International Journal of
Mathematical Modelling & Computations
Vol. 01, No. 04, 2011, 235 - 244

Introduction and Development of Surrogate Management Framework for Solving Optimization Problems

B. Azarkhalili[a], M. Rasouli[b], P. Moghadas[c], and B. Mehri[a]*

[a] *Mathematics Department, Sharif University of Technology, Azadi Ave, Tehran, Iran.*
[b] *Electrical Engineering Department, Sharif University of Technology, Azadi Ave, Tehran, Iran.*
[c] *Aerospace Engineering Department, Sharif University of Technology, Azadi Ave, Tehran, Iran.*

Received: 30 July 2011; Accepted: 20 September 2011.

Abstract. In this paper, we have outlined the surrogate management framework for optimization of expensive functions. An initial simple iterative method which we call the Strawman method illustrates how surrogates can be incorporated into optimization to stand in for the most expensive function. These ideas are made rigorous by incorporating them into the framework of pattern search methods. The SMF algorithm is presented, including mesh definition, and choice of polling points. In summarizing the ideas of surrogate-based optimization, we enrich this paper with an admittedly simplistic analogy which helps to compare optimization strategies.

Keywords: Optimization, Surrogate Method, Pattern Search, Polling, Searching Methods.

Index to information contained in this paper

1. Introduction

A growing numbers of engineering disciplines are now incorporating computer simulations into optimization problems, particularly for optimal design. In the case that the computational cost of these simulations is large, they present challenges that are not always met by traditional optimization techniques. In particular, these are problems in which the cost functions are expensive to evaluate, which are likely to be noisy or discontinuous, and for which there is often little or no available gradient information. In addition, multi-objective problems frequently require the use of several simulation codes to obtain cost function and constraint information.

*Corresponding author. Email: mehri@sharif.edu.

http://www.ijm2c.com

Computer simulations in engineering usually involve the solution of coupled sets of partial differential equations (PDE's). It is often desirable to use an optimization method that can treat a function as a "black box". Surrogate management framework (SMF) is an optimization method appropriate for this aim.

Imagine a hiker in some unexplored mountains who is trying to reach the lowest elevation point possible before dark. If our hiker uses a steepest-descent gradient method, then she is able to see the slope of the hill in her immediate surroundings. She points herself in the direction of steepest downhill, and walks for a short segment, and again looks for the steepest slope down. She continues this way until she reaches the bottom of the valley or the bottom of a high elevation bowl, and finds a point where the slope points uphill in all directions. There she sets up her camp.

On the other hand, let's suppose our hiker is using a surrogate based method. Perhaps she has a crude map in her backpack that indicates the general layout of the surrounding area, but is lacking full detail. This map acts as her surrogate. She sees that the next valley over the ridge looks promising. She gets out her paraglider and flies there, only to discover that the map was wrong and she's not at a lower point at all. Undeterred, she updates her map, and tries again. On her next flight she lands in a deep valley next to a great fishing lake. To push the analogy even further, the hiker should look around to make sure she's really at a flat spot on which to pitch her tent. Without this extra 'polling' step, she might be stuck sleeping on the side of the valley.

We see from our analogy that the gradient method will reliably and efficiently get the hiker to the bottom of the nearest valley. However, it offers her no way of hiking out of the valley if she finds that its depth is very shallow. The surrogate method offers her a potential to explore many valleys, however we also see that if her map is very inaccurate she may take several paragliding trips before finding a superior camp spot.

2. Incorporating Surrogate into Optimization

One method of surrogate construction, loosely termed approximation modeling, is to approximate or interpolate the true function value. In fact, even the linear approximation resulting from the use of the function value and a change depending on the approximation of the first derivative used in the Newton method [4] can be thought of as a surrogate of the true function. In general, one constructs a surrogate function to fit through a set of known data points. The surrogate functions can be polynomials, in which case the models are called response surface models or other interpolating functions such as Splines or Kriging functions. In these methods, one performs many predictive evaluations on the surrogate, which is cheap to evaluate, rather than on the expensive actual function. Using this method, many expensive evaluations of the true function can be avoided, resulting in a large cost savings.

Pattern search algorithms belong to the class of direct search optimization methods for numerical optimization. Direct search methods include many different optimization algorithms which do not explicitly use derivatives. Pattern search methods, including SMF, are characterized by two main requirements. First, they are restricted to a mesh, or lattice, which defines all potential locations where the function will be evaluated. This mesh is refined as the algorithm proceeds to generate a sequence of meshes. Second, pattern search methods rely on a set of polling conditions. A set of poll points defines a pattern which is used to search around the current iterates. The concept of polling and examples of poll sets will be discussed in Section 3.

Surrogates are a particularly good match for combining with pattern search methods. One reason for this is the inherent flexibility of pattern search methods, because they can be separated into a SEARCH step, which offers the user flexibility, and a POLL step which provides the basis for proof of convergence. Because of this, a surrogate can be incorporated to search the design space in a matter well-suited to the problem, and the choice of surrogate is flexible.

2.1 *Concept Illustration Using the "Strawman" Method*

Before outlining the SMF algorithm in detail, we will first look at a simpler method, which we term the "Strawman" method, which gives a preview of the concept of optimization using surrogates. Use of approximation modeling methods is also illustrated with example functions in Torczon & Trosset (1998b). The Strawman method is simply implemented as follows.

Let us assume we wish to find a minimum of the one-dimensional function $y = f(x)$ within an allowable domain $x_{\min} \leqslant x \leqslant x_{\max}$. First, we begin with a set of initial data points $\bar{x} = [x1, x2, ..., xn]$ where the function values are known. A surrogate model is constructed to fit through the known data points and approximates the actual function. We express the surrogate function as $y = \hat{f}(x)$. Because the surrogate function is inexpensive to evaluate, we can easily search for the minimum of the surrogate (within the allowable range of x) using standard optimization methods. In each iteration of the Strawman method, the surrogate is used to predict the minimum of the actual function. Then, the actual function value is computed at the predicted minimum and the surrogate model is updated to incorporate the new data. This process continues iteratively until convergence. Possible criteria for convergence are when sufficient cost function reduction has been achieved, or when the minimum point does not change from one iteration to the next. In summary, the steps in the Strawman algorithm (for one or more parameters) are as follows:

1. Fit a surrogate function through the set of known data points.
2. Estimate the function minimizer using the surrogate function.
3. Evaluate the true function value at the estimated minimum.
4. Check for convergence.
5. Update surrogate using new data point
6. Iterate until convergence.

To illustrate the Strawman method, we first give a one-dimensional example in which the exact function is known. The equation for a damped harmonic oscillator is

$$m\ddot{y} + k\dot{y} + c = 0 \tag{1}$$

Where $y(t)$ is the position of a mass at time t. The general solution to the above equation for the case of decaying oscillations is

$$y(t) = ae^{-\gamma t}\cos(w1t + \varphi) \tag{2}$$

Where $\gamma = c/2m$, $w1 = \sqrt{(w0^2 - \gamma^2)}$, and $w0 = \sqrt{k/m}$. As an example, we use the particular solution

$$y(t) = e^{-2t}\cos(5\pi t) \tag{3}$$

If we wish to find the minimum height reached by the oscillating mass in the time interval $0 \leqslant t \leqslant 1$, and the corresponding time t, we need to minimize the above cost function $y(t)$. The exact solution is plotted as the solid line in Figure 1 in the interval $0 \leqslant t \leqslant 1$. Let's assume that we do not know the exact function, and we start with two known initial data points at $t = 0.3$ and $t = 0.4$ as shown in the upper left plot. The function values at these two points are $y = 0.0$ and $y = 0.4493$, respectively. Given these two data points, our initial lowest cost function value is $y = 0.0$.

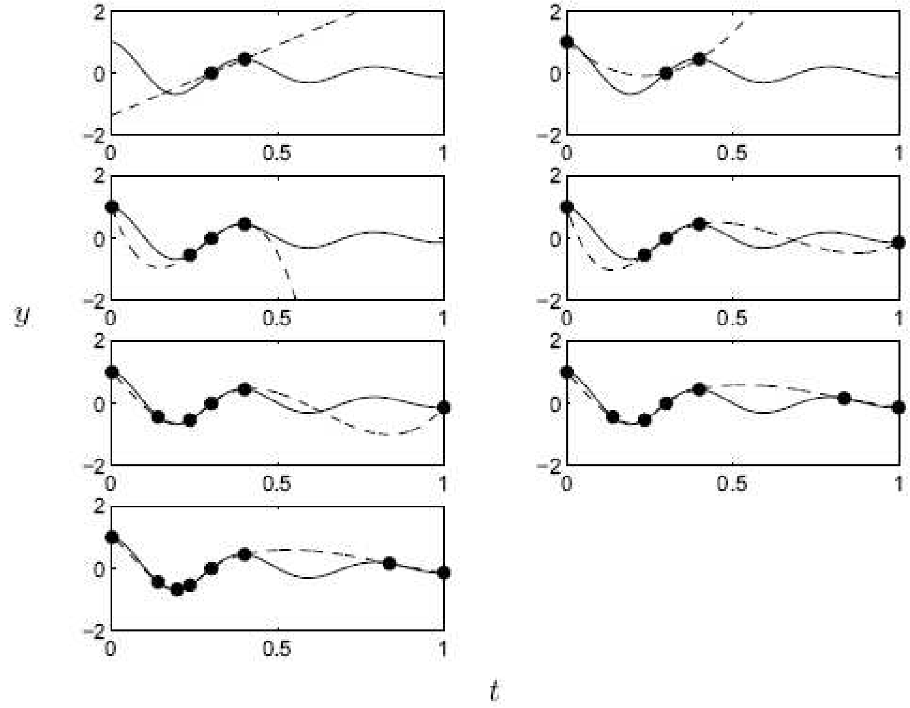

Figure 1. Exact solution (Solid Line) and surrogate fit (Dashed Line) for seven iterations of the Strawman method for the oscillator example. known data points are given y dots.

Figure 1 illustrates seven steps of the"Strawman" algorithm in which we approximate the known data using a spline as our surrogate function. From the first plot, the spline fit (dashed line) predicts the minimum to be at $t = 0$. Evaluating the function at this point and updating the surrogate gives the three points and corresponding spline fit shown in the second plot (upper right). We continue this process of evaluating the true function at the point given by minimizing the surrogate and we see that the surrogate function evolves as shown in the remaining plots in the figure. By the last step, we have obtained a value of $y = -0.6754$ at $t = 0.194$. The exact minimizer of the function is $t = 0.192$, and indeed the method will converge to this value if iterations continue. Our "cost function" which in this case is the minimum oscillator position, has been reduced from $y = 0.0$ in iteration 1, to $y = -0.6754$ in iteration 7. However, it should be pointed out that with a different, and unlucky choice of initial data, we could have easily converged to a different minimum value.

While the "Strawman" method is extremely easy to implement, and may produce a significant cost function reduction, it has several disadvantages. First, the Strawman approach is not strictly guaranteed to converge to a minimum of the true function. This can easily be seen by considering a counter example. Suppose that the initial data set consists of three points, where two points are symmetric around the center point. Using a parabolic surrogate function, the initial fit will be a symmetric parabola with its minimum point at the center data point. The Strawman approach will immediately converge to the location of the center point, which is a minimum of the surrogate, but is not necessarily a minimum of the actual function. Another disadvantage of the Strawman method is that the algorithm may take very small steps towards the minimum. We will see in the next section that this tendency may be avoided by restricting the algorithm to lie on a mesh.

3. Outline of the Surrogate Management Framework

In this section, we outline the general steps in the SMF algorithm. The surrogate management framework, introduced in Booker et al. (1999); Serafini (1998), is a pattern search method that incorporates surrogate functions to make the optimization cost effective. The main idea behind the SMF method is to use a surrogate function as a predictive tool, while retaining the robust convergence properties of pattern search methods. Like pattern search methods, SMF is a mesh based algorithm, so that all points evaluated are restricted to lie on a mesh. In this section, we outline the algorithm in general terms using known functions as examples.

We wish to solve the general optimization problem with bound constraints

$$\begin{array}{l} \min F(x) \\ \text{s.t. } \; x \in \Omega \end{array} \tag{4}$$

In the above problem statement, $F : R^n \to R$ is the cost function, and x is the vector of parameters. The bounds on the parameter space are de?ned by $\Omega = \{x \in R^n | l \leqslant x \leqslant u\}$ where $l \in R^n$ is a vector of lower bounds on x and $u \in R^n$ is a vector of upper bounds on x.

The first step in the optimization is to choose a set of initial data. Latin hypercube sampling (LHS), introduced by McKay et al. (1979), is commonly used to find a well distributed set of initial data in the parameter space, thus ensuring that each input variable has all portions of its range represented in the chosen data set. To choose a sample set of m vectors in the parameter space, each dimension is divided into m subintervals, and a point within each subinterval is selected (this is often done by randomly sampling from a uniform distribution over the subinterval). The sample set is then obtained by randomly grouping these points to form vectors. Consequently, for each dimension, each interval appears exactly once in the set.

Once the initial data set $\{x_1, \dots, x_m\}$ has been chosen, the cost function $J(x)$ is evaluated at these points, and an initial surrogate function is constructed, the surrogate interpolates the data and then it can be used to predict the value of the function at a particular location in the parameter space. As the optimization progresses, the surrogate should be updated to include new data.

After constructing an initial surrogate, all points subsequently evaluated by the algorithm are restricted to lie on a mesh in the parameter space. The mesh definition is flexible so long as it is defined by a set of vectors that positively span R^n (Lewis & Torczon, 1996). A positive spanning set of a matrix is simply the set of linear combinations of its column vectors with positive coe?cients. If none of the vectors in a given set can be formed from a non-negative combination of the others, the set is considered positively independent (Davis, 1954).

A positive basis is then defined as a positively independent set whose positive span is Rn .If we let D be a matrix whose columns form a positive spanning set in R^n , then the set of mesh points surrounding a point x are given by

$$M(x, \Delta) = \{x + \Delta Dz : z \in N^{nD}\} \tag{5}$$

Where Δ is the mesh size parameter, and n_D is the number of columns in D.The mesh may be refined or coarsened by changing $\Delta > 0$, and the mesh may be rotated from one iteration to the next as long as it satisfies this de?nition. Additional technical restrictions are discussed in Torczon (1997).

The SMF algorithm consists of two steps, SEARCH and POLL. The exploratory SEARCH step uses the surrogate to aid in the selection of points that are likely to improve the cost function. The SEARCH step provides means for local and global exploration of the parameter space, but is not strictly required for convergence. Because the SEARCH step is not integral to convergence, it affords the user a great deal of flexibility in choosing which points to evaluate and may be adapted by a knowledgeable user to a particular engineering problem. The only requirements of the SEARCH step are that it produces a finite number of points (which could be zero) and that these points lie on the mesh.

Convergence of the SMF algorithm is guaranteed by the POLL step, in which points neighboring the current best point on the mesh are evaluated in a positive spanning set of directions to check whether the current best point is a mesh local optimizer. A set of at least $n + 1$ poll points are required to generate a positive basis. An example of such a basis is constructed in R^n as follows. We let V be the matrix whose columns are the basis elements. Then construct $D = [V, -Ve]$, where e is the vector of ones and $-Ve$ is the negative sum of the columns of V. The columns of D form a positive basis for R^n using $n+1$ vectors. For example, in three dimensions such a basis could be given by $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$, $(-1, -1, -1)$. An example of directions making up a typical $2n$ positive basis in R^2 is shown on the left side of Figure 2, an example of directions forming an $n+1$ positive basis is shown in the center, and an example of three vectors that do not form a positive basis is shown on the right.

Following evaluation of the initial data, the first step in the optimization is a SEARCH step. In the SEARCH step, optimization is performed on the surrogate in order to predict the location of one or more minimizing points, and the function is evaluated at these points. If an improved cost function value is found, the SEARCH is considered successful, the surrogate is updated, and another SEARCH step is performed. If the SEARCH fails to find an improved

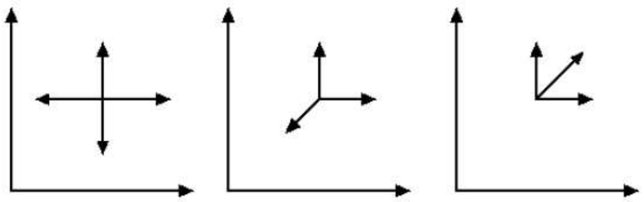

Figure 2. Positive bases in R^2 (Left and Center). Example of a set of typical $2n$ positive basis directions is shown on left, and a set of $n+1$ positive basis directions is shown in the center, and a set of $n+1$ vector which do not form a positive basis are shown on the right.

Point, then it is considered unsuccessful and a POLL step is performed, in which the cost function is evaluated at the set of poll points. It should be noted that as soon as an improved point is found, it is no longer necessary to evaluate the remainder of the set of poll points, thereby reducing the number of required function evaluations. If the POLL produces an improved point, then a SEARCH step is performed on the current mesh. Otherwise, if no improved points are found, then the current best point is defined to be a mesh local optimizer as in Audet & Dennis (2003). This terminology is to acknowledge that it may not be a local minimizer of the objective function on the mesh since the poll set is typically not the subset of all the points in the mesh adjacent to the poll center. For greater accuracy, the mesh may be refined, at which point the algorithm continues with a SEARCH. Convergence is reached when a local minimizer on the mesh is found, and the mesh has been refined to the desired accuracy. Each time new data points are found in a SEARCH or POLL step, the data is added to the surrogate and it is updated. The steps in the algorithm are summarized below, where the set of points in the initial mesh is M_0 , the mesh at iteration k is M_k, and the current best point is x_k.

1. Search
   a) Identify a finite set T_k of trial points on the mesh M_k.
   b) Evaluate $J(z)$ for all trial points $z \in T_k \subset M_k$.
   c) If for any trial point in T_k, $J(z) < J(x_k)$, a lower cost function value has been found, and the Search is successful. Increment k and go back to (a).
   d) Else, if no trial point in T_k improves the cost function, Search is unsuccessful. Increment k and go to Poll.
2. Poll
   a) Choose a set of positive spanning directions, and form the POLL set X_k as the set of mesh points adjacent to x_k in these directions.
   b) If $J(x_{poll}) < J(x_k)$ for any point $x_{poll} \in X_k$, then a lower cost function has been found and the POLL is successful. Set x_{k+1} to $xpoll$, increment k and go to SEARCH.
   c) Else, if no point in X_k improves the cost function, POLL is unsuccessful.
      i. If convergence criteria are satisfied, a converged solution has been found. STOP.
      ii. Else if convergence criteria are not met, refine mesh. Increment k and go to SEARCH.

The steps in the SMF algorithm are perhaps best illustrated graphically. As an example, Figures 3 through 7 illustrate surrogate construction, an unsuccessful SEARCH step, and an unsuccessful POLL step for a hypothetical two dimensional example. In these plots, x_1 and x_2 are the two optimization parameters, and f is the function value on the vertical axis. The mesh is the intersection of lines shown in the (x_1, x_2) plane.

Let us assume that we start by evaluating three points on the mesh as shown in Figure 3. One of these three points has a lower cost function value than the other two, and becomes the new incumbent point. Figure 4 illustrates surrogate building and selection of a search point. In this figure, the current incumbent point is marked with an asterisk. As indicated by the function surface sketched in the figure, a surrogate is fit through the three points to approximate the true function. The surrogate minimizer on the mesh is shown by the diamond in Figure 4.

Figure 5 illustrates the SEARCH step, in which the true function value is evaluated at the surrogate minimizer. The value obtained is shown by a dot, and in this case it is found that the true function value is higher than the current incumbent point. The SEARCH step has therefore failed to find an improved point, so the incumbent point remains the same.

After the SEARCH step fails, a POLL step is performed, and this is shown in Figure 6. Three points neighboring the incumbent point are evaluated, and are shown by open circles

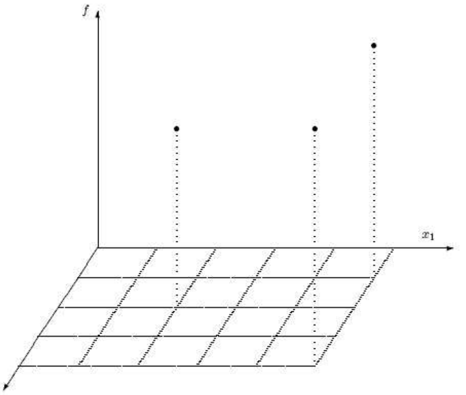

Figure 3. Evaluation of three initial points on the mesh. This is the first of five plots to illustrate the steps in the SMF method.

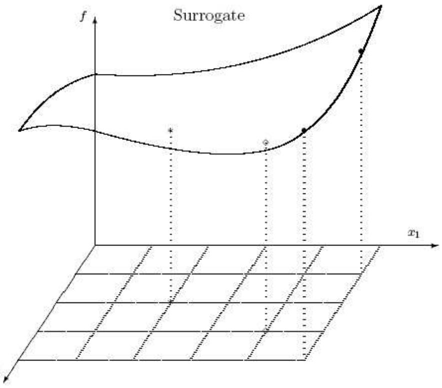

Figure 4. The Current incumbent point is marked with $a*$. A surrogate is constructed to approximate the function using known data points. The minimum of the surrogate is marked with a diamond (* is the incumbent solution).

In Figure 6, These three points form an n+1 positive basis in two dimensions. In this example, all the poll points are found to have a higher cost function value than the incumbent point, and so the POLL step is unsuccessful. Figure 7 shows that after the unsuccessful POLL step, the mesh in the parameter space is refined by half in each direction. At this point, a new iteration is performed, and the incumbent point remains the same.

An example of polling in two dimensions on a sequence of increasingly finer meshes is shown in Figure 8. The $n + 1$ poll points shown are $\{p^1, p^2, p^3\}$.The mesh M_k is the intersection of all lines, and the mesh is refined by half, and half again moving from left to right in the figure. We note that the poll directions can change, as shown in the figure, from one iteration to the next as long as they form a positive basis and points are evaluated on the mesh. Near the boundary of the parameter space, the poll directions need to conform to the boundaries for

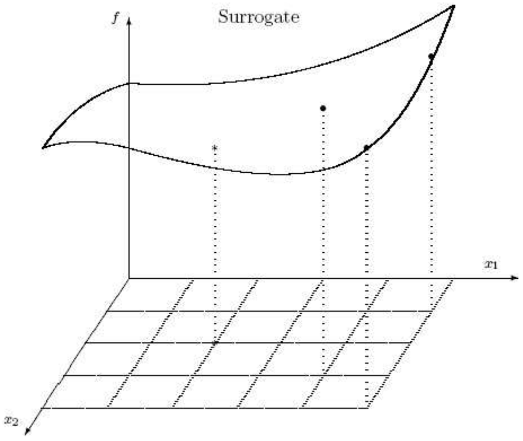

Figure 5. A SEARCH step is performed. The surrogate minimizer is evaluated by the true function, but is found to give a higher value than the incumbent point. The incumbent solution, still the same, is marked by $a*$ (* is still the incumbent solution).

convergence of the method. In the case shown, a 2n basis can be used since the boundary is rectangular.

Because the SMF method has distinct SEARCH and POLL steps, convergence theory for the method reduces to convergence of pattern search methods. Convergence of the SMF method is discussed at length by Booker et al. (1999) and by Serafini (1998).

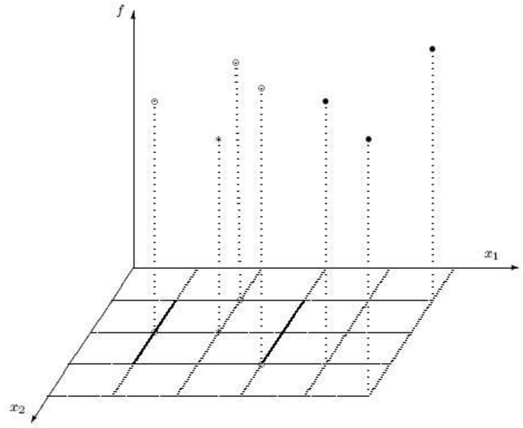

Figure 6. Polling around the incumbent solution, marked by $a*$. Three poll points are evaluated (shown by open circles) to form an $n + 1$ positive basis in two dimensions. The poll points in this example have higher cost function values than the incumbent. The POLL step is unsuccessful.

4. Conclusion

There are several variations of the surrogate management framework that can offer savings in computational cost, and improvements in cost functions reduction. In problems with full or even partial gradient information, large gains in efficiency and cost function reduction are often possible. Automatic differentiation (Bischof et al., 1992) and adjoint solvers (Jameson, 1995b,a; Jameson et al., 1998) are promising methods for obtaining gradient information, even for complex problems. The SMF method is general enough to incorporate gradient information in a number of ways.

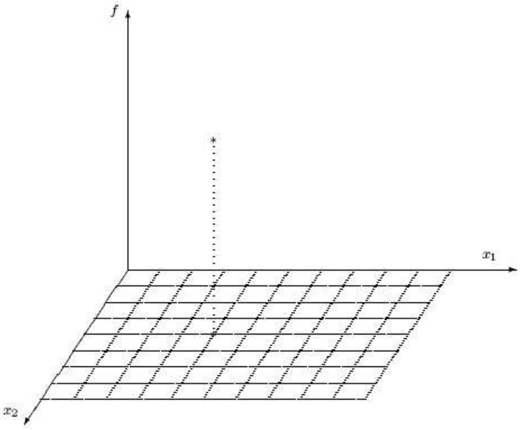

Figure 7. A new iteration begins, starting from the same incumbent point. The mesh is Refined by two in each direction, as shown in the (x_1, x_2) plane. The incumbent point, marked with $a*$ remains the same.

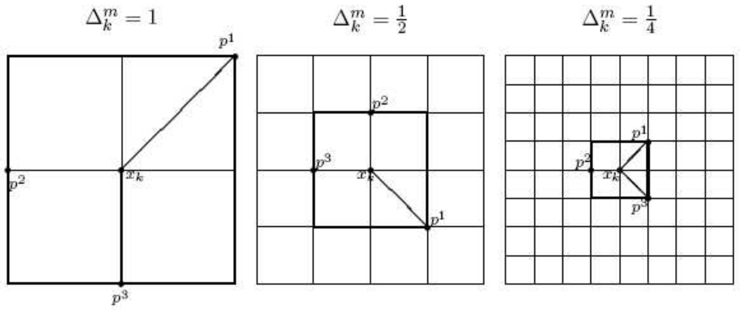

Figure 8. Example of polling using an $n+1$ positive basis. The poll set includes $n+1$ points, $X_k = \{x_k + \Delta_k^m d : d \in D_k\} = \{p^1, p^2, p^3\}$ for three subsequently finer meshes. In all three figures, the mesh Mk is the intersection of all lines.

Convergence theory is presented by Audet & Dennis (2003), Torczon (1997) and Lewis & Torczon (1999, 2000, 2002). An extensive review of methods for optimization by direct search is given by Kolda et al. (2003).

References

[1] Abramson M. A. Pattern search algorithms for mixed variable general constrained optimization problems. PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, TX, (2002).

[2] Abramson M. A., C. Audet, Dennis, J. R. Generalized pattern searches with derivative information. Mathematical Programming, Series B, **100**, (2004) 325.

[3] Booker A. J. Well-conditioned Kriging models for optimization of computer models. Mathematics and Computing Technology Report 002. Boeing Phantom Works, Seattle, WA, (2000).

[4] Booker A. J., Denni J. R., Frank J. E., Serafini P. D., Torczon D. B., V. & Trosset, M. W. A rigorous frameworks for optimization of expensive functions by surrogates. Structural Optimization **17 (1)**, (1999) 113.

[5] Gill P. E., Murray, W. & Wright, M. H. Practical Optimization. San Diego: Academic Press, (1981).

[6] Guinta A. A. Use of data sampling, surrogate models, and numerical optimization in engineering design. AIAA Paper, 2002-0538.

[7] Hansen N., Ostermeier, A. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. Proc. of the 1996 IEEE Int?l. Conf. on Evolutionary Computation, (1996), 312317.

[8] Lewis R. M., Torczon, V. A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds. SIAM J. Optim. **12**, (2002), 10751089.

[9] Ong Y. S., Nair, P. B., Keane, A. J. Evolutionary optimization of computationally expensive problems via surrogate modeling. AIAA J. **41 (4)**, (2003), 687696.

[10] Simpson T. W., Korte, J. J., Mauery, T. M. and Mistree, F. Comparison of response surface and kriging models for multidisciplinary design optimization. AIAA Paper (1998), 98-4755.