Factorial Run Orders by the Simulated Annealing Algorithm

by

Daniel C. Coster
Purdue University

Technical Report #88-52

Department of Statistics
Purdue University

October 1988

◇

Abstract

Factorial Run Orders by the Simulated Annealing Algorithm

Coster and Cheng (1988) introduced a Generalized Foldover Scheme (GFS) for generating run orders of fractional factorial designs which are simultaneously optimal with repect to two specific objective functions, namely, polynomial trend elimination from factor effect estimation and cost minimization in terms of the number of factor level changes. In this report, we study the effectiveness of the Simulated Annealing Algorithm, proposed by Kirkpatrick et. al. (1983), when applied to the factorial run order problem. In addition to assessing the performance of the annealing algorithm on the standard trend elimination and cost minimization problem, for which exact optimal solutions are available using the GFS, we study the usefulness of annealing when alternative design conditions or objective functions are established and exact solutions are (theoretically) unavailable or unknown.

1. Introduction.

Throughout this report, we assume that the reader is familiar with the Generalized Foldover Scheme (GFS) used by Coster and Cheng (1988) to generate run orders of fractional factorial designs for which (i) factor effects of interest are orthogonal to polynomial trends present in each block of the design and (ii) the cost of the run order, measured by the total number of factor level changes, is minimized. Recent references on theoretical approaches to this problem are Coster (1988a, 1988b), Cheng (1988), and Cheng and Jacroux (1987).

Notation and terminology for this report are taken from Coster and Cheng (1988). In particular, the design of interest is denoted by G, a s^{-p} fractional factorial design for n factors each at s levels, with the $N = s^{n-p}$ runs (treatment combinations) of G run in s^r blocks each of size $R = s^{n-p-r}$. Thus p denotes the level of fractionation and r the number of independent effects confounded with blocks. The model for the data in the standard setting is the usual linear model:

$$\mathbf{Y} = (\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)(\beta'_1, \beta'_2, \beta'_3)' + \varepsilon, \tag{1.1}$$

where $\mathbf{X}_1$ is that part of the design matrix corresponding to the factor effects, β_1, of interest, $\mathbf{X}_2$ represents the block effects, and $\mathbf{X}_3$ has k columns representing a polynomial of degree $k \geq 1$ over the (equally spaced) run positions in every block. The usual orthogonal treatment and block structure is assumed for the particular fraction chosen, that is, all the columns of $\mathbf{X}_1$ and $\mathbf{X}_2$ are mutually orthogonal. The *standard* assumptions made when using the GFS to find optimal run orders are:

(A1) all factor level changes are equally expensive;

(A2) the *same* polynomial trend of degree k is present in each block of G;

(A3) the errors ε are uncorrelated (and mean zero) in model (1.1).

Assumption (A2) ensures that $\mathbf{X}_2$ and $\mathbf{X}_3$ are mutually orthogonal also. With these assumptions, the standard design problem may be stated as follows: find run orders of G such that, simultaneously, (D1) $\mathbf{X'}_1\mathbf{X}_3 = \mathbf{0}$, that is, factor effects are orthogonal to trend effects;

(D2) the total number of factor level changes is minimized.

The simulated annealing algorithm is applied to both the standard combinatorial optimiziation problem outlined above and to versions of the run order problem involving modified assumptions and design objectives as follows:

(M1) level changes for different factors are *not* equally expensive;

(M2) different or non-polynomial trends may be present in different blocks of G;

(M3) there are *correlated* errors ε in the form of a first order autoregressive error process in model (1.1).

For example, if modification (M1) is made, the use of an approximate numerical algorithm such as simulated annealing rather than an exact theoretical method for finding optimal run orders may be justified by the following considerations. Let design G be represented by a graph whose nodes are the treatment combinations $\{\mathbf{g}_1, \ldots, \mathbf{g}_N\}$. If all factor level changes are equally expensive, the length of an arc joining nodes i and j, $i \neq j$ is the number of level changes between the two runs $\mathbf{g}_i$ and $\mathbf{g}_j$. Relaxing the assumption of equally expensive level changes, the length of arc (i,j) remains the cost of moving from run $\mathbf{g}_i$ to $\mathbf{g}_j$ but is no longer the number of level changes between these two runs. In this context, the search for a minimum cost run order is equivalent to the search for the shortest path through the graph, that is, a travelling salesman problem where each node of the graph must be visited exactly once by the shortest possible path. The travelling salesman problem is known to be NP-complete so there is no algorithm that guarantees a global optimal solution and has a search time that is a polynomial function of the problem size. An exhaustive search of all possible run orders would take an exponentially long time in terms of the problem size. The annealing algorithm provides a compromise between the practical limitation of available search time and the objective of a global optimal solution.

In Section 2 below, we introduce a version of the simulated annealing algorithm proposed by Lundy (1985). We adopt the same stopping condition as used by Lundy, but look at two different perturbation schemes in the context of our run order search problem. Section 3 provides some numerical evidence of the extent to which we can expect "success" from the annealing algorithm when applied to a small design for which an optimal solution under the standard assumptions (A1) - (A3) is available via the GFS. In Section 4, we look at examples involving the modifications (M1) - (M3) listed above. A short summary of the overall effectiveness of the algorithm is provided in Section 5.

2. Simulated Annealing Algorithm. The simulated annealing algorithm is one member of a class of probabilistic hill-climbing algorithms that has proven to be an effective method for solving combinatorial optimization problems. It differs from pure descent algorithms in that increases in the objective function are permitted with certain probabilities in each iteration. This hill-climbing aspect of the annealing algorithm helps it avoid being trapped in local optima. Simulated annealing also improves on randomizing algorithms - which generate the next state of the combinatorial system randomly and accept all objective function increases - in that increases are accepted with a probability that takes the cost of the move into consideration.

We begin by detailing the structure of the optimization problem as follows. Let S be the set of all possible configurations or states of the system. For our purposes, S is the set of all permissible run orders of the design G. Let $S(i)$ be the set of states that are neighbours of state i in the sense that state $j \varepsilon S(i)$ if it can be reached in one perturbation step of the annealing algorithm from state i. Assume that $i \not\varepsilon S(i)$. In the search for an optimal run order, the perturbation step of the algorithm consists of exchanging the positions of two (or possibly more) runs. Let each state $i \varepsilon S$ have value $C(i)$ with respect to some chosen objective function. We assume that the optimization problem is cast as one of minimization. In each iteration of the algorithm, a transition is made from the current state i to some pertubed state $j \varepsilon S(i)$ with a probability that depends on a sequence of control parameters $\{T_m, m=1,2, \cdots \}$, usually referred to as the temperature control sequence or cooling schedule, and on the difference $C(i)-C(j)$ between objective function values of the current and perturbed states. The cooling schedule $\{T_m\}$ satisifies

$$T_m \geq T_{m+1} \quad \text{and} \quad \lim_{m \to \infty} T_m = 0 . \tag{2.1}$$

The basic structure of the simulated annealing algorithm and the form of the transition probabilities is as stated below.

DEFINITION 1. **Annealing Algorithm:** Beginning in some random state i_0 with objective function value $C(i_0)$, and for some choice of cooling schedule $\{T_m,\ m=1,2,\cdots\}$, if the system is in state i_m at time m, repeat steps (i) – (iii) below until a specified stopping condition is met:

(i) perturb system from state i_m to state $j \in S(i_m)$;

(ii) make the transition from state i_m to state j with probability

$$\mathbf{P}_{i_m j} = \min\left\{1,\ e^{-\{C(j)-C(i_m)\}/T_m}\right\};$$

(iii) set m to $m+1$ and T_m to T_{m+1}.

In Section 3, we consider two alternatives for the perturbation scheme. The first perturbation scheme selects one member of $S(i)$ at random with each state having probability $1/|S(i)|$ of being chosen; the second makes a systematic pass through all permutations of two, (and possibly three or four) runs and/or blocks. The scheme that produces solutions of higher quality is used in Section 4 for examples involving the modified assumptions described by (M1) – (M3). If state $j \in S(i_m)$ is generated in the perturbation step (i), a transition is made with probability one to state j if $C(j) \leq C(i_m)$ and with probability $e^{-\{C(j)-C(i_m)\}/T_m}$ if $C(j) > C(i_m)$.

With these choices of perturbation scheme and transition probabilities, the success of the algorithm in its search for a global optimum depends on the choice of the cooling schedule. Hajek (1986) and Mitra et al. (1985) show that the Markov chain properties of the annealing algorithm guarantee convergence to a global optimum if the temperature control sequence takes the form $T_m = \frac{\gamma}{\log(m + m_0 + 1)}$, where m_0 is any positive constant greater than 1 and γ is a constant that depends on the size of the problem, and subject to the further condition that steps (i) and (ii) of the algorithm are performed infinitely often at each temperature T_m. Since guaranteed convergence would require infinitely many iterations at infinitely many temperatures, these results serve little practical purpose other than to suggest possible choices of cooling schedules and stopping conditions under which the algorithm performs satisfactorily. Following Lundy (1985), we find that the annealing algorithm produces acceptable results for a control sequence given by

Cooling Schedule : $$T_m = \frac{T_0}{1 + m\alpha T_0/U}, \qquad (2.2)$$

where $0 \le \alpha \ll 1$ and U is an upper bound on objective function increases that will be accepted by the algorithm. Notice that $T_{m+1} = \alpha(T_m)T_m$, where $0 \ll \alpha(T_m) < 1$, and that the temperature decreases more and more slowly as the number of iterations increases. The higher the temperature, the greater the probability that an increase in the objective function will be accepted while only small hill climbs are likely to be accepted when the temperature approaches its minimum value.

If a conservative approach is taken, U is set to the maximum possible increase in the objective function so that all hill climbs are allowed in transition scheme (ii) of Definition 1. For a particular problem, if cooling schedule (2.2) terminates the annealing algorithm before a solution of sufficient quality is reached, a slower cooling scedule closer to the log-rate suggested by the convergence results should be used. Furthermore, steps (i) and (ii) of the algorithm should be executed more than once at each temperature.

A stopping condition remains to be chosen. Again following Lundy, we define a parameter $T_f = \eta/\log(\mathbf{V})$, where η is a small positive number and **V** the total number of states of the system, and terminate the algorithm at the first iteration m^* satisfying

$$T_{m^*} \le T_f \,. \tag{2.3}$$

The number of iterations needed to reach T_f is at most

$$U \log(\mathrm{V})/(\alpha\eta)\,. \tag{2.4}$$

When searching for optimal run orders of fractional factorial designs in $B = s^r$ blocks of size $R = s^{n-p-r}$, an upper bound for log(V) is $B \log(R\,!) + \log(B\,!)$.

For $T_0 = U$ and fixed problem size V, the search time of the algorithm is determined by parameters α and η. Good starting values for these parameters are $\alpha = 0.1$ and $\eta = 1$ for an initial search and smaller values of either for a more refined search when the execution time of the algorithm is more accurately determined. In Section 3, we provide examples of the improvement in quality as α and η are decreased.

To apply the annealing algorithm to a bicriteria optimization problem involving distinct objective functions $C_1(i)$ and $C_2(i)$ over states $i \in S$, a scalar objective function

$$C = wC_1 + (1-w)C_2 \tag{2.5}$$

is minimized for each weight w satisifying $0 \le w \le 1$. For a single objective function, set $C_1 \equiv C_2$. In practice, the minimization is done for a finite selection of weights $\{w_1, \ldots, w_{nrep}\}$ for *nrep* repetitions of the annealing algorithm. Evenly spaced weights are recommended unless the experimenter has reason to restrict the range of weights to a subset of the unit interval. When the last repetition is completed, the experimenter implements one of the *nrep* admissible run orders found by the annealing algorithm.

3. Simulated Annealing Applied to Standard Criteria. In this section, we give examples of the annealing algorithm applied to the optimization problem outlined in Section 1 for which the optimality criteria are *linear* trend elimination (so $k = 1$) and minimum cost of level changes, under the standard assumptions (A1) – (A3).

As stated above, two perturbation schemes are compared. In the first scheme, $S(i)$ consists of all states that may be reached from state i by permuting any two, three or four runs (within a single block) or by permuting any two blocks, subject to the restriction that the run order always begins with run **1** and with the principle block. This restriction does not affect the performance of the annealing algorithm. Each state in $S(i)$ has probability $1/|S(i)|$ of being selected as the perturbed state. The second perturbation scheme involves a systematic pass through all permutations of two runs, followed by all permutations of three runs and then all four run permutations. The cycle is repeated until stopping condition (2.3) is met. Under this scheme, neighbourhood $S(i)$ contains a single state that depends on the current permutation.

To provide a measure of the variation in solution quality, for each of five choices of parameters α and η, five repetitions of the annealing algorithm are made at each of eleven equally spaced weights, $\{0.0, 0.1, \ldots, 1.0\}$. In the examples discussed below, the minimum and maximum values of the objective functions C_1 (linear trend elimination) and C_2 (cost of level changes) are given exactly by the results of Coster and Cheng (1988), so each objective function is rescaled to lie between zero and one. Consequently, the value of objective function (2.5) is a number between zero and one also. This allows the quality of solutions at different weights and different parameter settings to be compared directly. Solutions found at weight $w=0$ or $w=1$ are optimal with respect to one design criterion only. At any other weight, a solution with a value of zero in expression (2.5) is globally optimal.

Table I summarizes the performance of the annealing algorithm using the first perturbation scheme described above. For each weight and each choice of parameters α and η, the average and standard deviation of the five values of objective function (2.5) are reported. The number of optimal solutions in each group of five repetitions is indicated by an equal number of asterixes placed after the average. In a similar fashion, Table II provides a performance summary for the second perturbation scheme.

The design used for Tables I and II is a 2^{-2} fraction of a design for $n=6$ factors, each at $s=2$ levels, defined by the relation I = ABCD = ABEF, blocked into 2 blocks of size 8 with confounded effect ACE. It is a main effects only plan, so there are $n=6$ columns in $\mathbf{X}_1$. The minimum and maximum number of level changes for any run order are 44 and 60 repectively, using the results in Coster and Cheng (1988), while the maximum absolute time count between any main effect column of $\mathbf{X}_1$ and the single linear trend column of $\mathbf{X}_3$ lies between 0 and 32 for all run orders. The time count between a main effect column $\mathbf{x}$ of $\mathbf{X}_1$ and $\mathbf{X}_3$ is just $\mathbf{x}'\mathbf{X}_3$, so an optimal run order under the trend elimination criterion is one for which the maximum absolute time count for *all* six columns of $\mathbf{X}_1$ is zero, making main effects estimation orthogonal to trend estimation. The first objective function, C_1, is simply a rescaling of this maximum absolute time count to a number between 0 and 1.

A generator sequence that produces an optimal run order for this design using the GFS is available in Coster (1988b). This example design is labelled as plan 2.6.1 in Table II of Coster (1988b). In the table headings of this section, we use the same label, namely 2.6.1 (for $p=2$, $n=6$ and $s=1$). One immediately interesting aspect of the use of the annealing algorithm is the discovery of an optimal run order for this example that cannot be found by direct application of the GFS of Coster and Cheng. For example, an optimal non-foldover run order for plan 2.6.1 found by annealing is:

1 bce abef ade abcd bdf cdef acf

adf abcdef bcf cd ace ef bde ab

In each cell of Table I, the upper entry is the average of the five final values of objective function (2.5) when the stopping condition is reached, while the lower entry is the standard deviation of these five values. The last line contains the average and standard deviation for all fifty-five solutions at each choice of parameters α and η. The same layout is used for Table II on the following page.

The systematic perturbation scheme found optimal run orders on seven occasions. In comparison, the randomized scheme found two optimal orders. Observing the entries in the last row of each table, the systematic scheme shows consistent improvement in the average quality of the solutions as α and η are decreased and gives satisfactory performance (average value of objective function (2.5) below 0.1) when results are averaged over all weights. In contrast, the

Table I

Averages and Standard Deviations of Five Annealing Algorithm Solutions to Plan 2.6.1 Using Randomized Perturbation Scheme

Weight	$\alpha=.1$ $\eta=1$	$\alpha=.05$ $\eta=1$	$\alpha=.02$ $\eta=1$	$\alpha=.01$ $\eta=1$	$\alpha=.01$ $\eta=.5$
0.0	0.2000 0.1075	0.1625 0.1016	0.1500 0.1016	0.1625 0.1159	0.1125 0.0729
0.1	0.1575 0.0722	0.2050 0.0770	0.1925 0.0357	0.1937 0.0506	0.1950 0.0987
0.2	0.2775 0.0673	0.2275 0.0556	0.2125 0.0559	0.2225 0.0709	0.1800 0.0872
0.3	0.3000 0.0858	0.2512 0.0297	0.2025 0.0876	0.2075 0.0654	0.1638 0.0634
0.4	0.2675 0.0498	0.2400 0.0429	0.1600 0.0464	0.2125 0.0296	0.1800 0.0719
0.5	0.2125 0.0306	0.2062 0.0375	0.1875 0.0442	0.1875 0.0395	0.1312 0.0364
0.6	0.1900 0.0677	0.1950 0.0408	0.1725 0.0348	0.1375 0.0379	0.1450* 0.0801
0.7	0.1913 0.0327	0.1725 0.0419	0.1663 0.0522	0.1525 0.0525	0.1463 0.0600
0.8	0.1500 0.0454	0.1550 0.0551	0.1375 0.0426	0.1075 0.0595	0.0975 0.0330
0.9	0.1587 0.0315	0.0975 0.0620	0.0750 0.0240	0.0863* 0.0538	0.0388 0.0179
1.0	0.1375 0.0468	0.0750 0.0250	0.0875 0.0500	0.0375 0.0306	0.0125 0.0250
	0.2039 0.0818	0.1807 0.0773	0.1585 0.0707	0.1552 0.0818	0.1275 0.0850

Table II

Averages and Standard Deviations of Five Annealing Algorithm Solutions to Plan 2.6.1 Using Systematic Perturbation Scheme

Weight	$\alpha=.1$ $\eta=1$	$\alpha=.05$ $\eta=1$	$\alpha=.02$ $\eta=1$	$\alpha=.01$ $\eta=1$	$\alpha=.01$ $\eta=.5$
0.0	0.1875 0.0685	0.0625 0.0559	0.0375 0.0500	0.0875 0.0396	0.0500 0.0250
0.1	0.1338 0.0475	0.0900 0.0264	0.0850 0.0211	0.0725 0.0350	0.0713 0.0320
0.2	0.1850 0.0184	0.1125 0.0395	0.1275 0.0450	0.0875 0.0565	0.0700 0.0187
0.3	0.1938 0.0495	0.1500 0.0426	0.1350 0.0457	0.1200 0.0329	0.0588** 0.0503
0.4	0.2225 0.0255	0.1575 0.0350	0.1350 0.0521	0.1700 0.0292	0.0675 0.0170
0.5	0.2062 0.0319	0.1750 0.0468	0.1312 0.0415	0.1125 0.0468	0.0875 0.0234
0.6	0.2225 0.0509	0.1475 0.0357	0.1725 0.0930	0.1200 0.0322	0.0775* 0.0436
0.7	0.1625 0.0495	0.1713 0.0400	0.0850 0.0140	0.1513 0.0394	0.1713* 0.0432
0.8	0.1850 0.0339	0.1825 0.0605	0.1300* 0.1077	0.1275 0.0348	0.0650 0.0357
0.9	0.0975 0.0429	0.0988* 0.0550	0.1437 0.0143	0.0586 0.0161	0.0563* 0.0306
1.0	0.1750 0.0919	0.1125 0.0729	0.0875 0.0306	0.0625 0.0559	0.0375 0.0500
	0.1792 0.0616	0.1327 0.0611	0.1155 0.0655	0.1064 0.0519	0.0648 0.0378

improvement in average quality for the randomized scheme is considerably less than that observed for the systematic scheme. Even at the lowest parameter settings, the randomized scheme results in unsatisfactory average quality while the variability in quality remains high as α and η are decreased. Ignoring the rows of the table with weights $w=0$ and $w=1$, the systematic scheme has superior average quality in 38 of the 45 combinations of weight and control parameters. Based on these empirical observations, the systematic scheme is preferred.

An alternative optimization procedure is a randomizing algorithm: in each iteration, the run order is re-randomized and the value of the objective function computed. The best run order found after a chosen number of iterations is implemented. Each of the 550 annealing runs summarized in Tables I and II began with a randomized run order. The average value of objective function (2.5) over all 550 randomized orders was 0.4751 with a standard deviation of 0.1214. The best starting run order had value 0.125 and was one of only two randomized run orders not improved upon by the annealing algorithm, that is, the algorithm began in a local optimum from which it was unable to escape. In this respect, a randomizing algorithm is inferior to the annealing algorithm which, in a comparable number of iterations at parameter settings $(\alpha, \eta) = (0.02, 1)$, found at least one run order of superior quality at all weights. The superior quality of solutions found by the annealing algorithm was even more evident at lower values of α and η, athough the number of iterations until stopping condition (2.3) is met increases to approximately 2100 for the lowest parameter settings used.

The 275 randomized orders used as starting points for the annealing algorithm with the randomized perturbation scheme averaged 0.4808, while the solutions averaged 0.1652, an average improvement of 0.3156 (SD = 0.1264). The average improvement with the systematic scheme was 0.3497 (SD = 0.1303) from an average starting value of 0.4694. Again, the performance of the systematic perturbation scheme is judged to have been superior to that of the randomized scheme.

From Table II, the annealing algorithm found an optimal run order in seven of the two hundred and seventy five starts that used the systematic perturbation scheme. A greater

Table III

Plan 2.6.1 Annealing Solutions with $(\alpha,\eta)=(0.002,0.25)$					
Weight	Perturbations Accepted	Value of C_1	Value of C_2	Objective Function (2.5)	Optimal Run Order Found
0.1	16394	8	44	0.0250	No
0.2	6638	0	44	0.0000	Yes
0.3	7304	0	44	0.0000	Yes
0.4	12967	2	45	0.0625	No
0.5	12107	2	45	0.0625	No
0.6	11329	0	45	0.0250	No
0.7	5101	0	44	0.0000	Yes
0.8	6609	0	44	0.0000	Yes
0.9	3173	0	44	0.0000	Yes

success rate is possible with lower values of the control parameters α and η. For example, optimal orders for plan 2.6.1 were found in five of eighteen starts using settings $(\alpha,\eta)=(0.002,0.25)$ when two starts were made at each weight $w \in \{0.1, \ldots, 0.9\}$. The final scaled value of objective function (2.5), the final unscaled values of criterion functions C_1 and C_2 and the number of perturbations accepted by the transition step (ii) of the algorithm for the better of the two starts at each weight appear in Table III above.

Referring to Table III, for those starts leading to optimal run orders, the number of perturbations accepted was much less than the number accepted for unsuccessful starts, as expected since the algorithm terminated when a known optimal solution was reached. By (2.4), the number of iterations required for those starts that fail to reach an optimal solution is necessarily the maximum allowed by the stopping condition, while the number of perturbations actually accepted is much less than this maximum number of allowed iterations, approximately 38000 in this example. This suggests that, in any practical application of simulated annealing to run order problems of the type studied here, the algorithm should be monitored interactively and stopped after fewer iterations than allowed by (2.4) when the probability that a hill climb will be accepted becomes sufficiently small.

By step (ii) of Definition 1, the probability that a hill climb is accepted depends on the size of the increase in objective function (2.5) and on the temperature. For fixed temperature and weight, this probability is largest for the smallest possible objective function increase. With $T_0 = U = 1.0$ (as is always the case in the examples of this section), $T_m = 1/(1+m\alpha)$. Let the minimum possible objective function increase under the systematic perturbation scheme be $\delta_{\min}$. For this increase to be accepted with probability at most q, $0 < q < 1$, we require $e^{-(1+m\alpha)\delta_{\min}} \le q$, that is, the iteration number must satisify:

$$m \ge \frac{-\log_e q/\delta_{\min} - 1}{\alpha}. \tag{3.1}$$

For plan 2.6.1, with α and η set to the values used for the annealing runs summarized in Table III and at weight $w = 2/3$, $\delta_{\min} = 1/48$. If $q = 0.05$, then at least 71,398 iterations are required for all subsequent hill climbs to be accepted with probability at most q. This is almost twice the number of iterations allowed by stopping condition (2.3). Thus, requiring the acceptance probability of the smallest objective function increase to be bounded above may be too conservative. Instead, we suggest a modified stopping condition that requires the probability of the *average* hill climb, δ_{avg} say, to be at most q and replace stopping condition condition (3.1) by

$$m \ge \frac{-\log_e q/\delta_{avg} - 1}{\alpha}. \tag{3.2}$$

Computing δ_{avg} for an arbitrary perturbation scheme, objective function and weight is not generally feasible. To overcome this difficulty, we recommend using an iterative procedure that evaluates the average of all hill climbs attempted by the annealing algorithm and substitutes this value for δ_{avg} in (3.2). With this modification, the algorithm is terminated when the number of iterations exceeds the value given by (3.2), subject to the requirement that at least half the number of iterations allowed by stopping condition (3.1) are executed. (This last requirement ensures that the number of iterations is large enough for satisfactory quality to be reliably attained.) The choice of limiting probability q must be made by the experimenter. In an interactive environment, each time this modified stopping condition is met the experimenter may examine the current best solution and decide whether it is of sufficient quality. If not, a smaller value of q may be entered and the algorithm continues until either the modified version of (3.2) is again met or stopping condition (2.3) is achieved.

For example, with $(\alpha, \eta) = (1, 0.1)$, the average hill climb attempted for plan 2.6.1 over 45 starts at weights other than 0 and 1 came to 0.0895 (SD = 0.0140). A similar average was observed for other parameter settings. With $q = 0.05$ as before, the number of iterations required until condition (3.2) is met is approximately $\frac{32.5}{\alpha}$. When α is large and hence total execution time small, condition (2.3) will usually be met before (3.2); for $\alpha\,(\leq 0.002)$ small, condition (3.2) may be satisfied before (2.3) and the algorithm terminated earlier without seriously affecting the quality of the solution. In the particular case that $(\alpha, \eta) = (0.002, 0.5)$, (3.2) requires approximately 16,235 iterations while (2.3) needs in excess of 21,900 steps. From annealing runs made at each of fifty-one weights evenly spaced between zero and one and α and η as stated above, the iteration in which the best run order was found was, without exception, less than 14,200 and an average of 6884 iterations were required to reach the best solutions for each start. Thus, the use of stopping condition (3.2) would have been appropriate for these annealing runs. We note that with the stated parameter settings, twelve of the fifty-one starts led to optimal run orders, an excellent success rate.

Plan 2.6.1 is a small design. As the design size grows, the annealing algorithm is less successful at finding globally optimal run orders, while the execution time increases at a rate proportional to log(V) in expression (2.4) for fixed settings of parameters α and η. For example, with $(\alpha, \eta) = (0.002, 0.5)$, and for the plan labelled as 2.7.2 in Table II of Coster (1988b), a 2^{-2} fraction of a design for $n = 7$ factors blocked into 4 blocks of size 8, run orders were found at each of six equally spaced weights. Table IV below summarizes the results. Although the quality of the annealing algorithm solutions is satisfactory for all but weight $w = 0.6$, none of the run orders is optimal. Note that the minimum value of objective function C_2 for this

Table IV

Plan 2.7.2 Annealing Solutions with $(\alpha,\eta)=(0.002,0.5)$				
Weight	Starting Value of Function (2.5)	Final Value of Function (2.5)	Value of C_1	Value of C_2
0.0	0.5000	0.0667	32	98
0.2	0.4897	0.0596	2	98
0.4	0.4800	0.0750	4	99
0.6	0.4583	0.1254	2	110
0.8	0.4367	0.0767	0	117
1.0	0.4688	0.0000	0	124

design is 94 factor level changes. With high weight on objective function C_1, the number of factor level changes in the solutions is well above this minimum. However, at the extreme weights of $w=0$ and $w=1$, the solutions are quite satisfactory (optimal in the case $w=1$).

4. Modified Design Criteria. In this section, we present examples of run orders found by the annealing algorithm for each of the modified design criteria described in (M1) – (M3) of Section 1. Stopping condition (2.3) is used in all the examples of this section.

We begin with an example in which the level changes for different factors are not equally expensive. We again use plan 2.6.1. Suppose that level changes of factor **d** have unit cost and all other factor level changes are free (that is, have zero cost). A linear trend free minimum cost run order is wanted, as usual. A factor must make at least two level changes if it is to be linear trend free. So the unscaled cost function has a minimum value of two and a maximum of fifteen. An optimal run order must have all six factors linear trend free while factor **d** must change levels twice, once after four runs and a second time after twelve runs. For two choices of parameters α and η, five randomized starts were made at each of eleven equally spaced weights. The values of the unscaled objective functions C_1 and C_2 and scaled objective function C given by (2.5) for the best order found at each weight are presented in Table V below. With the lower parameter settings, the bicriteria search successfully found an optimal run order for five of the nine weights not equal to either zero or one.

Table V

Annealing Solutions for Plan 2.6.1 for Modified Criterion M1 With All Factors Except d Having Zero Cost						
	$\alpha=0.1, \eta=1$			$\alpha=0.01, \eta=0.5$		
Weight	Value of C_1	Value of C_2	Function (2.5)	Value of C_1	Value of C_2	Function (2.5)
0.0	16	2	0.0000	16	2	0.0000
0.1	8	3	0.1000	0	2	0.0000
0.2	8	3	0.1167	2	2	0.0125
0.3	4	2	0.0375	0	2	0.0000
0.4	6	2	0.0750	8	3	0.1462
0.5	8	3	0.1635	0	2	0.0000
0.6	6	2	0.1125	2	2	0.0375
0.7	4	6	0.1798	0	4	0.0462
0.8	4	2	0.1000	2	2	0.0500
0.9	2	8	0.1024	0	2	0.0000
1.0	4	8	0.1250	0	4	0.0000

Modifying the cost structure to a less trivial form, suppose that level changes of factors **d, e, f** are free while level changes of factors **a, b, c** cost 1, 2 and 3 respectively. With $(\alpha, \eta) = (0.01, 0.5)$, run orders with a minimum cost of twenty-three level changes were found. (The maximum number of level changes came to seventy-six). The best bicriteria run order had all factors linear trend free and required twenty-four level changes. This run order is:

1 bdf abef ade acf cdef bce abcd

abcdef ace cd bcf bde ab adf ef.

Modification (M2) involves using a non-polynomial trend. Possibilities include piece-wise linear trends, polynomial trends of different degrees in each block and cyclic trends. An example of the latter is a sine wave based on run position in each block. For example, with design $G = 2_0^{5-1}$ for five factors in one block of sixteen runs defined by $I = ABCDE$, the trend in run position i of $\mathbf{X}_3$ is now equal to $\sin(2\pi i / 16)$. The second design objective, C_2, remains

Table VI

Annealing Solutions for Plan 1.5.0 for Modified Criterion M2 With Trend Given by $\sin(\frac{2\pi i}{16})$

	$\alpha=0.01, \eta=1$			$\alpha=0.002, \eta=0.5$		
Weight	Value of C_1	Value of C_2	Function (2.5)	Value of C_1	Value of C_2	Function (2.5)
0.0	0.2697	30	0.0000	0.1580	30	0.0000
0.1	0.0070	30	0.0007	0.0156	30	0.0016
0.2	0.0267	30	0.0053	0.0046	30	0.0009
0.3	0.0156	30	0.0047	0.0054	30	0.0016
0.4	0.0313	30	0.0125	0.0039	30	0.0015
0.5	0.0642	30	0.0321	0.0156	30	0.0078
0.6	0.0313	30	0.0186	0.0009	30	0.0005
0.7	0.0321	30	0.0225	0.0092	30	0.0064
0.8	0.0217	32	0.0307	0.0017	30	0.0014
0.9	0.0196	34	0.0245	0.0156	30	0.0141
1.0	0.0033	45	0.0033	0.0008	45	0.0008

the total number of factor level changes. Unscaled objective function C_1 corresponding to the criterion of trend elimination is equal to the maximum of $\mathbf{x}'\mathbf{X}_3$ over the columns $\mathbf{x}$ of $\mathbf{X}_1$. Equivalently, we seek a run order that minimizes the maximum squared correlation coefficient between any factor effect and the trend effect and has minimum cost of level changes. Table VI above presents a summary of the objective function values for the best run orders found at each of eleven equally spaced weights for two choices of parameters α and η. Note that C_1 values are the squared correlation coefficients.

If the maximum squared correlation coefficient is less than 0.01, the maximum correlation between any factor effect column and the sine trend is less than 0.1. The average over randomized starting orders of the maximum correlation coefficient between factor effects and the sine trend was 0.491. Thus, some of the run orders found by the annealing algorithm are considerably better than any randomized run order with respect to both objective functions. For these design criteria, a run order is judged to be of satisfactory quality if the value of objective function (2.5) is less than 0.01 at each weight.

The minimum value of criterion function C_2 is 30 factor level changes. Eighteen of the twenty-two run orders achieve this minimum. The run order corresponding to the solution at weight $w = 0.6$ for the lower parameter settings is

1 cd ce acde abde ab bd be ae ac ad de bcde abce bc abcd

Modification (M3) refers to a correlated errors model in which successive random errors are correlated according to a first order autoregression with correlation λ. Let the variance-covariance matrix for error process ε in linear model (1.1) be denoted by $\mathbf{W}$. We assume that ε has mean zero. The inverse of $\mathbf{W}$ is given by

$$\mathbf{W}^{-1} = \begin{bmatrix} 1 & -\lambda & 0 & . & . & 0 \\ -\lambda & 1+\lambda^2 & -\lambda & 0 & . & . \\ 0 & -\lambda & 1+\lambda^2 & . & 0 & . \\ . & 0 & -\lambda & . & -\lambda & 0 \\ . & . & 0 & . & 1+\lambda^2 & -\lambda \\ 0 & . & . & 0 & -\lambda & 1 \end{bmatrix} . \tag{4.1}$$

Linear model (1.1) may be re-written as

$$\mathbf{Y} = (\mathbf{X}_1, \mathbf{X}_2) \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \varepsilon ,$$

where $\mathbf{X}_1$ is the design matrix for factor effects and $\mathbf{X}_2$ represents blocking effects. We no longer assume that any trend is influencing the observations. The information matrix for the factor effects is

$$\mathbf{M}_{\beta_1} = \mathbf{X}'_1 \left[\mathbf{W}^{-1} - \frac{1-\lambda}{\kappa-(\kappa-2)\lambda} \, \mathbf{diag} \left(\Lambda, \ldots, \Lambda \right) \right] \mathbf{X}_1 , \tag{4.2}$$

where κ is the block size and Λ is a $\kappa \times \kappa$ matrix of the form

$$\Lambda = \begin{bmatrix} 1 & 1-\lambda & . & 1-\lambda & 1 \\ 1-\lambda & (1-\lambda)^2 & . & (1-\lambda)^2 & 1-\lambda \\ . & . & . & . & . \\ 1-\lambda & (1-\lambda)^2 & . & (1-\lambda)^2 & 1-\lambda \\ 1 & 1-\lambda & . & 1-\lambda & 1 \end{bmatrix}.$$

We assume that correlation coefficient λ is so chosen that variance-covariance matrix **W** is positive definite. Note that the choice of a first order autoregressive error process is for convenience. Any variance-covariance matrix **W** is acceptable, but for an arbitrary error process the closed form inverse (4.1) must, in general, be replaced by a numerically computed inverse.

Frequently used optimality criteria are the D-, A- and E- optimality criteria. A run order is D-optimal if it minimizes the product of the eigenvalues of the inverse of information matrix $\mathbf{M}_{\beta_1}$; for A-optimality, the sum of the eigenvalues must be minimized, while E-optimality is achieved if the maximum eigenvalue is minimized. Other functions of the eigenvalues may also be used. An equivalent form of the D-criterion is that the determinant of $\mathbf{M}_{\beta_1}$ is maximized.

Table VII summarizes the results of annealing runs made for plan 2.6.1 with two different choices of correlation λ at parameter settings $(\alpha, \eta) = (0.01, 1.0)$ and with one objective function, the D-criterion. Because no closed form expression is known for the determinant of $\mathbf{M}_{\beta_1}$ when λ is non-zero, the eigenvalues were re-evaluated for each run order. Furthermore, neither the maximum nor the minimum possible determinant is known for arbitrary λ. Hence, without an exhaustive search, the optimal D-criterion value is unknown. Instead, we trust that the success of the annealing algorithm observed in the examples of Section 3 is repeated here and that the best run order found is close to D-optimal. Since only one objective function is involved, all weight choices are equivallent and it is sufficient to run the annealing algorithm at weight $w = 0$. A run order with determinant 18.031 when $\lambda = 0.2$ is

1 abef abcd cdef bce acf bdf ade

ace adf bcf bde ef abcdef ab cd

This run order requires the maximum possible number of level changes within each block. In

Table VII

Annealing Results for Plan 2.6.1 Under D-Criterion With Correlated Errors Model				
	$\lambda=0.1$		$\lambda=0.2$	
Trail	Initial Determinant	Final Determinant	Initial Determinant	Final Determinant
1	16.610	16.832	16.846	17.992
2	16.258	16.878	17.080	18.031
3	16.449	16.761	17.017	18.031
4	16.406	16.893	16.981	18.031
5	16.366	16.832	17.708	18.031
6	16.254	16.907	16.614	17.877
7	16.193	16.835	17.071	18.031
8	16.534	16.896	17.087	18.006
9	16.463	16.900	17.004	18.012
10	16.474	16.816	17.058	18.006

general, it is observed that run orders which perform well with respect to the D-criterion have a large number of level changes for positive correlation λ. Consequently, the search for D-optimal run orders may be aided by searching for maximum level change orders since this latter criterion function is more readily evaluated at each state visited by the annealing algorithm. In a bicriteria search involving the D-criterion with an autoregressive error process, the cost function used in the earlier chapters should not be used since the two criteria will be in conflict.

5. Conclusions. The annealing algorithm has proven to be an effective method for finding near-optimal run orders for a variety of scalar and vector optimization problems. In general, any objective function whose value depends on the run order of the design may be used. Although not demonstrated here, the design itself need not be restricted to the class of fractional factorial designs but may take an almost arbitrary form. We add one word of caution: in a bicriteria optimization problem, care should be taken to avoid competing objective functions.

The greatest disadvantage of the annealing algorithm is the amount of execution time used for each run at the settings of parameters α and η required for high quality solutions. With the standard design criteria used in Section 3, an average of 3.8 seconds of cpu time on a Vax 11/750 was required for each annealing run at $(\alpha, \eta) = (0.1, 1)$ for plan 2.6.1. With α and η lowered to 0.002 and 0.5 respectively, run time averaged two minutes per start. Plan 2.7.2 required almost twenty minutes per run at these latter parameter settings. With objective functions such as the D-criterion, for which the eigenvalues of the information matrix must be computed for each state visited by the annealing algorithm, average run time is even greater. Our simplest and strongest recommendation for the further application of simulated annealing to design problems of the type considered here is: use a super computer.

REFERENCES

Cheng, C-S. and Jacroux, M. (1987). On the construction of trend-free run orders of two-level factorial designs. *JASA*, to appear.

Coster, D.C. (1988a). Trend-free run orders of mixed level fractional factorial designs. Purdue University Statistics Department Technical Report #88-31. Submitted to *Ann. Statist.*, June, 1988.

Coster, D.C. (1988b). Tables of minimum-cost, linear trend-free run sequences of two- and three-level fractional factorial designs. Purdue University Statistics Department Technical Report #88-41.

Coster, D.C. and Cheng, C-S. (1988). Minimum cost trend free run orders of fractional factorial designs. *Ann. Statist.*, **16**, 1188-1205.

Hajek, B. (1986). Cooling schedules for optimal annealing. Preprint.

Kirkpatrick, S., Gelatt, Jr., C.D. and Vecchi, M.P. (1983). Optimization by simulated annealing. *Science*, **220**, 671-680.

Lundy, M. (1985). Applications of the annealing algorithm to combinatorial problems in statistics. *Biometrika*, **72**, 191-198.

Mitra, D., Romeo, F. and Sangiovanni-Vincentelli, A. (1985). Convergence and finite time behaviour of simulated annealing. *Proceedings of the* 24th Conference on Decision and Control, Ft. Lauderdale, FL. 761-767.

National Bureau of Standards Applied Mathematics Series 48, (1957). *Fractional Factorial Experiment Designs for Factors at Two Levels*. U.S. Department of Commerce.

National Bureau of Standards Applied Mathematics Series 54, (1959). *Fractional Factorial Experiment Designs for Factors at Three Levels*. U.S. Department of Commerce.