

D-Optimal Designs for Sensor and Actuator Locations

Rex K. Kincaid †

Department of Mathematics
The College of William and Mary
Williamsburg, Virginia 23187-9795

Sharon L. Padula

Multidisciplinary Optimization Branch
NASA Langley Research Center
Hampton, Virginia 23681

Abstract: Active control of noise and vibration is now possible in automobiles, aircraft, and many other devices. Where to place actuators, to control noise and vibration, and sensors, to measure the performance of the actuators, is a central question. Given a truss structure, we seek the k most effective locations to control and/or sense vibrations. A discrete D-optimal design has been proposed as a solution to this location problem. We develop a simple static tabu search and test its performance on an 80 node truss structures built at NASA-Langley Research Center. We show that our tabu search approach dominates the traditional approaches to finding D-optimal designs.

Keywords: location, D-optimal, tabu search, heuristics

1. Introduction.

From aircraft fuselages and space stations to vacuum cleaners and automobiles, active control of noise and/or vibration has come of age. Determining the number of active control devices (e.g. actuators) to be placed and where they are to be placed is the prototypical location problem. However, unlike typical location problems, where the customer is readily identified and is actively engaged in the assessment of the performance of the chosen locations, the customers that active control devices serve are not so easily identified and their impact on system performance issues may be unclear. For example, consider the problem of where to locate actuators to attenuate cabin noise in a propeller driven aircraft (c.f. Palumbo et al. 1996, Kincaid et al. 1997, Palumbo and Padula 1997, and Kincaid and Padula 1998). Clearly, the ultimate customers are the passengers who will travel in these aircraft. But to decide whether one set of actuator locations is better than another it is unlikely we will ask passengers to fly in the aircraft and fill out a questionnaire about noise levels. Instead a set of sensors (pseudo-customers) are placed and the system performance of the actuators, as measured by these sensors, is recorded. Hence, we have yet another location problem. How many sensors should there be and where should they be located? In many instances collocation of sensors and actuators is the answer but in other instances it is not.

† The author was supported in this research by NASA grant NAG-1-1783.

A variety of approaches have been taken to address these sensor/actuator location problems. With regard to damping vibrations in truss structures (space station prototypes) Kincaid and Berger (1993) and Kincaid (1995) formulated a new noxious location problem and generated high-quality solutions with a combination of LP-relaxations and heuristic search procedures. Other related efforts are summarized in Padula and Kincaid (1995). In Kincaid et al. (1997) the actuator location problem for a single frequency interior noise control problem was examined for an idealized aircraft cabin. A tabu search procedure was shown to generate better locations for the actuators than a modal decomposition approach. Kincaid and Padula (1998) extended the model to include multi-frequency information. The sensor location problem is addressed in both Kincaid et al. (1997) and Kincaid and Laba (1998). In the latter article a reactive tabu search scheme was shown to dominate a static tabu search approach.

Our focus here is to determine locations to control and/or sense vibrations on a truss structure. However, instead of using one of the earlier optimization models referenced in the above paragraph we adopt an experimental design approach. Given a matrix with one column for each mode to be identified (or sensed) and one row for each potential sensor location, we seek a submatrix of maximal determinant—a D-optimal design. Our interest in this experimental design approach comes largely from a series of papers by Kammer (1991,1992,1996). Kammer demonstrates that D-optimal designs are good solutions for sensor placement. Our purpose here is twofold—(1) to improve upon the solution approach given by Kammer and (2) to show how the performance of traditional experimental design algorithms can be improved by incorporating the basic elements of tabu search.

2. Problem Description

The optimization problem is easily stated in mathematical terms. Given an m by n matrix X , $m \gg n$, select k rows of X . The result, \bar{X} , is a k by n submatrix of X . The goal is to form an \bar{X} so that the determinant of $\bar{X}^t \bar{X}$ is a maximum over all possible k row combinations of X for fixed $k \geq n$. Unless the dimensions m and k are small the time to perform a complete enumeration and evaluation of all m choose k submatrices \bar{X} is prohibitively long (i.e. centuries). Clearly $\bar{X}^t \bar{X}$ is positive semi-definite (PSD) but may not be positive definite (PD). We will be interested in selecting \bar{X} so that the columns of $\bar{X}^t \bar{X}$ will be linearly independent (\bar{X} is PD). We know of no mathematical results or combinatorial algorithms that can guarantee a globally optimal solution to the above maximum determinant problem. There is, however, an extensive literature for this maximum determinant problem in design of experiments (cf. Atkinson 1988 and Nguyen and Miller 1992) and, closer to our interests, in sensor placement on structures (cf. Kubrusly 1985).

In an experimental design the goal is to choose k points from a finite set of candidate points of size m . Consider a linear model $y = \bar{X}\beta + e$ where y is a k dimensional vector of observations, \bar{X} is a k by n matrix of known elements, β is an n dimensional vector of unknown parameters, and e is a k dimensional vector of random residual components. Assume that e has a mean of zero and a dispersion

of $\sigma^2 I$. The least squares normal equation for estimating β is $\bar{X}^t \bar{X} \beta = \bar{X}^t y$. $\bar{X}^t \bar{X}$ is called the Fisher information matrix and its inverse is called the variance-covariance matrix. A discrete D-optimal design is one in which the k points chosen maximize the determinant of the Fisher information matrix (equivalently minimizes the determinant of the variance-covariance matrix).

The solution approach in the experimental design literature is to use multi-start greedy 2-exchange algorithms. Nguyen and Miller (1992) survey these algorithms and note that the earliest references in this regard are Fedorov (1969,1972). One interesting departure from the traditional exchange algorithms is the branch and bound scheme employed by Welch (1982). Two additional features—blocks and row repetition—common in the experimental design literature have no application in our sensor location format and will not be discussed here. Repeated rows, for example, would imply that more than one sensor can be located at the same node in the same orientation (x, y or z). One of the outcomes of this research is to demonstrate the ease with which these greedy exchange algorithms can be augmented with rudimentary tabu search concepts. DOPT is the 2-exchange algorithm we have chosen to embed within a tabu search framework. It was developed by Miller and Nguyen (1994) and is available online (<http://lib.stat.cmu.edu/designs/dopt>).

Although the tabu search code we develop is directly applicable to experimental design, our motivation is with respect to the placement of sensors and/or actuators on structures for vibrational control. Sensors are used to identify the significant modes of a structure and/or to evaluate the performance of the actuators in controlling the modes when excited. Sensors and actuators are often co-located since good locations for identifying the modes are also good locations for controlling the modes. For these applications the matrix X has one row for each potential sensor (or actuator) location and one column for each structural mode that is to be controlled, damped or identified. As before, our goal is to select k of the sensors (or actuators) to maximize the determinant of $\bar{X}^t \bar{X}$ with the additional constraint that $\bar{X}^t \bar{X}$ is PD. If this matrix is indefinite Kammer (1991) has shown that the resulting sensor placement (the rows of \bar{X}) will detract from the independent identification of the modes. Kammer (1991,1992,1996) generates local optima to the discrete D-optimal design problem for sensor placement as solutions to the modal identification problem. However, instead of using a multi-start greedy exchange approach he uses an iterative improvement scheme. He starts with the null set and successively adds one sensor location at a time based upon what he calls an *effective independence measure*. This measure ranks the candidate sensor locations by their contribution to the linear independence of the finite element model representation of the target modes. Yao et al. (1993) attempt to improve on the effective independence iterative improvement scheme (EFI) with a genetic algorithm (GA). The GA was able to produce improved determinant values over the EFI approach but at a large computational cost. Lastly, Tongco and Meldrum (1996) also use the discrete D-optimality criteria in selecting sensor locations for a planar truss structure, similar to the structure we examine, but there is no optimization involved. Instead they enumerate and evaluate all possible solutions which is clearly impossible for truss

structure of even a modest size.

3. *Experimental Model*

The test article is an 80 node planar Vierendeel truss (Figure 1) built at NASA-Langley Research Center. Each node of the truss may have a sensor placed in either the x , y or z direction for a total of 240 potential sensor locations. In addition, the truss has up to 20 vibrational modes that may be considered for identification and/or control. A NASTRAN finite element model was used to generate the first 20 eigenvectors. The NASTRAN output file was filtered to produce the $m = 240$ by $n = 20$ matrix X described in Section 2. The physical dimensions of the truss structure are 24 inches in length and width. Each truss element is 1/2 inch by 1/4 inch. It is hoped that the lessons learned on a small-scale truss will carry over when trusses of larger physical dimensions (though perhaps with the same number of nodes) are considered.

Figure 1. Sensor locations for Vierendeel planar truss

4. *Tabu Search Description*

A simple tabu search procedure was developed with the underlying neighborhood search scheme based upon the DOPT code developed by Miller and Nguyen (1994). DOPT is a greedy 2-exchange local search that seeks to maximize the determinant of $\bar{X}^t \bar{X}$. It allows repeated row selection and the ability to handle blocks. Both of these features were deleted from their code since they do not apply to the sensor/actuator selection problem.

DOPT begins by generating an initial selection of k rows from X . Half the rows are selected at random. The remaining rows are added sequentially to maximize

the rank and the determinant. To accelerate the construction process roughly \sqrt{m} of the rows are considered each time a new row is to be added. In addition, instead of X , a Cholesky factorization of X is maintained throughout the algorithm. If, after construction, \bar{X} is not of rank k a 2-exchange sweep through all rows not in \bar{X} is performed. A row is added if the rank of \bar{X} is increased. The process terminates when the rank of \bar{X} reaches k or all 2-exchanges have been performed.

Once an initial solution has been constructed, DOPT examines all pair-wise exchanges between rows of \bar{X} (These row indices form the k vector IN .) and the rows of X that are not in \bar{X} (These row indices form the $m - k$ vector OUT). If a row exchange increases the determinant of $\bar{X}^t \bar{X}$ then the exchange is made. The process terminates when no exchanges remain that improves the determinant. Since it is expensive to calculate the determinant ($O(n^3)$) for an n by n matrix, an efficient updating scheme is critical. Let $F = \bar{X}^t \bar{X}$, let x_+ denote the row of X to be considered for addition, and let x_- denote the row of \bar{X} to be considered for deletion. Then the change in the determinant

$$\Delta(x_+, x_-) = x_+^t F^{-1} x_+ - x_-^t F^{-1} x_- (1 + x_+^t F^{-1} x_+) + (x_+^t F^{-1} x_+)^2.$$

If we write $F = R^t R$ (R is an upper triangular matrix), let $R^t z_+ = x_+$ and let $R^t z_- = x_-$ then

$$\Delta(x_+, x_-) = z_+^t z_+ - z_-^t z_- (1 + z_+^t z_+) + (z_-^t z_+)^2. \quad (1)$$

From the Cauchy-Schwarz inequality, we have that

$$(z_-^t z_+)^2 \leq (z_+^t z_+)(z_-^t z_-). \quad (2)$$

Substituting the right hand side of (2) for $(z_-^t z_+)^2$ in (1) we get

$$\Delta(x_+, x_-) \leq z_+^t z_+ - z_-^t z_-. \quad (3)$$

(3) is preferred to (1) since the two products on the right hand side of (3) can be calculated once for each row on each iteration (see Miller and Nguyen 1994 for more details).

Tabu search is a metaheuristic strategy that can exploit and direct a variety of local search heuristics. Tabu search seeks to exploit historical information gathered during the local search phase so that the search will not remain stalled at a local optima. There are a variety of mechanisms developed to avoid local optima including tabu lists, recency and frequency based diversification schemes, and the detection of basins of attraction. Although the roots of tabu search are present in a variety of early work in artificial intelligence and operations research the seminal paper is Glover (1989). Glover (1996) and Glover and Laguna (1998) provide a comprehensive list of techniques and applications associated with tabu search specifically and adaptive memory programming generally.

Our efforts here center on improving the performance of DOPT. A very simple tabu search framework is placed around DOPT to aid in the escape from local

optima. A static length tabu list is maintained that records the 2-exchanges made on each iteration. In subsequent iterations the inverse move is disallowed so as to avoid (hopefully) a return to a previously observed solution. Equation (3) acts as a candidate selection filter to avoid excessive computations for solutions that will not improve upon the current best solution. It is interesting to note that Miller and Nguyen (1994) comment that by not allowing the previously selected candidate point to leave the solution on the next iteration a small time savings was incurred. This is equivalent to a static tabu list of size 1.

Inputs: S_1 a set of k rows of X (vector IN) in the current solution and a set of $m - k$ rows (vector OUT) not in the current solution; max_it the maximum number of iterations; and a cost function Det (the determinant of $\bar{X}^t \bar{X}$).

```

LogDetbest = Log(Det( $S_1$ ))
do  $i = 1 \dots max\_it$ 
   $\Delta bestDet = -\infty$ 
  do  $j = 1 \dots k$ 
    do  $k = 1 \dots m - k$ 
      exchange  $j$ th entry of  $IN$  with  $k$ th entry of  $OUT$  yielding  $\tilde{S}_i$ 
      compute bound on change in  $Det$  with Eq. (3)
      if (bound <  $\Delta bestDet$ ) then
        compute  $\Delta Det$  the change in  $Det$  if  $i$  and  $j$  are exchanged
        check tabu status of exchange
        if (exchange NOT tabu) then
          make exchange permanent
           $S_i = \tilde{S}_i$ 
           $\Delta bestDet = \Delta Det$ 
        else
          exchange not allowed
        endif
      endif
    end do
  end do
  update tabu list with best exchange
  if ( $LogDet(S_i) > LogDetbest$ ) then
     $LogDetbest = LogDet(S_i)$ 
  endif
end do

```

There are four additional features not outlined in the pseudocode. First, the loops that exchange $j \in IN$ and $k \in OUT$ are randomized. That is, random starting points in the lists IN and OUT are selected. The lists are then examined in order from the random starting point. The next two features both attempt to avoid making certain kinds of exchanges. If the change in the determinant (for

an exchange that passed the bound test of equation 3) is too small then the move is not allowed. The magnitude of ‘small’ is dependent on the magnitude of the determinants. For example when the determinants were on the order of 10^{40} we used a value of .001 for ‘small’. There is no reason to avoid these moves other than a greedy one—we want bigger changes in the determinant. Next, if a move was encountered that resulted in a singular matrix (not PD) then the move is made tabu five times longer than normal. Clearly, we want to avoid repeating such a move and a longer tabu status is warranted. Lastly, upon completion of *max_it* iterations a frequency-based restart is performed and the search continues again for another *max_it* iterations from this new starting solution. For the experiments we report on in Table 1 ten frequency-based restarts were tried. The frequency count tabulates the number of times each row has appeared in any solution during the searches previous *max_it* iterations. When a restart is calculated the k least frequently selected rows are chosen as the new initial solution. The intent of this procedure is to force the search to explore additional regions of the search space.

5. Computational Experiments

Table 1 below illustrates the improvement in the determinant value obtained with our tabu search code versus Kammer’s EFI approach and a multi-start scheme. The runtime for both approaches is small (a few seconds) and is not an issue for the truss described in Section 2. The multi-start scheme calls DOPT 100 times, each time with a different random number seed. The number of replications, 100, was selected so that the multi-start scheme would examine at least as many 2-exchanges as our tabu search. Why did we pick the test cases in Table 1? Our goal was to find a difficult set of test cases.

Our computational experience indicated that when the number of sensors to be located was equal to the number of modes to be identified (or controlled) that the resulting D-optimal design problem was more difficult. For the modes equal number of sensors type problems, the local optima appear to have a wider range of values and appear to lie in steeper valleys. Moreover, the static tabu list has great difficulty escaping the local optima in these cases. Consider a 13 mode 13 sensor example. Here the range of the log of the determinant values is 58.94 to 88.92. In addition, only one of the frequency-based restarts produced a local optima that was improved upon by the use of the static tabu list. Now consider a 13 mode 39 sensor example. The local optima are easier to escape and the values of the local optima appear more uniform. The range of the log determinant values for this example was 102.86 to 102.94. In 7 of the 10 restarts the tabu list was able to escape at least one local optima. As a side issue we note that only when the number of modes equals the number of sensors is the DOPT procedure unable to construct an initial set of rows for which $\bar{X}^t \bar{X}$ is nonsingular (only 6 of the 10 restarts were able to find nonsingular initial solutions in the 13 mode 13 sensor example). Hence, Table 1 records the outcomes of four problems in which the number of modes equals the number of sensors.

The multi-start scheme was able to find the same maximum determinant solu-

tion as tabu search in 2 out of the 4 test cases. In the 16 mode 16 sensor problem, however, even with 500 replications multi-start was unable to find the best solution generated by tabu search. In all four cases the Kammer EFI approach is dominated by both multi-start and tabu search.

modes	sensors	EFI Det	Multi-start	Tabu Det
10	10	1.885 E+28	2.312 E+28	2.312 E+28
13	13	3.023 E+38	4.158 E+38	5.561 E+38
16	16	5.896 E+48	6.853 E+48	6.975 E+48
20	20	1.589 E+62	1.956 E+62	1.956 E+62

Table 1. Best Determinants for EFI, Multi-start, and Tabu Search

A geometric comparison of the Kammer EFI solution and the tabu search solution is also possible. In Figure 2 we show the solutions for the 10 mode 10 sensor and the 16 mode 16 sensor cases. Recall that at each of the 80 potential node locations there are three possible sensor choices—x, y or z directions. In Figure 2 we make no distinction between the the x, y and z sensor locations, so some information is lost. However, the figure does allow us to see the actual node location differences between the two methods. The black X’s denote the Kammer EFI locations and the solid squares denote the tabu search locations. A set of dots to the right (left) of a location designate how many of the three (x, y and z) directions were chosen and if more than one was selected by EFI or Tabu Search.

In the 10 mode 10 sensor case the solutions generated by EFI and tabu search had strong similarities. The two solutions had 7 of the 10 locations in common. The most obvious difference is the selection of node 49 by EFI and node 12 by tabu search (please refer to Figure 1 for node numbers). In the 16 mode 16 sensor case only 4 of the 16 locations were in common between EFI and tabu search. Even though the actual node locations are quite different in these two solutions both placed the same number of sensors on each of the two vertical and two horizontal components of the truss. We don’t know what the global optimal is to either the 10 mode or 16 mode case. However, we speculate that the strong similarity between the EFI and tabu search solutions for the 10 mode case may be linked to the fact that the search terrain is simpler. We believe that the search terrain is simpler because both the multi-start and tabu search solutions are the same for the 10 mode 10 sensor problem. This observation is supported by the fact that for the 20 mode 20 sensor problem (where multi-start and tabu search again find the same solution) 16 of the 20 locations are shared between the tabu search and EFI solutions. Moreover, the 13 mode 13 sensor problem is similar to the 16 mode 16 sensor results. Here only 7 of the 13 locations are in common between EFI and tabu search and multi-start was unable to find as good a solution as tabu search.

Figure 2. EFI and tabu search solutions for 10 mode (top) and 16 mode (bottom) cases.

In an effort to determine the optimal number of sensors we plot the system performance versus the number of sensors locations for the 10 mode case. Figure 3 shows that although system performance (log of determinant) continues to improve as the number of sensors increases the performance flattens out around 100 sensors. Such a result is not unexpected. At some point, there should be a decrease in the marginal return for additional sensor information. It is important to know when this occurs as it may aid the designer in determining the number of sensors to

place. Other issues such as the cost of a sensor and the change in the structural properties of the truss due to the attachment of sensors will be at least as important as performance.

Figure 3. Performance versus number of sensors for 10 modes

The next set of results analyze the effectiveness of the three schemes that filter out unwanted exchanges. The three filters are: the bound filter from equation (3), the tabu list filter, and small changes in the determinant filter. The number of exchanges that fail to pass a particular filter is greatly affected by the order in which the filters are applied. We made no systematic study of what order is best. However, since the purpose of the bound filter is to avoid unnecessary computation for unproductive exchanges it seems logical to check this first. We check the tabu list filter second since more exchanges are likely to be tabu than to fail the small change in the determinant filter. Given this ordering, we examine two cases to test the effectiveness of each of the three filters. We consider the 13 mode 13 sensor and 13 mode 39 sensor location problems. As we discussed earlier in this section the first problem is more difficult than the second. The number of iterations in each case is $max_it = 50$ and 10 frequency-based restarts are performed.

Table 2 records the average number of exchanges that fail to pass each filter each iteration. For the 13 mode 13 sensor case we compute these averages for each iteration up to first occurrence of the best solution. This was early in the search (before iteration 15 in 5 of the 6 completed restarts) again indicating that the tabu list was unable to overcome the local optima. For case 2 we computed the averages over all 50 iterations for each restart. We did this since 9 out of 10 of the restarts encountered their best solution after iteration 40. The three filters are referenced in the table as *bound*, *tabu*, and *small*. Column 4, *bound*, records the average number

of exchanges per iteration that fail to meet the bound condition in Column 5, *tabu*, records the average number of exchanges per iteration that satisfy *bound* but are *tabu*. Column 6, *small*, records the average number of exchanges per iteration that satisfy *bound*, are not *tabu*, but result in an improvement in the determinant that is deemed too small. Column 3 records the total number of exchanges possible on a single iteration— $k * (m - k)$.

We note that the average values can be somewhat misleading. From these averages one might conclude that there are many more *tabu* exchanges (per iteration) than there are exchanges that result in a small change in the determinant (columns 5 and 6 of table 2). Yet in 3 of the 10 restarts in the 13 mode 39 sensor case the opposite was true (once at a rate of nearly 4 to 1). In other words, the variance is significant for the averages reported in columns 5 and 6. However, reporting the average values clearly show the relative effectiveness of these filters in relation to the bound filter of column 4. Percentages are not given in columns 5 and 6 since the percent values are all less than 1 percent. The values in column 4 show little variation and one can expect to see nearly 90 percent of all exchanges on every iteration filtered out by *bound*. This results in a large computational time savings. Our computational experience indicates that using the bound filter, as opposed to equation (1), cuts the runtime roughly in half.

modes	sensors	possible	bounds	tabu	small
13	13	2591	2251/87%	11.2	6.2
13	39	7839	7113/91%	18.0	13.1

Table 2. Filter performance (average per iteration)

6. Discussion

The usefulness of the D-optimal design as a solution for the location of sensors (and actuators) may depend on the number of sensors to be located. The D-optimal design appears to be quite effective when the number of sensors (actuators) to be located is close to the number of modes to be identified (or controlled). When many more sensors (actuators) are available than there are modes the D-optimal design solution tends to cluster locations. Of course, part of the clustering effect is due to the compression of the x , y , and z configurations to a single node, but since we must physically attach sensors devices to the nodes the clustering in the compressed space is important.

In Figure 4 we show a 10 mode 50 sensor solution found by tabu search and compare it to the 10 mode 10 sensor tabu search solution given in Figure 2. In Figure 4 the 50 sensor solution locations are the black squares while the 10 sensor solution locations are the black square outlines (a ‘T’ in the one location that was not chosen in the 50 sensor solution). A set of dots to the right (left) of a location

designate how many of the three (x , y and z) directions were chosen if more than one was selected for the 10 sensor (50 sensor) solution. Clearly, the 50 sensor solution is clustering about the 10 sensor solution. Even the distribution of the sensors in the 50 location case follows the distribution in the 10 location case. For example, the 10 sensor case chose 4 sensors near the middle of the right vertical component of the truss (27, 30, 31, and 35) while the 50 sensor case clusters 18 sensors along the middle. So, 40 percent of the sensors are positioned along the middle of the right vertical component in the 10 sensor case and 36 percent in the 50 sensor case.

Figure 4. 50 versus 10 locations for the 10 mode case.

The observed clustering effect caused by the D-optimal design solution may not be desirable. Future experiments are underway to test the effectiveness of the D-optimal design solutions for an actuator location problem on an aircraft wing. Here the actuators are to be placed to aid in the control of gust and turbulence. The wing has already been built and tested for other purposes. The wing model was first tested in a wind-tunnel to obtain basic flutter characteristics and transfer functions. McGowan et al. (1996) summarize the data produced from ground vibration tests and two wind-tunnel tests. Piezoelectric actuators covered approximately two-thirds of the internal composite plate in the wing. The model has a total of 72 actuators. Our future work will be to evaluate the performance of a D-optimal design solution when selecting k out of the 72 actuators for gust and turbulence control.

References

- Atkinson, A.C. "Recent developments in the methods of optimum and related experimental designs," *Int. Statist. Rev.*, **56** (1988) 99-115.
- Fedorov, V.V., "Theory of optimal experiments," Preprint No. 7 LSM, Izd-vo Moscow State University, Moscow, 1969.
- Fedorov, V.V., Theory of Optimal Experiments, Translated and edited by W.J. Studden and E.M. Klimko, eds. (Academic Press, New York, 1972).
- Glover, F. (1989) "Tabu Search, Part I," *ORSA J. on Computing*, **1**, 190-206.
- Glover, F. (1996) "Tabu Search and Adaptive Memory Programming—Advances, Applications, and Challenges," to appear in Interfaces in Computer Science and Operations Research, Barr, Helgason and Kennington (eds.) Kluwer Academic Publishers.
- Glover, F. and M. Laguna Tabu Search, Kluwer Academic Publishers, Norwell, MA, 1998.
- Kammer, D.C. "Sensor Placement for on-orbit modal identification and correlation of large space structures," *J. of Guidance, Control and Dynamics*, **14** (1991) 251-259.
- Kammer, D.C. "Efficient sensor placement for on-orbit modal identification of sequentially assembled large space structures," *International J. of Analytical and Experimental Modal Analysis*, **7** (1992) 95-109.
- Kammer, D.C. "Optimal sensor placement for modal identification using system-realization methods," *J. Guidance, Control, and Dynamics*, **19** (1996) 729-731.
- Kincaid, R.K. and R.T. Berger, "The Damper Placement Problem on Space Truss Structures," *Location Science* **1** (1993) pp. 219-234.
- Kincaid, R.K. "Solving the Damper Placement Problem using Local Search Heuristics," special issue of *OR Spektrum* on Applied Local Search, **17** (1995) pp. 149-158.
- Kincaid, R.K., K.E. Laba, and S.L. Padula, "Quelling Cabin Noise in Turboprop Aircraft via Active Control," *J. of Combinatorial Optimization* **1**, Issue 3 (1997) 1-22.
- Kincaid, R.K. and K.E. Laba, "Reactive Tabu Search and Sensor Selection in Active Structural Acoustic Control Problems," (with K. Laba) *J. of Heuristics*, **4** (1998) pp. 199-220.
- Kincaid, R.K. and S.L. Padula, "Actuator Selection for the Control of Multi-Frequency Noise in Aircraft Interiors", (with S.L. Padula) to appear *Proceedings of the 2nd International Conference on Metaheuristics*, Sophia Antipolis, France, July 21-24, 1997, Kluwer, Chapter 31, pp. 457-470.
- Kubrusly, C.S. and H. Malebranche, "Sensors and controllers location in distributed systems—a survey," *Automatica*, **21** (1985) 117-128.
- McGowan, A-M.R., J. Heeg, and R. C. Lake (1996) "Results of Wind-Tunnel Testing from the Piezoelectric Aeroelastic Response Tailoring Investigation," presented at 37th AIAA Structures, Structural Dynamics, and Materials Conference, April (AIAA 96-1511).

- Miller, A.J. and N.-K. Nguyen, "Algorithm AS 295: A Fedorov Exchange Algorithm for D-optimal design," *Appl. Statist.*, **43** (1994) 669-678.
- Nguyen, N-K. and A.J. Miller, "A review of some exchange algorithms for constructing discrete D-optimal designs," *Computational Statistics and Data Analysis*, **14** (1992) 489-498.
- Padula, S.L. and R.K. Kincaid, "Aerospace Applications of Integer and Combinatorial Optimization," (with S.L. Padula) NASA Technical Memorandum 110210, 1995.
- Palumbo, D., S.L. Padula, K.H. Lyle, J.H. Cline, and R.H. Cabell (1996) "Performance of Optimized Actuator and Sensor Arrays in an Active Noise Control System," presented at 2nd AIAA/CEAS Conference, AIAA 96-1724.
- Palumbo, D. and S.L. Padula (1997) "Optimizing an Actuator Array for the Control of Multi-Frequency Noise in Aircraft Interiors," presented at 3rd AIAA/CEAS Aeroacoustics Conference, May (AIAA 97-1615).
- Qureshi, Z.H., T.S. Ng, and G.C. Goodwin, "Optimum experimental design for identification of distributed parameter systems," *International J. of Control*, **31** (1980) 21-29.
- Tongco, E. and D. Meldrum, "Optimal sensor placement of large flexible space structures," *J. of Guidance, Control, and Dynamics*, **19** (1996) 961-963.
- Welch, W.J. "Branch-and-bound search for experimental designs based on D-optimality and other criteria," *Technometrics*, **24** (1982) 41-48.
- Yao, L., W.A. Sethares, and D.C. Kammer, "Sensor Placement for on-orbit modal identification via a genetic algorithm," *AIAA Journal*, **31** (1993) 1922-1928.