6

LAYOUT PLANNING MODELS AND DESIGN ALGORITHMS

6.1 INTRODUCTION

The generation and evaluation of a number of layout alternatives is a critical step in the facilities planning process, since the layout selected will serve to establish the material flow patterns and physical relationships between activities. Recognizing that the layout ultimately selected will be either chosen from or based on one of the alternatives generated, it is important for the facilities planner to be both creative and comprehensive in generating a reasonable number of layout alternatives.

Most facility layouts can be viewed at two levels: The *block layout* (which shows the location, shape, and size of each planning department) and the *detailed layout* (which shows the exact location of all equipment, work benches, and storage areas *within* each department). The block layout is concerned primarily with the "macro" flows in the facility, while the detailed layout is often concerned with the "micro" flows. For the facility layout to be complete and effective, both the block layout and the detailed layout need to be developed and evaluated carefully. Although some of the models and techniques we show in this chapter can be applied to detailed layouts as well, our focus is primarily on quantitative methods for developing and evaluating alternative block layouts.

We previously addressed how to determine the requirements. In Chapter 1, we treated the strategic relationship between facilities planning and manufacturing, distribution, and marketing. From that discussion, we recognized the importance of taking a long-range viewpoint and coordinating the facilities plan with the plans of other organizational units. A facilities layout strategy should emerge from the overall strategic plan. Product, manufacturing, marketing, distribution, management, and human resource plans will be impacted by and will have an impact on the facilities layout.

The facilities requirements resulting from product design, process design, and schedule design decisions were examined in Chapter 2. The impact of personnel requirements on space, proximities, and special features of the facility was treated in Chapter 4.

Chapter 3 provided a comprehensive treatment of activity relationships, space requirements, and types of planning departments as they relate to facilities planning. From that discussion, as well as the emphasis in Chapter 1 on the establishment of activity relationships and determination of space requirements, it is clear that both the block layout and the detailed layout are critically important in designing a layout for a facility.

Before proceeding further, it seems appropriate to address two questions, which frequently come up in a layout planning project:

1. Which comes first, the material handling system or the facility layout?
2. Which comes first, the block layout or the detailed layout?

Many appear to believe the layout should be designed first and then that the material handling system should be developed. Yet material handling decisions can have a significant impact on the effectiveness of a layout. For example, the following decisions will affect the layout:

1. Centralized versus decentralized storage of work-in-process (WIP), tooling, and supplies
2. Fixed-path versus variable-path material handling
3. The handling unit (unit load) planned for the systems versus one-piece flow
4. The degree of automation used in handling
5. The type or level of inventory control, physical control, and computer control of materials

Each of the above considerations affects the requirements for space, equipment, and personnel, as well as the degree of proximity required between various departments.

Why do people tend to focus first on layout? Perhaps one reason is an overemphasis on the manufacturing process. For example, it seems perfectly logical to place department B next to department A "if process B occurs immediately after process A." In such a situation, the handling problem is reduced to the question, "What is the best way to move materials from A to B?" Conventional wisdom suggests that the handling problem can be addressed after the layout is finalized.

However, if materials cannot flow directly from department A to department B, then WIP storage is required in A, B, and/or elsewhere. Depending on the storage and control requirements, a centralized WIP storage area might be used, so that materials flow from A to S (storage), and then from S to B. With such a centralized WIP storage system, materials do not flow from A to B, and B no longer needs to be placed next to A. Furthermore, the centralized system provides added flexibility when the process sequence changes. Yet lean manufacturing teaches us to redefine the departments and create "cells" so that the parts flow, one at a time, directly from one workstation to the next, which eliminates intermediate storage areas and significantly reduces the material handling distances and delays incurred due to batching.

Another reason for the "layout first" approach could be a misapplication of the "handling less is best" adage. For example, in one-piece flow (the make one, move one principle practiced in lean manufacturing), the parts are handled more times compared to batching (when all the parts are placed in one container and moved only once as a unit load). Yet, due to delays introduced by batching, in a large majority of cases, one-piece flow is the preferred approach, provided the workstations are placed next to each other. Hence, when one says "handling less is best," one needs to be careful in how handling is defined and how less or more handling impacts measures such as WIP inventory, the expected time it takes for the parts to travel through the system, and walk times/material handling times for the operators.

So, which comes first, the material handling system or the facility layout? Our answer is, "Both!" The layout and the handling system should be designed simultaneously. The complexity of the design problem, however, generally requires that a sequential process be used. For this reason, we recommend that a number of alternative layout plans be developed and the appropriate handling system be designed (at least at a conceptual level) for each alternative. The preferred layout will be that which results from a consideration of the system as a whole [72].

As for the second question, our recommendation is to first obtain basic requirements for each department (such as space requirements, shape constraints, and so on) and then develop a set of alternative block layouts. Once the desirable block layout is identified, the analyst can then develop a detailed layout for each department. In the process, he/she may very well go back and "massage" or "modify" the block layout, creating an iterative process, which can be repeated for some of the other block layout alternatives as well. Since the process often ends up being an iterative one, the starting point is less critical, although gathering basic information for each department and then starting with macro-flows is often a practical approach.

6.2 BASIC LAYOUT TYPES

In Chapter 3, we identified four types of planning departments:

1. Fixed material location departments
2. Production line departments
3. Product family departments
4. Process departments

Based on the above types of departments, we also identified four types of layouts, as shown earlier in Figure 3.37 and Table 3.2. (For the reader's convenience, Figure 3.37 is repeated here as Figure 6.1.)

Before we explain in the next section various procedures developed for generating and evaluating layout alternatives, we need to stress that identifying the right number and types of departments that are going to be used to populate the layout is a key step the facilities planner must perform first. A critical document for this purpose is the future state map (FSM). The facilities planner must establish very clearly the relationship between each "process box" in the FSM and each "department" that is to appear in the layout. Skipping this step (or performing it with a poorly defined

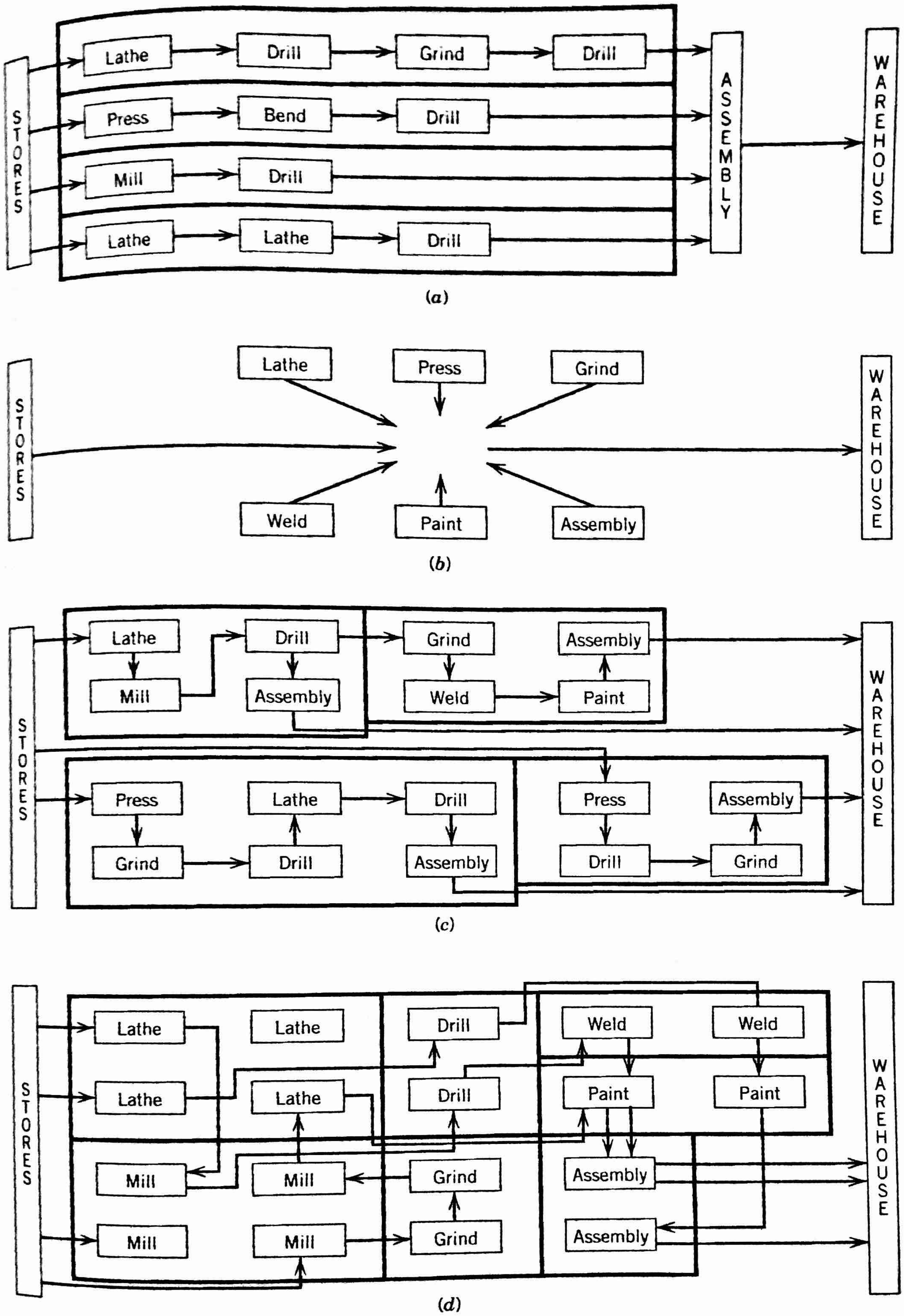


Figure 6.1 Alternative types of layouts. (*a*) Production line product layout. (*b*) Fixed product layout. (*c*) Product family layout. (*d*) Process layout.

FSM) may result in a layout that minimizes the flow distances between the "wrong" departments; that is, the flow distances in the layout may be minimized, but the departments that populate the layout may not be the right type of departments needed to support lean manufacturing. (For an introduction to value stream mapping and basic lean principles used to develop an FSM, the reader may refer to [54].)

6.3 LAYOUT PROCEDURES

A number of different procedures have been developed to aid the facilities planner in developing layout alternatives. These procedures can be classified into two main categories: construction type and improvement type. *Construction* layout methods basically involve developing a new layout "from scratch." *Improvement* procedures, on the other hand, generate layout alternatives by seeking improvements in an existing layout.

Although a number of papers in the facility layout literature focus on the development of construction-type procedures, most layout work still involves some form of improving the layout of an existing facility. As Immer [29, p. 32] observed as early as the 1950s,

> Much of [the] work will consist of making minor changes in [an] existing layout, locating new machines, revising a section of the plant, and making occasional studies for material handling. The plans for a complete new production line or a new factory may make the headlines, but except for a war or a new expansion, the average layout [planner] will very seldom have to consider such a problem.

While Immer's observation still holds today for the most part, some changes to the layout are no longer "minor" changes (due to the far-reaching impact lean manufacturing has had in the manufacturing sector), and there are new plants being built around the globe, as manufacturing spreads from traditional bases such as the United States to other regions of the world.

We begin our discussion of layout procedures by discussing some of the original approaches to the layout problem. The concepts in these approaches continue to serve as the foundation of many of the methodologies proposed today.

6.3.1 Apple's Plant Layout Procedure

Apple [2] proposed the following detailed sequence of steps in producing a plant layout.

1. Procure the basic data.
2. Analyze the basic data.
3. Design the productive process.
4. Plan the material flow pattern.
5. Consider the general material handling plan.
6. Calculate equipment requirements.
7. Plan individual workstations.
8. Select specific material handling equipment.
9. Coordinate groups of related operations.
10. Design activity interrelationships.
11. Determine storage requirements.
12. Plan service and auxiliary activities.
13. Determine space requirements.
14. Allocate activities to total space.
15. Consider building types.

16. Construct master layout.
17. Evaluate, adjust, and check the layout with the appropriate persons.
18. Obtain approvals.
19. Install the layout.
20. Follow up on implementation of the layout.

Apple noted that these steps are not necessarily performed in the sequence given. As he put it,

> Since no two layout design projects are the same, neither are the procedures for designing them. And, there will always be a considerable amount of jumping around among the steps, before it is possible to complete an earlier one under consideration. Likewise, there will be some backtracking going back to a step already done—to re-check or possibly re-do a portion, because of a development not foreseen [2, p. 14].

6.3.2 Reed's Plant Layout Procedure

Reed [53] recommended the following "systematic plan of attack" as required steps in "planning for and preparing the layout."

1. Analyze the product or products to be produced.
2. Determine the process required to manufacture the product.
3. Prepare layout planning charts.
4. Determine workstations.
5. Analyze storage area requirements.
6. Establish minimum aisle widths.
7. Establish office requirements.
8. Consider personnel facilities and services.
9. Survey plant services.
10. Provide for future expansion.

Reed calls the layout planning chart "the most important single phase of the entire layout process" [53, p. 10]. It incorporates the following:

1. Flow process, including operations, transportation, storage, and inspections
2. Standard times for each operation
3. Machine selection and balance
4. Manpower selection and balance
5. Material handling requirements

An example of a layout planning chart is given in Figure 6.2. Such charts can be viewed as the predecessors of value stream maps used in lean manufacturing today [54].

6.3.3 Muther's Systematic Layout Planning (SLP) Procedure

Muther [49] developed a layout procedure he named systematic layout planning, or SLP. The framework for SLP is given in Figure 6.3. It uses as its foundation the activity relationship chart described in Chapter 3 and illustrated in Figure 6.4.

LAYOUT PLANNING CHART

PART NO. 1 PART NAME PLASTIC CONTACT PAD PCS/ASSY 1 PCS REQ/HR 70.4
ASSY NO. — ASSY NAME — ASSY/PRODUCT 2 PRODUCTION HRS/DAY 6.0 SHEET 1 OF 1
MATERIAL PLASTIC SIZE 11/2" OD × 3/8" ID (from 4' × 8' × 1/4" SHEETS) PCS/DAY 422 LOT SIZE 1 PREPARED BY J.G.D. DATE 1-4-60
APPROVED BY DATE

ST NO.	F M S I	DESCRIPTION	OPER NO.	DEPT NO.	TIME PER PIECE	MACHINE OR EQUIPMENT	MACHINES REQD MACH FRAC	COMB WITH	MACH REQD	OPER PER MACH	TOTAL MANPOWER CREQ FRAC	MAN FRAC	COMB WITH	MEN REQD	HANDLING REQUIREMENTS HOW MOVED	CONT TYPE	LOAD SIZE	DIST MOVED	REMARKS
1		FROM MATERIALS STORAGE													FORK LIFT	PALLET	4 SHEETS	100'	
2		ON PALLET BY SAW		2															
3		TO SAW TABLE																	
4		SAW INTO STRIPS 2 1/2" × 8'	10	2	.02	TABLE SAW	.028	9 – 10 16 – 10	1	2	.028	.056	9 – 10 16 – 10	2					
5		TO RACK BY SAW																	
6		IN RACK		2									1 – 20 14 – 30						
7		TO HEATER											17 – 20 3 – 20						
8		IN RACK BY HEATER		2									6 – 30 13 – 20						
9		FEED INTO HEATER											14 – 20						
10		HEAT	10	2	.04	HEATER	.055	9 – 20 16 – 20	1	1	.055	.055	6 – 20 6 – 10	1					
													3 – 10 2 – 10						
													9 – 20 16 – 20						
													13 – 10 14 – 10						
													17 – 10						
11		FEED TO PUNCH PRESS																	
12		PUNCH TO SHAPE	10	2	.04	PUNCH PRESS	.055	4 – 10 4 – 20	1	1	.055	.055	4 – 10 4 – 20	1					
13		TP BIN BY PUNCH PRESS						9 – 30 16 – 30					9 – 30 16 – 30						
14		IN BIN		2				19A – 10 19B – 10					19A – 10 19B – 10						
15		TO PARTS STORAGE											19A – 20 5A – 10		4 WHEEL HAND TRUCK	TOTE BOX	400 PER BOX	150'	
16		IN PARTS STORAGE		5															
17		TO ASSEMBLY													4 WHEEL HAND TRUCK	TOTE BOX	400 PER BOX	60'	
18		IN BIN IN ASSEMBLY		6															
19		TO TABLE																	

Figure 6.2 Layout planning chart.

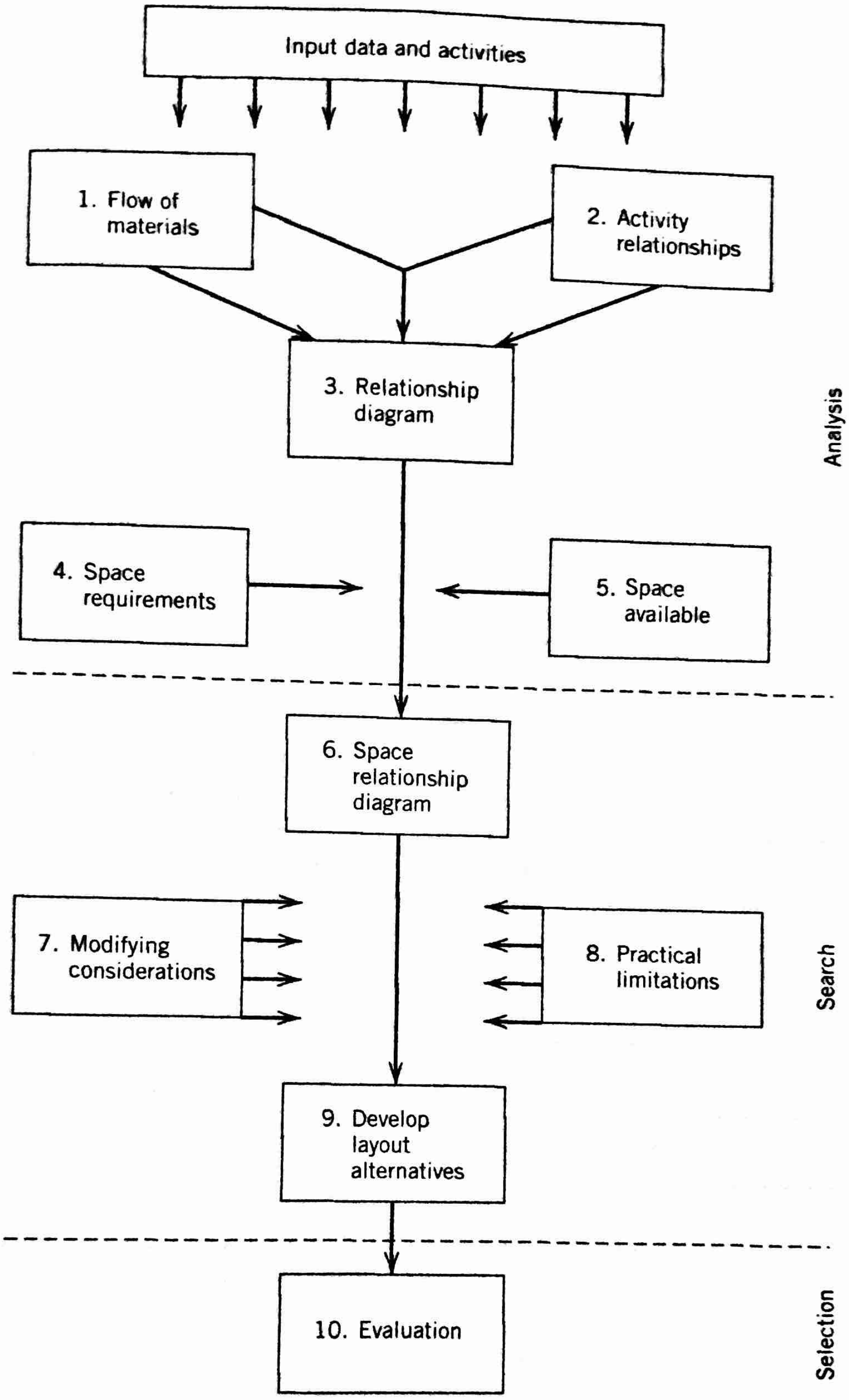


Figure 6.3 Systematic layout planning (SLP) procedure.

Based on the input data and an understanding of the roles and relationships between activities, a material flow analysis (from-to chart) and an activity relationship analysis (activity relationship chart) are performed. From the analyses performed, a relationship diagram is developed (Figure 6.5).

The relationship diagram positions activities spatially. Proximities are typically used to reflect the relationship between pairs of activities. Although the relationship diagram is normally two-dimensional, there have been instances in which three-dimensional diagrams have been developed when multistory buildings, mezzanines, and/or overhead space were being considered.

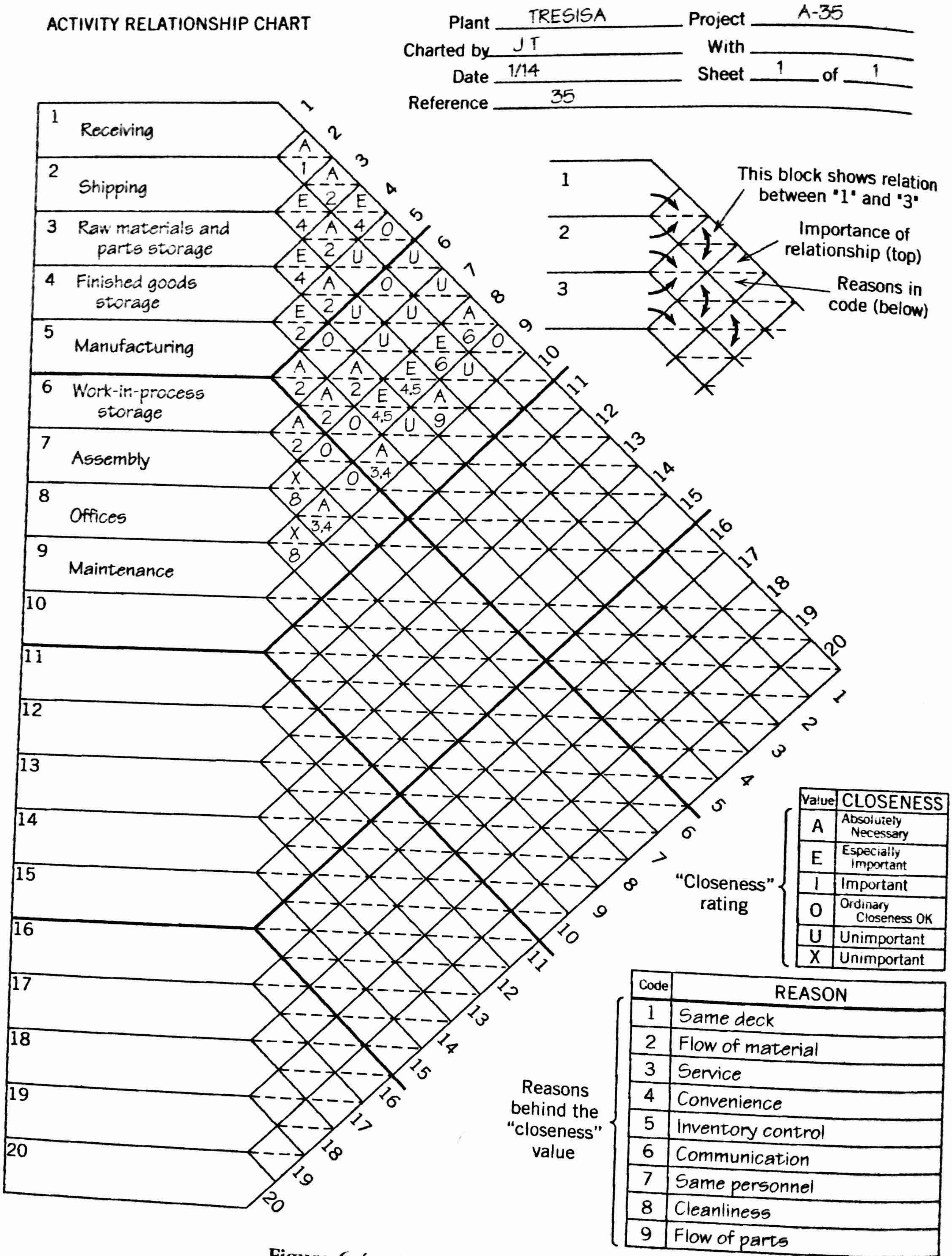


Figure 6.4 Activity relationship chart.

The next two steps involve the determination of the amount of space to be assigned to each activity. From Chapter 3, departmental service and area requirement sheets would be completed for each planning department. Once the space assignments have been made, space templates are developed for each planning department, and the space is "hung on the relationship diagram" to obtain the space relationship diagram (Figure 6.6).

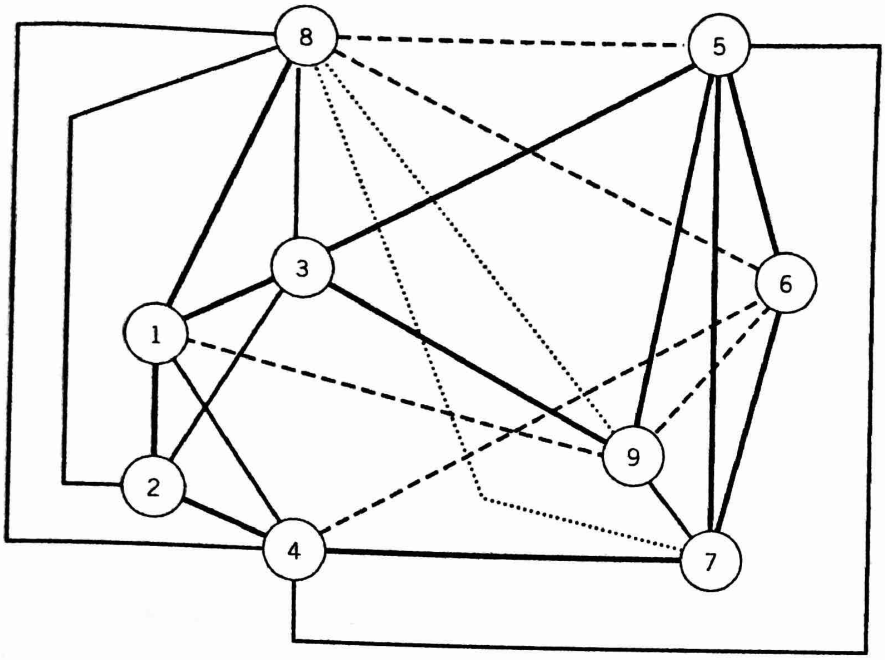


Figure 6.5 Relationship diagram.

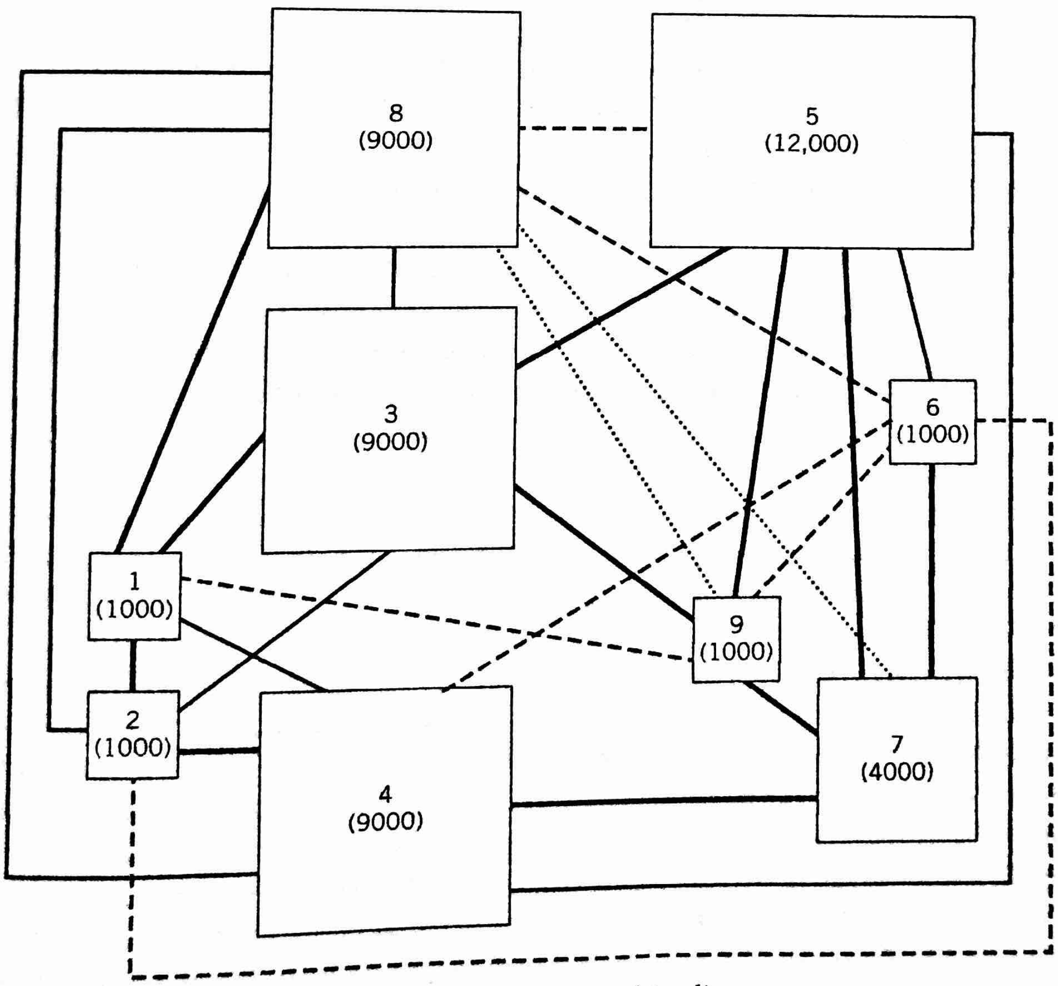


Figure 6.6 Space relationship diagram.

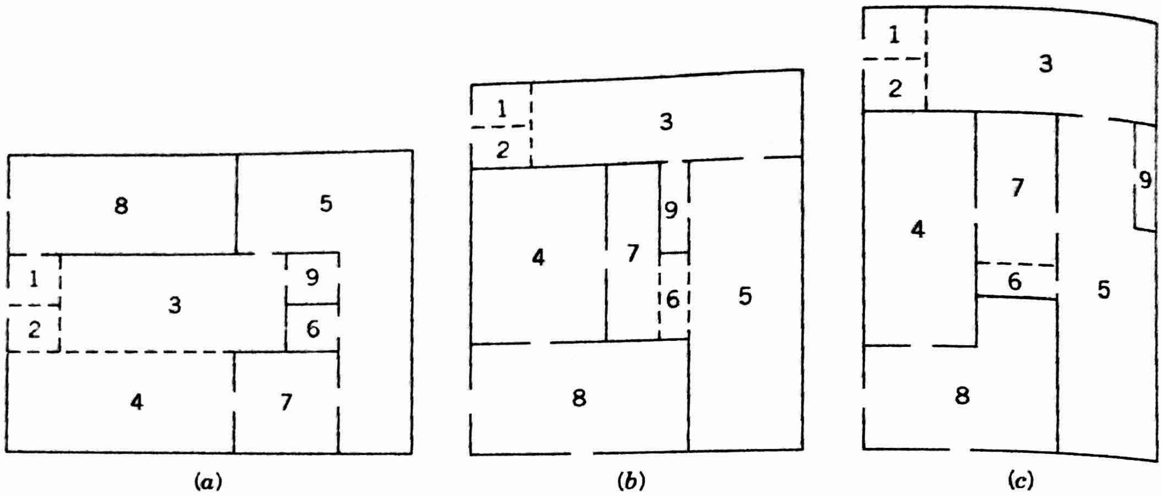


Figure 6.7 Alternative block layouts.

Based on modifying considerations and practical limitations, a number of layout alternatives are developed (Figure 6.7) and evaluated. The preferred alternative is then identified and recommended.

While the process involved in performing SLP is relatively straightforward, it does not necessarily follow that difficulties do not arise in its application. We addressed in Chapter 3 such issues as the *a priori* assignment of activity relationships and the use of proximity as a criterion for measuring the degree of satisfaction of activity relationships. In addition to those concerns, it should be noted that alternative relationship diagrams can often be developed, with apparent equivalent satisfaction of activity relationships. Likewise, the shapes of the individual space templates used in constructing the space relationship diagram can influence the generation of alternatives. Finally, the conversion of a space relationship diagram into several feasible layout alternatives is not a mechanical process: intuition, judgment, and experience are important ingredients in the process.

The SLP procedure can be used sequentially to develop first a block layout and then a detailed layout for each planning department. In the latter application, relationships between machines, workstations, storage locations, and entrances to and exits from the department are used to determine the relative location of activities within each department.

6.4 ALGORITHMIC APPROACHES

The relative placement of departments on the basis of their "closeness ratings" or "material flow intensities" is one that can be reduced to an algorithmic process. The layout procedures we presented in the previous section provide an excellent framework and overall process to construct or improve a layout, but they do not provide a formal procedure or algorithm for some of the critical steps associated with layout design and evaluation. The models and algorithms we present in this section are formal procedures that can help the layout analyst develop or improve a layout and at the same time provide him or her with objective criteria to facilitate the evaluation of various layout alternatives that emerge in the process.

The layout algorithms we present, at least in theory, can be executed by hand. However, for most practical, real-world problems, the algorithms we show here are intended or best-suited for computer implementation. Currently available computer-based layout algorithms cannot replace human judgment and experience, and they generally do not capture the qualitative characteristics of a layout. However, computerized layout algorithms can significantly enhance the productivity of the layout planner and the quality of the final solution by generating and numerically evaluating a large number of layout alternatives in a very short time. Computerized layout algorithms are also very effective in rapidly performing "what-if" analyses based on varying the input data or the layout itself.

The algorithms presented in this section, for the most part, represent the outgrowth of university research. As such, commercial versions of the algorithms we present either do not exist or they must be obtained through the original authors. Also, due to limited space, we are only able to present what we hope is a diverse and representative subset of the many layout algorithms that have been developed to date.

A number of commercial packages are available for facility layout. However, with some exceptions (see Section 6.8), such packages are either intended only for presentation purposes (that is, they are electronic drafting tools that facilitate the drawing or maintenance of a given layout) or they are designed primarily as a layout evaluation tool (that is, they can evaluate a layout, provided that one has been supplied by the layout planner). While such tools can also significantly enhance the productivity of the layout planner, in this section we will focus on those algorithms that can actually improve a given layout or develop a new layout from scratch.

6.4.1 Algorithm Classification

Most layout algorithms can be classified according to the type of input data they require. Some algorithms accept only qualitative "flow" data (such as a relationship chart), while others work with a (quantitative) flow matrix expressed as a from-to chart. Some algorithms (such as BLOCPLAN) accept both a relationship chart and a from-to chart; however, the charts are used only one at a time when evaluating a layout. The current trend appears to be toward algorithms that use a from-to chart, which generally requires more time and effort to prepare but provides more information on parts flow (or material handling trips) when completed. *Given that flow values can be converted to relationship ratings and vice versa*, most algorithms can be used with either type of data. Of course, if a relationship chart is converted to a from-to chart (by assigning numerical values to the closeness ratings), then the "flow" values picked by the layout planner represent only an ordinal scale.

With respect to data input, the time and effort required to compile a relationship chart or a from-to chart increases rapidly with the number of departments. Since the relationship chart is based on user-assigned closeness ratings, its construction often requires input from multiple individuals. After obtaining such input, the analyst needs to identify and resolve possible inconsistencies. For example, for the same pair of departments, one person may assign an "A," while another assigns an "I." Rather than attempt to decide who is "right" and who is "wrong," the analyst needs to meet with both individuals to understand the reasons behind their ratings and, with agreement from all concerned parties, determine the final closeness rating to be used for the pair. Given the significant increase in the number of all possible department pairs as a function of the number of departments, relationship charts are often not practical for problems with 20 or more departments.

The same increase in the number of department pairs also applies to from-to charts. For medium- to large-sized problems (say, 20–30 or more departments), filling out each entry in the from-to chart would not be practical. In such cases, however, the from-to chart can be constructed in reasonable time by using the production route data for each product (or product family). For example, if product type A is processed through departments 1-2-5-7, and is moved at a rate of, say, 20 loads per hour, then we set $f_{12} = f_{25} = f_{57} = 20$ in the from-to chart. Repeating this process for each product (or product type) completes the from-to chart, which is often a "sparse matrix" (i.e., it contains many blank entries). In fact, in many cases, it is useful to first construct a separate and complete from-to chart for each product (or product type) so product-level flow data remain available to the analyst at all times. Subsequently, individual product-level from-to charts can be combined into one cumulative chart, using appropriate weights for individual product types if necessary. Of course, the construction of the from-to chart in the above manner is done by computers in most cases. Also, if the unit of flow for the product changes as it moves from one process to the next, then appropriate multipliers can be inserted into the routing data to scale the flow intensity up or down based on the unit of flow.

Layout algorithms can also be classified according to their objective functions. There are two basic objectives: one aims at minimizing the sum of flows times distances, while the other aims at maximizing an adjacency score. Generally speaking, the former, that is, the "distance-based" objective—which is similar to the classical quadratic assignment problem (QAP) objective—is more suitable when the input data is expressed as a from-to chart, and the latter, that is, the "adjacency-based" objective, is more suitable for a relationship chart.

Consider first the distance-based objective. Let m denote the number of departments, f_{ij} denote the flow from department i to department j (expressed in number of unit loads moved per unit time), and c_{ij} denote the cost of moving a unit load one distance unit from department i to department j. The objective is to minimize the cost per unit time for movement among the departments. Expressed mathematically, the objective can be written as

$$\min z = \sum_{i=1}^{m} \sum_{j=1}^{m} f_{ij} c_{ij} d_{ij} \tag{6.1}$$

where d_{ij} is the distance from department i to j. In many layout algorithms, d_{ij} is measured rectilinearly between department centroids; however, it can also be measured according to a particular aisle structure (if one is specified).

Note that the c_{ij} values in Equation 6.1 are implicitly assumed to be independent of the utilization of the handling equipment, and they are linearly related to the length of the move. In those cases where the c_{ij} values do not satisfy the above assumptions, one may set $c_{ij} = 1$ for all i and j and focus only on *total unit load travel* in the facility (i.e., the product of the f_{ij} and the d_{ij} values). In some cases, it may also be possible to use the c_{ij} values as relative "weights" (based on unit load attributes such as size, weight, bulkiness, etc.) and minimize the weighted sum of unit load travel in the facility.

Consider next the adjacency-based objective, where the adjacency score is computed as the sum of all the flow values (or relationship values) between those departments that are adjacent in the layout. Letting $x_{ij} = 1$ if departments i and j are adjacent (that is, if they share a border) in the layout, and 0 otherwise, the objective is to maximize the adjacency score; that is,

$$\max z = \sum_{i=1}^{m} \sum_{j=1}^{m} f_{ij} x_{ij} \qquad (6.2)$$

Although the adjacency score obtained from Equation 6.2 is helpful in comparing two or more alternative layouts, it is often desirable to evaluate the *relative* efficiency of a particular layout with respect to a certain lower or upper bound. For this purpose, the layout planner may use the following "normalized" adjacency score:

$$z = \frac{\sum_{i=1}^{m} \sum_{j=1}^{m} f_{ij} x_{ij}}{\sum_{i=1}^{m} \sum_{j=1}^{m} f_{ij}} \qquad (6.3)$$

Note that the normalized adjacency score (which is also known as the *efficiency rating*) is obtained simply by dividing the adjacency score obtained from Equation 6.2 by the total flow in the facility. As a result, the normalized adjacency score is always between zero and 1. If the normalized adjacency score is equal to 1, it implies that all department pairs with positive flow between them are adjacent in the layout.

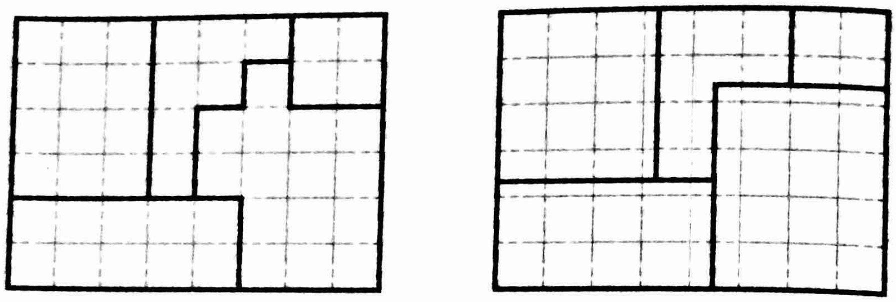
In some cases, the layout planner may represent an X relationship between departments i and j by assigning a negative value to f_{ij}. The exact negative value to be used should be determined with respect to the "real" (i.e., positive) flow values in the from-to chart. If such "negative flow" values are used, then the normalized adjacency score must be modified as follows:

$$z = \frac{\sum_{(i,j)\in F} f_{ij} x_{ij} + \sum_{(i,j)\in \bar{F}} f_{ij}(1 - x_{ij})}{\sum_{(i,j)\in F} f_{ij} - \sum_{(i,j)\in \bar{F}} f_{ij}} \qquad (6.4)$$

where F and $\bar{F}$ represent the set of department pairs with positive and negative flow values, respectively.

The adjacency-based objective has been used in a number of algorithms; see, for example, [15], [16], [17], and [47], among others. Although such an objective is easy to use and intuitive (i.e., department pairs with high closeness ratings or large flows need to be adjacent in the layout), it is generally not a complete measure of layout efficiency since it disregards the distance or separation between nonadjacent departments. Therefore, as remarked in [6], it is possible to construct two layouts that have identical or similar adjacency scores but different travel distances in terms of parts flow.

Layout algorithms can further be classified according to the format they use for layout representation. Most layout algorithms use the discrete representation (shown in Figure 6.8*a*), which allows the computer to store and manipulate the layout as a matrix. With such a representation, the area of each department is rounded off to the nearest integer number of grids. If the grid size is too large (too small), small (large) departments may have too few (too many) grids. Also, the grid size determines the overall "resolution" of the layout; a smaller grid size yields a finer resolution, which allows more flexibility in department shapes. However, since the



(*a*) (*b*)

Figure 6.8 Discrete versus continuous layout representation. (Adapted from [18].)

total area is fixed, a smaller grid size results in a larger number of grids (or a larger matrix), which can considerably increase the computational burden. Hence, in algorithms that use the discrete representation, selecting the appropriate grid size is an important decision that must be made early in the planning process.

The alternative representation (see Figure 6.8*b*) is the continuous representation where there is no underlying grid structure. (The grid in Figure 6.8*b* was retained only for comparison purposes.) Although such a representation is theoretically more flexible than its discrete counterpart, it is also more difficult to implement on a computer. In fact, except for one case (where the shapes of certain departments are "adjusted" to accommodate a nonrectangular building), present computerized layout algorithms that use the continuous representation are restricted to a rectangular building and rectangular department shapes.

While it is possible to model nonrectangular buildings by using fixed "dummy" departments (see the section on CRAFT), generally speaking, defining L-shaped, U-shaped, and other arbitrary, nonrectangular departments with the continuous representation is not straightforward. If a department is rectangular, and we know its area, then we need to know only the *x*, *y*-coordinate of its centroid and the length of its side in the north–south direction to specify its exact location and shape. (How would you specify the exact location and shape of an L-shaped department or a T-shaped department with a known area? Among alternative specifications, which one requires the minimum amount of data? Which one makes it easier to identify and avoid overlapping departments?)

Department shapes play an important role in computerized layout algorithms. Although we discuss specific shape measures in Section 6.5, first we need to present some basics. Since, by definition, a department represents the smallest indivisible entity in layout planning, a layout algorithm should not "split" a department into two or more pieces. If some departments are "too large," the layout planner can go back and reconsider how the departments were defined and change some of them as necessary. In the process, one large department may be redefined as two smaller departments. Once all the departments have been defined, however, a layout algorithm cannot change them or split them.

Although the human eye is adept at judging shapes and readily identifying split departments, for the computer to "recognize" a split department, we need to develop formal measures that can be incorporated into an algorithm. Consider, for example, the discrete representation, where a department is represented as a collection of grids. Suppose there is a "dot" that can move *only from one grid to an adjacent grid*. (Two grids are adjacent only if they share a border of positive length; two

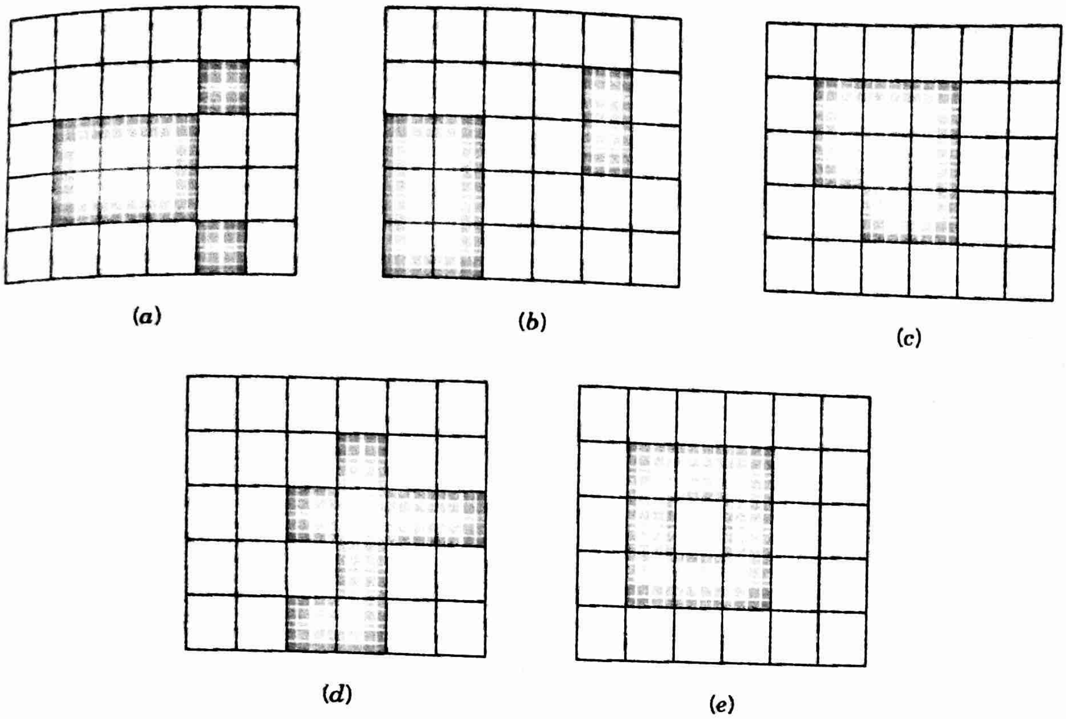


Figure 6.9 Examples of split and unsplit departments.

grids that "touch" each other at the corners are not considered adjacent.) We say that department *i* is not split if the above dot can start at any grid assigned to department *i* and travel to any other grid assigned to department *i* *without* visiting any grid that has not been assigned to department *i*. In other words, given the restrictions imposed on the movements of the dot, any grid assigned to department *i* must be "reachable" from any other such grid.

For example, the department shown in Figure 6.9*a* and *b* is split, while those shown in Figure 6.9*c* and *d* are not. However, according to the above definition, the department shown in Figure 6.9*e* is also not a split department. Departments such as the one shown in Figure 6.9*e* are said to contain an "enclosed void" [18] and as a rule of thumb are not considered practical or reasonable for facility layout purposes. (Of course, one possible exception to this rule is an "atrium." However, an atrium should generally be modeled by placing a fixed "dummy department" inside the building. Refer to the section on CRAFT for dummy departments.) The shape measures we present in Section 6.5 are devised to generally avoid department shapes such as those shown in Figure 6.9*d* and *e*. In the above discussion of department shapes, we focused on the discrete representation. Although it has been used effectively only with rectangular departments, our comments apply to the continuous representation as well.

Lastly, as we remarked earlier, layout algorithms can be classified according to their primary function—that is, layout *improvement* versus layout *construction*. Improvement-type algorithms generally start with an initial layout supplied by the analyst and seek to improve the objective function through "incremental" changes in the layout. Construction-type layout algorithms generally develop a layout "from scratch." They can be further divided into two categories: those that assume the building dimensions are given and those that assume they are not. The first type of construction algorithm is suitable when an operation is being moved into an existing, vacant building. The second type of construction algorithm is suitable for "green field" applications.

Even with "green field" applications, however, there is usually a *site plan*, which shows the property, surrounding roads, etc. Given the constraints imposed by the site plan, it is often necessary to construct the new building within a certain "envelope." If a construction algorithm of the second type is used, it is often difficult to ensure that the resulting building will properly fit into such an envelope. With construction algorithms of the first type, on the other hand, one may model the above envelope as an "existing building" to obtain a proper fit between the actual building and the envelope. Primarily for the above reason, the construction algorithms we present in this chapter are of the first type (i.e., they assume the building dimensions are given).

In the following sections, we describe the overall modeling techniques and/or methods used in various layout algorithms: namely, the pairwise exchange method, a graph-based method, CRAFT, BLOCPLAN, MIP, LOGIC, and MULTIPLE. (For ease of reference, in those cases where the original authors did not use an acronym, we created our own.) Detailed information such as input data format or output variables are not presented since such information is relevant and available only if the reader obtains a copy of the computer program. However, for each algorithm, we present sufficient detail not only so that the fundamental concepts are covered but also so that interested readers may develop their own basic implementation of the algorithm. The algorithms we present are relatively recent ones with a few exceptions, such as CRAFT, which we consider because it is one of the first layout algorithms developed and it provides a good platform for the others. Furthermore, all the algorithms we present require a basic understanding of heuristic search techniques (such as the "steepest-descent" procedure) and the difference between "locally" and "globally" optimal solutions. We selected these methods based on their conceptual contributions to layout planning methods and/or their unique approach to layout construction/improvement.

6.4.2 Pairwise Exchange Method

In an earlier discussion, we suggested that the majority of layout problems involve the redesign of an existing layout or existing facility, which is typically triggered by changes made to the building, kaizen events conducted in lean manufacturing, the addition of new machines, changes in product mixes, decisions related to the contraction and expansion of storage areas, or a simple realization that the old layout is no longer adequate for the firm's current needs. Thus, given an existing layout, the challenge is to develop an improved layout by making feasible or realistic changes to the existing layout.

The pairwise exchange method ([53], [9]) is an improvement-type layout algorithm. Although it can be used with both an adjacency-based and distance-based objective, it is often used with the latter. We will illustrate the pairwise exchange method below through an example based on equal-area departments for simplicity. Its implementation with unequal-area departments (which is the case in practice) will be shown later via CRAFT, MULTIPLE, and others.

Consider four departments of equal sizes. The material flow data are given in Table 6.1. The existing layout is shown in Figure 6.10*a*. A distance matrix can be obtained based on the existing layout as given in Table 6.2.

Table 6.1 ***Material Flow Matrix***

		To Department			
		1	2	3	4
From Department	1	—	10	15	20
	2		—	10	5
	3			—	5
	4				—

The distance-based objective function value (or "total cost") for the existing layout is computed as follows:

$$TC_{1234} = 10(1) + 15(2) + 20(3) + 10(1) + 5(2) + 5(1) = 125$$

The subscript notation indicates the order of the departments in the initial layout.

The pairwise exchange method simply states that for each iteration, all feasible exchanges in the locations of department pairs are evaluated *one at a time*, and the pair that results in the *largest reduction* in total cost is selected. (Moving in the direction of largest cost reduction is also known as "steepest descent" in optimization.) Evaluating each exchange one at a time means that when departments i and j are exchanged and the layout cost is computed, departments i and j are placed back in their current positions in the layout *before* the next pair of departments is exchanged.

Since all the departments are of equal size, and none of the departments has been declared fixed, the feasible exchanges for the above problem are 1-2, 1-3, 1-4, 2-3, 2-4, and 3-4. The distance matrix is recomputed each time an exchange is performed. The layout costs resulting from the above exchanges are

$$TC_{2134}(1\text{-}2) = 10(1) + 15(1) + 20(2) + 10(2) + 5(3) + 5(1) = 105$$
$$TC_{3214}(1\text{-}3) = 10(1) + 15(2) + 20(1) + 10(1) + 5(2) + 5(3) = \mathbf{95}$$
$$TC_{4231}(1\text{-}4) = 10(2) + 15(1) + 20(3) + 10(1) + 5(1) + 5(2) = 120$$
$$TC_{1324}(2\text{-}3) = 10(2) + 15(1) + 20(3) + 10(1) + 5(1) + 5(2) = 120$$
$$TC_{1432}(2\text{-}4) = 10(3) + 15(2) + 20(1) + 10(1) + 5(2) + 5(1) = 105$$
$$TC_{1243}(3\text{-}4) = 10(1) + 15(3) + 20(2) + 10(2) + 5(1) + 5(1) = 125$$

Using steepest descent, we select pair 1-3 and exchange the locations of departments 1 and 3 in the layout. The new layout is shown in Figure 6.10*b*.

(*a*) Iteration 0	1	2	3	4
(*b*) Iteration 1	3	2	1	4
(*c*) Iteration 2	2	3	1	4

Figure 6.10 Layouts corresponding to each iteration.

Table 6.2 *Distance Matrix Based on Existing Layout*

From Department	To Department 1	2	3	4
1	—	1	2	3
2		—	1	2
3			—	1
4				—

For the next iteration, we consider all feasible exchanges that consist of the same set as in iteration 1. The resulting layout costs are

$$TC_{3124}(1\text{-}2) = 10(1) + 15(1) + 20(2) + 10(2) + 5(1) + 5(3) = 105$$
$$TC_{1234}(1\text{-}3) = 10(1) + 15(2) + 20(3) + 10(1) + 5(2) + 5(1) = 125$$
$$TC_{3241}(1\text{-}4) = 10(2) + 15(3) + 20(1) + 10(1) + 5(1) + 5(2) = 110$$
$$TC_{2314}(2\text{-}3) = 10(2) + 15(1) + 20(1) + 10(1) + 5(3) + 5(2) = \mathbf{90}$$
$$TC_{3412}(2\text{-}4) = 10(1) + 15(2) + 20(1) + 10(3) + 5(2) + 5(1) = 105$$
$$TC_{4213}(3\text{-}4) = 10(1) + 15(1) + 20(2) + 10(2) + 5(1) + 5(3) = 105$$

The pair 2-3 is selected with a total layout cost of 90. Figure 6.10*c* shows the resulting layout after two iterations. Continuing on, the third iteration calculations are

$$TC_{1324}(1\text{-}2) = 10(2) + 15(1) + 20(3) + 10(1) + 5(3) + 5(1) = 120$$
$$TC_{2134}(1\text{-}3) = 10(1) + 15(1) + 20(2) + 10(2) + 5(3) + 5(1) = 105$$
$$TC_{2341}(1\text{-}4) = 10(3) + 15(2) + 20(1) + 10(1) + 5(2) + 5(1) = 105$$
$$TC_{3214}(2\text{-}3) = 10(1) + 15(2) + 20(1) + 10(1) + 5(2) + 5(3) = \mathbf{95}$$
$$TC_{4312}(2\text{-}4) = 10(1) + 15(1) + 20(2) + 10(2) + 5(3) + 5(1) = 105$$
$$TC_{2413}(3\text{-}4) = 10(2) + 15(1) + 20(1) + 10(3) + 5(1) + 5(2) = 100$$

Since the lowest layout cost for the third iteration, 95, is worse than the current layout cost of 90 (obtained in the second iteration), the procedure is terminated. The final layout arrangement is 2-3-1-4, as shown in Figure 6.10*c*. The final layout is also known as a "two-opt layout" since there are no two-way exchanges that can further reduce the layout cost.

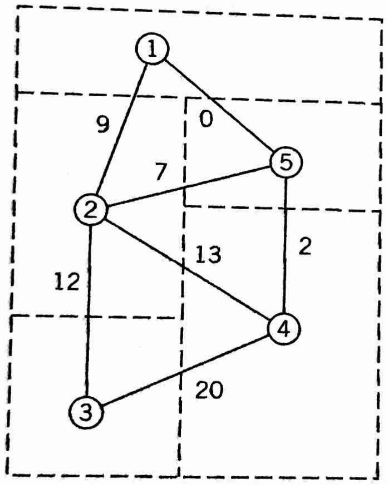
We stress that the pairwise exchange procedure described above is not guaranteed to yield the optimal layout because the final outcome is dependent on the initial layout; that is, a different initial layout often results in a different solution. Thus, we can only claim local optimality; that is, a two-opt layout is only locally optimal; it may or may not be globally optimal. Also, you may have observed that it is possible to cycle back to one of the alternative layout arrangements from a previous iteration. For instance, the layout arrangement 1-2-3-4 is what we started with, and we see the same arrangement in the second iteration when departments 1 and 3 were exchanged based on the solution obtained from iteration 1—that is, layout arrangement 3-2-1-4. Additionally, symmetric layout arrangements may also occur; for example, 4-3-2-1 in iteration 3 is identical to 1-2-3-4.

Note that a pairwise exchange can be easily accomplished if the pair of departments considered are of equal size (as we assumed in this example). Otherwise, we would have to expend some effort in rearranging the two departments being exchanged and possibly the other departments in the layout. We will discuss the exchange of unequal-area departments in CRAFT and subsequent algorithms.

6.4.3 Graph-Based Method

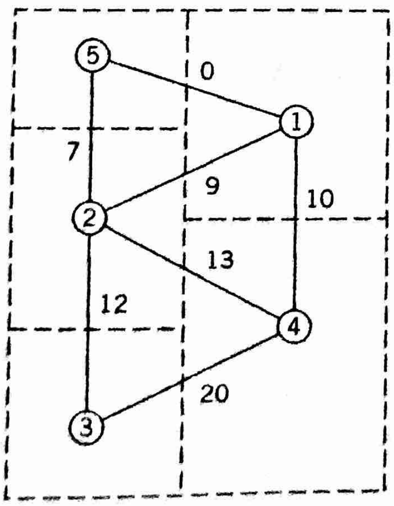
The graph-based method is a construction-type layout algorithm; it has its roots in graph theory. It is often used with an adjacency-based objective. The recognition of the usefulness of graph theory as a mathematical tool in the solution of facilities planning problems dates back to the late 1960s [35] and early 1970s [58]. Graph theory methods have similarities with the SLP method developed by Muther [49].

Consider the block layout shown in Figure 6.11*a*. We first construct an adjacency graph, where each node represents a department, with a connecting arc between two nodes indicating that two departments share a common border. A similar graph is constructed for the alternative block layout shown in Figure 6.11*b*. We observe that the two graphs shown in Figure 6.11 are subgraphs of the graph shown in



Arc	Weight
1-2	9
1-5	0
2-3	12
2-4	13
2-5	7
3-4	20
4-5	2
	63 (total)

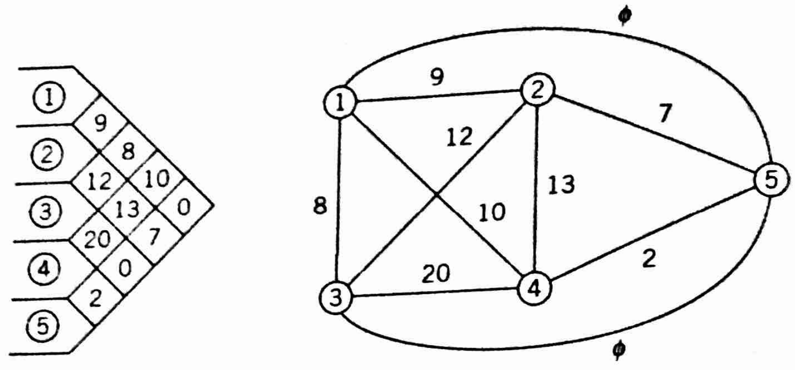
(*a*)



Arc	Weight
1-5	0
2-5	7
1-2	9
1-4	10
2-4	13
2-3	12
3-4	20
	71 (total)

(*b*)

Figure 6.11 Adjacency graphs for alternative block layouts.



(a) Relationship chart (b) Relationship diagram

Figure 6.12 Relationship chart and relationship diagram for graph-based example.

Figure 6.12*b*, which is derived from the relationship chart in Figure 6.12*a*. The relationship chart displays numerical "weights" rather than alphabetic closeness ratings.

Given the adjacency-based objective, block layout (b) is better than block layout (a) with scores of 71 and 63, respectively. Thus, finding a maximally weighted block layout is equivalent to obtaining an adjacency graph with the maximum sum of arc weights.

Before we describe a method for determining adjacency graphs, we first make the following observations:

a. The adjacency score does not account for distance, nor does it account for relationships other than those between adjacent departments.
b. Dimensional specifications of departments are not considered; the length of common boundaries between adjacent departments is also not considered.
c. The arcs do not intersect; this property of graphs is called planarity. We note that the graph obtained from the relationship diagram is usually a nonplanar graph.
d. The score is very sensitive to the assignment of numerical weights in the relationship chart.

There are two strategies we can follow in developing a maximally weighted planar adjacency graph. One way is to start with the graph from the relationship diagram and selectively prune connecting arcs while making sure that the final graph is planar. A second approach is to iteratively construct an adjacency graph via a node insertion algorithm while retaining planarity at all times. A heuristic procedure based on the second approach is described below.

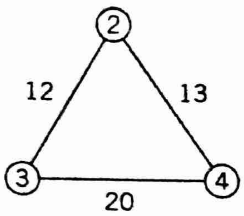
6.4.3.1 Procedure

Step 1. From the relationship chart in Figure 6.12*a*, select the department pair with the largest weight. Ties, if any, are broken arbitrarily. Thus, departments 3 and 4 are selected to enter the graph.

Step 2. Next, select the third department to enter. The third department is selected based on the sum of the weights with respect to departments 3 and 4. From Figure 6.13*a*, department 2 is chosen with a value of 25. The columns in this figure correspond to the departments already in the adjacency graph, and the rows correspond to departments not yet selected. The last column gives the sum of the weights for each unassigned department.

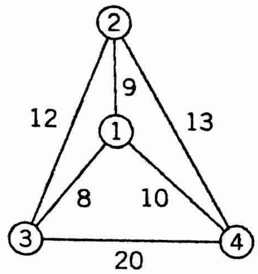
a. Step 2

	3	4	Total
1	8	10	18
2	12	13	25 (best)
5	0	2	2



b. Step 3

	2	3	4	Total
1	9	8	10	27 (best)
5	7	0	9	9



c. Step 4

	1	2	3	4
5	0	7	0	2

Faces	Total
1-2-3	7
1-2-4	9 (best)
1-3-4	2
2-3-4	9 (best)

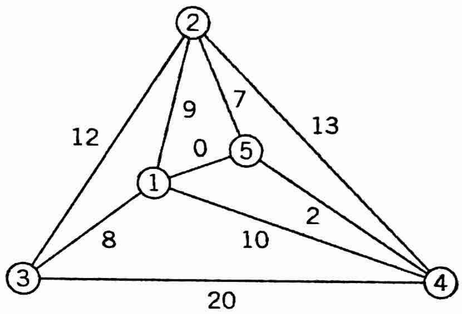
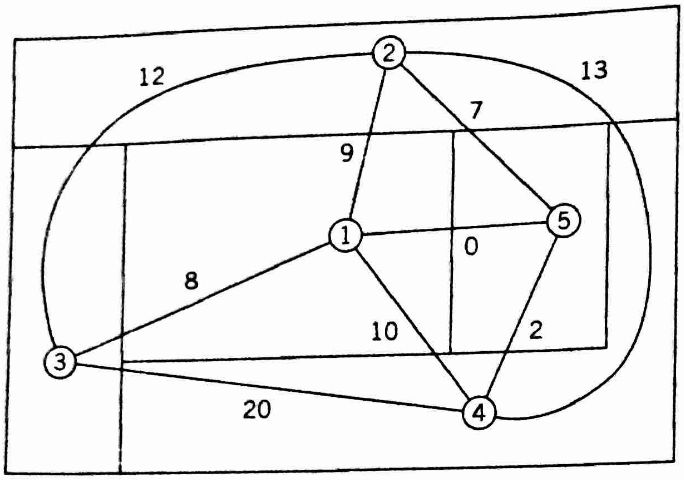


Figure 6.13 Steps of the graph-based procedure.

Step 3. We then pick the fourth department to enter by evaluating the value of adding one of the unassigned departments represented by a node on a face of the graph. A face of a graph is a bounded region of a graph. For instance, a triangular face is the region bounded by arcs 2-3, 3-4, and 4-2 in Figure 6.13*a*. We will denote this face as 2-3-4. The outside region is referred to as the external face. For our example, the value of adding departments 1 and 5 is 27 and 9, respectively. Department 1 is selected and placed inside the region 2-3-4, as shown in Figure 6.13*b*.

Step 4. The remaining task is to determine on which face to insert department 5. For this step, department 5 can be inserted on faces 1-2-3, 1-2-4, 1-3-4, and 2-3-4. Inserting 5 on faces 1-2-4 and 2-3-4 yields identical values of 9. We arbitrarily select 1-2-4. The final adjacency graph is given in Figure 6.13*c*. This solution is optimal, with a total sum of arc weights equal to 81.

Step 5. Having determined an adjacency graph, the final step is to construct a corresponding block layout. A block layout based on the final adjacency graph is shown in Figure 6.14. The manner by which we constructed the block layout is analogous to the SLP method. We should note that in constructing the block layout, the original department shapes had to be altered significantly in order to satisfy the requirements of the adjacency graph. In practice, we may not have as much latitude in making such alterations since department shapes are generally derived from the geometry of the individual machines within the department and the internal layout



Arc	Weight
1-2	9
1-3	8
1-4	10
1-5	0
2-3	12
2-4	13
2-5	7
3-4	20
4-5	2
	81 (total)

Figure 6.14 Block layout from the final adjacency graph.

configuration. We will discuss department shapes and their control later in this chapter (see Section 6.5). Finally, we should point out that there are algorithmic methods for performing this step as demonstrated by Giffin et al. [20] and Hassan and Hogg [23].

6.4.4 Craft

Introduced in 1963 by Armour, Buffa, and Vollman (see [3], [9]), the *Computerized Relative Allocation of Facilities Technique* (CRAFT) is one of the earliest layout algorithms presented in the literature. It uses a from-to chart as input data for the flow. Layout "cost" is measured by the distance-based objective function shown in Equation 6.1. Departments are not restricted to rectangular shapes, and the layout is represented in a discrete fashion.

Since CRAFT is an improvement-type layout algorithm, it starts with an initial layout, which typically represents the actual layout of an existing facility but may also represent a prospective layout developed by another algorithm. CRAFT begins by determining the centroids of the departments in the initial layout. It then calculates the rectilinear distance between pairs of department centroids and stores the values in a distance matrix. The initial layout cost is determined by multiplying each entry in the from-to chart with the corresponding entries in the unit cost matrix (i.e., the c_{ij} values) and the distance matrix.

CRAFT next considers all possible two-way (pairwise) or three-way department exchanges and identifies the best exchange, that is, the one that yields the largest reduction in the layout cost. (No department can be split as a result of a two-way or three-way exchange.) Once the best exchange is identified, CRAFT updates the layout according to the best exchange and computes the new department centroids as well as the new layout cost to complete the first iteration. The next iteration starts with CRAFT once again identifying the best exchange by considering all possible two-way or three-way exchanges in the (updated) layout. The process continues until no further reduction in layout cost can be obtained. The final layout obtained in such a manner is also known as a two-opt (three-opt) layout since no two-way (three-way) exchanges can further reduce the layout cost.

Since computers were relatively "slow" in the 1960s, the original implementation of CRAFT deviated slightly from the above description. When the program considered exchanging the locations of departments *i* and *j*, instead of actually exchanging the department locations to compute their new centroids and the actual layout cost, it computed an *estimated* layout cost simply by temporaily treating the centroid of department *i* in the current layout as the centroid of department *j* and vice versa; that is, it simply swapped the *centroids* of departments *i* and *j*.

The error incurred in estimating the layout cost as described above depends on the relative size of the two departments being exchanged. If the departments differ in size, then the estimated centroids may deviate significantly from their correct locations. (Of course, if the departments are equal in area, no error will be incurred.) As a result, the actual reduction in the layout cost may be overestimated or underestimated. Although it does not fully address the above error, once the best exchange *based on the estimated layout cost* is identified, CRAFT exchanges the locations of the departments and computes their new centroids (and the actual layout cost) before continuing to the next iteration.

A more important refinement we need to make in the above description of CRAFT is concerned with the exchange procedure. Although at first it may seem "too detailed," this refinement is important from a conceptual point of view since it demonstrates an intricate aspect of using computers for layout purposes. When CRAFT considers exchanging two departments, instead of examining all possible exchanges as we stated above, it actually considers exchanging only those departments that are either adjacent (i.e., that share a border) or that are equal in area. Such a restriction is not arbitrary. Given that departments cannot be split, it would be impossible to exchange two departments without "shifting" the location of the other departments in the layout, *unless the two departments are either adjacent or equal in area.* (Why?) Since CRAFT is not capable of "automatically" shifting departments in such a manner, it considers exchanging only those that are adjacent or equal in area.

Obviously, two departments with equal areas, whether they are adjacent or not, can always be exchanged without shifting the other departments in the layout. However, if two departments are not equal in area, then adjacency is a *necessary but not sufficient* condition for being able to exchange them without shifting the other departments. That is, in certain cases, even if two (unequal-area) departments are adjacent, it may not be possible to exchange them without shifting the other departments. We will later present an example for such a case.

We also need to stress that, while searching for a better solution, CRAFT picks only the best (estimated) exchange at each iteration, which makes it a "steepest descent" procedure. It also does not "look back" or "look forward" during the above search. Therefore, CRAFT will terminate at the first two-opt or three-opt solution that it encounters during the search. Such a solution is very likely to be only locally optimal. Furthermore, with such a search procedure, the termination point (or the final layout) will be strongly influenced by the starting point (or the initial layout). Consequently, CRAFT is a highly "path-dependent" heuristic and to use it effectively, we generally recommend trying different initial solutions (if possible) or trying different exchange options (two-way vs. three-way) at each iteration.

CRAFT is generally flexible with respect to department shapes. As long as the department is not split, it can accommodate virtually any department shape. Theoretically, due to the centroid-to-centroid distance measure, the optimum layout (which has an objective function value of zero) consists of concentric rectangles! Of course, the above problem stems from the fact that the centroids of some O-shaped, U-shaped, and L-shaped departments may lie outside the department itself. Unless the initial layout contains concentric departments, CRAFT will not construct such a layout. However, some department shapes may be irregular, and the objective function value may be underestimated due to the centroid-to-centroid measure.

CRAFT is normally restricted to rectangular buildings. However, through "dummy" departments, it can be used with nonrectangular buildings as well. Dummy departments have no flows or interaction with other departments but require a certain area specified by the layout planner. In general, dummy departments may be used to

1. Fill building irregularities.
2. Represent obstacles or unusable areas in the facility (such as stairways, elevators, plant services, and so on).
3. Represent extra space in the facility.
4. Aid in evaluating aisle locations in the final layout.

Note that, when a dummy department is used to represent an obstacle, its location must be fixed. Fortunately, CRAFT allows the user to fix the location of any department (dummy or otherwise). Such a feature is especially helpful in modeling obstacles, as well as in modeling departments such as receiving and shipping in an existing facility.

One of CRAFT's strengths is that it can capture the initial layout with reasonable accuracy. This strength stems primarily from CRAFT's ability to accommodate nonrectangular departments or obstacles located anywhere in a possibly nonrectangular building. However, in addition to being highly path dependent, one of CRAFT's weaknesses is that it will rarely generate department shapes that result in straight, uninterrupted aisles as is desired in the final layout. Fixing some departments to specific locations, and in some cases placing dummy departments in the layout to represent main aisles, may lead to more reasonable department shapes. Nevertheless, as is the case with virtually all computerized layout algorithms, the final layout generated by the computer should not be presented to the decision maker before the layout planner "molds" or "massages" it into a practical layout.

As we demonstrate in the following example, molding or massaging a layout involves relatively minor adjustments being made to the department shapes and/or department areas in the final layout. The fact that such adjustments are almost always necessary does not imply that computerized layout algorithms are of limited use. To the contrary, by considering a large number of alternatives in a very short time, a computerized layout algorithm narrows down the solution space for the layout planner, who can then concentrate on further evaluating and massaging a few "promising" solutions identified by the computer.