Journal of Global Optimization **14:** 283–298, 1999.
© 1999 *Kluwer Academic Publishers. Printed in the Netherlands.*

A Non-myopic Utility Function for Statistical Global Optimization Algorithms *

SIMON STRELTSOV[1] and PIROOZ VAKILI[2]
Manufacturing Engineering Department, Boston University, 15 St Mary's St., Boston, MA 02215, USA
[1]*e-mail: simon@alphatech.com;* [2]*e-mail: vakili@bu.edu*

(Received 30 May 1996; accepted in revised form 2 February 1998)

Abstract. The high cost of providing "worst-case" solutions to global optimization problems has motivated the development of "average-case" algorithms that rely on a statistical model of the objective function. The critical role of the statistical model is to guide the search for the optimum. The standard approach is to define a utility function $u(x)$ that in a certain sense reflects the benefit of evaluating the function at x. A proper utility function needs to strike a balance between the immediate benefit of evaluating the function at x – a myopic consideration; and the overall effect of this choice on the performance of the algorithm – a global criterion. The utility functions currently used in this context are heuristically modified versions of some myopic utility functions. We propose using a new utility function that is provably a globally optimal utility function in a non-adaptive context (where the model of the function values remains unchanged). In the adaptive context, this utility function is not necessarily optimal, however, given its global nature, we expect that its use will lead to the improved performance of statistical global optimization algorithms. To illustrate the approach, and to test the above assertion, we apply this utility function to an existing adaptive multi-dimensional statistical global optimization algorithm and provide experimental comparisons with the original algorithm.

Key words: Average-case, Statistical global optimization, Utility function

1. Introduction

We consider a class of global optimization methods that have two distinctive features: (a) they assume or construct a statistical model of the objective function, and, (b) they look for solutions that are good "on average" instead of considering worst-case scenarios.

It is known that worst-case approaches to global optimization lead to algorithms of exponential complexity. The appeal of average-case approaches is in the possibility of obtaining algorithms that can work efficiently for typical problems without paying a high premium for worst-case guarantees. A drawback of average-case algorithms, on the other hand, is that they often require a larger amount of

* The work in this paper was supported in part by the National Science Foundation under grant EID-9527422.

auxiliary computations to update the model of the objective function and to find the best location for function evaluation. See Žilinskas (1992) and Streltsov et al. (1996) for further discussions.

The critical role of the statistical model is to guide the search for the optimum. Typically, a utility function is defined that, in some sense, measures the potential of each point or region in the domain of the function to yield large values (assume we are maximizing). Then, at each iteration of the optimization algorithm, the next location of the function evaluation is selected as that which maximizes the utility function.

To make things more explicit, consider the following setting:

Let $f : A \to \mathbf{R}$ be a real valued deterministic function defined on a bounded set $A \subset \mathbf{R}^d$, and consider the following maximization problem:

$$\max_{x \in A} f(x) \tag{1}$$

Then, average-case algorithms typically work as follows:

Assume that f is evaluated as t points $\{x_i; i = 1, \dots, t\}$ and let all observations up to t be denoted by $\zeta_t = \{(x_i, y_i); i = 1, \dots, t\}$ where $y_i = f(x_i)$. Let

$$F(y; x, \zeta_t) = P(f(x) \leq y | \zeta_t). \tag{2}$$

denote the conditional distribution of $f(x)$ given ζ_t. Given this model of the function values, a utility function $u(x; \zeta_t)$ is defined to reflect in a certain way the "reward" of selecting x as the next location for function evaluation. The maximizer of $u(x; \zeta_t)$ is then chosen as the next point to evaluate the function. Different approaches vary in their choice of the model for the objective function, $F(y; x, \zeta_t)$, and their choice of the utility function $u(x; \zeta_t)$. See, e.g., Žilinskas (1992), Mockus (1989, 1994), and Betrò (1991) for reviews of the existing methods.

Most algorithms use a number of simplifying assumptions in order to reduce the computational complexity of the algorithm. One of these simplifications consists of using a utility function that reflects *the immediate* or one-step reward of evaluating the function at a point x, while disregarding the effect of this choice on the overall performance of the algorithm.

$$u(x; \zeta_t) = E[f(x) - Z_t]^+, \quad \text{or,} \quad u(x; \zeta_t) = P(f(x) \geq Z_t), \tag{3}$$

are examples of the utility functions considered (x^+ denotes $\max(x, 0)$, and $Z_t = \max\{y_1, \dots, y_t\}$).

Informally, we are interested in striking a balance between continuing the search in areas where large function values have already been found (and, typically, the expected function values $E[f(x)]$ are large), and searching in areas that are not sufficiently explored (and where, typically, $\text{Var}(f(x))$ is large). The utility functions (3), however, take only the immediate effect of each function evaluation into account and tend to ignore the unexplored areas, thus limiting the scope of the search. This is a well-known drawback and different heuristic adjustment to the utility function have been proposed to compensate for it. Kushner (1964), for example, proposed the following modified utility function in the context of a one-dimensional search algorithm:

$$u(x; \zeta_t) = P(f(x) \geq Z_t + \varepsilon_t), \tag{4}$$

where $0 \leq \varepsilon_t < \infty$ is a parameter of the algorithm. When ε_t is large, the search is conducted in areas with large variance and when $\varepsilon_t \to 0$ the search becomes local around the current best value, Z_t. Algorithms of Žilinskas (1992) and Mockus (1989) use essentially the same heuristic rule in order to make the multi-dimensional search more global. Another approach is to "blow up" the variance of distribution $F(y)$ by a constant (see, e.g., Törn and Žilinskas 1989).

We propose that an alternative utility function, denoted by $z^*(x; \zeta_t)$, be used in the above context. To specify $z^*(x; \zeta_t)$, we first re-define the optimization criterion to include the cost of computations. Our objective then is to maximize the expected total "reward" of the algorithm (that takes the cost of computations into account) instead of finding the best function value after a fixed number of iterations of the algorithm. More specifically, let $c(x)$ denote the cost of evaluating the function at x and other auxiliary computations. Then our objective is to maximize

$$E\left[\max_{1 \leq i \leq T}\{f(x_i)\} - \sum_{i=1}^{T} c(x_i)\right] = E\left[Z_T - \sum_{i=1}^{T} c(x_i)\right], \tag{5}$$

where T is a stopping time of the optimization algorithm, determined by the algorithm itself.

In this context, $z^*(x; \zeta_t)$ is defined as the solution to the following equation:

$$E[f(x) - z]^+ = c(x), \tag{6}$$

where the distribution of $f(x)$ is assumed to be $F(y; x, \zeta_t)$. The existence and uniqueness of $z^*(x; \zeta_t)$ are guaranteed under mild conditions. $z^*(x; \zeta_t)$ is the value at which the marginal benefit of evaluating the function at x is offset by the cost of computation.

In Section 3 of the paper, we describe a non-adaptive setting in which $z^*(x; \zeta_t) = z^*(x)$ provides the perfect guide for the search and the optimal search strategy is as follows: The utility function $z^*(x)$ is calculated at all points where the function is not yet evaluated. If $\max\{z^*(x); x \in A - \{x_1, \ldots, x_t\}\}$ is greater than the current highest value observed, i.e., Z_t, the next point for function evaluation is the maximizer of $\{z^*(x); x \in A - \{x_1, \ldots, x_t\}\}$. Otherwise the search is stopped at t, i.e., $T = t$. This search strategy optimizes (5). (In Section 3 we describe another context in which using $z^*(x)$ as a utility function yields an optimal search strategy.)

Therefore, in the above non-adaptive setting, $z^*(x)$ is a truly global utility function that takes into account the effect of selecting x as the next point for function evaluation on the overall performance of the optimization algorithm. In cases where criterion (5) is not adopted as the objective to be optimized, the “cost” $0 < c < \infty$ can be viewed as a parameter of the optimization algorithm, similar to ε above.

Given the more global character of the utility function $z^*(x; \zeta_t)$ when compared to the other utility functions specified above, we expect that by using $z^*(x; \zeta_t)$ one can improve a number of existing optimization algorithms. To illustrate, we give an example in Section 4 where such a modification of Žilinskas multi-dimensional algorithm UNT is performed.

The rest of the paper is organized as follows: in Section 2 we describe the statistical approach to global optimization. We discuss the new utility function $z^*(x; \zeta_t)$ in Section 3 and show how to apply this policy to the algorithm UNT in Section 4. We present computational results in Section 5 and state conclusions in Section 6.

2. Statistical global optimization

In statistical global optimization, one defines a statistical model that captures the global behavior of the objective function without pre-defining its local behavior. The model is updated as the function is evaluated at new points. In this context, it is possible to define the “average efficiency” of the algorithm as the expected optimal value found by the algorithm, given the model of the objective function.

It is often assumed that $f(x)$ can be modeled as a realization of a certain stochastic process on A. At each iteration of the optimization algorithm, the conditional distribution of the function values, $F(y; x, \zeta_t)$, is determined based on the set of all previous observations $\zeta_t = \{(x_i, y_i), i = 1, \ldots, t\}$. Gaussian models of the objective function $f(x|\zeta_t) \sim N(\mu(x), \sigma(x))$ with conditional mean $\mu(x|\zeta_t)$ and conditional variance $\sigma^2(x|\zeta_t)$ are commonly used. The one-dimensional Brownian motion model was proposed by Kushner (1964) and multi-dimensional algorithms are presented in Törn and Žilinskas (1989) and Mockus (1991, 1994).

Given the conditional distribution function $F(y; x, \zeta_t)$, an auxiliary problem is solved in order to find the point that maximizes a certain rational utility function $u(x; \zeta_t)$. A popular approach is to maximize a one-step expected reward

$$u(x; \zeta_t) = E[f(x) - Z_t]^+, \tag{7}$$

or the probability of selecting a point with a function value better than the current maximum Z_t:

$$u(x; \zeta_t) = P(f(x) \geq Z_t). \tag{8}$$

When utility functions (7) and (8) are used, selected points for function evaluation tend to be close to the existing good points. For example, utility function (7) leads to a policy that gives a large expected increase of the function value at the next immediate step, but this policy may not perform well in the long run. In other words, they are “myopic” utility functions.

Various heuristics are proposed in order to make the search more global. Kushner (1964) suggested using

$$u(x) = P(f(x) \geq Z_t + \varepsilon_t), \tag{9}$$

where parameter $0 \leq \varepsilon_t < \infty$ defines a trade-off between local and global search. When ε_t is large, the search is conducted in areas with large variance; when $\varepsilon_t \to 0$ search becomes local around the current best value Z_t. Therefore, ε should be chosen large at the beginning of the search and small at the end. Although Žilinskas (1989) gives more detailed recommendations based on extensive experimental research, the exact choice of ε_t is left with the user.

After the utility function is defined, the next point for function evaluation is chosen as the solution of an auxiliary optimization problem

$$x_{t+1} = \arg \max_{x \in A - \{x_1, \dots, x_t\}} u(x; \zeta_t). \tag{10}$$

In the one-dimensional case, the Brownian motion on the interval $[a, b] \subset R$ can be decomposed into independent processes in each of the subintervals between previously sampled points $[x_i, x_{i+1}]$. Therefore,

$$\max_{x \in A - \{x_1, \dots, x_t\}} u(x; \zeta_t) = \max_{i=1, \dots, t-1} \{ \max_{x \in (x_i, x_{i+1})} u(x; \zeta_t)\}, \tag{11}$$

where the maximum of $u(x; \zeta_t)$ in each of the subintervals can be found analytically for $u(x; \zeta_t)$ defined according to (9).

In the multi-dimensional case, the computation of the full conditional distribution $F(y; x, \zeta_t)$ becomes very expensive because the conditional distribution at any point x depends on all previous observations. Approximate models are developed that use the information from the neighboring points only. The maximum value of $u(x; \zeta_t)$ is usually found via a combination of random and local searches or via optimization over heuristically chosen subsets – such as line search between previous sampled points (see, e.g., Stuckman 1988; Stuckman and Stuckman 1993).

3. An alternative utility function

Our approach is based on using an alternative utility function in the above statistical global optimization context. To define this utility function, we first modify the optimization criterion by introducing the cost of computations into the objective function.

Let $c(x)$ denote the computational cost of evaluating the function at x and other auxiliary computations. Then our objective is to maximize

$$E\left[\max_{1 \leq i \leq T} \{f(x_i)\} - \sum_{i=1}^{T} c(x_i)\right] = E\left[Z_T - \sum_{i=1}^{T} c(x_i)\right], \tag{12}$$

where T is an appropriate stopping time of the optimization algorithm. This criterion was first suggested by Tang (1994) in the context of partitioned random search.

The optimal solution to the above problem in the adaptive case is not known. We consider the simpler non-adaptive case – when the model of function values is not updated – and use the optimal solution of the non-adaptive case to construct a heuristic solution for the original adaptive one.

Let $F(y; x)$ be the distribution of $f(x)$. Assume that the current maximum value is $Z_t = z$. Then, we can define a one-step expected reward of evaluating the function at x as

$$R(x; z) = E[(f(x) - z)^+] - c(x). \tag{13}$$

$R(x; z)$ is a non-increasing function of z, and for large enough z it becomes negative, i.e., $R(x; z) < 0$. We define $z^*(x)$ as the unique solution to the following identity:

$$R(x; z^*(x)) = 0.$$

In other words, $z^*(x)$ is the value for which the expected marginal benefit of evaluating the function at x is offset by the cost of computation. Under the following assumptions, this value is a perfect guide for the search for the optimum.

- The set A is finite.
- The distribution of the function value at x, $F(y; x)$, is known.
- The distributions $F(y; x)$, $x \in A$, are independent.

Assume that at each decision moment k we can do one more function evaluation or stop the search. Weitzman (1979) considered this problem in some more generality in the context of an optimal search for the best economic alternative and proved that the following index policy based on the values $z^*(x)$ is the optimal policy (i.e., optimal solution to (12)).

- **Stopping rule:** Stop the search if $Z_t \geq \max \{z^*(x); x \in A - \{x_1, \ldots, x_t\}\}$, or if all points are examined, i.e., $\{x_1, \ldots, x_t\} = A$.
- **Selection rule:** If the stopping condition is not satisfied, evaluate the function at the maximizer of $\{z^*(x); x \in A - \{x_1, \ldots, x_t\}\}$.

In other words, in the non-adaptive finite case, $z^*(x)$ is an optimal utility function. The assumption of finiteness of A is not very restrictive: For any bounded subset of $\mathbf{R}^d$, we can construct a finite mesh that approximates the set very closely; hence, we can approximate the original optimization problem by an optimization problem on the approximating finite mesh.

In the adaptive case, where, at each step of the optimization algorithm, the model of the function values is updated, we evaluate the utility function at each step based on the updated model, i.e., we evaluate $z^*(x;\zeta_t)$ based on $F(y;x,\zeta_t)$. As we stated before, the resulting policy in the non-adaptive case is not necessarily optimal. However, $z^*(x;\zeta_t)$ is a somewhat more “global” utility function when compared to the utility functions defined in Section 2, and we expect that using $z^*(x;\zeta_t)$ will improve the performance of the statistical global optimization algorithms discussed in Section 2. To test this assertion we apply the $z^*(x;\zeta_t)$ utility function to an existing statistical global optimization algorithm in the next section.

REMARK. It is worth noting another context in which $z^*(x)$ is an optimal utility function. Consider the following non-adaptive setting: Assume that sampling/function evaluation at each $x \in A$ yields a *random* reward $f(x)$ with distribution $F(y;x)$ at a cost $c(x)$ (A is assumed to be a compact set, not necessarily finite). In this case, unlike what we discussed above, multiple sampling/function evaluation at one point is possible, resulting in independent and identically distributed rewards. Castanon et al. (1996) proved that, under suitable conditions, the optimal policy (optimizing (12)) is to sample at the point that has the largest $z^*(x)$ and to stop sampling the moment a value larger than the largest $z^*(x)$ is obtained. Moreover, they showed that the optimal expected reward in this context is the largest $z^*(x)$.

4. Application to multi-dimensional global optimization

As already mentioned, we propose to use $z^*(x;\zeta_t)$ as a utility function in the context of statistical global optimization algorithms; no other changes in the local or global stages of the algorithm are required. To illustrate the approach, we use $z^*(x;\zeta_t)$ as a utility function in the context of the multi-dimensional axiomatic statistical optimization algorithm UNT (Žilinskas, 1992).

We describe the algorithm briefly: The statistical model is a Gaussian field on the set A. Therefore, the distribution of the function value $f(x;\zeta_t)$ at stage t of the algorithm is Gaussian for all $x \in A$:

$$f(x;\zeta_t) \sim N(\mu(x;\zeta_t), \sigma(x;\zeta_t)). \tag{14}$$

To simplify the computations, the following approximate updating of the conditional density was proposed by Žilinskas: The mean and variance of $f(x;\zeta_t)$ are computed based on the r nearest neighbors of x (r is a parameter of the algorithm) as follows:

$$\mu_t(x;\zeta_t) = \sum_{i=1}^{t} y_i w_i(x;\zeta_t), \tag{15}$$

$$\sigma_t(x;\zeta_t) = \gamma_t \sum_{i=1}^{t} ||x - x_i||\ w_i(x;\zeta_t),$$

$$w_i(x;\zeta_t) = d(x,x_i)/\sum_{j\in\mathcal{N}(x)} d(x,x_j), \quad i \in \mathcal{N}(x), \tag{16}$$

$$= 0, \quad \text{otherwise},$$

$$d(x,x_i) = \exp(-v||x-x_i||^2)/||x-x_i||,$$

where $\mathcal{N}(x)$ is the set of indices of the r nearest neighbors of x; $w_i(x;\zeta_t)$ defines the relative weight of observation i at point x, and v and γ_t are fixed parameters of the model that need to be estimated (see Žilinskas (1992) for details).

Therefore, $\mu(x)$ is the weighted average of the existing function values $\{y_i, i = 1,\dots,t\}$, weighted proportionally to an appropriately defined distance 0 between x and x_i's; $\sigma(x)$ depends on the distance of x from x_i's in such a way that it increases when moving further away from them.

Based on the above model, Žilinskas proposed the following heuristic optimization algorithm:

1. Sample N_0 initial samples randomly from A and estimate the parameters of the model γ_t and v.
2. Use a random search approach to find (approximately) the maximizer of the utility function $u(x;\zeta_t) = P(f(x) > Z_t + \varepsilon)$; denote this value by x^*.
3. Evaluate $y(x^*)$.
4. If $y(x^*)$ and values at K points closest to x^* form a concave set, assume that the area around a local maximum is found, and generate a local search starting from x^*. (K is a parameter of the algorithm.)
5. If a specified limit on the total number of points at which the function is evaluated or on the number of local searches is reached, stop.
   Otherwise, go to step (2).

We propose to modify the above algorithm by changing the utility function to $u(x) = z^*(x)$. We assume that $c(x) = c$, i.e., the cost of computations at all points is identical, and use c as a parameter of the modified algorithm. In order to provide a fair comparison between the two algorithms, we use the same stopping criterion and local search procedures as the original algorithm.

Given $f(x;\zeta_t) \sim N(\mu(x;\zeta_t), \sigma(x;\zeta_t))$, we evaluate the utility function $z^*(x;\zeta_t)$ as follows: $z^*(x;\zeta_t)$ is the solution of the equation

$$I(z;\mu,\sigma) = E\,[Y-z]^+ = c,$$

where the density of Y is $f(x;\zeta_t)$. Note that $I(z;\mu,\sigma) = E\ [Y-z]^+$ can be evaluated using a standard normal density:

$$\begin{aligned} I(z;\mu,\sigma) &= E\ [Y-z]^+ = \sigma E\ ((Y-\mu)/\sigma - (z-\mu)/\sigma) \\ &= \sigma I((z-\mu)/\sigma);0,1), \end{aligned} \tag{17}$$

where

$$I(z;0,1) = \int_z^\infty (1-\Phi(p))dp = \phi(z) - z(1-\Phi(z)) \tag{18}$$

(Φ, ϕ are, respectively, the distribution and density functions of the standard normal distribution.)

Therefore,

$$z^*(x) = I^{-1}(c;\mu(x),\sigma(x)) = \mu(x) + \sigma(x)\ I^{-1}(c/\sigma(x);0,1) \tag{19}$$

(see Rosenfield (1983) for a detailed discussion of computing the value $z^*(x)$ for normal distribution).

We can tabulate values of the monotone function $I^{-1}(c;0,1)$ in advance on a discretized subset of $[0,\infty)$ of the form $\{n\delta c; n \geq 0, \delta > 0\}$. Therefore, computing $z^*(x)$ will require only one lookup to a sorted table $\{n\delta c, z^*(n\delta c;0,1)\}$.

5. Experimental results

In this section we provide experimental results to compare the performance of the UNT optimization algorithm when (a) the original utility function is used, and (b) this utility function is replaced by $z^*(x;\zeta_t)$.

We modify the original FORTRAN algorithm by computing the utility function according to (19). We use the local search procedures and the stopping rule that are provided by the algorithm UNT without any modification.

We choose a sampling cost $c = 0.001$. The maximum number of local minima is set to 20. The maximum number of points at which the function is evaluated is chosen to be 5000 (the stopping usually occurred earlier when the specified number of local maxima were reached).

We ran the algorithm with different numbers of initial random points $N_0 = 30, 100,$ and 1000. For each pair of runs, we started both methods with the same set of initial random points. For each experiment, we report the average results of 100 independent replications – each replication starting from a new set of initial random numbers. Test problems are taken mostly from Aluffi-Pentini et al. (1988) and are described in Appendix A.

We compare the maximum values Z_T found by the two methods (Table 1) and the total number of iterations T (Table 2). We also compute the probability that the modified method (using $z^*(x;\zeta_t)$) gives a strictly better result than UNT in terms of the maximum value or the number of iterations (Table 3).

Table 1. Comparing UNT and z^* methods: Final maximal values.

Function	d	$N_0 = 30$		$N_0 = 100$		$N_0 = 1000$	
		UNT	z^*	UNT	z^*	UNT	z^*
4-order poly	1	0.351	0.351	0.352	0.352	0.352	0.352
Gold 6 order poly	1	−7.002	−7.004	−7.002	−7.002	−7.0	−7.0
Shubert	1	12.866	12.871	12.871	12.871	12.871	12.871
4 order poly	2	0.303	0.133	0.266	0.270	0.335	0.336
1 row of local min	2	−0.141	−0.263	−0.141	−0.117	−0.027	−0.010
6-hump camel	2	1.016	0.996	1.007	1.009	1.028	1.029
Shubert, $\beta = 0$	2	141.3	179.8	184.07	184.10	186.48	186.23
Shubert, $\beta = 0.5$	2	124.4	160.7	168.3	168.7	181.4	180.2
Shubert, $\beta = 1$	2	100.1	152.9	155.3	153.7	180.8	177.5
3 ill-cond min, $A = 10$	2	−177.8	−1040	−13.16	−13.16	−0.334	−0.334
3 ill-cond min, $A = 10^2$	2	−32.059	−115.03	−21.227	−21.227	9.500	9.500
Goldstein-Price	2	−4.318	−8.033	−4.788	−4.788	−3.267	−3.271
Branin	2	−0.424	−0.469	−0.432	−0.427	−0.405	−0.400
Levy-Mont, 1, 10	2	−0.055	−0.022	−0.021	−0.017	−0.010	−0.006
Levy-Mont, 3, 10	2	−0.034	−0.065	−0.035	−0.032	−0.017	−0.011
Small global min	2	−640.6	−1843.1	−896.0	−896.0	−156.1	−156.1
Goldstein-Price	2	−4.318	−8.730	−5.129	−5.129	−3.369	−3.369
Rasn	2	1.868	1.951	1.975	1.955	1.994	1.978
Hartman	3	3.813	3.853	3.850	3.859	3.859	3.859
Levy-Mont, 1, 10	3	−0.595	−0.258	−0.167	−0.126	−0.089	−0.070
Levy-Mont, 3, 10	3	−0.163	−0.255	−0.161	−0.119	−0.085	−0.024
Shekel, $M = 5$	4	1.407	4.934	5.771	5.201	7.273	6.114
Shekel, $M = 7$	4	1.822	6.830	7.315	5.917	7.713	6.687
Shekel, $M = 10$	4	1.771	7.192	7.555	6.235	8.126	6.770
Levy-Mont, 1, 10	4	−1.537	−0.818	−0.678	−0.576	−0.236	−0.243
Levy-Mont, 3, 10	4	−0.828	−0.686	−0.561	−0.459	−0.262	−0.097
Rasn	4	1.334	1.481	1.437	1.581	1.711	1.730
Levy-Mont, 2, 10	5	−30.089	−24.976	−23.222	−24.792	−10.497	−10.155
Levy-Mont, 3, 5	5	−0.536	−0.367	−0.374	−0.264	−0.216	−0.032
1 cusp-shaped min	5	−89.822	−61.949	−59.702	−56.060	−46.628	−40.695
Small global min	5	−1117	−982.5	−738.1	−717.0	−386.1	−367.3
Hartman	6	2.730	3.120	3.133	3.226	3.210	3.268
Levy-Mont, 3, 5	6	−0.880	−0.594	−0.604	−0.433	−0.367	−0.055
Levy-Mont, 3, 5	7	−1.481	−0.996	−0.973	−0.656	−0.556	−0.086
Levy-Mont, 2, 10	8	−83.165	−68.691	−66.170	−65.751	−37.008	−33.276
Levy-Mont, 2, 10	10	−108.4	−93.996	−85.866	−82.972	−59.787	−59.651
Shekel, $M = 10$	10	0.079	0.238	0.155	0.262	0.278	0.350
Rasn	10	0.578	0.673	0.683	0.849	0.892	0.968

Table 2. Comparing UNT and z^* methods: Number of iterations.

Function	d	$N_0 = 30$		$N_0 = 100$		$N_0 = 1000$	
		UNT	z^*	UNT	z^*	UNT	z^*
4-order poly	1	110.4	141.0	262.6	263.9	1432	1358
Gold 6 order poly	1	123.3	116.5	185.3	188.0	1174	1149
Shubert	1	343.6	166.6	237.5	256.9	1170	1192
4 order poly	2	196.0	131.1	310.9	303.4	1843	1735
1 row of local min	2	192.8	156.9	289.3	279.2	1305	1199
6-hump camel	2	196.0	176.6	366.1	340.0	1893	1513
Shubert, $\beta = 0$	2	354.7	387.3	548.0	527.1	1669	1542
Shubert, $\beta = 0.5$	2	355.6	387.1	545.8	510.1	1582	1450
Shubert, $\beta = 1$	2	350.5	350.8	496.3	442.6	1523	1434
3 ill-cond min, $A = 10$	2	304.4	243.9	423.8	423.8	2229	2229
3 ill-cond min, $A = 10^2$	2	302.4	245.0	425.2	425.2	2228	2228
Goldstein-Price	2	240.5	236.2	451.1	451.1	2319	2327
Branin	2	242.2	199.7	311.2	295.9	1384	1255
Levy-Mont, 1, 10	2	375.1	349.2	513.4	439.7	1508	1291
Levy-Mont, 3, 10	2	138.3	103.9	208.1	185.1	1412	1096
Small global min	2	129.9	102.9	195.3	195.3	1183	1183
Goldstein-Price	2	241.1	223.6	466.9	466.9	2344	2344
Rasn	2	338.0	313.8	567.3	392.6	1414	1378
Hartman	3	389.5	320.6	385.8	374.7	1312	1258
Levy-Mont, 1, 10	3	346.5	376.1	498.6	490.5	1933	1686
Levy-Mont, 3, 10	3	343.2	124.7	236.0	210.7	1226	1131
Shekel, $M = 5$	4	378.1	196.1	317.3	252.2	1244	1230
Shekel, $M = 7$	4	387.1	213.9	316.1	259.3	1255	1219
Shekel, $M = 10$	4	377.6	236.5	347.2	282.1	1265	1224
Levy-Mont, 1, 10	4	378.2	389.3	528.4	519.3	1863	1645
Levy-Mont, 3, 10	4	378.1	186.0	271.1	262.6	1263	1167
Rasn	4	377.1	414.3	549.6	608.8	2319	1597
Levy-Mont, 2, 10	5	381.3	386.4	508.4	508.5	1851	1812
Levy-Mont, 3, 5	5	376.3	274.7	334.8	323.1	1346	1207
1 cusp-shaped min	5	377.4	283.2	364.6	353.4	1218	1190
Small global min	5	383.1	237.3	311.9	310.2	1256	1266
Hartman	6	378.2	439.8	551.9	551.8	1443	1436
Levy-Mont, 3, 5	6	375.5	315.2	416.8	394.2	1475	1271
Levy-Mont, 3, 5	7	365.3	315.7	466.0	457.2	1606	1365
Levy-Mont, 2, 10	8	346.9	338.8	485.8	487.1	2135	2103
Levy-Mont, 2, 10	10	331.4	301.0	451.9	451.8	2175	2139
Shekel, $M = 10$	10	346.6	479.6	503.2	583.0	1656	1570
Rasn	10	332.4	361.2	462.9	599.0	2244	1803

Table 3. Comparing UNT and z^* methods: Percentage of runs when z^* performed better than UNT.

Function	d	In $z*$			In iterations		
N_0 :		30	100	1000	30	100	1000
4-order poly	1	46	30	36	13	47	91
Gold 6 order poly	1	39	45	96	57	50	68
Shubert	1	81	73	96	100	38	34
4 order poly	2	10	43	44	90	58	83
1 row of local min	2	29	52	79	70	53	94
6-hump camel	2	29	43	53	59	66	100
Shubert, $\beta = 0$	2	94	46	42	36	55	75
Shubert, $\beta = 0.5$	2	85	52	40	38	59	66
Shubert, $\beta = 1$	2	85	53	27	47	68	76
3 ill-cond min, $A = 10$	2	26	0	0	75	0	0
3 ill-cond min, $A = 10^2$	2	32	0	0	75	0	0
Goldstein-Price	2	15	0	3	51	0	23
Branin	2	17	56	89	72	66	95
Levy-Mont, 1, 10	2	69	53	56	60	72	97
Levy-Mont, 3, 10	2	25	48	68	83	88	100
Small global min	2	21	0	0	75	0	0
Goldstein-Price	2	12	0	0	48	0	0
Rasn	2	83	58	39	61	92	60
Hartman	3	95	87	61	73	51	76
Levy-Mont, 1, 10	3	73	59	53	28	53	96
Levy-Mont, 3, 10	3	26	61	98	99	68	96
Shekel, $M = 5$	4	98	41	31	97	73	58
Shekel, $M = 7$	4	96	34	33	94	70	64
Shekel, $M = 10$	4	98	41	29	91	69	67
Levy-Mont, 1, 10	4	77	54	48	41	53	97
Levy-Mont, 3, 10	4	53	64	91	96	58	90
Rasn	4	70	78	58	29	23	100
Levy-Mont, 2, 10	5	63	49	50	40	56	60
Levy-Mont, 3, 5	5	77	70	100	78	58	87
1 cusp-shaped min	5	93	61	70	86	50	69
Small global min	5	59	31	45	90	40	41
Hartman	6	94	79	85	30	47	46
Levy-Mont, 3, 5	6	74	68	100	64	58	87
Levy-Mont, 3, 5	7	78	76	99	62	49	89
Levy-Mont, 2, 10	8	65	51	58	47	52	63
y Levy-Mont, 2, 10	10	55	52	51	57	49	61
Shekel, $M = 10$	10	97	76	65	19	39	80
Rasn	10	59	76	71	33	6	99

We observe that the largest improvement is achieved in high-dimensional problems ($d \geq 5$), where the modified method is almost always better than the original UNT algorithm. The difference in performance is less significant in low-dimensional problems and in a number of problems both methods produce similar results.

Our explanation for the above performance results is that in the higher-dimensional problems the adaptation process takes place much more slowly than in the low-dimensional problems, i.e., the conditional distribution at each point changes more slowly in high-dimensional problems. As a result, in high-dimensional problems the setting is closer to a non-adaptive case and the merit of the new utility function more apparent.

REMARK. Note that the goal of the numerical experiments in this section is to evaluate the impact of using the utility function proposed in this paper when compared to using the original utility function of the *UNT* algorithm. Therefore, we use all test functions for this purpose only. Some of these problems can be solved more efficiently by methods from other classes of optimization algorithms, such as multistart or line search.

6. Conclusions

We proposed a new utility function in order to improve the performance of statistical global optimization algorithms. This utility function takes the overall goal of optimization into account, is not myopic, and is an optimal utility function in a non-adaptive setting. Our computational results, in the context of an existing statistical global optimization algorithm, suggest that the advantage of using the new utility function is more apparent in high-dimensional problems.

The new utility function can also be applied to a number of other optimization methods that include a selection stage, such as one-dimensional (Kushner, 1964; Žilinskas, 1992), multi-dimensional (Mockus, 1989) statistical algorithms, adaptive partitioning (Pintér, 1996; Tang, 1994; Norkin et al., 1994), global line search heuristics (Stuckman, 1988; Stuckman and Stuckman, 1993; Streltsov and Muchnik, 1996). The use of the new utility function in these contexts and the evaluation of its impact is a subject for future research.

Acknowledgments

We would like to acknowledge Ilya Muchnik of DIMACS for many fruitful discussions, and two anonymous reviewers for their remarks that helped to clarify the exposition.

Appendix A: Test Functions

- 4-order polynomial, $d = 1, 2$
  $d = 1$: $f(x) = ((0.25x_1^2 - 0.5)x_1 + 0.1)x_1$
  $d = 2$: $f(x) = ((0.25x_1^2 - 0.5)x_1 + 0.1)x_1 + 0.5x_2^2$
  $-10 \le x_i \le 10, \quad i = 1, \dots, d.$
- Goldstein 6-order polynomial, $d = 1$
  $f(x) = ((x_1^2 - 15)x_1^2 + 27)x_1^2 + 250$
  $-4 \le x_i \le 4, \quad i = 1, \dots, d$
- Shubert, $d = 1, 2$; $\beta = 0, 0.5, 1$
  $d = 1$: $f(x) = \sum_{i=1}^{5} i \cos[(i + 1)x_1 + i]$
  $d = 2$:
  $$\begin{aligned} f(x) =& \beta((x_1 + 1.4251284)^2 + (x_2 + 0.8003211)^2) + \\ &+ \sum_{i=1}^{5} i \cos[(i + 1)x_1 + i] * \sum_{i=1}^{5} i \cos[(i + 1)x_2 + i] \end{aligned}$$
  $-10 \le x_i \le 10, \quad i = 1, \dots, d$
- A function with a single row of local minima, $d = 2$
  $f(x) = 0.5(0.1x_1^2 + 1 - \cos(2x_1)) + x_2^2$
  $-15 \le x_1 \le 25, -5 \le x_2 \le 15$
- 6-hump camel function, $d = 2$
  $f(x) = ((x_1^2/3 - 2.1)x_1^2 + 4)x_1^2 + x_1x_2 + 4(x_2^2 - 1)x_2^2$
  $i - 4 \le x_i \le 4 - i, \quad i = 1, \dots, d$
- A function with 3 ill-conditioned minima, $d = 2$; $A = 10, 10^2$
  $f(x) = Ax_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + (x_1^2 + x_2^2)^4/A$
  $-10^i \le x_i \le 10^i, \quad i = 1, \dots, d$
- Goldstein-Price, $d = 2$
  $$\begin{aligned} f(x) =& [1 + (x_1 + x_2 + 1)^2(36 - 20(x_1 + x_2 + 1) + 3(x_1 + x_2 + 1)^2)] \\ & [30 + (2x_1 - 3x_2)^2(18 - 16(2x_1 - 3x_2) + 3(2x_1 - 3x_2)^2)] \end{aligned}$$
  $-2.5 \le x_i \le 2, \quad i = 1, \dots, d$
- Branin, $d = 2$
  $f(x) = [x_2 - 1.275(\frac{x_1}{\pi})^2 + \frac{5}{\pi}x_1 - 6]^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$
  $-5 \le x_1 \le 10, \quad 0 \le x_2 \le 15$
- Levy-Montalvo, type=1,2; $R = 10$
  $y_i = 1 + (x_i - 1)/4, \quad i = 1, \dots, d$ for type $= 1$
  $y_i = x_i, \quad i = 1, \dots, d$ for type $= 2$
  $$\begin{aligned} f(x) =& \pi/d\Bigg[10\sin^2(\pi y_1) + (y_d - 1)^2 \\ &+ \sum_{i=2}^{d}(y_{i-1} - 1)^2(1 + 10\sin^2(\pi y_i))\Bigg] \end{aligned}$$

$-R \leq x_i \leq R, \quad i = 1, \ldots, d$

- Levy-Montalvo, type = 3; $R = 5, 10$

$$f(x) = 0.1 \Bigg[\sin^2(3\pi x_1) + (x_d - 1)^2(1 + \sin^2(2\pi x_d)) + \sum_{i=2}^{d} (x_{i-1} - 1)^2 (1 + \sin^2(3\pi x_i)) \Bigg]$$

$-R \leq x_i \leq R, \quad i = 1, \ldots, d$

- A function with a small-attraction-region global minimum, $d = 2, 5$

$$f(x) = \sum_{i=1}^{d} x_i^2 - I \left\{ \sum_{i=2}^{d} x_i^2 + (x_1 - R)^2 < 0.98 \right\} \times (10 + R^2) \exp \left\{ \frac{-(\sum_{i=2}^{d} x_i^2 + (x_1 - R)^2)}{1 - \sum_{i=2}^{d} x_i^2 + (x_1 - R)^2} \right\}$$

$d = 2: \quad R = 100; \; -1000 \leq x_i \leq 1000, \quad i = 1, \ldots, d$
$d = 5: \quad R = 10; \; -100 \leq x_i \leq 100, \quad i = 1, \ldots, d$

- Rasn
  $f(x) = \frac{2}{d} \sum_{i=1}^{d} (x_i^2 - \cos(18 x_i))$
  $-1 \leq x_i \leq 1, \quad i = 1, \ldots, d$
- Hartman $d = 3, 6$
  $f(x) = -\sum_{i=1}^{4} c_i \exp\{-\sum_{j=1}^{d} a_{ij} (x_j - p_{ij})^2\}$
  $d = 3$: $C = (1, 1.2, 3, 3.2)$, $P = ((0.3689, 0.117, 0.2673), (0.4699, 0.4387, 0.7470), (0.1091, 0.8732, 0.5547), (0.03815, 0.5743, 0.8828))$,
  $A = ((3, 10, 30), (0.1, 10, 35), (3, 10, 30), (90.1, 10, 35))$.
  $d = 6$: $C = (1, 1.2, 3, 3.2)$, $P = ((0.1312, 0.1696, 0.5569, 0.0124, 0.8283, 0.5886)$, (0.2329,0.4135,0.8307,0.3736,0.1004,0.9991), (0.2348,0.1451, 0.3522,0.2883,0.3047,0.6650), (0.4047,0.8828,0.8732,0.5743,0.1091, 0.0381)), $A = ((10, 3, 17, 3.5, 1.7, 8), (0.05, 10, 17, 0.1, 8, 14), (3, 3.5, 1.7, 10, 17, 8), (17, 8, 0.05, 10, 0.1, 14))$
  $0 \leq x_i \leq 1, \quad i = 1, \ldots, d$
- Shekel, $d = 4$; $M = 5, 7, 10$
  $f(x) = \sum_{i=1}^{M} \frac{1}{(x - a_i)(x - a_i)^T + c_i}$
  $C = (0.1, 0.2, 0.2, 0.4, 0.4, 0.6, 0.3, 0.7, 0.5, 0.5)$, $A = ((4, 4, 4, 4), (1, 1, 1, 1), (8, 8, 8, 8), (6, 6, 6, 6), (3, 7, 3, 7), (2, 9, 2, 9), (5, 5, 3, 3), (8, 1, 8, 1), (6, 2, 6, 2), (73.6, 7, 3.6))$
  $0 \leq x_i \leq 10, \quad i = 1, \ldots, d$
- A single cusp-shaped min, $d = 5$
  $f(x) = (\sum_{i=1}^{d} i x_i^2)^{1/4}$
  $-20000 \leq x_i \leq 10000, \quad i = 1, \ldots, d$

References

Aluffi-Pentini, F., Parisi, V. and Zirilli, F. (1988), A Global Optimization Algorithm Using Stochastic Differential Equations. *ACM Transactions on Mathematical Software* 14(4), 345–365.

Betrò, B. (1991), Bayesian Methods in Global Optimization, *J. of Global Optimization* 1(1), 1–14.

Betrò, B. and Schoen, F. (1992), Optimal and Sub-optimal Stopping Rules for Multistart Algorithm in Global Optimization, *Mathematical Programming* 38, 271–286.

Castañon, D.A., Streltsov, S. and Vakili, P. (1998), Optimality of Index Policies for a Sequential Sampling, to appear in *IEEE Transactions on Automatic Control*.

Kushner, H. (1964), A New Method of Locating the Maximum Point of an Arbitrary Multipeak Curve in the Presence of Noise, *Transactions of the ASME, Series D, J. of Basic Engineering* 86, 97–105.

Mockus, J. (1989), *Bayesian Approach to Global Optimization*. Kluwer, Dordrecht.

Mockus, J. (1994), Application of Bayesian Approach to Numerical Methods of Global and Stochastic Optimization. *J. of Global Optimization* 4(4), 347–356.

Mockus, J. and Mockus, L. (1991), Bayesian Approach to Global Optimization and Application to Multiobjective and Constrained Problems. *J. of Optimization Theory and Applications* 70(1), 157–172.

Norkin, V., Ermoliev, Yu. and Ruszczynski, A. (1994), On Optimal Allocation of Indivisibles under Uncertainty, *Working Paper WP-94-21, IIASA*.

Pintér, J. (1996), *Global Optimization in Action. Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications*, Kluwer, Dordrecht.

Rosenfield, D. B. (1983), Optimal Strategies for Selling an Asset, *Management Science* 29(9), 1051–1058.

Streltsov, S. and Muchnik I. (1996), Global Optimization and Line Search, *Working Paper. Boston University*.

Streltsov, S., Vakili, P. and Muchnik I. (1996), Competing intelligent Search Agents in Global Optimization, *Proceedings of NIST Conference "Intelligent Systems: A Semiotic Perspective"*, 293–298.

Stuckman, B. (1988), A Global Search Method for Optimizing Nonlinear Systems, *IEEE Transactions on Systems, Man, and Cybernetics* 18(6), 965–977.

Stuckman, B. and Stuckman, P. (1993), Find the "best" Optimal Control Using Global Search, *Computers & Electrical Engineering* 19(1), 9–18.

Tang, Z. (1994), Adaptive Partitioned Random Search to Global Optimization, *IEEE Transactions on Automatic Control* 32(11), 2235–2244.

Törn, A. and Žilinskas, A. (1989) Global Optimization, *Lecture Notes in Computer Science, 350*, Springer-Verlag.

Weitzman, M. L. (1979), Optimal Search for the Best Alternative, *Econometrica* 47(3), 641–654.

Žilinskas, A. (1992), A Review of Statistical Models for Global Optimization, *J. of Global Optimization* 2(2), 145–153.

Zhigliavskii, A.A. (1991), *Theory of Global Random Search*, Kluwer, Dordrecht.