An Initialization Strategy for High-Dimensional Surrogate-Based Expensive Black-Box Optimization

Rommel G. Regis

Citation:

Regis, R.G. 2013. An Initialization Strategy for High-Dimensional Surrogate-Based Expensive Black-Box Optimization. In: Zuluaga L., Terlaky T. (eds) *Modeling and Optimization: Theory and Applications. Springer Proceedings in Mathematics & Statistics, Vol. 62.* Springer, New York, NY, pp. 51–85.

An Initialization Strategy for High-Dimensional Surrogate-Based Expensive Black-Box Optimization

ROMMEL G. REGIS
Department of Mathematics
Saint Joseph's University
Philadelphia, Pennsylvania 19131, USA
E-mail: rregis@sju.edu

June 23, 2013

Abstract. Surrogate-based optimization methods build surrogate models of expensive black-box objective and constraint functions using previously evaluated points and use these models to guide the search for an optimal solution. These methods require considerably more computational overhead and memory than other optimization methods so their applicability to high-dimensional problems is somewhat limited. Many surrogates, such as radial basis functions (RBF) with linear polynomial tails, require a maximal set of affinely independent points to fit the initial model. This paper proposes an initialization strategy for surrogate-based methods called *Underdetermined Simplex Gradient Descent (USGD)* that uses underdetermined simplex gradients to make progress towards the optimum while building a maximal set of affinely independent points. Numerical experiments on a 72-dimensional groundwater bioremediation problem and on 200-dimensional and 1000-dimensional instances of 16 well-known test problems demonstrate that the proposed USGD initialization strategy yields dramatic improvements in the objective function value compared to standard initialization procedures. Moreover, USGD initialization substantially improves the performance of two optimization algorithms that use RBF surrogates compared to standard initialization methods on the same test problems.

Keywords: Engineering optimization, high-dimensional black-box optimization, simplex gradient, surrogate model, function approximation, radial basis function, expensive function.

1 Introduction

Many engineering optimization problems involve black-box functions whose values are outcomes of computationally expensive simulations. They can be found in engineering design problems in the aerospace and automotive industries and also in parameter estimation problems for mathematical models describing complex physical, chemical and biological systems. These problems are challenging because only a relatively small number of function evaluations can be used to search for an optimal solution, and they are even more challenging when there are large numbers of decision variables and constraints. Hence, surrogates such as kriging models, radial basis functions (RBFs), and linear and quadratic models are widely used to solve these problems. However, surrogate-based methods tend to require considerably more computational overhead and memory than other optimization methods so their applicability to high-dimensional problems with hundreds of decision variables is somewhat limited. Moreover, the ability of surrogates to guide the selection of promising iterates tends to diminish as the problem dimension increases. For instance, kriging-based methods have mostly been applied to problems with less than 15 decision variables. However, there are practical optimization problems that involve a large number of decision variables and constraints (e.g., see Jones (2008)). This paper proposes an initialization strategy for high-dimensional surrogate-based optimization called *Underdetermined Simplex Gradient Descent (USGD)* that uses underdetermined simplex gradients to make progress towards the optimum while building a set of initial function evaluation points. Numerical results show that the proposed USGD strategy generally yields much better objective function values than standard initialization methods on a 72-dimensional groundwater bioremediation management problem and on 200-dimensional and 1000-dimensional instances of 16 well-known test problems. This paper also explores the performance of two surrogate-based optimization algorithms initialized by USGD in comparison to the same algorithms initialized by standard methods on the same problems. The results also show that USGD substantially improves the performance of two optimization algorithms that use RBF surrogates in comparison to standard initialization methods. Hence, the proposed initialization technique facilitates the application of black-box optimization methods to problems with a much larger number of decision variables than those typically considered in the related literature.

The focus of this paper is on the bound constrained black-box optimization problem:

$$\begin{array}{l} \min \ f(x) \\ \text{s.t.} \\ x \in \mathbb{R}^d, \ a \leq x \leq b \end{array} \tag{1}$$

Here, $a, b \in \mathbb{R}^d$, where d is large (possibly in the hundreds or thousands), and f is a deterministic, computationally expensive black-box function. Here, *computationally expensive* means that the evaluation of the objective function completely dominates the cost of the entire optimization process. Moreover, *black-box* means that analytical expressions for the objective or constraint functions are not available, and in many cases, the function values are obtained from time-consuming computer simulations. In addition, in many practical applications, the derivatives of the objective function are not explicitly available.

There are also many practical optimization applications with black-box constraints that arise from expensive simulations. For example, Jones (2008) presented a black-box automotive problem involving 124 decision variables and 68 black-box inequality constraints and Regis (2011, 2012) developed RBF methods that work well on this problem. In general, it is difficult to deal with hundreds of decision variables and many constraints in the computationally expensive setting where only relatively few function evaluations can be carried out. In fact, for high-dimensional problems, many algorithms that employ surrogates either run out of memory or take an enormous amount of time for each iteration and sometimes they do not even make substantial progress over the starting solution. For simplicity, this paper focuses only on bound constrained problems with a large number of decision variables. Future work will develop extensions of the methods proposed in this paper to high-dimensional problems with black-box constraints.

Before proceeding, it is important to clarify that “solving” a high-dimensional and expensive black-box optimization problem does not actually mean finding its global optimum since this is not realistic. Global optimization of inexpensive black-box functions with even a moderate number of dimensions is already computationally challenging. In addition, even convergence to a first-order critical point is hard to achieve for high-dimensional black-box problems when the number of function evaluations is severely limited. In general, algorithms that can be proved to converge to a critical point or to a global minimizer should be preferred. However, another important criterion in practice is how well an algorithm performs when given a severely limited computational budget. Hence, in this paper, algorithms that yield the best objective value after a fixed number of function evaluations and for a wide range of computational budgets and test problems are considered better. Thus, in this context, “solving” a problem simply means providing a good objective function value, in comparison with alternatives, when given a fixed computational budget.

The remaining sections are organized as follows. Section 2 provides a review of literature on black-box optimization methods and surrogate-based methods, including methods for high dimensional problems. Section 3 presents initialization strategies for high-dimensional surrogate-based optimization, including an algorithm that uses underdetermined simplex gradients. Section 4 presents some numerical experiments. Finally, Section 5 presents a summary and some conclusions.

2 Review of Literature

A natural approach for expensive black-box optimization problems is to use surrogate models or function approximation models for the expensive functions. Commonly used surrogate modeling techniques include linear and quadratic polynomials, kriging or Gaussian process models (Sacks et al. 1989, Cressie 1993), radial basis functions (RBFs) (Powell 1992, Buhmann 2003), neural networks (Chambers and Mount-Campbell 2002, Jin et al. 2002) and support vector machines (SVMs) (Vapnik 1995, Loschilov et al. 2012). Kriging is an interpolation method where the observed function values are assumed to be outcomes of a stochastic process. An advantage of kriging interpolation is that there is a natural way to quantify prediction error at points where the black-box function has not been evaluated. A potential disadvantage of kriging is that it is computationally intensive especially in high dimensions since fitting the model requires numerically solving a maximum likelihood estimation problem. Neural networks are also computationally intensive for high dimensional problems since fitting the model also requires numerically solving a nonlinear least squares problem. In contrast, the RBF model in Powell (1992) can be fit by solving a simple linear system with good theoretical properties.

Surrogate-type approaches have been used in optimization for quite some time. For example, linear and quadratic regression models have been used in response surface methodology (Myers and Montgomery 2009). Kriging was used by Jones et al. (1998) to develop the Efficient Global Optimization (EGO) method where the next iterate is a global maximizer of an expected improvement function. The convergence of EGO was proved by Vazquez and Bect (2010) and explored further by Bull (2011). Variants of EGO have been developed by Huang et al. (2006) for stochastic black-box systems and by Aleman et al. (2009) for IMRT treatment planning. Kriging was also used by Villemonteix et al. (2009) to develop a method that uses minimizer entropy to determine new iterates. On the other hand, RBF interpolation was used by Gutmann (2001) to develop a global optimization algorithm where the next iterate is a global minimizer of a measure of bumpiness of the RBF model. This method was also shown to converge to a global minimum by Gutmann (2001). Variants of Gutmann's RBF method have been developed by Björkman and Holmström (2000), Regis and Shoemaker (2007b), Holmström (2008), Jakobsson et al. (2009), Cassioli and Schoen (2011), and Regis and Shoemaker (2012).

A promising class of derivative-free expensive black-box optimization methods are those that utilize interpolation models within a trust-region framework. These methods are meant for unconstrained optimization but they can be easily adapted to handle bound constraints. For example, the DFO method (Conn, Scheinberg and Toint 1997, Conn, Scheinberg and Vicente 2009b), UOBYQA (Powell 2002) and NEWUOA (Powell 2006) use quadratic models while BOOSTERS (Oeuvray 2005, Oeuvray and Bierlaire 2009) and ORBIT (Wild et al. 2008, Wild and Shoemaker 2011) use RBFs. Global convergence of a class of derivative-free trust region methods to first- and second-order critical points is established by Conn, Scheinberg and Vicente (2009a).

A widely used approach for derivative-free black-box optimization is pattern search (Torczon 1997), including extensions such as generating set search (GSS) (Kolda et al. 2003) and Mesh Adaptive Direct Search (MADS) (Audet and Dennis 2006, Abramson and Audet 2006). MADS is implemented in the NOMAD software (Le Digabel 2011). When the problems are computationally expensive, these methods are sometimes combined with surrogates. For example, Booker et al. (1999) and Marsden et al. (2004) used kriging with pattern search. Conn and Le Digabel (2011) used quadratic models in MADS. Le Thi et al. (2012) and Rocha et al. (2012) used RBFs while Custódio et al. (2010) used minimum Frobenius norm quadratic models in pattern search. Moreover, pattern search can also be made more efficient for expensive functions by using simplex gradients obtained from previous function evaluations (Custódio and Vicente 2007). Finally, these methods have parallel implementations such as APPSPACK (Kolda and Torczon 2004, Gray and Kolda 2006), HOPSPACK (Plantenga and Kolda 2009) and PSD-MADS (Audet et al. 2008).

Another approach for black-box numerical optimization that is very popular in the engineering optimization community are heuristics (e.g., simulated annealing) and nature-inspired algorithms such as evolutionary algorithms (including genetic algorithms, evolution strategies, evolutionary programming and scatter search) and swarm intelligence algorithms (e.g., particle swarm and ant colony algorithms). Although genetic algorithms are among the popular metaheuristics, some researchers in evolutionary computation have noted that evolution strategies and evolutionary programming are more suitable for continuous numerical optimization. In particular, CMA-ES (evolution strategy with covariance matrix adaptation) (Hansen and Ostermeier 2001, Hansen 2006) and its variants and extensions have performed well in the annual benchmarking competition in the evolutionary computation community that includes comparisons with other derivative-free methods such as NEWUOA (Powell 2006). In contrast to pattern search methods and derivative-free trust-region methods, many of these metaheuristics lack theoretical convergence guarantees and might not perform well on computationally expensive problems when given a very limited computational budget. However, as with pattern search methods, many of these metaheuristics can be combined with surrogates, making them competitive with other derivative-free algorithms. For example, kriging has been used in combination with scatter search (Egea et al. 2009) and a ranking SVM surrogate has been used for CMA-ES (Loschilov et al. 2012). Jin (2011) provides a survey on surrogate-assisted evolutionary computation. In addition, these metaheuristics can also be combined with other methods with convergence guarantees. For example, Vaz and Vicente (2007, 2009) combined particle swarm with pattern search. Finally, as with pattern search methods, many of these metaheuristics are parallelizable, making it possible to use them on large-scale problems.

When the optimization problems are high-dimensional, some surrogate-based approaches, particularly those that rely on kriging models, do not seem suitable because they can become computationally prohibitive and require an enormous amount of memory. In particular, most papers on kriging-based methods, including relatively recent ones, involve less than 15 decision variables (e.g., Egea et al. 2009, Viana et al. 2010, Parr et al. 2012, Chen et al. 2012). Hence, in the area of surrogate-based expensive black-box optimization, problems with more than 15 decision variables are sometimes considered high-dimensional and problems with hundreds of decision variables are definitely considered large-scale.

One approach for high-dimensional expensive black-box problems is to use dimension reduction techniques such as sequential bifurcation (Bettonvil and Kleijnen 1997) before applying any optimization methods. Another approach is parallelization (e.g., Audet et al. 2008, García-Palomares et al. 2012). Shan and Wang (2010) provides a survey of some approaches for high-dimensional, expensive, black-box problems. Because many surrogate models such as kriging can be expensive to maintain, it is not surprising that relatively few surrogate-based or model-based methods have been applied to high-dimensional problems. For example, BOOSTERS (Oeuvray and Bierlaire 2009) and DYCORS (Regis and Shoemaker 2013) have been applied to 200-dimensional problems. Moreover, Shan and Wang (2011) developed the RBF-HDMR model and tested it on bound-constrained problems with up to 300 decision variables. Regis (2011, 2012) also developed RBF methods that are suitable for problems with expensive black-box constraints and applied it to a 124-dimensional automotive problem with 68 black-box inequality constraints. Note that what these methods have in common is that they all use RBF surrogates, suggesting that RBF interpolation is promising for high dimensional expensive black-box problems. Hence, pattern search guided by RBF surrogates and RBF-assisted metaheuristics are also potentially promising for high dimensional expensive black-box optimization.

3 Initialization Strategies for High-Dimensional Surrogate-Based Methods

3.1 Standard Initialization Methods

Given a starting point $x_0 \in \mathbb{R}^d$, a standard initialization procedure for surrogate- or model-based methods is to use $d+1$ initial points consisting of x_0 and the points obtained by moving along each of the d positive coordinate directions from x_0. That is, use the points $\{x_0, x_0 + \Delta e_1, \dots, x_0 + \Delta e_d\}$, where Δ is the step size and the vectors $e_1, \dots, e_d$ form the natural basis for $\mathbb{R}^d$, i.e., $e_i = [0, \dots, 0, 1, 0, \dots, 0]^T$, where the 1 is in the ith position. The step size is required to satisfy $\Delta \leq 0.5 \min_{1 \leq i \leq d}(b_i - a_i)$ so that if $x_0 + \Delta e_i$ goes outside the bounds, then it can be replaced by $x_0 - \Delta e_i \in [a, b]$. This procedure is referred to as the *Static Simplex (SS)* initialization procedure.

In the high dimensional and computationally expensive setting, some progress towards the optimum can be made by modifying the Static Simplex procedure so that the center of the algorithm is always moved to the current best point. Below is the pseudo-code for this method, which is referred to as the *Dynamic Simplex (DS)* initialization procedure.

Dynamic Simplex (DS) Initialization Procedure:

Inputs:

(1) Function to minimize: $f : [a, b] \to \mathbb{R}$, where $[a, b] \subseteq \mathbb{R}^d$

(2) Starting point: $x_0 \in [a, b]$

(3) Step size: $\Delta \leq 0.5 \min_{1 \leq i \leq d}(b_i - a_i)$

Outputs: A set of $d+1$ affinely independent points $\mathcal{X} = \{x_0, x_1, \dots, x_d\} \subseteq \mathbb{R}^d$ and their objective function values $\mathcal{F} = \{f(x_0), f(x_1), \dots, f(x_d)\}$.

```
1. (Evaluate Starting Point) Calculate $f(x_0)$. Initialize $\mathcal{X} = \{x_0\}$ and $\mathcal{F} = \{f(x_0)\}$. Also,
   set $x_{\text{best}} = x_0$ and $f_{\text{best}} = f(x_0)$.
2. For $k = 1$ to $d$ do
     (2a) (Select New Point) Let $x_k = x_{\text{best}} + \Delta e_k$. If $x_k$ goes outside the bounds, reset
          $x_k = x_{\text{best}} - \Delta e_k$.
     (2b) (Evaluate Selected Point) Calculate $f(x_k)$ and update $x_{\text{best}}$ and $f_{\text{best}}$.
     (2c) (Update Information) Reset $\mathcal{X} = \mathcal{X} \cup \{x_k\}$ and $\mathcal{F} = \mathcal{F} \cup \{f(x_k)\}$.
   End for.
3. (Return Outputs) Return $\mathcal{X}$ and $\mathcal{F}$.
```

3.2 The Simplex Gradient and Other Preliminaries

Let $\mathcal{X} = \{x_0, x_1, \ldots, x_k\}$ be a set of $k + 1 \leq d + 1$ points in $\mathbb{R}^d$ such that the function values $f(x_0), f(x_1), \ldots, f(x_k)$ are known. If the points in $\mathcal{X}$ are affinely independent, then there exists a linear function (an infinite number if $k < d$) that interpolates the data points $\{(x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_k, f(x_k))\}$. More precisely, if $p(x) = c_0 + c^T x$, where $c = [c_1, \ldots, c_d]^T$, is a linear polynomial in d variables that interpolates these data points, then

$$\begin{bmatrix} 1 & x_0^T \\ 1 & x_1^T \\ \vdots & \vdots \\ 1 & x_k^T \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_d \end{bmatrix} = \begin{bmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_k) \end{bmatrix}.$$

The numerical stability of this linear interpolation depends on the condition number of the $(k + 1) \times (d + 1)$ interpolation matrix

$$L(\mathcal{X}) := \begin{bmatrix} 1 & x_0^T \\ 1 & x_1^T \\ \vdots & \vdots \\ 1 & x_k^T \end{bmatrix}$$

If $\mathcal{X}$ is affinely dependent, then $\text{cond}(L(\mathcal{X})) = \infty$. On the other hand, if $\mathcal{X}$ is affinely independent, then $\text{cond}(L(\mathcal{X})) < \infty$ and the smaller the value the better is the geometry of the data points for linear interpolation. A comprehensive treatment of the geometry of sample sets of points for interpolation (determined and underdetermined cases) and regression (overdetermined case) in derivative-free optimization can be found in Conn, Scheinberg and Vicente (2008a, 2008b) and in Scheinberg and Toint (2010).

The next section presents an algorithm that iteratively constructs a set $\mathcal{X} = \{x_0, x_1, \ldots, x_d\}$ of $d + 1$ affinely independent points in $\mathbb{R}^d$ while making progress in finding a minimum for f and while keeping $\text{cond}(L(\mathcal{X}))$ relatively small. The points in $\mathcal{X}$ are used to initialize a surrogate-based method so it is important that the resulting data points have good geometry for interpolation. Moreover, since function evaluations are expensive, it is also important to make progress on the optimization process even during this initialization phase. That is, it is desirable to have a good value for $\min_{0 \leq i \leq d} f(x_i)$ while keeping $\text{cond}(L(\mathcal{X}))$ relatively small. This is accomplished by using the concept of a simplex gradient, which is defined next.

Let $\mathcal{X} = \langle x_0, x_1, \ldots, x_k \rangle$ be an ordered set of $k + 1$ affinely independent points in $\mathbb{R}^d$, where $k \leq d$. Define

$$S(\mathcal{X}) := [x_1 - x_0 \quad \ldots \quad x_k - x_0] \in \mathbb{R}^{d \times k} \quad \text{and} \quad \delta(\mathcal{X}) := \begin{bmatrix} f(x_1) - f(x_0) \\ \vdots \\ f(x_k) - f(x_0) \end{bmatrix} \in \mathbb{R}^k.$$

When $k = d$ (the determined case), $S(\mathcal{X})$ is invertible and the *simplex gradient with respect to* $\mathcal{X}$, denoted by $\nabla_s f(\mathcal{X})$, is given by

$$\nabla_s f(\mathcal{X}) = S(\mathcal{X})^{-T} \delta(\mathcal{X}).$$

When $k < d$ (the underdetermined case), the *simplex gradient with respect to* $\mathcal{X}$ is the minimum 2-norm solution to the system

$$S(\mathcal{X})^T \nabla_s f(\mathcal{X}) = \delta(\mathcal{X}),$$

which is given by $\nabla_s f(\mathcal{X}) = S(\mathcal{X})(S(\mathcal{X})^T S(\mathcal{X}))^{-1}\delta(\mathcal{X})$. In this case, $\nabla_s f(\mathcal{X})$ is a linear combination of $x_1 - x_0, x_2 - x_0, \dots, x_k - x_0$ since $\nabla_s f(\mathcal{X}) = S(\mathcal{X})v$, where $v = (S(\mathcal{X})^T S(\mathcal{X}))^{-1}\delta(\mathcal{X}) \in \mathbb{R}^k$.

The following proposition shows that it is not necessary that $\mathcal{X}$ be an ordered set when defining the simplex gradient.

Proposition 1. Suppose $\mathcal{X} = \langle x_0, x_1, \dots, x_k\rangle$ is an ordered set of $k+1$ affinely independent points in $\mathbb{R}^d$, where $k \le d$. Let α be a permutation of the indices $\{0, 1, \dots, k\}$ and let $\mathcal{X}_\alpha = \langle x_{\alpha(0)}, x_{\alpha(1)}, \dots, x_{\alpha(k)}\rangle$. Then $\nabla_s f(\mathcal{X}_\alpha) = \nabla_s f(\mathcal{X})$.

Proof. First consider the case where $\alpha(0) \neq 0$. Then $\alpha(j) = 0$ for some index $1 \le j \le k$ and $S(\mathcal{X}_\alpha) = [x_{\alpha(1)} - x_{\alpha(0)} \quad \dots \quad x_{\alpha(k)} - x_{\alpha(0)}]$ can be transformed to $S(\mathcal{X}) = [x_1 - x_0 \quad \dots \quad x_k - x_0]$ by applying a series of elementary column operations to $S(\mathcal{X}_\alpha)$. To see this, begin by multiplying the jth column of $S(\mathcal{X}_\alpha)$ by -1. The result is also given by $S(\mathcal{X}_\alpha)M$, where M is the elementary matrix obtained by replacing the jth diagonal entry of I_d by -1. Next, for each $i = 1, \dots, k,\ i \neq j$, perform an elementary column operation that consist of adding the jth column of $S(\mathcal{X}_\alpha)M$ to the ith column and storing the result in the latter column. The result is $S(\mathcal{X}_\beta)$ for some permutation β of the indices $\{0, 1, \dots, k\}$ that fixes 0. That is,

$$S(\mathcal{X}_\alpha)ME_1E_2 \dots E_{k-1} = [x_{\beta(1)} - x_0 \quad \dots \quad x_{\beta(k)} - x_0],$$

where $E_1, E_2, \dots, E_{k-1}$ are the elementary matrices obtained by adding the jth column of I_d to the other columns and storing the results in those columns. Finally, $S(\mathcal{X})$ can be obtained by applying a series of column interchanges to $S(\mathcal{X}_\alpha)ME_1E_2 \dots E_{k-1}$, i.e.,

$$S(\mathcal{X}_\alpha)ME_1E_2 \dots E_{k-1}P = S(\mathcal{X}),$$

for some permutation matrix P.

Let $F = ME_1E_2 \dots E_{k-1}P$. Then $S(\mathcal{X}_\alpha)F = S(\mathcal{X})$ and F is nonsingular because it is the product of nonsingular matrices. Observe that

$$F^T\delta(\mathcal{X}_\alpha) = (ME_1E_2 \dots E_{k-1}P)^T\delta(\mathcal{X}_\alpha) = P^T E_{k-1}^T \dots E_2^T E_1^T M^T \delta(\mathcal{X}_\alpha) = \delta(\mathcal{X}).$$

Hence,

$$\nabla_s f(\mathcal{X}) = S(\mathcal{X})(S(\mathcal{X})^T S(\mathcal{X}))^{-1}\delta(\mathcal{X}) = S(\mathcal{X}_\alpha)F\ (\ (S(\mathcal{X}_\alpha)F)^T(S(\mathcal{X}_\alpha)F)\ )^{-1}\delta(\mathcal{X})$$

$$= S(\mathcal{X}_\alpha)F\ (\ F^T S(\mathcal{X}_\alpha)^T S(\mathcal{X}_\alpha)F\ )^{-1}\delta(\mathcal{X}) = S(\mathcal{X}_\alpha)FF^{-1}\ (\ S(\mathcal{X}_\alpha)^T S(\mathcal{X}_\alpha)\ )^{-1}(F^T)^{-1}\delta(\mathcal{X})$$

$$= S(\mathcal{X}_\alpha)(S(\mathcal{X}_\alpha)^T S(\mathcal{X}_\alpha))^{-1}\delta(\mathcal{X}_\alpha) = \nabla_s f(\mathcal{X}_\alpha).$$

The proof for the case where $\alpha(0) = 0$ is similar, and in fact, simpler because only permutations are involved. □

Next, the following well-known result is useful for understanding the initialization strategies that are given in the succeeding sections.

Proposition 2. Let $\mathcal{X} = \{x_0, x_1, \dots, x_k\}$ be a set of $k+1 < d+1$ affinely independent points in $\mathbb{R}^d$. If $x_{k+1} \notin \mathrm{aff}(\mathcal{X})$ (the affine hull of the points in $\mathcal{X}$), then $\mathcal{X} \cup \{x_{k+1}\}$ is also affinely independent.

3.3 Using Underdetermined Simplex Gradients to Initialize Surrogate-Based Optimization Methods

The main idea of the algorithm below is to build a set of $d + 1$ affinely independent points by iteratively adding new points that are not in the affine hull of the previously chosen points. To increase the chances of making progress in finding the minimum of f, a new point is sometimes chosen such that the vector from the current best point (in terms of the value of f) to this new point makes an acute angle with the negative of the simplex gradient. However, preliminary numerical experiments show that the condition number of $L(\mathcal{X})$ can quickly deteriorate as more points are added to $\mathcal{X}$ in this particular way. Hence, the algorithm proceeds in two phases. In the first phase, the algorithm performs a series of iterations where it adds a new point such that the vector from the current best point to the new point is perpendicular to the affine hull of the previously chosen points. In this phase, the condition number does not deteriorate much. In the second phase, the algorithm iteratively adds a new point such that the vector from the current best point to this new point makes an acute angle with the negative of the simplex gradient at the current best point. Moreover, to keep the condition number of $L(\mathcal{X})$ to a reasonable value, we find a point $\widetilde{x}$ that minimizes $\text{cond}(\mathcal{X} \cup \{\widetilde{x}\})$ whenever the condition number exceeds a particular threshold.

Underdetermined Simplex Gradient Descent (USGD):

Inputs:

(1) Function to minimize: $f : [a, b] \to \mathbb{R}$, where $[a, b] \subseteq \mathbb{R}^d$

(2) Starting point: $x_0 \in [a, b]$

(3) Step size: $\Delta \leq 0.5 \min_{1 \leq i \leq d}(b_i - a_i)$

(4) Number of perpendicular moves: $0 \leq n_p < d$

(5) Acute angles with the negative simplex gradient: $0 < \theta_k < \pi/2$ for $k = n_p + 1, \ldots, d$

(6) Threshold condition number: κ_{max}

Outputs: A set of $d+1$ affinely independent points $\mathcal{X} = \{x_0, x_1, \ldots, x_d\} \subseteq \mathbb{R}^d$ and their objective function values $\mathcal{F} = \{f(x_0), f(x_1), \ldots, f(x_d)\}$.

Step 1. (Evaluate Starting Point) Calculate $f(x_0)$. Initialize $\mathcal{X} = \{x_0\}$ and $\mathcal{F} = \{f(x_0)\}$. Also, set $x_{\text{best}} = x_0$ and $f_{\text{best}} = f(x_0)$.

Phase I: Perpendicular Moves

Step 2. (Initialize Set of Coordinates) Set $\mathcal{C} = \{1, \ldots, d\}$.

Step 3. (Select Points) For $k = 1$ to n_p do

(3a) **(Select New Point)** Consider the set of points $\mathcal{T} = \bigcup_{j \in \mathcal{C}} \{x_{\text{best}} \pm \Delta e_j\}$ and let $\widetilde{x}$ be a point in $\mathcal{T} \cap [a, b]$ that minimizes $\text{cond}(L(\mathcal{X} \cup \{\widetilde{x}\}))$. Let $\widetilde{j}$ be the element of $\mathcal{C}$ that gave rise to $\widetilde{x}$.

(3b) **(Evaluate Selected Point)** Set $x_k = \widetilde{x}$ and calculate $f(x_k)$.

(3c) **(Update Information)** Update x_{best} and f_{best}. Reset $\mathcal{X} = \mathcal{X} \cup \{x_k\}$ and $\mathcal{F} = \mathcal{F} \cup \{f(x_k)\}$. Also, reset $\mathcal{C} = \mathcal{C} \setminus \{\widetilde{j}\}$.

End for.

Phase II: Acute-Angled Moves Guided by the Negative Simplex Gradient

Step 4. (Select Points) For $k = n_p + 1$ to d do

(4a) **(Identify Best Point)** Let ℓ be the index such that $x_{\text{best}} = x_\ell$.

(4b) **(Determine Simplex Gradient)** Calculate $\nabla_s f(\mathcal{X})$.

(4c) **(Determine Vectors Orthogonal to Affine Hull of Previous Points)** Determine an orthonormal basis $\{z_1, \ldots, z_{d-k}\}$ for $\text{Null}(S(\mathcal{X})^T)$. (Since $\mathcal{X} = \{x_0, x_1, \ldots, x_{k-1}\}$ consists of k affinely independent points, $\dim(\text{Null}(S(\mathcal{X})^T)) = d - k + 1$.

(4d) For $j = 1$ to $d - k + 1$ do

If $\nabla_s f(\mathcal{X}) \neq 0$, then define $y_{k,j} := w_k z_j - \dfrac{\nabla_s f(\mathcal{X})}{\|\nabla_s f(\mathcal{X})\|}$, where $w_k = \tan(\theta_k)$. Else, define $y_{k,j} := z_j$.

End for.

(4e) Consider the set of trial points $\mathcal{T} = \displaystyle\bigcup_{j=1}^{d-k+1} \left\{ x_{\text{best}} + \Delta \frac{y_{k,j}}{\|y_{k,j}\|} \right\}$. Let $\widetilde{x}$ be a point in $\mathcal{T} \cap [a, b]$ that minimizes $\text{cond}(L(\mathcal{X} \cup \{\widetilde{x}\}))$.

(4f) If the minimum condition number of $\text{cond}(L(\mathcal{X} \cup \{\widetilde{x}\}))$ from (4e) exceeds κ_{max}, then replace $\widetilde{x}$ by a point in $[a, b]$ that minimizes $\text{cond}(L(\mathcal{X} \cup \{\widetilde{x}\}))$.

(4g) **(Evaluate Selected Point)** Set $x_k = \widetilde{x}$ and calculate $f(x_k)$.

(4h) **(Update Information)** Update x_{best} and f_{best}. Reset $\mathcal{X} = \mathcal{X} \cup \{x_k\}$ and $\mathcal{F} = \mathcal{F} \cup \{f(x_k)\}$.

End for.

Step 5. (Return Outputs) Return $\mathcal{X}$ and $\mathcal{F}$.

The next proposition is used to verify that the angle between the negative simplex gradient and the vector $\Delta(y_{k,j}/\|y_{k,j}\|)$ (i.e., the vector from the current best point to a trial point) in Step (4e) is the specified angle θ_k.

Proposition 3. Let $v, z \in \mathbb{R}^d$ such that $v \neq 0$ and $z^T v = 0$. Moreover, let w be any positive real number and let $u = wz + v$. Then $u \neq 0$ and the angle between u and v is $\tan^{-1}(w\|z\|/\|v\|)$.

Proof. Let θ be the angle between $u = wz + v$ and v. Since $z^T v = 0$, it follows that

$$\|u\|^2 = u^T u = (wz + v)^T (wz + v) = w^2 z^T z + v^T v = w^2 \|z\|^2 + \|v\|^2. \tag{2}$$

Moreover, since $v \neq 0$, it follows that $\|u\|^2 > 0$, and so, $u \neq 0$. From elementary linear algebra, $0 \leq \theta \leq \pi$ and

$$\cos\theta = \frac{u^T v}{\|u\|\|v\|} = \frac{(wz + v)^T v}{\|u\|\|v\|} = \frac{w(z^T v) + v^T v}{\|u\|\|v\|} = \frac{\|v\|^2}{\|u\|\|v\|} = \frac{\|v\|}{\|u\|}.$$

From the previous equation, $\cos\theta > 0$, and so, $0 \leq \theta < \pi/2$. Now from (2),

$$\tan^2\theta = \sec^2\theta - 1 = \frac{\|u\|^2}{\|v\|^2} - 1 = \frac{w^2\|z\|^2 + \|v\|^2}{\|v\|^2} - 1 = \frac{w^2\|z\|^2}{\|v\|^2}.$$

Note that $\tan\theta > 0$ since $0 \leq \theta < \pi/2$. Moreover, since $w > 0$, it follows that $\tan\theta = w\|z\|/\|v\|$, and so, $\theta = \tan^{-1}(w\|z\|/\|v\|)$. □

In Step 4(d) of USGD, note that z_j and $-\nabla_s f(\mathcal{X})/\|\nabla_s f(\mathcal{X})\|$ are unit vectors. By Proposition 3, $y_{k,j} \neq 0$ and the angle between $y_{k,j}$ and $-\nabla_s f(\mathcal{X})/\|\nabla_s f(\mathcal{X})\|$ is

$$\tan^{-1}\left(\frac{w_k\|z_j\|}{\| - \nabla_s f(\mathcal{X})/\|\nabla_s f(\mathcal{X})\|\|}\right) = \tan^{-1}(w_k) = \theta_k.$$

Hence, the angle between the negative simplex gradient $-\nabla_s f(\mathcal{X})$ and $y_{k,j}/\|y_{k,j}\|$ (the vector from the current best point to the jth trial point in Step 4(e)) is also θ_k.

To show that the above algorithm yields $d+1$ affinely independent points, we first prove the following simple result.

Proposition 4. Suppose $\mathcal{X} = \langle x_0, x_1, \ldots, x_k\rangle$ is an ordered set of $k+1$ affinely independent points in $\mathbb{R}^d$, where $k \leq d$. Let α be a permutation of the indices $\{0, 1, \ldots, k\}$ and let $\mathcal{X}_\alpha = \langle x_{\alpha(0)}, x_{\alpha(1)}, \ldots, x_{\alpha(k)}\rangle$. Then $\text{Null}(S(\mathcal{X}_\alpha)^T) = \text{Null}(S(\mathcal{X})^T)$.

Proof. By definition,

$$\text{Null}(S(\mathcal{X}_\alpha)^T) = \{z \in \mathbb{R}^d \mid z^T(x_{\alpha(i)} - x_{\alpha(0)}) = 0 \text{ for } i = 1, 2, \ldots, k\}.$$

If $\alpha(0) = 0$, then $\text{Null}(S(\mathcal{X}_\alpha)^T) = \text{Null}(S(\mathcal{X})^T)$. Next, suppose $\alpha(0) \neq 0$. Let $z \in \text{Null}(S(\mathcal{X}_\alpha)^T)$. For any $i = 1, \ldots, k$, note that

$$z^T(x_i - x_0) = z^T(\ (x_i - x_{\alpha(0)}) + (x_{\alpha(0)} - x_0)\ ) = z^T(x_i - x_{\alpha(0)}) - z^T(x_0 - x_{\alpha(0)}) = 0.$$

This shows that $z \in \text{Null}(S(\mathcal{X})^T)$. Hence, $\text{Null}(S(\mathcal{X}_\alpha)^T) \subseteq \text{Null}(S(\mathcal{X})^T)$. A similar argument shows that $\text{Null}(S(\mathcal{X})^T) \subseteq \text{Null}(S(\mathcal{X}_\alpha)^T)$. □

Next, the following result shows that the acute-angled moves guided by the negative simplex gradient result in affinely independent points.

Proposition 5. Let $\mathcal{X} = \{x_0, x_1, \ldots, x_k\}$ be a set of $k+1 < d+1$ affinely independent points in $\mathbb{R}^d$ whose function values $f(x_0), f(x_1), \ldots, f(x_k)$ are known and let $x_\ell \in \mathcal{X}$ be a point with the smallest function value (i.e., $f(x_\ell) \leq f(x_i)$ for all $i = 0, 1, \ldots, k$). Moreover, let $z \in \text{Null}(S(\mathcal{X})^T)$ with $z \neq 0$. Then for any $\alpha \neq 0$ and any constant β, $x_\ell + (\alpha z + \beta\nabla_s f(\mathcal{X})) \notin \text{aff}(\mathcal{X})$, and so, the set $\mathcal{X} \cup \{x_\ell + (\alpha z + \beta\nabla_s f(\mathcal{X}))\}$ is also affinely independent.

Proof. Suppose $x_\ell + (\alpha z + \beta\nabla_s f(\mathcal{X})) \in \text{aff}(\mathcal{X})$. Then

$$x_\ell + (\alpha z + \beta\nabla_s f(\mathcal{X})) = a_0x_0 + a_1x_1 + \ldots + a_kx_k,$$

where $a_0, a_1, \ldots, a_k$ are constants such that $a_0 + a_1 + \ldots + a_k = 1$. Now

$$\alpha z + \beta\nabla_s f(\mathcal{X}) = a_0x_0 + a_1x_1 + \ldots + a_kx_k - (a_0 + a_1 + \ldots + a_k)x_\ell,$$

and so,

$$\alpha z = a_0(x_0 - x_\ell) + a_1(x_1 - x_\ell) + \ldots + a_k(x_k - x_\ell) - \beta \nabla_s f(\mathcal{X}).$$

By Proposition 4,

$$z \in \mathrm{Null}(S(\mathcal{X})^T) = \mathrm{Null}(S(\{x_\ell, x_0, x_1, \ldots, x_{\ell-1}, x_{\ell+1}, \ldots, x_k\})^T)$$

$$= \mathrm{Null}([x_0 - x_\ell, x_1 - x_\ell, \ldots, x_{\ell-1} - x_\ell, x_{\ell+1} - x_\ell, \ldots, x_k - x_\ell]^T).$$

Hence, z is a nonzero vector that is perpendicular to $x_i - x_\ell$ for all $i = 0, 1, \ldots, k$, $i \neq \ell$. Since $\nabla_s f(\mathcal{X})$ is a linear combination of the vectors $x_i - x_\ell$ with $i \neq \ell$, it follows that $z^T \nabla_s f(\mathcal{X}) = 0$ and

$$\alpha \|z\|^2 = \alpha z^T z = z^T(\alpha z) = a_0 z^T(x_0 - x_\ell) + a_1 z^T(x_1 - x_\ell) + \ldots + a_k z^T(x_k - x_\ell) - \beta z^T \nabla_s f(\mathcal{X}) = 0.$$

This leads to $\alpha \|z\|^2 = 0$, which is a contradiction since $z \neq 0$ and $\alpha \neq 0$. □

By Proposition 5, each trial point in Step 4(e) does *not* belong to aff($\mathcal{X}$). To see this, note that

$$x_{\text{best}} + \Delta \frac{y_{k,j}}{\|y_{k,j}\|} = x_{\text{best}} + \left(\frac{\Delta w_k}{\|y_{k,j}\|} z_j - \frac{\Delta}{\|y_{k,j}\| \|\nabla_s f(\mathcal{X})\|} \nabla_s f(\mathcal{X}) \right) \notin \mathrm{aff}(\mathcal{X}).$$

Hence, USGD generates a set of $d + 1$ affinely independent points.

The minimization of the condition number in Step 4(f) can be carried out by any standard numerical optimization solver. In this study, this is implemented by creating a Matlab function whose input is a point $x \in [a, b] \subseteq \mathbb{R}^d$ and whose output is cond($L(\mathcal{X} \cup \{\widetilde{x}\}$), where $\mathcal{X}$ is the set of previously evaluated points. The Fmincon routine from the Matlab Optimization Toolbox (The Mathworks, Inc. 2009) is then used to find a point $x \in [a, b]$ that minimizes cond($L(\mathcal{X} \cup \{\widetilde{x}\}$).

Note that Step 4(e) can take a long time, especially for high-dimensional problems. To speed up the algorithm, one modification is to form the set $\mathcal{T}$ in Step 4(e) using only a random sample of the orthonormal basis from Step 4(c). This modified method is referred to as *USGD-Fast*.

The minimization of the condition number in Step 4(f) can also take a long time. The running time can be reduced by allowing the solver to stop after a maximum number of iterations. Another possibility is to find a quick way to update the condition number of $L(\mathcal{X})$ when a new point is added and this will be explored in future work.

4 Numerical Comparison of Initialization Strategies

4.1 Test Problems

The three initialization strategies (Static Simplex (SS), Dynamic Simplex (DS) and Underdetermined Simplex Gradient Descent (USGD)) are compared on the 200-D and 1000-D versions of 16 well-known test functions. Table 1 summarizes the characteristics of these test problems. The first twelve test functions in Table 1 are taken from Moré et al. (1981) while the remaining four test functions (the Ackley, Rastrigin, Griewank and Keane functions) are well-known in the engineering optimization community. For the Linear Function - Full Rank and Linear Function - Rank 1 problems from Moré et al. (1981), the number of residual functions, denoted by m, is set to 300 for the 200-D instances and 1500 for the 1000-D instances. Most of the twelve test problems from Moré et al. (1981) each have only one local minimum while the last four problems each have a large

Table 1: Test problems for the computational experiments.

Test Function	Domain	Global min value
Extended Rosenbrock	$[-2,2]^d$	0
Extended Powell Singular	$[-1,3]^d$	0
Penalty Function I	$[-1,3]^d$	unknown
Variably Dimensioned	$[-2,2]^d$	0
Trigonometric	$[-1,3]^d$	0
Brown Almost-Linear	$[-2,2]^d$	0
Discrete Boundary Value	$[-3,3]^d$	0
Discrete Integral Equation	$[-1,3]^d$	0
Broyden Tridiagonal	$[-1,1]^d$	0
Broyden Banded	$[-1,1]^d$	0
Linear Function – Full Rank	$[-2,1]^d$	100 (for $d = 200, m = 300$)
		500 (for $d = 1000, m = 1500$)
Linear Function – Rank 1	$[-1,3]^d$	74.6256 (for $d = 200$)
		374.6251 (for $d = 1000$)
Ackley	$[-15,20]^d$	$-20 - \exp(1)$
Rastrigin	$[-4,5]^d$	$-d$
Griewank	$[-500,700]^d$	0
Keane	$[1,10]^d$	unknown

number of local minima. Note that 200 and 1000 dimensions are considered very high dimensional in surrogate-based optimization since most papers in this area generally deal with problems with dimensions $d < 15$.

The above test problems are not really computationally expensive to evaluate and the different strategies are compared by pretending that these functions are expensive. This is accomplished by keeping track of the best function values obtained by the different algorithms as the number of function evaluations increases. The relative performance of algorithms on the test problems are expected to be similar to their relative performance on truly expensive functions that have the same general surface as these test problems.

4.2 Management of Groundwater Bioremediation

The initialization strategies are also compared on a groundwater bioremediation problem (Minsker and Shoemaker 1998, Yoon and Shoemaker 1999). Groundwater bioremediation involves using injection wells that supply electron acceptors (e.g., oxygen) or electron donors (e.g., hydrogen) into the groundwater to promote the growth of the soil bacteria that can transform contaminants into harmless substances. Monitoring wells are also used to measure the concentration of the contaminant at specific locations and ensure that it is below some threshold level at specified time periods.

The setup involves a hypothetical contaminated aquifer whose characteristics are symmetric about a horizontal axis. The aquifer is discretized using a 2-D finite element mesh with 18 nodes in the horizontal direction and 9 nodes in the vertical direction. There are 6 injection wells and 84 monitoring wells that are also symmetrically arranged. Oxygenated water is pumped into the injection wells. The optimization formulation involves a 2-D finite element simulation model that describes groundwater flow and changes in the concentrations of the contaminant, oxygen and biomass. The entire planning horizon is evenly divided into 24 management periods and the problem is to determine the pumping rates for each injection well at the beginning of each management period in order to minimize the total pumping cost subject to some constraints on the contaminant concentration at the monitoring wells. The problem can be reformulated as a box-constrained global optimization problem by incorporating the constraints into the objective using a penalty term (Yoon and Shoemaker 1999). Because the wells are symmetrically arranged, pumping decisions are only needed for the 3 injection wells on one side of the axis of symmetry resulting in 72 decision variables for the optimization problem. The maximum pumping rate is rescaled to 1 so the search space is $[0,1]^{72}$. This groundwater bioremediation problem is referred to as GWB72 and it is an extension of the 12-dimensional problem in Regis and Shoemaker (2007a). This groundwater bioremediation model uses a relatively coarse grid so its simulation time is only about 0.1 second on an Intel(R) Core(TM) i7 CPU 860 2.8 GHz desktop machine. However, it is representative of more complex groundwater bioremediation problems whose simulation times can sometimes take several hours (Shoemaker et al. 2001).

4.3 Experimental Setup

Two sets of numerical experiments are performed to assess the effectiveness of the USGD initialization strategy. In the first set of experiments, the USGD strategy is compared with the Static Simplex (SS) and Dynamic Simplex (DS) strategies in terms of the mean of the best objective function value. In the second set of experiments, each surrogate-based algorithm is run with each of the three different initialization strategies (SS, DS and USGD).

For the first set of experiments, each initialization strategy is run for 30 trials on the 72-D groundwater bioremediation problem GWB72 and on each of the 200-D test problems. However, since these strategies can become computationally expensive on the 1000-D test problems, they are only run for 10 trials on these problems. Each trial corresponds to a fixed starting point that is the same for all methods and all strategies use the same step size of $\Delta = 0.2 \min_{1 \leq i \leq d}(b_i - a_i)$. For GWB72 and the 200-D problems, USGD is run with $n_p = \lfloor d/2 \rfloor$, $\theta_k = 75^o$ for all k and $\kappa_{\max} = 10^5$. However, for the 1000-D problems, the faster implementation USGD-Fast is run with $n_p = \lfloor 3d/4 \rfloor$, $\theta_k = 80^o$ for all k and $\kappa_{\max} = 10^6$. The reason for delaying the acute-angled moves guided by the negative simplex gradient and for using steep angles for the 1000-D problems is to prevent the condition number of the linear interpolation matrix from becoming very large. The threshold condition numbers $\kappa_{\max} = 10^5$ or 10^6 are set based on numerical experiments with the condition numbers of the linear interpolation matrices of randomly generated points. More precisely, when $d = 200$, the mean and median condition numbers (out of 10,000 trials) of linear interpolation matrices of uniform random points on $[0,1]^d$ are 6.28×10^4 and 9.99×10^3, respectively. When $d = 1000$, the mean and median condition numbers (out of 10,000 trials) of linear interpolation matrices of uniform random points on $[0,1]^d$ are 5.68×10^5 and 1.14×10^5, respectively. All methods are run in Matlab 7.11 using an Intel(R) Core(TM) i7 CPU 860 2.8 GHz desktop machine.

For the second set of numerical experiments, two surrogate-based methods for high-dimensional, bound-constrained, expensive black-box optimization are run with each of the three initialization strategies (SS, DS and USGD or USGD-Fast) on GWB72 and on the 200-D and 1000-D test problems. Again, USGD is used for GWB72 and the 200-D problems while USGD-Fast is used for the 1000-D problems. The surrogate-based methods used are DYCORS-DDSRBF (Regis and Shoemaker 2013) and a Matlab implementation of a pattern search algorithm that uses RBF surrogates (Le Thi et al. 2012), which is denoted by PS-RBF. The particular RBF model used for both DYCORS-DDSRBF and PS-RBF is a cubic RBF augmented by a linear polynomial tail, which has been used by Björkman and Holmström (2000), Wild, Regis and Shoemaker (2008), Regis (2011), Le Thi et al. (2012), and Regis and Shoemaker (2013). Thus, six methods are compared: DYCORS-DDSRBF (SS), DYCORS-DDSRBF (DS), DYCORS-DDSRBF (USGD or USGD-Fast), PS-RBF (SS), PS-RBF (DS) and PS-RBF (USGD or USGD-Fast). Each combination of optimization method and initialization strategy is run for 30 trials on GWB72 and on the 200-D problems and for only 10 trials on the 1000-D problems. Each trial begins with the initial points generated by the assigned initialization strategy and this is the same for DYCORS-DDSRBF and PS-RBF with this initialization strategy. All computational runs are also carried in Matlab 7.11 using the same machine that was used to test the initialization strategies.

DYCORS-DDSRBF is an RBF-assisted modification of the *DDS (Dynamically Dimensioned Search)* heuristic by Tolson and Shoemaker (2007). It follows the *DYCORS (DYnamic COordinate search using Response Surface models)* framework for bound constrained, high dimensional, expensive black-box optimization by Regis and Shoemaker (2013). In the DYCORS framework, a response surface (or surrogate) model is used to select the iterate from a set of random trial solutions generated by perturbing only a subset of the coordinates of the current best solution. Moreover, the probability of perturbing a coordinate of the current best solution decreases as the algorithm progresses. PS-RBF is a pattern search algorithm guided by RBF surrogates (Le Thi et al. 2012) that is implemented in the PSwarm solver (Vaz and Vicente 2007, 2009). In PS-RBF, an RBF model is built and minimized during the search step of the underlying pattern search method. The RBF model is minimized within an ∞-norm trust region whose radius is proportional to the step size and the resulting box constrained problem is solved using a d.c. programming method. In addition, the RBF model is used to sort the directions in the poll step of the pattern search method.

Combining the DYCORS-DDSRBF solver with the initialization strategies is straightforward. The set of affinely independent points and objective function values obtained by each initialization strategy are used as inputs to the solver. The DYCORS-DDSRBF solver then begins by fitting the initial RBF model that is used to select the next iterate. Ideally, merging an initialization strategy with the PS-RBF solver should be done in the same manner. However, the PS-RBF code from the PSwarm solver does not accept a set of points as input. It only accepts a single starting point so the best point (in terms of objective function value) obtained by an initialization strategy is used as the starting point for PS-RBF. Hence, one limitation of this study is that the PS-RBF solver is not properly merged with the initialization strategies, and any improvement of PS-RBF (USGD or USGD-Fast) over PS-RBF (DS) or PS-RBF (SS) is most likely due only to the fact the starting point of the former has better objective function value than the starting points of the latter algorithms. The performance of PS-RBF can be further improved if the entire set of points obtained by an initialization strategy is used by the solver.

Before proceeding with the comparisons, it is important to clarify that the purpose of the comparisons below is *not* to demonstrate that one algorithm is superior to the other methods. In particular, it is not the intent to the paper to show that DYCORS-DDSRBF (USGD) is better than PS-RBF (USGD) or even to prove that DYCORS-DDSRBF (USGD) is always better than DYCORS-DDSRBF (SS). First of all, the comments in the previous paragraph suggest that any comparison between DYCORS-DDSRBF and PS-RBF combined with any initialization strategy would not be fair. Second, the test problems are relatively limited so one cannot really generalize the results to a larger class of problems. After all, it is widely believed that there is no universally best optimization method. Finally, it is difficult to guarantee that each method is run with the best parameter settings for the given problems. Although default and reasonable parameter settings are used for the different methods, it might still be possible to improve their performance by more carefully tuning their parameters (some of which are not visible to the user). The goal is simply to

Table 2: Mean and Standard Error of the Best Objective Function Values in 30 trials for Three Initialization Strategies for Surrogate-Based Optimization Methods on GWB72 and on the 200-D Problems.

Test Function	USGD	Dynamic Simplex (DS)	Static Simplex (SS)
GWB72	548.78 (18.27)	1919.57 (37.88)	2368.24 (52.41)
Extended Rosenbrock	3347.54 (131.95)	5959.02 (162.63)	44334.37 (1153.21)
Extended Powell Singular	7342.31 (207.92)	20917.62 (627.58)	80834.69 (3313.60)
Penalty Function I	6534.78 (308.14)	96551.29 (2571.04)	217584.45 (5761.92)
Variably Dimensioned	1.37×10^{15} (2.08×10^{14})	3.49×10^{15} (4.32×10^{14})	1.64×10^{17} (1.04×10^{16})
Trigonometric	4.92×10^{5} (1.91×10^{4})	7.46×10^{6} (1.72×10^{5})	1.18×10^{7} (2.71×10^{5})
Brown Almost-Linear	1.21×10^{5} (1.84×10^{4})	1.10×10^{6} (7.21×10^{4})	8.00×10^{6} (2.24×10^{5})
Discrete Boundary Value	422.43 (12.02)	964.61 (21.58)	3522.71 (65.03)
Discrete Integral Equation	94.75 (2.72)	752.14 (14.49)	1176.89 (23.23)
Broyden Tridiagonal	231.48 (4.16)	327.81 (6.51)	993.18 (16.51)
Broyden Banded	399.72 (7.59)	756.48 (10.95)	3225.48 (53.12)
Linear Function – Full Rank	182.50 (1.83)	212.83 (1.61)	299.58 (2.54)
Linear Function – Rank 1	2.65×10^{12} (1.04×10^{12})	2.59×10^{15} (8.62×10^{13})	3.64×10^{15} (1.21×10^{14})
Ackley	-9.09 (0.15)	-6.21 (0.05)	-3.50 (0.03)
Rastrigin	156.30 (8.51)	548.82 (7.82)	1398.10 (15.73)
Griewank	604.68 (15.62)	2841.80 (39.97)	6492.08 (74.85)
Keane	-0.1439 (0.0016)	-0.1537 (0.0009)	-0.0895 (0.0014)

Table 3: Mean and Standard Error of the Best Objective Function Values in 10 trials for Three Initialization Strategies for Surrogate-Based Optimization Methods on the 1000-D Problems.

Test Function	USGD-Fast	Dynamic Simplex (DS)	Static Simplex (SS)
Extended Rosenbrock	26,156.39 (741.33)	30,285.17 (401.52)	229,789.24 (3678.87)
Extended Powell Singular	66,230.64 (1146.68)	107,335.62 (2210.79)	449,075.45 (10161.50)
Penalty Function I	822,745.49 (18160.28)	2,394,676.33 (56040.92)	5,396,250.76 (115621.31)
Variably Dimensioned	3.65×10^{20} (4.11×10^{19})	1.22×10^{21} (1.12×10^{20})	6.54×10^{22} (2.84×10^{21})
Trigonometric	2.91×10^{8} (5.55×10^{6})	9.27×10^{8} (1.75×10^{7})	1.47×10^{9} (2.53×10^{7})
Brown Almost-Linear	2.73×10^{7} (2.47×10^{6})	1.37×10^{8} (7.65×10^{6})	1.02×10^{9} (2.51×10^{7})
Discrete Boundary Value	2534.10 (283.85)	4993.86 (77.45)	18506.10 (340.80)
Discrete Integral Equation	1472.63 (28.91)	3718.43 (60.45)	5864.37 (103.47)
Broyden Tridiagonal	1277.67 (50.33)	1671.79 (28.70)	5271.13 (93.70)
Broyden Banded	3289.52 (44.03)	3801.36 (31.98)	16536.81 (154.76)
Linear Function – Full Rank	764.87 (6.93)	1062.23 (6.85)	1497.04 (9.56)
Linear Function – Rank 1	1.28×10^{19} (9.35×10^{17})	1.95×10^{20} (5.04×10^{18})	2.76×10^{20} (6.76×10^{18})
Ackley	-8.77 (0.24)	-6.19 (0.04)	-3.47 (0.02)
Rastrigin	1890.34 (22.24)	2722.35 (35.24)	7013.49 (49.99)
Griewank	6862.44 (98.96)	14180.55 (169.62)	32518.19 (239.73)
Keane	-0.1449 (0.0008)	-0.1552 (0.0007)	-0.0894 (0.0009)

make a case why some of these methods (e.g., DYCORS-DDSRBF (USGD)) should be seriously considered for solving high-dimensional expensive black-box optimization problems.

4.4 Results of Simplex Gradient Descent

Table 2 shows the results of applying the USGD initialization procedure and the two other alternatives (SS and DS) on the 200-D test problems. Table 3 shows the results of applying USGD-Fast and the other methods on the 1000-D problems. Note that the simple modification provided by DS already yields large improvements on the best objective function value over SS. However, USGD

Table 4: Mean and Standard Error of the Condition Numbers of the Resulting Interpolation Matrix in 30 trials for Three Initialization Strategies for Surrogate-Based Optimization Methods on GWB72 and on the 200-D Problems.

Test Function	USGD	Dynamic Simplex (DS)	Static Simplex (SS)
GWB72	481.84 (13.78)	2012.91 (59.30)	2816.69 (118.67)
Extended Rosenbrock	17,418.74 (1151.65)	3766.98 (54.66)	38,613.17 (1009.10)
Extended Powell Singular	18,529.61 (952.13)	10,900.17 (157.00)	13,224.10 (865.08)
Penalty Function I	4173.08 (295.32)	10,101.70 (179.56)	13,224.10 (865.08)
Variably Dimensioned	14,442.34 (989.06)	6481.60 (90.49)	38,613.17 (1009.10)
Trigonometric	4914.77 (208.29)	10,110.82 (178.31)	13,224.10 (865.08)
Brown Almost-Linear	7774.41 (460.17)	6481.60 (90.49)	38,613.17 (1009.10)
Discrete Boundary Value	16,309.45 (1329.72)	6365.37 (120.69)	57,676.07 (1516.08)
Discrete Integral Equation	4879.38 (212.35)	10,026.23 (266.24)	13,224.10 (865.08)
Broyden Tridiagonal	11,248.22 (369.68)	2568.09 (45.59)	19,743.66 (500.35)
Broyden Banded	5693.20 (186.77)	1476.10 (19.41)	19,743.66 (500.35)
Linear Function – Full Rank	19,626.08 (1265.89)	4217.09 (89.41)	53,051.24 (986.62)
Linear Function – Rank 1	7619.04 (442.15)	9998.30 (268.55)	13,224.10 (865.08)
Ackley	31,655.90 (3313.84)	31,634.32 (500.27)	259,032.55 (9336.27)
Rastrigin	27,343.19 (2146.34)	7514.11 (99.70)	70,841.79 (2351.12)
Griewank	576,644.29 (32668.25)	1,156,648.70 (17580.00)	8,477,462.48 (325536.92)
Keane	138,487.01 (12079.53)	82,492.99 (1160.75)	245,775.76 (10121.11)

Table 5: Mean and Standard Error of the Condition Numbers of the Resulting Interpolation Matrix in 10 trials for Three Initialization Strategies for Surrogate-Based Optimization Methods on the 1000-D Problems.

Test Function	USGD-Fast	Dynamic Simplex (DS)	Static Simplex (SS)
Extended Rosenbrock	132,474.48 (12,672.99)	41,834.45 (372.65)	942,541.41 (13,273.37)
Extended Powell Singular	297,399.59 (38,993.88)	120,619.52 (1481.65)	142,210.10 (11,636.73)
Penalty Function I	244,556.11 (46,266.62)	113,910.31 (831.45)	142,210.10 (11,636.73)
Variably Dimensioned	108,801.26 (9040.40)	72,204.04 (753.03)	942,541.41 (13,273.37)
Trigonometric	141,579.52 (25,123.47)	113,903.08 (832.55)	142,210.10 (11636.73)
Brown Almost-Linear	88,640.16 (6399.78)	72,204.04 (753.03)	942,541.41 (13273.37)
Discrete Boundary Value	142,872.55 (16285.88)	68,829.80 (1205.76)	1,412,591.21 (19916.05)
Discrete Integral Equation	170,963.99 (19170.73)	112,284.62 (721.91)	142,210.10 (11636.73)
Broyden Tridiagonal	69,791.48 (4071.08)	26,786.60 (348.40)	473,466.72 (6625.97)
Broyden Banded	91,703.76 (4934.80)	15,279.28 (165.98)	473,466.72 (6625.97)
Linear Function – Full Rank	115,467.93 (11176.48)	46,730.66 (900.64)	1,320,508.47 (19257.26)
Linear Function – Rank 1	162,938.86 (20,618.71)	112,057.14 (926.10)	142,210.10 (11636.73)
Ackley	453,153.51 (64,588.20)	355,482.96 (3976.48)	6,241,984.99 (102133.16)
Rastrigin	203,772.79 (19900.39)	84,665.57 (915.34)	1,713,659.81 (26601.73)
Griewank	1.62×10^7 (5.99×10^6)	1.30×10^7 (1.32×10^5)	2.03×10^8 (3.49×10^6)
Keane	1.23×10^6 (9.84×10^4)	9.17×10^5 (9.93×10^3)	6.23×10^6 (9.33×10^4)

and USGD-Fast yield even better improvements over DS on 15 of the 16 problems (all except the Keane problem). Moreover, on the Keane problem, DS is only slightly better than USGD or USGD-Fast. These results suggest that moving in the direction that makes an acute angle with the negative simplex gradient during the initialization phase for a surrogate-based optimization method results in significant improvement on the objective function value on high dimensional problems.

Tables 4 and 5 show the means and standard errors of the condition numbers of the resulting linear interpolation matrices for the three initialization strategies on each of the test problems. The means of the condition numbers for USGD are worse than those for DS on 10 of the 17 problems in

Table 4 and on all 16 problems in Table 5. However, they are better than those for SS on 16 of the problems in Table 4 (all except the 200-D Extended Powell Singular) and on 12 of the problems in Table 5. Moreover, the means of the condition numbers for USGD are 481.84 for GWB72 and $< 4 \times 10^4$ on 14 of the 200-D problems (all except Griewank200 and Keane200), and these are well within the threshold condition number $\kappa_{\max} = 10^5$ for GWB72 and the 200-D problems. In addition, the means of the condition numbers for USGD-Fast are $< 5 \times 10^5$ on 14 of the 1000-D problems (all except Griewank1000 and Keane1000), and these are again within the threshold condition number $\kappa_{\max} = 10^6$ for the 1000-D problems.

4.5 Performance and Data Profiles

In the second set of experiments, DYCORS-DDSRBF and PS-RBF initialized by USGD are compared with these same algorithms initialized by SS and DS using performance and data profiles (Moré and Wild 2009). Let $\mathcal{P}$ be the set of problems where a given problem p corresponds to a particular test problem and a particular starting point (prior to the initialization strategy). The performance and data profiles are created separately for the 200-D and 1000-D test problems. Since there are sixteen 200-D (and also sixteen 1000-D) problems and 30 starting points (corresponding to the 30 trials), there are $16 \times 30 = 480$ problems for the profiles. Moreover, let $\mathcal{S}$ be the set of solvers. Here, there are 6 solvers (DYCORS-DDSRBF (USGD), DYCORS-DDSRBF (DS), DYCORS-DDSRBF (SS), PS-RBF (USGD), PS-RBF (DS) and PS-RBF (SS)). For any pair (p, s) of a problem p and a solver s, the *performance ratio* is defined by

$$r_{p,s} := \frac{t_{p,s}}{\min\{t_{p,s} \ : \ s \in \mathcal{S}\}},$$

where $t_{p,s}$ is the number of function evaluations required to satisfy the convergence test that is defined below. Note that $r_{p,s} \geq 1$ for any $p \in \mathcal{P}$ and $s \in \mathcal{S}$. Moreover, for a given problem p, the best solver s for this problem attains $r_{p,s} = 1$. Furthermore, by convention, set $r_{p,s} = \infty$ whenever solver s fails to yield a solution that satisfies the convergence test.

Now, for any solver $s \in \mathcal{S}$ and for any $\alpha \geq 1$, the *performance profile of s with respect to α* is the fraction of problems where the performance ratio is at most α, i.e.,

$$\rho_s(\alpha) = \frac{1}{|\mathcal{P}|} \left|\{p \in \mathcal{P} \ : \ r_{p,s} \leq \alpha\}\right|.$$

For any solver $s \in \mathcal{S}$, the *performance profile curve of s* is the graph of the performance profiles of s for a range of values of α.

Derivative-free algorithms for expensive black-box optimization are typically compared given a fixed and relatively limited number of function evaluations. In particular, the convergence test by Moré and Wild (2009) uses a tolerance $\tau > 0$ and the minimum function value f_L obtained by *any* of the solvers on a particular problem within a given number μ_f of function evaluations and it checks if a point x obtained by a solver satisfies

$$f(x_0) - f(x) \geq (1 - \tau)(f(x_0) - f_L),$$

where x_0 is a starting point corresponding to the problem under consideration. In the above expression, x is required to achieve a reduction that is $1 - \tau$ times the best possible reduction $f(x_0) - f_L$.

Next, given a solver $s \in \mathcal{S}$ and $\alpha > 0$, the *data profile of a solver s with respect to α* (Moré and Wild 2009) is given by

$$d_s(\alpha) = \frac{1}{|\mathcal{P}|} \left| \left\{ p \in \mathcal{P} \ : \ \frac{t_{p,s}}{n_p + 1} \leq \alpha \right\} \right|,$$

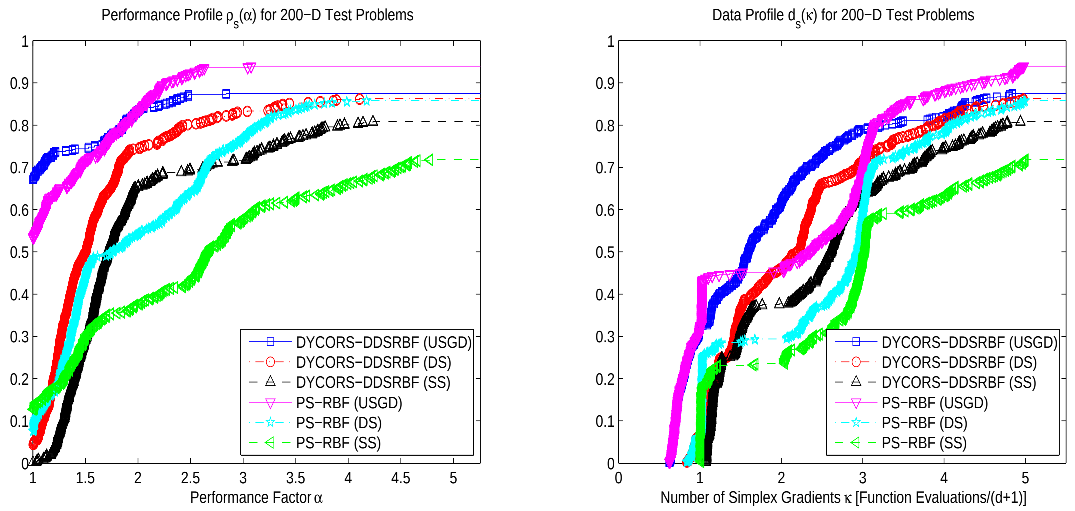


Figure 1: Performance and data profiles for the alternative optimization methods on the 200-D test problems.

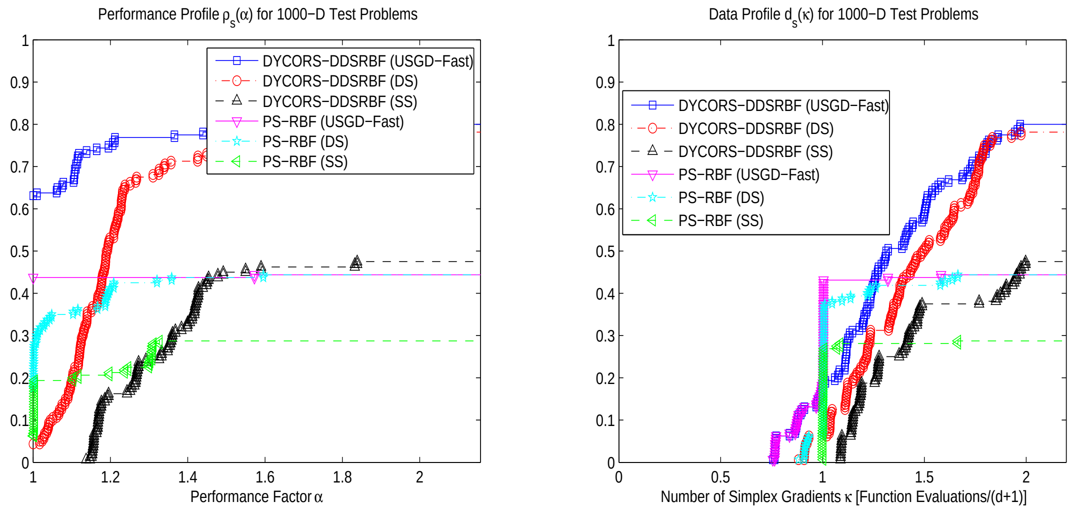


Figure 2: Performance and data profiles for the alternative optimization methods on the 1000-D test problems.

where $t_{p,s}$ is the number of function evaluations required by solver s to satisfy the convergence test on problem p and n_p is the number of variables in problem p. For any solver $s \in \mathcal{S}$, the *data profile curve of* s is the graph of the data profiles of s for a range of values of α. For a given solver s and any $\alpha > 0$, $d_s(\alpha)$ is the fraction of problems "solved" (i.e., problems where the solver generated a point satisfying the convergence test) by s within $\alpha(n_p + 1)$ function evaluations (equivalent to α simplex gradient estimates (Moré and Wild 2009)).

Figures 1 and 2 show the performance profile and data profiles (Moré and Wild 2009) for the 200-D and 1000-D problems. For the 200-D problems, all algorithms are run up to 1000 function evaluations, while for the 1000-D problems, the algorithms are run up to 2000 function evaluations. Recall, however, that the first 201 function evaluations for the 200-D problems and the first 1001 function evaluations on the 1000-D problems are spent on the initialization procedure (USGD, DS

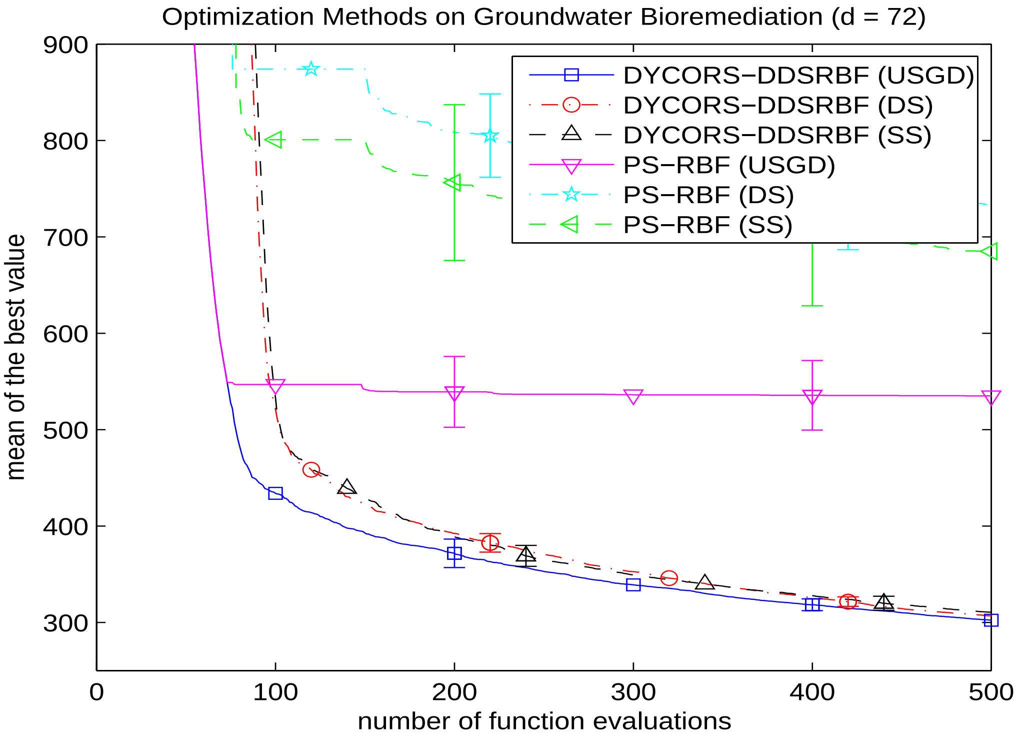


Figure 3: Mean of the best objective function value (over 30 trials) vs number of function evaluations for the alternative optimization methods on the 72-D groundwater bioremediation problem. Error bars represent 95% t-confidence intervals for the mean.

or SS). The parameter τ for the performance and data profiles described above is set to $\tau = 0.05$.

The performance and data profiles show that DYCORS-DDSRBF (USGD) is generally much better than both DYCORS-DDSRBF (SS) and DYCORS-DDSRBF (DS) on the 200-D and 1000-D test problems. Moreover, PS-RBF (USGD) is generally much better than both PS-RBF (SS) and PS-RBF (DS) on the 200-D problems and it is also generally better than PS-RBF (SS) on the 1000-D problems. The profiles for the 1000-D problems do not show any clear improvement of PS-RBF (USGD) over PS-RBF (DS). However, USGD does not seem to hurt the performance of PS-RBF on the 1000-D problems. Hence, these results suggest that the USGD initialization strategy is generally helpful in improving the performance of DYCORS-DDSRBF and, to a limited extent, of PS-RBF. These results also suggest that USGD is potentially helpful for other surrogate-based methods, especially those that require a maximal set of affinely independent points for initialization.

Recall, however, that the points generated by the various initialization strategies (USGD, DS and SS) are not fully integrated into PS-RBF as explained in Section 4.3. PS-RBF only uses the best point from the set of points generated by these initialization strategies and ignores the rest of the points. Hence, the improvements of PS-RBF (USGD) over either PS-RBF (SS) or PS-RBF (DS) are mostly due to the fact that the starting point of PS-RBF after USGD typically has a much better objective function value than the starting point of PS-RBF after either SS or DS. In contrast, DYCORS-DDSRBF uses all the points generated by these initialization strategies. Because of this, it would not be fair to compare DYCORS-DDSRBF and PS-RBF in this paper.

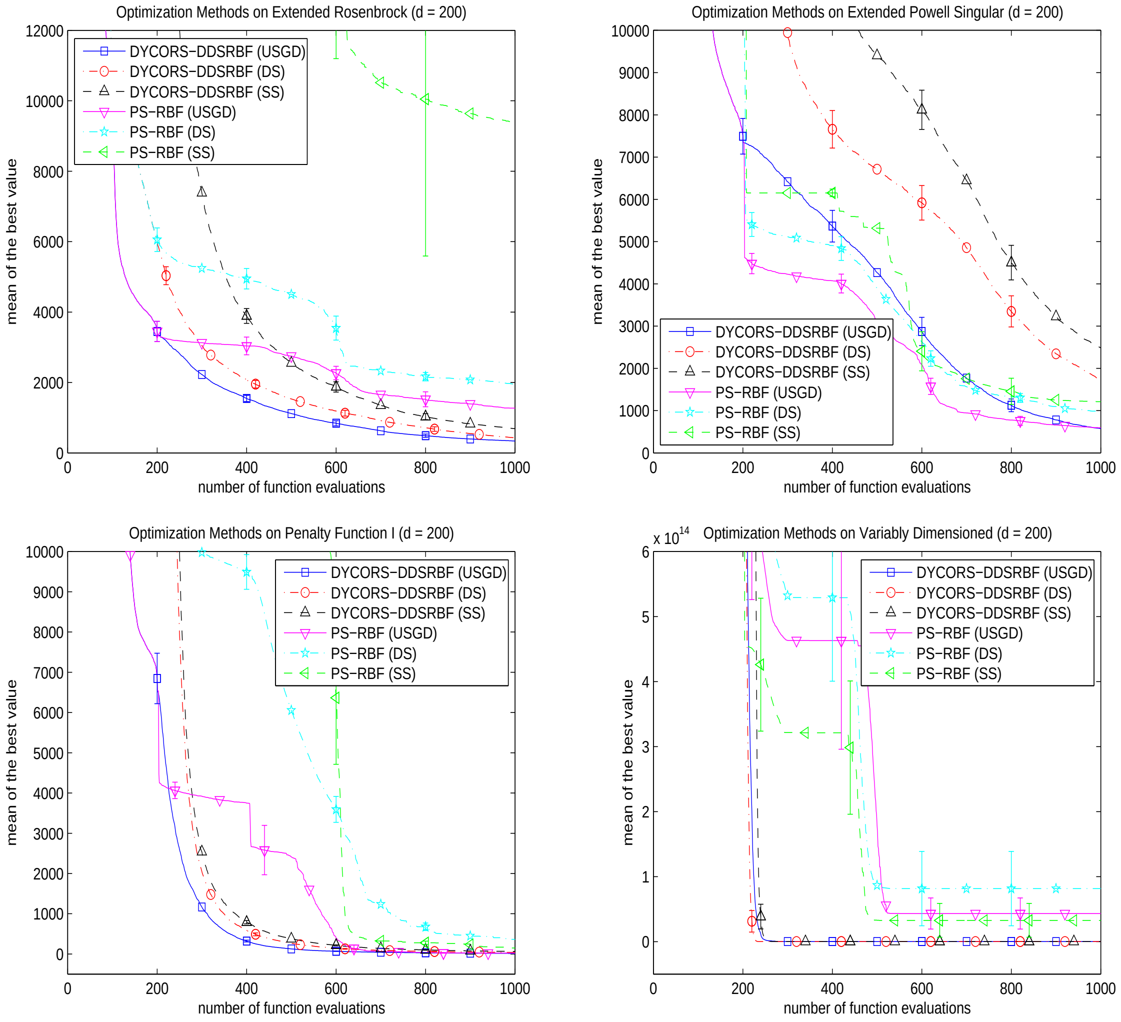


Figure 4: Mean of the best feasible objective function value (over 30 trials) vs number of function evaluations for the alternative optimization methods on the 200-D test problems. Error bars represent 95% t-confidence intervals for the mean.

4.6 Average Progress Curves

The different combinations of surrogate-based method and initialization strategy are also compared using average progress curves. An *average progress curve* is a graph of the mean of the best objective function value obtained by an algorithm as the number of function evaluations increases. Figures 3-10 show the average progress curves when the various optimization algorithms are applied to the 72-D groundwater bioremediation problem GWB72 and to the 200-D and 1000-D test problems. Figures 8-10 are in the appendix. The error bars in these figures represent 95% t confidence intervals for the mean. That is, each side of the error bar has length equal to 2.045 (for 30 trials) or 2.262 (for 10 trials) times the standard deviation of the best function value divided by the square root of the number of trials. Here, 2.045 and 2.262 are the critical values corresponding to a 95% confidence level for a t distribution with 29 and 9 degrees of freedom, respectively.

Figure 3 shows that DYCORS-DDSRBF (USGD) is better than both DYCORS-DDSRBF (SS) and DYCORS-DDSRBF (DS) on GWB72. Moreover, PS-RBF (USGD) is also better than both

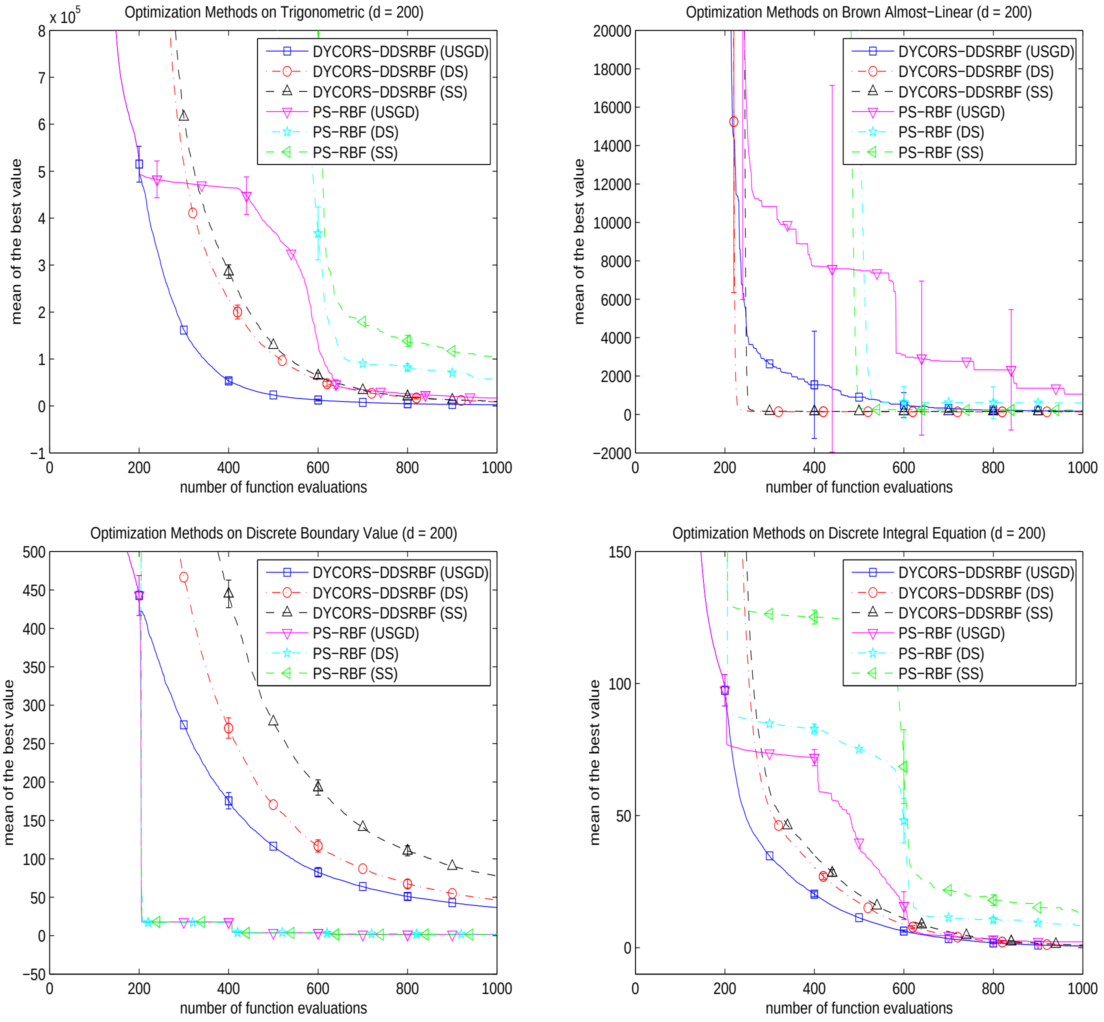
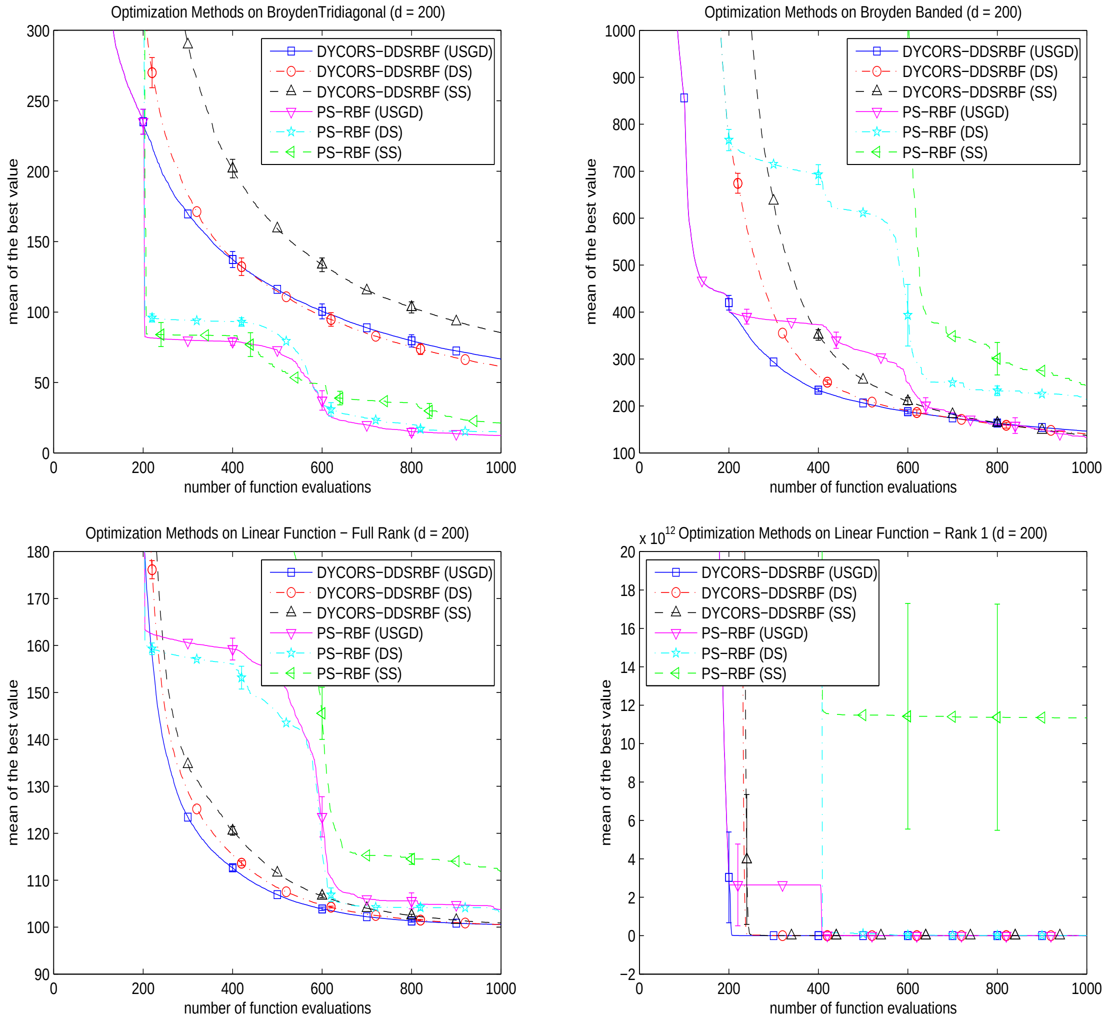


Figure 5: Mean of the best feasible objective function value (over 30 trials) vs number of function evaluations for the alternative optimization methods on the 200-D test problems. Error bars represent 95% t-confidence intervals for the mean.

PS-RBF (SS) and PS-RBF (DS) on GWB72. However, multiple trials of PS-RBF (USGD) appear to be stuck at a local minimum relatively early in the search.

Figures 4-7 show that DYCORS-DDSRBF (USGD) is better than both DYCORS-DDSRBF (SS) and DYCORS-DDSRBF (DS) on 13 of the 200-D test problems (all except Variably Dimensioned, Broyden Tridiagonal and Brown Almost-Linear). Moreover, the performance of DYCORS-DDSRBF (USGD) is comparable to that of the latter algorithms on the 200-D Variably Dimensioned and Broyden Tridiagonal problems. Note that although USGD yielded a much better objective function value than the SS and DS initialization methods on the 200-D Brown Almost-Linear problem, the performance of DYCORS-DDSRBF (USGD) is somewhat worse than DYCORS-DDSRBF (SS) and DYCORS-DDSRBF (DS) on this problem.

Figures 4-7 also show that PS-RBF (USGD) is better than both PS-RBF (SS) and PS-RBF (DS) on 12 of the 200-D test problems (all except Variably Dimensioned, Brown Almost-Linear, Discrete Boundary Value and Linear Function - Full Rank). Moreover, the performance of PS-RBF (USGD) is comparable to that of the latter algorithms on the 200-D Variably Dimensioned and


Optimization Methods on BroydenTridiagonal (d = 200)

Optimization Methods on Broyden Banded (d = 200)

Optimization Methods on Linear Function − Full Rank (d = 200)

Optimization Methods on Linear Function − Rank 1 (d = 200)

Figure 6: Mean of the best feasible objective function value (over 30 trials) vs number of function evaluations for the alternative optimization methods on the 200-D test problems. Error bars represent 95% t-confidence intervals for the mean.

Discrete Boundary Value problems. However, the performance of PS-RBF (USGD) is somewhat worse than that of PS-RBF (SS) and PS-RBF (DS) on the 200-D Brown Almost-Linear in later iterations even though Table 2 shows that the former started with better objective function values. In addition, PS-RBF (USGD) performed slightly worse than PS-RBF (DS) on the 200-D Linear Function - Full Rank even though the former also started with better objective function values.

Next, Figures 8-10 (see Appendix) show that DYCORS-DDSRBF (USGD) is better than both DYCORS-DDSRBF (SS) and DYCORS-DDSRBF (DS) on 12 of the 1000-D test problems (all except Extended Powell Singular, Variably Dimensioned, Rastrigin and Keane). Moreover, the performance of the former is comparable to that of the latter algorithms on the 1000-D Variably Dimensioned and Rastrigin problems.

Figures 8 and 10 also show that PS-RBF (USGD) is better than both PS-RBF (SS) and PS-RBF (DS) on seven of the 1000-D test problems (Extended Rosenbrock, Brown Almost-Linear, Broyden Banded, Linear Function - Full Rank, Ackley, Rastrigin and Griewank). Moreover, its performance is comparable to those of the latter algorithms on six other 1000-D test problems (Extended Powell

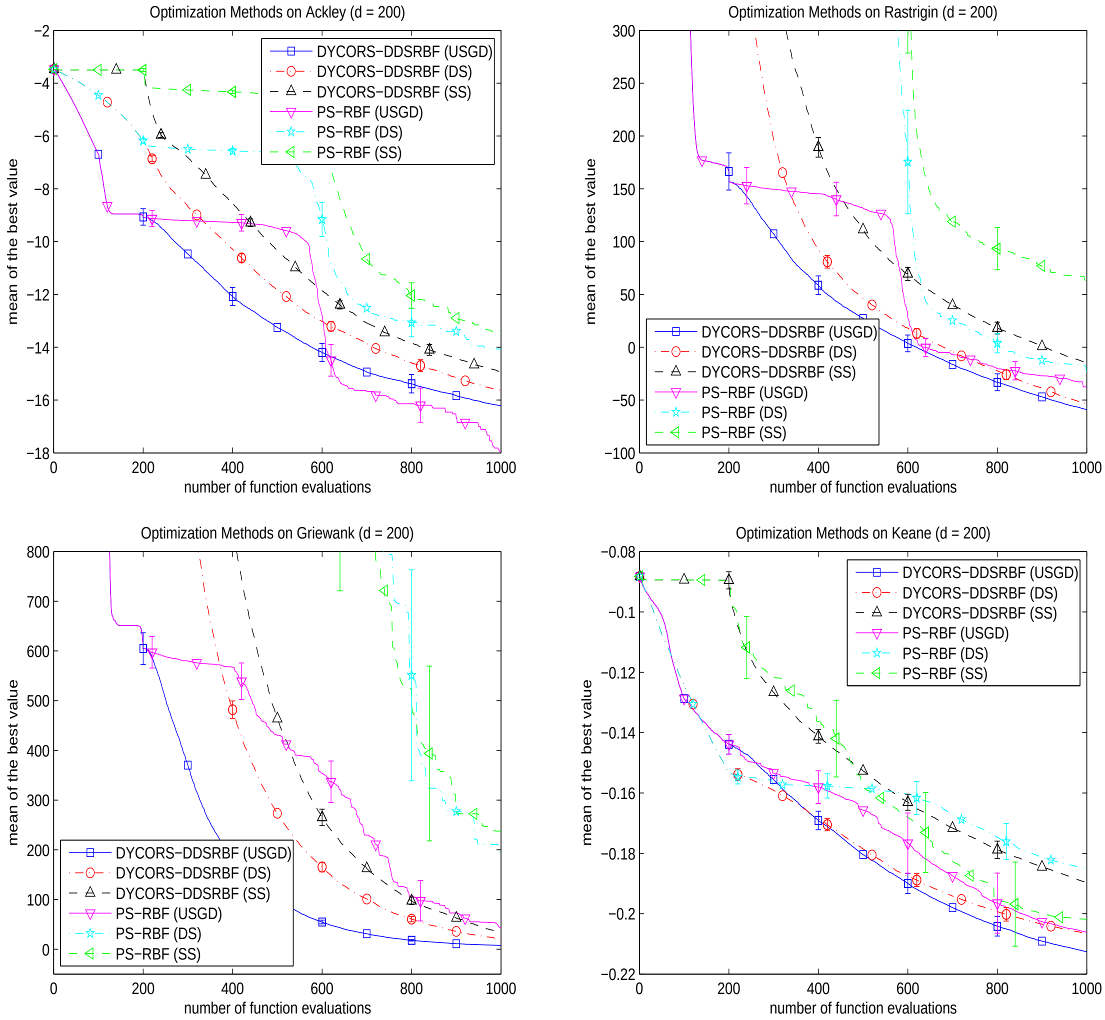


Figure 7: Mean of the best feasible objective function value (over 30 trials) vs number of function evaluations for the alternative optimization methods on the 200-D test problems with many local minima. Error bars represent 95% t-confidence intervals for the mean.

Singular, Penalty Function I, Variably Dimensioned, Discrete Boundary Value, Discrete Integral Equation and Broyden Tridiagonal). However, PS-RBF (USGD) is worse than PS-RBF (DS) on the 1000-D Trigonometric, Linear Function - Rank 1, and Keane problems.

Overall, USGD yielded improvements for DYCORS-DDSRBF over the simpler SS and DS initialization on the 72-D groundwater application and on most of the 200-D and 1000-D test problems. USGD also yielded improvements for PS-RBF over SS and DS on GWB72 and on many of the 200-D problems. Moreover, it yielded improvements for PS-RBF on only seven of the 1000-D problems but it did not really hurt performance on most of the remaining 1000-D test problems. As noted earlier, the main reason USGD was more effective for DYCORS-DDSRBF is that the USGD points are actually used by the algorithm whereas only the best of the USGD points is used by PS-RBF.

4.7 Running Times

To get an idea of the overhead computational effort required by the different algorithms, Table 6 reports the average running times on the 1000-D Extended Rosenbrock problem for 2000 function evaluations, excluding the time required by the initialization procedures. Note that DYCORS-DDSRBF have much longer average running times compared to PS-RBF. However, for truly expensive problems where each function evaluation could take hours, these running times are still much smaller than the total time required for all function evaluations.

Table 6: Average running times (over 10 trials) of the surrogate-based methods for 2000 function evaluations on the 1000-D Extended Rosenbrock problem. These running times exclude the times required by the initialization procedures.

Algorithm	Average Running Time
DYCORS-DDSRBF (USGD-Fast)	6070.18 sec (1.69 hrs)
DYCORS-DDSRBF (DS)	6146.93 sec (1.71 hrs)
DYCORS-DDSRBF (SS)	6127.76 sec (1.70 hrs)
PS-RBF (USGD-Fast)	370.34 sec
PS-RBF (DS)	390.42 sec
PS-RBF (SS)	475.85 sec

5 Summary and Conclusions

This paper presented an initialization strategy for surrogate-based optimization method called *Underdetermined Simplex Gradient Descent (USGD)* that generates a set of $d+1$ affinely independent points in $\mathbb{R}^d$ while making progress towards the optimum and also while keeping the condition number of the corresponding linear interpolation matrix within a reasonable value. Numerical experiments on 200-D and 1000-D instances of 16 well-known test problems and on a 72-D groundwater bioremediation application suggest that this approach results in much better objective function values compared to simpler and more standard initialization strategies, called *Static Simplex (SS)* and *Dynamic Simplex (DS)*.

This paper also compared the performance of the DYCORS-DDSRBF algorithm (Regis and Shoemaker 2012) initialized by USGD with the same algorithm initialized by the SS and DS strategies on the same test problems. Moreover, a pattern search algorithm that uses RBF surrogates (PS-RBF) (Le Thi et al. 2012) initialized by USGD was also compared with the same algorithm initialized by the SS and DS procedures. The numerical results given in the performance and data profiles and also the average progress curves consistently showed that DYCORS-DDSRBF (USGD) was a substantial improvement over both DYCORS-DDSRBF (SS) and DYCORS-DDSRBF (DS) on the 72-D groundwater bioremediation problem and on the 200-D and 1000-D test problems. Similarly, PS-RBF (USGD) was also generally an improvement over both PS-RBF (SS) and PS-RBF (DS) on the same set of problems. Overall, the numerical results suggest that the USGD initialization strategy is promising for surrogate-based algorithms for very high-dimensional expensive black-box problems involving hundreds of decision variables when only a relatively limited number of function evaluations can be performed.

Acknowledgements

I would like to thank Ismael Vaz and Luís Vicente for the PSwarm package, that includes a pattern search Matlab code with options for combining with RBF models. I am also grateful to Jorge Moré and Stefan Wild for their Matlab code that creates performance and data profiles.

References

1. Abramson, M.A., C. Audet. 2006. Convergence of mesh adaptive direct search to second-order stationary points. *SIAM Journal on Optimization* **17(2)** 606–619.

2. Aleman, D.M., H.E. Romeijn, J.F. Dempsey. 2009. A response surface approach to beam orientation optimization in intensity modulated radiation therapy treatment planning. *INFORMS Journal on Computing* **21(1)** 62–76.

3. Audet, C., J.E. Dennis, Jr. 2006. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization* **17(2)** 188–217.

4. Audet, C., J.E. Dennis, Jr., S. Le Digabel. 2008. Parallel space decomposition of the mesh adaptive direct search algorithm. *SIAM Journal on Optimization* **19(3)** 1150-1170.

5. Bettonvil, B., J.P.C. Kleijnen. 1997. Searching for important factors in simulation models with many factors: Sequential bifurcation. *European Journal of Operational Research* **96(1)** 180-194.

6. Björkman, M., K. Holmström. 2000. Global optimization of costly nonconvex functions using radial basis functions. *Optimization and Engineering* **1(4)** 373–397.

7. Booker, A.J., J.E. Dennis, Jr., P.D. Frank, D.B. Serafini, V. Torczon, M.W. Trosset. 1999. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization* **17(1)** 1–13.

8. Buhmann, M.D. 2003. *Radial Basis Functions.* Cambridge University Press, Cambridge, U.K.

9. Bull, A.D. 2011. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research* **12(Oct)** 2879–2904.

10. Cassioli, A., F. Schoen. 2011. Global optimization of expensive black box problems with a known lower bound. *Journal of Global Optimization*, DOI: 10.1007/s10898-011-9834-7.

11. Chambers, M., C. A. Mount-Campbell. 2002. Process optimization via neural network metamodeling. *International Journal of Production Economics* **79(2)** 93–100.

12. Chen, L.-L., C. Liao, W. -B. Lin, L. Chang, X. -M. Zhong. 2012. Hybrid-surrogate-model-based efficient global optimization for high-dimensional antenna design. *Progress In Electromagnetics Research*, **124** 85–100.

13. Conn, A.R., S. Le Digabel, 2013. Use of quadratic models with mesh-adaptive direct search for constrained black box optimization. *Optimization Methods and Software* **28(1)** 139–158.

14. Conn, A.R., K. Scheinberg, Ph.L. Toint. 1997. Recent progress in unconstrained nonlinear optimization without derivatives. *Mathematical Programming* **79(3)** 397–414.

15. Conn, A.R., K. Scheinberg, L.N. Vicente. 2008a. Geometry of interpolation sets in derivative free optimization. *Mathematical Programming* **111(1-2)** 141–172.

16. Conn, A.R., K. Scheinberg, L.N. Vicente. 2008b. Geometry of sample sets in derivative-free optimization: polynomial regression and underdetermined interpolation. *IMA Journal of Numerical Analysis* **28(4)** 721–748.

17. Conn, A.R., K. Scheinberg, L.N. Vicente. 2009a. Global convergence of general derivative-free trust-region algorithms to first- and second-order critical points. *SIAM Journal on Optimization* **20(1)** 387–415.

18. Conn, A.R., K. Scheinberg, L.N. Vicente. 2009b. *Introduction to Derivative-Free Optimization.* SIAM, Philadelphia, PA.

19. Cressie, N. 1993. *Statistics for Spatial Data.* Wiley, New York.

20. Custódio, A.L., H. Rocha, L.N. Vicente. 2010. Incorporating minimum Frobenius norm models in direct search. *Computational Optimization and Applications* **46(2)** 265–278.

21. Custódio, A.L., Vicente, L.N. 2007. Using sampling and simplex derivatives in pattern search methods. *SIAM Journal on Optimization* **18(2)** 537–555.

22. Egea, J.A., E. Vazquez, J.R. Banga, R. Marti. 2009. Improved scatter search for the global optimization of computationally expensive dynamic models. *Journal of Global Optimization* **43(2-3)** 175–190.

23. García-Palomares, U. M., I. J. García-Urrea, P. S. Rodríguez-Hernández. 2012. On sequential and parallel non-monotone derivative-free algorithms for box constrained optimization. *Optimization Methods and Software*, DOI:10.1080/10556788.2012.693926.

24. Gray, G.A., T.G. Kolda. 2006. Algorithm 856: APPSPACK 4.0: asynchronous parallel pattern search for derivative-free optimization. *ACM Transactions on Mathematical Software* **32(3)**, 485–507.

25. Gutmann, H.-M. 2001. A radial basis function method for global optimization. *Journal of Global Optimization* **19(3)** 201–227.

26. Hansen, N. 2006. The CMA evolution strategy: a comparing review. J.A. Lozano, P. Larranga, I. Inza, E. Bengoetxea, eds. *Towards a new evolutionary computation.* Springer. 75–102.

27. Hansen, N., A. Ostermeier. 2001. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation* **9(2)** 159–195.

28. Holmström, K. 2008. An adaptive radial basis algorithm (ARBF) for expensive black-box global optimization. *Journal of Global Optimization* **41(3)** 447–464.

29. Huang, D., T.T. Allen, W.I. Notz, N. Zeng. 2006. Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of Global Optimization* **34(3)** 441–466.

30. Jakobsson, S., M. Patriksson, J. Rudholm, A. Wojciechowski. 2010. A method for simulation based optimization using radial basis functions. *Optimization and Engineering* **11(4)**, 501–532.

31. Jin Y. 2011. Surrogate-assisted evolutionary computation: recent advances and future challenges. Swarm and Evolutionary Computation **1(2)** 61–70.

32. Jin, Y., M. Olhofer, B. Sendhoff. 2002. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transactions on Evolutionary Computation* **6(5)** 481–494.

33. Jones, D.R., M. Schonlau, W.J. Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization* **13(4)** 455–492.

34. Jones, D.R. 2008. Large-scale multi-disciplinary mass optimization in the auto industry. Presented at the *Modeling and Optimization: Theory and Applications (MOPTA) 2008 Conference*, Ontario, Canada.

35. Kolda, T.G., R.M. Lewis, V. Torczon. 2003. Optimization by direct search: new perspectives on some classical and modern methods. *SIAM Review* **45(3)** 385–482.

36. Kolda, T.G., V.J. Torczon. 2004. On the convergence of asynchronous parallel pattern search. *SIAM Journal on Optimization* **14(4)** 939–964.

37. Le Digabel, S., 2011. Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm. *ACM Transactions on Mathematical Software* **37(4)** 44:144:15.

38. Le Thi H.A., Vaz, A.I.F., Vicente, L.N. 2012. Optimizing radial basis functions by D.C. programming and its use in direct search for global derivative-free optimization. *TOP* **20(1)** 190–214.

39. Loshchilov, I., M. Schoenauer, M. Sebag. 2012. Self-adaptive surrogate-assisted covariance matrix adaptation evolution strategy. In: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2012)*, ACM Press.

40. Marsden, A.L., M. Wang, J.E. Dennis, Jr., P. Moin. 2004. Optimal aeroacoustic shape design using the surrogate management framework. *Optimization and Engineering* **5(2)** 235–262.

41. Minsker, B.S., C.A. Shoemaker. 1998. Dynamic optimal control of in-situ bioremediation of groundwater. *Journal of Water Resources Planning and Management* **124(3)** 149–161.

42. Moré, J., B. Garbow, K. Hillstrom. 1981. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software* **7(1)** 17-41.

43. Moré, J., S. Wild. 2009. Benchmarking derivative-free optimization algorithms. *SIAM Journal on Optimization* **20 (1)** 172–191.

44. Myers, R.H., D.C. Montgomery. 2009. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments, 3rd Edition.* Wiley, New York.

45. Oeuvray, R., M. Bierlaire. 2009. BOOSTERS: A derivative-free algorithm based on radial basis functions. *International Journal of Modelling and Simulation* **29(1)** 26–36.

46. Oeuvray, R. 2005. Trust-Region Methods Based On Radial Basis Functions With Application To Biomedical Imaging. *Ph.D. thesis*, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland.

47. Parr, J.M., A.J. Keane, A.I.J., Forrester, C.M.E. Holden. 2012. Infill sampling criteria for surrogate-based optimization with constraint handling. *Engineering Optimization* **44(10)** 1147–1166.

48. Plantenga, T., Kolda, T. 2009. HOPSPACK: Software Framework for Parallel Derivative-Free Optimization. *Sandia Technical Report (SAND 2009-6265).*

49. Powell, M.J.D. 1992. The theory of radial basis function approximation in 1990. W. Light, ed. *Advances in Numerical Analysis, Volume 2: Wavelets, Subdivision Algorithms and Radial Basis Functions.* Oxford University Press, Oxford, U.K. 105–210.

50. Powell, M.J.D. 2002. UOBYQA: Unconstrained optimization by quadratic approximation. *Mathematical Programming* **92(3)** 555–582.

51. Powell, M.J.D. 2006. The NEWUOA software for unconstrained optimization without derivatives. G. Di Pillo and M. Roma, eds. *Large-Scale Nonlinear Optimization.* Springer, US. 255–297.

52. Regis, R.G. 2011. Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions. *Computers and Operations Research* **38(5)** 837–853.

53. Regis, R.G. 2013. Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Engineering Optimization*, DOI: 10.1080/0305215X.2013.765000.

54. Regis, R.G., C.A. Shoemaker. 2007a. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS Journal on Computing* **19(4)** 497–509.

55. Regis, R.G., C.A. Shoemaker. 2007b. Improved strategies for radial basis function methods for global optimization. *Journal of Global Optimization* **37(1)** 113–135.

56. Regis, R.G., C.A. Shoemaker. 2012. A quasi-multistart framework for global optimization of expensive functions using response surface models. *Journal of Global Optimization*, DOI: 10.1007/s10898-012-9940-1.

57. Regis, R.G., C.A. Shoemaker. 2013. Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization. *Engineering Optimization* **45(5)** 529–555.

58. Rocha, H., J. M. Dias, B. C. Ferreira, M. C. Lopes. 2012. Selection of intensity modulated radiation therapy treatment beam directions using radial basis functions within a pattern search methods framework. *Journal of Global Optimization*, DOI: 10.1007/s10898-012-0002-5.

59. Sacks, J., W.J. Welch, T.J. Mitchell, H.P. Wynn. 1989. Design and analysis of computer experiments. *Statistical Science* **4(4)** 409–435.

60. Scheinberg, K., Ph.L. Toint. 2010. Self-correcting geometry in model-based algorithms for derivative-free unconstrained optimization. *SIAM Journal on Optimization* **20(6)** 3512–3532.

61. Shan, S., G. Wang. 2010. Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions. *Structural and Multidisciplinary Optimization* **41(2)** 219–241.

62. Shoemaker, C.A., M. Willis, W. Zhang, J. Gossett. 2001. Model analysis of reductive dechlorination with data from Cape Canaveral field site. V. Magar, T. Vogel, C. Aelion, A. Leeson, eds. *Innovative Methods in Support of Bioremediation.* Battelle Press, Columbus, OH. 125–131.

63. Tolson, B.A., C.A. Shoemaker. 2007. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resources Research* **43, W01413**, doi:10.1029/2005WR004723.

64. Vapnik, V. 1995. *The Nature of Statistical Learning Theory.* Springer-Verlag New York, Inc. New York, NY, USA.

65. Vaz, A. I. F., L. N. Vicente. 2007. A particle swarm pattern search method for bound constrained global optimization. *Journal of Global Optimization* **39(2)** 197-219.

66. Vaz, A. I. F., L. N. Vicente. 2009. PSwarm: A hybrid solver for linearly constrained global derivative-free optimization. *Optimization Methods and Software* **24(4-5)**, 669-685.

67. Vazquez, E., J. Bect. 2010. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and Inference* **140(11)** 3088-3095.

68. Viana, F.A.C., R.T. Haftka, L.T. Watson. 2010. Why not run the efficient global optimization algorithm with multiple surrogates?. *51th AIAA/ASME /ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA 2010-3090, Orlando, USA.

69. Villemonteix, J., E. Vazquez, E. Walter. 2009. An informational approach to the global optimization of expensive-to-evaluate functions. *Journal of Global Optimization* **44(4)** 509–534.

70. Wild, S.M., R.G. Regis, C.A. Shoemaker. 2008. ORBIT: Optimization by radial basis function interpolation in trust-regions. *SIAM Journal on Scientific Computing* **30(6)** 3197-3219.

71. Wild, S.M., C.A. Shoemaker. 2011. Global convergence of radial basis function trust region derivative-free algorithms. *SIAM Journal on Optimization* **21(3)** 761–781.

72. Yoon, J.-H., C.A. Shoemaker. 1999. Comparison of optimization methods for ground-water bioremediation. *Journal of Water Resources Planning and Management* **125(1)** 54–63.

73. The Mathworks, Inc. 2009. *Matlab Optimization Toolbox: User's Guide, Version 4.* Natick, MA.

Appendix

Below are the average progress curves for the surrogate-based optimization algorithms on the 1000-D test problems. Because the computational overheads of running some of these methods are enormous, the algorithms are only run for 10 trials instead of 30 trials on each problem.

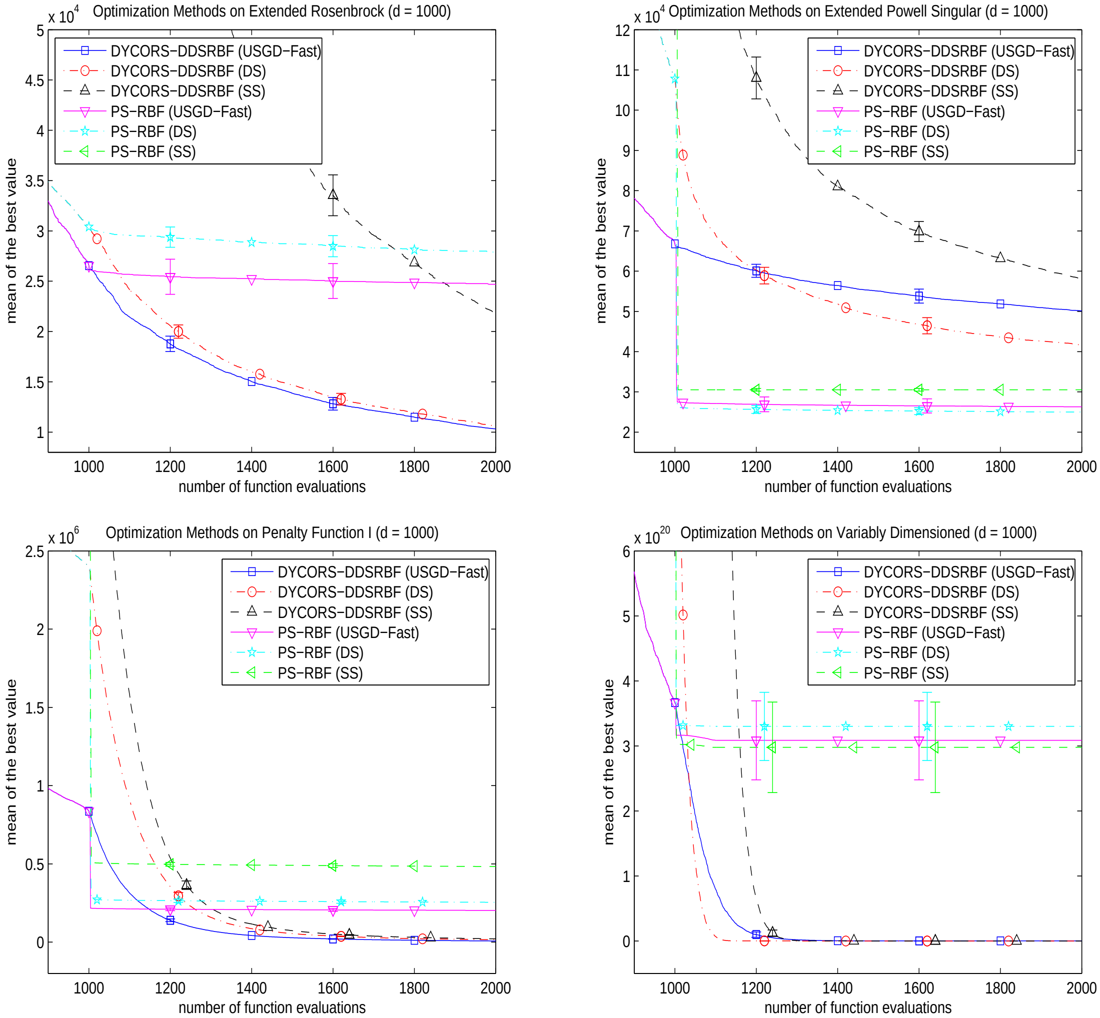


Figure 8: Mean of the best objective function value (over 10 trials) vs number of function evaluations for the alternative optimization methods on the 1000-D test problems. Error bars represent 95% t-confidence intervals for the mean.

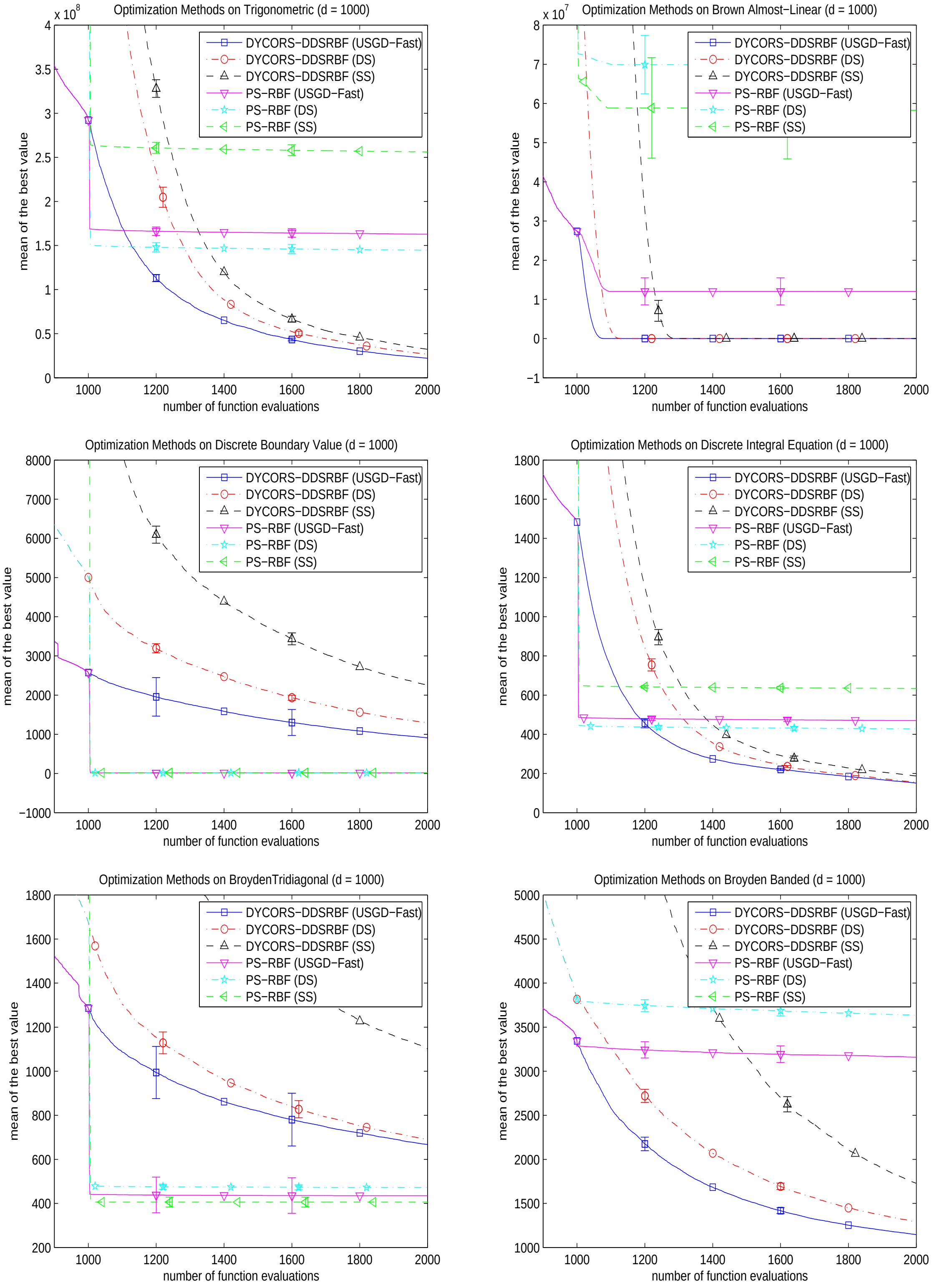


Figure 9: Mean of the best objective function value (over 10 trials) vs number of function evaluations for the alternative optimization methods on the 1000-D test problems. Error bars represent 95% t-confidence intervals for the mean.

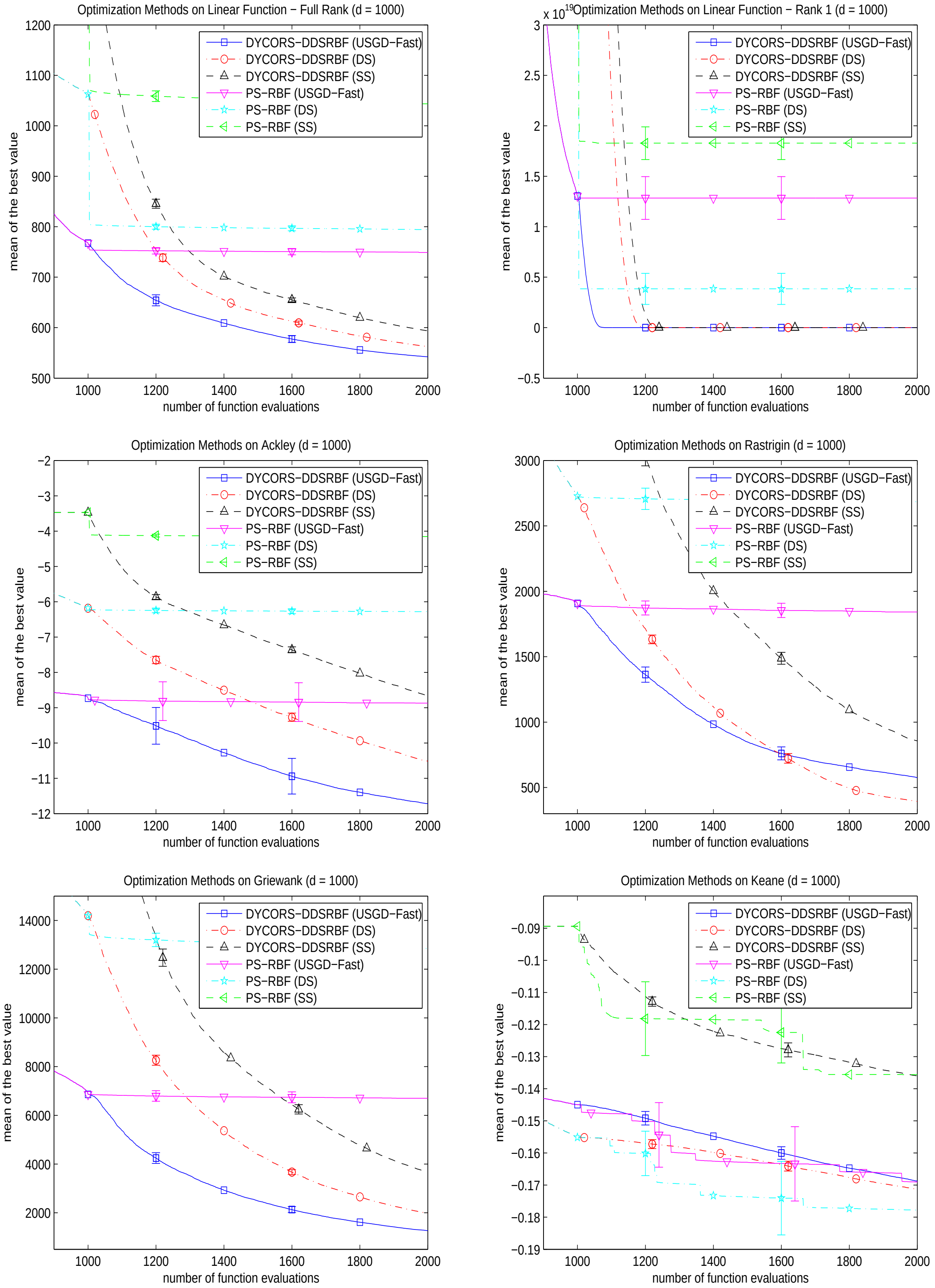


Figure 10: Mean of the best objective function value (over 10 trials) vs number of function evaluations for the alternative optimization methods on the 1000-D test problems. Error bars represent 95% t-confidence intervals for the mean.