IIE Transactions
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/uiie20

Solving Quadratic Assignment Problems by 'Simulated Annealing'

Mickey R. Wilhelm Ph.D., P.E. [a] & Thomas L. Ward Ph.D., P.E. [a]
[a] Department of Industrial Engineering , University of Louisville , Louisville, Kentucky, 40292
Published online: 09 Jul 2007.

To cite this article: Mickey R. Wilhelm Ph.D., P.E. & Thomas L. Ward Ph.D., P.E. (1987) Solving Quadratic Assignment Problems by 'Simulated Annealing', IIE Transactions, 19:1, 107-119, DOI: 10.1080/07408178708975376

To link to this article: http://dx.doi.org/10.1080/07408178708975376

Solving Quadratic Assignment Problems by 'Simulated Annealing'

MICKEY R. WILHELM, Ph.D., P.E.
SENIOR MEMBER, IIE

THOMAS L. WARD, Ph.D., P.E.
SENIOR MEMBER, IIE
Department of Industrial Engineering
University of Louisville
Louisville, Kentucky 40292

Abstract: Recently, an interesting analogy between problems in combinatorial optimization and statistical mechanics has been developed and has proven useful in solving certain traditional optimization problems such as computer design, partitioning, component placement, wiring, and traveling salesman problems. The analogy has resulted in a methodology, termed "simulated annealing," which, in the process of iterating to an optimum, uses Monte Carlo sampling to occasionally accept solutions to discrete optimization problems which increase rather than decrease the objective function value. This process is counter to the normal 'steepest-descent' algorithmic approach. However, it is argued in the analogy that by taking such controlled uphill steps, the optimizing algorithm need not get "stuck" on inferior solutions.

This paper presents an application of the simulated annealing method to solve the quadratic assignment problem (QAP). Performance is tested on a set of "standard" problems, as well as some newly generated larger problems ($n = 50$ and $n = 100$). The results are compared to those from other traditional heuristics, e.g., CRAFT, biased sampling, and a revised Hillier procedure. It is shown that under certain conditions simulated annealing can yield higher quality (lower cost) solutions at comparable CPU times. However, the simulated annealing algorithm is sensitive to a number of parameters, some of whose effects are investigated and reported herein through the analysis of an experimental design.

■ The quadratic assignment problem (QAP) arises in facilities location and layout problems when facilities are to be assigned to sites and when there are interactions between the facilities that depend upon their location (e.g., [1]). For example, if c_{ikjh} is the "cost" of locating facility i at site k and facility j at site h (usually defined as the product of the flow between facilities i and j and the distance between sites k and h), then the formulation of the QAP has been written as the minisum problem P1: below [2]:

$$\text{P1:} \quad \text{Minimize } f(x) = (1/2) \sum_{i=1}^{n} \sum_{k=1}^{n} \sum_{j=1}^{n} \sum_{h=1}^{n} c_{ikjh}\, x_{ik}\, x_{jh}$$

$$\text{Subject to: } \sum_{i=1}^{n} x_{ik} = 1,\ k = 1, 2, \ldots, n$$

$$\sum_{k=1}^{n} x_{ik} = 1,\ i = 1, 2, \ldots, n$$

$$x_{ik} = 0 \text{ or } 1, \text{ for all } i, k.$$

Received June 1985; revised November 1985. Handled by the Facilities Design/Location Analysis.

In this formulation, the first two constraint equation sets ensure that exactly n facilities are to be assigned to exactly n sites. Such a QAP is said to have a **problem size** n. The last constraint implies that each variable, x_{ik}, for all i, k, is one if facility i is located at site k, otherwise it is zero.

As a matter of computational convenience, a QAP assignment (or solution) can be represented by the n-vector:

$$\mathbf{a} = [a(1), a(2), \ldots, a(n)],$$

where the element $a(i)$ in the assignment vector **a** denotes the number of the site to which facility i has been assigned [2]. Therefore, the assignment vector **a** provides the subscripts of the variables x_{ik}, where $k = a(i)$, in problem P1 which should be set equal to one in order to provide a complete, minimum cost assignment of facilities to sites. For example, if $n = 4$ and the final assignment vector is $\mathbf{a} = [2, 4, 1, 3]$, then **a** implies that $x_{12} = x_{24} = x_{31} = x_{43} = 1$, which means that facilities 1, 2, 3, and 4 should be assigned to sites 2, 4, 1, and 3, respectively, and all other $x_{ik} = 0$. It is easy to test a particular assignment vector **a** to see whether it satisfies the constraints given in problem P1. It is also easy (at least con-

0470-817X/85/0600-0182/$2.00/0

ceptually) to totally enumerate all possible assignments of facilities to sites.

The QAP is a powerful model with a large number of potential real world applications [2, 3]. For example, QAP applications include the assignment of a number of plants, warehouses or indivisible operations to a number of geographical sites, facility layout, the layout of indicators and controls on a control panel, the layout of electronic components on a printed circuit board, assigning storage space on computer disc storage devices, sequencing work through a production facility, routing (traveling salesman) problems and so on [4].

The extent to which the QAP formulation can fulfill its application potential depends on the existence of computationally feasible and efficient solution procedures. The computational efficiency of a QAP solution technique is often measured by the functional dependence of solution time on problem size. However, Sahni and Gonzalez [5] have shown that the QAP is **NP complete.** It has been conjectured [6] that there is no solution technique which has a polynomially bounded solution time for problems of this class. In particular, Foulds [7] has suggested that QAP solution times are likely to be an exponential function of the problem size, n.

Although the literature reports numerous algorithms capable of determining exact (optimal) solutions to quadratic assignment problems [e.g., 8], they are currently only computationally feasible for very small problems ($n \leq 17$), or for special cases of the QAP [3, 7]. It is not surprising, therefore, that much research effort has been devoted to devising heuristic solution procedures which run in "reasonable" computer time and yield solutions of "acceptable", if not optimal, quality.

Heuristic approaches to quadratic assignment problems have been generally classified as either 'construction' procedures or 'improvement' procedures [9]. Construction procedures attempt to build a solution from the null solution by making successive assignments of facilities to sites; whereas, improvement procedures start with a complete initial assignment of facilities to sites and attempt to iteratively improve upon it.

The purpose of this paper is to present the results of experimental applications of a new heuristic approach, which has been termed 'simulated annealing', to the solution of quadratic assignment problems.

Previous Work

Koopmans and Beckmann [10] first formulated the QAP in 1957 and applied optimization procedures to solve it. Because of the computational difficulties initially encountered in solving QAP's, numerous other researchers have attempted the development of exact solution algorithms, e.g., Lawler [11], Gavett and Plyter [12], Graves and Whinston [13], Pierce and Crowston [4], Bazaraa and Elshafei [14], and Bazaraa and Sherali [15]. Most of the solution approaches taken by those researchers are based on a branch and bound strategy. However, as previously stated, the exact solution approaches have not been able to overcome the intractability of the QAP, traceable to the fact that it is **NP complete,** and thus are limited in application to small problems ($n \leq 17$).

As previously discussed, heuristics (of both the construction and improvement types) have been introduced to solve larger problems than those solvable by exact approaches. Construction type heuristics, e.g., Hanan and Kurtzberg [8], Burkard and Stratmann [16] and Liggett [17], do not, in general, yield solutions that are near optimal. Improvement type heuristics which start with an initial solution and, through facility locational exchanges, attempt to reduce the cost of assignment have been found to yield superior solutions, e.g., Hillier [18], Hillier and Connors [19], Nugent, Vollman and Ruml [9], Wilhelm, et. al. [1], and Picone and Wilhelm [20].

Two of the best improvement procedures, by general consensus, are that employed by CRAFT (Computerized Relative Allocation of Facilities Technique) [21], and that employed in a variation on CRAFT by Nugent, Vollman and Ruml [9], known as the Biased Sampling Technique. The attribute of biased sampling that gives rise to its ability to yield better solutions than CRAFT is the technique used to select facilities for location interchange in the improvement iterations. Biased sampling assigns to the facility pairwise exchange with the greatest potential cost reduction a probability of selection less than one, thereby allowing the possible selection of any pairwise exchange which leads to a cost reduction. This capability circumvents the strictly steepest-descent algorithmic approach, used in original versions of CRAFT as well as similarly based algorithms contributed by other researchers, which could get "stuck" on inferior solutions. Nugent, Vollman and Ruml, concluded that biased sampling produced better solutions than both CRAFT [21] and the Hillier and Connors procedure [19], but at relatively higher computational costs.

Typical of an approach taken by a number of researchers recently is that of Picone and Wilhelm [20] which extends the methods of Armour and Buffa [21], Hillier [18] and Hillier and Connors [19] by expanding the 'neighborhood' searched in making λ-wise interchanges of facility locations. Termed a Revised Hillier procedure, they investigate 3- and/or 4-way perturbations at each iteration of the Hillier procedure with the goal of improving the quality of the solution (minimizing cost) while controlling (minimizing) run time. They report yields of solutions of higher quality than those obtained by CRAFT and the original Hillier procedure, while requiring only modest increases in computer run time.

Simulated Annealing

An improvement procedure bearing some similarity to the biased sampling approach [9] has been termed **simulated annealing.** This interesting name has been proffered because of a useful analogy between statistical mechanics (the behavior of systems with many degrees of freedom in thermal equi-

librium at a finite temperature) and combinatorial optimization (exemplified by problems like the QAP) that was recently proposed [22].

"Statistical mechanics is the central discipline of condensed matter physics, a body of methods for analyzing aggregate properties of the large numbers of atoms to be found in samples of liquid or solid matter" [22]. The reason for the term "statistical" mechanics is that because of the large number of atoms (approximately 10^{23} per cubic centimeter), their aggregate behavior is characterized by observation of random fluctuations about the most probable (average) behavior of the system at a specific temperature [23]. A fundamental question in statistical mechanics concerns what happens to the system in the limit as the temperature approaches the ground state, e.g., whether the atoms remain fluid or solidify.

The low temperature is referred to as the "ground state" and is the lowest energy state of the system [24]. Ground states, and the atomic configurations of macroscopic bodies resembling them, are extremely rare at elevated temperatures, but are predominate properties at low temperatures. However, low temperatures alone are not sufficient for finding the ground states of matter [22]. In practice, experiments designed to find the ground states are performed by careful **annealing,** i.e., by first melting the system at a high temperature, then lowering the temperature slowly (according to an annealing schedule), finally spending a long time at temperatures in the vicinity of the freezing, or solidification, point. The amount of time spent at each temperature during the annealing process must be sufficiently long to allow the system to reach thermal equilibrium (steady state). If care is not taken in adhering to the annealing schedule (the combination of a set of temperatures and length of time to maintain the system at each temperature), undesirable random fluctuations may be frozen into the material thereby making the attainment of the ground state impossible.

Metropolis, et. al. [25], devised a simple Monte Carlo approach to simulate the behavior of a collection of atoms in achieving thermal equilibrium at a given temperature. The procedure may be stated as [25, 23, 24]:

> Given a configuration of the elements of the system, randomly displace the elements, one at a time, by a small amount and calculate the resulting change in energy, ΔE. If $\Delta E < 0$ then accept the displacement and use the resulting configuration as the starting point for the next iteration. If $\Delta E \geqslant 0$, then the displacement is accepted with probability $P(\Delta E) = \exp(-\Delta E/k_b T)$, where T is the temperature and k_b is Boltzmann's constant, which is not required when applying the Metropolis algorithm to combinatorial problems.

The Monte Carlo sampling aspect of this simulation is incorporated by comparing $P(\Delta E)$ with a random variable drawn from a uniform distribution on the interval (0, 1). Notice that there is a probability of $P(\Delta E)$ that an atomic rearrangement which **increases** the energy of the system will be accepted. This process is continued until equilibrium is achieved, then the temperature is lowered according to the annealing schedule and the procedure is repeated until the system freezes. At each temperature, the annealing schedule **must** allow the simulation to proceed long enough for the system to reach steady state.

The analogy which suggests that the simulated annealing approach may be effective and efficient in solving combinatorial optimization problems like the QAP should now be apparent. If each feasible assignment of facilities to sites in a QAP is viewed as a configuration of atoms in the mechanical system, and the value of the objective cost function in P1 is viewed as the energy of the system, then determining the least cost assignment of facilities to sites is analogous to finding the arrangement of atoms in the mechanical system which results in the lowest energy (ground) state.

Kirkpatrick, Gelatt and Vecchi [22], after drawing attention to the analogy between statistical mechanics and combinatorial optimization problems, applied the simulated annealing approach to the solution of a circuit board layout and wiring problem, and to the traveling salesman problem (TSP). They concluded that good quality solutions to both of these problem types are attainable with annealing schedules for which the amount of computational effort scales as n, or as a small power of n. Encouraged by these results, they hypothesized the fruitfulness of a wide application of this heuristic technique to other combinatorial optimization problems.

Golden and Skiscim [23, 24] investigated the application of simulated annealing to traveling salesman problems and to p-median network location problems. They developed an algorithm and conducted experiments regarding the effects of various parameters, e.g., annealing schedule, tests for steady state at each temperature, stopping rules, etc., on solution time and quality. They found the simulated annealing procedure to be very sensitive to a number of control parameters and stopping criteria. They were unable to determine an implementation strategy which performs consistently well on the TSP. Further, their experiments with simulated annealing in solving p-median problems were also discouraging: a substantial increase in computation time to obtain solutions only slightly better than those obtained from the use of simple node interchange heuristics.

However, they pointed out that TSP's have been studied for many years and it may be unfair to expect a new procedure like simulated annealing to perform as well as the best known TSP algorithms without more extensive evaluation. Therefore, they recommended more research to learn more about the potential benefits of using simulated annealing to solve TSP and other combinatorial optimization problems.

Bonomi and Lutton [26] also studied the application of a simulated annealing approach in solving TSP's. However, in contrast to Golden and Skiscim, their conclusions regarding the usefulness of simulated annealing were quite positive. They noted, "The resulting Metropolis algorithm seems to be a powerful numerical tool for solving large difficult problems provided it is supplemented by an efficient trial configuration selection mechanism favouring the choice of 'important' configurations."

Burkard and Rendl [27] presented some computational results from the application of simulated annealing to QAP's. They compared the simulated annealing results with those from other fast, sophisticated heuristics for solving QAP's which they had previously developed and noted that it performed well. They concluded that the simulated annealing procedure should be considered an efficient algorithm for solving QAP's with respect to both solution quality and computational time required.

However, Burkard and Rendl did not provide very many details regarding the settings of the control parameters (or annealing schedule) used in their algorithm aside from the advice that computation time and efficiency may be controlled by "setting the 'cooling parameter' less than but close to one, or, independently setting the number of iterations sufficiently high." This lack of information makes it very difficult, if not impossible, to duplicate their results.

In the following sections, a simulated annealing algorithm for solving QAP's, which performs better than that in [27], is presented and evaluated experimentally.

The Simulated Annealing Algorithm

This section outlines the general algorithm and the features of the implementation of a simulated annealing heuristic for solving quadratic assignment problems.

Nomenclature

S = $\{t_1, t_2, \ldots, t_r\}$, a set of annealing schedule temperatures, where $t_i = (10)(0.9)^{i-1}$ $\forall$ $i = 1, 2, \ldots, r$ therefore; $t_1 > t_2 > t_3 > \ldots > t_{r-1} > t_r$.

e = Epoch interval – an *a priori* number of accepted pairwise interchanges of facility locations at temperature t_i, where e is an integer.

$\bar{f}_e = \left(\sum_{j=1}^{e} f_j(\mathbf{a})\right)/e$ where e is defined above and where $\bar{f}_e(\mathbf{a})$ = mean of the total costs $f_j(\mathbf{a})$ for each of the accepted facility locational assignment interchanges $j = 1, 2, \ldots, e$, during the current epoch interval.

$\bar{f}'_e$ = Value of the grand mean of the $\bar{f}_e$ for all previous epochs at a specific temperature, t_i.

ϵ = an error constant used to determine whether the system is in equilibrium at a specific temperature, t_i.

J = a constant representing the total number of interchanges *attempted* at the current temperature t_i.

N' = a constant which when multiplied by the problem size, n, defines the maximum value of J.

I'_p = number of *accepted* interchanged positions for facility p, $\forall p = 1, 2, \ldots, n$, at the current temperature t_i.

N = a constant defining the minimum required value of I'_p per epoch.

I_c = iteration counter.

$\mathbf{a}^0$ = $[a(1), a(2), \ldots, a(n)]$, the initial assignment vector of facilities to locations.

$\mathbf{a}^*$ = "current best" least cost assignment of facilities to sites.

$\mathbf{a}$ = the current assignment of facilities to locations at any iteration of the algorithm.

Adapting the Metropolis, et al., [25] procedure stated previously, we may outline a general algorithm for application of simulated annealing in solving the QAP.

General Algorithm

For a given annealing schedule of temperatures, $S = \{t_1, t_2, \ldots, t_r\}$,

Step 1 Determine an initial assignment (configuration), $\mathbf{a}^0$, of facilities to sites, perhaps randomly. Set $\mathbf{a} = \mathbf{a}^0$ and $i = 1$.

Step 2 Compute the total cost (energy) of the current assignment, $f(\mathbf{a})$, from the objective function in problem P1.

Step 3 a. Randomly select two facilities (elements), u and v, for location exchange, resulting in an assignment $\mathbf{a}'$. Evaluate the consequent change in total cost (energy), $\Delta f = f(\mathbf{a}) - f(\mathbf{a}')$. If $\Delta f \leq 0$, go to step 3 b; otherwise, go to step 3 c.

b. Select a random variable $x \sim U(0, 1)$. If

$$x < P(\Delta f) \equiv \exp(-\Delta f/t_i),$$

then go to step 3 c; otherwise go to step 3 a.

c. Accept the pairwise exchange, set $\mathbf{a} = \mathbf{a}'$ and $f(\mathbf{a}) = f(\mathbf{a}')$, then go to step 4.

Step 4 If equilibrium has *not* been reached at temperature t_i, i.e., if the system has *not* dwelled at the current temperature for a sufficient length of time, go to step 3 a. Otherwise, set $i = i + 1$, which amounts to reducing the temperature to the next one in the annealing schedule, and go to step 5.

Step 5 If all temperatures in the annealing schedule have been used, i.e., if $i > r$, **STOP.** Otherwise, go to step 3 a.

Now, as was pointed out by Kirkpatrick, Gelatt and Vecchi [22], two of the most important issues in implementing the general simulated annealing approach are those of the annealing schedule, S, and the test for equilibrium at each temperature in the foregoing algorithm. With respect to the temperatures in the annealing schedule, tests were conducted

to determine how high 'temperatures' should be set in order to ensure 'melting' of the system. Then various schema for determining the individual temperatures, t_i, in the annealing schedule were plotted for analysis. The schema ranged from a constant step size to various non-linear step sizes. Finally, it was decided that the equation used by both Kirkpatrick, Gelatt and Vecchi [22] and Golden and Skiscim [23] would be used, i.e.,

$$t_i = (10)(0.9)^{i-1}, \text{ for } i = 1, 2, \ldots, r.$$

Regarding the system equilibrium at each temperature issue, here again Golden and Skiscim's [23] work provided some insights. They developed the concept of only testing for equilibrium of the system at each **epoch,** where for the QAP implementation an epoch, e, is defined as an *a priori* specified number of accepted pairwise interchanges of facility locations. Thus, for each temperature t_i in the annealing schedule, after execution of each epoch, i.e., after e pairwise interchanges have been accepted, the mean of the total costs for all the assignments accepted during the epoch, $\bar{f}_e$, is compared with the grand mean, $\bar{f}'_e$, of the total costs of assignments for all preceding epochs at temperature t_i. If the mean total cost for the most recent epoch is sufficiently close (within ϵ) of the grand mean, the system is assumed to be in equilibrium at temperature t_i. Then, the next annealing temperature is selected and the procedure is repeated.

In an effort to avoid excessive running times, particularly at low temperatures when the system is 'freezing-up' (relatively few interchanges accepted), Kirkpatrick's, Gelatt and Vecchi [22] suggestions were taken, i.e., at each temperature t_i, enough exchanges, J, are attempted so that either: (1) at least an a priori constant, N, number of interchanges of location per facility, I'_p, are accepted, or (2) the number of attempted pairwise interchanges exceeds an *a priori* constant N' times the number of facilities, n. If the desired number of acceptances of location interchanges is not achieved at three successive temperatures, the system is considered frozen, and the last, least cost facility assignment $\mathbf{a}^*$ is considered the 'best' facility location assignment.

With the foregoing explanation of the concepts, assumptions and parameters involved in using the simulated annealing approach to solve the QAP, the following heuristic was implemented and used to experiment with this application [23]:

Simulated Annealing Heuristic

Step 1 Set $i = 1$, $I_c = 0$.

Step 2 Execute step 3 of the general algorithm for one epoch, e. Compute $\bar{f}_e$.

Step 3 Equilibrium test. If $|\bar{f}_e - \bar{f}'_e|/\bar{f}'_e \leq \epsilon$, go to step 4; otherwise, return to step 2.

Step 4 Annealing temperature change. $i = i + 1$. If all departments have been involved in at least N interchanges ($I'_p \geq N$), set $I_c = 0$ and go to step 2; otherwise, go to step 5.

Step 5 If the total number of attempted interchanges is **not** at least $N'n$ ($J \not\geq N'n$), go to step 2; otherwise, set $I_c = I_c + 1$ and go to step 6.

Step 6 If $I_c = 3$, **STOP;** otherwise, go to step 2.

The heuristic has been implemented on a DEC System 10 mainframe computer system. A flowchart of that implementation is shown in Figure 1.

Experimental Evaluation

In the form presented, the effectiveness and/or efficiency of the algorithm is influenced by a number of parameters. In order to gauge their influence, some of the parameters hypothesized to be most important with respect to solution time and quality, were selected for study. Specifically, the effects of problem size, n, the length of the epoch interval (number of interchanges per epoch), e, the constant N', and the steady state error constant, ϵ, were chosen for study.

An experiment with four factors (n, e, ϵ and N') was designed, with replication within each combination of factors to be achieved by varying the starting assignment of facilities to sites, $\mathbf{a}^0$.

Based upon some experimental results by Golden and Skiscim [23, 24], the following levels of three of the factors were employed in the experimental design:

Factor	Levels
e	5, 15, 25, 50
ϵ	0.25, 0.10, 0.01
N'	10, 50, 100

In a frequently cited paper in QAP literature, Nugent, Vollman and Ruml [9], compared a number of well accepted improvement type solution procedures: H63 (Hillier [18]), HC63-66 (Hillier and Connors [19]), CRAFT (Armour and Buffa [21]), and biased sampling (Nugent, Vollman and Ruml [9]). These procedures were applied to a common set of eight problems having $n = 5, 6, 7, 8, 12, 15, 20$, and 30 facilities, respectively, for five random starting assignments per problem. This same set of test problems plus problems for $n = 50$ and $n = 100$, for which two starting assignments per problem and flow data were generated from uniform distributions, were used in the experimental design. The site layouts (locations) are shown in Figure 2 for the problems solved in the experimental evaluation. The squares in Figure 2 represent sites and the numbers in the squares are site numbers.

The 'costs', c_{ikjh}, in the objective function of problem P1 are assumed to be the product of the flow between facilities i and j and the rectilinear distance between the centroids of their site locations, k and h, respectively. The squares in the layouts of Figure 2 are assumed to be "unit squares," so for

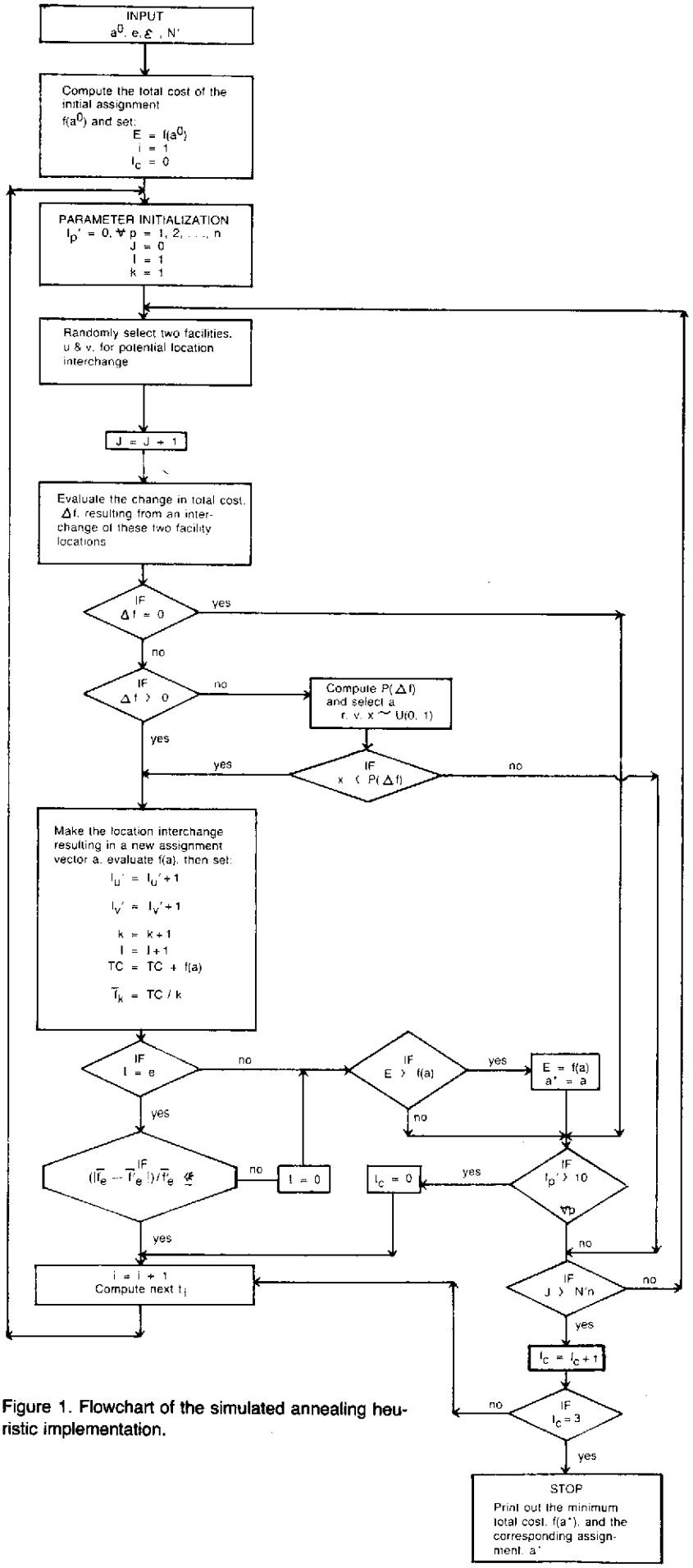

Figure 1. Flowchart of the simulated annealing heuristic implementation.

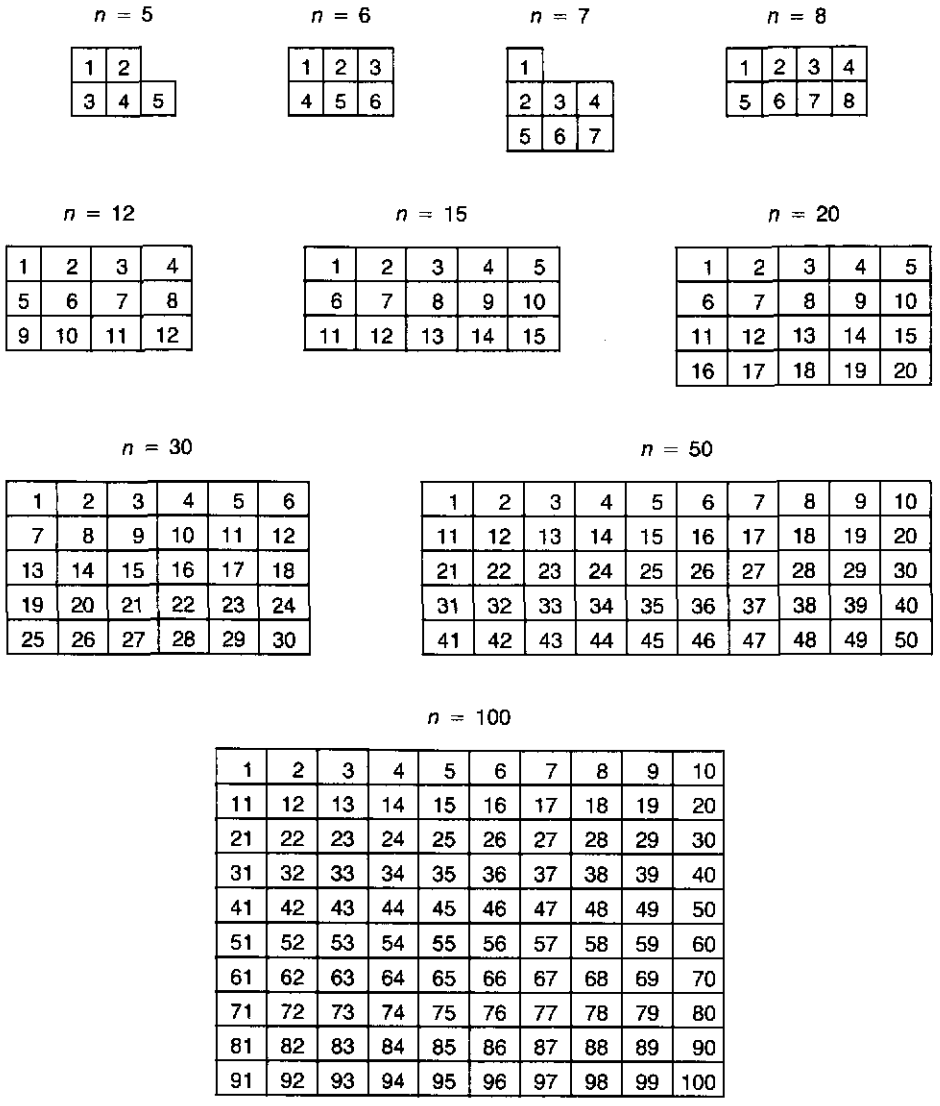

Figure 2. Layouts of site locations for the problems solved.

the layouts given it is quite simple to construct matrices whose elements are the distances between sites. Both the flow and distance matrices for the first eight experimental problems as well as the five starting assignments per problem are given in [9]. The flow and distance matrices for the new problems with $n = 50$ and $n = 100$, as well as the two starting assignments used per problem, are available by contacting the authors.

The experimental design was implemented using the data layout shown in Figure 3. This data layout required the evaluation of 1440 experimental conditions for the Nugent, Vollman and Ruml, problems plus 144 experimental conditions for the $n = 50$ and $n = 100$ problems. The purpose of the design was to study the effect of the factors on two dependent variables: CPU time and solution quality (objective function value of the least-cost assignment found). Hence, each data point obtained in the data layout represents the execution of the simulated annealing heuristic, coded in FORTRAN IV, for each of the 1584 experimental conditions on a DEC KL 1059 mainframe computer. For each experimental condition both the minimum total cost and the associated least cost assignment as well as the CPU time required for the system to "freeze" were noted.

Throughout the experiment, the other parameters of the algorithm, e.g., N, maximum I_c, and the annealing temperature equation for calculation of t_i were held constant at $N = 10$, maximum $I_c = 3$, and $t_i = (10)(0.9)^{i-1}$ for $i = 1, 2, \ldots$, respectively, as recommended by Kirkpatrick, et. al. [11].

Results

Tables 1 and 2 illustrate the results of the computational study outlined in the preceding section. Table 1 contains the percentage above the best known solution, averaged over starting solutions (replication), obtained by the simulated annealing heuristic for all values of n, e, ϵ and N'. Similarly, Table 2 contains the CPU times (seconds), averaged over

For each $n = 5, 6, \ldots, 30$

N'	Rep \ ϵ	$e = 5$			$e = 15$			$e = 25$			$e = 50$		
		0.25	0.10	0.01	0.25	0.10	0.01	0.25	0.10	0.01	0.25	0.10	0.01
10	1												
	2												
	3												
	4												
	5												
50	1												
	2												
	3												
	4												
	5												
100	1												
	2												
	3												
	4												
	5												

Figure 3. Format of experimental design for $n = 5, 6, 7, 8, 12, 15, 20$ and 30

Table 1. Percentages above best known solutions, averaged over starting solutions a^0, obtained by the simulated annealing heuristic for all n, e, ϵ, N'

N'	Best Known Soln.	n \ ϵ	$e = 5$			$e = 15$			$e = 25$			$e = 50$		
			0.25	0.10	0.01	0.25	0.10	0.01	0.25	0.10	0.01	0.25	0.10	0.01
10	25	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	43	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.40	0.00	1.40	0.00
	74	7	1.08	1.08	0.54	0.27	0.27	1.35	0.00	1.35	1.62	2.43	2.43	2.43
	107	8	0.56	0.56	0.93	1.87	1.31	3.18	2.43	2.62	3.55	3.93	2.24	3.93
	289	12	3.46	2.98	3.25	2.77	3.67	5.61	7.34	4.36	9.00	9.13	9.27	9.27
	575	15	4.87	4.87	3.72	3.90	4.24	6.40	2.78	3.65	8.00	9.15	9.15	7.69
	1285	20	3.83	3.83	3.52	4.87	4.87	6.19	4.68	5.31	7.30	7.91	8.17	7.30
	3064	30	4.00	4.00	5.48	4.46	4.46	4.78	4.86	4.86	6.18	6.48	7.00	7.53
	24472	50	1.11	1.60	1.33	1.52	1.52	1.02	1.69	1.69	1.37	2.03	2.03	1.90
	137011	100	0.88	0.62	0.62	1.35	1.35	1.35	1.78	1.55	1.55	1.14	1.14	1.14
50	25	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	43	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	74	7	0.27	0.27	0.54	0.00	0.27	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	107	8	0.00	0.00	0.37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	289	12	2.70	2.70	2.35	1.80	1.66	1.38	1.04	1.11	1.31	0.90	1.18	2.01
	575	15	1.77	1.77	1.84	0.56	0.90	1.04	1.32	1.36	1.81	0.63	1.08	2.40
	1285	20	2.83	2.83	2.77	1.63	1.63	1.71	1.77	1.85	1.73	2.65	2.77	2.65
	3064	30	2.75	2.75	3.39	2.15	2.15	1.86	1.95	1.95	1.33	2.20	1.70	3.37
	24472	50	1.17	1.17	1.17	0.74	6.51	0.45	0.53	0.53	0.10	0.47	0.47	0.04
	137011	100	0.43	0.43	0.43	0.77	0.77	0.77	0.95	0.95	0.95	0.24	0.24	0.24
100	25	5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	43	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	74	7	0.27	0.27	0.54	0.00	0.27	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	107	8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	289	12	2.70	2.70	2.35	1.80	1.52	0.90	0.69	1.59	0.69	1.87	1.38	1.31
	575	15	1.67	1.67	1.84	0.56	0.90	0.77	1.32	1.29	0.56	0.52	0.59	0.77
	1285	20	2.71	2.71	3.02	1.60	1.60	1.98	1.65	1.53	1.79	1.11	1.26	1.26
	3064	30	2.70	2.70	3.08	1.83	1.83	1.55	1.61	1.61	1.17	2.00	1.36	1.58
	24472	50	1.17	1.17	1.17	0.57	0.57	0.37	0.43	0.43	0.06	0.35	0.35	0.09
	137011	100	0.41	0.41	0.41	0.68	0.68	0.68	0.81	0.81	0.81	0.17	0.17	0.39

Table 2. CPU times in seconds, averaged over starting solution a^0, required by the simulated annealing heuristic to satisfy the stopping criterion for all n, e, ϵ, and N'

		$e = 5$			$e = 15$			$e = 25$			$e = 50$		
N'	n \ ϵ	0.25	0.10	0.01	0.25	0.10	0.01	0.25	0.10	0.01	0.25	0.10	0.01
10	5	0.52	0.53	0.51	0.54	0.52	0.44	0.49	0.47	0.44	0.44	0.44	0.44
	6	0.61	0.60	0.57	0.65	0.58	0.49	0.59	0.51	0.44	0.44	0.44	0.43
	7	0.64	0.67	0.67	0.66	0.71	0.53	0.63	0.60	0.48	1.47	0.47	0.47
	8	0.70	0.67	0.75	0.72	0.74	0.54	0.69	0.59	0.53	0.51	0.53	0.50
	12	1.04	1.04	1.13	1.21	1.23	0.95	1.06	1.08	0.85	0.88	0.83	0.81
	15	1.67	1.66	1.78	1.56	1.57	1.34	1.42	1.47	1.20	1.12	1.12	3.66
	20	2.74	2.74	2.84	2.41	2.42	2.15	2.26	2.19	1.80	1.85	1.80	1.76
	30	5.82	5.79	5.62	5.27	5.27	5.29	4.92	4.92	4.56	4.51	4.51	4.21
	50	21.48	34.65	31.93	14.95	14.98	14.52	12.98	12.98	13.21	19.32	19.31	17.61
	100	80.83	93.17	93.02	49.39	49.42	49.42	33.89	39.21	39.20	71.07	71.08	71.13
50	5	0.96	1.00	1.17	1.14	1.28	1.40	1.33	1.45	1.36	1.41	1.41	1.41
	6	1.22	1.20	1.37	1.59	1.79	1.88	1.75	1.75	1.87	1.56	1.56	1.56
	7	1.72	1.52	1.93	2.04	1.90	1.93	1.72	2.01	2.01	1.88	1.89	2.12
	8	1.72	1.80	2.29	2.53	2.58	2.53	2.26	2.55	2.12	2.23	2.31	2.26
	12	3.52	3.52	4.35	3.68	3.55	4.51	3.87	3.63	3.76	3.70	3.98	3.44
	15	5.36	5.36	6.51	5.53	5.30	5.38	5.55	5.11	5.40	5.08	5.89	5.13
	20	9.91	9.91	8.57	9.12	9.12	10.82	8.15	8.02	8.20	8.07	8.36	7.51
	30	16.31	16.26	18.40	18.76	18.76	19.97	20.17	20.14	21.89	16.70	16.49	13.32
	50	43.49	43.49	43.49	49.13	49.56	46.11	44.03	44.02	56.70	54.96	54.84	62.97
	100	229.57	229.62	229.60	171.66	171.48	171.51	140.37	140.43	140.42	244.66	244.08	244.30
100	5	1.36	1.35	1.69	1.50	1.73	1.88	1.94	1.84	2.05	2.76	2.76	2.76
	6	2.35	2.07	1.98	2.59	3.29	2.99	2.78	3.12	2.86	2.71	2.70	2.70
	7	3.02	2.69	3.18	3.13	3.58	3.85	3.88	3.81	3.81	3.85	3.51	3.50
	8	3.03	3.13	3.76	4.34	4.31	4.61	4.53	4.45	4.79	4.54	4.60	4.58
	12	6.98	6.98	7.75	7.46	7.36	8.79	7.28	6.87	6.93	7.77	7.06	7.00
	15	8.88	8.89	11.98	9.90	9.35	10.22	9.88	10.91	11.63	10.37	10.60	10.95
	20	19.54	19.54	20.55	15.61	15.62	20.21	15.24	15.67	18.82	15.13	17.23	16.44
	30	31.18	31.20	34.19	39.28	39.28	37.65	36.93	36.90	42.27	35.36	36.93	40.89
	50	75.33	75.32	75.21	88.32	88.29	86.01	91.39	96.61	86.72	99.46	99.44	92.85
	100	335.82	335.47	335.49	275.25	275.07	275.07	307.33	307.35	307.09	408.87	409.37	370.34

starting solutions, required by the heuristic to satisfy the stopping criterion for all values of n, e, ϵ and N'.

The data reported in Tables 1 and 2 were averaged over starting solutions (replication), for two reasons. First, there was very little variation in both solution quality and CPU time with starting solution (in the range 0% to 5% for $n \geq 12$). Second, publication space constraints prohibit inclusion of all data (including the complete data set would require 20 pages). Interested readers may contact the authors for a complete set of data.

The data in Tables 1 and 2 provides important insights into the performance of the simulated annealing heuristic with variation in the four main parameter settings, [n, e, ϵ, and N']. For example, with the exception of $N' = 10$ for $n = 7$, 8, for the small problems, $5 \leq n \leq 8$, very little change is seen in either solution quality or CPU time for the various settings of e and ϵ. The probable reason for the exception at $N' = 10$ is that an insufficient number of interchanges were attempted at each temperature to reach equilibrium. This problem gets worse as n increases. Further, as N' increases for each n, the CPU time required by the heuristic increases.

By observing Tables 1 and 2, we see that the best quality solutions for problems of size $8 \leq n \leq 30$ are achieved for the highest level of $N' = 100$, a mid-range level of $e = 15$ or $e = 25$, and the lowest level for $\epsilon = 0.01$. However, for the large problems having $n = 50$ and $n = 100$, the best parameter settings appear to be [e, ϵ, N'] = [50, 0.01, 100]. These best quality solutions are obtained by sacrificing CPU time (the $e = 50$ results require about twice the time required by $e = 15$, and yield only slightly better solution quality).

The high quality solutions obtained for the largest sized problems cause the authors to reason that perhaps e and N' have not been set high enough nor ϵ low enough to observe the best solutions obtainable from application of the simulated annealing heuristic. But, increasing these parameters will certainly greatly increase the CPU time required. Furthermore, the solutions obtained at the current parameter settings (even with $e = 15$) are probably acceptable and are achieved with reasonable CPU time expenditure.

From the foregoing observations and within the parametric bounds of the experimental framework discussed, it may be postulated that the simulated annealing heuristic yields good quality solutions if $e = 15$, $\epsilon = 0.01$ and $N' = 100$. Of course, these parameter settings lead to longer CPU times to achieve the least cost assignment than different values of these parameters. However, some trade-off must be made between

Table 3. Comparison of Solutions and CPU Times for a Number of QAP Solution Procedures

Number of Depts.	Starting Solution No.	Best Cost for Simulated Annealing	CPU Time (Sec.)	CRAFT Solution	CPU Time (Sec.)	CRAFT Biased Sampling Solution	CPU Time (Sec.)	Revised Hillier Solution	CPU* Time (Sec.)
	1	25	1.87	29	0.56	29	1.99	25**	
	2	25	1.87	29	0.55	29	1.87	25**	
5	3	25	1.87	25	0.56	25	1.98	25**	
	4	25	1.87	29	0.58	29	2.08	25**	
	5	25	1.87	29	0.59	26	2.06	25**	
Means		25	1.87	28.2	0.57	27.6	2.00	25**	0.24
	1	43	2.95	43	0.54	43	1.82	43	
	2	43	2.94	43	0.59	43	1.92	43	
6	3	43	2.74	46	0.56	46	1.38	43	
	4	43	2.74	43	0.57	43	1.90	43	
	5	43	2.92	46	0.61	43	2.02	43	
Means		43	2.86	44.2	0.57	43.6	1.91	43	0.24
	1	74	4.13	79	0.73	76	2.63	72**	
	2	74	3.45	78	0.81	74	3.19	72**	
7	3	74	4.27	74	0.78	74	3.11	76**	
	4	74	3.83	84	0.79	74	3.07	74**	
	5	74	3.37	83	0.71	76	2.75	72**	
Means		74	3.81	79.6	0.76	74.8	2.95	73.2**	0.55
	1	107	4.99	119	0.72	113	2.98	107	
	2	107	5.05	107	0.81	107	3.29	107	
8	3	107	3.84	107	0.72	107	2.82	109	
	4	107	4.99	110	0.71	107	2.98	107	
	5	107	5.07	107	0.73	107	2.98	107	
Means		107	4.79	110	0.74	108.2	3.01	107.4	0.44
	1	289	6.59	298	1.32	296	7.06	304	
	2	291	7.32	308	1.33	298	6.82	293	
12	3	293	6.65	291	1.46	296	7.72	307	
	4	293	7.56	295	1.52	295	8.28	297	
	5	289	6.54	289	1.33	289	5.32	291	
Means		291	6.93	296.2	1.39	294.8	7.04	298.4	1.35
	1	576	15.53	628	1.83	588	11.36	596**	
	2	575	9.40	588	2.44	580	17.10	580**	
15	3	576	11.49	591	2.41	587	15.38	575**	
	4	584	10.72	640	2.01	575	14.87	576**	
	5	580	11.02	583	2.39	575	18.31	584**	
Means		578.2	11.63	606	2.22	581	15.40	582.2**	2.38
	1	1303	18.67	1334	5.81	1317	41.00	1328**	
	2	1313	15.45	1354	4.05	1317	35.50	1297**	
20	3	1315	20.75	1351	4.71	1328	34.60	1348**	
	4	1300	20.14	1324	4.57	1331	39.60	1328**	
	5	1309	19.10	1332	4.91	1312	41.80	1322**	
Means		1308.0	18.82	1339	4.81	1321	39.50	1324.6**	5.83
	1	3064	30.85	3090	28.54	3120	220.00	3070	
	2	3111	43.75	3192	19.67	3117	217.00	3147	
30	3	3095	49.24	3197	20.48	3156	204.00	3153	
	4	3079	40.15	3273	15.68	3119	202.00	3124	
	5	3150	47.35	3237	18.86	3106	214.00	3077	
Means		3099.8	42.27	3197.8	20.65	3124	211.00	3114.2	22.86

Note: Unless otherwise indicated, all algorithmic results presented in this table were obtained from FORTRAN codes implemented on a DEC KL 1059 mainframe computer. All simulated annealing results are for the parameter settings$[n, e, \epsilon, N'] = [n, 15, 0.01, 100]$.

*Results of FORTRAN code run on a Prime 750 computer, as reported in [20].

**Different initial site layouts, and hence different distance matrices, from those in Nugent, et. al. were used for these problems in [20].

solution quality and CPU time as this is in reality a bi-criteria optimization process.

In order to compare the effectiveness of the simulated annealing heuristic with other techniques for solving QAP's, Table 3 was constructed. It compares the costs of assignments and the corresponding CPU times for the Nugent, Vollman and Ruml problem set obtained by simulated annealing with $[e, \epsilon, N']$ fixed at [15, 0.01, 100] for all n, CRAFT, CRAFT Biased Sampling, and the Revised Hillier Procedure [20]. As seen, the simulated annealing algorithm obtained optimal as-

signments for $n = 5, 6, 7$, and 8. Since the optimal solutions for the larger problems with $n = 12, 15, 20$ and 30 are unknown, the authors are content with the fact that the simulated annealing algorithm obtained assignments with smaller costs (better quality) in practically every case, and required significantly less CPU time than biased sampling and about twice the time required by CRAFT and the revised Hillier procedure. Table 4 shows both the minimum and average costs and CPU times for all problems solved using the simulated annealing algorithm, including the constructed problems having $n = 50$ and $n = 100$ facilities.

A power function of the form $y = ax^b$ was fitted to the average CPU time required by the algorithm (shown in Table 4) versus problem size, n. A plot of the resulting function is shown in Figure 4. The equation of best fit was determined to be:

$$\text{CPU time (seconds)} = 0.072\, n^{1.679},$$

with coefficient of determination $r^2 = 0.994$. This curve implies that, as expected, as problem size increases so does the required CPU time. However, average solution time increases at a rate less than the square of problem size.

Table 4. Statistics for simulated annealing experiments across all parameters [e, ϵ, N', a^0] for each problem size, n

	Total Cost			CPU Time		
n	Minimum	Average	Std. Dev.	Minimum	Average	Std. Dev.
5	25	25.00	0.00	0.37	1.24	0.69
6	43	43.03	0.31	0.40	1.60	0.94
7	74	74.36	0.86	0.45	1.99	1.24
8	107	107.82	1.87	0.50	2.37	1.53
12	289	297.81	8.92	0.77	4.05	2.72
15	575	590.55	16.90	1.08	5.80	3.88
20	1285	1326.70	28.58	1.75	9.51	6.67
30	3064	3161.60	59.47	4.10	19.99	13.84
50	24472	24736.47	362.32	8.86	52.10	30.36
100	137011	138065.65	620.82	26.72	195.54	118.21

The layouts of the problems solved showing the minimum cost assignments of facilities to sites, together with the CPU time required to obtain the solution, are shown in Figure 5 (refer to Figure 2 for the corresponding site location numbers). These are presented for the benefit of other QAP researchers who wish to compare their solutions with those achieved in this study.

Conclusions

This paper has studied the application of Kirkpatrick's, et. al. [22], framework for solving discrete combinatorial optimization problems through its application to QAP's by a simulation-type heuristic. Tables 1 and 2 provide insight into the behavior of the heuristic with variation in four of the many possible parameters which may influence its performance. And, as seen in Table 3, the solution quality obtained by

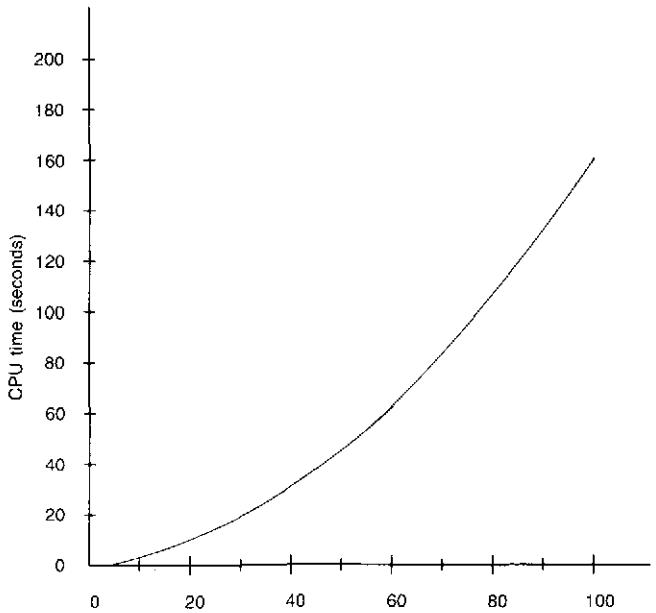

Figure 4. Average CPU time required by all simulated annealing runs versus problem size, n

simulated annealing, when applied to a common problem set, was superior to that obtained by the traditional approaches to QAP for practically every starting solution and for the means over starting solutions. Further, the average CPU times required by simulated annealing was less than that required by biased sampling and only slightly longer than that required by CRAFT and the revised Hillier procedure. Figure 4 also correlates well with a conclusion drawn by Kirkpatrick, et. al. [22], in which they state, "Our numerical studies suggest that results of good quality are obtained with annealing schedules in which the amount of computational effort scales as n or as a small power of n." And, as seen in Figure 4, for $5 \leq n \leq 100$, average CPU time scales as the 1.679 power of n for the example QAP's solved in this study.

However, the results in Tables 1 and 2 came from a significant effort in experimental design and analysis requiring 1584 separate runs of the simulated annealing algorithm under different combinations of parametric values. This computational experience has taught the authors the same lessons learned by Golden and Skiscim [23, 24], i.e., the simulated annealing procedure is sensitive to a number of control parameters and stopping rules. And, perusal of Tables 1 and 2 does not indicate consistently superior settings of the control parameters studied [e, ϵ, and N']. Other potentially influential parameters on simulated annealing performance which were **not** studied in this research are N, maximum I_c, and the annealing temperature equation for t_i.

Given the above constraints on the study, application of simulated annealing to the "standard" Nugent, Vollman and Ruml [9], problem set yielded some superior solutions to the larger problems which have not been reported heretofore (see Figure 5). Larger problems ($n = 50$ and $n = 100$) were generated and solved for the benefit of computational experience with simulated annealing.

Even though the simulated annealing methodology is sen-

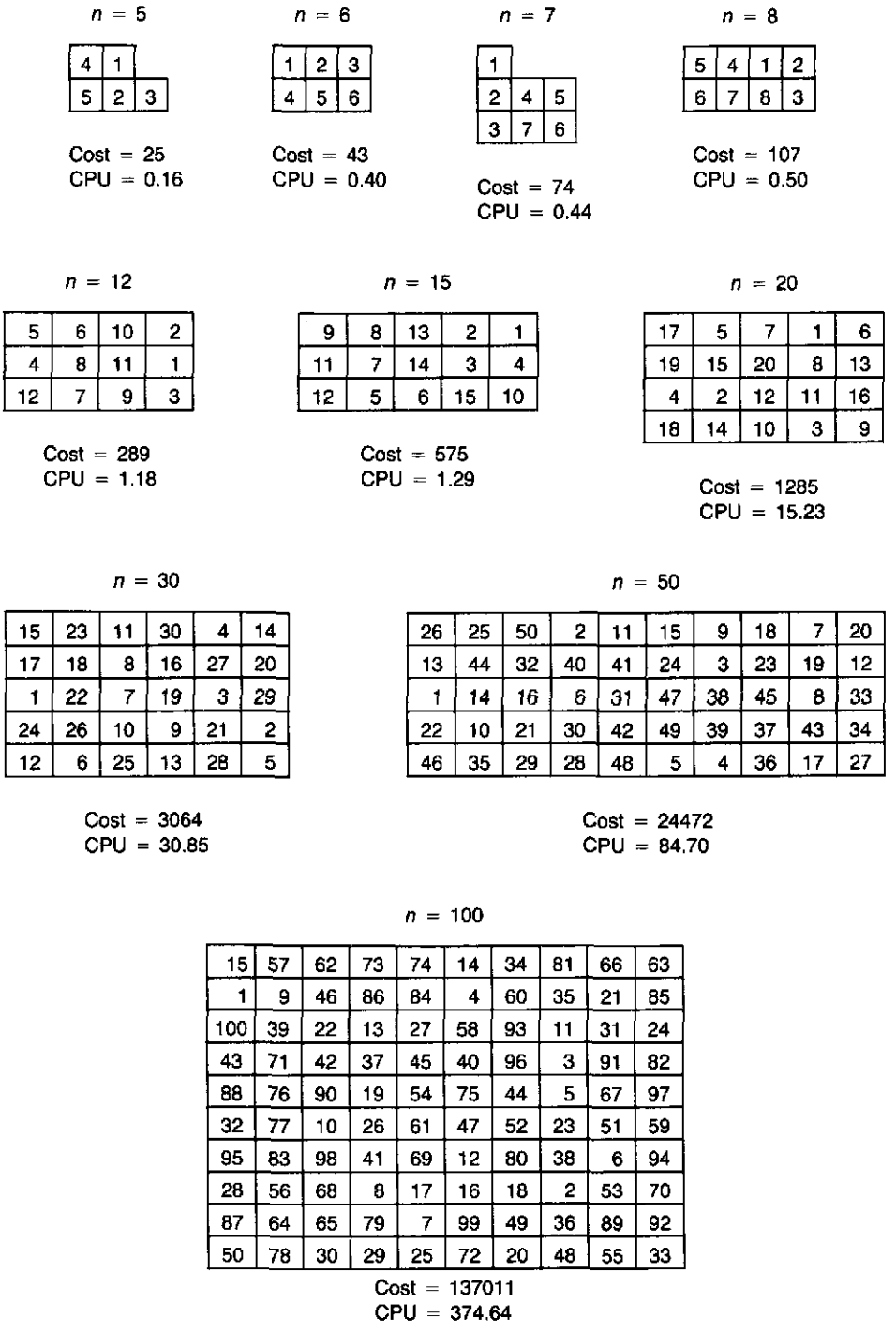

Figure 5. Minimum cost assignments of facilities to sites (CPU times are in seconds on a DEC System 10)

sitive to a number of parameters, the authors believe, as do others, that it is a promising approach for solving combinatorial optimization problems like the QAP and that more research is warranted to define appropriate parameter values to optimize its effectiveness and efficiency.

Acknowledgement

The authors wish to thank the anonymous referees for their helpful comments toward revising this paper for the better.

REFERENCES

[1] Wilhelm, M. R., T. L. Ward, R. A. Rankin, R. Gupta and R. Dutton, "Innovative Improvement Procedures for Solving Quadratic Assignment Problems," Proceedings of the 1984 Conference of the Institute of Industrial Engineers, May 1984, with subsequent publication in *Computerized Facilities Planning: Selected Readings*, H. L. Hales ed., Industrial Engineering and Management Press, Norcross, GA 1985.

[2] Francis, R. L. and J. A. White, *Facility Layout and Location: An Analytical Approach*, Prentice Hall, Englewood Cliffs, NJ, 1974.

[3] Francis, R. L., L. F. McGinnis and J. A. White, "Location Analysis," invited review in *European Journal of Operational Research*, 12 (1983), 220-252.

[4] Pierce, J. F. and W. B. Crowston, "Tree Search Algorithms for Quadratic Assignment Problems," *Naval Research Logistics Quarterly*, 18 (1971), 1-36.

[5] Sahni, S. and T. Gonzalez, "P—Complete Approximation Problems," *Journal of the Association of Computing Machinery*, 23, 1976.

[6] Parker, G. R. and R. L. Rardin, "An Overview of Complexity Theory in Discrete Optimization: Part I. Concepts," *IIE Transactions*, 14, 1 (1982).

[7] Foulds, L. R., "Techniques for Facilities Layout: Deciding Which Pairs of Activities Should Be Adjacent," *Management Science*, 29, 12 (1983), 1414-1426.

[8] Hanan, M. and J. M. Kurtzberg, "A Review of the Placement and Quadratic Assignment Problems," *SIAM Review*, 14, 2 (1972), 324-341.

[9] Nugent, C. E., R. E. Vollman and J. Ruml, "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations," *Operations Research*, 16 (1968), 150-173.

[10] Koopmans, T. and M. Beckmann, "Assignment Problems and the Location of Economic Activities, *Econometrica*, 25, (1957), 53-76.

[11] Lawler, E. L., "The Quadratic Assignment Problem," *Management Science*, 9, (1963), 586-599.

[12] Gavett, J. W. and N. V. Plyter, "The Optimal Assignment of Facilities to Locations by Branch and Bound," *Operations Research*, 14, (1966), 210-232.

[13] Graves, G. W. and A. B. Whinston, "An Algorithm for the Quadratic Assignment Problem," *Management Science*, 17, (1970), 453-471.

[14] Bazaara, M. S. and A. N. Elshafei, "An Exact Branch-and-Bound Procedure for Quadratic Assignment Problem," *Naval Research Logistics Quarterly*, 26, 1 (March 1979), 109-120.

[15] Bazaara, M. S. and H. D. Sherali, "Benders' Partitioning Scheme Applied to a New Formula of the Quadratic Assignment Problem," *Naval Research Logistics Quarterly*, 27, 1 (March 1980), 29-41.

[16] Burkard, R. E. and K. H. Stratmann, "Numerical Investigations on Quadratic Assignment Problems," *Naval Research Logistics Quarterly*, 25, (March 1978), 129-144.

[17] Liggett, R. S., "The Quadratic Assignment Problem: An Experimental Evaluation of Solution Strategies," *Management Science*, 27, (1981), 442-458.

[18] Hillier, F. S., "Quantitative Tools for Plant Layout Analysis," *Journal of Industrial Engineering*, 14, 1 (1963), 33-40.

[19] Hillier, F. S. and M. M. Connors, "Quadratic Assignment Problem Algorithms and the Location of Indivisible Facilities," *Management Science*, 13, 1 (1966), 42-57.

[20] Picone, C. J. and W. E. Wilhelm, "A Perturbation Scheme to Improve Hillier's Solution to the Facilities Location Problem," *Management Science*, 30, 10 (1984), 1238-1249.

[21] Armour, G. C. and E. S. Buffa, "A Heuristic Algorithm and Simulation Approach to Relative Locations of Facilities," *Management Science*, 9 (1963).

[22] Kirkpatrick, S., C. D. Gelatt, Jr. and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, 220, 4598 (1983), 671-680.

[23] Golden, B. L. and C. C. Skiscim, "Using Simulated Annealing to Solve Routing and Location Problems," Working Paper Series MS/S 84-001, University of Maryland, January 1984, accepted for publication in *Naval Research Logistics Quarterly*.

[24] Skiscim, C. C. and B. L. Golden, "Optimization by Simulated Annealing: A Preliminary Computational Study for the TSP," Proceedings of the 1983 Winter Simulation Conference, Washington, D.C.

[25] Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller, *Journal of Chemical Physics*, 21 (1953), 1087-1092.

[26] Bonomi, E. and J. Lutton, "The N-City Travelling Salesman Problem: Statistical Mechanics and the Metropolis Algorithm," *SIAM Review*, 26, 4 (1984), 551-568.

[27] Burkard, R. E. and F. Rendl, "A Thermodynamically Motivated Simulation Procedure for Combinatorial Optimization Problems," *European Journal of Operational Research*, 17, (1984), 169-174.

Mickey R. Wilhelm is a Professor of Engineering Management and Industrial Engineering, and Chairman of the Department of Industrial Engineering, at the University of Louisville. His research interests center around facilities planning, including location, layout and materials handling. He received the BSE in Electrical Engineering, and the MSE and Ph.D. degrees in Industrial and Systems Engineering from the University of Alabama in Huntsville. Dr. Wilhelm is a Senior Member of IIE, and a member of SME and ASEE. He is currently Chairman of the Academic Programs and Activities Committee of the College-Industry Council on Material Handling Education (CICMHE) of the Material Handling Institute (MHI). He is a registered Professional Engineer in the Commonwealth of Kentucky in the branch of Industrial Engineering.

Thomas L. Ward is a Professor of Engineering Management and Industrial Engineering in the James B. Speed Scientific School at the University of Louisville. His research interests include industrial engineering systems that require computer intensive methods for their analysis and design. Following twenty years of experience in industry, he held academic positions at the University of Southern California, California State Polytechnic University at Pamona, and the Jet Propulsion Laboratory at the California Institute of Technology. He received a B.S. in Physics from the University of Texas at Austin, and an M.S. and Ph.D. in Industrial and Systems Engineering from the University of Southern California. Dr. Ward is a Senior Member of IIE, IEEE, and SME. He is a member of ASME and holds California and Kentucky registration as a Professional Engineer in the electrical, industrial, and mechanical branches.