AIAA-2000-4935

IMPLEMENTATION OF A TRUST REGION MODEL MANAGEMENT STRATEGY IN THE DAKOTA OPTIMIZATION TOOLKIT

Anthony A. Giunta* and Michael S. Eldred*
Sandia National Laboratories, Optimization and Uncertainty Estimation Department †
Mail Stop 0847, Albuquerque, NM 87185 USA
e-mail: aagiunt@sandia.gov, mseldre@sandia.gov

Abstract

A trust region-based optimization method has been incorporated into the DAKOTA optimization software toolkit. This trust region approach is designed to manage surrogate models of the objective and constraint functions during the optimization process. In this method, the surrogate functions are employed in a sequence of optimization steps, where the original expensive objective and constraint functions are used to update the surrogates during the optimization process. This sequential approximate optimization (SAO) strategy is demonstrated on two test cases, with comparisons to optimization results obtained with a quasi-Newton method. For both test cases the SAO strategy exhibits desirable convergence trends. In the first test case involving a smooth function, the SAO strategy converges to a slightly better minimum than the quasi-Newton method, although it uses twice as many function evaluations. In the second test case involving a function with many local minima, the SAO strategy generally finds better local minima than does the quasi-Newton method. The performance of the SAO strategy on this second test case demonstrates the utility of using this optimization method on engineering optimization problems, many of which contain multiple local optima.

1 Introduction

Many engineering analysis and optimization problems involve computationally expensive physics-based simulation software. In some cases the computational expense of a full-physics simulation may be so great as to preclude the direct coupling of the simulation code to numerical optimization software. Another drawback to the use of optimization software is the existence of numerical noise (nonsmooth trends) in the output of the simulation software due to sources such as poorly resolved computational meshes, early termination of iterative solvers, and software errors. This numerical noise in the objective function and constraints inhibits traditional gradient-based optimization, while the computational expense of the simulation precludes gradient-free optimization.

Engineering optimization methods are needed to bridge the gap between gradient-based and gradient-free techniques. One such method is multi-start local optimization. Another more advanced approach is the implicit filtering method of Gilmore and Kelley [1]. A third approach is broadly classified as *sequential approximate optimization* (SAO) and has been widely practiced in the engineering community. Several types of SAO techniques are described by Haftka and Gürdal [2].

A typical SAO strategy decomposes the optimization process into a sequence of optimization subproblems, where each subproblem is confined to a small region of the parameter space. Then, in each subregion, the optimizer is given surrogate functions in place of the computationally expensive, or nonsmooth, objective function and constraints. These surrogate functions can take the form of surface fits to a small number of data samples, or the surrogates may be ob-

*Senior Member AIAA.

†This work was performed at Sandia National Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

tained from the physics simulation using a lower fidelity model. In the case where the surrogate models are provided by surface fits, this sequential optimization strategy is performed as follows:

1. compute the expensive and/or nonsmooth objective function and constraints at a small number of sample sites in the subregion,
2. construct surface fits to the objective function and constraint data,
3. optimize within the subregion using the surface fits as surrogates for the true objective function and constraints,
4. compute the true objective function and constraints at the optimum identified in Step 3,
5. check for convergence,
6. move/shrink/expand the subregion according to the accuracy of the approximation surfaces compared to the true function and constraint values, and
7. return to Step 1.

Various approaches have been developed in the engineering community for implementing Steps 1-7 ([3],[4]), but the convergence of these methods is problem dependent.

In conjunction with the engineering SAO efforts, there has been considerable research in the mathematics community on SAO techniques that are provably convergent to a local optimum ([5], [6], [7], [8]). These methods generally follow Steps 1-7, but with a gradient-matching requirement in Step 2. That is, the surface fits (or low fidelity models) must match the value and the gradient of the original (high fidelity) functions at the center of the current subregion. Thus, gradients are required for the original functions during each pass through Steps 1-7. While this approach is more expensive than a typical engineering SAO strategy, it is less than, or at most equal to, the expense of traditional gradient-based optimization [7].

In an ongoing project we are developing an SAO strategy that combines elements from a variety of sources in the engineering and mathematics fields. Our goal is to add an SAO capability to the DAKOTA optimization software package that will permit the user to choose between a provably convergent but potentially computationally expensive strategy, and a less reliable but possibly less computationally expensive strategy. This SAO strategy will augment the existing capabilities in gradient-based and gradient-free optimization methods in DAKOTA.

This paper describes the current status of this project and is organized as follows. Section 2 provides a brief overview of the capabilities in the DAKOTA optimization toolkit. Section 3 describes our formulation of an SAO strategy. Sections 4 and 5 demonstrate our SAO strategy on Rosenbrock's function and on a quasi-sinusoidal test function, respectively. Section 6 describes topics of future research and Section 7 summarizes our current work.

2 DAKOTA Software Overview

The DAKOTA (Design and Analysis Kit for OpTimizAtion) software [9] is an object-oriented suite of analysis and optimization methods under development at Sandia National Laboratories. DAKOTA provides a flexible framework for conducting parameter estimation, sensitivity analysis, uncertainty quantification, design of experiments sampling, and optimization. Included in the optimization tools are methods for solving continuous, discrete, and mixed continuous-discrete problems. The analysis and optimization methods in DAKOTA are designed to exploit massively parallel computers (typically having $\mathcal{O}[10^3 - 10^4]$ processors) which were developed under the Department of Energy's Accelerated Strategic Computing Initiative (ASCI). While intended for MP computers, DAKOTA also may be used on a single workstation or on a cluster of workstations. To date, DAKOTA has been ported to most common UNIX-based workstations including Sun, SGI, DEC, IBM, and LINUX-based PCs.

The DAKOTA software provides much of the infrastructure needed to implement an SAO strategy. DAKOTA has a library of surrogate modeling methods including polynomial regression [10], artificial neural networks [11], multivariate regression splines [12], and kriging interpolation [13]. Data sampling methods in DAKOTA include Latin hypercube sampling [14], orthogonal array sampling [15], and pure random sampling.

3 Trust Region Methods

3.1 Background

Consider a nonlinear inequality-constrained problem (NLP) of the form

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0,\ i = 1, \ldots, m \\ & x_l \leq x \leq x_u, \end{array} \tag{1}$$

where $x \in \Re^n$ is the vector of design variables, and x_l and x_u are the lower and upper bounds, respectively, on x.

In the SAO methods developed by Rodriguez, et al [5] and Alexandrov, et al [7], the NLP is recast into a simply constrained problem using an augmented Lagrangian function that combines the objective, constraints, Lagrange multipliers, and various penalty terms and/or slack variables. For this discussion, let $L(x)$ represent a generic augmented Lagrangian function. Next, an *approximate augmented Lagrangian* function is constructed using surrogates for the objective and constraints. These surrogates are denoted as $\hat{f}(x)$ and $\hat{g}_i(x)$, and we let $\hat{L}(x)$ represent the approximate augmented Lagrangian function constructed from the surrogates.

The generic SAO procedure outlined in Steps 1-7 (see Section 1) is performed in a sequence of k iterations, where the optimizer is restricted to a subregion of the design space (i.e., the trust region) during each iteration. For the k^{th} iteration the trust region is the parameter space where the design variables, x, satisfy

$$\| x - x_c^k \|_\infty \leq \Delta^k, \tag{2}$$

where x_c^k is the center point of the trust region, and the initial value for Δ^k at $k = 0$ is selected by the user. A specific strategy for updating Δ^{k+1} is described in Section 3.2.

One of the key elements in proving the convergence of the SAO methods developed in [5] and [7] is the enforcement of consistency conditions between the original functions and the surrogates. Specifically, the following conditions must hold at x_c^k

$$L(x_c^k) = \hat{L}(x_c^k), \tag{3}$$

$$\nabla L(x_c^k) = \nabla \hat{L}(x_c^k). \tag{4}$$

However, evaluating the term $\nabla L(x_c^k)$ can be computationally expensive if finite difference methods are used. Additional problems can occur in evaluating $\nabla L(x_c^k)$ when there are nonsmooth trends in the objective and constraint values.

For these reasons, it would be advantageous to have an SAO strategy that allows the user either to enforce or not enforce the consistency conditions. When gradients are easily evaluated and reliable, the user will have the advantage of a provably convergent SAO strategy. In cases where gradients cannot be easily or reliably estimated, an SAO strategy can still be employed, but without convergence guarantees.

3.2 SAO Strategy Development

The initial SAO strategy implemented in DAKOTA is a hybrid of the sequential quadratic programming method developed by Alexandrov, et al [7], and the trust region update methods employed by Rodriguez, et al [5]. Rather than implementing an augmented Lagrangian approach, we elected to retain separate objective and constraint functions. Our SAO formulation yields a sequence of k NLP trust region optimization subproblems of the form

$$\begin{array}{ll} \text{minimize} & \hat{f}^k(x) \\ \text{subject to} & \hat{g}_i^k(x) \leq 0, i = 1, \ldots, m \\ & x_l \leq x_l^k \leq x \leq x_u^k \leq x_u, \\ & k = 0, 1, 2, ..., k_{max}, \end{array} \tag{5}$$

where

$$x_l^k = x_c^k - \Delta^k, \tag{6}$$

$$x_u^k = x_c^k + \Delta^k. \tag{7}$$

The trust region update method used in this study is similar to the approach employed in Reference [5]. After each of the k iterations in the SAO strategy the trust region size, Δ^k, is updated based on a measure of the accuracy of the surrogate functions at the k^{th} optimum point, x_*^k. This measure of merit is designated ρ^k and is calculated using

$$\rho^k = \min(\rho_f^k, \rho_{g_i}^k), \text{ for } i = 1, ..., m, \tag{8}$$

where

$$\rho_f^k = \frac{f(x_c^k) - f(x_*^k)}{\hat{f}(x_c^k) - \hat{f}(x_*^k)}, \tag{9}$$

and

$$\rho_{g_i}^k = \frac{g_i(x_c^k) - g_i(x_*^k)}{\hat{g}_i(x_c^k) - \hat{g}_i(x_*^k)}. \tag{10}$$

Equations 9 and 10 are a measure of the actual versus predicted change in the function values at the k^{th} optimum. This value for ρ^k is then used in updating Δ^k as follows:

$$\begin{array}{rcll} \Delta^{k+1} & = & 0.25\Delta^k, & \text{if } \rho^k \leq 0.25, \\ & = & \Delta^k, & \text{if } 0.25 < \rho^k < 0.75, \\ & = & \gamma\Delta^k, & \text{if } \rho^k \geq 0.75, \end{array} \tag{11}$$

where

$$\begin{array}{rcll} \gamma & = & 2, & \text{if} \| x_*^k - x_c^k \|_\infty = \Delta^k, \\ & = & 1, & \text{if} \| x_*^k - x_c^k \|_\infty < \Delta^k. \end{array} \tag{12}$$

The starting point for the $k+1$ SAO iteration is determined by

$$\begin{array}{rll} x_c^{k+1} & = x_*^k, & \text{if } \rho^k > 0, \\ & = x_c^k, & \text{if } \rho^k \leq 0. \end{array} \tag{13}$$

The trust region updating procedure established in Equations 8–13 can be summarized by the following cases:

1. $\rho^k \leq 0$ – The surrogates are inaccurate. Reject the k^{th} optimum x_*^k and let $x_c^{k+1} = x_c^k$. Shrink the trust region by a factor of 0.25 to improve surrogate function accuracy.

2. $0 < \rho^k \leq 0.25$ – The surrogates are marginally accurate. Let $x_c^{k+1} = x_*^k$, but shrink the trust region size for the $k+1$ iteration.

3. $0.25 < \rho^k < 0.75$ – The surrogates are moderately accurate. Let $x_c^{k+1} = x_*^k$, and maintain the current trust region size.

4. $\rho^k \geq 0.75$ and $\| x_*^k - x_c^k \|_\infty < \Delta^k$ – The surrogates are accurate and x_*^k lies inside the trust region bounds. Let $x_c^{k+1} = x_*^k$ and maintain the current trust region size.

5. $\rho^k \geq 0.75$ and $\| x_*^k - x_c^k \|_\infty = \Delta^k$ – The surrogates are accurate and x_*^k lies on the trust region bounds. Let $x_c^{k+1} = x_*^k$ and increase the trust region size by a factor of two.

3.3 SAO Strategy Limitations

The current SAO strategy in DAKOTA does not enforce the consistency conditions in Equations 3 and 4. This is due to limitations in the design of experiments sampling methods and the surrogate function methods. Thus we cannot guarantee that this SAO strategy will converge to a local minimum. However, even with these limitations this SAO strategy can be applied to optimization problems. This is demonstrated below in an example problem involving Rosenbrock's function, and in a quasi-sinusoidal test problem that mimics the nonsmooth trends that can occur in some engineering optimization problems.

4 Rosenbrock Test Function

4.1 Background

Shown in Figure 1 is the well known Rosenbrock function

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2. \qquad (14)$$

The minimum of this function occurs at $(x_{1*}, x_{2*}) = (1, 1)$ where $f(x_{1*}, x_{2*}) = 0$.

To provide a basis of comparison, we first minimized Rosenbrock's function using the quasi-Newton algorithm (with a BFGS Hessian update strategy) available in the commercial optimizer DOT [16]. This provides a control case for comparison with the SAO optimization results. Both analytic gradients and finite difference gradients were used, with a step size of 0.0001 for the forward step finite difference method. The starting point for all optimization cases was $(x_1, x_2) = (0, 0)$, where $f(x_1, x_2) = 1.0$.

For unconstrained optimization, the quasi-Newton method in DOT was terminated if one of the following three conditions was met

$$\nabla f(x_1, x_2) < 0.0001, \qquad (15)$$

$$\left| \frac{(f^j - f^{j-1})}{f^{j-1}} \right| < 0.0001 \qquad (16)$$

$$\left| (f^j - f^{j-1}) \right| < 0.001, \qquad (17)$$

where j is the internal DOT iteration counter.

The SAO strategy uses DOT's quasi-Newton method to solve each of the k optimization problems in Equation 5. For each of the k iterations a surrogate model, $\hat{f}^k(x)$, was created using a quadratic polynomial of the form

$$\hat{f}^k(x_1, x_2) = c_0 + c_1 x_1 + c_2 x_2 + c_3 x_1 x_2 + c_4 x_1^2 + c_5 x_2^2. \qquad (18)$$

The coefficients, $c_0, ..., c_5$, were found using a least squares fit with six data samples taken inside each trust region. A six-level Latin hypercube method was used to generate the samples. The SAO strategy was terminated if either of the following conditions was satisfied

$$\left| \frac{(f^k - f^{k-1})}{f^{k-1}} \right| < 0.0001, \qquad (19)$$

$$\left| (f^k - f^{k-1}) \right| < 0.001. \qquad (20)$$

4.2 Optimization Results

Table 1 lists the optimization results for Rosenbrock's function using DOT's quasi-Newton method and the SAO strategy. The DOT method with analytic gradients finds the minimum of this function, while the DOT-finite difference method and the SAO strategy terminate prior to reaching the minimum. Figure 2 shows the convergence history for the DOT-finite difference method and the SAO strategy. Both methods make good progress toward the minimum, with the SAO strategy taking a few more iterations before terminating. It is clear from the results in Table 1 and Figure 2 that the SAO strategy performs more function evaluations than the DOT quasi-Newton method. This behavior is expected since the SAO strategy does not use any gradient information. That is, the consistency condition in Equation 4 is not enforced with the quadratic polynomial used for $\hat{f}^k(x_1, x_2)$. Even though the convergence is slow, the intent of this test problem is to demonstrate that the SAO strategy exhibits the correct convergence trends.

5 Quasi-sinusoidal Test Function

5.1 Background

Another demonstration of the SAO strategy is performed on the quasi-sinusoidal test function developed in Reference [13]

$$f(x_i) = \sum_{i=1}^{n_v} [\tfrac{3}{10} + \sin(\tfrac{16}{15}x_i - \tfrac{7}{10}) + \sin^2(\tfrac{16}{15}x_i - \tfrac{7}{10}) + \tfrac{1}{50}\sin(40(\tfrac{16}{15}x_i - \tfrac{7}{10}))]. \quad (21)$$

This sine test function is shown in Figure 3 for two parameters, x_1 and x_2. A contour plot of this surface is shown in Figure 4, and a cut through this surface at $x_2 = 0$ is shown in Figure 5. The global minimum for this function on the range $[-1, 1]^2$ is $(x_{1*}, x_{2*}) = (0.177, 0.177)$ where $f(x_{1*}, x_{2*}) = 0.060$.

The last term on the right side of Equation 21 creates a high frequency, low amplitude variation in the test function. This high frequency term simulates the nonsmooth behavior that occurs in many engineering analyses by creating many local minima in the test function (although real-world nonsmooth trends are seldom this well-behaved). In such situations there often is a clear global trend, but gradient-based optimizers are easily trapped in a local minimum close to their starting point. The SAO strategy is well-suited to this type of problem, since a surrogate model will smooth out the high-frequency variations.

5.2 Optimization Results

Table 2 lists the optimization results obtained with DOT's quasi-Newton method, with various finite difference step sizes ranging from 0.0001 to 0.5 (i.e., 0.01% to 50%). All optimization cases were started from the point $(-0.3, -0.3)$, where $f(x_1, x_2) = 0.346$. All of the optimization cases terminated at approximately the same point $(x_1, x_2) = (-0.253, -0.253)$, which is the nearest local minimum to the starting point.

Table 3 lists the optimization results for the SAO strategy, using various initial trust region sizes. Here the initial trust region size is expressed as a percentage of the global bounds. For example, an initial trust region size of 20 percent gives bounds of $[-0.7, 0.1]$ around $x_1 = -0.3$ and $x_2 = -0.3$. For small trust region sizes the SAO strategy performs like a traditional gradient-based optimizer and locates the nearest local minimum. As the initial trust region size increases, the SAO strategy generally moves further away from the starting point and finds another local minimum, including several points at or near the global minimum. Note that the one case that did not converge was terminated prematurely when the first two SAO iterations were rejected.

The results in Tables 2 and 3 demonstrate the utility of using SAO methods over traditional gradient-based methods in situations where multiple local minima exist. While not guaranteed to find a global minimum, the SAO strategy permits the optimizer to move outside of the immediate vicinity of the starting point, and avoids becoming trapped in the nearest local minimum.

6 Future Work

The initial SAO capability in DAKOTA described in this paper will be extended in several areas. The first area of improvement focuses on satisfying the consistency conditions in Equations 3 and 4. This will involve changes to the design of experiments sampling methods and to the surrogate modeling methods in DAKOTA. Once the consistency conditions can be satisfied, we will explore a convergence proof for the SAO strategy described in Equation 5. It is likely that this convergence proof will necessitate changes to the convergence criteria that currently are used, especially when considering constrained optimization problems where the Karush-Kuhn-Tucker conditions must be met.

7 Conclusions

A sequential approximate optimization strategy has been developed and implemented in the DAKOTA optimization software toolkit. The two test problems examined in this study demonstrate the utility of this SAO strategy.

In the Rosenbrock test problem the SAO strategy converges to a point near the global minimum, but does so using twice as many function evaluations as a quasi-Newton method. This slow convergences is expected since the SAO strategy does not use gradient information to guide it to convergence.

For the quasi-sinusoidal test problem the SAO strategy avoids the nearby local minima that trap the quasi-Newton method. As a result, the SAO strategy generally finds better local minima than the quasi-Newton method. As expected, the performance of the SAO strategy depends on the initial size of the trust region search space.

This SAO strategy fills a need for a capability between local gradient-based optimization and global

gradient-free optimization. While the SAO strategy developed in this paper lacks a proof of convergence, it is a useful technique for engineering optimization problems, many of which exhibit nonsmooth data trends. Future development of a convergence proof will promote confidence in the use of this SAO strategy.

References

[1] Gilmore, P. and Kelley, C. T., "An Implicit Filtering Algorithm for Optimization of Functions with Many Local Minima," *SIAM J. Opt.*, vol. 5, no. 2, pp. 269–285, 1995.

[2] Haftka, R. T. and Gürdal, Z., *Elements of Structural Optimization*, 3rd edn., Kluwer, Boston, MA, 1993.

[3] Toropov, V. V., Filatov, A. A., and Polynkin, A. A., "Multiparameter Structural Optimization Using FEM and Multipoint Explicit Approximations," *Structural Optimization*, vol. 6, pp. 7–14, 1993.

[4] Hutchison, M. G., Unger, E. R., Mason, W. H., and Grossman, B., "Variable-Complexity Aerodynamic Optimization of an HSCT Wing Using Structural Wing-Weight Equations," *J. Aircraft*, vol. 31, no. 1, pp. 110–116, 1994.

[5] Rodriguez, J. F., Renaud, J. E., and Watson, L. T., "Convergence of Trust Region Augmented Lagrangian Methods Using Variable Fidelity Approximation Data," *Structural Optimization*, vol. 15, pp. 1–7, 1998.

[6] Alexandrov, N. M., Dennis, Jr., J. E., Lewis, R. M., and Torczon, V., "A Trust-region Framework for Managing the Use of Approximation Models in Optimization," *Structural Optimization*, vol. 15, pp. 16–23, 1998.

[7] Alexandrov, N. M., Lewis, R. M., Gumbert, C. R., Green, L. L., and Newman, P. A., "Optimization with Variable-Fidelity Models Applied to Wing Design," AIAA Paper 2000–0841, 38th Aerospace Sciences Meeting and Exhibit, Reno, NV, 2000.

[8] Dennis, Jr., J. E., El-Alem, M., and Maciel, M. C., "A Global Convergence Theory for General Trust-Region-Based Algorithms for Equality Constrained Optimization," *SIAM J. Opt.*, vol. 7, no. 1, pp. 177–207, 1997.

[9] Eldred, M. S., Bohnhoff, W. J., and Hart, W. E., "DAKOTA, An Object-Oriented Framework for Design Optimization Parameter Estimation, Sensitivity Analysis, and Uncertainty Quantification," Sandia Technical Report SAND2000-XXXX, Draft available from authors., (in preparation).

[10] Myers, R. H. and Montgomery, D. C., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, John Wiley & Sons, Inc., New York, NY, pp. 1–141, 279–401, 462–480, 1995.

[11] Zimmerman, D., "Genetic Algorithms for Navigating Expensive and Complex Design Spaces," Final Research Report prepared for Sandia National Laboratories, (technical contact: M. Eldred), 1996.

[12] Friedman, J. H., "Multivariate Adaptive Regression Splines," Stanford Linear Accelerator Center, Tech Report PUB-4960, (also in The Annals of Statistics, 1990), 1990.

[13] Giunta, A. A. and Watson, L. T., "A Comparison of Approximation Modeling Techniques: Polynomial Versus Interpolating Models," in *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Design*, St. Louis, MO, pp. 392–404, AIAA Paper 98–4758, Sept. 1998.

[14] McKay, M. D., Beckman, R. J., and Conover, W. J., "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.

[15] Koehler, J. R. and Owen, A. B., *Computer Experiments*, vol. 13 of *Handbook of Statistics*, Elsevier Science, New York, NY, pp. 261–308, eds. S. Ghosh and C. R. Rao, 1996.

[16] Vanderplaats Research & Development, Inc., Colorado Springs, CO, *DOT User's Manual, Version 4.20*, 1995.

Table 1: Optimization results for Rosenbrock's function using DOT-BFGS and sequential approximate optimization. The starting point was $(x_1, x_2) = (0, 0)$.

Optimizer	Num. $\mathbf{f(x_1, x_2)}$ Evals.	Num. $\hat{\mathbf{f}}\mathbf{(x_1, x_2)}$ Evals.	Stopping Point $(\mathbf{x_{1*}, x_{2*}})$	Num. Iterations	$\mathbf{f(x_{1*}, x_{2*})}$
DOT-BFGS(analytic grad.)	93	*N/A*	$(0.999, 0.998)$	23	0.000
DOT-BFGS(finite diff. grad.)	116	*N/A*	$(0.923, 0.852)$	18	0.006
SAO	208	906	$(0.935, 0.873)$	25	0.004

Table 2: Optimization results for the quasi-sinusoidal test function using DOT's quasi-Newton method with BFGS update and finite difference gradients. The starting point was $(x_1, x_2) = (-0.3, -0.3)$.

Finite Diff. Step Size	Num. $\mathbf{f(x_1, x_2)}$ Evals.	Stopping Point $(\mathbf{x_{1*}, x_{2*}})$	Num. Iterations	$\mathbf{f(x_{1*}, x_{2*})}$
0.01%	24	$(-0.253, -0.253)$	4	0.275
0.1%	21	$(-0.253, -0.253)$	4	0.275
1.0%	21	$(-0.253, -0.253)$	4	0.275
10.0%	15	$(-0.255, -0.255)$	3	0.275
50.0%	25	$(-0.253, -0.253)$	4	0.275

Table 3: Optimization results for the quasi-sinusoidal test function using the SAO strategy with various initial trust region sizes. The starting point was $(x_1, x_2) = (-0.3, -0.3)$.

Initial Trust Region Size	Num. $\mathbf{f(x)}$ Evals.	Num. $\hat{\mathbf{f}}\mathbf{(x)}$ Evals.	Stopping Point $(\mathbf{x_{1*}, x_{2*}})$	Num. Iterations	$\mathbf{f(x_{1*}, x_{2*})}$
0.1%	80	152	$(-0.252, -0.254)$	10	0.276
1.0%	32	45	$(-0.251, -0.253)$	4	0.276
5%	72	148	$(-0.107, -0.253)$	9	0.221
10%	32	66	$(-0.107, -0.108)$	4	0.167
20%	72	143	$(0.177, 0.036)$	9	0.074
30%	48	113	$(0.316, -0.551)$	9	0.295
40%	80	195	$(0.177, 0.455)$	10	0.148
50%	64	134	$(0.036, -0.108)$	8	0.127
60%	72	158	$(0.036, -0.551)$	9	0.287
70%	16	27	$(-0.300, -0.300)^*$	2	0.346
80%	80	155	$(0.177, 0.036)$	10	0.074
90%	40	62	$(0.177, 0.177)$	5	0.060
100%	64	148	$(0.177, 0.177)$	8	0.060

(* did not converge)

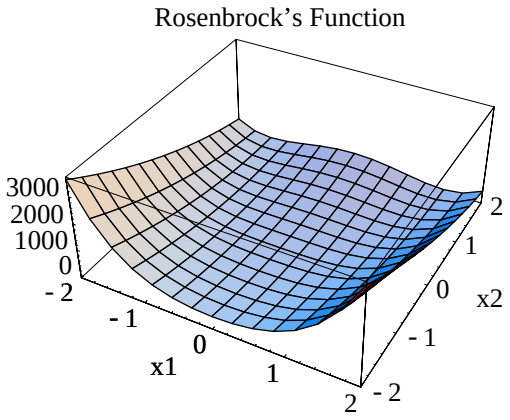

Figure 1: Rosenbrock's function plotted versus x_1 and x_2. The global minimum is at the point (1,1).

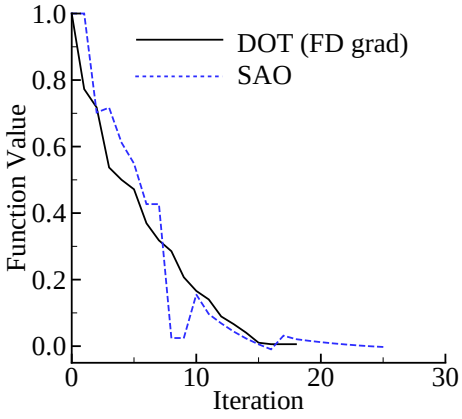

Figure 2: Convergence history for minimizing Rosenbrock's function using DOT's quasi-Newton method (with finite difference gradients) and sequential approximate optimization.

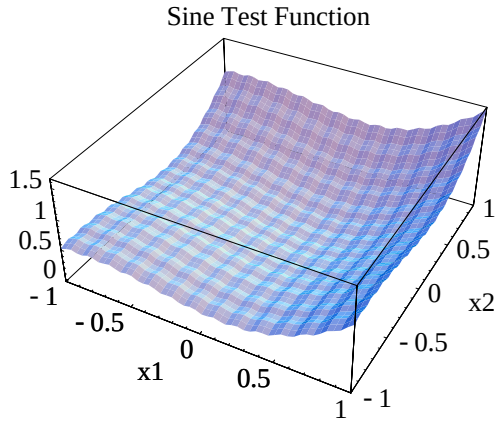

Figure 3: The quasi-sinusoidal test function plotted versus x_1 and x_2. The global minimum is at the point (0.177,0.177).

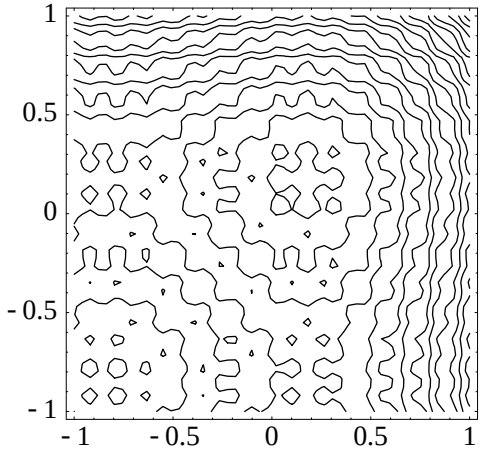

Figure 4: Contours of the quasi-sinusoidal test function, with the x_1 axis along the bottom of the figure.

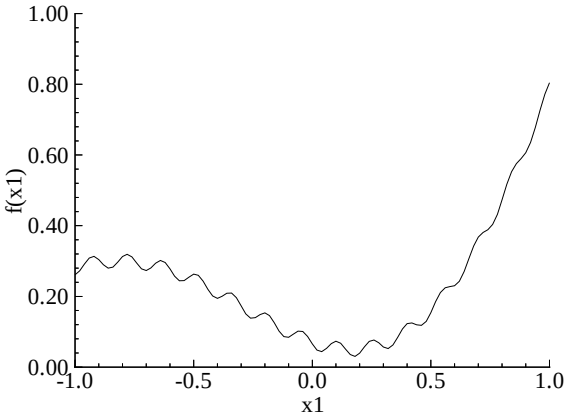

Figure 5: A cut through the quasi-sinusoidal test function with $x_2 = 0$.