Journal of Global Optimization **21:** 415–433, 2001.

A Comparison of Global Optimization Methods for the Design of a High-speed Civil Transport

STEVEN E. COX[1], RAPHAEL T. HAFTKA[1], CHUCK A. BAKER[2,*], BERNARD GROSSMAN[2], WILLIAM H. MASON[2] and LAYNE T. WATSON[3]
[1]*Department of Aerospace Engineering, Mechanics & Engineering Science, P.O. Box 116250, University of Florida, Gainesville, FL 32611-6250, USA;* [2]*Department of Aerospace and Ocean Engineering, Virginia Polytechnic Institute and State University, 215 Randolph Hall, Mail Stop 0203, Blacksburg, VA 24061, USA;* [3]*Departments of Computer Science & Mathmatics, Virginia Polytechnic Institute and State University, 630 McBryde Hall, Mail Stop 0106, Blacksburg, VA 24061, USA; *Current address: Engineous Software, Inc., 1800 Perimeter Park West, Suite 275, Morrisville, NC 27560, USA*

Abstract. The conceptual design of aircraft often entails a large number of nonlinear constraints that result in a nonconvex feasible design space and multiple local optima. The design of the high-speed civil transport (HSCT) is used as an example of a highly complex conceptual design with 26 design variables and 68 constraints. This paper compares three global optimization techniques on the HSCT problem and two test problems containing thousands of local optima and noise: multistart local optimizations using either sequential quadratic programming (SQP) as implemented in the *d*esign *o*ptimization *t*ools (DOT) program or Snyman's dynamic search method, and a modified form of Jones' DIRECT global optimization algorithm. SQP is a local optimizer, while Snyman's algorithm is capable of moving through shallow local minima. The modified DIRECT algorithm is a global search method based on Lipschitzian optimization that locates small promising regions of design space and then uses a local optimizer to converge to the optimum. DOT and the dynamic search algorithms proved to be superior for finding a single optimum masked by noise of trigonometric form. The modified DIRECT algorithm was found to be better for locating the global optimum of functions with many widely separated true local optima.

1. Introduction

The conceptual design of complex systems often involves optimization with a large number of design variables involving multiple disciplines. As the number of variables increases, the volume of the design space increases exponentially and gradient calculations become more expensive. This increases the cost to optimize the design over even a small range for each variable. Often the feasible design space or the objective functions are nonconvex and may contain multiple local optima that can trap local optimizers and prevent them from locating the best design. To solve this problem, either multiple starting points or a global optimization algorithm may be used. Both of these methods will increase the cost of the optimization.

Discretization errors, round-off errors, and less than fully converged iterative calculations within analysis codes can result in noisy constraints and objective functions. This can create additional spurious optima. A distinction is made between these noise-generated, pseudo local optima and genuine multiple optima due to nonconvexity. Designers are primarily concerned with locating the physically meaningful local optima modeled by the analysis functions and need a way to bypass the numerical noise. In our research we employed approximations, dynamic search techniques, and space partitioning methods to try to overcome numerical noise.

Much research has been done to find computationally efficient methods to perform global optimization for high dimensional, nonconvex design spaces. Many global optimization codes have been developed and tested for use with different classes of problems, Floudas & Pardalos (1996). However, most of these global optimization algorithms are specialized to a narrow class of problems. Similarly, different local optimizers have been compared in terms of their performance as multistart optimizers (e.g., Haim et al., 1999). One general purpose global optimization algorithm to draw attention lately is a Lipschitzian optimizer called DIRECT, Jones et al. (1993). It has recently been applied to airfoil design, Cramer (1998), and modifications have been proposed to speed up the convergence, Nelson and Papalambros (1998).

The configurational design of the high-speed civil transport (HSCT) is an example of a high-dimensional design problem with a nonconvex feasible design space due to nonlinear constraints and numerical noise (Knill et al., 1999). This paper compares three global optimization methods for optimizing the design of the HSCT. The first method uses multiple starting points to optimize the design using sequential quadratic programming (SQP) as implemented in *d*esign *o*ptimization *t*ools (DOT), Vanderplaats (1995). The second method, Snyman's dynamic search algorithm (Snyman, 1983, 2000), is capable of passing through shallow local minima to locate a better optimum but still requires multiple starting points. The third method is the modified DIRECT, which is run only once.

In addition to the HSCT problem, two test functions were examined to explore the performance of the optimizers on different classes of problems. The first is a quartic polynomial with a large number of widely separated local optima. The second is the so-called Griewank function which has a single global optimum masked by trigonometric 'noise'. The comparisons help to show the strengths of each optimizer and suggest the types of problems each is best suited for.

2. Optimizers

2.1. OPTIMIZATION PROBLEM

Three optimizers are compared based on their ability to solve the constrained minimization problem,

$$\begin{array}{lll} \text{minimize} & f(\mathbf{x}), & \mathbf{x} = \{x_1, x_2, \ldots, x_n\}^{\mathbf{T}} \\ \text{such that} & g_j(\mathbf{x}) \leqslant 0, & j = 1, 2, \ldots, m \\ & x_{li} \leqslant x_i \leqslant x_{ui}, & i = 1, 2, \ldots, n \end{array}$$

where $f(\mathbf{x})$ and $g_j(\mathbf{x})$ may be nonconvex, nonsmooth functions.

2.2. SEQUENTIAL QUADRATIC PROGRAMMING

The commercial program DOT was used to provide multi-start local optimization using SQP. SQP forms a quadratic approximation of the objective function over several iterations and linear approximations for the constraints. For the initial iteration, it uses the function value and first derivatives of the objective function and constraints to create the approximations and substitutes the identity matrix for the Hessian to form a quadratic approximation of the objective function. It then moves towards the optimum within given move limits and samples the objective function and constraints at the stopping point to refine the quadratic approximation and create new approximations for the constraints. DOT repeats this process until it reaches a local optimum. Due to the use of approximations, DOT is relatively quick to perform a single optimization and will handle a limited amount of noise without becoming stuck in spurious local minima. DOT has been successfully used in the past with the HSCT problem, and compared favorably to other local optimizers (Haim et al., 1999). In order to perform a global search of the design space, DOT was started at 100 random initial designs for the HSCT problem and up to 20 000 starting points for the test problems.

2.3. DYNAMIC SEARCH

The second optimizer tested was Snyman's dynamic search method, Leap Frog Optimization Procedure with Constraints, Version 3 (LFOPCV3) (Snyman, 1983, 2000). LFOPCV3 is a semiglobal optimization method that is capable of moving through shallow local minima. The method is based on the physical analogy of a particle rolling down hill. As the particle moves down, it builds momentum, which carries it out of small dips in its path. At each step, LFOPCV3 subtracts a certain fraction of the gradient of the objective function from the velocity of the particle at the previous step. The new velocity determines the step size and direction for the current step. When the velocity vector makes an angle of more than 90° with the gradient vector, i.e., when it is descending, the velocity increases and the particle builds up momentum to go through areas where it is ascending.

When ascent occurs, a damping strategy is used to extract energy from the particle to prevent endless oscillation about a minimum. The dampening strategy used in the version of LFOPCV3 used here has been modified to increase its ability to search further away from a local optimum. Whenever the optimizer is at a point where the function value is higher than the best function value located so far, it creates a shallow quadratic function centered at the best location found and uses the slope of the quadratic function plus a small percentage of the objective function slope to calculate the next step. This reduces the dampening when it is moving uphill while preventing it from becoming trapped in a local optimum that is worse than the best point found so far. For the two test functions, analytical expressions for the gradient of the objective functions were used but the cost of LFOPCV3 was computed as if first-order finite difference methods were used. For the HSCT problem, first-order finite differences were used with step sizes equal to 6% of the value of each design variable.

LFOPCV3 handles constraints with a standard quadratic penalty function approach (Snyman, 2000). This causes the gradient of the penalty function to increase as the constraint violation increases which gives a smooth gradient at the constraint boundaries. The penalty parameter was set at 100 for the bulk of the optimization and then switched to 10 000 for the final convergence as instructed in the code to maximize the accuracy. In our initial experiments with this algorithm it demonstrated the ability to move further than DOT when searching for the global minimum and has been successfully used on functions with hundreds of local minima. However, it still requires multiple starts to sample the entire design box. For this comparison, we used the same 100 initial designs to compare the performance with DOT for the HSCT problem and up to 20 000 starting points for the test problems.

2.4. DIRECT

The DIRECT algorithm (Jones et al., 1993; Gablonsky, 1998; Carter et al., 2000) is a variation of Lipschitzian optimization that uses all values for the Lipschitz constant. Lipschitzian optimization requires the user to specify the Lipschitz constant, K, which is used as a prediction of the maximum possible slope of the objective function. Lipschitzian optimization uses the value of the objective function at the vertices of each hypercube and K to find the hypercube with potentially the lowest objective function value. By starting at each vertex and drawing a surface that slopes away from the vertex with a slope of K, the minimum possible function value at each point in the design space can be determined. The minimum possible value is the value of the highest surface at each point. The objective function is evaluated at the predicted minimum possible value and the process is repeated for a set number of iterations (see Fig. 1).

DIRECT does not require the user to predict the Lipschitz constant. It uses the function value at the center of each hypercube and the hypercube size to find

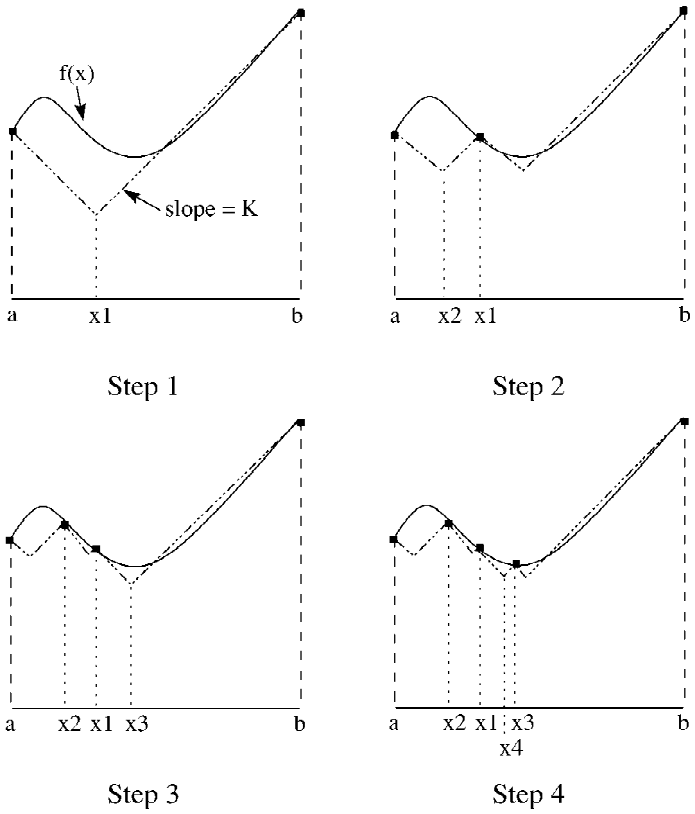

Figure 1. Lipschitzian optimization

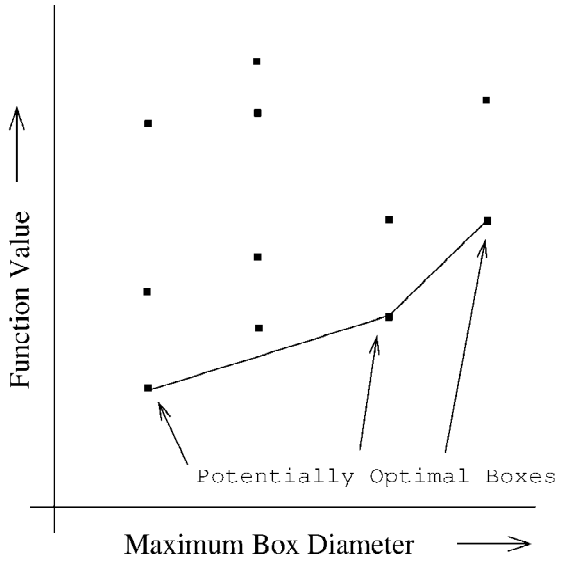

Figure 2. DIRECT point selection.

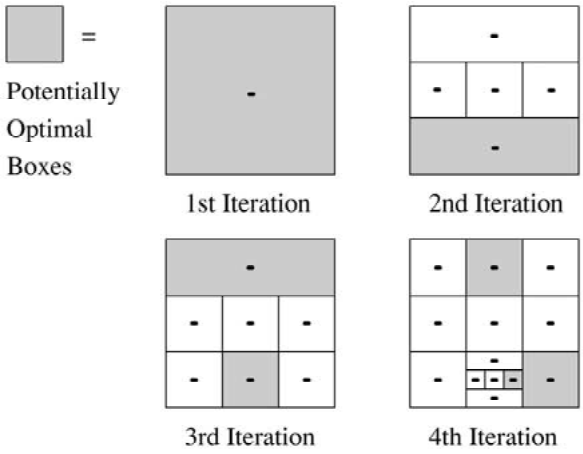

Figure 3. DIRECT box division.

the hypercubes which potentially contain the optimum. A hypercube is selected if using some Lipschitz constant K, that box could contain the lowest function value (see Fig. 2). It can be shown that this requires the box to lie on the bottom part of the convex hull of the set of boxes in a graph such as Fig. 2. In Fig. 2, each box is one of four sizes. The best design from the smallest box and largest and next to largest boxes are potentially optimal because, for each there is some value of K where that box could contain a better design than any other box.

The DIRECT algorithm is as follows:

(1)) Normalize the search space to the unit hypercube. Let c_1 be the centerpoint of the hypercube and evaluate $f(c_I)$

(2) Identify the set, S, of potentially optimal boxes.

(3) For each box $j \in S$:

  (a) Identify the set of dimensions, I, which correspond to the longest sides of the box. Let δ equal one-third of the length of these sides.

  (b) Sample the function at the points $c \pm \delta e_i$ for all $i \in I$, where c is the center of the box and e_i is the i^{th} unit vector.

  (c) Divide only the box containing c into thirds along the dimensions in I, starting with the dimension with the lowest value of $f(c \pm \delta e_i)$ and continuing to the dimension with the highest $f(c \pm \delta e)_i$. A each dimension is divided, the points $c \pm \delta e_i$ will be the center of the new boxes formed by the division.

(4) Repeat (2) and (3) until stopping criterion is met.

Graham's Scan routine is used to identify the potentially optimal boxes (Jones et al., 1993). Boxes that were not previously potentially optimal can become potentially optimal in later iterations as the boxes are divided (see Fig. 3). DIRECT can select more than one box to divide at each iteration. This can allow for a wider search than Lipschitzian optimization and makes DIRECT ideal for parallelization.

DIRECT was found to be quick to locate regions of local optima but slow to converge. To speed up the convergence, here DIRECT is stopped once the smallest box reaches a specified percentage of the original box size and a local optimizer is used for the final optimization. The optimization was stopped when the smallest box reached was 0.01% of the original box size for the HSCT comparisons and 0.001% for the test functions presented in the next section. These values were chosen based on initial experiments with other problems. These experiments indicated that the optimizer was not very sensitive to the percentage used in the stopping criteria and worked well for values from 0.1% down to 0.00001%. DOT was then used starting from up to 15 of the best points analyzed by DIRECT, which were at least a certain percentage of the width of the design space away from the other starting points selected for local optimization. For the HSCT and quartic functions 5% separation was used, while 0.5% separation was used for the Griewank function. The separation used was based on general knowledge of the types of problems being optimized. The Griewank function has trigonometric noise with many clustered optima, the Quartic function has no noise but many widely separated optima. The HSCT problem has some noise and a few widely separated optima.

DIRECT uses a linear penalty function to handle constraints, $g_i(\mathbf{x}) \leqslant 0, i = 1, 2, \ldots, m$. Initial work with this code was based on a quadratic penalty function. This tended to over penalize design points with moderate constraint violations. Due to the sparse sampling of the design space in the first few iterations, this penalty function could eliminate large regions from consideration. A linear penalty function was better able to sample more of the design space while still concentrating on the most promising regions. The augmented objective function $F(x)$ is given as

$$F(x) = f(x) + \sum_{i=1}^{n} g_i(x) P_i, \tag{1}$$

where f is the objective function, $\boldsymbol{g} = (g_1, g_2, \ldots, g_m)^{\mathbf{T}}$ is the constraint vector, and P_i is the penalty parameter, which is zero for unviolated constraints and a small positive number for violated constraints.

The performance of DIRECT depends on the magnitude of the penalty parameter, P. The parameter should be as small as possible while still preventing the optimizer from selecting an infeasible design. Choosing a value that is twice the estimated magnitude of the largest Lagrange multiplier would help prevent the optimizer from converging to an infeasible solution without over penalizing boxes with an infeasible center.

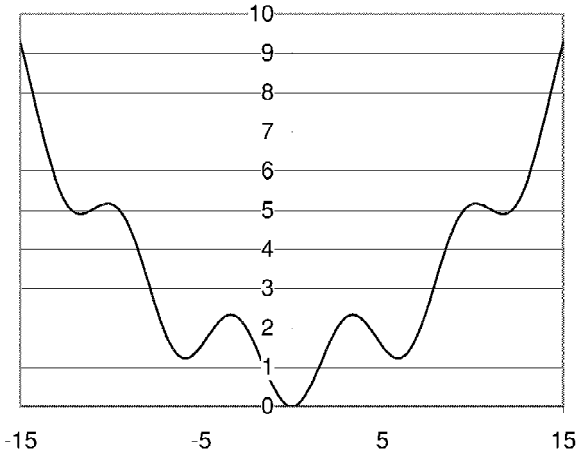

Figure 4. Griewank Function in One Dimension.

3. Simple Test Functions

The optimizers were compared for two algebraic test functions in addition to the HSCT problems. The Griewank function was used as an example of a problem with one true global optimum superimposed with noise. A simple quartic function was optimized in a hypercube to examine the optimizer's ability to locate the global optimum from a large number of widely separated local optima. Each optimizer was run multiple times for five, 10, and 20 design variables (DV) and the results are given below.

The Griewank function is a quadratic function with noise in the form of a cosine function. Without the noise, the function is convex with one optimum. This function was used to test how well the optimizers could move through the noise to locate the actual optimum. The one-dimensional Griewank function is depicted in Fig. 4.

The *n*-dimensional Griewank function is defined as

$$F(\mathbf{x}) = 1 + \sum_{i=1}^{p} \frac{x_i^2}{d} - \Pi_{i=1}^{p} \cos\left(\frac{x_i}{\sqrt{i}}\right).$$

The relative strength of the noise can be controlled by adjusting d. When d is small, the Griewank function is primarily a quadratic function with a small noise component from the cosine terms. As d is increased, the objective function becomes flatter and the cosine portion becomes more important. The function changes from a noisy function with one global optimum, to an almost flat surface with hundreds of nearly equal local optima.

The constant d is taken to be 200, 1000 and 20 000 for the 5, 10, and 20 DV cases, respectively. The design domain was $[-400,600]^n$ for the DOT and LFOPCV3 optimizers with random starting points. For DIRECT, the box had edges

Table 1. Comparisons for 5, 10, and 20 DV Griewank functions

5 DV	# of times global optimum located / no. of runs	function evaluations per global optimum	90% Confidence
DIRECT	87 / 100	3400	3335
DOT	5 / 300	10260	23416
LFOPCV3	60 / 400	9050	19235
10 DV			
DIRECT	56 / 100	11810	18550
DOT	96 / 500	1260	2604
LFOPCV3	136 / 500	32240	63598
20 DV			
DIRECT	8 / 30	102650	231593
DOT	16 / 2000	19740	45265
LFOPCV3	128 / 2000	7220	16084

of length 1000 with the upper limit randomly set between 100 and 900 to randomize the results for the different runs. Each optimizer was run enough times to compute the average number of evaluations for each optimum found. The global optimum is at $\mathbf{x} = \mathbf{0}$. For this comparison, the optimizer was considered to have reached the global optimum if all of the design variables were in the range ± 0.1. The optimizers were compared based on the number of function evaluations required and the number of times each optimizer located the global optimum. The results are given in Table 1.

We use two measures of efficiency of the optimizers. The first is the average number of function evaluations per optimum. The second measure is the number of function evaluations needed to achieve 90% confidence that the global optimum was found. Even though DIRECT is not random there is still chance involved. To calculate the number of function evaluations for 90% confidence, we start by observing that the probability of not locating the optimum with one optimization run is

$$p_1 = (n-1)/n,$$

where n is the total number of optimization runs per optimum located. The probability of not locating the optimum in r optimizations is

$$p_r = ((n-1)/n)^r.$$

We set p_r=0.1, and solve for r, which gives the number of runs needed for 90% confidence that the global optimum was found. Multiplying this number by the average number of function evaluations per run, a, gives the desired number of function evaluations for 90% confidence

$$f_{90} = a\frac{\ln(0.1)}{\ln\left(1-\frac{1}{n}\right)}.$$

When p_1 is close to one, it is easy to check that the number of function evaluations for 90% confidence is approximately 2.3, (ln10), times the average number of function evaluations per optimum. To illustrate the difference between the two measures of performance, consider the following example. Method A finds the optimum 40% of the time with 100 function evaluations per run. Method B requires 300 function evaluations but finds the optimum every time. On the average measure, method A is better, with an average of 250 evaluations per optimum. However, three runs with method A, requiring 300 evaluations, have $(1-0.6^3)$ = 21.6% chance of missing the optimum while method B is sure to find it.

The results in Table 1 do not give a clear advantage to any one optimizer but they do indicate that DIRECT is not the most efficient optimizer in higher-dimensional space. The Griewank function is smooth with an underlying quadratic shape. This is suitable for gradient based optimization if the algorithm is capable of moving past the weak local optima. LFOPCV3 does this by design while DOT is able to do this by using approximations based on widely separated points.

The lack of a clear trend in Table 1 may be due to the effect of d, which was different for each case. The relative strength of the noise changed as the number of design variables increased. To investigate the effect of the noise on the different optimizers, the 10 variable case was repeated for different values of d. The results for this comparison are given in Table 2.

As expected, the optimization is easier for small values of d, where the global optimum is more distinct from other local optima. DOT appears to be the least sensitive to this parameter while LFOPCV3 was the most sensitive to d. The approximations used by DOT appear to allow it to move past the noise and capture the underlying quadratic function. LFOPCV3, however, is slowed down as it passes through the noise and is not able to locate small improvements in the local optima for the large value of d.

The one-dimensional quartic function is given in Fig. 5. The n-dimensional quartic function is defined as

$$f(\mathbf{x}) = \sum_{i=1}^{n}[2.2(x_i + e_i)^2 - (x_i + e_i)^4],$$

where n is the number of dimensions and e_i is a random number in the range [0.2, 0.4]. The design space was the hypercube $[-2,2]^n$. The global optimum is at $\mathbf{x}$ = 2. This problem contains 3^n local optima located at the constraint boundaries and

Table 2. Effect of variation of *d* in 10-dimensional Griewank function

	No. of times global optimum located	Average no. of function evaluations per global opt.	90% Confidence
DIRECT, 100 runs			
d = 4000	19	39 480	81 958
d = 1000	56	11 810	18 550
d = 200	83	6990	7539
DOT, 500 runs			
d = 4000	37	3970	8801
d = 1000	96	1260	2604
d = 200	160	620	1188
LFOPCV, 500 runs			
d = 4000	25	193 410	434 117
d = 1000	136	32 240	63 598
d = 200	390	830	989

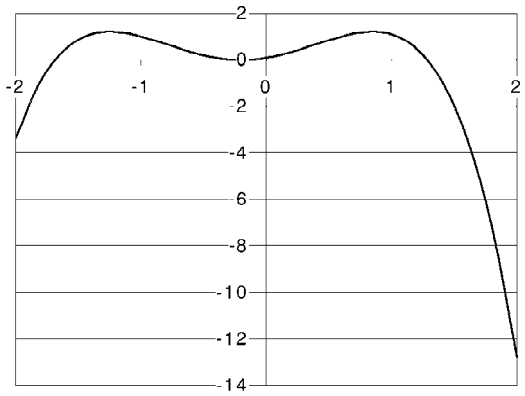

Figure 5. Quartic Test Function (*e*=0.3).

close to the center of each variable. This problem was used to examine the optimizer's abilities to locate the global optimum for a nonconvex objective function with a large number of widely separated local optima. The optimizer was considered to have located the optimum if all of the design variables were greater than 1.9.

DIRECT was run for 200 random values of ***e*** for each case. LFOPCV3 and DOT used 20,000 different starting points with random values of ***e*** in order to get a statistically meaningful average number of function evaluations per optimum located. The results are given in Table 3.

For the Griewank function, none of the optimizers were 100% effective in locating the optimum. For the Quartic function, DIRECT found the optimum every time. This means that a single run of DIRECT more than satisfies the 90% requirement. For this reason the 90% confidence column equals the function evaluations per optimum column for DIRECT.

DIRECT is clearly better suited for this problem than either DOT or LFOPCV3. The space partitioning method employed by DIRECT allows it to examine a wider range of points, which increases the likelihood of locating the basin that contains the global optimum. It does not remain in a single basin but continues to search other regions while refining the optima located previously.

DOT is unable to adequately capture the shape of the entire design space with the quadratic approximation. The quadratic approximation is only good for describing small portions of the design space, which prevents it from accurately extrapolating to the regions of other local optima.

LFOPCV3 is strictly a local optimizer for this problem. It is designed to move through noise and weak local minima, not the deep basins found in this problem. In order for this optimizer to find the global optimum, it must start in the basin that contains the global optimum. For $\boldsymbol{e} = 0.3$, the odds of starting in this region for the 5, 10, and 20 DV cases are 1 in 333, 1 in 111 000 and 1 in 123×10^8 respectively. This makes multistart local optimization methods a poor choice for this type of problem. LFOPCV3's performance in the 5 DV case if fact indicates that, based on the above probabilities, it is not as good at locating the global optimum as one would expect. In many cases LFOPCV3 stopped at saddle points and points where the slope was not zero.

4. HSCT design problem

The design problem used to compare the three optimization methods is the configuration design of a HSCT. The HSCT design code uses up to 26 design variables and up to 68 constraints (MacMillin et al., 1997). The design goal is to minimize the gross take off weight (GTOW) while satisfying 68 range, performance, and geometric constraints.

The HSCT code uses simple structural and aerodynamic models to analyze a conceptual design of the aircraft. In more recent versions of the code, the aerodynamic analysis for the drag is replaced by a response surface constructed on the basis of a large number of simulations.

The current version of the HSCT code employs an aerodynamic drag response surface constructed by Knill et al. (1999) based on solutions of the Euler equations (McGrory et al., 1993). Once the variable designs are selected using design of experiments theory, the wing camber for each design is found using Carlson's modified linear theory optimization method WINGDES (Carlson and Miller, 1974); Carlson and Walkley, 1984). This design is then analyzed using the Euler equations. The Euler analysis is made at two angles of attack and a response surface is

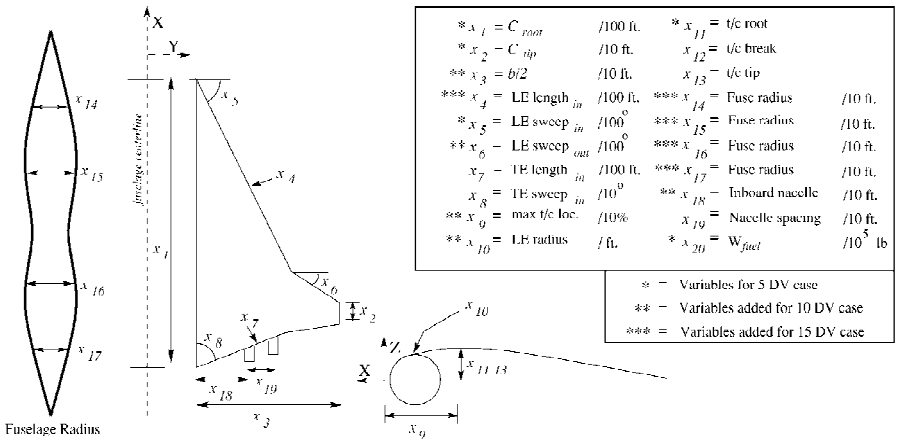

Figure 6. Definitions of design variables.

constructed for the drag polar in terms of the intervening variables C_{Do} and K:

$$C_D = C_{Do} + KC_L^2,$$

where C_D is the drag coefficient, K is the drag polar shape parameter and C_L is the lift coefficient. The viscous contribution to the drag, C_{Do}, is obtained from standard algebraic estimates of the skin friction assuming turbulent flow (Hopkins, 1972).

Previous work with the HSCT code has demonstrated the nonconvexity of the feasible domain and the existence of multiple local minima (Knill et al., 1999). In addition, several analyses and suboptimizations, including range calculations and structural weights, result in noisy performance functions. This noise adds additional local minima and makes accurate gradient calculations difficult. Optimization methods must be able to deal with this noise and move from one region of design space to another without becoming trapped in local minima.

The HSCT code optimization can work with subsets of 5, 10, 15, or 20 of the design variables as defined in Fig. 6. The variables designated with one asterisk are used in the 5 DV case. The 10, 15 and 20 DV cases add the variables designated with two, three and four asterisks respectively. The variables used are described in more detail in Knill et al. (1999).

These design cases were used to compare the performance of the optimizers. The 5 DV cases have only one optimum, while the higher-dimensional designs contain many local optima and nonconvex feasible regions. This allowed us to compare the optimizers on problems of different complexities. The lowest objective function value found for each case was taken as the global minimum.

For the HSCT problem, the range constraint had the largest effect on the objective function. Based on previous work, the HSCT requires approximately 90 lb of fuel per mile of range deficiency, so this provides an estimate of the Lagrange multiplier associated with that constraint. Therefore, for DIRECT the penalty constant was chosen to increase the objective function by twice this much per mile of

Table 3. Comparisons for 5, 10, and 20 DV Quartic functions

	No. of times global optimum located / no. of runs	Average no. of function evaluations per global opt.	90% confidence
5 DV, DIRECT	200 / 200	1025	1025
DOT	410 / 20 000	3397	7741
LFOPCV3	8 / 20 000	2581418	5942746
10 DV			
DIRECT	200 / 200	2192	2192
DOT	16 / 20 000	174189	400925
LFOPCV3	0 / 20 000	–	–
20 DV			
DIRECT	200 / 200	11266	11266
DOT	0 / 20 000	–	–
LFOPCV3	0 / 20 000	–	–

violation. The range constraint is given as

$$g_1 = 20\left(\frac{R}{5500} - 1\right) \tag{2}$$

where R is the range. The objective function, f, is the gross weight normalized by 700 000 lb. This gives a value of 0.071 for the penalty multiplier. For our study we rounded this up to 0.1.

5. HSCT results

For each design case, the HSCT code was run for 100 starting points for DOT and LFOPCV3, and once for DIRECT. Table 4 shows the results from these runs. The number of optima and 90% confidence columns here refer to designs within 1000 lb of the best optimum design out of all three optimizers and the number of function evaluations needed to find such a design, respectively. In order to ensure that DIRECT could locate the same optimum with slightly different starting conditions and numerical noise, it was run a second time for each case with the design box perturbed by 1% (larger perturbations would degrade the quality of the response surface). This altered the noise component of each analysis without substantially changing the design space. In each case, DIRECT located an optimum that was within 300 lb of the original optimum. For a small number of design variables DOT has a large advantage over DIRECT, but this advantage diminishes with dimension, and for the 20 design variable case, DIRECT is slightly better. LFOPCV3 did very

Table 4. Comparison of 5, 10, 15, and 20 DV HSCT problem

	Best GTOW Located	Function evaluations	No. of Opt[a]	90% Confidence
5 DV DIRECT	638 238	2345	1	2345
DOT	638 231	10 870	94	89
LFOPCV3	638 813	33 234	5	14 919
10 DV				
DIRECT	624 751	9067	1	9067
DOT	624 731	29 791	12	5366
LFOPCV3	631 300	137 929	0	–
15 DV				
DIRECT	603 127	40 662	1	40 662
DOT	602 685	47 440	12	8545
LFOPCV3	620 538	1,437 312	0	-
20 DV				
DIRECT	588 404	51 147	1	51 147
DOT	588 586	57 428	2	65 453
LFOPCV3	620 092	418 315	0	–

[a] One run for DIRECT, 100 runs each for DOT and LFOPCV3.

poorly, and was not able to find the global optimum except for the five-variable case. LFOPCV3 was slowed down by the need to continuously calculate gradients by finite differences while DOT was able to construct an approximation to the problem which reduced the number of gradient calculations it required.

For the 5 DV case, each optimizer found approximately the same global optimum. This was expected due to earlier experiments that showed the 5 DV feasible set was convex with only one local optimum. The difference in the design variables is less than 1%, which can be attributed to noise in the objective function causing premature convergence for LFOPCV3. Our implementation of DIRECT uses DOT for the final convergence, which should cause both to reach the same design when optimizing in the same local region. Here DOT was the most efficient optimizer by far since it did not require more than a few starting points to locate the optimum.

For the 10 DV case, DIRECT and DOT located a design of the same weight, with DOT being more efficient by a factor of two. The 10 DV case has distinct local optima (Knill et al., 1999), and LFOPCV3 could not locate the global optimum.

For the 15 DV case DOT found a slightly lower weight than DIRECT and was also much more efficient. However, this difference is less than 0.08% of the total weight of the HSCT. This case illustrates the unpredictable performance of

Table 5. 15 DV optima for DOT and DIRECT[a]

Design variables	DOT	DIRECT	% Difference
Root chord	1.7306	1.7116	4.2
Tip chord	0.8425	0.8563	2.3
Semispan	6.9164	6.9336	1.1
LE length in.	1.2052	1.2074	1.0
LE sweep in.	0.7128	0.7121	1.2
LE sweep out.	0.1205	0.1300	4.8
Max t/c loc.	4.8400	4.8502	0.6
LE radius	3.0660	2.4353	33.2
t/c ratio	1.9872	2.0304	5.4
Fuse radius 1	0.5176	0.5144	2.1
Fuse radius 2	0.5669	0.5699	2.0
Fuse radius 3	0.5659	0.5699	2.7
Fuse radius 4	0.4966	0.4928	2.5
Inboard nacelle	2.8293	2.8519	0.9
Fuel wt.	3.0938	3.1005	1.1

[a] The design variables in Table 5 are defined in Fig. 6.

a multistart method. Out of the initial 100 random starting points, DOT was only able to locate the global optimum once but it found seven designs with weights as good as DIRECT's result. To evaluate the likelihood of finding the optimum with DOT, another 100 random starting points were chosen for DOT. DOT was unable to locate the best optimum for this set of starting points. Examination of the design space near the best optimum indicated that it lies in a small feasible region, which explains the difficulty locating it. The designs found by DOT and DIRECT are shown in Table 5. They differ only slightly in most of the design variables. This indicates that the difference between the two is primarily due to noise preventing the optimizer from moving through a narrow region of similar weights to find the best optimum.

For the 20 DV case, DIRECT performed slightly better than DOT. The weights they found differed by only 200 lb, but the design variables differed by as much as 35% of their respective ranges. This case showed that DIRECT is still able to perform well for a 20-dimensional design space. The number of function evaluations increased rapidly for more than 10 dimensions, but it was still less expensive than the two multistart methods for the 20 DV case. Without prior knowledge of the design space, it is difficult to decide on the number of starting points to choose for the multistart methods.

In an effort to determine the number of local optima scattered throughout the design space, the optimum designs from the 10 DV DOT optimizations were com-

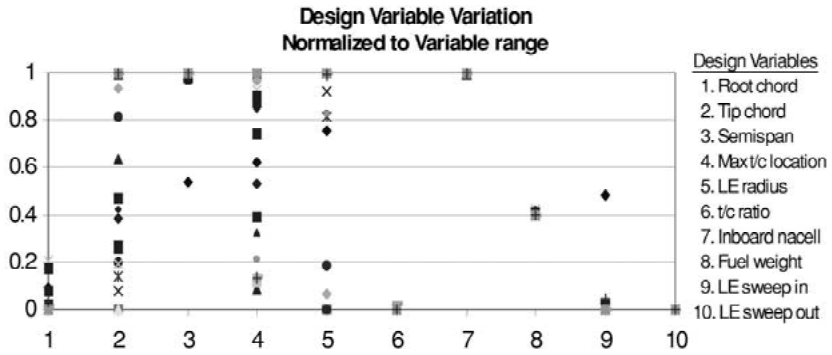

Figure 7. Distribution of design variables for the best 25 designs from the 10 DV HSCT case.

pared. Figure 7 shows the distribution of the design variables for the 25 best optimizations found by DOT over the range of each design variable. The range of the GTOW was only 16 000 lb. but some of the variables differ by their entire range. This plot indicates that there are some variables, such as the root chord, t/c ratio, the inboard nacelle location, fuel weight, and the outboard sweep angle, which tend to a small range of values for the optimized design. The other variables fluctuate widely without affecting the GTOW much. This indicates that the design space contains narrow channels with slowly varying weights. DOT easily locates these regions with similar weights but can not always move through them without becoming trapped by minor fluctuations in the objective function.

The HSCT problem has both physical local optima and numerical noise. The good performance of DIRECT and DOT for this problem is consistent with the fact that DOT handled well low amplitude noise for the Griewank test function, and DIRECT handled well widely separated local optima for the quartic function. As the dimensionality of the problem increases, the number of substantially separated optima can increase exponentially, and the effectiveness of multistart methods appears to deteriorate faster than that of DIRECT. The poor performance of LFOPCV3 compared to the Griewank problem may have to do with the fact that the noise for the HSCT problem is not differentiable.

6. Concluding remarks

Three optimization procedures were tested for the global optimization of a HSCT. DOT (a local optimizer used with sequential quadratic programming) and LFOPCV3, (a semi-global optimizer) were applied with random multistarts. A modification of DIRECT, (a global Lipschitzian optimizer which used SQP for the final convergence) was applied in a single run.

The three optimization procedures were first tested on two simple algebraic problems. The Griewank function is a convex function superimposed with low amplitude noise from a trigonometric function. The quartic test function does not

have any noise but displays a large number of widely separated local optima. These two functions revealed that the two multistart procedures, DOT and LFOPCV3, dealt well with trigonometric noise, while DIRECT was much more effective in finding the global minimum among widely separated local optima.

The HSCT problem presents a combination of numerical noise and multiple physical optima, and thus has elements from both test functions. For a small number of design variables DOT had a large advantage over DIRECT, but the advantage decreased with increasing dimensionality, with the two having similar performance for the 20 variable case. LFOPCV3 did poorly for the HSCT problem, possibly because of the effect of the noise on the derivatives that have to be calculated by its search procedure more frequently than by DOT.

7. Acknowledgement

This research was supported in part by NASA grant NAG2-1179.

References

1. Carlson, H. W. and Miller, D. S. (1974), Numerical Methods for the Design and Analysis of Wings at Supersonic Speeds, NASA TN D-7713, Dec.
2. Carlson, H.W. and Walkley, K.B. (1984), Numerical Methods and a Computer Program for Subsonic and Supersonic Aerodynamic Design and Analysis of Wings with Attainable Thrust Corrections, NASA CR-3808
3. Carter, R., Gablonsky, J.M., Patrick, A., Kelley, C.T. and Eslinger, O.J. (2000), Algorithms for Noisy Problems in Gas Transition Pipeline Optimization, Tech. Rep. CRSC-TR00-10, Centerfor Research in Sci. Computation, North Carolina State University, Raleigh, NC
4. Cramer, E.J. (1998), Using Approximate Models for Engineering Design, *7^{th} AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St Louis, Missouri, AIAA Paper-98-4716, Sept.
5. Floudas, C.A. and Pardalos, P.M. (1996), *State of the Art in Global Optimization,* Kluwer Academic Publishers, Dordrecht, The Netherlands
6. Gablonsky, J. M. (1998), An Implementation of the DIRECT Algorithm, Tech. Rep. CRSC-TR98-29, Center for Research in Sci. Computation, North Carolina State Univ., Raleigh, NC
7. Haim, D., Giunta, A.A., Holzwarth, M.M., Mason, W.H., Watson, L.T. and Haftka, R.T. (1999), Comparison of optimization software packages for an aircraft multidisciplinary design optimization problem, *Design Optimization* 1, 9–23.
8. Hopkins, E. J. (1972), Charts for Predicting Turbulent Skin Friction from the Van Driest Method, NASA TN D-6945, Oct.
9. Jones, D.R., Perttunen, C.D. and Stuckman, B.E. (1993), Lipschitzan optimization without the Lipschitz constant, *Journal of Optimization Theory and Application* 79, 157–181.
10. Knill, D.L., Giunta, A.A., Baker, C.A., Grossman, B., Mason, W.H., Haftka, R.T. and Watson, L.T. (1999), Response surface models combining linear and Euler aerodynamics for supersonic transport design, *Journal of Aircraft* 36, pp. 75–86.
11. MacMillin, P.E., Golovidov, O.B., Mason, W.H., Grossman, B. and Haftka, R.T. (1997), An MDO Investigation of the Impact of Practical Constraints on an HSCT Configuration, *AIAA 35^{th} Aerospace Sciences Meeting and Exhibit*, Reno, NV, AIAA Paper 97-0098, Jan.
12. McGrory, W.D., Slack, D.C., Applebaum, M.P. and Walters, R.W. (1993), *GASP Version 2.2 Users Manual*, Aerosoft, Inc., Blacksburg, VA,

13. Nelson, S.A. II, and Papalambros, P.Y. (1998), A Modification to Jones' Global Optimization Algorithm for Fast Local Convergence, *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St Louis, Missouri, AIAA Paper-98-4751, Sept.
14. Snyman, J.A. (1983), An Improved version of the original Leap-Frog dynamic method for unconstrained minimization: LFOP1(b), *Appl. Math. Modelling* 7, 216–218.
15. Snyman, J.A. (2000), The LFOPC Leap-Frog Algorithm for Constrained Optimization, *Computers and Mathmatics with Applications*, 40, 1085–1096
16. Vanderplaats Research & Development, Inc. (1995), *DOT: Design Optimization Tools*, Version 4.20