

AIAA-2002-5573

## GLOBAL OPTIMIZATION OF PROBLEMS WITH DISCONNECTED FEASIBLE REGIONS VIA SURROGATE MODELING

Michael Sasena,\* Panos Papalambros† Pierre Goovaerts‡  
 University of Michigan, Ann Arbor, MI 48109-2125  
 {msasena, pyp, goovaert}@umich.edu

### ABSTRACT

Approximation methods have successfully solved a variety of global optimization problems. One of the major remaining challenges involves problems where the feasible design space consists of disconnected regions. Traditionally, researchers have attempted to solve such problems by combining global and local searching algorithms such as Simulated Annealing and SQP. While this approach is often successful, it is not efficient in terms of the number of function evaluations required. When the design problem is comprised of expensive functions such as computer simulations, new techniques must be applied. We use an approximation-based global optimization algorithm, superEGO, to solve such problems. We propose taking advantage of superEGO's flexibility to design alternative search strategies that locate multiple feasible regions in a new way. The technique was successfully applied to two analytical examples and required far fewer function calls than three competing techniques.

**Keywords:** Global Optimization, Disconnected Feasible Region, Approximations, Constrained Optimization

### 1 INTRODUCTION

Approximations have been used extensively in design optimization to address the problems inherent to the use of computer simulations<sup>1,2,12,15</sup>. Two of the main difficulties are the computational expense of the objective and/or constraint functions and the discontinuous or noisy behavior of the responses. A wide variety of algorithms exist for utilizing approximations to overcome these problems by providing a smooth function that is inexpensive to compute. The framework used here is a constrained, nonlinear

optimization algorithm known as *superEGO*<sup>12</sup>, an extension of the EGO algorithm<sup>15</sup>.

The algorithm works by taking an initial sample and fitting kriging approximations to the data<sup>4</sup>. It then uses an infill sampling criterion (ISC) as the objective function for an auxiliary optimization problem that selects the next design point to evaluate. The kriging models are iteratively updated with the new samples until the termination criterion is reached (e.g., the maximum number of samples is reached). The flowchart of the superEGO algorithm is shown in Figure 1.

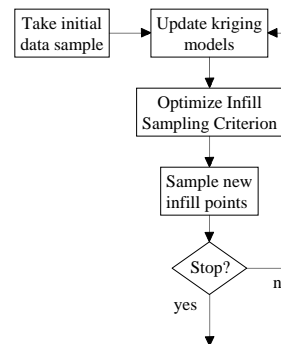


Figure 1: Flowchart of the superEGO algorithm

To demonstrate superEGO's search strategy for traditional optimization problems, a one-dimensional multimodal example is shown in Figure 2. The w-shaped dashed line is the true objective function we wish to minimize, while the solid line is the kriging approximation conditional to the sample points shown as circles. The plot at the bottom is the sampling criterion, normalized to facilitate comparisons between iterations.

Observing the progression of the algorithm, one can see that it searches for the optimum by looking for points of good local improvement as well as regions of high model uncertainty where there is a

\*PhD, Dept. of Mechanical Engineering  
 †Professor, Dept. of Mechanical Engineering  
 ‡Assistant Professor, Dept. of Civil and Environmental Engineering

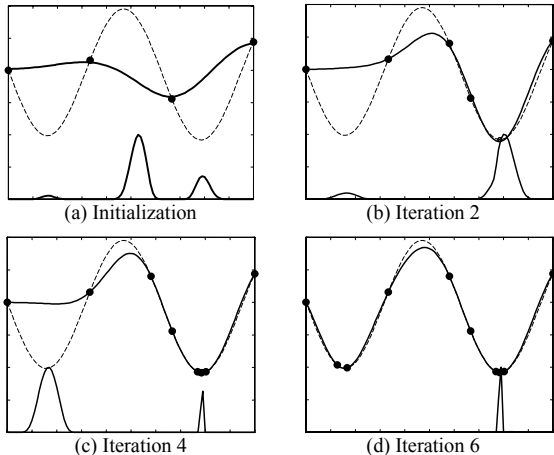


Figure 2: Demonstration of superEGO. Dashed line is the true function to be minimized, solid line is the approximation, circles are the sample points. The plot at the bottom is the sampling criterion.

possibility of an overlooked optimum. After the initial sample of 4 points is taken, the resulting kriging model is a poor fit compared to the true function. However, the ISC leads the algorithm to sample points where the uncertainty in the model is highest. After two iterations, the model has improved in the region of the local optimum on the right. Two iterations later, the region on the right has been explored somewhat, but the uncertainty in the model on the left portion of the model drives superEGO to sample points in that region. After six iterations, both local optima have been discovered and the true solution has been found quite accurately.

The search behavior is controlled by the choice of sampling criterion. In the above example, the expected improvement function<sup>15</sup> was used as the ISC in order to locate the minimum of the test function. SuperEGO can use any one of 16 predefined criteria, and the flexibility of the algorithm allows for any ISC to be incorporated quite easily. It is this flexibility that allows us to expand upon the current capabilities of superEGO by introducing new search strategies<sup>13</sup>.

While the framework has been applied to a variety of optimization problems, some interesting challenges remain. The specific difficulty addressed in this work is how to solve global optimization problems where the feasible region is relatively small and/or disconnected. Typically, a combination of global and local techniques is applied in an attempt to sweep through the space, searching for feasible regions and then finding their associated local minima. For example, one may use a multistart SQP or clustering algorithm, possibly combined heuristi-

cally with a global search such as Simulated Annealing. While this is often effective, it generally requires a large number of function evaluations. Here we explore the approximation-based framework described above to solve such problems using fewer function evaluations.

## 2 METHODOLOGY

New challenges in approximation-based optimization arise when constraints are introduced. Several methods have been proposed for incorporating inequality constraints into the algorithm<sup>12</sup>. The method preferred by this author is to include the constraints directly in the ISC subproblem formulation. It has been shown to produce accurate results and can be applied regardless of the choice of sampling criterion. The resulting ISC subproblem is of the form:

$$\begin{aligned} \min \quad & ISC(\mathbf{x}) \\ \text{subject to: } \quad & \hat{g}_{exp}(\mathbf{x}) \leq 0 \\ & g_{inexp}(\mathbf{x}) \leq 0, \end{aligned} \quad (1)$$

where  $ISC(\mathbf{x})$  is the infill sampling criterion,  $\hat{g}_{exp}(\mathbf{x})$  is the predicted value of the kriging model for each expensive constraint, and  $g_{inexp}(\mathbf{x})$  is the actual value for each inexpensive constraint. We distinguish between functions that are expensive to compute and must be modeled with kriging approximations, and functions that are inexpensive enough to evaluate directly during the ISC subproblem. By doing so, one can easily incorporate inexpensive constraint information into the search process, making for a more efficient approach<sup>14</sup>.

Simply optimizing the expected improvement function subject to the constraints as shown above does not guarantee that solutions to the original design problem can be efficiently generated if the feasible space is difficult to locate or model. In order to search the design space better, we propose two criteria: one for locating an initial feasible point, and one for locating subsequent feasible points in unexplored regions.

### 2.1 Locating an initial feasible point

When no feasible point has yet been found, one can simply search for the location in the design space that yields the highest probability of feasibility. One way to quantify this is to assume the constraints are independent and multiply the probability that each one is satisfied:

$$\max \quad ISC_1(\mathbf{x}) = \prod_{i=1}^m P(g_i(\mathbf{x}) \leq 0), \quad (2)$$

Because the inexpensive functions are evaluated directly, the probability of feasibility is simply:

$$P_{inexp} = \begin{cases} 1, & \text{if } g_{inexp}(\mathbf{x}) \leq 0 \\ 0, & \text{otherwise} \end{cases} . \quad (3)$$

For expensive constraints, the kriging model is used to estimate the probability of feasibility as:

$$P_{exp} = \Phi \left( \frac{0 - \hat{g}_{exp}(\mathbf{x})}{\hat{\sigma}(\mathbf{x})} \right) , \quad (4)$$

where  $\Phi$  is the Gaussian cumulative distribution function,  $\hat{g}_{exp}(\mathbf{x})$  is the kriging estimate of the constraint function and  $\hat{\sigma}^2(\mathbf{x})$  is its associated kriging variance. While a constraint boundary of 0 is shown in the equations above, the user is free to set any constraint tolerance they wish. Also, the examples shown utilize kriging models; but any approximating function may be used, provided it has an associated measure for the local uncertainty in the model (i.e., the variance).

To illustrate the behavior of the criterion, a two-dimensional example with one constraint function is introduced. The objective is a simple quadratic function, and the Branin test function<sup>2</sup> is applied as a constraint. The example problem, referred to here as the *newBranin* example, is of the form:

$$\begin{aligned} \min f(\mathbf{x}) &= -(x_1 - 10)^2 - (x_2 - 15)^2 & (5) \\ \text{s.to: } g(\mathbf{x}) &= \left( x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + \dots \\ &10 \left( 1 - \frac{1}{8\pi} \right) \cos x_1 + 10 \leq 5 . \end{aligned}$$

A contour plot of the newBranin example is shown in Figure 3. The problem consists of three disconnected feasible regions (shown as dashed lines).

A plot of the  $ISC_1$  criterion using an initial 21 point sample (none of which are feasible) is shown in Figure 4. Because there is only one constraint function,  $ISC_1$  is the probability that the constraint in Equation (5) will be satisfied. As expected, there are three regions of high probability of feasibility. The remainder of the design space is very unlikely to be feasible and therefore has a value for  $ISC_1$  close to zero.

Optimizing the  $ISC_1$  criterion searches for points where the probability of feasibility is highest. Of course, this will only work for cases where feasible points have yet to be found. Otherwise, there will be a region of probability 1, and the algorithm will simply continue sampling in that region. Therefore, a second strategy must be employed to find subsequent feasible points.

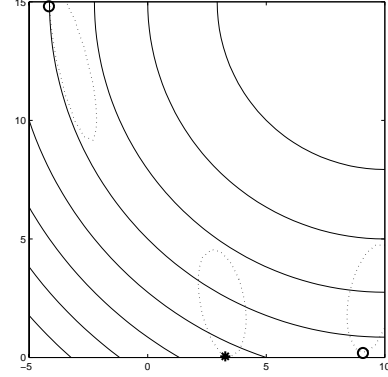


Figure 3: Contour plot of the newBranin example. Feasible region boundaries are shown as dashed lines, local optima as circles, and the global optimum as an asterisk.

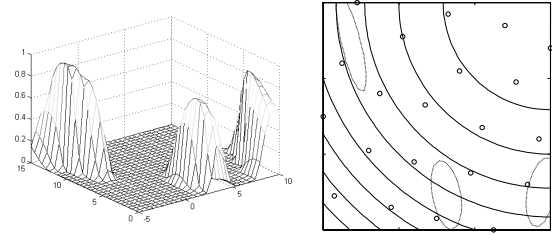


Figure 4: Mesh and contour plot of  $ISC_1$  for the newBranin example. Sample points are shown as circles.

## 2.2 Locating subsequent feasible points

Once an initial feasible point has been found, the challenge is to force the algorithm to search elsewhere for other feasible regions. We propose using an approach similar to *tunnelling*<sup>3</sup> whereby the ISC subproblem maximizes the distance from the current best point while searching for areas that have a high probability of feasibility. The two terms in the criterion are the probability (P) and the distance to the nearest feasible point (D).

$$\max ISC_2(\mathbf{x}) = \prod_{i=1}^m P_i \cdot D_i , \quad (6)$$

where P is calculated by Equations (3) and (4) and the distance, D, is calculated as

$$D = \min_{x_{feas}} \left( \frac{\|\mathbf{x}_{feas} - \mathbf{x}\|}{range} \right) . \quad (7)$$

Dividing the distance by the range is done to balance the magnitudes of the distance in each dimension. Failure to do so might lead to the distance portion of  $ISC_2$  being dominated by one design variable. Multiplying the distance by the probability has the

effect of reducing the value of the criterion wherever there is a low probability of feasibility.

To illustrate the behavior of  $ISC_2$ , we continue the same newBranin example. After the initial 21 point design,  $ISC_1$  guides superEGO to sample a point inside the feasible region in the upper left portion of the design space.  $ISC_2$  is then used to distinguish between these regions of likely feasibility by their distance from the one feasible sample point. A plot of the sampling criterion is shown in Figure 5. The initial sample points are shown as circles, and the only feasible point is shown as an x.

Observe how the surface takes a noticeable dip in the immediate vicinity of the known feasible point. The algorithm is guided to the center of the feasible region farthest away. Once that location is sampled, only the middle feasible region is left unexplored. Figure 6 shows  $ISC_2$  after two feasible points have been found. Note that the criterion has small values in the immediate neighborhood of those points, but high values in the middle region where the constraint is likely to be satisfied and there are no nearby feasible points. Thus utilizing  $ISC_1$  followed by  $ISC_2$  successfully locates all three feasible regions in just three iterations.

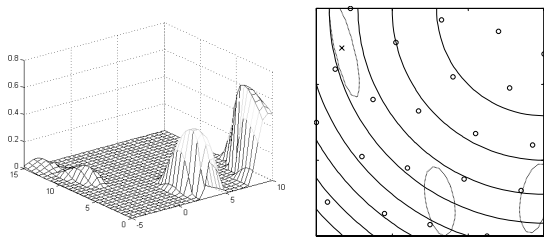


Figure 5: Mesh and contour plot of  $ISC_2$  for the newBranin example. Initial sample points are shown as circles, and the only feasible sample is shown as an x.

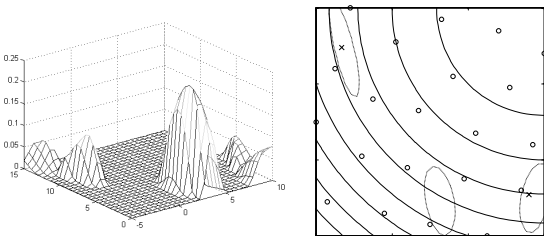


Figure 6: Mesh and contour plot of  $ISC_2$  for the newBranin example after an additional iteration. Initial sample points are shown as circles, and the two feasible samples are shown as x's.

### 2.3 Implementation

Equipped with the necessary tools to locate feasible points throughout the design space, one must decide how to implement the search strategy logic. Ideally, one would locate an initial feasible point, find the local optimum in its vicinity, then move on to locate new feasible regions and their associated local optima. The only missing tool then is the local search criterion. For this work, we attempt to find local optima by finding the minimum of the approximate model as:

$$\begin{aligned} \min \quad & ISC_3(\mathbf{x}) = \hat{y}(\mathbf{x}) \\ \text{subject to:} \quad & \hat{g}_{exp}(\mathbf{x}) \leq 0 \\ & g_{inexp}(\mathbf{x}) \leq 0, \end{aligned} \quad (8)$$

where  $\hat{y}(\mathbf{x})$  and  $\hat{g}_{exp}(\mathbf{x})$  are the kriging models of the objective and expensive constraint functions, respectively, and  $g_{inexp}(\mathbf{x})$  are inexpensive constraint functions.

Putting these criteria together, we have the following search strategy which will be referred to as *Search Strategy 1*:

1. Take an initial sample of the true functions, and model the expensive functions with kriging approximations.
2. If no feasible point has yet been sampled, solve an unconstrained ISC subproblem using  $ISC_1$ , otherwise go to 3.
3. Locate local optimum of new feasible region using  $ISC_3$  as a constrained ISC subproblem. If at least  $ncluster$  points have already been sampled within  $tol_1$  of the last iterate, then go to 4.
4. Attempt to find an additional feasible point using  $ISC_2$  as an unconstrained ISC subproblem. Once a feasible point is sampled at least  $tol_2$  away from other feasible samples, begin another local search (go to 3). Terminate once  $maxfCount$  function calls have been made.

In our current implementation, we have tested the parameter values  $ncluster=3$ ,  $tol_1=0.5\%$  and  $tol_2=5\%$  where the latter two are percentages of the maximum Euclidean distance of the design space (i.e., corner to corner). The user is free to specify the  $maxfCount$  as they see fit, but a value of  $50 * d$  where  $d$  is the number of design variables tends to produce adequate results.

A shortcoming of Search Strategy 1 is that it does not consider information about the objective function when searching for new feasible regions in Step

4. This may cause the algorithm to search in regions of poor objective function values if these are locations where there is a high probability of finding a new feasible region. To address this issue, *Search Strategy 2* introduces a constraint into the ISC subproblem of Step 4 whereby the probability of the objective being at least as good as the current best value  $f_{min}$  must exceed some threshold  $P_{tol}$ :

$$\begin{aligned} \max \quad & ISC_4(\mathbf{x}) = \prod_{i=1}^m P_i \cdot D_i \quad (9) \\ \text{subject to:} \quad & P(Y(\mathbf{x}) \leq f_{min}) \geq P_{tol} , \end{aligned}$$

where  $Y(\mathbf{x})$  is the true objective at the candidate point and  $P_{tol} = 90\%$  by default. Incorporating this constraint restricts superEGO to search for new feasible regions only where it is likely to improve upon the best point. Search Strategy 2 is identical to Search Strategy 1 except that  $ISC_2$  in Step 4 is replaced by  $ISC_4$ .

In all cases, the global searching, gradient-free DIRECT algorithm of Jones<sup>5,7</sup> is used to solve the ISC subproblem (either constrained or unconstrained). Because DIRECT is an interval method (i.e., it finds solutions by sequentially subdividing the design space), it is possible that it can choose the same exact design point in different iterations. Sampling identical locations would cause numerical instabilities for the kriging approximations, therefore some slight adjustments were made to the search strategies to accommodate such situations. While this is not a frequent occurrence, it has been observed in several instances as design points begin to cluster around the best known solution.

If DIRECT cannot locate a unique candidate point within the function call limit, superEGO switches sampling criteria for one iteration, and searches for the location of maximum uncertainty in the kriging models. Doing so helps in two ways. It ensures that the algorithm can continue searching by providing a candidate that has not yet been sampled, and it improves the model accuracy by sampling the region of greatest uncertainty.

Another difficulty that arises during the solution of the ISC subproblem is the possibility that no feasible solution exists due to the inaccuracy in the models of the constraint functions. When DIRECT is unable to locate a feasible point, superEGO performs one iteration using the maximum uncertainty criterion instead. At each iteration, superEGO verifies the acceptability of the next candidate and reacts accordingly as often as needed until DIRECT is able to find a unique, feasible solution to the original ISC subproblem.

### 3 EXAMPLES

In this section, we evaluate how well the proposed methodology locates the optimum of a problem with disconnected feasible regions. We compare it against three other algorithms: Sequential Quadratic Programming (SQP)<sup>11</sup>, Simulated Annealing (SA)<sup>8</sup>, and DIvided RECTangles (DIRECT)<sup>5,7</sup>. SQP is among the most widely used optimization algorithms. It is a deterministic, gradient-based algorithm that finds local solutions efficiently. SA is a stochastic, derivative-free algorithm that requires many more function evaluations to find a good solution, but is less likely to get trapped in local optima than SQP. DIRECT is a deterministic, derivative-free algorithm that also performs global searching, but tends to be more efficient than global algorithms such as SA or genetic algorithms.

#### 3.1 Problem Description

To measure how well each algorithm locates the solution, we counted the number of function evaluations required before a feasible point was sampled within a box the size of  $\pm 1\%$  of the design space range centered around the true solution. Each algorithm must be handled slightly differently in order to create a meaningful comparison, which we refer to as the  $x_{1\%}$  metric.

The superEGO algorithm is strongly impacted by the selection of the initial sample, therefore ten different ten-point Latin hypercube design of experiments (DOE) were applied. The  $x_{1\%}$  metric was computed after each run and the average value was recorded. The results of the DIRECT algorithm were straightforward to compare because it is deterministic and does not require a starting point.

Both the SQP and SA algorithms are strongly influenced by the starting point. Additionally, the SA algorithm is stochastic (i.e., it will result in a different search path each time because it uses random numbers). Each algorithm was started from 100 locations in a 10 by 10 grid around the design space. Not every run converged to a point within the  $x_{1\%}$  boundaries, so an effective  $x_{1\%}$  metric was computed as the average of the successful runs divided by the success rate. For example, if SQP required on average 100 function calls to first evaluate a point within the  $x_{1\%}$  boundaries, but half of the runs never successfully found the solution, then the effective  $x_{1\%}$  metric value would be  $100/50\%=200$  function calls.

The first example is the newBranin example used in the previous section. However, the constraint was set to  $g(\mathbf{x}) \leq 2$  rather than  $g(\mathbf{x}) \leq 5$  as was used in Equation (5), in order to make the feasible regions somewhat smaller for this example (see Figure 7).

The three feasible regions cover approximately 3% of the design space in total, making it difficult to locate the true optimum at  $\mathbf{x}_*=[3.2143 \ 0.9633]$  at the center of the square in Figure 7.

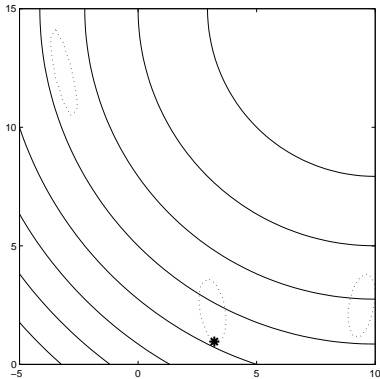


Figure 7: Contour plot of the newBranin example. Feasible region boundaries are shown as dashed lines and the global optimum as an asterisk.

The second example is the Gomez #3 test function<sup>3</sup>, shown in Figure 8. The problem statement is

$$\begin{aligned} \min f(\mathbf{x}) &= (4 - 2.1x_1^2 + x_1^4/3)x_1^2 + \dots \quad (10) \\ &+ x_1x_2 + (-4 + 4x_2^2)x_2^2 \\ \text{s. to: } g(\mathbf{x}) &: -\sin(4\pi x_1) + 2\sin^2(2\pi x_2) \leq 0 \end{aligned}$$

defined over the range  $-1 \leq x_i \leq 1, i = 1, 2$ . The example is quite difficult to solve because the sine terms in the constraint function create 20 disconnected feasible regions that cover approximately 19% of the design space in total. The true solution lies at  $\mathbf{x}_*=[0.1093 \ -0.6234]$ . Gradient-based methods are not able to solve such problems efficiently because they require a large number of multistarts before the global solution can be found with any confidence. Approximation-based methods also have difficulty with this problem because the constraint function is quite challenging to model accurately.

### 3.2 Results

Table 1 shows the  $x_{1\%}$  performance metrics for both examples. SQP converged to a point within the  $x_{1\%}$  boundaries for only 15 of the 100 runs in the newBranin example and 4 of 100 runs in the Gomez example. The SA algorithm was successful for only 19 and 20 of the 100 runs in the newBranin and Gomez examples, respectively. This supports the claim that local search methods are prone to failure for problems with disconnected feasible regions, and that SA is better at avoiding local optima than SQP.

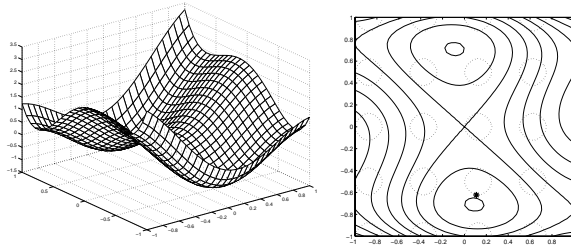


Figure 8: Plot of the Gomez #3 test function. The solid and dashed contours are the objective and constraint boundaries, respectively. The feasible space is inside each of the dashed circles, and the optimum is shown as an asterisk.

Table 1: Comparison of  $x_{1\%}$  performance metrics

Algorithm	Avg # of function calls	
	newBranin	Gomez
superEGO 1	22.2	66.3
superEGO 2	22.0	36.5
DIRECT	76	93
SQP	363	831
SA	5371	7150

The results from these two examples demonstrate that the proposed search strategies enable approximation-based algorithms to solve problems with disconnected feasible regions successfully. Both of the superEGO search strategies were able to locate the solution for each of the ten initial samples in both examples (40 cases in all). More importantly, the search was very efficient. Even when compared to DIRECT, a global searching algorithm that is relatively efficient, superEGO was able to find the solution in fewer function evaluations. As expected, SQP was less efficient due to the low percentage of successful runs. Finally, SA was the least efficient, requiring two orders of magnitude more function calls than superEGO.

Another observation from these examples is that imposing the improvement constraint in Equation (9) for Search Strategy 2 appeared to bolster the efficiency of superEGO. Search Strategy 2 required the same or fewer function evaluation in 8 of 10 runs for identical initial DOE's in the newBranin example – 10 of 10 in the Gomez example. Comparing each individual run, Search Strategy 2 required on average only 0.8% fewer function calls for the newBranin example, but 40.5% fewer function calls for the Gomez example. The improvement was much more pronounced in the Gomez example because

there were spurious feasible regions. By preventing superEGO from searching areas of likely high objective function values, far fewer function calls were wasted exploring suboptimal regions.

A consequence of the additional constraint was that Search Strategy 2 often failed to find a unique candidate sample point within the allotted 1000 function evaluations during the ISC subproblem for  $ISC_2$ . The improvement constraint limited the area of the design space superEGO could search, and much of the remaining area had a sampling criterion value near zero because it was either close to an existing feasible point or likely to be infeasible. As a result, Search Strategy 2 was forced to rely on the substitution of the maximum variance criterion more frequently than Search Strategy 1. This was not significantly detrimental as it aided in modeling the constraint function, leading to more accurate searches in subsequent iterations. On two of the 20 tests, the additional variance criterion iterations resulted in Search Strategy 2 requiring more function evaluations than Search Strategy 1 to find a solution within the  $x_{1\%}$  boundaries. However, the difference was very slight – one evaluation in one case, and 5 evaluations in the other. Moreover, Search Strategy 2 proved to be a more effective method for solving problems with disconnected feasible regions.

#### 4 CONCLUSIONS AND FUTURE WORK

While existing techniques can successfully solve problems with disconnected feasible regions, they often require a large number of function evaluations. To address this shortcoming, a global searching, approximation-based algorithm was employed. Utilizing approximations of the constraint functions, a new search strategy was introduced to locate an initial feasible design point, refine the solution, then search for new feasible regions, and repeat. The resulting contribution is an efficient approach to solving problems where the feasible region is relatively small or disconnected.

Of course, the benefits do not come without a price. The drawback of the method is that each iteration of the algorithm requires solving another optimization problem to search for the next iterate. Therefore, the proposed methodology should only be applied if the objective and/or constraint functions are expensive to compute, say, requiring more than a minute of CPU time.

As this was an initial investigation of the search strategy, there is still much room for improvement. Most important would be an exploration of new implementation strategies. More work needs to be done to assess the impact of the parameters

$ncluster$ ,  $tol_1$ ,  $tol_2$  and  $P_{tol}$ . Entirely new implementations should also be explored. The heuristic approach for switching between  $ISC_1$ ,  $ISC_2$  (or  $ISC_4$ ), and  $ISC_3$  was successful here; however, there may be more rigorous or efficient ways to proceed. Statistical hypothesis testing may be an avenue to bring more formality to the decisions of when to alternate between looking for new feasible points and refining existing solutions. Results to date are promising, but further simulation-based studies must be performed to understand the remaining challenges.

#### ACKNOWLEDGMENTS

This research has been partially supported by the Automotive Research Center at the University of Michigan, a US Army Center of Excellence in Modeling and Simulation of Ground Vehicles, by the US Army Dual Use Science and Technology Program, and by the General Motors Collaborative Research Laboratory at the University of Michigan. This support is gratefully acknowledged.

#### REFERENCES

- [1] Cox D D and John S. SDO: A Statistical Method for Global Optimization. In MN Alexandrov and MY Hussaini (Eds.), *Multidisciplinary Design Optimization: State of the Art*. pp 315–329. SIAM, Philadelphia, PA, 1997.
- [2] Dixon, L C W and Szegő, G P. The Global Optimisation Problem: An Introduction, in *Towards Global Optimisation 2*, North-Holland Publishing Company, New York, 1978.
- [3] Gomez S and Levy A. The tunnelling method for solving the constrained global optimization problem with several non-connected feasible regions, Dold, A. and Eckmann, B. (Eds.) *Lecture Notes in Mathematics 909*, pp. 34–47, Springer-Verlag, 1982.
- [4] Goovaerts P. *Geostatistics for Natural Resources Evaluation*. New York: Oxford University Press, 1997.
- [5] Jones, D R. The DIRECT Global Optimization Algorithm, *Encyclopedia of Optimization*, 1:431–440, 2001.
- [6] Jones D R, Schonlau M, and Welch W J. Efficient global optimization of expensive black-box functions. *J of Global Optimization*, 13(4): 455–492, 1998.

- [7] Jones D R, Perttunen C D and Stuckman B E. Lipschitzian Optimization Without the Lipschitz Constant. *J of Optimization Theory and Application*, 79: 157–181, 1993.
- [8] Kirkpatrick S, Gelatt C D Jr., Vecchi M P. Optimization by Simulated Annealing, *Science*, 220, 4598, pp. 671–680, 1983.
- [9] Kodiyalam S and Sobieszczycki-Sobieski J. Alternate Sampling Methods for use with Multidisciplinary Design Optimization in a High Performance Computing Environment, *Proceedings ASME DETC/IEC Conference*, Paper No. DETC2001/DAC-21081, 2001.
- [10] Lin Y, Krishnapur K, Allen J K, and Mistree F. Robust Concept Exploration in Engineering Design: Metamodeling Techniques and Goal Formulations, *Proceedings ASME DETC/IEC Conference*, Paper No. DETC2000/DAC-14283, 2000.
- [11] Papalambros P Y and Wilde D J. *Principles of Optimal Design: Modeling and Computation*, Second Edition, Cambridge University Press, New York, 2000.
- [12] Sasena M J, *Flexibility and Efficiency Enhancements for Constrained Global Design Optimization with Kriging Approximations*, PhD Thesis, University of Michigan, Ann Arbor, MI, Department of Mechanical Engineering, 2002.
- [13] Sasena M J, Papalambros P Y and Goovaerts P. Exploration of Metamodeling Sampling Criteria for Constrained Global Optimization, *Engineering Optimization*, 34(3):263–278, 2002.
- [14] Sasena M J, Papalambros P Y and Goovaerts P. The Use of Surrogate Modeling Algorithms to Exploit Disparities in Function Computation Time Within Simulation-Based Optimization, *The Fourth World Congress of Structural and Multidisciplinary Optimization*, 2001.
- [15] Schonlau M. *Computer Experiments and Global Optimization*. Doctoral thesis, University of Waterloo, Department of Statistics, Ontario, Canada, 1997.