S Y S T E M S
R E S E A R C H
C E N T E R

Class: Computerized LAyout Solutions Using Simulated Annealing

by I. Minis, G. Harhalakis,
S. Jajodia and J.M. Proth

TR 90-71

CLASS: Computerized LAyout Solutions using Simulated annealing

Ioannis Minis [1]

George Harhalakis [2]

Satish Jajodia[1]

Jean Marrie Proth[3]

[1] Department of Mechanical Engineering
[2] Department of Mechanical Engineering and Systems Research Center,
University of Maryland, College Park, MD 20742
[3] INRIA-Lorraine, Metz, France

ABSTRACT

A new method (Computerized LAyout Solutions using Simulated annealing
- CLASS) that considers the inter-cell and intra-cell layout problems in a
cellular manufacturing environment is presented. It addresses the relative
placement of equidimensional manufacturing entities within a discrete
solution space in an attempt to minimize the total material flow (cost)
between these entities. An approach to accommodate the relative sizes of the
entities is also presented. The method is based on Simulated Annealing,
which has been successfully applied for the solution of combinatorial
problems. A major advantage of this technique is the insensitivity of the
final solution to the initial conditions. In addition, some important
practical issues such as intra-cell layout of machines in pre-determined
configurations (e.g., row-wise or circular arrangements), have been
addressed. Several comparisons were made with some of the existing
approaches for facility layout, such as CRAFT, HC63-66, etc., that yielded
results of equal or better quality for each of eight classical test problems.

1. Introduction

The production function of a manufacturing company is significantly affected by the layout of its manufacturing shop. While a well designed layout can considerably improve the efficiency of the shop, a poor one can lead to increased work-in-process inventory, overload the materials handling system and contribute to inefficient set-ups, longer queues, etc. (Ham *et al.* 1985).

The cellular layout approach has been recognized as an effective means of enhancing the efficiency of a discrete parts manufacturing shop. The design problem of a cellular facility is formulated in three distinct stages: i) grouping of the production equipment into cells, ii) allocation of the machine cells to areas within the shop-floor (inter-cell or facility layout), and iii) layout of the machines within each cell (intra-cell or machine layout). This paper does not address the first design stage which has been considered by numerous studies in the past. An efficient grouping method has been presented by the authors in Jajodia *et al.* (1990). Assuming that the composition of the cells is known, the problems of inter-cell and intra-cell layout are considered here. The latter are critical since their solutions dictate the material flow patterns and costs within and between cells.

A comprehensive survey of the existing methods for facility (inter-cell) layout can be found in Kusiak and Heragu (1987). The most typical formulation considers the Quadratic Assignment Problem (QAP) for n facilities and m locations ($m \geq n$). Both optimal and suboptimal algorithms have been proposed for its solution. Given that the QAP is NP-complete (Kusiak and Heragu 1987),

the size of the problems which can be solved by optimal methods is limited (≤15) and, thus, suboptimal methods are more appropriate for cases of larger size.

An important class of suboptimal methods consider an initial layout and attempt to improve it by successive pairwise swaps of facilities. At each step, the swap that yields the maximum reduction of a certain criterion, typically material handling costs, is performed. This class includes the Computerized Relative Allocation of Facilities Technique (CRAFT) of Armour and Buffa (1963) and Buffa et al. (1964), the Automated Layout DEsign Program (ALDEP) of Seehof and Evans (1967), the methods of Hillier (1963), Hillier and Connors (1966), Fortenberry and Cox (1985), Malakooti (1989), Co et al. (1989), and the Biased Sampling approach of Nugent et al. (1968). These heuristics have one common drawback: the solution obtained is very sensitive to the initial layout. Since they are based on so-called "greedy algorithms," which only accept configuration changes that improve the objective function, the final solution usually corresponds to the local minimum that is nearest to the initial configuration.

A second class of suboptimal methods includes construction algorithms that assign facilities to points in the solution space one by one. The assignment is based upon certain relationships between these facilities, such as material flow, proximity requirements, etc. Representative methods include the COmputerized RElationship LAyout Planning (CORELAP) algorithm of Lee and Moore (1967) and the Modular Allocation Technique (MAT) of Edwards et al. (1970). Such algorithms are also 'myopic'; i.e., the best local assignment is performed at each step. This might prevent the convergence of the final solution to the global optimum.

Golany and Rosenblatt (1989) propose a hybrid method which combines the advantages of both improvement and construction algorithms. However, it still remains sensitive to the initial layout.

The machine (intra-cell) layout problem has received substantially less attention. Although it is closely related to the facility layout problem, certain practical issues prevent direct application of the methods discussed above for its solution. For example, most of these methods consider only equidimensional facilities, which is clearly an unrealistic assumption in the case of intra-cell layout (Heragu and Kusiak 1988). Although certain techniques like CRAFT and CORELAP do account for facility areas, they do not consider the shape of the manufacturing entities. Moreover, most of the facility layout methods require that the locations for facility placement, as well as the distances between them, be known *a priori*, which is not always possible (e.g., design of a new facility).

Leskowsky *et al.* (1987) address both the inter- and intra-cell layout problems using a common approach. They use the methodology of CRAFT in the layout algorithm. Realizing that the solution can converge to a local minimum, they suggest a method to force it away from it. A small number of local minima solutions are examined under the control of the user, and the best one is retained. No formal approach, however, is provided on how to conduct and terminate this procedure.

Heragu and Kusiak (1988) propose two construction methods for the layout of machines within flexible manufacturing cells. The first one only specifies the order in which the machines are to be placed, based upon "adjusted flow" values between machines. The second algorithm constructs triangles of

machines which correspond to maximum "adjusted flow" values (weights) and ranks them in descending order of weights. The machines are then assigned to pre-specified sites based on these rankings. According to the authors, the first algorithm does not guarantee optimal solution for problems with more than 3 machines. They also note that, for some classical test problems, the second algorithm yields solutions inferior to those obtained using improvement algorithms.

Heragu and Kusiak (1990) have also presented a knowledge based system for the layout problem. Formulations for single- and multi-row layouts are presented. Two layout algorithms are used in an expert system which takes into account a variety of practical issues such as layout type, material handling requirements, etc. Their expert system may also utilize any other layout algorithm, such as the one proposed in this paper.

The CLASS (Computerized LAyout Solutions using Simulated annealing) algorithm presented here is especially designed to overcome the dependence of the solution on the initial layout. It addresses the problem of relative location of equidimensional manufacturing entities within a discrete solution space based on the criterion of minimizing the total material flow (cost) between these entities. An approach to accommodate relative entity size is also presented. Thus, the proposed method is, in principle, capable of addressing both the inter- and intra-cell layout problems. Although locations for entity placement need not be known *a priori*, entity placement at pre-defined locations can also be addressed.

In Section 2 the formulation of the layout problem is presented. A brief overview of Simulated Annealing, and the relevant parameters used for the present

problem are discussed in Section 3. The CLASS algorithm is presented in Section 4. In Section 5 the performance of the proposed method is assessed by solving eight problems commonly cited in literature. Some practical issues, such as the incorporation of entity sizes, are discussed in Section 6. The conclusions of the study are drawn in Section 7 and presented along with the recommendations for future work.

2. Problem Formulation

The layout problem consists of determining the relative positions of the k entities in the set $\{m_1, m_2, ..., m_k\}$, which may represent either the set of machines belonging to a cell or the set of manufacturing cells within the shop. This analysis seeks to minimize the total distance traveled by the manufactured parts, or pallets of parts, between these entities. The objective function is defined as:

Minimize:

$$E = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} T_{ij} d_{ij} \qquad (1)$$

where T_{ij} is the number of pallet transfers between entities m_i and m_j, and d_{ij} is the corresponding distance. The value of T_{ij} is the number of pallets trips required for transferring all the parts which use the two entities consecutively. Note that the quantity T_{ij} may also be weighted by any combination of parameters to consider special material handling requirements, part weights, actual material handling costs, etc. To quantify the distance d_{ij} between the

5

entities m_i and m_j, the following two measures can be used: Cartesian distance; i.e.,

$$d_{ij} = [(x_i - x_j)^2 + (y_i - y_j)^2]^{1/2} \qquad (2)$$

or the "Manhattan" distance; i.e.,

$$d_{ij} = |x_i - x_j| + |y_i - y_j| \qquad (3)$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of the geometric centers of entities m_i and m_j, respectively. Since the coordinate space considered is discrete and finite, each point in the space is assigned a unique 'position number' in order to simplify the analysis. The problem, thus, consists of determining the position number corresponding to each of the k entities, in order to minimize the objective function of Eq. (1). It is assumed that the entities do not overlap when they are assigned to adjacent positions. Furthermore, each position cannot be occupied by more than one entity.

3. Solution Methodology

As mentioned in Section 1, the layout problem defined above is NP-complete. The proposed approach utilizes the method of Simulated Annealing which has proven very successful for the solution of such problems.

3.1 Simulated Annealing

Simulated annealing (SA) is a relatively recent development among methods that address combinatorial optimization problems. According to Kirkpatrick *et al.* (1983), it is based on certain analogies that exist between problems of

combinatorial difficulty and large systems with many degrees of freedom, such as those usually encountered in statistical mechanics.

Consider, for example, the annealing process utilized for bringing a metal to its lowest energy state, i.e. free of internal stresses. The metal is first heated up to its melting temperature and is then cooled slowly to allow the release of all internal stresses. During this process, a change in the state of the metal is likely to occur if it leads to a lower energy state. The same principle is utilized in order to minimize the objective function of a combinatorial problem within finite time. A change in the configuration of the system is allowed if it leads to a reduction in the value of the objective function characterizing the system. Configuration changes that increase the objective function are also allowed, but with lower probabilities. For example, the Metropolis criterion (Metropolis *et al.* 1953) specifies that such a configuration change is accepted with a probability given by the value of $e^{-\Delta E/t}$, where ΔE is the change in the objective function and t is a control parameter (temperature) defined in the same units (Kirkpatrick *et al.* 1983).

The method of simulated annealing is directly applicable to those optimization problems wherein the decision variables can be expressed as points in a discrete, n-dimensional space. Typical applications include the traveling salesman problem (Kirkpatrick 1984, Lin 1965, Press *et al.* 1988) and the problem of chip placement on microprocessor circuit boards (Kirkpatrick *et al.* 1983, Vecchi and Kirkpatrick 1983, Darema *et al.* 1987, Casotto *et al.* 1987 and Kravitz and Rutenbar 1987).

To apply the SA method, the following must be defined: i) the solution-space ii) the appropriate procedure to vary the decision variables within this space, iii)

the initial 'melting' temperature t, and iv) the 'annealing schedule', i.e., the manner in which the temperature t is to be reduced during the solution process. The annealing schedule can be static or dynamic and is typically a constant multiplier less than unity, or a logarithmically decreasing function. It is noted that the temperature is decreased in order to steadily reduce the probability of acceptance of configuration changes which increase the value of the objective function. Sufficiently long intervals, in terms of the number of configuration changes attempted, are allowed at each stage to let the system reach a stable configuration.

The basic advantage of SA over iterative improvement procedures is that transitions out of local minima are possible, since configuration changes which increase the objective function are also allowed (Kirkpatrick *et al.* 1983). Hence, the solution obtained usually corresponds to the global minimum.

3.2 Problem parameters

Solution-Space

The solution space consists of a kxk grid wherein each point is assigned a unique identification number. Note that only k of the k^2 positions are actually occupied by the k entities throughout the solution process. However, this solution space can accommodate even the extreme case of facility placement in a straight line, if such a layout is necessitated by the physical requirements of the system. The user may also specify a custom solution space of smaller size that consists of at least k positions. Thus, by appropriately defining the solution space and the distances between the available positions, any type of intra-cell

8

layout, such as single-, double- or multi-row linear, or circular, can be addressed (see Section 5.1).

Configuration changes

Two distinct actions are considered to change the configuration of the system:

(i) A 'move' of an entity from its current position to another, previously unoccupied, position.

(ii) A 'swap' of the positions of any two entities.

Both these actions involve the exchange of the 'contents' of two positions, at least one of which contains an entity. Exchanges involving more than two positions can also be accommodated, although this increases the complexity of the method and does not offer any advantages in terms of improving solution quality. Any such exchange can be accomplished by an appropriate combination of pairwise swaps.

Annealing Schedule

As discussed in Section 3.1, the annealing schedule consists of defining the initial (melting) temperature t and the equation(s) which govern the variation of system temperature. The initial temperature should be sufficiently large so that virtually all configuration changes are accepted. The simplest temperature change schedule is a constant positive multiplier (*TFACTOR*) with value less than unity.

Configuration change acceptance criterion

The Metropolis criterion (Metropolis *et al.* 1953) was selected to govern the acceptance or rejection of configuration changes. It considers the following cases:

(i) If the configuration change results in a net reduction of the objective function given by Eq. (1), then it is accepted.

(ii) If the configuration change increases the objective function, then it is accepted with a probability of $e^{-\Delta E/t}$, where ΔE represents the change in the value of the objective function and t is the temperature of the system at the corresponding stage of the procedure. In the present application, the configuration change is accepted if a randomly generated number between 0 and 1 is less than the value $e^{-\Delta E/t}$.

4. The CLASS Algorithm

The flow chart of the CLASS algorithm is shown in Figure 1, and a detailed description of its steps is given below.

Step 1

The number of entities, k, to be allocated and the traffic T_{ij} ($i,j = 1,...,k$) between them are specified. T_{ij} is either provided directly or determined indirectly given: i) the production routings of the manufactured parts, ii) their production volumes over a certain time horizon, and iii) the quantities of each part that can be accommodated in a standard pallet or any other means of material transfer.

Step 2

A $k \times k$ grid is generated for the placement of the k entities and the distance between all pairs of positions is determined using either the geometric or the Manhattan distance measure. The user may also specify a grid of a different

size or a custom solution space consisting of at least k positions, as discussed in Section 3.2. This feature is especially useful in the case of large problems, wherein the default solution space becomes unreasonably large, or when the available positions for entity placement are known *a priori*.

Step3

The entities are randomly placed within the solution space using a random number generator. The user can override this default option and specify the initial positions of the entities.

Step4

The total material handling distance E between the entities is calculated for the initial placement using Eq.(1). The initial placement is stored as the best layout so far and the corresponding value of E is stored as the minimum material flow distance.

Step5

The annealing schedule is set up in this step. The initial temperature is determined by identifying the lowest value at which at least 80 percent of a certain number of random configuration changes are accepted (Kirkpatrick 1984). A provision for determining this temperature is provided in the program. The initial temperature can also be specified by the user. The following parameters are also defined: the temperature reduction factor (*TFACTOR*) (<1.0), the number of configuration changes to be attempted at each temperature (*nover*), the number of successful configuration changes allowed at each step (*nsucc*) and the total number of temperature steps in the

annealing procedure (*ntsteps*). The default values for *nover*, *nsucc* and *ntsteps* are (100*k), (10*k) and 100, respectively (Press *et al.* 1988).

Step 6

Two positions from the solution space, at least one of which contains an entity, are selected using the random number generator. A configuration change is considered by swapping the 'contents' of these two positions. This may correspond to either a 'move' or a 'swap', as discussed in Section 3.2.

Step 7

The cost ΔE corresponding to the configuration change under consideration is evaluated as the change in the total material handling distance between the current and previous configurations.

Step 8

The configuration change is allowed if it meets the acceptance criterion described in Section 3.2. If the configuration change attempt is successful, the contents of the two candidate positions are swapped and the value of the objective function is updated. Note that as a result of the configuration change the objective function might increase, decrease or remain stationary. At low temperatures, however, the probability of a large increase in the objective function is significantly lower. If the total material handling distance for the resulting layout is less than that of the best layout stored in memory thus far, both the new layout and the corresponding material handling distance are saved.

Steps 6 through 8 are repeated until either the number of configuration changes attempted equals *nover* or the number of successful configuration changes equals *nsucc* during a single temperature step.

Step9

The annealing temperature is reduced by the temperature reduction factor *TFACTOR*.

Steps 6 through 9 are repeated until the number of temperature steps equals *ntsteps*. The annealing procedure is also terminated if the number of successful configuration changes at any temperature equals zero. The value of the objective function for the final layout is compared with that of the layout stored in memory and the better one is saved.

5. Performance Evaluation

The performance of the proposed algorithm was compared to that of twelve other facility layout methods for eight commonly used test problems proposed in Nugent *et al.* (1968). The test results for the 12 layout methods have been taken directly from Kusiak and Heragu (1987) and are shown in the upper block of Table 2. The lower block of this table contains the results obtained from the CLASS algorithm. The c.p.u times indicated for CLASS were obtained on a Sun 3/50 workstation. Comparisons were made in terms of both the quality of the solutions obtained and the speed of convergence. The latter cannot be directly used for comparison purposes, since it depends on the type of computer system

used. However, for each algorithm, the ratio of the c.p.u times corresponding to the two cases with the maximum and minimum number of facilities, respectively, does give an indication of the efficiency of the algorithm as the size of the problem increases. The quality of the solutions was assessed by the following equation (Kusiak and Heragu 1987):

$$\text{Solution Quality} = \frac{E}{L.B.}(100) \tag{4}$$

where E is the value of the objective function for the final layout obtained and L.B. is a lower bound determined as follows: First, the $k(k-1)/2$ traffic values between the entities are arranged in descending order, while the distances between the k positions ($k(k-1)/2$ values) are arranged in ascending order of magnitudes. T^d_i and d^a_i give the ith elements of these lists, respectively. The value of L.B. is then determined using Eq.(5).

$$L.B. = \sum_{i=1}^{k(k-1)/2} T^d_i \, d^a_i \tag{5}$$

Thus, the lower the value of the solution quality measure, the better the solution. Note that the Manhattan distance criterion was used for all algorithms.

Table 2 shows that in every single case CLASS either equals the performance of, or outperforms each of the twelve methods. For the cases with 5,6,7 and 8 facilities, the solution shown is the optimal solution (Nugent *et al.* 1968). Note that in each case CLASS was initiated with a random assignment of facilities within the solution space.

In order to test the sensitivity of CLASS to the initial conditions, each of the cases with 5, 6, 7 and 8 entities was considered 5 times. In all runs a random initial placement of the facilities was used. The optimal solution was obtained

in each case. This indicates that the quality of the solutions obtained using CLASS is insensitive to the initial conditions.

Table 2 also shows that the computation time for CLASS does not significantly increase with an increase in the problem size. For example, the ratio of the c.p.u. times for the cases of 30 facilities and 5 facilities, respectively, is only 5.31 for CLASS, whereas the value of this ratio for the other methods (excluding Heuristics 1, 3 and FLAC) ranges from a minimum value of 11.57 for FATE to a maximum of 3150 for CRAFT. These results are indicative of the potential of the algorithm for application to problems of substantially larger dimensions.

6. Practical Issues

The proposed method addresses the problem of determining the proximity between equidimensional manufacturing entities in order to minimize the total material flow distance between them. Thus, it can serve as the basis for the solution of both the inter- and intra-cell layout problems. However, certain important issues need to be considered in each case in order to address practical applications.

5.1 Intra-cell layout

Two issues assume importance in the intra-cell layout problem (Heragu and Kusiak 1988): i) the sizes and shapes of the machines that belong to the cell, and, ii) the desired configuration of the cell (single, double multi-row, or circular arrangement), in order to utilize automated material handling devices such as AGV's, handling robots, etc.

So far the proposed algorithm has only considered the case of equidimensional entities. Subsequently the method was enhanced so as to accommodate entities of various sizes, by considering each entity to be composed of an integer number of square building blocks. This approximation is similar to the one proposed by Hillier and Connors (1966). It is recognized that a building-block representation of the size cannot cater for the shapes of entities with more than two building blocks.

To incorporate the enhancement to the proposed algorithm, the traffic between any pair of building blocks in the system is defined as follows: First, the traffic between two blocks belonging to different entities m_i and m_j is given by the following equation:

$$t_{ij} = \frac{T_{ij}}{n_i \, n_j} \tag{6}$$

where T_{ij} is the traffic between the two entities under consideration and n_i and n_j are the number of building blocks contained within these entities. According to this definition, the total traffic between the entities remains unaffected. Secondly, artificial values have to be assigned to the traffic between blocks that compose a given entity. This is necessary in order to ensure that these blocks are placed together in the final solution. Care has to be taken in defining such traffic values. If a low traffic value is used, these blocks may be separated in the final solution, leading to an unrealistic configuration. On the other hand, a very high traffic would dictate the annealing procedure, whereas the real traffic between the entities would play a relatively insignificant role in the solution process. A value for this artificial traffic which seems to work fairly well is 1.5 to 2.0 times the maximum traffic between any pair of blocks which belong to different entities.

Once the traffic between individual building blocks is defined, each of these blocks is considered as a separate entity to be placed in a discrete solution space. Subsequently, the procedure outlined in Section 4 is used.

To demonstrate the effectiveness of the proposed approach in the case of entities with unequal sizes, the example shown in Table 1 is considered. The table provides the number of building blocks required for representing each entity and the traffic between them.

The above data and a random initial placement of the blocks were used as inputs to the layout procedure. The traffic between blocks of the same entity was defined as two times the maximum traffic between blocks of different entities. An initial temperature of 15 units and an annealing schedule of 0.9 were used. The final layout obtained is shown in Figure 2, which corresponds to a solution quality of 1.13. It can be seen that the blocks composing each of the entities have been placed together in the final solution.

The second important issue for the intra-cell layout case, i.e., machine placement in pre-determined configurations (single, double or multi-row, or circular etc.) can be addressed by the proposed method if such an arrangement has been selected *a priori*. This selection can be guided by the availability of a specific material handling system. For example, the presence of a handling robot may necessitate the circular placement of machines, whereas a linear or double-row arrangement would be more appropriate in the presence of an automated guided vehicle (AGV) (Heragu and Kusiak 1988).

To demonstrate the capability of CLASS in addressing specific cell configurations, the problem of Table 1 was considered again. In this case, it was decided *a priori* that the machines were to be placed in a linear

arrangement. It was also assumed that the loading points for machines 1, 2, 3 and 4 were located along their shorter sides. The solution obtained for this case is shown in Figure 3, and corresponds to a solution quality (see Eq.(4)) of 1.25.

5.2 Inter-cell Layout

Certain improvements that enhance the applicability of the proposed method method can also be considered for the case of inter-cell layout, e.g.: i) Raw material and finished goods stores can be included in the formulation by identifying them as artificial cells. The appropriate traffic between these and the remaining cells can be defined directly from the part production routings. ii) Cell sizes can be accommodated using the facility described in Section 5.1.

7. Conclusions and Future Work

In this paper, we have proposed a new approach for the layout of manufacturing entities based on the criterion of minimum material flow distance. The method of Simulated Annealing, which has proven effective in solving other problems of combinatorial nature, has been utilized in the algorithm. The major advantage of the proposed method is its insensitivity to the initial conditions. Comparisons with some of the more prominent existing layout methods have yielded superior solutions. Both the inter- and intra-cell layout problems have been considered. The ability of the method to address a set of practical issues was examined. For example, the issue of approximate entity sizes has been accommodated with the use of a building-block approach.

As part of future work in this area, a critical consideration that needs to be examined is the robustness of the solutions obtained, in terms of the material flow, to projected changes in the product mix. Since a manufacturing system is dynamic, the parts produced in the system may undergo changes in terms of both design characteristics and production volumes. Therefore, an efficient layout should fulfil not only current, but also the future production requirements.

Appropriate enhancements should be implemented to further consider other practical issues. The proposed algorithm, in its present form, does not cater for actual entity sizes, neither does it guarantee the retention of the shape of any entity comprising of more than 2 building blocks in the final solution.

Actual shop-floor dimensions can be taken into account by defining an appropriate solution space. Immovable blocks may be defined to represent the walls and other physical obstacles in an actual shop-floor which impede regular traffic flow. In this case, the algorithm would have to be suitably enhanced to determine the traffic distances along regular traffic flow paths around these obstacles.

Although the present analysis uses material handling distances as the only parameter that governs entity placement, any combination of certain other parameters, such as material flow-times, traffic congestion along channels, accessibility, intersecting flow-lines, RELationship charts (Lee and Moore 1967), should also be considered.

References

Armour, G.C. and Buffa, E.S., 1963, A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities. *Management Science*, **9**, 294-309.

Buffa, E.S., Armour, G.C. and Vollman, T.E., 1964, Allocating Facilities with CRAFT. *Harvard Business Review*, **42**, 136-159.

Casotto, A., Romeo, F. and Sangiovanni-Vincentelli, A., 1987, A Parallel Simulated Annealing Algorithm for the Placement of Macro-Cells. *IEEE Transactions on Computer-Aided Design*, **CAD-6**(5), 838-847.

Co, H., Wu, A. and Reisman, A., 1989, A Throughput-Maximizing Facility Planning and Layout Model. *International Journal of Production Research*, **27**(1), 1-12.

Darema, F., Kirkpatrick, S. and Norton, V.A., 1987, Parallel Algorithm for Chip Placement by Simulated Annealing. *IBM Journal of Research and Development*, **31**(3), 391-402.

Edwards, H.K., Gillett, B.E. and Hale, M.E., 1970, Modular Allocation Technique (MAT). *Management Science*, **17**(3), 161-169.

Fortenberry, J.C. and Cox, J.F., 1985, Multiple Criteria Approach to the Facilities Layout Problem. *International Journal of Production Research*, **23**(4), 773-782.

Golany, B. and Rosenblatt, M.J., 1989, A Heuristic Algorithm for the Quadratic Assignment Formulation to the Plant Layout Problem. *International Journal of Production Research*, 27(2), 293-308.

Ham, I., Hitomi, K. and Yoshida, T., 1985, *Group Technology: Applications to Production Management* (Boston:Kluwer-Nijhoff Publishing).

Heragu, S.S. and Kusiak, A., 1988, Machine Layout Problems in Flexible Manufacturing Systems. *Operations Research*, 36(2), 258-268.

Heragu, S.S. and Kusiak, A., 1990, Machine Layout: An Optimization and Knowledge-Based Approach. *International Journal of Production Research*, 28(4), 615-635.

Hillier, F.S., 1963, Quantitative Tools for Plant Layout Analysis. *The Journal of Industrial Engineering*, 14, 33-40.

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

Jajodia, S., Minis, I. and Harhalakis, G., 1990, Manufacturing Cell Formation with Multiple, Functionally Identical Machines. Paper submitted for publication to *Manufacturing Review*.

Kirkpatrick, S., 1984, Optimization by Simulated Annealing: Quantitative Studies. *Journal of Statistical Physics*, 34(5/6), 975-986.

Kirkpatrick, S., Gelatt, Jr., C.D. and Vecchi, M.P., 1983, Optimization by Simulated Annealing. *Science*, 220, 671-680.

Kravitz, S.A. and Rutenbar, R.A., 1987, Placement by Simulated Annealing on a Microprocessor. *IEEE Transactions on Computer-Aided Design*, **CAD-6**(4), 534-549.

Kusiak, A. and Heragu, S.S., 1987, The Facility Layout Problem. *European Journal of Operations Research*, **29**, 229-251.

Lee, R.C. and Moore, J.M., 1967, CORELAP - Computerized Relationship Layout Planning. *The Journal of Industrial Engineering*, **18**(3), 195-200.

Leskowsky, Z., Logan, L. and Vannelli, A., 1987, Group Technology Decision Aids in an Expert System for Plant Layout. *Modern Production Management Systems*, Edited by A.Kusiak, (New York: Elsevier Science Publisher), 561-583.

Lin, S., 1965, Computer Solution of the Travelling Salesman Problem. *The Bell System Technical Journal*, **Dec. 1965**, 2245-2269.

Malakooti, B., 1989, Multiple Objective Facility Layout: A Heuristic to Generate Efficient Alternatives. *International Journal of Production Research*, **27**(7), 1225-1238.

Metropolis, N., Rosenbluth, A.N., Rosenbluth, M.N., Teller, A.H. and Teller, E., 1953, Equation of State Calculation by Fast Computing Machines. *The Journal of Chemical Physics*, **21**(6), 1087-1092.

Nugent, C.E., Vollman, T.E. and Ruml, J., 1968, An Experimental Comparison of Techniques for the Assignment of Facilities to Locations. *Operations Research*, **16**, 150-173.

Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T., 1988, *Numerical Recipes in C: The Art of Scientific Computing.* (New York: Cambridge University Press).

Seehof, J.M. and Evans, W.O., 1967, Automated Layout Design Program. *The Journal of Industrial Engineering*, 18, 690-695.

Vecchi, M.P. and Kirkpatrick, S., 1983, Global Wiring by Simulated Annealing. *IEEE Transactions on Computer-Aided Design*, CAD-2(4), 215-222.

Table 1. Entity sizes and traffic for the example with unequal entity sizes

No. of building blocks		1	2	3	4	5	6	7
3	1	0						
2	2	9	0					
2	3	6	4	0				
2	4	0	4	0	0			
1	5	0	4	0	4	0		
1	6	0	4	4	0	4	0	
1	7	3	0	0	4	0	0	0

Table 2. Performance comparison of CLASS with 12 other facility layout methods for 8 common test problems

No. of facilities	5		6		7		8		12		15		20		30	
Layout Method	Sol'n Qual.	CPU time	Sol'n Qual.	CPU time	Sol'n Qual.	CPU time	Sol'n Qual.	CPU time	Sol'n Qual.	CPU time	Sol'n Qual.	CPU time	Sol'n Qual.	CPU time	Sol'n Qual.	CPU time
MAT	100.0	1.02	134.1	1.26	119.4	1.62	140.7	2.10	138.7	6.12	154.7	13.20	143.0	47.49	165.8	346.7
LPA	100.0	0.05	104.9	0.10	110.5	0.30	129.7	0.74	137.5	1.50	129.0	2.50	136.6	3.60	145.7	4.50
FATE	104.0	2.80	123.4	2.40	116.4	3.10	139.2	3.30	134.3	4.80	138.0	6.10	141.6	11.90	151.5	32.40
FLAT	100.0	0.04	104.9	0.04	110.5	0.05	127.5	0.05	128.4	0.07	128.2	0.11	139.5	0.23	150.3	1.27
CRAFT	112.8	1.00	107.8	2.00	118.8	3.00	124.6	10.00	121.9	70.00	126.5	160.00	132.1	528.00	142.5	3150.0
HC63-66	117.6	10.00	107.8	9.00	117.0	12.00	121.1	14.00	127.7	19.00	125.3	40.00	132.6	75.00	143.3	285.0
FRAT	112.8	0.11	109.3	0.15	119.1	0.25	122.0	0.33	124.1	1.25	125.6	2.96	131.7	10.87	141.2	58.30
TSP	100.0	0.25	104.9	0.30	110.5	0.75	117.6	1.23	118.9	6.84	122.1	20.13	127.9	59.87	137.9	383.2
Heuristic 1	-	-	-	-	-	-	-	-	-	-	117.5	382.3	131.1	251.6	138.5	364.5
Heuristic 2	100.0	1.33	104.9	2.61	110.5	4.49	117.6	7.13	118.9	12.90	120.0	25.65	127.2	70.88	138.0	261.6
Heuristic 3	-	-	-	-	-	-	-	-	-	-	120.0	31.49	126.7	100.0	137.6	400.8
FLAC	100.0	-	104.9	-	110.4	-	117.6	-	118.9	4.30	122.1	5.00	128.5	14.60	137.6	45.00
CLASS	100.0	1.38	104.9	2.26	110.4	1.20	117.6	0.92	118.9	1.43	120.0	1.95	126.4	1.52	137.4	7.33

25

List of Figures

Start

Input:
no. of entities: k
traffic between entities: T_{ij} ;i,j = 1,...,k

Random placement of all *k* entities on
k x k grid

Compute total distance *E* for the
initial layout

Define: t, TFACTOR, nover, nlimit, ntsteps.
ncount = 0; nsucc = 0; no_temp_steps = 0

Select (at random) candidate positions
p1,p2, for swapping

Compute ΔE = cost of configuration
change (in terms of total distance)

$\Delta E < 0$

No

generate random number
between 0 and 1: ran

$ran < \exp(-\Delta E / t)$

Yes

No

Yes

swap contents of p1, p2
nsucc = nsucc + 1;
$E = E + \Delta E$

ncount = 0
nsucc = 0

ncount = ncount + 1

ncount = nover
or
nsucc = nlimit

No

Yes

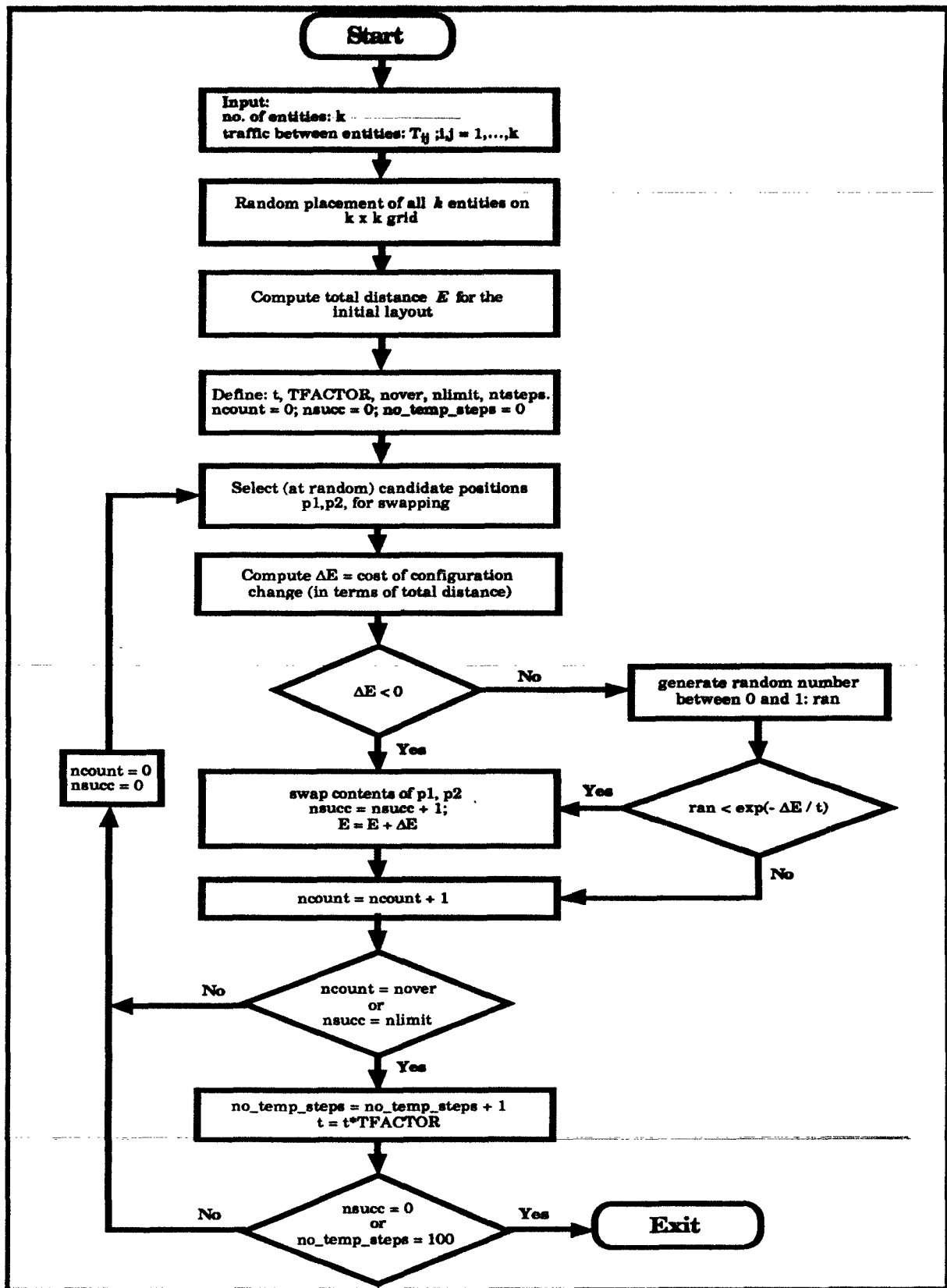no_temp_steps = no_temp_steps + 1
t = t*TFACTOR

nsucc = 0
or
no_temp_steps = 100

No

Yes

Exit

Figure 1

27

Figure 2

Figure 3