and B. E. Stuckman (1993). Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications* (79), 157–181.
- Moré, J. J., B. S. Garbow, and K. E. Hillstrom (1981). Testing unconstrained optimization software. *ACM Transactions on Mathematical Software* (7), 17–41.
- Price, C. J., M. Reale, and B. L. Robertson (2012). A cover partitioning method for bound constrained global optimization. *Optimization Methods and Software* (27), 1059–1072.
- Price, C. J., M. Reale, and B. L. Robertson (2013). One side cut accelerated random search. *Optimization Letters*, early access online.
- Zabinsky, Z. B. (2003). *Stochastic Adaptive Search for Global Optimization*. Massachusetts: Kluwer Academic Publishers.
- Zhigljavsky, A. A. (1991). *Theory of Global Random Search*. Netherlands: Kluwer Academic Publishers.