THESIS REPORT
Master's Degree

UNIVERSITY OF MARYLAND
COLLEGE PARK

SYSTEMS
RESEARCH
CENTER

HARVARD

Supported by the
National Science Foundation
Engineering Research Center
Program (NSFD CD 8803012),
Industry and the University

Design of Manufacturing Cells with Multiple, Functionally Identical Machines

by S. K. Jajodia
Advisors: G. Harhalakis and I. Minis

M.S. 90-12

Design of Manufacturing Cells with Multiple, Functionally Identical Machines

by

Satish Kumar Jajodia

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1990

Advisory Committee :

Associate Professor	George Harhalakis, Co-Advisor
Assistant Professor	Ioannis Minis, Co-Advisor
Assistant Professor	Guangming Zhang

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The production function of a manufacturing company is significantly affected by the layout of its manufacturing shop. Whereas a well designed layout can contribute significantly to 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, and other deficiencies (see Ham *et al.* [21]). Consequently, the importance of layout planning in a manufacturing facility cannot be overemphasized.

Recent years have seen a growing trend among manufacturing companies world-wide to introduce various innovative methods of production. One concept which has gained significant popularity as an effective means of increasing manufacturing efficiency and throughput, and which, among other issues, addresses the layout of a manufacturing facility, has been that of Group Technology (GT). Group Technology aims at introducing the advantages of mass production into batch manufacturing. It recognizes and exploits the similarities between parts with respect to both design characteristics and production requirements for streamlining the functions of design, process planning, manufacturing, and material flow. Various new elements such as GT part codes and machine cells, are used to implement the GT concept in a manufacturing organization.

Cellular manufacturing is an important and widely accepted application of the GT concept wherein the equipment comprising a manufacturing facility is grouped into cells, which are clusters of (mostly) dissimilar machines dedicated to the manufacture of one or more families of parts with similar processing requirements. This is in contrast to the functional arrangement, wherein the machines which perform similar operations are grouped together into functional departments. In the ideal case of a cellular manufacturing environment, the manufacturing cells are independent, i.e., each cell completely satisfies the processing requirements of one or more part families. In reality, however, the case of perfectly decoupled cells is highly unlikely to exist. More common is a hybrid environment which displays the characteristics of both functional and cellular layouts.

Equipment grouping is just one aspect of the design of a manufacturing shop. Other important issues include the actual or relative placement of the production equipment within each cell, as well as the placement of the resulting cells within the shop. The former is critical since it dictates the material flow pattern within each cell. The relative location of the cells on the shop floor becomes less important as the degree of dependence between the cells decreases. However, in a hybrid manufacturing environment this issue still needs to be addressed in order to exploit the benefits of cellular manufacturing.

This thesis addresses the entire design problem of a cellular manufacturing facility. The disaggregation of the shop into cells is performed in such a manner that the traffic of parts within the shop is minimized. A general manufacturing shop of discrete parts consisting of both unique and functionally identical machines is considered. In such an environment, the process plans of the manufactured parts usually define the sequence of machine *types* used in the parts' production. Thus, the grouping method also needs to address the determination of unique routings that indicate the specific machines to be utilized for the manufacture of the parts produced in the system. Machine capacity constraints are also considered during the assignment of parts to machines. Having determined the machines that comprise each cell, the physical layout of each cell and the allocation of the cells to distinct areas on the shop floor is addressed. The proposed methodology can accommodate both planned as well as existing facilities. It is also unique in the manner in which the issue of functionally identical machines is handled in the process of defining the manufacturing cells.

This thesis is structured as follows: The benefits of cellular manufacturing are briefly discussed in Chapter 2, which also presents a detailed survey of the available methods for both the cell formation and the layout problems. Chapter 3 presents the formulation and the solution algorithm for the cell formation problem. Chapter 4 addresses the layout problem. The application of the methods proposed to a large manufacturing facility is presented in Chapter 5. The conclusions and recommendations for future work are discussed in Chapter 6.

Chapter 2

Literature Survey

2.1 Cellular Manufacturing

The layout of a discrete parts manufacturing shop can be classified into two types: (i) Functional, wherein each group of functionally similar machines occupies a dedicated area of the shop floor. (ii) Cellular, wherein the production equipment is disaggregated into manufacturing cells. In this case, each cell contains most of the machines that are required to produce one or more families of parts with similar processing requirements. Various definitions have been proposed for underscoring the concept of cellular manufacturing (see Martin [36]).[1]

---

[1] *" A manufacturing cell is the grouping of people and processes into a specific area dedicated to the production of a family of parts or products."*

Dennis E. Wisnowsky, President,
Wizdom Systems, Inc., Naperville, IL.

" One or more machine tools linked together by common material handling and under the control of a centralized cell controller for the purpose of producing the given requirements of a family of parts."

Jim Simon, Vice-President of Engineering,
Giddings & Lewis, Fond du Lac, WI.

Some of the most important benefits of cellular manufacturing are listed below (see Ham *et al.* [21]).

- Reduction in materials handling distances and cost:

  The localization of part movement within a cell results in shorter part travel distances, and leads to reduced materials handling costs.

- Simplified material flow:

  Since most of the part operations are performed within one cell, the erratic material flow-paths between departments normally observed in a functional layout are avoided.

- Reduction in tooling/fixturing costs:

  Universal jigs/fixtures can be designed to meet the requirements of a family of parts with similar design features. This could lead to substantial reductions in the fixturing costs.

- Reduction in set-up times and costs:

  Since the parts that belong to a part family have similar processing requirements and can be accommodated on common jigs/fixtures, set-up times are significantly reduced.

- Reduction in Work-In-Process (WIP) inventory:

  Each manufacturing cell may be served by a dedicated material handling system which caters to the needs of the parts visiting the cell for processing. This could lead to significant reductions in the time that a part waits to be moved between work-centers.

Furthermore, the reduced set-up times also result in significant reductions in the total time spent by the parts on the shop floor.

- Simplification of scheduling and production control:

  Individual cells can be considered as quasi-autonomous units within the larger manufacturing facility and, therefore, the scheduling problem can be addressed at the cell level instead of the aggregate shop-floor level. Since only subsets of the manufacturing equipment and the product range need to be considered at this level, scheduling and production control are less complex.

- Compatibility with the JIT philosophy:

  The just-in-time (JIT) philosophy is compatible with the application of GT since the latter leads to improved work flow and shorter production times.

In a recent survey of companies that have implemented cellular manufacture (see Wemmerlov and Hyer [49]), many of these benefits have been quantified. The companies surveyed reported an average of more than 40 percent reduction in set-up times and over 21 percent reduction in materials handling as a direct result of establishing manufacturing cells. In addition, throughput time was reduced by 24 percent and WIP inventory by 20 percent. These results confirm the justification of applying the cellular manufacturing concept in batch manufacturing shops.

2.2 Methods for Cell Formation

Extensive research efforts have been focused on the development of methods to disaggregate a machine shop into manufacturing cells. The existing literature can be broadly classified into three major categories: (i) clustering approaches based on similarity measures, (ii) methods which permutate the part-machine incidence matrix to form part-machine clusters, and (iii) methods which are based on considerations other than the above, such as inter-cell traffic minimization and cost considerations.

Prominent among methods which utilize similarity measures between parts and/or machines in order to form part and machine clusters are: (i) The single linkage algorithm of McAuley [37], that defines the similarity between two machines in terms of the number of parts which visit both machines and the number of parts visiting either machine; he then aggregates machines with high similarity into manufacturing cells. (ii) The method proposed by Leskowsky *et al.* [32], which uses the Average Common Part Weighing Metric to quantify similarities among parts. (iii) The Knowledge Based Group Technology System of Kusiak [30], which forms work center cells and part families in such a way, that each part is manufactured exclusively by work centers in its corresponding cell or by bottleneck machines. (iv) The Numerical Taxonomy approach of Carrie [6], which is similar to the one proposed by McAuley. (v) The method of

Wei and Kern [48], who, again, enumerate similarities between machines for use in their linear clustering algorithm.

A second class of algorithms utilize the part-machine incidence matrix, each row of which represents a part and each column a machine. The elements of this matrix assume values of 0 or 1, a value of 1 for the entry a_{ij} representing an operation upon part 'i' by machine 'j'. Rows and columns are permutated to form a set of blocks with high densities of 1's along the diagonal. Each of these blocks identifies machine clusters and part families which are processed by the same. Methods which belong to this class of algorithms are: (i) The method of McCormick *et al.* [38], which maximizes the total 'bond-energy' of the part-machine incidence matrix. (ii) The rank order clustering approach of King [25], which rearranges the part-machine incidence matrix based on the 'binary rank orders' of its rows and columns. (iii) The direct clustering algorithm of Chan and Milner [9], which forms part and machine families by rearranging the rows and the columns of the incidence matrix, based on the number of non-zero elements in each. (iv) The GPM method of Garcia and Proth [18,19] that successively aggregates machines into cells and parts into families, using a procedure which awards operations on parts within their own cell and penalizes operations outside the cell. The common objective of all these methods is to maximize the number of operations performed on the part families within their corresponding machine cells.

The third class of cell-formation methods includes: (i) The cost-based heuristic of Askin and Subramanian [4], which considers the costs of work-in-process and cycle inventory, intra-group material handling, and machine set-up and run costs. (ii) The Inter Class Traffic Minimization Method [22,40], which forms machine cells and the corresponding part families in order to minimize the total inter-cell traffic of parts within the shop. (iii) The ICRMA (Identification, Clustering, Refinement, Merging and Allocation) heuristic of Tabucanon and Ojha [45], which also aims at reducing the inter-cell material flow in the system through four stages of machine formation and subsequent solution refinement. Part families are formed in the fifth stage by assigning each part to a unique machine cell.

The grouping methods that utilize similarity measures for clustering or simply re-arrange the part-machine incidence matrix do not consider certain important characteristics of the production system. Namely: (i) The sequence of operations in a part's production routing, although critical for material flow applications, is not taken into account. Most of the similarity measures employed in the clustering approaches do not consider the order of operations in part routings. Only recently, Choobineh [11] and Tam [46] proposed improved similarity measures that account for the sequence of operations. On the other hand, each entry a_{ij} of the incidence matrix can only signify whether or not machine j is used to manufacture part i, and does not indicate the sequence. (ii) Workload data such as capacity requirements of parts and available capacity on machines are also not considered. As a result, the final cell formation may not be feasible due to overloads of certain machines within some cells.

None of the methods discussed above adequately addresses the machine grouping problem in an environment consisting of functionally identical machines. It is emphasized that the presence of one or more groups of functionally identical machines that are represented by identical work-center designations is quite common in a job shop. In such an environment the part routings usually specify the machine *types* required for the manufacture of each part. In a cellular manufacturing environment, however, the routings should identify the specific machine for each part operation, since the functionally identical machines might be distributed in different cells. Consequently, the proper assignment of the parts among such machines becomes critical, since the inter-cell material flow is directly influenced by such an assignment. Capacity considerations are also critical in this case since overloading any work-center would be detrimental to the efficient functioning of the manufacturing organization.

Grouping methods that do provide for the case of functionally identical machines include the Knowledge Based Group Technology system of Kusiak [30] and Choobineh's two-stage grouping technique [11]. Both methods, however, create perfect cells between which no traffic flow is allowable. Consequently, the first technique results in a large general shop which processes those parts that cannot be manufactured entirely in a single cell. Furthermore, in this case, each group of functionally identical machines is assigned in its entirety to a single cell, which constitutes a sub-optimal solution. The second of these method assigns to each cell as many machines as are needed to satisfy both the processing and the capacity requirements of the corresponding part family. However, no constraints are considered on the number of machines available. Co and Araar [12] have also examined the case of functionally identical machines. During the first stage of their three-stage procedure, they distribute the parts processed by a group of functionally identical machines in such a way that the workloads are almost even among members of the group. The assignment procedure, however, does not consider the remaining operations in the parts' routings. Thus, very similar parts may be assigned to different machines of the group, resulting either in unnecessary traffic within the shop, or in the grouping of two or more identical machines to the same cell, even if this is not necessitated by capacity requirements. In Tabucanon and Ojha's [45] ICRMA heuristic, identical machines are only considered at the conclusion of the cell formation process, i.e., the they are assigned to the cells *after* the cells have been formed. This might clearly lead to a suboptimal solution since the cell formation directly depends on the presence of such machines. Moreover, the first stage of this procedure, which identifies dissimilar machines, is based on similarity measures and does not address traffic minimization, although this is the final objective of the method.

Seifoddini [44] examines the possibility of achieving reductions in materials handling costs by machine duplication in *existing* manufacturing cells. Machine acquisition costs are compared to the potential reductions in material handling costs for decision making. This method could be used to supplement existing grouping methods, but cannot independently address the cell-formation problem. Nagi *et al.* [41] address the issue of functionally identical machines in their two stage algorithm. The part assignment and cell formation processes are addressed independently, but with the common objective of minimizing the total inter-class traffic. The former problem is formulated as a linear programming problem and is solved using a simplex algorithm. The latter is solved using the ICTMM (Inter-Class Traffic Minimization Method) algorithm of Harhalakis *et al.* [22] and Nagi [40]. Since the problem has been decomposed into two distinct stages, optimal or near-optimal solutions obtained for either or both the problems does not necessarily guarantee a good solution for the global problem.

In conclusion, none of the above methods, with the exception of Nagi *et al.* [41], proposes a comprehensive methodology for manufacturing cell formation in the presence of functionally identical machines. This provided sufficient motivation for directing research efforts toward this area.

2.3 Methods for Facility Layout

A rich body of literature also addresses the relative placement of the facilities of a manufacturing system. The methods proposed encompass a wide range of heuristic techniques and exact optimization algorithms. Discussed below are a few of the most prominent among these methods.

A large segment of the methods for facility layout utilize variations of iterative improvement procedures addressing different criteria, the most commonly used being minimization of materials handling costs. One of the earliest methods based on an iterative improvement procedure was the CRAFT (Computerized Relative Allocation of Facilities Technique) algorithm of Buffa *et al.* [5] and Armour and Buffa [3]. Their algorithm is considered to be classical in the area of facility layout. The algorithm generates suboptimal layout alternatives by performing pairwise swaps of departments in order to minimize the total inter-departmental material handling cost. The ALDEP (Automated Layout DEsign Program) method of Seehof and Evans [43] utilizes an algorithm similar to that of CRAFT, except that a preference table is used instead of the material flow to indicate the proximity desirability codes between departments. Also, departments are assumed to occupy equal areas in this analysis. A variation of this program generates random layouts and scores them, the best few among which are retained. Hillier [23] proposes a very similar algorithm whereby an initial layout of equidimensional departments is first generated and pairwise exchanges of department positions are performed in order to minimize material handling costs. He also defines a criterion for evaluating the efficiency of the proposed solutions in terms of their closeness to the 'optimal' layout. The objective function proposed by Fortenberry and Cox [17] includes some additional measures of 'closeness' besides the material handling costs between departments. Their heuristic is again one of iterative improvement. Malakooti [35] formulates the layout design as a multiple criteria non-linear optimization problem and assigns certain weights to each criterion to obtain a set of efficient layout alternatives. A pairwise interchange method similar to CRAFT is utilized for the layout generation. Malakooti and Tsurushima [34] propose an expert system which addresses the problem as one of multiple criteria decision making. In the first stage of their two-stage procedure, layouts are generated based upon a set of rules, restrictions and preferences. The second stage seeks to improve the results of the first one through interaction with the decision maker.

All of the above methods have one common drawback: the solution obtained is very sensitive to the initial configuration provided. It is noted that the facility layout problem is one of combinatorial difficulty with a large number of local minima. Since each of the methods mentioned above are based on so-called "greedy algorithms," which only accept configuration changes that improve the objective function, the solution obtained usually corresponds to the nearest local minimum. Different local minima solutions can be obtained by using different starting points. Malakooti and Tsurushima [34] allow some flexibility by providing a user interface for the selection of a satisfactory layout. However, the inherent drawback of the greedy algorithm still remains. There is no guarantee that the layout obtained would be optimum without resorting to exhaustive search.

Golany and Rosenblatt [20] seek to eliminate this by providing their improvement algorithm with a 'good' initial layout. The initial layout is based upon a matching between sites and departments ranked on the basis of 'accessibility' and total 'connectivity,' respectively. This method, although claimed to yield better results in comparison to methods that start with a random initial assignment, still does not guarantee optimality. This is simply due to the fact that the initial assignment is again based on a heuristic which iteratively assigns the most connected department to the best available site in terms of accessibility. Thus, it is again a 'myopic' approach akin to that of greedy algorithms. Lee and Moore [31] in their CORELAP program place departments iteratively based upon connectivities and preference relationships between them. Their procedure resembles the initial assignment technique of Golany and Rosenblatt.

Chittajallu [10] formulates an optimization problem for the layout wherein the total weighted material handling distance is to be minimized. Devices (machines) are represented by their two-dimensional footprints. He then utilizes a penalty function method for solving the constrained optimization problem. This method is also very sensitive to the initial placement of the device shapes. This drawback is even more pronounced in his optimization formulation due to the combinatorial nature of the problem. According to Drezner [15], "*... the facilities will tend, during the solution process, to retain their original orientations around the center.*" Based on this observation, Drezner seeks to avoid the dependency of the solution on the initial assignment by incorporating a 'dispersion' phase wherein all the devices are placed at a common center and allowed to 'explode' outwards using a Lagrangian Differential Gradient (LDG) method. This technique, according to the author, provides a good starting point for the succeeding concentration phase which brings the devices together to provide the final solution. No formal proof of convergence or optimality is provided, although an intuitive justification for the algorithm has been discussed.

Co *et al.* [13] modify the methodology of CRAFT and use a measure of the production rate to form the objective function to be maximized. This method also shares the common drawbacks of iterative improvement algorithms. Aneke and Carrie [1] propose a technique for the layout of multi-product flowlines. However they consider only single-dimensional (linear) flow, which is not feasible in a batch shop.

In addition to the drawback mentioned earlier, none of the layout methods discussed above adequately considers the relative dimensions of the facilities. In the majority of them, the facilities are considered to be equidimensional. Some of them like CRAFT and CORELAP only consider the relative floor area requirements for the facilities. This has the disadvantage that the shape of the areas allocated to some facility may not correspond to the actual shape of the facility itself.

The preceding discussion establishes the need for a new method for facility allocation that does not depend on the initial conditions. The approach used in this research is based on Simulated Annealing (SA) and succeeds to a great extent in overcoming the shortcomings of the methods discussed above. Simulated Annealing was initially proposed by Kirkpatrick *et al.* [27] and has since been used with a great degree of success in many applications where gradient based methods and heuristics had produced less than satisfactory results. Combinatorial problems, for which monotonicity cannot usually be established and the objective function can have a large number of local minima, are ideally suited for the application of SA. Since the problem of facility layout is combinatorial, it is a prime candidate for application of the SA method.

The proposed method yields optimal or near-optimal results for the layout problem. The major advantage of the proposed method over the methods discussed in the preceding section is that the layouts generated are not dependent on the initial layout. Moreover, the approximate dimensions of facilities are taken into account to generate realistic layout solutions. Application to the classical examples presented in the literature produced results at least as good as, if not better than, the results produced by some of the more prominent methods discussed above.

Chapter 3

The Cell Formation Problem

3.1 Problem Definition

The problem of cell formation in an environment consisting of one or more groups of functionally identical machines can be considered to be two-fold. The major goal is to partition the production equipment into a set of smaller, relatively independent groups of machines. A secondary issue to be addressed is the development of specific process plans for parts which visit one or more groups of functionally identical machines for their manufacture. The criterion that drives the entire cell formation process is the minimization of the total inter-cell material traffic in the resulting cellular manufacturing system. The grouping and routing assignment problems are mutually dependent and thus, have to be considered together in order to obtain a realistic partitioning of the system.

3.1.1 Assumptions

The following assumptions were made in order to simplify the analysis:

- A discrete parts manufacturing environment is considered.
- The manufacturing shop consists of both unique as well as functionally identical machine types. The processing capacity requirement for the unique machines is assumed to be satisfied by their available capacity due to the unavailability of extra machines of such type for part assignment. The numbers and capacities of the functionally identical machines types are known.
- A unique process plan exists for each product which specifies the sequence of machine *types* required for its production, as well as the corresponding set-up and processing times for each operation.
- Consecutive operations on the same machine type are aggregated.
- Machines belonging to the same work-center type are assumed to be interchangeable; i.e., any member of the group can be utilized to process a part. Also, the processing capacities of all the machines of a certain type are considered equal, although this assumption can be easily relaxed.

3.1.2 Required Inputs

The following input information is necessary for the proposed disaggregation method:

- The set of machine types.
- The production routings of all parts. Each routing identifies the sequence of workcenter types visited by the part, and the corresponding set-up and processing times.

- The number and capacities of the machines belonging to each work-center type.
- The production volume of each part over a certain time horizon, either forecasted (in the case of a planned facility) or real (in the case of an existing facility).
- The average production batch-size of each part. Also required is the average number of units of each part that can be accommodated in one standard pallet (or any other means of material transfer).
- The maximum number of machines allowed in a cell.
- The least traffic between two cells that justifies their merging.

3.2 Problem Formulation

The problem consists of forming the set of manufacturing cells $C = \{c_1, c_2, \ldots, c_w\}$ in a way that the total traffic between cells (inter-cell traffic) is minimized. Furthermore, the production routings of parts which visit one or more groups of identical work-centers have to be rewritten in order to reflect specific machines from each of these groups. The latter is performed concurrently with the cell formation process, in a manner consistent with the final objective. Finite capacities of machines are also taken into account while assigning parts to them. Figure 3.1 illustrates the framework of the problem and the appropriate notations employed.

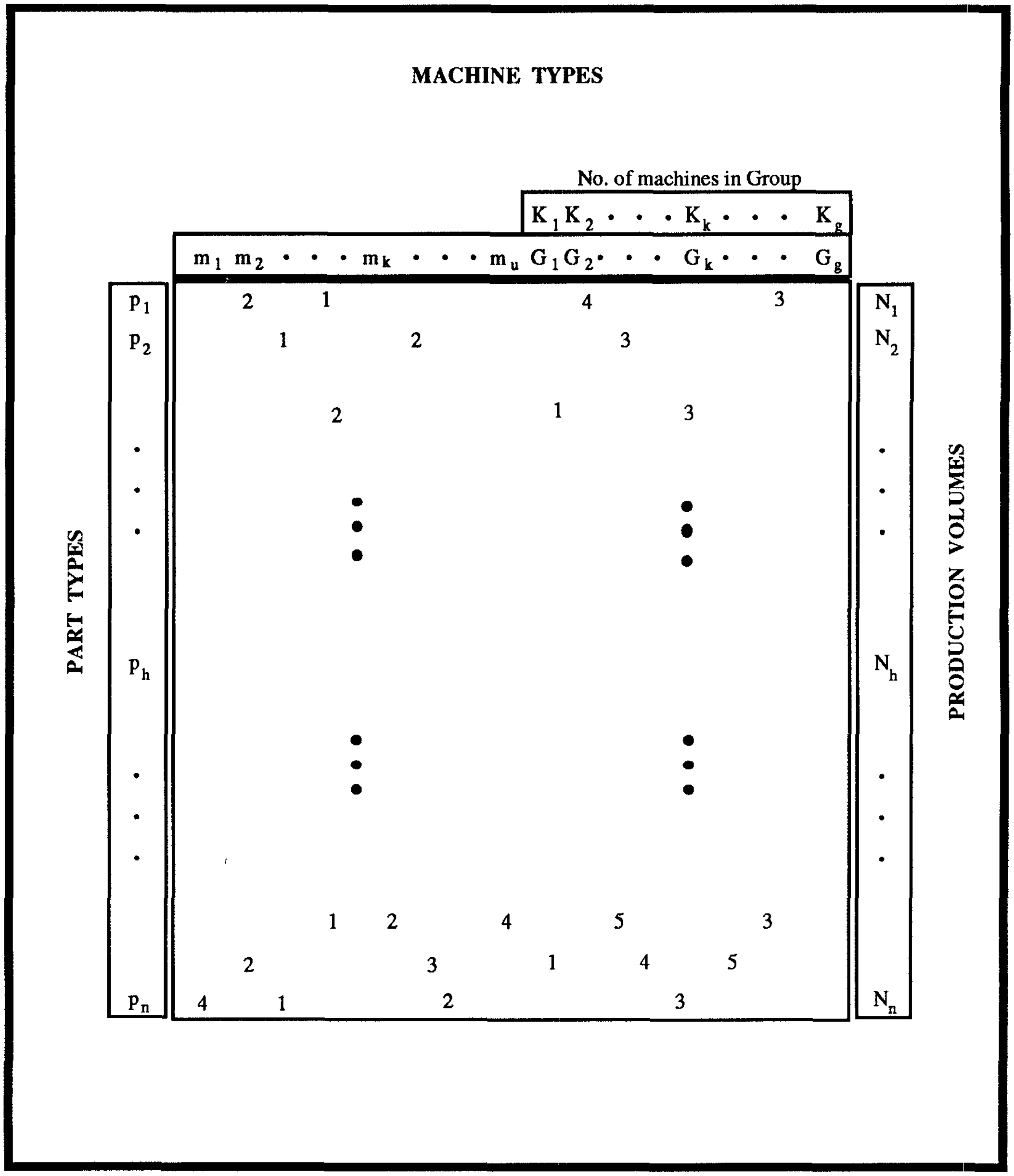


Figure 3.1. Framework of the cell formation problem

A set of '(u+g)' machine types and a set of 'n' part types are considered, along with pre-specified relationships between the two sets in the form of part production routings. The set of machine types is divided into two subsets: (i) the set of 'u' unique machines $M_U = \{m_1,$

$m_2,\ldots,m_u\}$, and (ii) the set of 'g' groups of identical machines $M_G = \{G_1, G_2, \ldots, G_g\}$, wherein each group G_k includes K_k identical machines. A unique production routing r_h corresponds to each part type $p_h : p_h \in P = \{p_1, p_2, \ldots, p_n\}$, P being the set of part types. This routing identifies the sequence of machine types used to manufacture the part under consideration, i.e.,

$$r_h = < m_h(1), m_h(2), \ldots, m_h(L_h) > \; ; \quad h = 1, \ldots, n$$

where $m_h(\lambda) \in M = M_U \cup M_G$ is the work-center required for the λ^{th} operation of part type p_h. L_h is the total number of distinct operations required for the manufacture of part p_h. Also included in the part routings are the set-up and run times for each operation, which are utilized for capacity requirement calculations.

Note that a common designation is used to identify an operation performed on a part by any machine within a group of identical work-centers. This is the usual practice in most functionally arranged manufacturing shops and signifies the interchangeability among the machines of each group. Note also that the designation G_k identifies a group of identical machines, whereas the designation G^j_k represents one particular machine of this group which has been assigned to a cell. Thus, an operation designated by G_k in a part routing signifies that any of the machines in the group G_k can be utilized for that particular operation. On the other hand, G^j_k denotes an operation to be performed by the specific machine G^j_k from the group.

The production volume of part type p_h over a given time horizon is represented by N_h.

The objective function of the problem can now be stated as follows :

Minimize:

$$T = \sum_{i=1}^{w-1} \sum_{j=i+1}^{w} \frac{t_{ij}}{q_i + q_j} \tag{3.1}$$

where w is the total number of manufacturing cells, q_i and q_j represent the number of machines in cells c_i and c_j respectively, and t_{ij} is the traffic between these cells, in terms of the number of pallet transfers between them; i.e.,

$$t_{ij} = \sum_{h=1}^{n} B_h(i,j)\, x_h(i,j) \tag{3.2}$$

where $x_h(i,j)$ is the number of times part p_h has to be transferred between cells c_i and c_j during its manufacture. This number is determined by examining the number of times two machines, one belonging to cell c_i and the other belonging to cell c_j, appear in sequence within the production routing of part p_h. $B_h(i,j)$ is the number of pallet transfers required to move the corresponding number of units of part type p_h between cells c_i and c_j, and is defined as follows:

$$B_h(i,j) = \begin{cases} s_h(i,j)/\mu_h & \text{if Mod}[s_h(i,j),\mu_h] = 0 \\ \text{Int}\left(\dfrac{s_h(i,j)}{\mu_h}\right) + 1 & \text{if Mod}[s_h(i,j),\mu_h] \neq 0 \end{cases} \tag{3.3}$$

where μ_h is the pallet size of part type p_h and $s_h(i,j)$ is the number of units of part type p_h transferred between cells c_i and c_j. The value of $s_h(i,j)$ is usually equal to N_h, the production volume of part type p_h, unless the latter has been divided among two or more alternate routings due to capacity considerations.

Note that the ratio $t_{ij}/(q_i+q_j)$ used in the objective function of Eq.(3.1) represents the "normalized" traffic between the two cells. It is employed in order to favor the union of smaller cells at each step of the merging process (see Nagi [40]).

The minimization of inter-cell traffic is subject to two sets of constraints:

(i) cell size constraints :

$$q_i \leq Q \qquad i = 1,\ldots,w, \qquad (3.4)$$

where Q is the maximum number of machines allowable per cell, determined by the user. The value of Q depends on several factors, such as the volume of work, the interdependence between workers and machines, the labor skills required, the sizes of the machines, etc. (see Ang and Willey [2]).

(ii) machine capacity constraints for each machine $G^k{}_j$ that belongs to group G_j :

$$\sum_{h=1}^{n} S_h(G^k{}_j)\left[\frac{\theta_h{}^{su}(G^k{}_j)}{\beta_h} + \theta_h{}^{run}(G^k{}_j)\right] \leq T_j \quad k = 1,\ldots,K_j;\, j = 1,\ldots,g \qquad (3.5)$$

where :

$S_h(G^k{}_j)$ is the number of units of part p_h processed by machine $G^k{}_j$,

$\theta_h{}^{su}(G^k{}_j)$ and $\theta_h{}^{run}(G^k{}_j)$ are the set-up and run times for part type p_h on machine $G^k{}_j$,

β_h is the average batch size of part type p_h, and

T_j is the capacity of each machine in group G_j.

It is noted that capacity constraints are only considered for those machines that belong to a group of functionally identical machines. Capacity overloads of unique machines are assumed unavoidable and are only reported by the system.

3.3 The Algorithm

The proposed method utilizes a bottom-up heuristic aggregation procedure wherein the maximum normalized inter-cell traffic is reduced at each iteration. The basic steps of the method are similar to the ones in the Inter Class Traffic Minimization Method (ICTMM) (see Harhalakis *et al.* [22] and Nagi [40]). However, major enhancements were necessary in order to address a general manufacturing environment that consists not only of unique machines, but also contains groups of functionally identical machines with finite capacities. The principal steps of the method are summarized in the flow-chart of Figure 3.2 and are described in the following discussion.

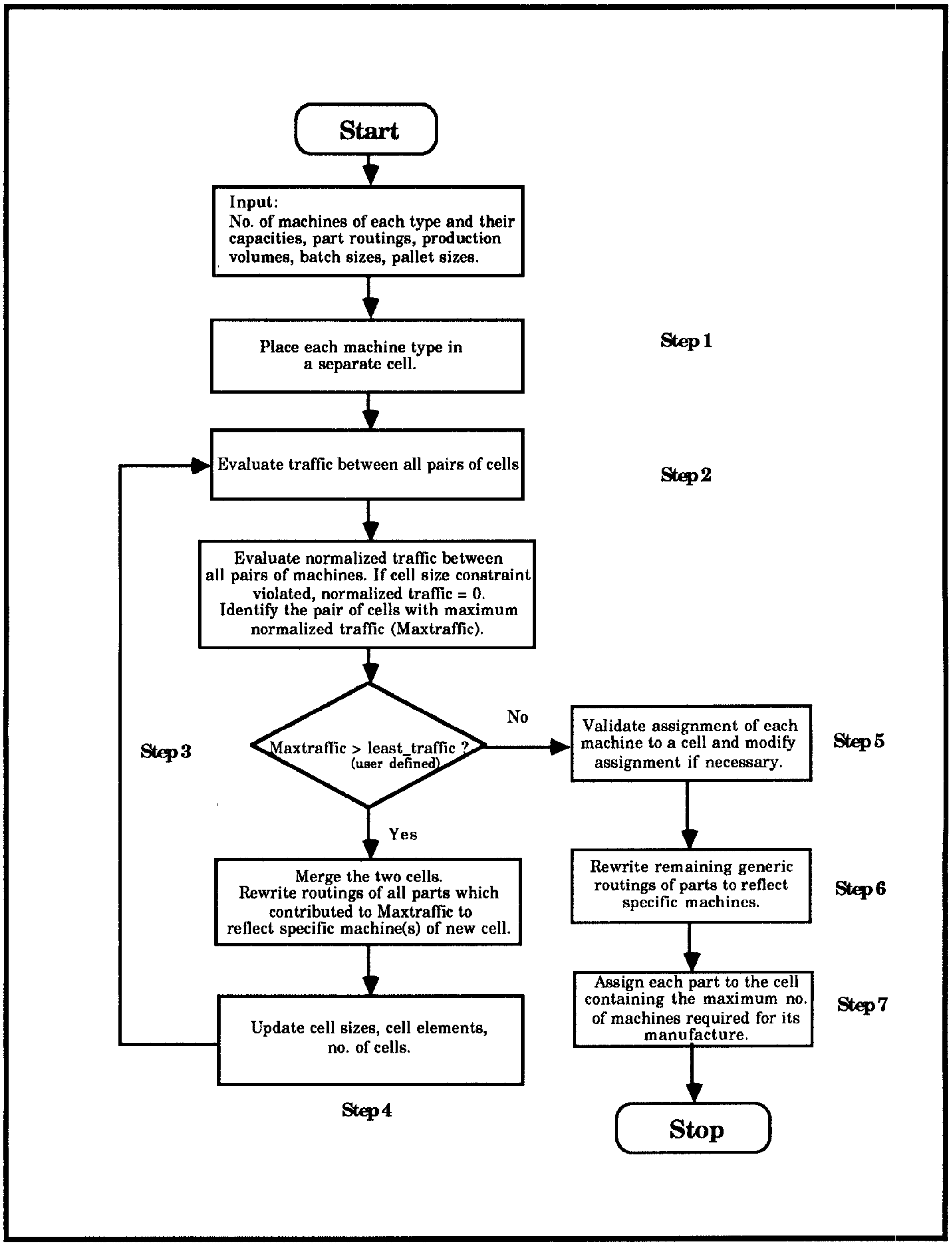


Figure 3.2: Flow-chart of the cell formation method

Step 1

Initially each machine from the set of unique machines M_U and each group of functionally identical machines from the set M_G is placed in a separate cell. Thus, at the beginning of the algorithm, the total number of cells is equal to the number of machine types available :

$c_i = \{m_i\} : i = 1,\ldots,u$

$c_{u+j} = \{G^1_j, G^2_j, \ldots, G^{K_j}_j\} : j = 1,\ldots,g$

Step 2

The inter-cell traffic t_{ij} is calculated for all pairs of cells (c_i, c_j). The general procedure for the evaluation of traffic between two cells is presented below.

Consider two cells c_s and c_r:

$c_s = M_s \cup G_s$

$c_r = M_r \cup G_r$

where M_x represents the set of unique machines and G_x represents the set of representatives from the groups of functionally identical machines that belong to cell c_x $(x = s,r)$. The traffic between these two cells is the traffic between all possible pairs of machines formed by considering one machine from either cell.

Three distinct cases are considered in these traffic calculations, and the appropriate procedures are described below:

Case (1) The traffic between a pair of unique machines (m_i,m_j) is determined using Eq.(3.2) for machines m_i and m_j.

Case (2) The traffic between a pair (m_i,G_j), formed by a unique machine m_i and a group G_j, is determined by arbitrarily considering a single remaining representative of the group G_j. The flow-chart for this case is shown in Figure 3.3.

Parts from among those that contain the sequence $\{m_i,G_j\}$, or the sequence $\{G_j,m_i\}$, are selected in such a manner that this traffic figure is maximized within the available capacity limit (T_j) of the machine under consideration. This is accomplished by first calculating the average processing time $b_h(G_j)$ of part p_h on the work-center G_j using the equation:

$$b_h(G_j) = \frac{\theta_h^{su}(G_j)}{\beta_h} + \theta_h^{run}(G_j) \qquad (3.6)$$

where $\theta_h^{su}(G_j)$ and $\theta_h^{run}(G_j)$ are the set-up and run times, respectively, of part p_h on work-center G_j, and β_h is the average batch size of part type p_h. All parts that contain the above sequence are then ranked in ascending order of the average processing time $b_h(G_j)$. In the case that work-center G_j is utilized for more than one operation on part type p_h, each occurrence of the work-center in the part's production routing is considered separately.

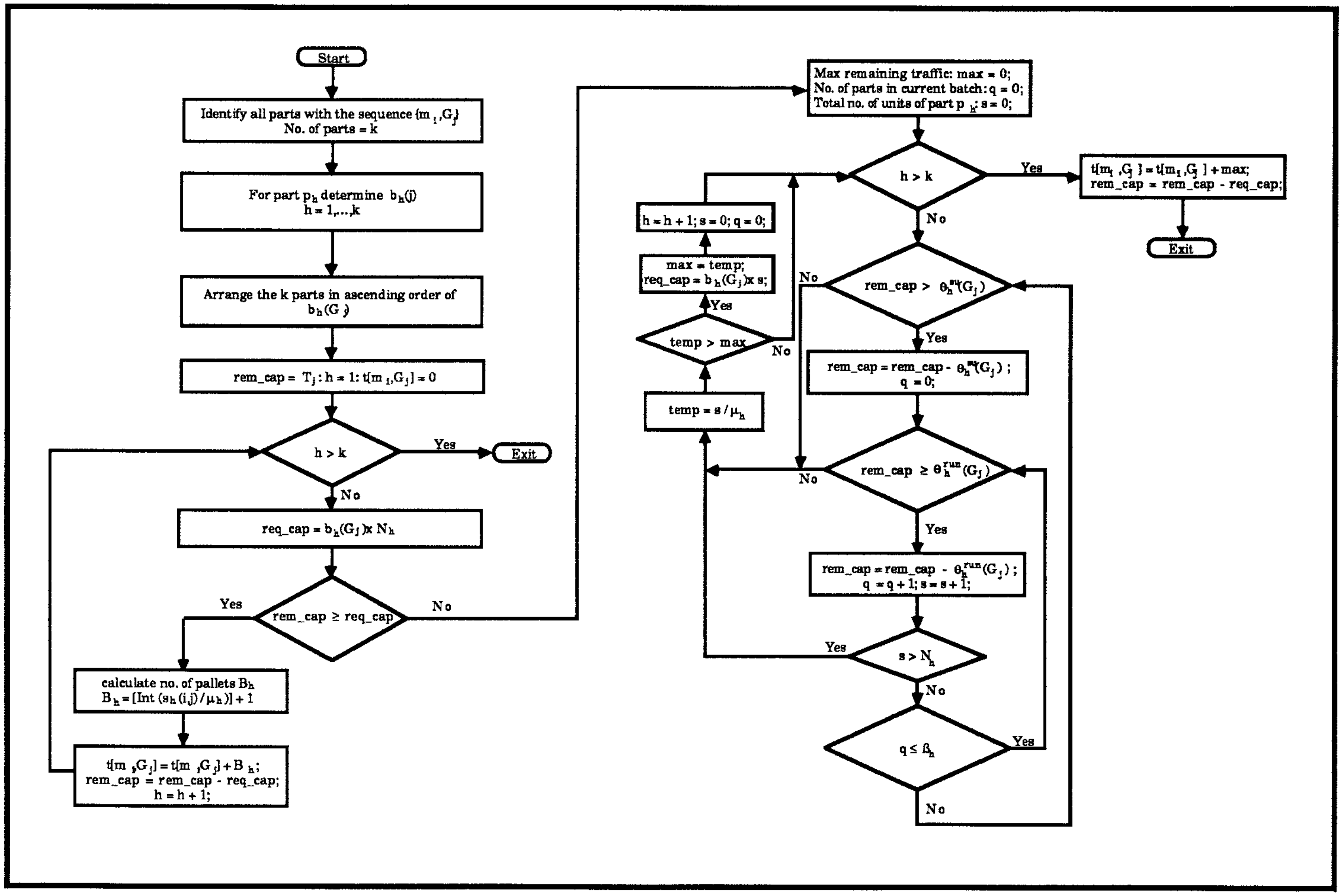


Figure 3.3: Flow-chart for the traffic calculation procedure

This might result in the part appearing more than once in the above mentioned list, and allows the flexibility to assign multiple operations on the same work-center type to different machines, if necessitated by traffic considerations. The advantage of this particular utility is illustrated in the example of section 3.4. Starting from the top of this ordered list, we then select as many parts as allowed by the capacity T_j. Thus, if the two cells under consideration are merged in step 3 of the algorithm, the maximum possible traffic between the two cells, within the existing capacity constraints, will be eliminated.

In the case that the entire production volume of the part p_z, which is selected last from the ordered list, cannot be processed by the selected machine due to the capacity constraint, a special procedure must be used. This is necessary since part p_z may have a high set-up time on machine G_j, which may consume most of its remaining capacity. Thus, this part may no longer be the one among the remaining parts in the list that contributes the maximum pallet traffic to the traffic value under consideration. The procedure used selects the most appropriate part in terms of traffic contribution from the remainder of the list. It should be noted that the capacity constraints of Eq.(3.5) are automatically satisfied, since parts are assigned to a particular machine only if its capacity has not been exhausted.

The procedure outlined above is also used when determining the traffic between a cell c_p and a machine $G^k{}_j$ that belongs to cell c_q. In this case, the presence of both the sequence $\{m_l, G^k_j\}$, where m_l is *any* machine belonging to cell c_p, and the sequence $\{m_l, G_j\}$ should be considered. Thus, if machine G^k_j has any remaining capacity after the assignment of parts that contain the first sequence, it can be utilized for the processing of any remaining parts which visit machine type G_j.

Case (3) The traffic between two groups of functionally identical machines (G_i, G_j) is determined by considering a single representative from each group and their respective capacities T_i and T_j. Parts from among those that contain the sequence $\{G_i, G_j\}$ or the sequence $\{G_j, G_i\}$, are selected in such a manner that this traffic figure is maximized. This is accomplished by first ranking all the above mentioned parts in ascending order of the higher among the two processing times $b_h(G_i)$ and $b_h(G_j)$, expressed as percentages of the remaining capacities of the two machines, respectively. Starting from the top of this list, as many parts as allowed by the capacities T_i and T_j are selected to determine the value of the traffic under consideration. This procedure also yields the maximum possible reduction in traffic if the two cells are merged in the next step of the algorithm. The procedure for the traffic evaluation between machines G_i and G_j is essentially the same as the one shown in Figure 3.3. However, the initial ranking of the parts is modified as described above, and, at each step, the appropriate capacity constraints are considered for both machine types.

The procedure outlined above is also used when determining the traffic between the machines G^f_i and G^k_j belonging to cells c_p and c_q respectively. This is for the same reason mentioned in case 2(b) above.

Step 3

The normalized traffic T_{ij} between all pairs of cells c_i and c_j is calculated. The pair that corresponds to the maximum normalized traffic value is identified, and the following operations are performed:

(a) The two cells are merged, subject to the merging rules described below.

Merging Rules

(i) The size of the newly formed cell should not violate the cell size constraint.

(ii) The size of a newly formed cell is the sum of the sizes of the two cells being merged. This, however, is not the case if a cell containing only a group of functionally identical machines is merged with any other cell. In this case, only one machine from the former is merged with the latter, since the capacity of a single machine of this group is considered in the traffic evaluation procedure. Moreover, the machine selected for merging from the group of functionally identical machines is renamed to obtain its original entity.

(b) The routings of all parts which were considered in the traffic evaluation procedure, and which are processed by one or more groups of functionally identical machines, are changed accordingly to reflect the specific machine(s) that belong to the newly formed cell. Thus, these parts now have unique production routings.

At the end of this step, the total inter-cell traffic is reduced by the traffic value between the two cells being merged. It is emphasized that this is the maximum possible traffic that could have been reduced by merging any pair of cells at this iteration, based on the way in which the inter-cell traffic is calculated.

Step 4

The cell configurations and the total number of cells are updated.

Steps (2) through (4) are repeated until either the traffic between all pairs of cells becomes lower than a user defined value (which is normally zero), or it is impossible to further merge any cells without violating the cell size constraint.

Step 5

This step validates the initial assignment of each machine to a cell. According to the aggregation procedure of the ICTMM (see Harhalakis *et al.* [22], Nagi [40]), and the method of Tabucanon and Ojha[45]), once a machine is assigned to a particular cell it cannot be withdrawn, even if its assignment to a new cell becomes more

- A unique routing for each part which identifies the sequence of (specific) machines visited by the part for processing.
- The production volume corresponding to each alternate routing of a part.

3.4 Example

Consider a manufacturing system consisting of 30 part types and 16 machine types. Work-centers 15 and 16 represent groups of functionally identical machines, and include 5 and 2 machines, respectively. The part and machine data is shown in Table 3.1. The first column of this table indicates the part production volume over a given period of time. Columns 2 and 3 show the average batch-size and pallet size, respectively, for each part type. Note that in this example, by setting the pallet-size for all parts equal to unity, only transfers of individual parts were considered. The part identification numbers are indicated in column 4. Each row of this incidence matrix represents the production routing of a part and indicates the sequence of machines required for its manufacture. Table 3.2 shows the production routing, along with the corresponding set-up and run times, for each part.

appropriate. This problem is addressed here, wherein each machine is examined for placement in each of the cells formed in steps (2) through (4). It is then assigned to the cell which yields the greatest reduction in the inter-cell traffic. The reassignment is done only if permitted by the cell size constraint.

Step 6

Parts which are processed by a group of identical machines, but are not yet assigned to a specific representative of this group, are now assigned in such a manner that the traffic is minimized, subject to the capacity constraints. Potential capacity overloads, if any, are reported.

Step 7

Each part is assigned to the cell which contains the maximum number of machines required for its manufacture. This helps in forming families that contain parts with similar processing requirements.

3.3.1 Output

The following is the output obtained from the application of the cell formation algorithm to a manufacturing system:

- The list of manufacturing cells which defines the machines belonging to each cell.
- A list of part families corresponding to the manufacturing cells.

Table 3.1: Part and Machine Data for the example

Part Details				Machines															
prod vol	batch size	pallet size	I.D	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
1	1	1	1	•	1	•	2	•	•	•	•	•	•	4	•	•	•	3	5
12	3	1	2	1	•	•	•	2	•	•	3	•	•	•	•	•	•	4	•
2	1	1	3	•	•	5	•	•	4	•	•	3	•	•	•	•	2	1	•
1	1	1	4	•	•	4	•	•	5	•	•	1	•	•	•	•	3	2	•
6	2	1	5	4	•	•	•	1	•	•	3	•	•	•	•	•	•	2	•
1	1	1	6	1	•	•	•	4	•	•	2	•	•	•	•	•	•	3	•
2	2	1	7	•	•	•	•	•	•	3	•	•	•	•	1	•	•	2	4
1	1	1	8	•	2	•	3	•	•	•	•	•	•	5	•	•	•	1	4
3	2	1	9	2	•	•	•	3	•	•	4	•	•	•	•	•	•	1	•
8	4	1	10	5	•	4	•	•	2	•	1	•	•	•	•	•	•	6	•
2	1	1	11	•	4	•	1	•	•	•	•	•	•	2	•	•	•	•	3
2	1	1	12	•	•	•	•	•	•	4	•	•	•	2	•	•	•	1	3
3	3	1	13	•	•	•	•	•	•	2	•	•	•	3	•	•	•	4	1
3	1	1	14	•	•	•	•	•	•	•	•	•	1	•	•	2	•	•	•
1	1	1	15	•	3	•	4	•	•	•	•	•	•	1	•	•	•	2	5
5	4	1	16	•	•	1	•	•	2	•	•	5	•	•	•	•	4	3	•
2	1	1	17	•	•	•	•	•	•	•	•	•	2	•	•	1	•	•	•
3	3	1	18	4	•	•	•	2	•	•	3	•	•	•	•	•	•	1	•
1	1	1	19	•	•	•	•	•	•	2	•	•	•	•	•	•	•	3	1
2	1	1	20	•	•	1	•	•	•	•	•	3	•	•	•	•	2	4	•
1	1	1	21	•	•	4	•	•	3	•	•	2	•	•	•	•	1	5	•
6	2	1	22	3	•	•	•	1	•	•	4	•	•	•	•	•	•	2	•
1	1	1	23	•	4	•	2	•	•	•	•	•	•	1	•	•	•	3	5
2	2	1	24	•	5	•	2	•	•	•	•	•	•	3	•	•	•	4	1
1	2	1	25	•	•	•	•	•	•	1	•	•	•	•	3	•	•	2	4
4	2	1	26	•	•	•	•	•	•	2	•	•	•	•	1	•	•	•	3
1	3	1	27	•	•	•	•	1	•	•	2	•	•	•	•	•	•	3	•
4	3	1	28	•	•	•	•	•	•	2	•	•	•	•	4	•	•	1	3
3	1	1	29	•	•	•	•	•	•	•	•	•	1	•	•	2	•	•	•
1	2	1	30	4	•	•	•	3	•	•	1	•	•	•	•	•	•	2	•

Table 3.2: Part Routings for the Example

Part ID, Sequence of Machines, Set-up and Run Times

1	2 0.1 0.2	4 0.1 0.2	15 0.4 .42	11 0.1 0.2	16 0.2 0.4
2	1 0.2 .15	5 0.3 0.2	8 0.2 0.1	15 0.2 .15	
3	15 .32 .45	14 0.1 0.4	9 .25 0.4	6 0.2 0.3	3 0.2 0.2
4	9 .25 0.2	15 0.4 .51	14 0.1 0.2	3 0.1 0.3	6 0.2 0.1
5	5 0.2 0.5	15 0.1 0.3	8 0.3 0.1	1 0.5 0.4	
6	1 0.3 0.2	8 0.1 0.1	15 .16 .25	5 0.6 0.1	
7	12 0.2 0.7	15 0.3 .31	7 .23 0.3	16 0.2 .25	
8	15 0.6 .11	2 0.3 0.9	4 .34 0.4	16 0.3 0.3	11 0.5 0.1
9	15 .09 0.3	1 0.3 0.1	5 0.3 0.2	8 0.2 0.2	
10	8 0.2 0.4	6 0.5 0.2	15 .09 0.1	3 .21 0.1	1 0.3 .15
	15 0.5 0.1				
11	4 0.2 .18	11 0.5 0.2	16 0.1 .25	2 0.5 0.3	
12	15 0.4 0.2	12 0.3 0.1	16 .23 0.3	7 0.1 0.1	
13	16 0.2 0.2	7 0.2 0.1	12 0.1 0.1	15 0.5 .12	
14	10 0.3 .25	13 0.2 0.2			
15	11 0.2 0.2	15 .78 .25	2 0.3 0.2	4 0.3 0.2	16 .18 0.3
16	3 0.3 0.1	6 0.5 0.1	15 0.3 .15	14 0.5 0.2	9 0.2 0.1
17	13 0.3 0.2	10 0.2 0.2			
18	15 .12 0.5	5 0.3 0.2	8 0.1 0.1	1 0.3 0.2	
19	16 0.1 0.1	7 0.4 0.1	15 0.2 0.4		
20	3 0.5 0.3	14 0.2 0.1	9 0.1 0.1	15 0.6 .38	
21	14 0.2 0.2	9 0.3 0.2	6 0.2 0.3	3 1.0 0.2	15 0.8 .55
22	5 0.2 0.3	15 0.3 0.2	1 0.6 0.3	8 0.5 0.2	
23	11 0.1 0.1	4 0.1 .05	15 .62 .35	2 0.1 0.2	16 0.4 0.2
24	16 0.1 0.4	4 0.2 0.3	11 0.6 0.3	15 .49 0.4	2 0.4 0.1
25	7 0.7 0.3	15 0.1 .16	12 0.9 0.3	16 0.2 .15	
26	12 0.5 0.2	7 0.4 .08	16 0.2 0.2		
27	5 0.2 0.3	8 0.6 0.3	15 .26 0.1		
28	15 0.1 .11	7 0.1 0.1	16 0.3 0.2	12 0.1 .06	
29	10 0.4 0.2	13 0.6 0.1			
30	8 0.2 0.1	15 0.4 .22	5 0.3 0.2	1 0.1 0.1	

The capacity of each work-center of type 15 and 16, both of which contain multiple machines, is set to 8.0 units over the given time horizon. The maximum number of machines allowed per cell is set to 6. The results of the grouping are shown in Table 3.3.

Table 3.3: Results of the grouping for the example

```
us|Part| 0 0 0 1 2| 0 0 1 2 2| 0 0 0 1 1| 0 1 1 2| 1 1|
us|I.D.| 1 5 8 7 0| 2 4 1 1 3| 3 6 9 4 8| 7 2 9 2| 0 3|
-----------------------------------------------------------
-----------------------------------------------------------
12|   2| 1 2 3 4 .| . . . . .| . . . . .| . . . .| . .|
 6|   5| 4 1 3 2 .| . . . . .| . . . . .| . . . .| . .|
 1|   6| 1 4 2 3 .| . . . . .| . . . . .| . . . .| . .|
 3|   9| 2 3 4 . 1| . . . . .| . . . . .| . . . .| . .|
 8|  10| 5 . 1 . 6| . . . . .| 4 2 . . 3| . . . .| . .|
 3|  18| 4 2 3 . 1| . . . . .| . . . . .| . . . .| . .|
 6|  22| 3 1 4 . 2| . . . . .| . . . . .| . . . .| . .|
 1|  27| . 1 2 3 .| . . . . .| . . . . .| . . . .| . .|
 1|  30| 4 3 1 2 .| . . . . .| . . . . .| . . . .| . .|
 1|  31| 2 3 4 1 .| . . . . .| . . . . .| . . . .| . .|
-----------------------------------------------------------
 1|   1| . . . . .| 1 2 4 3 5| . . . . .| . . . .| . .|
 1|   8| . . . . .| 2 3 5 1 4| . . . . .| . . . .| . .|
 2|  11| . . . . .| 4 1 2 . 3| . . . . .| . . . .| . .|
 1|  15| . . . . .| 3 4 1 2 5| . . . . .| . . . .| . .|
 1|  23| . . . . .| 4 2 1 3 5| . . . . .| . . . .| . .|
 2|  24| . . . . .| 5 2 3 4 1| . . . . .| . . . .| . .|
-----------------------------------------------------------
 2|   3| . . . . .| . . . . .| 5 4 3 2 1| . . . .| . .|
 1|   4| . . . . .| . . . . .| 4 5 1 3 2| . . . .| . .|
 5|  16| . . . . .| . . . . .| 1 2 5 4 3| . . . .| . .|
 2|  20| . . . . .| . . . . .| 1 . 3 2 4| . . . .| . .|
 1|  21| . . . . .| . . . . .| 4 3 2 1 5| . . . .| . .|
-----------------------------------------------------------
 2|   7| . . . . .| . . . . .| . . . . .| 3 1 2 4| . .|
 2|  12| . . . . .| . . . . .| . . . . .| 4 2 1 3| . .|
 3|  13| . . . . .| . . . . .| . . . . .| 2 3 4 1| . .|
 1|  19| . . . . .| . . . . .| . . . . .| 2 . 3 1| . .|
 1|  25| . . . . .| . . . . .| . . . . .| 1 3 2 4| . .|
 4|  26| . . . . .| . . . . .| . . . . .| 2 1 . 3| . .|
 4|  28| . . . . .| . . . . .| . . . . .| 2 4 1 3| . .|
-----------------------------------------------------------
 3|  14| . . . . .| . . . . .| . . . . .| . . . .| 1 2|
 2|  17| . . . . .| . . . . .| . . . . .| . . . .| 2 1|
 3|  29| . . . . .| . . . . .| . . . . .| . . . .| 1 2|
-----------------------------------------------------------

Global Efficiency = 97.619049
Group Efficiency = 98.936172
G. T. Efficiency = 97.894737
```

The five machines corresponding to work-center number 15 are renumbered during the cell formation process to machine numbers 17 through 21. Similarly, the two machines corresponding to work-center 16 are renumbered to 22 and 23. As can be seen from the final results, two machines of type 15 have been included in the second cell. This is necessary, since the capacity of a single machine is not sufficient to satisfy the capacity requirements of all the parts that visit this cell. The remaining three machines of this type have been assigned to different cells based on traffic considerations. It can also be seen from Table 3.3 that the total number of parts is now 31, since the total production volume of part no.9 has been assigned to two different routings owing to capacity constraints.

The following criteria, as suggested in Nagi [40], were used to evaluate the solutions obtained:

(1) **Global Efficiency :** It is the ratio of the total number of operations that are performed within each part's own cell, to the total number of operations in the system. This criterion quantifies the effectiveness of the method in confining the operations on parts within their corresponding cells.

(2) **Group Efficiency :** It is given by the difference between the maximum number of external cells that could be visited and the total number of external cells actually visited by parts in the cellular layout, divided by the former number. The value of this criterion quantifies

the effectiveness of the proposed solution in confining the manufacture of each part to as few external cells as possible.

(3) **Group Technology Efficiency :** This is the ratio of the difference of the maximum number of inter-cell pallet transfers possible and the number of inter-cell pallet transfers actually required in the cellular layout, to the maximum number of inter-cell pallet transfers possible.

The values of the three measures for the present example are shown at the bottom of Table 3.3.

The proposed algorithm has been coded in the 'C' language and runs on a Sun/Unix system. The results of this example were obtained in less than 2 seconds of c.p.u. time, which indicates a good capability for applications of much larger dimensions. The short execution time allows for the possibility of carrying out simulations with varying numbers of machines for each machine type, or even for varying cell-size limits.

Chapter 4

Facility Layout

The algorithm presented in Chapter 3 groups the production equipment of a manufacturing facility into cells and forms a set of part families which are processed within these cells. It does not, however, address either the relative placement of the machines within each cell or the relative placement of the cells on the shop-floor. Both these issues are critical in terms of the total material flow within the shop and are addressed in this chapter. The proposed approach is based on the method of Simulated Annealing, which is outlined below.

4.1 Background on Simulated Annealing

Simulated annealing is a relatively recent development among available methods for the solution of combinatorial optimization problems. According to Kirkpatrick *et al.* [27], the method has its origins in statistical mechanics. It has been developed based on analogies defined between large systems with many degrees-of-freedom and problems of combinatorial difficulty. Specific analogies between various characteristics of these systems can be illustrated using a hypothetical fluid as an example for the former (see Vecchi and Kirkpatrick [47]):

fluid	optimization problems
internal energy	objective function
atomic position	decision variables
cool into a stable, low energy state	find a near optimal configuration

The method utilized for bringing a metal to its lowest energy state, i.e. free of internal stresses, is that of annealing, wherein the metal is first heated up to its melting temperature and then slowly cooled to allow the release of all internal stresses.

In annealing, a change in the state of a 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 minimization 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 lead to an increase in the objective function are also allowed, but with lower probabilities. The Monte Carlo technique of Metropolis *et al.* [39] can be applied to such problems to define configuration changes within a discrete solution space as well as the probability of acceptance of such changes towards the minimization of the objective function. An example of a configuration change would be a swap in the positions of two chips in the problem of chip placement on a microprocessor circuit board in which the wire density in channels (see Kirkpatrick *et* *al.* [27]) is to be minimized. The Metropolis criterion [39] states that the configuration change would be accepted not only if it reduces the objective function, but also in a few cases for which the configuration change results in a net increase in the objective function. The probability of acceptance of such configuration changes is 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) which is defined in the same units as the objective function (see Kirkpatrick *et al.* [27]).

The method of simulated annealing essentially consists of defining: i) the appropriate procedure to vary the decision variables, ii) the initial 'melting' temperature, and iii) the 'annealing schedule', which specifies the manner in which the temperature is to be changed (reduced) during the solution process. The annealing schedule can be static or dynamic and is typically a constant multiplier less than unity, or a logarithmic decrement function. The temperature is decreased in order to steadily reduce the probability of acceptance of configuration changes which increase the objective function. Sufficiently long intervals, in terms of the number of configuration changes attempted, are allowed at each temperature step to let the system reach a stable configuration at each stage. The advantage of this method over iterative improvement procedures is that transitions out of local minima are possible, since configuration changes which increase the objective function are also allowed (see Kirkpatrick *et al.* [27]). Thus the solution obtained usually corresponds to the global minimum.

The simulated annealing method is relatively simple to apply to those optimization problems wherein the decision variables can be expressed as points in a discrete, n-dimensional space. Notable among its applications are the traveling salesman problem and the problem of chip placement on microprocessor circuit boards. The former is a classical NP-complete problem and has successfully been solved by SA (see Kirkpatrick [26], Lin [33] and Press *et al.* [42]). A substantial body of literature has focused on the latter problem, which addresses the placement of chips on a microprocessor circuit board in order to optimize certain design parameters, such as wire density in channels, total wiring length, etc. (see Kirkpatrick *et al.* [27], Vecchi and Kirkpatrick [47], Darema *et al.* [14], Casotto *et al.* [8] and Kravitz and Rutenbar [28]).

The problem of facility layout can be considered analogous to that of chip placement on a microprocessor circuit board, since both involve the placement of entities in a given two dimensional discrete space. This analogy has been exploited in the proposed algorithm in order to generate efficient alternatives for the layout of a manufacturing shop. The machine and cell placement problems are similar, although they are addressed at different levels in the hierarchy of the manufacturing facility. Thus, they can be solved using a common approach which is described in the following sections. Throughout the discussion, the term 'entity' represents either a machine within a cell, or an entire cell within the shop, depending on the context in which it is being used.

4.2 Setting of the Problem

4.2.1 Assumptions

The following assumptions were made in order to simplify the analysis.

- The entities (machines or cells) are represented by equidimensional square blocks to be placed in a finite, two-dimensional discrete space. A procedure to relax the assumption of equidimensionality in order to account for the physical dimensions of the entities is presented in section 4.6.
- The distance between two entities is defined as the distance between their geometric centers, either the cartesian distance or the "Manhattan" distance.

4.2.2 Inputs

The following input information is necessary for determining the layout:

- The traffic between the entities in terms of the number of pallet transfers. This information can be provided either directly, or indirectly by specifying the unique process plans of all the parts produced in the system, their production volumes over a certain time horizon, and the number of units of each part that can be accommodated in a standard pallet.

- In case entity sizes are considered (see section 4.6), the relative sizes and dimensions of the entities are also required in terms of the number of square building blocks required for representing each entity.

4.3 Problem Formulation

The placement problem consists of determining the relative positions of the 'k' entities in the set $\{m_1, m_2, \ldots, m_k\}$, which might be either the set of machines belonging to a cell or the set of manufacturing cells within the shop. The criterion involved in this analysis is the minimization of the total distance travelled by the parts (pallets) between the entities. The objective function can, therefore, be defined as:

Minimize:

$$E = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} T_{ij} d_{ij} \tag{4.1}$$

where T_{ij} is the (pallet) traffic between the entities m_i and m_j and d_{ij} is the distance between them. Note that the parameter T_{ij} can be weighted by any combination of various other parameters such as part bulk, actual material handling costs, etc. One of the following two measures of distance can be used.

The cartesian distance between the entities m_i and m_j is defined as:

$$d_{ij} = [(x_i - x_j)^2 + (y_i - y_j)^2]^{1/2} \tag{4.2}$$

The "Manhattan" distance between the entities is defined as:

$$d_{ij} = |x_i - x_j| + |y_i - y_j| \tag{4.3}$$

where (x_i, y_i) and (x_j, y_j) are the coordinates corresponding to 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. (4.1).

It is assumed that each position on the grid can completely accommodate an entity, i.e., the entities do not overlap when they are assigned to adjacent positions. Thus, the minimization problem is subject to one simple set of constraints, which state that each position can be occupied by not more than one entity.

4.4 Solution Using Simulated Annealing

4.4.1 Problem Set-up

Solution-Space

The 'k' entities under consideration have to be placed on a (kxk) grid wherein each point is assigned a unique identification number. It is noted that only 'k' of the 'k^2' positions on the grid would be actually occupied throughout the solution process. The reason for considering such a large solution space is to accommodate even the extreme case of facility placement in a straight line.

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 substantial advantages.

Annealing Schedule

The annealing schedule consists of defining the initial (melting) temperature *t* and the equation(s) which govern the change in the temperature of the system at each step of the procedure. The initial temperature should be sufficiently large so that virtually all configuration changes are accepted by the criterion presented in the next paragraph. The temperature change schedule, in its simplest form, can be a constant positive multiplier (*TFACTOR*) with value less than unity.

Configuration change acceptance criterion

The Metropolis criterion (see Metropolis *et al.* [39]) 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 in the objective function of Eq. (4.1), 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 time the configuration change is attempted. Thus, the configuration change is accepted if a randomly generated number between 0 and 1 is less than the value $e^{-\Delta E/t}$.

4.4.2 The Algorithm

Once the parameters of the method are completely defined, the method of simulated annealing can be applied to generate efficient layout alternatives for the manufacturing system. The flow chart of the proposed algorithm is shown in Figure 4.1 below, followed by a detailed description of its various steps.

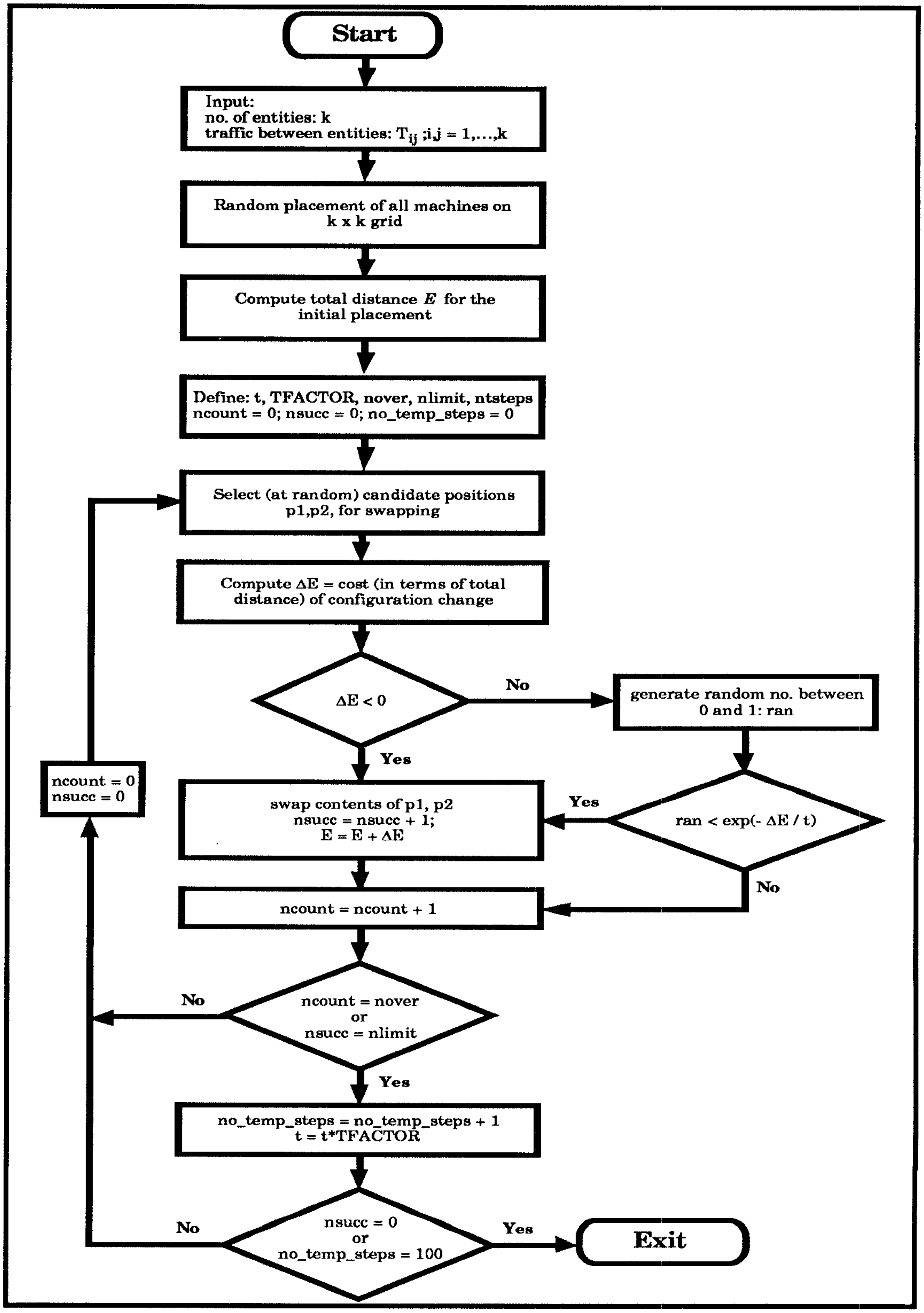


Figure 4.1: Flow-chart for the layout algorithm

Step 1

The number of entities to be allocated and the traffic between them are specified, either directly or indirectly as mentioned in section 4.2.2.

Step 2

A (k×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 criterion.

Step 3

The entities are randomly placed within the solution space. This is done with the help of a random number generator which assigns each entity to an empty position. The user can override this default option to specify the initial placement of the entities.

Step 4

The total material handling distance E between the entities is calculated for the initial placement using Eq.(4.1). The initial placement is stored as the best layout and the corresponding value of E is stored as the minimum material flow distance.

Step 5

The annealing schedule is set up in this step. The initial temperature is determined by identifying the lowest temperature at which at least 80 percent of a certain number of random configuration changes would be accepted. The initial temperature can also be directly specified by the user. The following parameters are also specified: 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 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.

Step 6

Two positions from the solution space, at least one of which contains an entity, are randomly selected using the random number generator. The attempted configuration change consists of swapping the 'contents' of these two positions. This might result in either a 'move' or a 'swap', as discussed in section 4.4.1.

Step 7

The total cost ΔE is evaluated as the change in the material handling distance between the current and previous configurations.

Step 8

The configuration change is allowed if it meets the configuration change acceptance criterion described in section 4.4.1. 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 an increase in the objective function is significantly lower. If the material handling distance for the resulting layout is less than that of the best layout stored in memory, the best layout is replaced by the former and the corresponding material handling distance replaces the minimum distance stored in memory.

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*.

Step 9

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 is also stopped 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.

4.5 Example

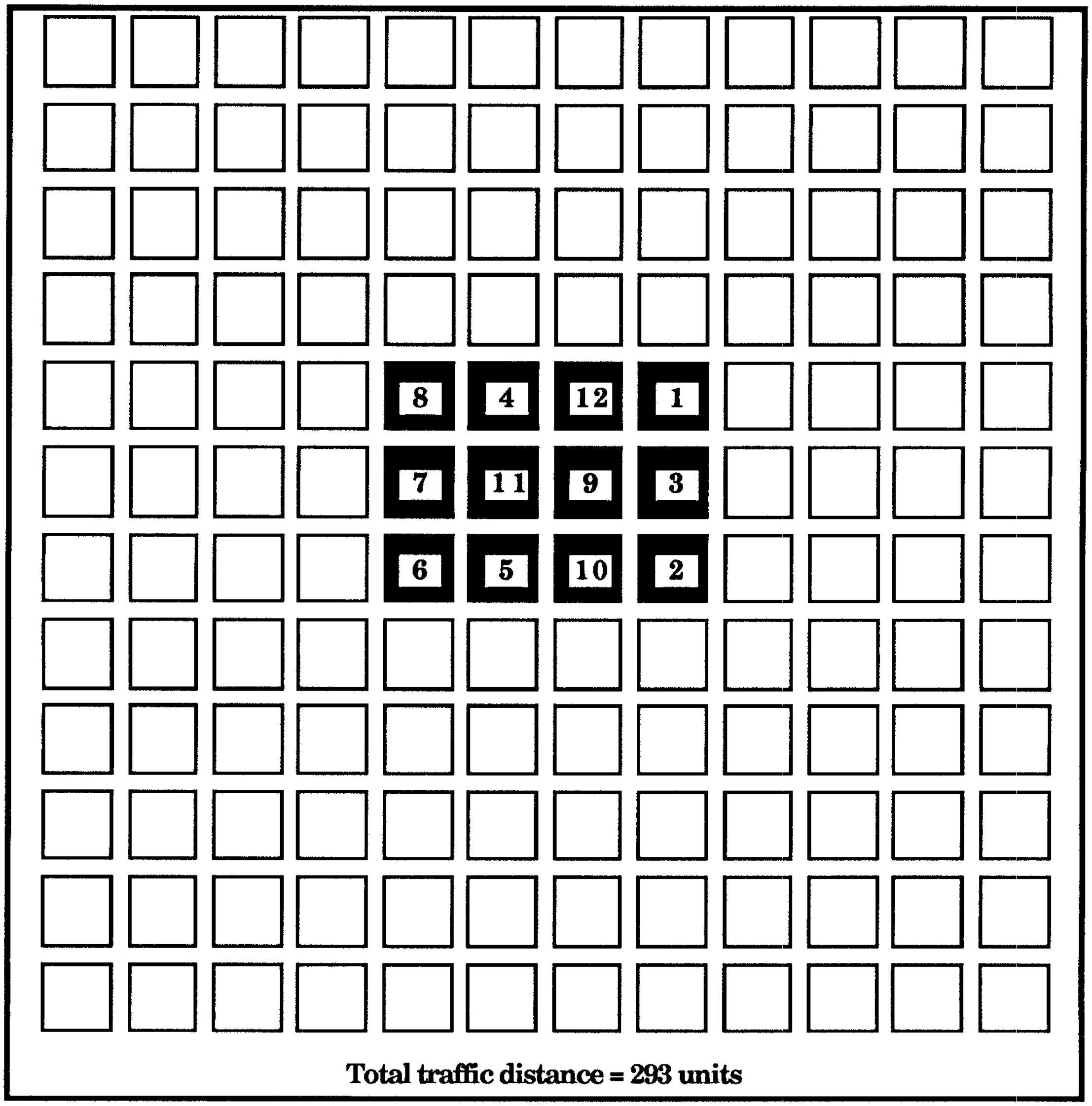
The case of a manufacturing cell with 12 machines is considered. This example was first proposed by Hillier [23] and has also been addressed by Golany and Rosenblatt [20]. In agreement with these studies, the entities were considered to be equidimensional and the Manhattan distance criterion was used . The traffic values between the machines are shown in Table 4.1.

Table 4.1: Traffic matrix for the example case

Entity Numbers

Entity Numbers	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	0											
2	5	0										
3	2	3	0									
4	4	0	0	0								
5	1	2	0	5	0							
6	0	2	0	2	10	0						
7	0	2	0	2	0	5	0					
8	6	0	5	10	0	1	10	0				
9	2	4	5	0	0	1	5	0	0			
10	1	5	2	0	5	5	2	0	0	0		
11	1	0	2	5	1	4	3	5	10	5	0	
12	1	0	2	5	1	0	3	0	10	0	2	0

The default values of the parameters *nover*, *nlimit* and *ntsteps* were used (see section 4.4.2). The starting temperature *t* and the parameter *TFACTOR* were defined as 30.0 and 0.9 respectively. The initial layout on the 12x12 grid was randomly generated by the program. The final layout is shown in figure 4.2 and corresponds to a total material handling distance of 293 units.



Total traffic distance = 293 units

Figure 4.2: Final Layout for the example case

It is noted that the layouts presented by Hillier, and Golany and Rosenblatt correspond to a material handling distance of 297 units. Consequently, in the present case Simulated Annealing has yielded superior results. Though not conclusive, these results certainly provide justification for the algorithm used and are indicators of its potential.

In order to facilitate comparison between various results and also to provide some means of quantifying the efficiency of the layout, the following criterion is proposed:

(1) **Global efficiency**, defined as:

$$\text{Global Efficiency} = \left[\frac{D}{E}\right] \tag{4.4}$$

where D is the total material handling distance assuming that all material transfers involve unit distance, and E is the material handling distance for the proposed solution. The higher the value of this criterion, the better the solution. Note that the optimal value of unity may never be obtained for a practical case involving more than two entities. Moreover, as the number of entities increases, the value of this criterion tends to decrease.

For the example case the Global Efficiency obtained is 59.4 percent, compared to 60.2 percent for the solution proposed by Hillier.

4.6 Entity Sizes

The method presented is based on the assumption that all entities are of equal dimensions, and can be placed at any of the positions in the solution space. This, however, is not true in most practical cases. In order to accommodate the approximate size of various entities, each entity is considered to be composed of an integer number of square building blocks.

The traffic between two blocks m_i and m_j belonging to different entities, hereafter referred to as 'non-identical' blocks, is given by the following equation:

$$t_{ij} = \frac{T_{ij}}{n_i\, n_j} \qquad (4.5)$$

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. Note that this definition does not alter the total traffic between the entities.

Artificial values have to be assigned to the traffic between blocks that belong to the same entity, hereafter referred to as 'identical' blocks. Care has to be taken while defining such traffic values. A low traffic value might cause these blocks to be placed away from each other in the final solution, leading to an unrealistic solution. On the other hand, defining very high traffic between identical blocks would have an adverse effect on the layout procedure. In this case the artificial traffic between the identical blocks would dictate the annealing procedure, whereas the real traffic between the non-identical blocks would play a relatively insignificant role in the solution process. A value for the traffic between identical blocks which seems to work fairly well is 1.5 to 2.0 times the maximum traffic between any pair of non-identical blocks.

Once the traffic between individual building blocks has been defined, each of these blocks is then considered as a separate entity to be placed in a discrete solution space, and the procedure outlined in the previous section is used.

4.6.1 Layout example with entities of unequal sizes

To demonstrate the effectiveness of the proposed approach in the case of entities with unequal sizes, the theoretical example of Figure 4.3 is considered. The system contains 7 entities with areas as shown in Table 4.2. The traffic between entities was defined in such a manner that the configuration of Figure 4.3 corresponds to the best solution. The traffic between the entities is shown in Table 4.2. This data and a random initial placement of the blocks were used as inputs to the layout procedure.

Table 4.2. Entity sizes and traffic for the unequal entity size example

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

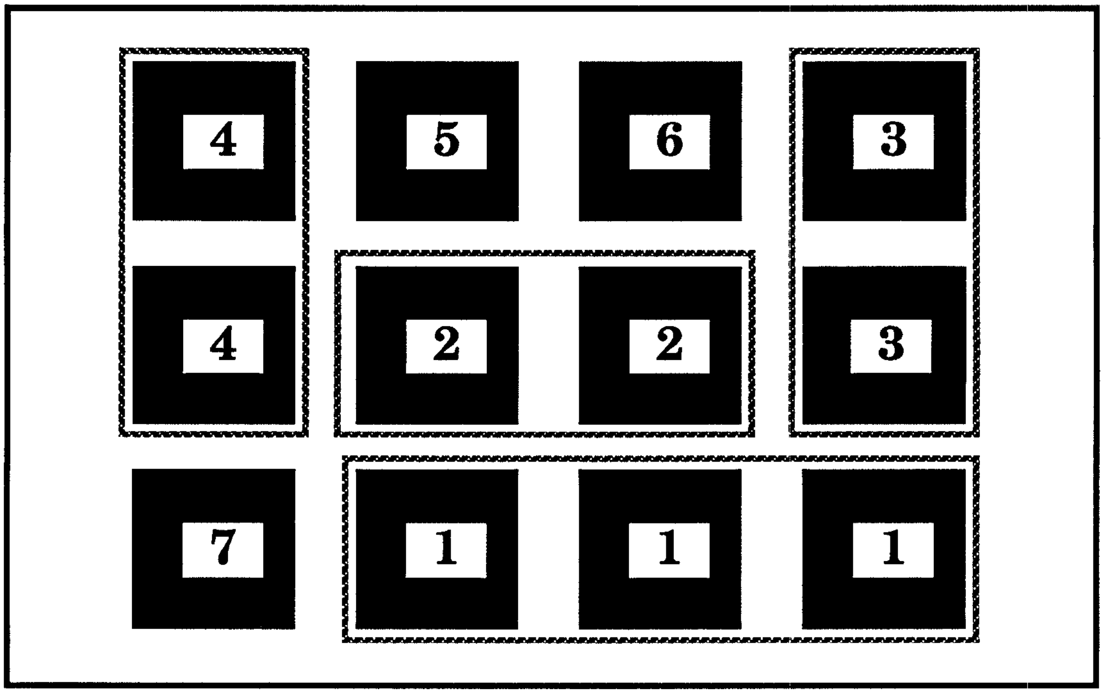


Figure 4.3: Layout for example with unequal sizes

The traffic between identical blocks was defined as 2.0 times the maximum traffic between non-identical blocks. An initial temperature of 15 units and an annealing schedule of 0.9 was used. The layout obtained corresponded exactly to the optimal layout of Figure 4.3. It can be seen that the blocks representing entities 1 through 4, which are the ones represented by more than one building block, have been placed together in the final solution.

Chapter 5

Implementation

The programs for both the cell formation and the layout problems have been coded in the 'C' programming language and run on a Unix-based Sun 3/50 workstation. The software has been structured in such a way that either the entire shop-floor design problem can be addressed sequentially or its various modules can be individually used for cell-formation or shop/intra-cell layout. Since the programs are very efficient, a large number of simulations may be carried out to test various alternatives and obtain better results. The cell-formation example of section 3.4 was solved in less than 2 seconds of c.p.u time, while the layout example of section 4.5 required 5.4 seconds of c.p.u. time.

5.1 Industrial Application

The proposed cell-formation and layout methods were utilized to partially re-arrange the existing production facility of a large manufacturing company. The approach followed and the solutions obtained are discussed below.

5.1.1 Company Background

Kop-Flex, Inc. is a manufacturer of flexible shaft couplings, clutches and related rotational parts. The company's product lines consist of mechanically flexible gear couplings, of which Kop-Flex is the world's largest supplier, and a wide range of material-flexible couplings, which incorporate either elastomer materials or metal membrane elements.

Kop-Flex manufactures almost 1,200 types of make-to-stock parts. In addition, an average of 1,500 make-to-order part types are in process at any given point in time. All manufactured parts are cylindrical and span a wide range of sizes and shapes. The machine shop includes over 200 work-centers, such as lathes, gear-shapers, drills, milling machines, etc., which are currently functionally arranged.

5.1.2 Case Analysis

The make-to-stock part base was used as the foundation of the analysis, since; (i) it is almost static over time, (ii) strong similarities exist between the make-to-stock and make-to-order parts with respect to both design features and manufacturing characteristics, and (iii) common work-centers are used for the manufacture of both these segments of the part population.

The data required for the case analysis included: (i) the list of work-centers, (ii) the production routings of all 1186 make-to-stock parts, including the set-up and run times for each operation, (iii) the production volume of these parts over a certain period of time (16 months), and (iv) the average production batch-size and the pallet-size for each part. The validity of the part data was verified by comparing the standard production routings against the time keeping records, with both the actual sequence of operations performed and the actual set-up and run times for each operation. Where necessary, the routings were revised to reflect consistent current practices. Note that the functionally identical machines were represented by common designations in the part routings. This indicated that a part could be processed by any machine in a group of identical machines. Thus, the available routing data was in a format consistent with the requirements of the method applied. As discussed in section 3.3, these routings would be transformed during the cell-formation process to identify the specific machine within its group on which each part is to be processed.

After the completion of the data collection and validation phase, a certain amount of data pre-processing had to be performed. This mainly consisted of the removal of certain work-centers from the part routings. These included: (i) Central facilities such as packing, storage areas, work-ups, etc., which are visited by almost all the parts in the system. (ii) All inspection work-centers, since those inspection operations that are not performed in-process require special metrology equipment which cannot be assigned to any particular cell. (iii) Work-centers that process most of the manufactured parts, such as saws, rough-out drills, etc. The following data was used as input to the cell-formation method:

Inputs:

Total number of parts = 1186

Number of machine types = 70

Number of groups of functionally identical machines = 9

The I.D.'s of the functionally identical machine groups, the number of machines in each group and the capacity of each machine within a period of 16 months are shown below.

I.D.	Number in machine type	Capacity in hours
12	2	6250
13	10	6250
18	10	6250
19	3	6250
20	2	6250
27	2	6250
36	2	6250
53	2	6250
55	5	6250

The batch and pallet-sizes for all the parts were set to unity in the absence of such data in the company records.

5.1.3 Results of the cell-formation method

The following parameters were used for the proposed cell-formation algorithm: (i) The maximum number of machines allowed in any cell (cell size constraint) was set to 8 after consultation with the company's staff. (ii) The least traffic between any two cells which justified merging of the cells was set to zero.

Only 10 proper cells were obtained. Five of these contained eight machines each, two consisted of seven machines each, two of the cells contained two machines each and one cell contained three machines. Almost all of the 1186 parts belong to the part families that correspond to one of these cells.

Of the remaining machines, 20 were assigned to individual cells since there was no traffic between them and the other machines. Seven machines from group 13, seven from group 18 and four from group 55 were not placed in any cell. These machines are not required for processing the parts in the make-to-stock segment of the company, and can be used elsewhere.

Of the resulting ten cells, only seven major cells were proposed for implementation on the company's shop-floor. The small number of parts visiting the other three cells does not justify their implementation. The compositions of these cells are shown below. The numbers in the parentheses indicate the machine types.

Cell	Machines
1	1, 9, 10, 22, 73(13), 83(18), 99(27)
2	2, 3, 15, 26, 71(12), 74(13), 84(18), 102(53)
3	4, 5, 6, 21, 72(12), 85(18), 95(19), 100(36)

Cell	Machines
4	8, 30, 32, 38, 39, 41, 43, 61
5	16, 17, 93(19), 96(20), 97(20), 98(27), 101(36)
6	25, 56, 59, 64, 68, 69, 103(53), 104(55)
7	44, 45, 47, 51, 54, 62, 63, 66

The above results indicate that all identical machines, with the exception of the two machines of type 20, have been placed in different cells. The placement of both the machines of type 20 in the same cell was necessitated by capacity requirements. Moreover, these machines are only used by the parts which belong to cell 5, thus justifying their placement in that cell.

The efficiency values for the proposed cellular layout are as follows:

Global Efficiency = 76.459984

Group Efficiency = 65.857986

G.T. Efficiency = 79.956663

The last of these efficiency values indicates the potential reduction in the inter-cell traffic if the layout type is changed from the present (functional) configuration to a cellular one.

For comparison purposes the KBGT system [30] proposed by Kusiak and Wadood, which also addresses the case of identical machines, was also applied to the problem at hand for comparison purposes. This method yielded 18 manufacturing cells with sizes ranging from 2 to 7 machines. However, only 475 of the 1186 parts were assigned to part families, while the remaining 711 parts were placed in the part waiting list, since none of them could be manufactured in a single cell. This is consistent with the objective of the KBGT system, which forms cells and part families in such a way that each part is manufactured exclusively by the work centers in the corresponding cell, or by bottleneck machines. The value of the G.T. efficiency for the cellular layout obtained from this method was only 21 percent. Note that this efficiency measure favours the grouping method proposed here since it is directly related to the objective of the method. Nevertheless, the results obtained for the industrial case still indicate the effectiveness of the proposed method in restricting material flow between the manufacturing cells. Note that the company has started establishing some of the proposed cells.

5.1.4 Results of the layout method

The intra-cell layout was determined for the seven major cells obtained from the cell-formation algorithm. The specific routings generated by the cell-formation method were used to determine the traffic between individual machines. All machines were considered to be equidimensional in this analysis. An initial temperature of 15.0 and an annealing schedule of 0.9 was used for generating these layouts. The results are shown in Figure 5.1.

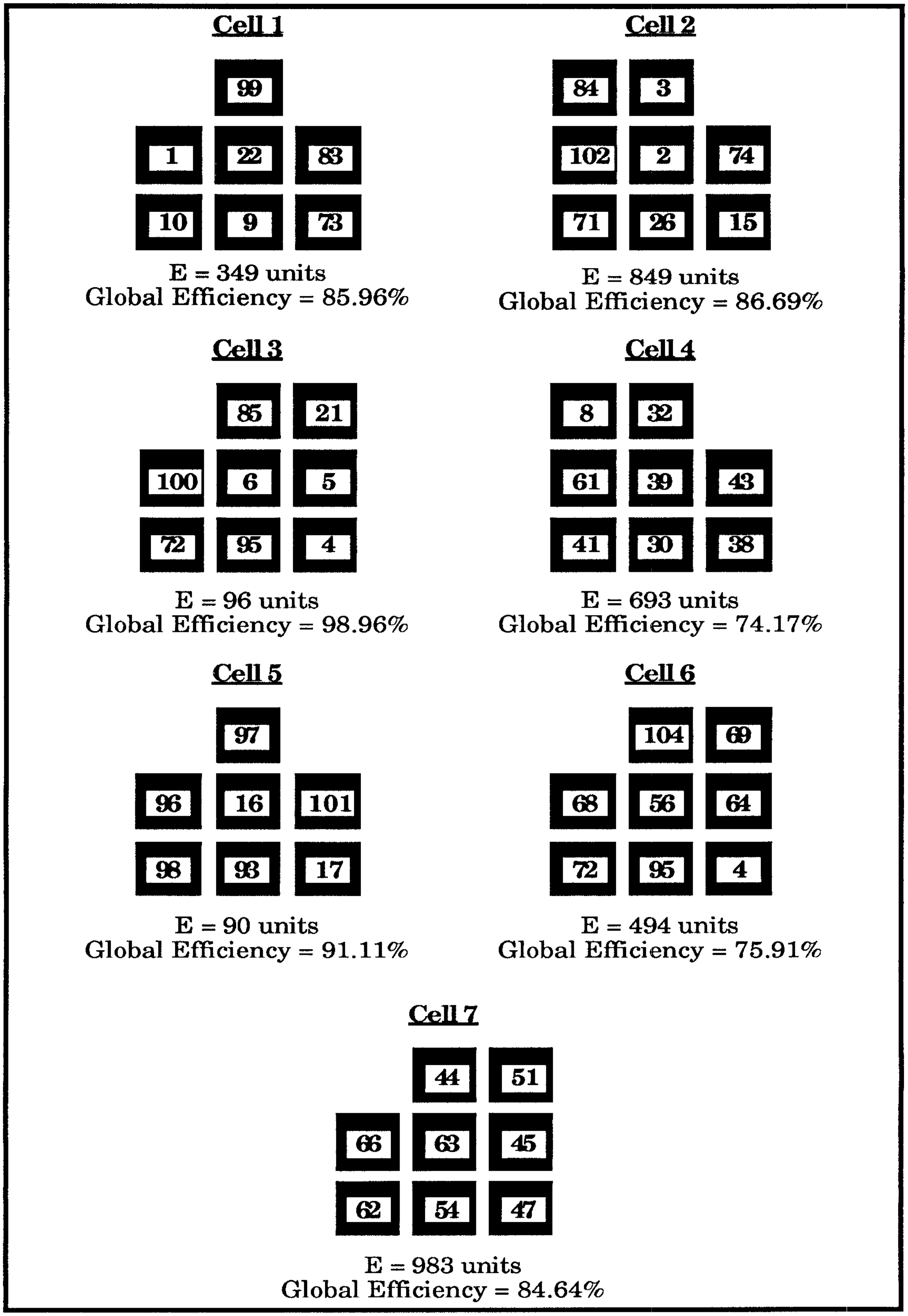


Figure 5.1: Intra-cell layouts for the industrial application

Implementation of the proposed layouts in the company's shop-floor would yield substantial savings in the annual material handling and work-in-process inventory costs and would also help in simplifying production control to a large extent.

Chapter 6

Conclusions & Recommendations for further work

6.1 Conclusions

New methods for both the cell-formation and the layout problems have been presented. Both methods address a host of practical issues, some of which had either been ignored or inadequately treated by past methods.

The cell-formation method minimizes the inter-cell material flow within the shop and distributes the functionally identical machines to the various cells in a way consistent with this objective, while respecting the capacity of each machine. Given a set of generic production routings for the parts manufactured in the system, the method also selects specific machines on which these parts are to be processed. Although the primary application of the system would be to re-arrange an existing functionally laid out job shop into manufacturing cells, it is equally applicable to planned facilities with forecasted product demand and known manufacturing processes. Moreover, the algorithm allows for simulations with "what if" questions regarding the number of machines of each type available, and cell-size limits, and provides the appropriate cell configurations for each case.

The major features of the proposed cell-formation system can be summarized as follows:

- The functionally identical machines are assigned to cells based on both traffic and capacity considerations.
- The sequence of operations is taken into account while evaluating the traffic between machines and/or cells.
- Multiple, non-consecutive operations on the same machine type are taken into account and are individually treated.
- The method minimizes the total number of inter-class movements of pallets as opposed to movements of individual parts. However, individual part transfers can also be considered as a special case, by setting the pallet size equal to unity for all parts.
- Both the set-up and run times along with the average batch-sizes are used to compute the capacity requirements for each part.
- Encroached cases, wherein a perfect decomposition of the manufacturing facility into cells is not possible, can also be addressed.

- The algorithm provides for simulations for determining the optimum number of machines of each type to minimize the traffic while ensuring sufficient capacity availability.

The proposed cell-formation method can be complemented by the work of Ang and Willey [19], who consider a given layout and propose a number of strategies to optimize the real-time performance of the production system by transferring workloads between functionally identical machines belonging to different cells, based on a given cellular layout. These strategies can be used for enhancing system performance once the manufacturing cells have been formed.

The proposed layout method overcomes the major drawback of other methods which address this problem, i.e. the dependence of the final solution on the initial layout. Thus the solution obtained usually corresponds to the best solution for the given problem. An additional feature of this method is that a number of alternate efficient layout designs of nearly the same quality can be easily obtained, the most suitable among which can be selected for implementation. This is possible since the proposed method is a probabilistic one, and consequently, different solutions can be obtained starting from the same initial conditions. This is not possible in deterministic heuristics like CRAFT, CORELAP, etc., wherein for a given set of initial conditions, the same final solution is obtained. Different initial layouts have to be provided to these methods to generate alternate solutions.

This becomes very difficult and tedious as the magnitude of the problem increases.

An attempt has also been made to account for the approximate entity sizes, which is a very important and practical consideration. This feature, although not yet subjected to rigorous testing, has a lot of potential for application to those practical cases wherein the assumption of equidimensionality for the entities might lead to unrealistic solutions. Use of this feature in some hypothetical problems has resulted in a significant amount of success in obtaining the optimal layout for entities with unequal area requirements.

6.2 Future Work

A very important consideration that needs to be examined for the practical application of the proposed methods is the robustness of the solutions obtained, in terms of the material flow, to projected changes in the product mix. A manufacturing system is dynamic and the parts produced in the system may undergo vast changes in terms of both design characteristics and production volumes. Therefore, an efficient layout should fulfil not only the current requirements of the production facility, but also the future production requirements.

Factors other than the traffic flow, such as the costs associated with machine relocation, production costs on different machines, etc. may also be considered in the manufacturing cell formation stage. In addition, workload criteria such as machine utilization and load balancing could be incorporated in the grouping algorithm for the assignment of parts to machines. In the proposed algorithm the functionally identical machines were considered to be completely interchangeable. Relaxation of this constraint to consider work-centers which can only process *some* common parts would be a step in the right direction. This would entail accommodating the presence of alternate routings for the parts produced. This kind of a situation is very common in a manufacturing facility and involves the use of non-identical machines for performing the same operation.

Apart from these, exact machine dimensions (or areas) and shapes could be explicitly included in the cell-size constraints. This consideration, assumes great significance in the case of a manufacturing facility containing entities with a wide range of sizes and shapes, which require to be allocated to pre-specified departmental areas with finite sizes.

For the case of facility layout, the validity of entity representation by building blocks must be examined. This could be accomplished by subjecting the method to rigorous testing using a wide variety of hypothetical and practical cases involving entities with different sizes. Development of empirical formulae for defining the traffic between identical blocks is also required.

Another major enhancement of the layout method would be the consideration of actual shop-floor dimensions along with the walls and other obstacles which impede regular traffic flow. In this case traffic patterns around the obstacles and along regular traffic flow paths would have to be determined. The simple objective function which only includes material handling distance can be enhanced to include certain other parameters such as traffic congestion along channels, accessibility, intersecting flow-lines, etc.

Finally, the proposed methods could be enhanced by including actual part dimensions and other physical criteria such as weight and volume as input parameters to the system. This feature, coupled to a graphics interface for simulation purposes, would aid decision-making regarding the inter- and intra-cell material handling system requirements. The enhanced method could thus be integrated into an expert system to address the entire decision-making process during the design of a manufacturing facility. Such a prototype expert system would comprise methods for machine selection, machine grouping and inter- and intra-cell layout.

Appendix

User's Manual

Introduction

This manual to provides a description of the working of the software for the layout design of a cellular manufacturing shop. The functions performed by each of the programs, together with the input required and the output generated are presented.

This system basically determines the machine cells and the corresponding part families on the basis of the part and machine population in the system, and the part routings. The latter need not specify the particular machines to be visited. Instead, the machine type is sufficient for the system to select which specific machine must be used for the processing of a part, so that capacity constraints are respected. Finally, the system is capable of indicating the proximity between cells and the proximity of machines within a cell based on material flow between them.

All programs have been coded in the 'C' programming language. The approach used has been such that the whole shop-floor design problem can be addressed in the proper sequence. On the other hand, the modularity of the whole package is maintained so that one can directly access the program of one's choice without having to go through the preceding steps. Sufficient flexibility has been incorporated in the layout program to allow its use even in the absence of pre-specified part routings, provided some quantifiable measures of required closeness between entities are available.

Cell Formation

Following is a description of the programs related to the disaggregation of the manufacturing system into cells.

Main Program

The Program: C-Form.c

What it does: This program generates the manufacturing cells based upon the criterion of minimizing the total inter-cell traffic of parts in the system. Another important output of this program is the set of specific routings for the parts which visit representatives of one or more groups of functionally identical machines. At each step of this program, the following functions are performed: i) The traffic between all pairs of cells is calculated. For a detailed description of the traffic calculation procedure, see Section 3.3. ii) The two cells with the maximum normalized traffic between them are identified and merged, subject to the cell size constraint. iii) If one or both of the cells involved in the merging contain(s) a group of functionally identical machines, one machine from each group is renumbered to distinguish it from the other members of the group. In addition, the routings of all the parts which visit such machine type(s) and which contributed to the traffic under consideration, are changed to reflect this renumbering.

After the conclusion of the iterative cell-formation procedure, each member of the groups of functionally identical machines has a unique identification, as opposed to group I.D.'s at the beginning of the program. If, however, some part routings still remain which do not reflect the new machine designations, the appropriate changes are made at this stage.

Inputs:

Information	Input format
(1) Number of part and machine types	file 'routings'
(2) Generic routings of each part type including the machine types visited and the set-up and unit run times for each operation	file 'routings'
(3) Production volume of each part over a given	file 'partdata'

period of time, its average batch and pallet sizes	
(4) Number of functionally identical work-center groups, I.D. of each, number of machines in each group and the capacity of each machine over the same period of time as in (3) above	file 'rwcidfile'
(5) Maximum number of machines allowed in a cell	interactive
(6) Minimum traffic between two cells to allow their merging	interactive

The maximum number of machines allowed per cell (item # 5 above) depends on several factors, such as the volume of work, the interdependence between workers and machines, the labor skills required, the sizes of the machines, etc. This value can be varied and several iterations carried out in order to obtain a suitable disaggregation of the manufacturing system into cells. The minimum traffic between two cells which justifies their merging (item # 6 above) is usually maintained at zero. A higher value may be specified if the user does not wish to allow the merging of two cells with too low a value of traffic between them.

Outputs:

Information	Output format
(1) Specific routing of each part indicating the machines visited for processing	file 'inf'
(2) Production volume of each part, its average batch and pallet sizes	file 'usage_for_layout'

(3) Production volume of each part	file 'usage'
(4) Cell number corresponding to each machine	file 'cells' (also system file 'pos2')
(5) Cross-reference for new machine I.D.'s versus original I.D.'s	file 'machxref

Item number (2) above (output file 'usage_for_layout) need not be the same as the input file 'partdata'. This is due to the reason that, owing to capacity constraints during the process of cell formation, the production volume of a certain part type may be fulfilled by two or more alternate routings. In such a case the number of part types in the output of the program would not be the same as that provided as input to it.

The files 'pos2' and 'usage' are generated since they are input requirements for some of the other programs which form the part families and structure the output in a matrix arrangement .

The file system for the cell formation program is shown in Fig.(A1). Note that one has to run certain other programs (courtesy Nagi [40]) in order to determine the part families and obtain the final formatted output in the form of files 'struc1' through 'struc4'. The former is performed by the program 'arrpron', while the program 'structure' performs the latter function, as well as determines the values of the efficiencies discussed in Section 3.4. For a description of these programs, see Nagi [40].

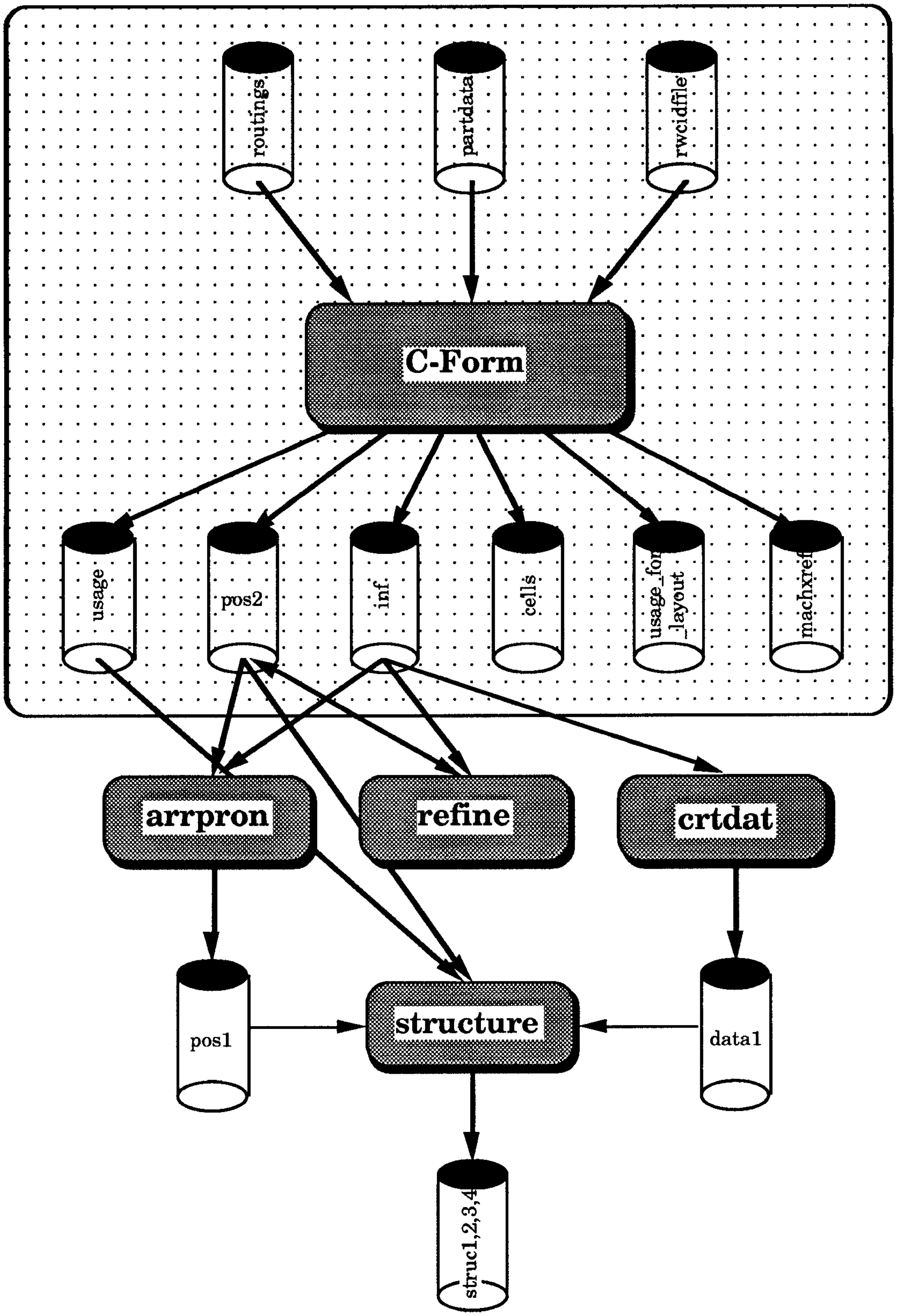


Figure A1: File system for the cell formation program

Performing Simulations with C-Form

The cell formation program can also be used to carry out simulations to assist decision-making regarding procurement of additional machines of various types. This can be done as follows: If, after running the program and observing the output, the user detects the presence of any bottleneck machine(s) which contributes to substantial inter-cell traffic, he/she can specify the presence of more machines of such type in the file 'rwcidfile' (item # 4 in the inputs list) and re-run the program. If this results in a better disaggregation of the manufacturing system, with higher efficiency values, (the decision to procure more machines of such type may be justified.

Data Validation Program

The Program: validate.c

What it does: This programs validates the data contained in the input files 'routings', 'partdata' and 'rwcidfile' by checking if the data is logically correct. Diagnostic messages are displayed on the standard output (monitor). The major diagnostics are performed on the file 'routings' to check if: i) the number of routings matches the number indicated at the top of the file, ii) any set-up or run time is missing in a part routing, iii) if consecutive operations are performed on the same machine type and iv) if work-center I.D.'s indicated in part routings remain within the maximum number of machine available.

Inputs:

Information	Input format
(1) Part routing information	file 'routings'
(2) Part production and material handling information	file 'partdata'
(3) Functionally identical work-center information	file 'rwcidfile'

Outputs:

Information	Input format
(1) Diagnostic Messages	display on screen

Efficiency Calculation Program

The Program: gteffcalc.c

What it does: This program calculates the G.T. Efficiency of the disaggregation achieved through the program 'C-Form'. Note that the program 'structure' also provides the value of this efficiency, along with the other two efficiency values, in the formatted output file 'struc1'.

However, this program only considers part transfers instead of pallet transfers as used in the cell formation procedure. Thus, discrepancies may exist between the two efficiency values. This program gives the true G.T. Efficiency of the disaggregation of the system into manufacturing cells.

Inputs:

Information	Input format
(1) Part routing information	file 'inf'
(2) Part production volume, and pallet-size information	file 'usage_for_layout'
(3) Cell information	file 'cells'

Outputs:

Information	Output format
(1) For each part, number of moves in current layout, number of moves in cellular layout and the saving in moves, G.T.Efficiency.	file 'out'

Layout

Following is a description of the program which determine the physical layout of the manufacturing entities. This program can be used for the layout of both the machines within a cell and the cells on the shop-floor.

The Program: CLASS.c

What it does: This program performs the physical layout of the manufacturing entities based on the method of Simulated Annealing. The criterion which drives the layout process is the minimization of the the total material handling distance (cost) between the manufacturing entities. Traffic values between the entities are provided as inputs, along with the solution space wherein the entities are to be placed. Traffic information can be provided directly or indirectly in the form of part routings, production volumes and pallet-sizes. The program performs pairwise interchanges of position 'contents' in order to minimize the objective function. The acceptance or rejection of the interchanges are governed by the (probabilistic) Metropolis criterion. The 'temperature' is reduced after a certain number of iterations in order to reduce the probability of acceptance of unfavourable interchanges. The procedure stops after a certain number of temperature steps, or when the solution converges, whichever occurs earlier. A graphic display of the system configuration is provided when the program is used in a Suntools environment. The display is automatically updated at the end of each temperature step.

Although originally designed for equidimensional entities, the program is also equipped to handle entities of different sizes, the relative sizes of the entities being represented by means of an integer number of building blocks for each entity. Traffic, both between blocks of the same entity and different entities, are automatically calculated within the program. Traffic between blocks of different entities are calculated in accordance with Eq.(4.5), while traffic between blocks belonging to the same entity are specified to be 2.0 times the maximum of these values. The traffic thus defined, the individual blocks are then considered as entities and the same procedure is utilized for their placement.

Inputs:

Information	Input format
(1) Entity information	
Indirect input	
(i) Traffic information	
Part routings	file 'inf'
Part production volume, and pallet-size information	file 'usage_for_layout'
(ii) Cell and entity size information	file 'cellswsize'

or

Direct input	
(i) Traffic, cell and entity size information	file 'dir_trf'
(2) Solution Space	interactive
(3) Starting Temperature and annealing schedule (*TFACTOR*)	interactive
Optional	
(4) Initial placement of entities in the solution space	interactive

As mentioned earlier, the entity and traffic information can be provided either directly through the file 'dir_trf', or it can be derived from the output of the cell formation program, in which case the program automatically calculates the traffic from the files 'inf' and 'usage_for_layout' and accesses cell information from the file 'cellswsize' (the file 'cells', modified to include machine sizes). Note that the layout of a single cell, or the inter-cell layout, can only be performed if the input is provided through the file 'dir_trf'. The file system for the layout program is shown in Figure A2.

The solution space can also be interactively provided in two ways. The program initially asks the user to specify the gridsize (a single numeral specifying the size of a square grid) for display purposes, which also becomes the solution space if the user so desires. In this case, the program automatically calculates the distances between the positions in the grid using the Manhattan distance criterion. On the other hand, the user can specify a custom solution space consisting of at least the minimum number of positions to accommodate all the entities. In this case, the user has to interactively input the distances between the positions, and the gridsize loses all significance apart from controlling the graphic display.

The user also has to specify the starting temperature and the annealing schedule (item # 3 above) for carrying out the annealing. The starting temperature should be sufficiently high so that virtually all configuration changes are accepted at that temperature. This temperature can be determined by trial and error. The annealing schedule can be any fraction between 0 and 1. A value of 0.9 for this works reasonably well for all applications.

The initial placement of the entities in the solution space (item # 4 above) can be interactively input by the user if so desired, or the program automatically places each entity at a distinct position, using a random number generator.

Outputs:

Information	Output format
(1) Entity location status at the conclusion of each temperature stage	graphic display in Suntools environment
(2) Final locations of all the entities and the layout Global Efficiency	file 'res_layout'

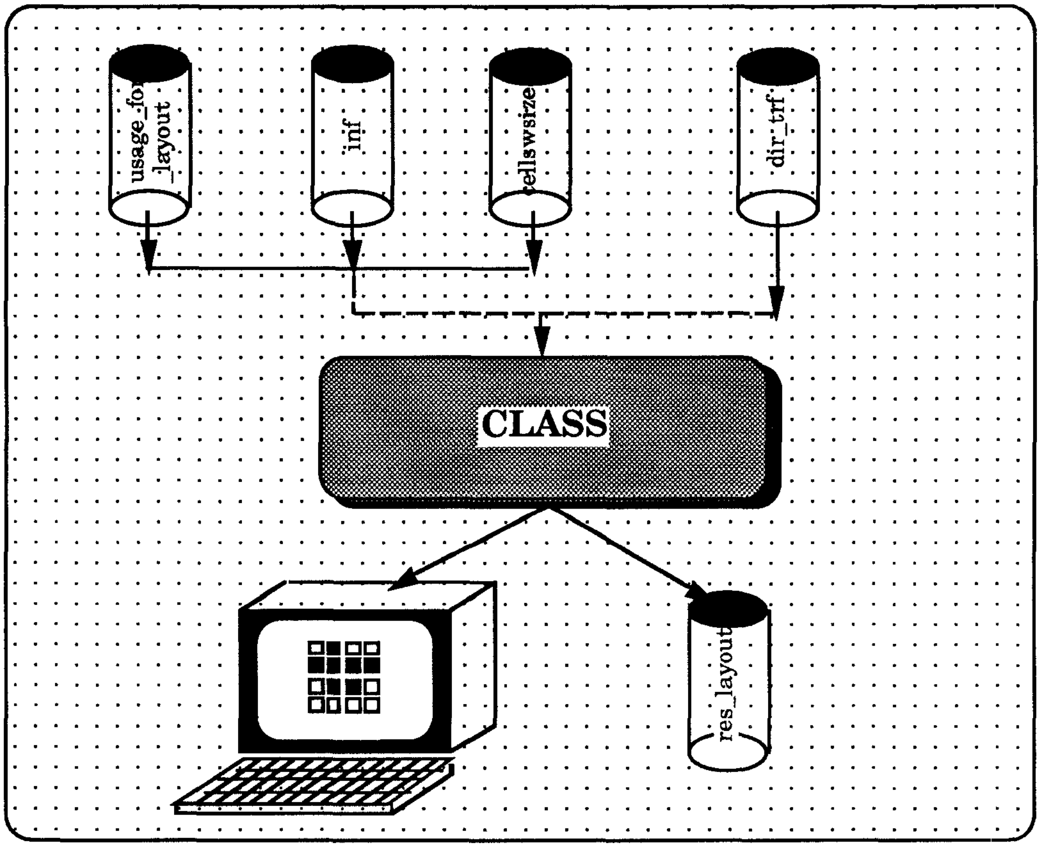


Figure A2: File system for the layout program

File Formats

1. routings:

n **m** <--- line 1

n	number of part types
m	number of machine types

lines 2 through (n+1)

1 $\mathbf{m_1(1)\ \ \theta_1^{su}(1)\ \ \theta_1^{run}(1)\ \ m_1(2)\ \ \theta_1^{su}(2)\ \ \theta_1^{run}(2)\ldots}$

2 $\mathbf{m_2(1)\ \ \theta_2^{su}(1)\ \ \theta_2^{run}(1)\ \ m_2(2)\ \ \theta_2^{su}(2)\ \ \theta_2^{run}(2)\ldots}$

•

•

•

n $\mathbf{m_n(1)\ \theta_n^{su}(1)\ \theta_n^{run}(1)\ m_n(2)\ \theta_n^{su}(2)\ \theta_n^{run}(2)}$ **...**

$m_1(1)$	first machine in routing of part 1
$\theta_1^{su}(1)$	set-up time for first operation on part 1
$\theta_1^{run}(1)$	run time for first operation on part 1

2. partdata

lines 1 through n

$\mathbf{N_1 \quad \beta_1 \quad \mu_1}$

$\mathbf{N_2 \quad \beta_2 \quad \mu_2}$

•

•

•

$\mathbf{N_n \quad \beta_n \quad \mu_n}$

N_1	production volume of part 1
β_1	average batch-size of part 1
μ_1	pallet-size of part 1

3. rwcidfile

p <--- line 1

p	number of functionally identical work-center groups

lines 2 through (p+1)

G_1 K_1 T_1
G_2 K_2 T_2
•
•
•
G_p K_p T_p

G_1	I.D. of first functionally identical work-center group
K_1	number of machines in first functionally identical work-center group
T_1	capacity of each machine in first functionally identical work-center group

4. inf

Same as 'routings', except that only the machines in each routing are indicated, without the set-up and run times. Moreover, the number of routings in this file may be more than the number present in the file 'routings'. This happens (during the program C-Form) when the total production volume of part is fulfilled by two or more alternate routings owing to capacity constraints.

5. usage_for_layout

Same as 'partdata'. The number of entries in this file is equal to the number of routings in the file 'inf'. Note that the batchsize is <u>not</u> used in the program 'CLASS'. However, this format was retained to maintain consistency with the format of the file 'partdata'.

6. usage

Same as 'usage_for_layout', without the batch and pallet-sizes for each part.

7. cells

lines 1 through m (≥m)

$\mathbf{b_1}$

$\mathbf{b_2}$

•

•

•

$\mathbf{b_m}$

b_1	cell number to which machine 1 belongs

8. cellswsize

Same as 'cells', except that it has a second column indicating the number of building blocks for representing each machine.

9. dir_trf

k <--- line 1

References

[1] Aneke, N.A.G. and Carrie, A.S., "A Design Technique for the Layout of Multi-Product Flowlines," *International Journal of Production Research*, 24(3), 1986, pp.471-481.

[2] Ang, C.L. and Willey, P.C.T., " A Comparative Study of the Performance of Pure and Hybrid Group Technology Manufacturing Systems using Computer Simulation Techniques," *International Journal of Production Research*, 22(2), 1984, pp.193-233.

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

[4] Askin, R. and Subramanian, S.B., "A Cost-Based Heuristic for Group Technology Configuration," *International Journal of Production Research,* 25(1), 1987, pp.101-113.

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

[6] Carrie, A.S., "Numerical Taxonomy applied to Group Technology and Plant Layout," *International Journal of Production Research,* 11(4), 1973, pp.399-416.

[7] Carrie, A.S., "Computer-Aided Layout Planning - The Way Ahead," *International Journal of Production Research,* 18(3), 1980, pp.283-291.

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

[9] Chan, H.M. and Milner, D.A. "Direct Clustering Algorithm for Group Formation in Cellular Manufacturing," *Journal of Manufacturing Systems*, 1 (1), 1982, pp.65-74.

[10] Chittajallu, S.K., "Methodology for the Economic Evaluation of Alternative Part Presentation Systems," *Ph.D. Thesis*, The Pennsylvania State University, 1989.

[11] Choobineh, F., "A Framework for the Design of Cellular Manufacturing Systems," *International Journal of Production Research*, 26(7), 1988, pp.1161-1172.

[12] Co, H.C. and Araar, A., "Configuring Cellular Manufacturing Systems," *International Journal of Production Research,* 26(9), 1988, pp.1511-1522.

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

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

[15] Drezner, Z., "DISCON: A New Method for the Layout Problem," *Operations Research*, 28(6), 1980, pp.1375-1384.

[16] Flynn, B.B. and Jacobs, F.R., "A Simulation Comparison of Group Technology with Traditional Job Shop Manufacturing," *International Journal of Production Research,* 24(5),1986, pp.1171-1192.

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

[18] Garcia, H. and Proth, J.M., "Group Technology in Production Management: The Short Horizon Planning Level," *Applied Stochastic Models and Data Analysis*, 1, 1985, pp. 25-34.

[19] Garcia, H. and Proth, J.M., "A New Cross-Decomposition Algorithm: The GPM. Comparison with the Bond Energy Method," *Control and Cybernetics,* 15 (2), 1986, pp. 155-164.

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

[21] Ham, I., Hitomi, K. and Yoshida, T., "Group Technology: Applications to Production Management," *Kluwer-Nijhoff Publishing*, 1985.

[22] Harhalakis, G., Nagi, R. and Proth, J.M., "An Efficient Heuristic in Manufacturing Cell Formation for Group Technology Applications," *International Journal of Production Research,* 28(1), 1990, pp.185-198.

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

[24] Kernighan, B.W. and Ritchie, D.M., "The C Programming Language," *Prentice-Hall of India Pvt. Ltd.*, 1988.

[25] King, J.R., "Machine-Component Grouping Using ROC Algorithm," *International Journal of Production Research,* 18 (2), 1980.

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

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

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

[29] Kusiak, A. and Herangu, S.S., "Group Technology," *Computers in Industry*, 9, 1987, pp. 83-91.

[30] Kusiak, A. and Wadood, I., "Knowledge Based System for Group Technology (KBGT)," *Proceedings of the 1st International Conference on CIM*, RPI, Troy, NY, 1988, pp. 184-193.

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

[32] Leskowsky, Z., Logan, L. and Vannelli, A., "Group Technology Decision Aids in an Expert System for Plant Layout," *Modern Production Management Systems*, Elsevier Science Publisher, 1987, pp. 561-583.

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

[34] Malakooti, B. and Tsurushima, A., "An Expert System using Priorities for Solving Multiple-Criteria Layout Problems," *International Journal of Production Research*, 27(5), 1989, pp.793-808.

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

[36] Martin, J.M., "Cells Drive Manufacturing Strategy," *Manufacturing Engineering*, Jan., 1989, pp.49-54.

[37] McAuley, J., "Machine Grouping for Efficient Production," *The Production Engineer*, 51 (53), 1972, pp. 53-57.

[38] McCormick, W.T., Schweitzer, P.J. and White, T.E., "Problem Decomposition and Data Re-organization by a Clustering Technique," *Operations Research*, 20 (5), 1972, pp. 993-1009.

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

[40] Nagi, R., "Selection and Layout of Facilities for Cellular Manufacturing," *M.S. Thesis,* University of Maryland, Dec., 1988.

[41] Nagi, R., Harhalakis, G. and Proth, J.M., "Multiple Machines and Capacity Considerations in Group Technology Applications," accepted for publication in the *International Journal of Production Research.*

[42] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T., "Numerical Recipes in C: The Art of Scientific Computing," Cambridge University Press, 1988.

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

[44] Seifoddini, H., "Duplication Process in Machine Cells Formation in Group Technology," *IIE Transactions*, 21(4), 1989, pp.1161-1165.

[45] Tabucanon, M.T. and Ojha, R., "ICRMA - A Heuristic Approach for Intercell Flow Reduction in Cellular Manufacturing Systems," *Material Flow*, 4, 1987, pp.189-197.

[46] Tam, K. "An Operation Sequence Based Similarity Coefficient for Part Families Formations," Journal of Manufacturing Systems, 9(1), 1990, pp.55-68.

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

[48] Wei, J.C. and Kern, G.M., "Commonality Analysis: A Linear Cell Clustering Algorithm for Group Technology," *International Journal of Production Research*, 27(12), 1989, pp.2053-2062.

[49] Wemmerlov, U. and Hyer, N.L., "Cellular Manufacturing in the U.S. Industry: A Survey of Users," *International Journal of Production Research*, 27(9), 1989, pp.1511-1530.