Journal of the Operations Research
Society of Japan
Vol. 38, No. 1, March 1995

SIMULATED ANNEALING SCHEME INCORPORATING MOVE DESIRABILITY TABLE FOR SOLUTION OF FACILITY LAYOUT PROBLEMS

Seyed Hessameddin Zegordi Kenji Itoh Takao Enkawa Shih Li Chung
Tokyo Institute of Technology

(Received March 2 1992; Final July 4 1994)

Abstract *Simulated annealing*, an analogy between statistical mechanics and combinatorial optimization, has attracted considerable attention due to its potential in dealing with certain traditional optimization problems. However, it has been shown to be parameter sensitive and require more computational effort to produce high quality solutions than traditional heuristics. The present paper reports on a new approach to applying this method to a class of quadratic assignment problems, i.e., the facility layout problem.

This approach combines the simulated annealing methodology with a specific layout design rule, which in the case of this study is a "*Move Desirability Table*". Two annealing methods are proposed based on this approach which differ only in cooling schedule. The first method produces high quality solutions, while the second method is faster with a slight degradation in the quality of solutions and therefore suitable for larger problems.

Performance of the two methods was numerically tested on standard problems as well as two larger problems ($n = 50$ and $n = 100$; n is the number of facilities) and was compared to that of Wilhelm & Ward, Connolly and other heuristics, e.g., QAPH4, CRAFT, biased sampling and the revised Hillier procedure. As the experimental results, it was found that except in the case of $n = 50, 100$, the proposed methods are computationally faster than Wilhelm & Ward and that the solution qualities of the proposed methods in all the cases are superior to those of Wilhelm & Ward, Connolly and other heuristic procedures.

1 Introduction

Most production control problems, such as scheduling and transportation problems, are known to be combinatorial optimization problems in nature. In these problems, it is difficult to derive high quality solutions due to the explosion in possible number of combinations which takes place as the problem size increases. Due to such characteristics this class of problems has largely relied on heuristic methods, and therefore, algorithms that efficiently derive good solutions are highly desired.

The facility layout problem, which is one of the major problems in the field of production control, shares these characteristics. This problem is normally formulated as a *quadratic assignment problem (QAP)* when facilities are to be assigned to sites and when there is an interaction between the facilities that is dependent upon their locations [17]. In addition to the facility layout problem, several combinatorial optimization problems are included in QAP, for example, the assignments of inter-communicating objects to locations to minimize the total cost of communication between them, and the allocation of industrial, residential, and shopping areas, schools, etc. to the land sites to maximize the total utility.

The general form of QAP formulation [9] may be written as follows:

$$Min \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} C_{ijkl} \cdot x_{ik}x_{jl}$$

$$s.t. \quad \begin{cases} C_{ijkl} = c_{ij} d_{kl} \\ c_{ij} = f_{ij} + f_{ji} \\ \sum_{k=1}^{n} x_{ik} = 1 \quad \text{for } i = 1, \ldots, n \\ \sum_{i=1}^{n} x_{ik} = 1 \quad \text{for } k = 1, \ldots, n \\ x_{ik} = \begin{cases} 1 & \text{if Facility } i \text{ is assigned to Location } k \\ 0 & \text{otherwise} \end{cases} \end{cases}$$

c_{ij}: Total material flow or cost between Facilities i and j
d_{kl}: Distance between Locations k and l
f_{ij}: Flow or cost of material from Facility i to Facility j
n : Number of facilities or locations

In the above equations, the decision variable x_{ik} is equal to one if Facility i is assigned to Location k, and otherwise is equal to zero. The constraints ensure that exactly n facilities are to be assigned to exactly n locations.

QAP is an NP-complete problem [26], i.e., there is no general solution technique available that is able to reach an optimal solution for problems of various sizes in polynomially bounded computational time. It is not surprising, therefore, that so much effort has been devoted to devising heuristic methods which yield solutions of acceptable quality in "reasonable" computational time, even if not optimal [30]. These heuristic methods may be classified as either *constructive procedures* or *improvement procedures*. In constructive procedures a solution is built from the null solution by making successive assignments of facilities to locations. The best known constructive procedures include CORELAP [18], MAT [7] and the method of Graves & Whinston [11].

In contrast, improvement procedures start from an initial assignment or the existing layout, and try to improve it through assignment exchanges [23]. Improvement procedures have generally been found to yield better solutions than constructive procedures. The most cited improvement methods are CRAFT [2], H63(the method of Hillier [13]), HC63-66(the modified version of H63) [14], Biased Sampling [23], COL [29], FRAT [15], and the revised Hillier method [25]. However, improvement procedures frequently get stuck in local optima after executing some iterations, and their performance depends on the initial solution given at the starting point.

Recently, *Simulated Annealing (SA)* [16], an analogy between statistical mechanics and combinatorial optimization, has been applied to support the solution of QAP by a number of researchers. Although classified as iterative improvement procedures the methods based on this approach are able to escape from local optima, which are not global optima, by accepting assignments that momentarily deteriorate the objective function value under specific conditions. Theoretically, it has been proved that simulated annealing algorithm converges asymptotically with probability one to the set of globally minimum costs(GC1, i.e., Global Convergence with probability one) [12, 22]. In application to combinatorial optimization problems, it has also been shown that these methods are able to obtain high quality solution when applied to different problems [3, 6, 10, 30].

Although SA has great ability for deriving good solutions, this technique is known to be parameter sensitive and time consuming. The performance and computational time of this scheme heavily depend on the *annealing schedule* and its parameter tuning. In general, high quality solutions can be obtained by tuning the parameters leading to a large number of iterations. Such parameter settings also lead to considerable computational time. Concerning the parameter setting, however, an appropriate setting for each parameter may

be obtained that leads to fewer iterations, therefore less computational time, and good solutions. Furthermore, it is difficult to obtain appropriate parameter settings for all cases. Such parameter settings vary according to the type of problems and their sizes.

A modification of Tabu Search technique [28] is also applied in order to overcome local optimality as well as Simulated Annealing. In that study, a number of numerical experiments with different parameters were conducted and the performance was compared with that of one of traditional SA algorithms, i.e., QAPH4 [3], and it was reported that this method is superior to the traditional SA.

The objective of the present study is to create an annealing scheme that leads to superior performance with fewer iterations and therefore less computational time; even though it might lose the GC1 property. For this purpose, we propose a new approach for solving the facility layout problem which combines the simulated annealing methodology with a problem-specific structure or information. This approach requires no parameter setting or tuning. We develop two algorithms based on this approach, which differ only in cooling schedule, and examine their performance using numerical experiments with problems of varying sizes.

In the next section, the simulated annealing algorithm and previous studies are briefly reviewed. The proposed simulated annealing methods are described in Section **3**. The results of the numerical experiments are mentioned, and some discussion are given in Section **4**. Finally, conclusions are given in Section **5**.

2 Methodology of simulated annealing

2.1 General algorithm

The principal idea used in simulated annealing was first proposed by Metropolis et al. [21]. Kirkpatrick et al. [16] applied the concept of the annealing to combinatorial optimization problems. This concept is based on an analogy between the physical annealing process of solids and solution process of combinatorial optimization problems.

The physical annealing process which obtains low energy states of a solid in a heat bath may be modeled as follows [16]: In each step an atom is given a small random displacement and the resulting change in the energy of the system, ΔE, is computed. If $\Delta E \leq 0$, then the displacement is accepted, and the configuration with the displaced atom is used as the starting point of the next step. The case of $\Delta E > 0$ is treated probabilistically, i.e., the probability that the configuration is accepted is determined using Equation (**1**), where T is the temperature and k_b is Boltzmann's constant.

$$P(\Delta E) = e^{-\frac{\Delta E}{k_b T}} \tag{1}$$

A random number uniformly distributed on the interval [0,1] is selected and compared with $P(\Delta E)$, which is calculated by Equation (**1**). If it is less than $P(\Delta E)$, then the new configuration is accepted and used to start the next step. Otherwise, the configuration is rejected. The process is continued until an "*equilibrium*" state is achieved, then the temperature is lowered according to the annealing schedule. This procedure is repeated until the system *freezes*. At each temperature, the annealing schedule must allow the simulation to proceed sufficiently long for the system to reach steady state condition (equilibrium point).

The general algorithm of SA derived from analogy with the physical annealing process described above is illustrated in pseudo-code in Figure **1**. In this figure, the outer loop represents the freezing process in which the temperature is changed, whereas the inner

Figure 1: General simulated annealing algorithm for QAP

```
Select an initial allocation a = a^0 ∈ S;
Select an initial temperature T = T_0 > 0;
Set temperature change counter t = 0;
Repeat ——— Freezing process
    Set repetition counter i = 0;
    Repeat ——— Equilibrium process
        Generate an allocation b, a neighbour of a;
        Calculate Δf = f(a) − f(b);
        If Δf > 0  then a := b
                   else if random(0,1) < exp(−Δf/T) then a := b;
        i := i + 1;
    until i = N(t);
    t := t + 1;
    T := T(t);
until stopping criterion true.
```

loop describes the equilibrium procedure that determines how many exchanges are to be attempted at each temperature.

2.2 Comparison of SA algorithms

(1) Annealing framework

Many researchers have applied the SA methodology to various combinatorial problems with slight variations of the annealing schedule, and have reported that SA provides a promising approach for solving combinatorial optimization problems. Nonetheless, this methodology has several weak points as presently applied. One of the weakest points may be that the solution quality is sensitive to certain control parameters, and it is hard to set appropriate values for the parameters which enable good solutions. For this reason, we intend to develop a new SA approach that can derive good solutions while achieving robustness to parameter settings and enabling easy tuning of the parameters.

In order to cope with the parameter sensitiveness in SA, at least two different approaches may be considered. One is to solve the problem using problem-specific structures and/or information during execution of the annealing process. The other approach is to decrease the number of control parameters required as input variables. This parameter reduction approach may be achieved either by utilizing added information or by changing the procedures that constitute the annealing process itself. Table 1(a) summarized the general characteristics of SA schemes as they relate to these two approaches.

Regarding the first approach, there are few existing schemes of SA in which problem-specific structures or rules are utilized. In the traditional improvement procedures for QAP, however, several methods employed such problem-specific rules or knowledge. Among these problem specific-rules and structures, the "*Move Desirability Table (MDT)*" on which Hillier & Connors [14] method was based, is deemed the 'best buy' considering solution quality and CPU times [23]. Therefore, in this study, it was decided to combine MDT with the annealing process. This combination may be considered as one of the most distinguishing features of the proposal approach.

On the other hand, most of existing SA schemes have multiple parameters which constitute the annealing schedule. The larger the number of parameters, the more difficult it is to tune them, and therefore the more difficult it is to derive high quality solutions for all

Table 1: Comparison of simulated annealing schemes for combinatorial problems

(a) General characteristics

Scheme	W & W[30]	G & S[10]	B & R[3]	Connolly[6]	Prop. methods
Use of problem structure	No	No	No	No	Yes (MDT)
Use of trial information	No	No	No	Yes	Yes
No. of parameters required	7	3	3	2	1 (SA-MDT1) 2 (SA-MDT2)

(b) Parameters of annealing schedule

Scheme	W & W	G & S	B & R	Connolly	Prop. methods
Pairwise exchange	Randomly among any pairs	Randomly among any pairs	Randomly among any pairs	Sequentially among neighborhood facility pairs	Based on desirability of move within neighborhood facility pairs
Initial temperature	Fixed and relatively low	Fixed and relatively high	Fixed and relatively high	Determined by trials	Determined by trials*
Temperature tuning	According to prescribed annealing schedule; quick lowering	According to prescribed annealing schedule; quick lowering	According to prescribed annealing schedule; quick lowering	Using information from trials; slow lowering	Using information from trials* (SA-MDT1) According to prescribed annealing schedule** (SA-MDT2)
Equilibrium test criterion	solution quality after every prescribed duration	solution quality after every prescribed duration	fixed no. of iterations	None	Anytime during annealing process according to problem-specific conditions
Stopping condition	After three steps if the desired no. of acceptances of exchanges is not achieved	Related to initial and final temperatures	Rejection of all exchanges during a no. of iterations	Consecutive no. of rejected uphill exchanges	The total no. of exchanges correspondent to problem size

*: The same procedure as Connolly
**: The same procedure as W & W and B & R

cases.

For this reason, most schemes introduce a few parameters except the method of Wilhelm & Ward [30]. In contrast, Connolly [6], which has the fewest parameters among the existing schemes, conducts trial executions to elicit information on the annealing schedule at the expense of some computational time. These trials contribute to reduction in the number of annealing parameters and may be useful to give high quality solutions. Problem-specific rules like the MDT mentioned above also play roles in the parameter reduction.

The basic idea of the proposed scheme is to combine the above two approaches in SA methodology. In addition to this idea, we can obtain further insight for improvement by comparing characteristics of annealing process among the existing schemes as shown in Table 1(b).

(2) Annealing procedure

(a) Pairwise exchange:

Previous studies can be categorized into two types with regard to the manner of selecting a pairwise exchange. One type is to select randomly among any possible pairs. In the other type of exchange, the pairs are selected in order. In the method of Connolly [6], for example, a neighborhood pair (e.g. neighboring facilities in location) is picked up sequentially (in order) from among all possible pairs during the iterations, which is called sequential neighborhood search.

In the same study, Connolly also examined the effects of random search versus sequential neighborhood search at the pairwise exchange in terms of solution quality. He concluded that the sequential neighborhood search is superior for QAPs. From his findings, we adopt

a policy of sequential neighborhood exchange criterion in this study. Furthermore, since we employ MDT as problem-specific information, a pair can be selected according to the *desirability of move* to any adjacent facilities, while in Connolly's method a pair is selected sequentially among all neighborhood exchange pairs.

(b) Initial temperature:

The number of iterations during the annealing process partly depends on the initial temperature. The procedures for setting the initial temperature may be broadly classified into two types. Most schemes determine the initial temperature as fixed numbers prior to execution of the annealing process. Golden & Skiscim [10] and Burkard & Rendl [3] set high initial temperatures, while Wilhelm & Ward [30] starts at a low temperature.

On the other hand, Connolly [6] determines the initial temperature using information obtained during the trials prior to the annealing process. He also suggested the possible existence of an optimal temperature between the extremely high and low temperatures. We adopt his suggestion, which justifies starting the annealing from a low temperature, hoping to reduce the computational time. According to this policy, the proposed scheme estimates the initial and final temperatures to try to maximize the proportion of the search performed near the unknown optimal temperature by using trial information like Connolly's.

(c) Temperature tuning:

One of the major issues related to the annealing schedule is how to tune the temperature during the annealing process. In the existing SA schemes, there are several types of procedures for tuning temperature. One is to employ a temperature lowering function based on the annealing schedule. Each annealing scheme has its own individual function. For example, Wilhelm & Ward [30] and Burkard & Rendl [3] calculate the next temperature, T_{i+1}, using Equation(**2**), where the parameter α is usually set close to one. Golden & Skiscim [10] used Equation (**3**) which reduces the temperature by 1/25 of the initial temperature at each stage, likewise resulting in a linear and therefore fast temperature reduction.

In another type of temperature tuning, information obtained from trials prior to the execution of annealing process is utilized. During these trials, for example in Connolly [6], the initial temperature T_0 and the final temperature T_f are determined. Using these temperatures and the constant M determined by the problem size, the next temperature is calculated by Equation(**4**). In this equation, parameter β usually has a small value, and therefore the temperature reduction proceeds relatively slowly.

$$T_{i+1} = \alpha T_i \qquad 0 < \alpha < 1 \tag{2}$$

$$T_{i+1} = T_i - \frac{T_0}{25} \qquad i = 0, \ldots, 25 \tag{3}$$

$$T_{i+1} = \frac{T_i}{1 + \beta T_i} \qquad \beta = \frac{T_0 - T_f}{M T_0 T_f} \tag{4}$$

It is clear that when the temperature is too high, a lot of poor uphill moves are accepted. Conversely, when the temperature is too low the probability of falling into a local minimum is very high. Kirkpatrick et al. [16] mentioned that between these two extremes there is a critical band of the temperatures in the whole annealing process where very slow cooling is required. Equation(**4**) and the procedures for determining the initial temperature used in Connolly [6] were designed based on this idea. Besides these two types of temperature tuning, Otten & van Ginnekenet [24] suggested a tuning method in which temperature is lowered adaptively during the execution of annealing process.

From the standpoint of solution quality, we adopt Connolly's temperature tuning procedure in one of the proposed methods, SA-MDT1 (Simulated Annealing based on MDT

1). This procedure which uses trial information contributes to reduction in the number of control parameters and leads to high quality solutions. However, it usually requires considerable time to reach the frozen state of the system. Therefore, we also employ another tuning procedure to lower the temperature more quickly in the other proposed method, SA-MDT2. This second method significantly decreases computational effort, especially for large size problems, with a slight degradation in solution quality. From evaluation of test results for the various tuning procedures, Equation(**2**) was selected for use in SA-MDT2.

(d) Equilibrium test:

Each SA scheme has its own means for testing equilibrium state, i.e., testing whether the annealing process should proceed to the next temperature. Most existing schemes test the equilibrium state according to prescribed criteria independent of the problem characteristics.

For instance in Wilhelm & Ward [30] and Golden & Skiscim [10], a test as to whether the equilibrium state has been reached is conducted after a certain duration at each temperature. In both studies, this duration is called an "*epoch*", though its definition differs slightly between these two schemes. In Golden & Skiscim [10], it is represented as the *a priori* specified number of *attempted* exchanges including non-accepted exchanges, while in Wilhelm & Ward [30] it is defined as the *a priori* specified number of only *accepted* exchanges. In both methods, after the execution of each epoch, the value of the objective function is calculated, and the equilibrium test is conducted based on the current and previous values of the objective function. If the system reaches the equilibrium state, then the temperature is lowered. Otherwise, exchanges are repeated during the next epoch at the same temperature, and at the end of that epoch the equilibrium test is conducted again.

There is also a slight difference in the procedures for determining the equilibrium state between the above two schemes, as well as the definition of the epoch. In Golden & Skiscim [10], as shown in Equation(**5**), if the mean value of the objective function from the most recent epoch, j, at temperature T_i which is defined as l_j^i in this equation, is sufficiently close to any of the mean values at previous epochs, i.e., $l \in \{l_0^i, \cdots, l_{j-1}^i\}$, then the system is assumed to be at equilibrium.

On the other hand, Wilhelm & Ward [30] uses Equation(**6**) as the equilibrium test, where $\bar{f}_e$ is the mean value of the objective function during the most recent epoch at temperature T_i, and $\bar{f}_e'$ is the grand mean of the objective function for all preceding epochs at temperature T_i.

$$|l_j^i - l| \leq \epsilon \tag{5}$$

$$\frac{\bar{f}_e - \bar{f}_e'}{\bar{f}_e'} \leq \epsilon \tag{6}$$

A simpler procedure than the above was employed in Burkard & Rendl [3]. In this scheme, the number of pairwise exchanges is specified prior to the execution of the annealing process, and the equilibrium state is determined to be reached after that number of exchanges at each temperature. This fixed number, called "*rep*" in their study, is set to twice the number of facilities and multiplied by the factor 1.1, i.e., $1.1 \times 2n$ for the next *rep*.

In contrast with these two types of existing schemes, our proposed scheme uses problem-specific information in the form of the MDT. This information can be also utilized in the equilibrium test, in place of the parameter values determined prior to execution. This enables more flexible and more frequent tests since no additional calculation or processing is required for the test. Details are given in the next section.

(e) Frozen test:

Table 2: Material flows between facilities for the example problem

Facility	1	2	3	4	5	6	7	8
1	–							
2	5	–						
3	2	3	–					
4	4	0	0	–				
5	1	2	0	5	–			
6	0	2	0	2	10	–		
7	0	2	0	2	0	5	–	
8	6	0	5	10	0	1	10	–

1	2	3	4
5	6	7	8

Figure 2: Initial assignment for example problem

Two broad types of stopping criterion have traditionally been used: historical data on acceptance and rejection of exchanges during the annealing process(Wilhelm & Ward [30], Burkard & Rendl [3], Connolly [6], etc.), and information on the annealing schedule and its initial and final temperatures (Golden & Skiscim [10], etc.).

Ward [30] was stated as follows:

It was determined from preliminary results in this study that regardless of the criterion selected and incorporated into the annealing process, the total number of iterations until the system reaches the frozen state heavily depends on the problem size. From these results, the proposed scheme employs more straightforward stopping criterion based on the total number of exchanges.

3 Simulated annealing combined with MDT

3.1 Procedures of annealing process

(1) Pairwise exchanges using MDT

The most distinctive feature of the proposed methods is the incorporation of a problem-specific structure, MDT, into the SA methodology. MDT can be utilized in selecting pairwise exchange and test for equilibrium state. In choosing the pairwise exchange at each iteration, MDT is applied in the proposed methods. The criterion adopted by MDT defines the desirability of moving one facility to different directions. This table contains the cost changes that would result from moving of a facility to each alternative location, i.e., left, right, up and down, in relation to its current location.

A procedure for calculating each entry in the MDT is shown next through an example in which eight facilities are to be assigned to eight locations. The material flow between any combination of facilities, c_{ij}, is shown in Table **2**, and the initial assignment is illustrated in Figure **2**. In this example, it is assumed that the distance to be traveled between two locations can be approximated by the distance between their centers measured along directions parallel to the sides of the locations, and the unit of distance is the length of the side of a location. Thus, if a load travels to an adjacent location, the distance is one, and if it travels to another location adjacent to an adjacent location, the distance is two.

Based on the above assumptions, the desirability of moving a given facility to adjacent

Table 3: Move Desirability Table for the initial assignment

Facility	Left	Right	Up	Down
1	–	+16	–	−4
2	0	−4	–	−2
3	0	0	–	0
4	+3	–	–	+15
5	–	+16	−2	–
6	0	−4	−12	–
7	−5	+5	−11	–
8	+12	–	+10	–

neighborhood positions, i.e., each entry in the MDT, is determined. For example, in Figure **2**, if Facility 7 is moved one step to the left, the effect on material handling costs is to save one distance unit for each load traveling between facilities assigned to columns on the left of Facility 7, i.e., Facilities 1, 2, 5 and 6, and to increase by one distance unit for each load traveling between facilities assigned to columns on the right of Facility 7 and facility in the same column, i.e., Facilities 4, 8 and 3. Using information in Table **2** and Figure **2**, the net decrease on the cost by the move would be $(0+2+5)-(0+2+10)=-5$, meaning that if Facility 7 is moved to the left the total cost will be increased by 5. Thus, the calculated number represents a "move desirability index". The desirability of moving all the facilities to the right, up and down are calculated in the same way. The result of such calculations for the state shown in Figure **2** is presented in Table **3**.

The entries in Table **3**, however, do not represent the actual cost reduction to be obtained by the next exchange since they reflect only the desirability of moving a single facility. The candidates for moves are those which have positive values in MDT. The facility that presents the maximum cost improvement from the entries of the MDT, is selected as facility to exchange without consideration of the value of the other facility in the exchange pair. At this time, its indicated move defines the other facility for the exchange. For example, in Table **3**, moving Facility 1 to the right is the entry with maximum desirability. It means Facility 1 shifts one step to the right and is replaced by Facility 2.

Utilizing one of the useful features of tabu search, in order to prevent an infinite loop, the same exchange is not allowed to take place more than one time at each temperature by defining a forbidden set which consists of all accepted pairs at the same temperature. When a facility with a negative value on the move desirability index is replaced as the other pair in an iteration, the move desirability of this facility to the former location will appear as a large positive entry in the next MDT. In this situation, based only on the value of the desirability index, it may be highly possible to select the same pair which was already exchanged before, and therefore this may lead to an infinite loop.

Whether the present exchange is accepted or not is tested according to the general acceptance procedure. At this time, the resulting change in the cost, Δf, is computed efficiently by applying the move desirability indexes in MDT. For example, in the case of exchange of Facilities 1 and 2, $\Delta f=(16-5)+(0-5)=6$ where the value of 5 is the flow material between Facilities 1 and 2.

In the case of accepting a pairwise exchange, the MDT must be revised for the next exchange of facilities. At this time, the revised MDT can also be generated easily using the structure of MDT. It should be clear that most of the entries are still valid for the succeeding state of the table. For example, in the state of Figure **2**, if the locations of Facility 6 and

Facility 5 are exchanged, those entries in Table **3** which need to be changed to reflect this exchange are the 'left' entries of Facilities 2, 5, and 6, and the 'right' entries of Facilities 1, 5 and 6. The other entries remain without changes. This is because, for example, the facilities located on the left side of Facility 7, i.e., Facilities 5 and 6, are still assigned in the left side, event hough their locations have been exchanged.

Since the state space coverage is not obvious under the proposed state transition scheme, the resulting SA procedure may have lost the GC1 property.

(2) Initial temperature setting

Using the same procedure of Connolly [6], the initial and final temperatures are determined using information obtained in trials prior to the annealing process. In these trials, 100 random pairwise exchanges are performed without regard to the resulting changes in the cost. From the results, the minimum value Δf_{min} and the maximum value Δf_{max} for the changes in cost are calculated for these exchanges. Using these values, the initial temperature, T_0 and the final temperature, T_f are then set according to Equations (**7**) and (**8**), respectively.

$$T_0 = \Delta f_{min} + \frac{1}{10}(\Delta f_{max} - \Delta f_{min}) \tag{7}$$

$$T_f = \Delta f_{min} \tag{8}$$

(3) Tuning temperature

In SA-MDT1, the temperature is controlled by Connolly [6]'s procedure, i.e., Equation(**4**). In the equation, M refers to the number of pairwise exchanges examined, and is calculated by Equation (**9**) in the present study[1].

$$M = 50 \times \frac{n(n-1)}{2} \tag{9}$$

In the second method, SA-MDT2, in order to lower the temperature faster, the temperature is controlled by Equation(**2**) as mentioned in Section **2.2**.

(4) Equilibrium test

The MDT is also utilized in the equilibrium test. As mentioned in Section **2.2**, each entry of the MDT represents the degree of desirability of moving to a neighborhood location. If no facilities remain with positive values for the move desirability index in the MDT, it may be considered that a steady state has been reached. Therefore, our proposed methods adopt this state of the MDT as the equilibrium state of the system. This procedure allows us to test for equilibrium anytime during the annealing process, with no additional effort needed beyond normal processing of pairwise exchanges.

(5) Frozen test

As mentioned in Section **2.2**, the proposed methods employ a simple, straightforward stopping criterion wherein the total number of iterations from the start of the annealing process is compared with a certain number corresponding to the problem size. The user can easily control the CPU time by selecting an appropriate value for this number. The annealing process terminates when either the total number of pairwise exchanges exceeds M, or the temperature reaches the final temperature T_f.

3.2 Algorithm

The complete algorithms for the annealing process procedures mentioned above are described here. The flowchart of one of the proposed methods, SA-MDT1, is illustrated in Figure **3**.

[1]This equation is also same as the one used in Connolly [6].

STEP 0: Determine an initial assignment a^0, and calculate T_0 and T_f. Concerning the initial assignment, the current existing layout may be used, or it may be generated at random. In the numerical experiments that are mentioned in Section 4, the initial assignment is input in order to compare the results with those of the existing schemes. After a hundred trials, T_0 and T_f are calculated by Equations (7) and (8), respectively. The temperature T is initialized to T_0 and the iteration counter m set to 0.

STEP 1: Compute the total cost of initial assignment $f(a^0)$, set the temporary solution $a^* = a^0$, and the temporary least cost $E = f(a^0)$.

In executing SA-MDT1, the tuning parameter on temperature β and M are calculated by Equations(4) and (9), respectively, while the parameter α is determined for SA-MDT2 by the user.

STEP 2: Generate the "Move Desirability Table" for the current assignment in the manner described previously.

STEP 3: Pairwise exchanges selection procedure.

a) Select the facility with the maximum entry of the MDT. If the related exchange is the same as the former exchanges at the current temperature, go to STEP 3b; otherwise go to STEP 4.

b) Ignore this exchange and return to STEP 3a.

STEP 4: Equilibrium test procedure

a) If the maximum entry of the MDT is less than or equal to zero, then go to STEP 4b; otherwise go to STEP 4c.

b) Change the temperature according to Equation (4) for SA-MDT1, or Equation (2) for SA-MDT2, and return to STEP 3a.

c) Increment the iteration counter $m = m + 1$, calculate the resulting change in cost Δf and go to STEP 5

STEP 5: Exchange acceptance process

a) If $\Delta f \leq 0$, then go to STEP 5b; otherwise go to STEP 5c

b) Compute $P(\Delta f) = e^{\frac{\Delta f}{T}}$ and select a random digit[2] $x \sim U(0,1)$. If $P(\Delta f) > x$, then go to STEP 5c; otherwise return to STEP 3a.

c) Accept this pairwise exchange, replace new assignment, calculate the new cost ($f(a) = f(a_{-1})^3 - \Delta f$) and go to STEP 6.

STEP 6: Change the temporary solution. If $E > f(a)$, then set $E = f(a)$, and $a^* = a$; go to STEP 7.

STEP 7: If m, the number of pairwise exchanges examined, is greater than M, go to STEP 8; otherwise return to STEP 2.

STEP 8: STOP.

4 Numerical experiments

4.1 Test method

Numerical experiments were conducted to evaluate the proposed methods. To compare the proposed methods with other existing methods, we used the standard data for facility layout problems which were first presented by Nugent et al. [23] as well as two types of larger problems. Burkard et al. [4] offered a QAP library in which a number of problem

[2] $U(0,1)$ means the uniform distribution with the range [0, 1].

[3] The total cost of previous assignment.

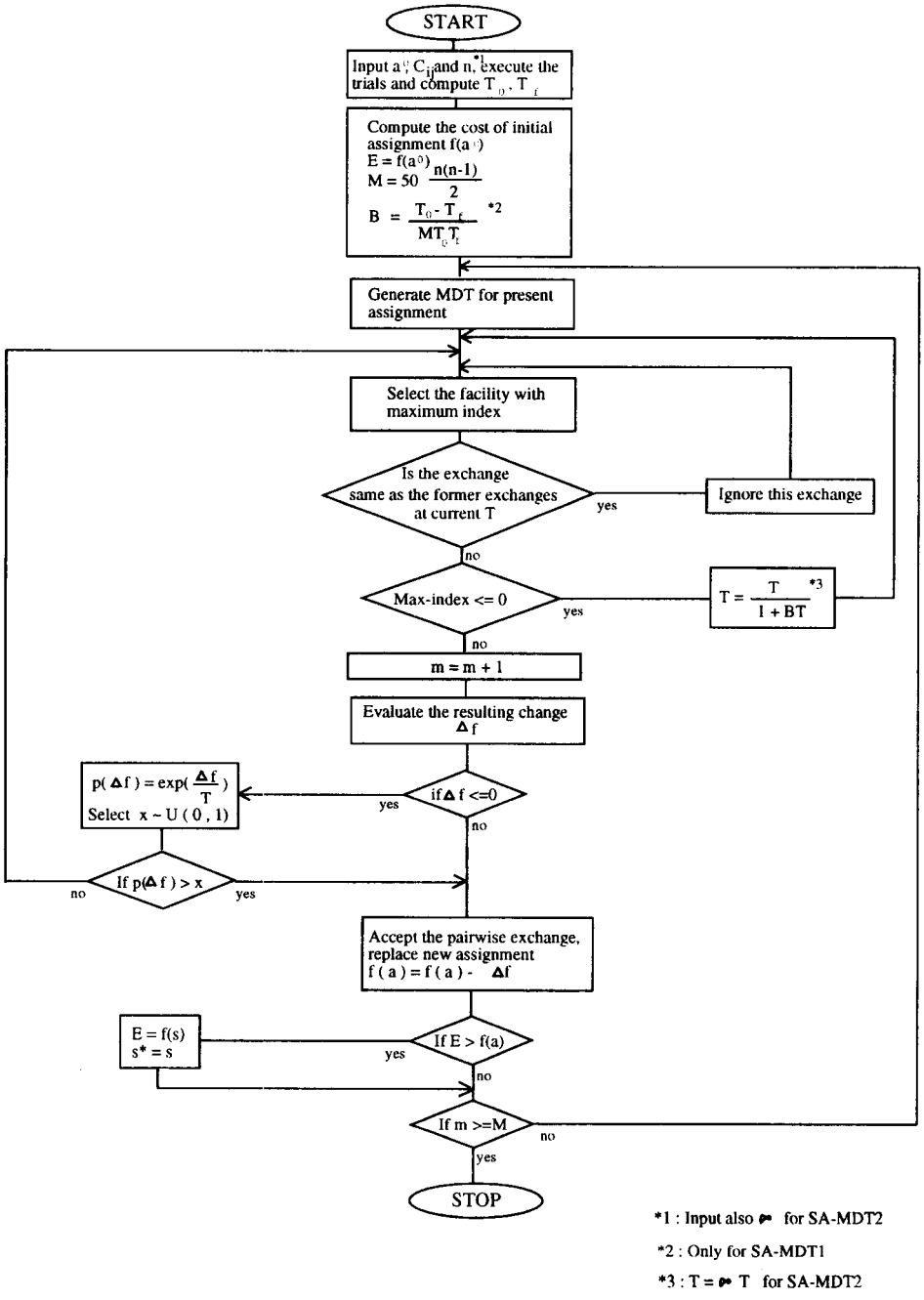


Figure 3: Flow chart of SA-MDT1

instances was stored. So far, the Nugent et al. were most common standard data utilized to evaluate heuristic methods suggested for QAP.

In particular, the proposed methods were compared with Wilhelm & Ward [30] and Connolly [6], which are known most effective traditional SA schemes for solution of QAPs. The computer programs of the proposed methods, Wilhelm & Ward and Connolly were written in C language and implemented on a workstation, SUN SPARC Station 2. The pseudo random numbers were generated by rand() function. The Unix^{TM} online manual for rand() states that "the spectral properties [...] leave a great deal to be desired" and that "the low bits of the numbers generated are not very random [...] in particular the lowest bit alternates between 0 and 1." An implicit premise for the numerical results in this paper is, therefore, that the quality of random numbers is unimportant for the performance comparison among SA and other schemes so long as the same quality is kept throughout.

All the above methods were tested for various sizes of problems, i.e., n = 6, 8, 12, 15, 20, 30, using five different initial assignments which were also used in Wilhelm & Ward [30] and others. For larger problems, SA-MDT2 was compared with the other methods. Three problems for n = 50 and one for n = 100 generated in this study in which flow materials data were generated from uniform distributions. Furthermore, for n = 100, two problems reported in [5] were also used. The number of runs for each initial assignment was limited to five for the data up to n = 30 as in the experiments of Wilhelm & Ward [30], and to one for the data of n = 50, 100.

The method of common random numbers [1] was not used because the gain in computation time would not seem to compensate the increase in programming complexity.

Concerning the parameter settings, the Wilhelm & Ward method was employed with e = 15, ϵ = 0.01 and N' = 100 for all problem sizes, as they recommended, while Connolly method was run with MAXFAIL = $\frac{n(n-1)}{100}$, same as Connolly's paper. In one of the proposed methods, SA-MDT2, the tuning parameter for temperature α was set at 0.97 from the results of preliminary experiments.

In addition, we also compared the quality of the results for the proposed methods with the results of other methods as reported in the literature [3, 30]. There are many methods by which computational tests were conducted using the same standard data of Nugent et al.. Here, we compared with QAPH4 [3], CRAFT [2], Biased Sampling [23] and the Revised Hillier method [25].

4.2 Computational Results

(1) Comparison with traditional SA methods

Results of the numerical experiments are shown in Tables 4, 5 and 6. These tables compare the total cost and CPU times obtained by the proposed methods and by the Wilhelm & Ward and Connolly methods. The results obtained in this study for the Wilhelm & Ward method was nearly the same results as those reported in their paper [30], while for Connolly method the results are different from those reported. This is due to different test methods used and some details which might not be given in his paper. For instance, the number of runs for an initial assignment (unknown for us) was limited to 100 in his study, whereas we tested all of SA methods by five runs for five specific initial assignments presented in [23]. This test method was reported in Nugent et al. and commonly used in the literature.(For instance see [3, 25, 30].)

Table 4 shows the computational results for relatively small size problems in terms of the mean and best cost for solutions and the mean CPU times for every five runs of each initial assignment. As seen in this table, SA-MDT2, which employs a fast cooling function,

Table 4: Comparison of results for $n = 6, 8, 12, 15$ problems (CPU time in second)

Prob. size	Init. assign. no.	W & W Cost Best	W & W Cost Mean	W & W CPU time	Connolly Cost Best	Connolly Cost Mean	Connolly CPU time	SA-MDT1 Cost Best	SA-MDT1 Cost Mean	SA-MDT1 CPU time	SA-MDT2 Cost Best	SA-MDT2 Cost Mean	SA-MDT2 CPU time
6 (43)*	1	43	43.0	0.98	43	43.0	0.11	43	43.0	0.20	43	43.0	0.03
	2	43	43.0	0.95	43	43.0	0.07	43	43.0	0.19	43	43.0	0.03
	3	43	43.0	1.07	43	43.0	0.10	43	43.0	0.21	43	44.8	0.06
	4	43	43.0	1.09	43	43.0	0.12	43	43.0	0.21	43	45.0	0.07
	5	43	43.0	0.99	43	43.0	0.12	43	43.0	0.21	43	44.8	0.06
	Ave.	43.0	43.0	1.02	43.0	43.0	0.11	43.0	43.0	0.20	43.0	44.1	0.05
	S.D.	0.0	0.0	0.15	0.0	0.0	0.04	0.0	0.0	0.02	0.0	1.8	0.03
8 (107)*	1	107	107.0	1.77	107	107.4	0.27	107	107.0	0.63	107	109.4	0.20
	2	107	107.0	1.52	107	107.4	0.24	107	107.0	0.65	107	109.8	0.19
	3	107	107.0	1.76	107	109.0	0.17	107	107.0	0.64	107	108.2	0.17
	4	107	107.0	1.91	107	109.2	0.25	107	107.0	0.67	109	109.0	0.17
	5	107	107.0	1.76	107	107.4	0.33	107	107.0	0.64	107	107.8	0.15
	Ave.	107.0	107.0	1.74	107.0	108.1	0.25	107.0	107.0	0.65	107.4	108.8	0.18
	S.D.	0.0	0.0	0.36	0.0	1.71	0.14	0.0	0.0	0.02	0.9	2.8	0.07
12 (289)*	1	289	292.4	4.09	291	294.2	1.22	289	289.8	3.34	293	295.6	0.93
	2	289	292.8	4.93	293	297.8	1.28	289	290.6	3.32	289	292.2	0.68
	3	289	292.6	4.95	293	296.6	1.29	289	289.4	3.32	289	295.4	0.74
	4	289	291.8	4.55	293	294.6	1.45	289	289.4	3.22	293	297.0	0.85
	5	289	292.2	3.93	293	296.2	1.29	289	289.8	3.29	289	293.4	0.56
	Ave.	289.0	292.4	4.49	292.6	295.9	1.31	289.0	289.8	3.29	290.6	294.7	0.75
	S.D.	0.0	2.3	1.05	0.89	3.09	0.57	0.0	1.5	0.14	2.2	4.2	0.25
15 (575)*	1	576	579.6	9.85	580	582.8	3.23	575	575.8	7.78	576	576.8	1.46
	2	575	578.8	7.15	583	585.2	2.78	576	576.0	7.80	576	580.8	1.78
	3	576	578.2	9.89	580	582.8	4.39	575	575.6	7.71	575	582.8	1.94
	4	575	579.4	6.56	576	581.8	3.35	575	575.2	7.86	576	585.4	2.23
	5	576	579.6	6.13	576	585.0	2.71	575	575.4	7.78	575	585.2	2.01
	Ave.	575.6	579.1	7.91	579.0	583.5	3.29	575.2	575.6	7.79	575.6	582.2	1.88
	S.D.	0.5	3.4	2.96	3.0	4.0	1.33	0.5	0.5	0.14	0.6	6.7	0.39

*: Best known solution

is the fastest to obtain solutions among all four methods compared here. Although this method gives slightly lower quality solutions than the W &W and SA-MDT1 methods, its performance is superior to Connolly method in the cases of $n = 12$, 15. Concerning the solution quality, SA-MDT1, Wilhelm & Ward and Connolly reached the best known solution for $n = 6, 8$ in all initial assignments. In the case of $n = 12, 15$, SA-MDT1 obtained higher solution quality than Wilhelm & Ward as well as Connolly. As the same tendency was identified in larger size problems, the standard deviation of cost values obtained by SA-MDT1 is smaller than those of Wilhelm & Ward and Connolly methods. This fluctuation in cost for the same initial assignment was due to the sequence of random numbers used. This means that SA-MDT1 does not depend so much on given initial solutions or events occurred by chance compared with other methods, and therefore can derive good solution consistently. From these results, SA-MDT1 is superior to SA-MDT2, Wilhelm & Ward and Connolly methods in terms of solution quality, and is faster than Wilhelm & Ward method for relatively small size problems.

The computational results for medium size problems are shown in Table 5. For the case of $n = 20, 30$, the solutions obtained by SA-MDT1 are better than those of Wilhelm & Ward as well as those of Connolly and their differences of values in the objective function are highly significant with level less than 0.005 in every case, although this method consumes considerable CPU time. That is due to the fact that in SA-MDT1 the system is cooled very slowly in order to search for solutions near an unknown optimal temperature. This slow change in temperature generates a very high quality solution at the cost of more computational time.

On the other hand, SA-MDT2 requires less CPU time to derive a good solution than both Wilhelm & Ward and Connolly methods. As the size of problems increases, the difference in computational times between these methods also increases. Regarding the solution quality, as may be seen in Table 5 for medium size problems, SA-MDT2 is superior to both Wilhelm

Table 5: Comparison of results for $n = 20, 30$ problems (CPU time in second)

Init. sol. no.	W & W Cost Best	W & W Cost Mean	W & W CPU time	Connolly Cost Best	Connolly Cost Mean	Connolly CPU time	SA-MDT1 Cost Best	SA-MDT1 Cost Mean	SA-MDT1 CPU time	SA-MDT2 Cost Best	SA-MDT2 Cost Mean	SA-MDT2 CPU time
$n = 20; (1285)^*$												
1	1297	1308.0	12.9	1301	1308.0	12.1	1287	1295.0	24.8	1301	1307.8	9.4
2	1299	1307.4	12.8	1300	1317.6	7.29	1290	1293.6	25.4	1292	1306.6	9.6
3	1302	1307.4	13.2	1301	1311.8	12.3	1285	1294.2	25.6	1287	1302.4	12.3
4	1287	1308.2	12.6	1294	1311.8	8.90	1287	1293.6	25.1	1285	1300.8	10.3
5	1300	1308.6	13.2	1287	1305.0	10.7	1297	1297.6	25.4	1299	1302.8	14.0
Ave.	1297.0	1307.9	12.9	1296.6	1310.8	10.3	1289.2	1294.8	25.2	1292.8	1304.0	11.1
S.D.	5.9	9.0	2.0	6.1	11.1	4.8	4.7	4.2	0.9	7.1	9.5	2.3
$n = 30; (3064)^*$												
1	3079	3124.0	58.0	3097	3127.8	75.6	3081	3095.8	136.0	3080	3099.8	30.2
2	3119	3125.8	56.2	3109	3118.2	73.9	3078	3094.6	135.0	3100	3114.4	25.7
3	3113	3117.4	57.6	3082	3108.8	46.6	3077	3092.8	134.0	3064	3091.4	28.5
4	3068	3126.6	67.9	3091	3107.6	56.7	3086	3096.6	135.0	3073	3092.0	26.4
5	3117	3127.2	68.2	3098	3108.2	69.9	3080	3093.2	135.0	3091	3109.4	33.9
Ave.	3099.2	3124.2	61.6	3095.4	3114.1	64.5	3080.4	3095.6	135.0	3081.6	3101.4	28.9
S.D.	23.8	18.7	24.1	9.9	17.7	24.4	3.5	8.6	3.2	14.2	18.3	5.9

*: Best known solution

Table 6: Comparison of results for $n = 50$ and $n = 100$ problems (CPU time in minute)

Prob. size	Best S.D	W & W Cost Best	W & W Cost Mean	W & W CPU time	Connolly Cost Best	Connolly Cost Mean	Connolly CPU time	SA-MDT2 Cost Best	SA-MDT2 Cost Mean	SA-MDT2 CPU time
50	50130*	50238	50465	4.47	50208	50346	10.3	50130	50225	7.30
	S.D.		128	1.9		91	2.6		71	1.5
	50532*	50681	50871	4.3	50558	50721	8.8	50532	50596	6.0
	S.D.		120	2.8		167	3.4		39	1.8
	49825*	50002	50133	5.1	49976	50093	11.0	49825	49897	7.2
	S.D.		129	2.2		125	2.0		74	1.7
100	281408*	282621	283092	34.9	281933	282397	171.3	281408	281681	103.3
	S.D.		305	8.8		294	33.1		232	15.5
	152014**	154178	155212	35.6	152822	153383	149.1	152402	152704	76.9
	S.D.		600	7.1		371	37.8		200	14.7
	153900**	155488	156432	39.6	154708	154916	177.2	154196	154516	76.6
	S.D.		572	2.0		163	47.8		224	14.9

*: Best known solution obtained in this study.
**: Best known solution reported in [5].

& Ward and Connolly, and there are only slight differences between SA-MDT1 and SA-MDT2. From these results, considering running time as well as solution quality, it may be suggested that SA-MDT2 is appropriate for larger problems. Therefore, SA-MDT1 was not tested for larger sizes of problems.

The results of numerical experiments for larger problems, i.e., $n = 50, 100$, are given in Table **6**. As it can be seen, SA-MDT2 can yield higher quality solutions than both Wilhelm & Ward and Connolly at the cost of more CPU time than Wilhelm &Ward method (but less CPU time than the Connolly method). This is due to the higher computational efforts which is required for generating MDT for extremely large problems. We note that in two problems of [5], SA-MDT2 could not reach the best solutions obtained by Tabu Search methods.

Table **7** presents the comparison results between SA methods and Tabu Search methods reported in [5]. As we mentioned above the best solutions obtained by Tabu Search methods are superior to those of SA methods, including SA-MDT2. Since the Tabu Search methods in [5] were run in a connected machine system, a direct CPU time comparison cannot be conducted. It is obvious that running one iteration of the Tabu Search method is more time consuming than that of the traditional SA. In addition, the number of iterations required to gain such good results by the Tabu Search method is also nearly four times larger than that of the traditional SA methods as shown in this table. In comparison with SA-MDT2, we cannot verify which method is faster without calculating the CPU time required for one

Table 7: Comparison between SA methods and Tabu Search methods

suboptimal methods	100b*		100c*	
	Best	No. of Iterations	Best	No. of Iterations
Par-tabu**	152014	224296	153910	182675
Aug. Par-tabu**	152014	199882	153900	274480
W & W***	154178	42281	155488	48218
Connolly***	152822	181766	154708	216071
SA-MDT2***	152402	47156	154196	51583

*: Denoted by 100a and 100b in [5].
**: Results are reported in [5].
***: Results are obtained in this study.

iteration in the same machine. However, as seen in Table 7 the total number of iterations for Connolly(traditional SA) is less than both Tabu Search methods and the CPU time of SA-MDT2 is about a half Connolly's CPU time.

Finally, Figure 4 shows the mean improvement of the proposed methods on traditional SA methods and relative CPU time required for such improvement.

(2) Comparison with other methods

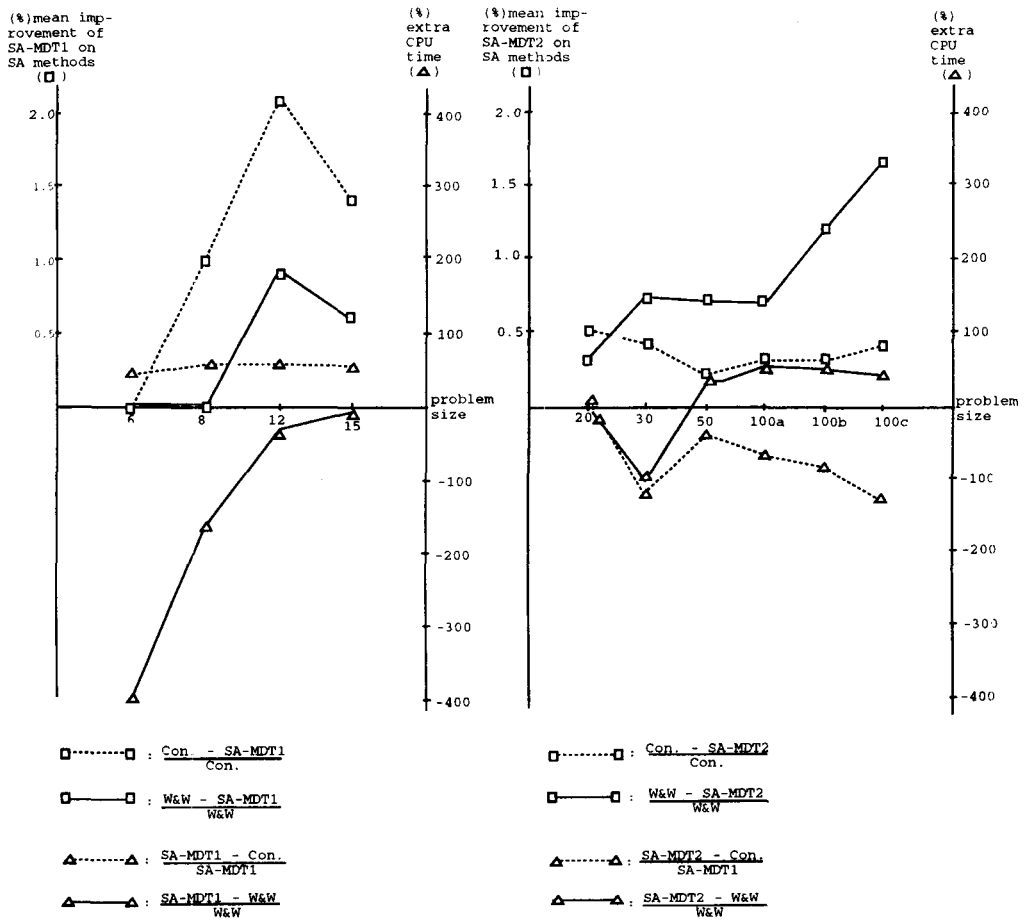
As mentioned previously, the performance of many methods has been tested using the same standard data for facility layout problems. Thus, we can also compare the solution performance of the proposed methods with other methods using the original results cited in the literatures [3, 23, 30]. Here, we can compare only by solution quality, not by CPU time, since every method in the literature was run on different computers from the one which was used in this study.

Table 8 shows the solution quality of the proposed methods in comparison with another SA method, i.e., QAPH4(the method of Burkard & Rendl [3]), and several traditional heuristic methods for layout problem, i.e., CRAFT [2], Biased Sampling [23] and the revised Hillier method [25] as applied to the standard data of Nugent et al. [23]. It can be seen that solution qualities obtained by SA-MDT1 and SA-MDT2 are generally better than the others.

In particular, it can be seen that there is large difference in the existing methods between the best cost and the mean cost for each method, while in the proposed methods, especially for SA-MDT1, the value of the mean cost is closer to the best one in each case. This means, as mentioned previously, that the proposed methods are not as sensitive to the initial assignment and random number sequence. Therefore, only one or a few executions are needed to obtain a good quality solution for a given problem.

5 Conclusion

In the present study, we proposed a new approach which combines simulated annealing with a problem specific structure. The facility layout planning problem formulated as QAP was treated as an application of this approach, and the "*Move Desirability Table*" was employed as the structure that is specific to this problem information. Using information from the MDT pairwise exchanges selection and the equilibrium test are conducted. In addition, several procedures for the annealing process were suggested from analysis and comparison with existing methods. Adopting MDT and these suggestions, fewer control parameters



(a) : Smaller cases (compared with SA-MDT1)

(b) : Larger cases (compared with SA-MDT2)

Figure 4: Relative performance of proposed methods

need be defined and hence, less computational effort is required for parameters setting.

Based on the above concepts, we proposed two methods which differed only in cooling schedule and hence affected in computational time. In the first method, SA-MDT1, information obtained during trials prior to the annealing process is utilized, and the system is cooled very slowly near an unknown critical temperature, thereby making this method appropriate for relatively small size problems.

On the other hand, the second method, SA-MDT2, uses a simpler procedure for temperature tuning, i.e., it determines the next temperature according to a prescribed cooling function. This function used in the present study cools faster, and thereby contributes to saving computational time.

In order to evaluate the proposed methods, we conducted a number of numerical computations using the standard problems of Nugent, et al. [23] in the facility layout problem. Based on the computational results, SA-MDT1 was shown to obtain higher quality solutions than all other SA methods for all size of problems, but the expense of relatively high computational effort in the case of large scale problems. In addition, this method was less

Table 8: Comparison of results for QAP suboptimal solution methods

Prob. size	Best known solut.	SA-MDT1 Best cost	SA-MDT1 Mean cost	SA-MDT2 Best cost	SA-MDT2 Mean cost	QAPH4* Best cost	QAPH4* Mean cost
6	43	43	43.0	43	44.1	–	–
8	107	107	107.0	107	108.8	–	–
12	289	289	289.8	289	294.7	295	300.5
15	575	575	575.4	575	582.5	580	593.5
20	1285	1285	1294.2	1285	1304.0	1299	1328.0
30	3064	3077	3095.6	3064	3101.4	3091	3140.5

Prob. size	Best known solut.	CRAFT** Best cost	CRAFT** Mean cost	Biased Sampl.** Best cost	Biased Sampl.** Mean cost	Revised Hillier** Best cost	Revised Hillier** Mean cost
6	43	43	44.2	43	43.6	43	43.0
8	107	107	110.0	107	108.2	107	107.4
12	289	289	296.2	289	294.8	291	298.4
15	575	583	606.0	575	581.0	575	582.2
20	1285	1324	1339.0	1312	1321.0	1297	1324.6
30	3064	3090	3197.8	3106	3124.0	3070	3114.2

–: Not applied.
*: Results are reported in [3].
**: Results are reported in [30].

sensitive to given initial assignments and random number seeds. In the case of large problems, the other proposed method, SA-MDT2 produced solutions of quite satisfactory quality with relatively less computational time. Therefore, considering solution quality as well as computational time, SA-MDT1 may be considered efficient for dealing with small size of layout problems, and SA-MDT2 for larger problems.

Finally, although the simulated annealing algorithms proposed here are for the facility layout problem, it is possible to apply these methods to other combinatorial optimization problems by finding rules or structure based on the characteristics of the problem, such as the MDT used in the facility layout problem. It may provide higher quality solutions with increasing efficiency in comparison with other heuristic methods available in the literature.

Acknowledgment

This research was supported in part by the Grants in Aid for Scientific Research(C), No.04832017, the Ministry of Education, Science and Culture, Japan.

References

[1] Bratly,P., Fox,B.L. and Schrage,L.E. : *A Guide to Simulation*, 2ed., Springer-Verlagf, 1987.

[2] Buffa,E.S., Armour,G.C. and Vollman,T.E. : "Allocating facilities with CRAFT", *Harvard Business Review*, Vol.**42**, No.**2** (1964), 136 – 158.

[3] Burkard,R.E. and Rendl,F. : "A thermodynamically motivated simulation procedure for combinatorial optimization problems", *European Journal of Operational Research*, Vol.**17** (1984), 169 – 174.

[4] Burkard,R.E., Karisch, S. and Rendl,F. : "QAPLIB-A quadratic assignment problem library", *European Journal of Operational Research*, Vol.**55** (1991), 115 – 119.

[5] Chakrapani,J. and Skorin-Kapof,J. : "Massively parallel tabu search for the quadratic assignment problem", Working Paper HAR-91-06, Harriman School for Management and Policy, State University of New York at Stony Brook, January 1992.

[6] Connolly,D.T. : "An improved annealing scheme for the QAP", *European Journal of Operational Research*, Vol.**46** (1990), 93 – 100.
[7] Edwards,H.K., Gillet,B.E. and Hale,M.E. : "Modular allocation technique (MAT)", *Management Science*, Vol.**17**, No.**3** (1970), 161 – 169.
[8] Eglese,R.W. : "Simulated annealing: A tool for operational research ", *European Journal of Operational Research*, Vol.**46** (1990), 271 – 281.
[9] Francis,R.L. and White,J.A. : *Facility layout and location : An analytical approach*, Prentice Hall, Englewood Cliffs, NJ, 1974.
[10] Golden,B.L. and Skiscim,C.C. : "Using simulated annealing to solve routing and location problems", *Naval Research Logistic Quarterly*, Vol.**33** (1986), 261 – 279.
[11] Graves,G.W. and Whinston,A.B. : "An algorithm for the quadratic assignment problem", *Management Science*, Vol.**15**, No.**4** (1970), 453 – 471.
[12] Hajek,B. : "Cooling schedules for optimal annealing", *Mathematics of Operations Research*, Vol.**13**, No.**2** (1988), 311 – 329.
[13] Hillier,F.S. : "Quantitative tools for plant layout analysis", *Journal of Industrial Engineering*, Vol.**14**, No.**1** (1963), 33 – 40.
[14] Hillier,F.S. and Connors,M.M. : "Quadratic assignment problem algorithms and the location of indivisible facilities", *Management Science*, Vol.**13**, No.**1** (1966), 42 – 57.
[15] Khalil,T.M. : "Facilities relative allocation technique (FRAT)", *International Journal of Production Research*, Vol.**11**, No.**2** (1973), 183 – 194.
[16] Kirkpatrick,S., Gelatt,Jr.,C.D. and Vecchi,M.P. : "Optimization by simulated annealing", *Science*, Vol.**220** (1983), 671 – 680.
[17] Koopmans,T. and Beckmann,M. : "Assignment problems and the location of economic activities ", *Econometrica*, Vol.**25** (1957), 53 – 76.
[18] Lee,R.C. and Moore,J.M. : "CORELAP – Computerized relationships layout planning",*Journal of Industrial Engineering*, Vol.**18**, No.**3** (1967), 195 – 200.
[19] Ligget,R.S. : "The quadratic assignment problem: An experimental evaluation of solution strategies", *Management Science*, Vol.**27**, No.**4** (1981), 442 – 458.
[20] Lundy,M. and Mees,A. : "Convergence of an annealing algorithm", *Mathematical Programming*, Vol.**34** (1986), 111 – 124.
[21] Metropolis,N., Rosenbluth,A., Rosenbluth,N., Teller,A. and Teller,E. : "Equation of state calculations by fast computing", *Journal Chemical Physics*, Vol.**21** (1953), 1087–1092.
[22] Mitra,D., Romeo,F. and Sangiovanni-Vincentelli,A.L. : "Convergence and finite-time behavior of simulated annealing", *Advances in Applied Probability*, Vol.**18** (1986), 747–771.
[23] Nugent,C.E., Vollman,R.E. and Ruml,J. : "An experimental comparison of techniques for the assignment of facilities to locations", *Operations Research*, Vol.**16** (1968), 150–173.
[24] Otten,R.H.J.M. and van Ginneken,L.P.P.P. : *The annealing algorithm*, Kluwer,1989.
[25] Picone,C.J. and Wilhelm,W.E. : "A perturbation scheme to improve Hillier's solution to the facilities layout problem", *Management Science*, Vol.**30**, No.**10** (1984), 1238 – 1249.
[26] Sahni,S. and Gonzales,T. : "P-complete approximation problems", *Journal of the Association of Computing Machinery*, Vol.**23** (1976), 555 – 565.
[27] Scriabin,M. and Virgin,C.R. : "A cluster-analytic approach to facility layout", *Management Science*, Vol.**31**, No.**1** (1985), 33 – 49.
[28] Skorin-Kapof,J. : "Tabu search applied to the quadratic assignment problem", *ORSA*

Journal on Computing, Vol.**2**, No.**1** (1990), 33 – 45.

[29] Vollmann,T.E., Nugent,C.E. and Zartler,R.L. : "A computerized model for office layout", *Journal of Industrial Engineering*, Vol.**19**, No.**7** (1968), 321 – 329.

[30] Wilhelm,M.R. and Ward,T.L. : "Solving quadratic assignment problem by 'Simulated Annealing' ", *IIE Transactions*, Vol.**19**, No.**1** (1987), 107 – 119

Takao Enkawa : Department of IE and Management,
Faculty of Engineering, Tokyo Institute of Technology.
2-12-1, Ookayama, Meguro-ku, Tokyo 152, Japan.