

Modul

Layout and Design

Prof. Dr. Richard F. Hartl
Dr. Margaretha Preusser

SS 2009



© Produktion und Logistik

Inhaltsverzeichnis

1. Introduction	4
2. Job shop production	9
2.1. The Linear Assignment Problem	10
2.1.1. Formulation as transportation problem	11
2.1.2. Assignment Method (Kuhn's Algorithm)	13
2.2. The Quadratic Assignment Problem (QAP)	18
2.2.1. QAP: Mathematical formulation	20
2.2.2. Starting heuristics	22
2.2.3. Improvement methods	23
2.2.4. „Umlaufmethode“	24
1. Appendix A: Methodological Basics	34
Complexity	34
Costs and distances	35
Basics on Graph Theory	35

1. Introduction¹²

Layout decisions are one of the key facts determining the long-run efficiency of operations. Layouts have numerous strategic implications because they establish an organization's competitive priorities in regard to capacity, processes, flexibility, and cost. They are associated with the tactical decision horizon and are dedicated to the concretion of strategic decisions like, e.g., facility location. Configured production systems are input for the operational level, where the goal is to run the given system as efficiently as possible.

An efficient layout facilitates and reduces costs of material flow, people, and information between areas. To achieve these objectives, a variety of configuration designs have been developed. The most relevant ones, in the context of this course, are:

1. **Fixed-position layout**: addresses the layout requirements of large, bulky projects
2. **Job shop production** (Process-oriented layout): deals with low-volume, high-variety production
3. **Cellular manufacturing** systems (work cell layout): arranges machinery and equipment to focus on production of a single product or group of related products
4. **Flow shop production** (Product-oriented layout): seeks the best personnel and machine utilization in repetitive or continuous production.

As a matter of fact layouts 1 and 2 are often described as centralized, and layouts 3 and 4 as decentralized manufacturing systems.

Example²: To illustrate the differences in fixed-position layout, job shop production, cellular manufacturing systems, and flow shop production consider a situation in which four parts (A, B, C, D) are to be produced and assembled into a single product. The processing sequence for part A is saw, turn, mill, and drill; for part B it is saw, mill, drill, and paint; for part C the processing sequence is grind, mill, drill, and paint; and for part D the sequence is weld, grind, turn, and drill. All parts go to a central assembly department. The following table contains the proportional capacity requirement of each part on each machine relative to the capacity availability of the machine in one time period.

Part	Equipment requirements						
	Weld	Grind	Saw	Turn	Mill	Drill	Paint
A	-	-	0.5	0.5	0.3	0.2	-
B	-	-	0.4	-	0.5	0.3	0.2
C	-	0.4	-	-	0.3	0.5	0.3
D	0.3	0.5	-	0.3	-	0.2	-

¹ Heizer, J., Render, B., Operations Management, Prentice Hall, 2006, Chapter 9

² Francis, R., McGinnis, L., White, J., Facility Layout and Location: An Analytical Approach, Prentice Hall, 1992

Based on the given capacity requirements we know that the minimum equipment needed is: 1 weld, 1 grind, 1 saw, 1 turning machine, 2 mills (0.3+0.5+0.3 > 1), 2 drills, and 1 painting machine.

According to the layout concepts listed above the following configurations for the example problem could be realized (this is not a complete list of all possible configurations but an illustrative selection of possible realizations).

1. In case of a fixed-position layout it may be sufficient to have the minimum machine equipment (see above). But depending on how production is scheduled it could also be necessary to install more machines in order to come up with the needed production output.

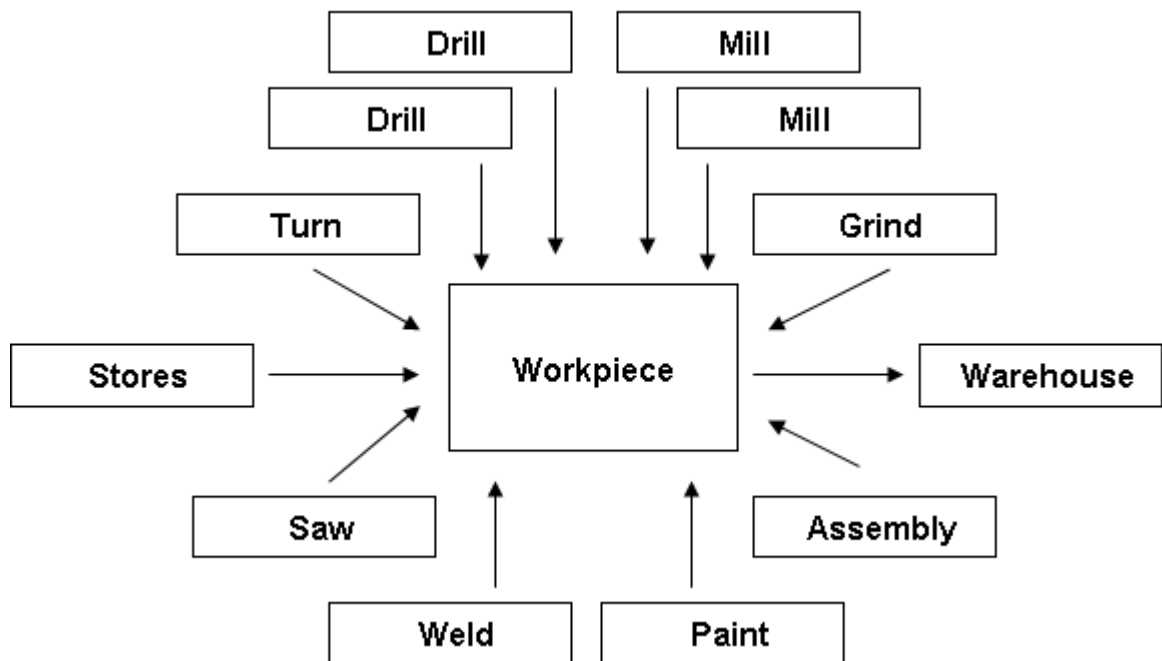


Figure 1-1: Fixed-position layout

2. By applying a job shop production system we are able to reach the minimum machine equipment. Clearly, depending on production scheduling it may become necessary to install more machines than the minimum equipment.

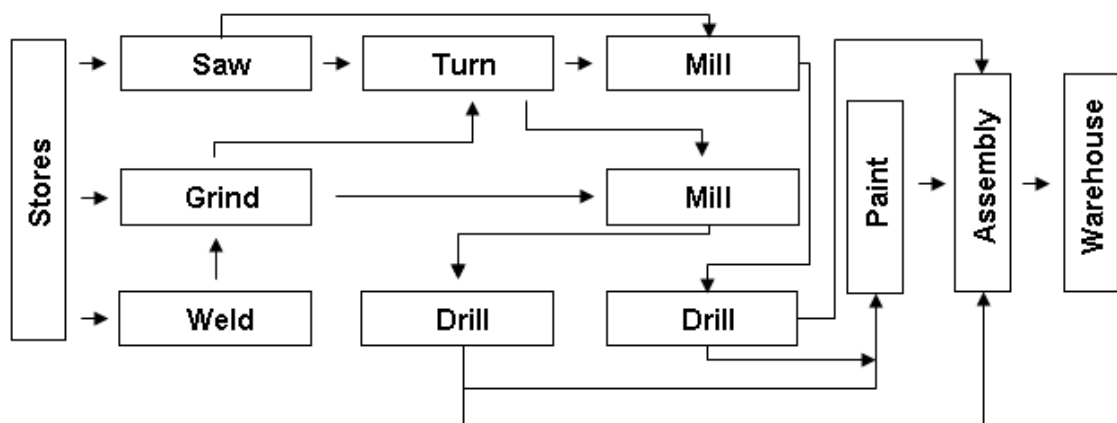


Figure 1-2: Job shop production

3. Figure 1-3 illustrates a cellular manufacturing system for the example problem. For the chosen configuration (2 work cells) it is not possible to realize the minimum machine equipment. We need an additional turning machine and an additional painter.

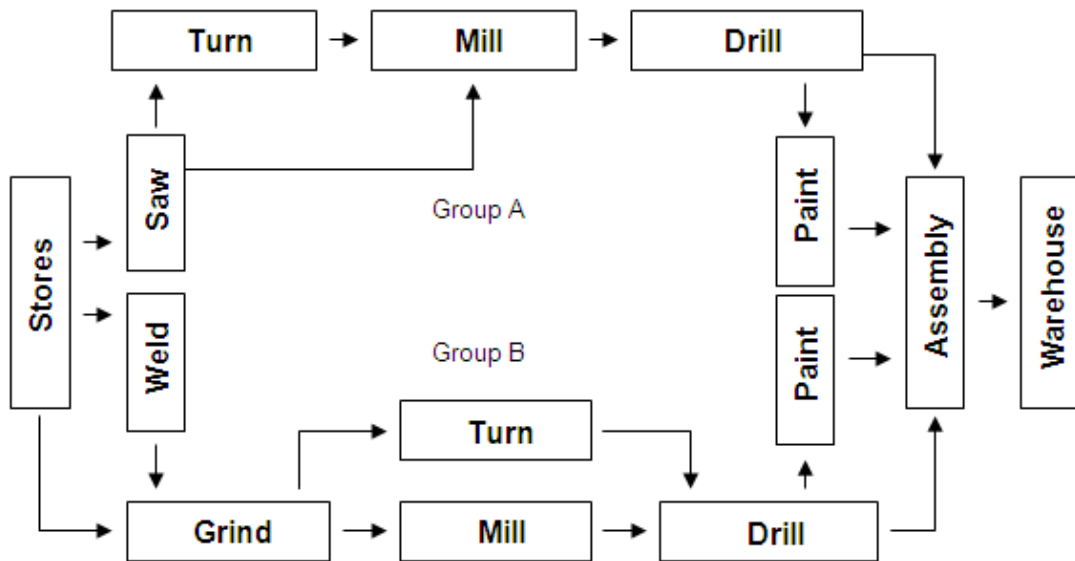


Figure 1-3: Cellular manufacturing system

4. Figure 1-4 shows a flow shop production system for the example problem. In this case we need 5 machines additional to the minimum equipment (1 grind, 1 saw, 1 turning machine, 1 mill, and 1 paint):

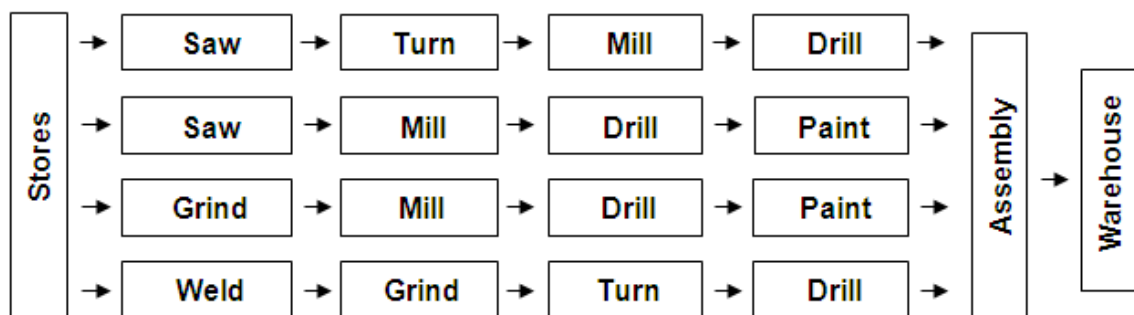


Figure 1-4: Flow shop production

The decision to use a fixed-position layout is generally dictated by a particular characteristic of the workpiece. It layout is used when the product is too large or cumbersome to be moved through the various processing steps. Consequently, the processes are brought to the product rather than taking the product to the processes (e.g. aircraft industry,...).

This concept is realized by locating workstations or production centres around the product in the appropriate processing sequence. Considerable logistics are involved in ensuring that the right processes are brought to the product at the right times and are located in the right places.

Advantages:

- Material movement is reduced.
- Promotes job enlargement by allowing individuals or teams to perform the “whole job”.
- Highly flexible; can accommodate changes in product design, product mix, and production volume.
- Independence of production centres allows scheduling to achieve minimum total production time.

Limitations:

- Increased movement of personnel and equipment.
- Equipment duplication may occur.
- Higher skill requirements for personnel.
- General supervision required.
- Cumbersome and costly positioning of material and machinery.
- Low equipment utilization.

However, the decision to use either a job shop, work cell, or flow shop layout generally depends on the volumes of production and variety of products being manufactured. Figure 1-5 illustrates a volume-variety chart³.

³ Francis, R., McGinnis, L., White, J., Facility Layout and Location: An Analytical Approach, Prentice Hall, 1992

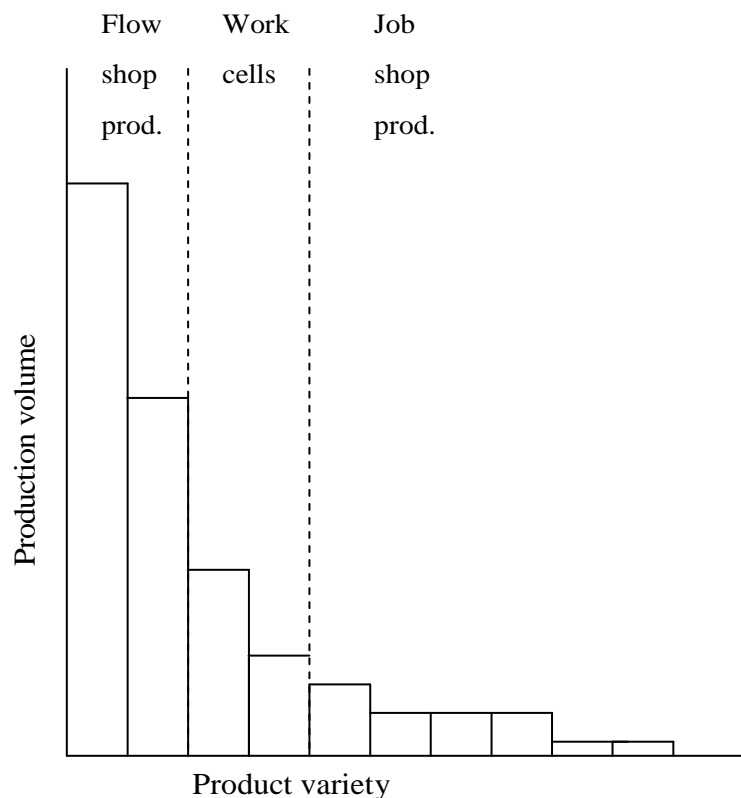


Figure 1-5: Volume-variety chart

Flow shop production is appropriate for high-volume, low variety conditions. Working cell manufacturing systems are usually used for “in between” conditions, and job shop production is applied for low-volume high-variety settings. In fact, many real world layouts tend to be a combination of all three of them (hybrid layout). The volume-variety mix among products can be such that a few products are manufactured using flow shop production, others using job shop production, and the remainder using working cell manufacturing. Similarly, it may be useful to appropriate to use either job shop production or working cells for the production of individual components and to use a flow shop system for the assembly of the components.

In the following we are going to discuss job shop production, cellular manufacturing systems and flow shop production in more detail. Occurring optimization problems and dedicated solution methods will be discussed as well.

2. Job shop production⁴⁵

The process-oriented layout can simultaneously handle a wide variety of products. It is typically the low-volume, high-variety strategy. Each product or product group undergoes a different sequence of operations. It is produced moving it from one department to another in the required sequence. Different products have different material flows. Thus, it is not efficient to arrange machines due to a product-oriented layout (flow shop system) but according to a process-oriented layout.

A process-oriented layout consists of a collection of processing departments or cells. All machines involved in performing a particular process are grouped together in a machine shop (e.g. drill, weld,..). This concept is used when there are many low-volume, dissimilar products. It is also used in case of rapid changes in product mix or volume, as well as when conditions are such that neither product-oriented nor cellular manufacturing systems are useful. In comparison with cellular manufacturing systems this layout concept is characterized by high degrees of interdepartmental flow. A big advantage of this process-oriented layout is its flexibility in equipment and labor assignment. The breakdown of one machine need not halt an entire process; work can be transferred to other machines in the department.

Advantages:

- Better utilization of machines can result; consequently, fewer machines are required.
- A high degree of flexibility exists relative to equipment or manpower allocation for specific tasks.
- Comparatively low investment in machines is required.
- The diversity of tasks offers a more interesting and satisfying occupation for the operator.
- Specialized supervision is possible.

Limitations:

- Since longer flow lines are needed, material handling is more expensive.
- Production planning and control systems are more involved than for other layouts.
- Usually, total production time is longer than for other layouts.

⁴ Heizer, J., Render, B., Operations Management, Prentice Hall, 2006, Chapter 9

⁵ Francis, R., McGinnis, L., White, J., Facility Layout and Location: An Analytical Approach, Prentice Hall, 1992

- Due to the fact that jobs have to queue before being processed in a machine job comparatively large amounts of in-process inventory occur.
- Comparatively high degree of (machine) idle time because machines have to wait until the subsequent job is finished with its foregoing process.
- Space and capital are tied up by work in process.
- Because of the diversity of the jobs in specialized departments, higher grades of skill are required.

Negative effects like idle or waiting times should be reduced by using dedicated production planning methods (on the operational level) and optimized machine shop arrangement on the tactical level. These tactical optimization problems referring to the optimal arrangement of machines in job shop production systems are known as “Assignment problems”. Although the typical problem in this context refers to the “Quadratic assignment problem” we first want to introduce the basic model: the “Linear assignment problem”.

2.1. The Linear Assignment Problem

The *Linear Assignment Problem* (LAP) is one of the most famous problems in linear programming and in combinatorial optimization. Apart from its application to intra-company location planning it can be used for a number of other planning problems.

Given

- n machines (jobs, workers)
- n potential locations (periods, projects)
- c_{ij} ... cost of running machine i on position j .

Any machine can be assigned to any location, incurring some cost that may vary depending on the machine-location assignment. It is required to use all locations by assigning exactly one machine to each location in such a way that the total costs of the assignments are minimized.

The LP is formulated as follows.

$x_{ij} = 1$, if machine i is assigned to location j and 0 otherwise

$$C = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min \quad \text{s.t.} \quad \sum_{j=1}^n x_{ij} = 1 \quad \text{for } i=1, \dots, n \quad \dots \text{assign all machines}$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \text{for } j=1, \dots, n \quad \dots \text{1 machine at each location}$$

$$x_{ij} \geq 0 \quad \text{for } i=1, \dots, n \quad \text{and } j=1, \dots, n$$

Example 1: 3 machines, 4 locations and the following costs c_{ij} (machine 2 may not be assigned to location 2 - > cost ∞):

		location $j =$			
		1	2	3	4
$i =$	maschine 1	13	10	12	11
	2	15	∞	13	20
	3	5	7	10	6

Convert the given problem into a symmetric one by adding dummy-machines (-rows) or dummy-locations (-columns) with cost 0:

		location $j =$			
		1	2	3	4
$i =$	maschine 1	13	10	12	11
	2	15	∞	13	20
	3	5	7	10	6
<i>dummy</i>	4	0	0	0	0

Locations being assigned to a dummy-machine remain empty. A machine being assigned to a dummy-location means that this machine is physically not allocated to any of the potential locations.

2.1.1. Formulation as transportation problem

Linear assignment problems can be interpreted as special case of a general *transportation problem* (TP). The latter is formulated as follows:

m supplier with supply $s_i, i = 1, \dots, m$

n consumer with demand $d_j, j = 1, \dots, n$

transportation costs c_{ij} per unit transported from i to j

decision variables x_{ij} indicate the amount of units transported from i to j

$$\text{Transportation cost } K = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$$

$$\text{Supply } s_i = \sum_{j=1}^n x_{ij} \quad i = 1, \dots, m$$

$$\text{Demand } d_j = \sum_{i=1}^m x_{ij} \quad j = 1, \dots, n.$$

$$\text{Nonnegativity } x_{ij} \geq 0 \quad i = 1, \dots, m; j = 1, \dots, n.$$

In order to derive the LAP from the TP we interpret machines as suppliers with a capacity of 1 and each location as consumer with a demand of 1. Thus, each LAP can be solved as special case of transportation problems. It is known that due to the problem structure the optimal solution of a TP consists of $(m+n-1)$ integer basis variables (even if we do not explicitly constrain integer property!). Clearly, this problem characteristic is valid for the LAP (as it has been derived as special case of the TP) as well. Since for the LAP we assume “supply” (machines) = “demand” (locations) = 1 it is automatically guaranteed to obtain an optimal solution consisting of exactly n decision variables with value 1 while all other variables are 0 (although the formulation basically allows non-integer variables as well). Thus, we obtain a feasible solution for the LAP. Clearly, finding a feasible solution premises to have an equal number of machines and locations ($m=n$), which has already been mentioned above and is part of the mathematical LAP formulation as well.

Example 1: as TP:

$i \setminus j$	1	2	3	4	s_i
1	13	10	12	11	1
2	15	∞	13	20	1
3	5	7	10	6	1
4	0	0	0	0	1
d_j	1	1	1	1	

There is another problem characteristic which we are going to make use of in order to solve the LAP: it is always possible to reduce (or increase) all entries of any column or row by a certain value without changing the optimal solution (only the absolute costs change, the relation stays the same). We use this characteristic in order to generate the maximum amount of 0 entries in the cost matrix. By subtracting the smallest element of each column and row from all elements of this column/row we generate the maximum number of 0 entries while not having any influence on the optimal solution. Clearly, the absolute cost factors do change by this matrix reduction, but the relation of assignment costs for each machine/location definitely stays the same.

Example 2: Cost reduction

Cost matrix:

	I	II	III
A	1	8	15
B	6	2	10
C	7	9	3

In this case the column minimum method finds the optimal solution. Machines A, B, and C are assigned to locations I, II, and III, respectively. We end up with total assignment costs of 6 (=1+2+3).

Cost reduction:

	I	II	III
A	0	6	12
B	5	0	7
C	6	7	0
	-1	-2	-3

Again the column minimum method leads to the following assignment: A-I, B-II, C-III. This solution has total (reduced) assignment costs of 0 which implies that we found an optimal solution. By adding the sum of reduction values to the reduced assignment costs we again determine the total assignment costs: $0+1+2+3=6$.

Usually, especially when solving larger problems, it is necessary to apply some iterations of the Transport-Simplex-Method in order to find the optimal solution.

However, for the LAP we know specialized methods leading to the optimal solution more quickly (cf. Appendix A referring to problem complexity). The most famous one will be presented in the following.

2.1.2. Assignment Method (Kuhn's Algorithm)⁶

Kuhn's Algorithm involves adding/subtracting appropriate values to/from the given cost factors in order to find the lowest opportunity cost (foregone or not-obtained profits) for each assignment.

“There are 3 steps to be followed:

1. *Subtract the smallest number in each column from every number in that column and then, from the resulting matrix, subtract the smallest number in each row from every number in that row. This step has the effect of reducing the numbers in the table until a series of zeros (at least 1 per column and row), meaning zero opportunity costs, appear.”⁵*
2. Draw the minimum number of vertical and horizontal straight lines necessary to cover all zeros in the table. This minimum number of lines equals the maximum number of zero cost assignments. Thus, if the number of lines equals the number of rows/columns in the

⁶ Heizer, J., Render, B., Operations Management, Prentice Hall, 2006, Chapter 15

table, then we can make an optimal assignment. If the number of lines is less than the number of rows or columns, we proceed to step 3.

At this point we want to complete this step of the Assignment Method by specifying the procedure during step 2. In fact, finding the minimum number of vertical and horizontal lines necessary to cover all zeros in a matrix may be trivial in case of very small matrices, but should be solved systematically in case of larger ones. Thus, we want to introduce the following procedure in addition to step 2:

We proceed systematically by choosing a column or row with as few as possible zero entries (preferably exactly one 0) and framing (shading) a 0 in this column or row. This leads to an interim assignment.

Then we cross all remaining zeros in this column or row. Now in each column or row related to a framed 0 all other zeros are crossed which means that in this column or row no further assignments are possible.

Now the next column or row with as few as possible non-marked (not crossed and not framed) zeros is chosen and so on. We stop as soon as we do not have zeros left to be framed. Now we have an arrangement of marked columns and rows including all zeros.

If we are able to make an assignment with (reduced) costs of 0 for each machine we have found an optimal assignment otherwise we proceed as follows:

- 2.1. Mark (for example „X“) all rows with no framed 0
 - 2.2. Mark all columns having at least 1 crossed 0 in a marked row
 - 2.3. Mark all rows having a framed 0 in a marked column
 - 2.4. Repeat 2.2 and 2.3 until there is no column or row left to be marked
 - 2.5. Mark each non-marked row and each marked column (shaded) with a continuous line -> all framed zeros are crossed now and we have the minimum number of crossed -> lines and rows needed to cover all zeros, i.e. the maximum number of zero cost assignments. If this number equals the number of rows or columns an optimal assignment is already found (in this case it would not have been necessary to perform the given subprocedure (2.1.-2.6.) because we should already have succeeded in finding a zero cost assignment as described above).
3. *“Subtract the smallest number not covered by a line from every other uncovered number. Add the same number to any number(s) lying at the intersection of any two lines. Do not change the value of the numbers that are covered by only one line. Return to step 2 and continue until an optimal assignment is possible.*

Some assignment problems entail maximizing profit, effectiveness, or payoff of an assignment of people to tasks or of jobs to machines. It is easy to obtain an equivalent minimization problem by converting every number in the matrix to an opportunity loss. To convert a maximization problem to an equivalent minimization problem, we subtract every number in the original matrix from the largest single number in that matrix. We then proceed to step 1. It turns out that

minimizing the opportunity loss produces the same assignment solution as the original maximization problem.”⁷

Example 1:

$i \setminus j$	1	2	3	4	s_i	
1	13	10	12	11	1	-10
2	15	∞	13	20	1	-13
3	5	7	10	6	1	-5
4	0	0	0	0	1	
d_j	1	1	1	1		

Step 1: Reduced costs

$i \setminus j$	1	2	3	4	s_i
1	3	0	2	1	1
2	2	∞	0	7	1
3	0	2	5	1	1
4	0	0	0	0	1
d_j	1	1	1	1	

Step 2: Optimal solution?

$i \setminus j$	1	2	3	4	s_i
1	3	0	2	1	1
2	2	∞	0	7	1
3	0	2	5	1	1
4	0	0	0	0	1
d_j	1	1	1	1	

In this case we see at first glance that an optimal solution is obtained with the following assignment: 1-2 2-3 3-1 4-4 (we have a zero cost assignment for each machine; the minimum number of lines needed to cover all zero elements would be equal to the number of rows/columns).

Example 2:

⁷ Heizer, J., Render, B., Operations Management, Prentice Hall, 2006, Chapter 15

Step 1: Cost reduction

17,5	15	9	5,5	12
16	16,5	10,5	5	10,5
12	15,5	14,5	11	5,5
4,5	8	14	17,5	13
13	9,5	8,5	12	17,5

-4,5 -8 -8,5 -5 -5,5

⇒

-0,5

⇒

Step 2: Optimal solution?

12,5	6,5	0	0	6
11,5	8,5	2	0	5
7,5	7,5	6	6	0
0	0	5,5	12,5	7,5
8,5	1,5	0	7	12

We are not able to make an assignment with (reduced) costs of 0 for each machine. Thus, we proceed with finding the minimum arrangement of marked columns and rows including all 0 elements.

12,5	6,5	0	0	6
11,5	8,5	2	0	5
7,5	7,5	6	6	0
0	0	5,5	12,5	7,5
8,5	1,5	0	7	12

12,5	6,5	0	0	6	X (1c)
11,5	8,5	2	0	5	X (2c)
7,5	7,5	6	6	0	
0	0	5,5	12,5	7,5	
8,5	1,5	0	7	12	X (1a)
		X (1b)	X (2b)		

Step 3: Generation of additional zeros.

From all covered elements we choose the smallest. This element a is going to be subtracted from all not covered elements and is going to be added to all elements being covered twice.

11	5	0	0	4,5	
10	7	2	0	3,5	
7,5	7,5	7,5	7,5	0	
0	0	7	14	7,5	
7	0	0	7	10,5	1 additional zero (assignment 5 → 2) increases the chance the find an assignment with total (reduced) costs of 0.

Step 2: Optimal solution?

Again we have to find out if we already are able to determine the optimal assignment.

Iteration 2:

11	5	0	0	4,5
10	7	2	0	3,5
7,5	7,5	7,5	7,5	0
0	0	7	14	7,5
7	0	0	7	10,5

We have found the optimal solution with reduced assignment costs of 0.

The total costs are calculated by summing up all reduction values (clearly, element "a" determined in step 3 is a reduction value as well):

$$K = (4,5 + 8 + 8,5 + 5 + 5,5 + 0,5) + (1,5) = 33,5$$

2.2. The Quadratic Assignment Problem (QAP)

The more common mathematical formulation for intra-company location problems (especially in case of job shop production) is the *Quadratic Assignment Problem* (QAP). For the QAP the cost of an assignment is determined by the distances and the material flows between all given entities. While, in case of LAP the costs for assigning a machine to a location do not depend on the location chosen for any other machine we now want to take distances of locations and material flow between entities into account as well. In fact, we are now going to minimize the total *transportation* costs occurring due to the chosen assignment whereas for the LAP we minimize isolated location-oriented costs.

So called “*Activity Relationship Charts*” are useful graphical means of representing the desirability of locating pairs of machines/operations near to each other. The following letter codes have been suggested in literature for determining a “closeness” rating:⁸

- “A *Absolutely necessary. Because two machines/operations use the same equipment or facilities, they must be located near each other.*
- E *Especially important. The facilities may for example require the same personnel or records.*
- I *Important. The activities may be located in sequence in the normal work flow.*
- O *Ordinary importance. It would be convenient to have the facilities near each other, but it is not essential.*
- U *Unimportant. It does not matter whether the facilities are located near each other or not.*
- X *Undesirable. Locating a wedding department near one that uses flammable liquids would be an example of this category.”⁷*

Example⁸: “*Met Me, Inc., is a franchised chain of fast-food hamburger restaurants. A new restaurant is being located in a growing suburban community near Reston, Virginia. Each restaurant has the following departments:*

1. *Cooking burgers*
2. *Cooking fries*
3. *Packing and storing burgers*
4. *Drink dispensers*
5. *Counter servers*
6. *Drive-up server*

⁸ Nahmias, S.: Production and Operations Analysis, 4th ed., McGraw-Hill, 2000, Chapter 10

The burgers are cooked on a large grill, and the fries are deep fried in hot oil. For safety reasons the company requires that these cooking areas not be located near each other. All hamburgers are individually wrapped after cooking and stored near the counter. The service counter can accommodate six servers, and the site has an area reserved for a drive-up window.

An activity relationship chart for this facility appears in the following. In the chart, each pair of activities is given one of the letter designations A, E, I, O, U, or X. Once a final layout is determined, the proximity of the various departments can be compared to the closeness ratings in the chart. Figure 2-2 illustrates the activity relationship chart for Met me Inc .”⁸

In the original conception of the QAP a number giving the reason for each closeness rating is needed as well. In case of closeness rating “X” a negative value would be used to indicate the undesirability of closeness for the according machines/operations.

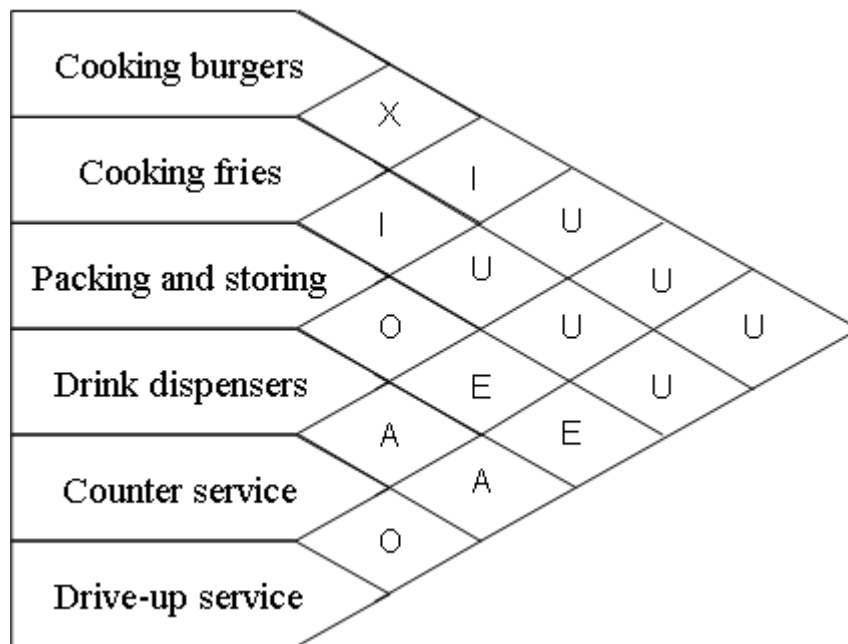


Figure 2-1: Activity relationship chart

2.2.1. QAP: Mathematical formulation

For the model formulation we need both distances between the locations and material flow between *organizational entities* (OE):

- n organizational entities (OE): all of them are of same size and can therefore be interchanged with each other.
- n locations: each can be provided with exactly one OE.
- t_{hi} ... intensity of material flow from OE h to OE i
- d_{jk} ... distance between location j and location k (e.g. shortest distance of central points); distances are not necessarily symmetric. Transportation costs are proportional to amount transported *and* to distance.

If OE h is allocated to location j and OE i to location k the transportation costs per unit from OE h to OE i are defined by d_{jk} . Similar to the LAP we define

$$\text{binary decision variable } x_{hj} = \begin{cases} 1 & \text{if OE } h \text{ is assigned to location } j \\ 0 & \text{otherwise} \end{cases}$$

Transportation costs per unit from OE h to OE i are $\sum_{j=1}^n \sum_{k=1}^n d_{jk} x_{hj} x_{ik}$.

The objective function minimizes total transportation costs between all OE:

$$\sum_{h=1}^n \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n t_{hi} d_{jk} x_{hj} x_{ik} \rightarrow \min$$

where we refer to the following constraints (similar to the LAP):

$$\sum_{j=1}^n x_{hj} = 1 \quad \text{for } h = 1, \dots, n \quad \dots \text{ each OE } h \text{ on exactly 1 location } j$$

$$\sum_{h=1}^n x_{hj} = 1 \quad \text{for } j = 1, \dots, n \quad \dots \text{ each location } j \text{ gets exactly 1 OE } h$$

$$x_{hj} = 0 \text{ or } 1 \quad \dots \text{ binary decision variable}$$

While all constraints are still linear we now face a non-linear objective function. Due to the combination of integer property and non-linearity finding optimal solutions for larger problems is almost impossible (cf. Appendix A). Thus, heuristic methods are applied in most cases. As usual we distinguish between starting heuristics and improvement methods or a combination of both of them.

Example: Calculation of costs of 3 OE (1, 2, 3) and 3 locations (A, B, C)

	Distances between locations d_{jk}	Intensity of material flow t_{hi}																																
	<table style="border-collapse: collapse;"> <tr> <td style="border: none;"></td> <td style="border: none; text-align: center;">A</td> <td style="border: none; text-align: center;">B</td> <td style="border: none; text-align: center;">C</td> </tr> <tr> <td style="border: none; text-align: center;">A</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">1</td> <td style="border: 1px solid black; text-align: center;">2</td> </tr> <tr> <td style="border: none; text-align: center;">B</td> <td style="border: 1px solid black; text-align: center;">1</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">1</td> </tr> <tr> <td style="border: none; text-align: center;">C</td> <td style="border: 1px solid black; text-align: center;">3</td> <td style="border: 1px solid black; text-align: center;">1</td> <td style="border: 1px solid black; text-align: center;">0</td> </tr> </table>		A	B	C	A	0	1	2	B	1	0	1	C	3	1	0	<table style="border-collapse: collapse;"> <tr> <td style="border: none;"></td> <td style="border: none; text-align: center;">1</td> <td style="border: none; text-align: center;">2</td> <td style="border: none; text-align: center;">3</td> </tr> <tr> <td style="border: none; text-align: center;">1</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">1</td> <td style="border: 1px solid black; text-align: center;">1</td> </tr> <tr> <td style="border: none; text-align: center;">2</td> <td style="border: 1px solid black; text-align: center;">2</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">2</td> </tr> <tr> <td style="border: none; text-align: center;">3</td> <td style="border: 1px solid black; text-align: center;">3</td> <td style="border: 1px solid black; text-align: center;">1</td> <td style="border: 1px solid black; text-align: center;">0</td> </tr> </table>		1	2	3	1	0	1	1	2	2	0	2	3	3	1	0
	A	B	C																															
A	0	1	2																															
B	1	0	1																															
C	3	1	0																															
	1	2	3																															
1	0	1	1																															
2	2	0	2																															
3	3	1	0																															

One possible solution would be 1 → A, 2 → B and 3 → C, i.e. $x_{1A} = 1, x_{2B} = 1, x_{3C} = 1$ and all other $t_{hj} = 0$. All constraints are fulfilled.
 Total transportation cost: $0*0 + 1*1 + 2*1 + 1*2 + 0*0 + 1*2 + 3*3 + 1*1 + 0*0 = 17$

Obviously, this solution is not optimal since OE 1 and 3 (which have the highest intensity of material flow) are assigned to the locations with the highest distance between them (A and C).

A better solution would for example be 1 → C, 2 → A and 3 → B, i.e. $x_{1C} = 1, x_{2A} = 1, x_{3B} = 1$.
 Total transportation cost: $0*0 + 3*1 + 1*1 + 2*2 + 0*0 + 2*1 + 1*3 + 1*1 + 0*0 = 14$

	Distances	Intensity of material flow																																
	<table style="border-collapse: collapse;"> <tr> <td style="border: none;"></td> <td style="border: none; text-align: center;">C</td> <td style="border: none; text-align: center;">A</td> <td style="border: none; text-align: center;">B</td> </tr> <tr> <td style="border: none; text-align: center;">2→A</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">3</td> <td style="border: 1px solid black; text-align: center;">1</td> </tr> <tr> <td style="border: none; text-align: center;">3→B</td> <td style="border: 1px solid black; text-align: center;">2</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">1</td> </tr> <tr> <td style="border: none; text-align: center;">1→C</td> <td style="border: 1px solid black; text-align: center;">1</td> <td style="border: 1px solid black; text-align: center;">1</td> <td style="border: 1px solid black; text-align: center;">0</td> </tr> </table>		C	A	B	2→A	0	3	1	3→B	2	0	1	1→C	1	1	0	<table style="border-collapse: collapse;"> <tr> <td style="border: none;"></td> <td style="border: none; text-align: center;">1</td> <td style="border: none; text-align: center;">2</td> <td style="border: none; text-align: center;">3</td> </tr> <tr> <td style="border: none; text-align: center;">1</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">1</td> <td style="border: 1px solid black; text-align: center;">1</td> </tr> <tr> <td style="border: none; text-align: center;">2</td> <td style="border: 1px solid black; text-align: center;">2</td> <td style="border: 1px solid black; text-align: center;">0</td> <td style="border: 1px solid black; text-align: center;">2</td> </tr> <tr> <td style="border: none; text-align: center;">3</td> <td style="border: 1px solid black; text-align: center;">3</td> <td style="border: 1px solid black; text-align: center;">1</td> <td style="border: 1px solid black; text-align: center;">0</td> </tr> </table>		1	2	3	1	0	1	1	2	2	0	2	3	3	1	0
	C	A	B																															
2→A	0	3	1																															
3→B	2	0	1																															
1→C	1	1	0																															
	1	2	3																															
1	0	1	1																															
2	2	0	2																															
3	3	1	0																															

2.2.2. Starting heuristics

Some starting heuristics refer to the combination of one of the following possibilities to select an OE and a location. The *core* is defined by the already chosen OE. After each iteration another OE is added to the core due to one of the following *priorities*:

- **Selection of (non-assigned) OE**

- A1. those having the maximum sum of material flow to all (other) OE
- A2. a) those having the maximum material flow to the last-assigned OE
b) those having the maximum material flow to an assigned OE
- A3. those having the maximum material flow to all assigned OE (core)
- A4. random choice

- **Selection of (non-assigned) locations**

- B1. those having the minimum total distance to all other locations
- B2. those being neighbouring to the last-chosen location
- B3. a) those leading to the minimum sum of transportation cost to the core
b) like a) but furthermore we try to exchange the location with neighboured OE
c) a location (empty or allocated) such that the sum of transportation costs within the new core is minimized (in case an allocated location is selected, the displaced OE is assigned to an empty location)
- B4. random choice

Example: By combining the simple rules A1 and B1 we have to arrange all OE referring to decreasing sum of material flow and all locations referring to increasing sum of distances to all other OE/locations (i.e. the last columns in the following tables):

OE	1	2	3	4	5	6	7	8	9	Σ	St.	A	B	C	D	E	F	G	H	I	Σ
1	-	-	-	-	3	-	-	-	-		A	-	1	2	1	2	3	2	3	4	
2		-	3	1	2	-	4	-	-		B		-	1	2	1	2	3	2	3	
3			-	3	5	2	-	3	4		C			-	3	2	1	4	3	2	
4				-	-	-	1	-	-		D				-	1	2	1	2	3	
5					-	2	2	1	-		E					-	1	2	1	2	
6						-	-	-	-		F						-	3	2	1	
7							-	-	-		G							-	1	2	
8								-	-		H								-	1	
9									-		I									-	

This leads to the following sequence of OE and locations: 3, 5, 2, 7, 4, 6, 8, 9, 1 ; E, B, D, F, H, A, C, G, I.

We obtain the following assignment of OE to locations:

OE	1	2	3	4	5	6	7	8	9
St.	I	D	E	H	B	A	F	C	G

OE	1	2	3	4	5	6	7	8	9
1	-	-	-	-	3×3	-	-	-	-
2		-	3×1	1×2	2×2	-	4×2	-	-
3			-	3×1	5×1	2×2	-	3×2	4×2
4				-	-	-	1×2	-	-
5					-	2×1	2×2	1×1	-
6						-	-	-	-
7							-	-	-
8								-	-
9									-

Calculation of costs:

1 and 5 are assigned to I and B with material flow of 3 and

3 (distance 1-5) × 3 (flow I-B)

and so on

Total cost = 61

In case of random choice (rules A4 and/or B4) one has the possibility to generate a set of different solutions and to choose the best one out of it.

2.2.3. Improvement methods

We basically try to improve solutions (i.e. reduce costs) by exchanging OE-pairs (see also the introductory example above). In case of acceptable computational times one can also try to exchange OE-triples. Even in case of pair wise exchanges we have different possibilities:

- **Selection of potential pairwise exchanges:**

- C1. all $n(n - 1)/2$ pairs
- C2. a subset of pairs
- C3. random choice

- **Selection of pairwise exchanges that are finally performed:**

- D1. that pair whose exchange of locations leads to the highest cost reduction. (best pair)
- D2. the first pair whose exchange of locations leads to a cost reduction (first pair)

A combination of C1 and D1 increases solution quality but also computational time. A common method is to start with C2 and skip to C1 as soon as the solution is reasonably good. (A combination of C1 and D2 is equivalent to the 2-opt method which we use to solve TSP).

A well-known (heuristic) method is *CRAFT* (computerized relative allocation of facilities techniques) which equals (in case of OE with similar place requirements) a combination of C1 and D1 (this method will be introduced later in this chapter in the context of OE with unequal place requirements).

In case of random choice (C3 and D2) we quite often find good results. Especially the fact that sometimes the best exchange of all exchanges which have been checked leads to an increase of costs is no disadvantage, because it reduces the risk to be trapped in local optima.

The basic idea and several adaptations/combinations of A, B, C, and D are discussed in literature.

2.2.4. „Umlaufmethode“

„Umlaufmethode“ is one of the numerous heuristics which combine the idea of starting heuristics and improvement methods. This method consists of the following components:

Initialization ($i = 1$):

Those OE having the maximum sum of material flow [A1] is assigned to the centre of locations (i.e. the location having the minimum sum of distances to all other locations [B1]).

Iteration i ($i = 2, \dots, n$): assign OE i

Part 1: (*Selection of OE and of free location*):

- select those OE with the maximum sum of material flow to all OE assigned to the core [A3]
- assign the selected OE to a free location so that the sum of transportation costs to the core (within the core) is minimized [B3a]

Part 2: (*Improvement step in iteration $i = 4$*):

- check pair wise exchanges of the last-assigned OE with all other OE in the core [C2]
- if an improvement is found, the exchange is conducted and we start again with Part 2 [D2].

The method ends with the finalization of iteration $i = n$ having assigned all OE.

Example, at first without improvement step (only Part 1):

Initialization ($i = 1$):

E is the centre

We assign OE 3 to the centre (E)

A	B	C
D	E 3	F
G	H	I

Sequence of assignments:

$i =$	1	2	3	4	5	6	7	8
OE	3							
1	0							
2	3							
3	↑							
4	3							
5	5							
6	2							
7	0							
8	3							
9	4							

$i = 3$: 2 has the max. mat.fl. to the core (3, 5)

$i = 1$: at first we assign 3

$i = 5$

$i = 2$: 5 has the max. mat.fl. to 3

$i = 6$

$i = 4$

$i = 7$

$i = 8$

Iteration $i = 2$ (Part 1): the maximum material flow to the core (3) is from OE 5.

Distances $d_{BE} = d_{DE} = d_{FE} = d_{HE} = 1$ is equally minimal \Rightarrow we select D

\Rightarrow In step $i = 2$ we assign D-5.

Iteration $i = 3$ (Part 1): the maximum material flow to the core (3, 5) is from OE 2.

Find a location X so that $d_{XE} \cdot t_{23} + d_{XD} \cdot t_{25} = d_{XE} \cdot 3 + d_{XD} \cdot 2$ is minimal (only A, B, G or H)

$$X = A \quad d_{AE} \cdot 3 \quad + \quad d_{AD} \cdot 2 \quad = \quad 2 \cdot 3 \quad + \quad 1 \cdot 2 \quad = \quad 8$$

$$X = B \quad d_{BE} \cdot 3 \quad + \quad d_{BD} \cdot 2 \quad = \quad 1 \cdot 3 \quad + \quad 2 \cdot 2 \quad = \quad 7$$

$$X = F \quad d_{FE} \cdot 3 \quad + \quad d_{FD} \cdot 2 \quad = \quad 1 \cdot 3 \quad + \quad 2 \cdot 2 \quad = \quad 7$$

$$X = G \quad d_{GE} \cdot 3 \quad + \quad d_{GD} \cdot 2 \quad = \quad 2 \cdot 3 \quad + \quad 1 \cdot 2 \quad = \quad 8$$

$$X = H \quad d_{HE} \cdot 3 + d_{HD} \cdot 2 = 1 \cdot 3 + 2 \cdot 2 = 7 \quad \text{B, F or H} \Rightarrow \text{B is chosen}$$

\Rightarrow In Step $i = 3$ we assign B-2.

Iteration $i = 4$ (Part 1) the maximum material flow to the core (2, 3, 5) is from OE 7.

Find a location X so that $d_{XE} \cdot t_{73} + d_{XD} \cdot t_{75} + d_{XB} \cdot t_{72} = d_{XE} \cdot 0 + d_{XD} \cdot 2 + d_{XB} \cdot 4$ is minimal in the map we see that A is the best choice

\Rightarrow in iteration $i = 4$ we tentatively assign A-7.

(Part 2) try to exchange A with E, B or D and calculate the costs of these assignments:

From Part 1: E-3, D-5, B-2, A-7	Cost	=	$1 \cdot 5 + 1 \cdot 3 + 2 \cdot 0 + 2 \cdot 2 + 1 \cdot 2 + 1 \cdot 4$	=	18
E-3, D-5, A-2 , B-7	Cost	=	$1 \cdot 5 + 2 \cdot 3 + 1 \cdot 0 + 1 \cdot 2 + 2 \cdot 2 + 1 \cdot 4$	=	21
E-3, A-5 , B-2, D-7	Cost	=	$2 \cdot 5 + 1 \cdot 3 + 1 \cdot 0 + 1 \cdot 2 + 1 \cdot 2 + 2 \cdot 4$	=	25
A-3 , D-5, B-2, E-7	Cost	=	$1 \cdot 5 + 1 \cdot 3 + 2 \cdot 0 + 2 \cdot 2 + 1 \cdot 2 + 1 \cdot 4$	=	18

An exchange of A and E is possible but does not lead to a cost reduction. Thus, we do not conduct this exchange but take the solution determined in Part 1.

After 8 iterations (without part 2) we end up with the solution from above with total costs = 54.

Inclusion of part 2 leads to an exchange of the last-assigned OE 9 with OE 4 in iteration 8. By this exchange we increase total costs to 51.

While a manual calculation of larger problems is obviously quite time consuming an implementation and therefore computerized calculation is relatively simple.

2.2.5. Different space requirements

The solution methods discussed above are also feasible for problems considering OE with different space requirements (OE are assumed to be either rectangularly shaped or composed of rectangular pieces). But here an exchange of OE may have an influence on the shapes and locations of other (even not-exchanged) OE. Furthermore, one has to define the way of measuring distances between locations, since the distance between two OE may depend on their shapes. The most common distances in this context probably are:

- Orthogonal distance between OE-boundaries: the (shortest) distance between 2 OE is determined by the orthogonal distance between the closest points of them. Thus, OE having at least 1 vertex in common have a distance of 0.
- Rectilinear distance of centre points: the distance between 2 OE is assumed to be the rectilinear distance between centroid locations. This implies the assumption that OE are located at their centroids. The centroid is another term for the coordinates of the centre of gravity or balance point. The accuracy of assuming that an OE is located at its centroid depends upon the shape of the OE. The assumption is most accurate when the shape of the OE is square or rectangular, but is less accurate for oddly shaped OE.

Rectilinear distances between centre points are, e.g., used for the well known CRAFT algorithm, which we are going to discuss in the following.

2.2.5.1. CRAFT Algorithm⁹¹⁰

CRAFT (computerized relative allocation of facilities techniques) was one of the first computer-aided layout routines developed. It is an improvement method which means that it requires an initial layout to be used as a starting solution.

Again we try to improve the given solution by moving around OE. The additional challenge now is that the shapes of OE are not fixed. Thus, the problem simply has too many degrees of freedom for us to devise a good method for modifying the starting solution. All the common improvement methods are based on limiting the kinds of changes that are permitted. This has already been addressed in the context of problems with similar space requirements.

We know that a pair of OE that can be exchanged without a direct influence on the shapes or locations of all remaining OE has to satisfy one of the following conditions:

1. the OE have the same space requirement,
2. the OE share a common boundary (having a common boundary means that the OE share at least 1 side boundary of their rectangles). E.g. OE 2 and OE 5 in Figure 2-2 share 1 side boundary.

In Figure 2-2 you see that an exchange of OE 2 and OE 5 would be possible without affecting the shape or the location of the remaining OE. In this case it does not matter that OE 2 and OE 5 have different space requirements because the total area used for them stays the same. Clearly, an exchange of same sized OE (no matter if they are neighbored or not) is always possible. On the other hand exchanging, e.g., OE 1 and OE 3, is not possible without changing the shape or the location of for example OE 2.

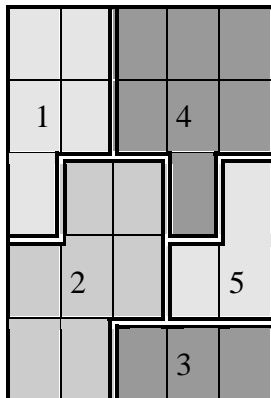


Figure 2-2: Different sized OE

⁹ Francis, R., McGinnis, L., White, J., Facility Layout and Location: An Analytical Approach, Prentice Hall, 1992

¹⁰ Nahmias, S.: Production and Operations Analysis, 4th ed., McGraw-Hill, 2000, Chapter 10

The CRAFT algorithm, which is probably the earliest widely known improvement algorithm, uses an estimate of the transportation cost that is based on the rectilinear distance between the centroid locations. If, e.g., OE 2 and 5 are considered for an exchange, the new costs is estimated by assuming that the new centroid of OE 2 is the old centroid of OE 5 and vice versa. This method of estimating transportation costs for the new layout is exact if OE have the same space requirement, but can be in error if the requirements are different. In this case we revise the estimated transportation costs by developing a distance chart for the new layout and calculating the “real” total transportation costs. This is done whenever an exchange has been identified to be the most useful (based on the estimation of costs) in an iteration. The algorithm continues until no further reductions in the predicted transportation costs can be achieved.

So we summarize the steps to be followed according to CRAFT:

1. Estimate total transportation costs considering all pairwise exchanges of OE that share at least 1 border or that are of same size (i.e. equal number of rectangles).
2. Perform that exchange that leads to the minimum estimated total transportation costs (based on an estimation of distances as described above). If all possible exchanges lead to an increase of predicted total costs, stop here.
3. Revise the estimated distance chart and calculate the new total costs. Go back to step 1.

You see that by applying this procedure the “best” exchange could be passed over, due to estimation errors. This generally will be the case for any improvement algorithm that does not actually evaluate every exchange possible.

Example¹¹: A local manufacturing firm has recently completed construction of a new plant to house 4 departments: A, B, C, and D. The plant is 100m² by 50m². The plant manager has chosen an initial layout of the 4 departments. This layout is given in Figure 2-3. From the figure we see that department A requires 1800 m², department B 1200m², department C 800m², and department D 1200m².

¹¹ Nahmias, S.: Production and Operations Analysis, 4th ed., McGraw-Hill, 2000, Chapter 10

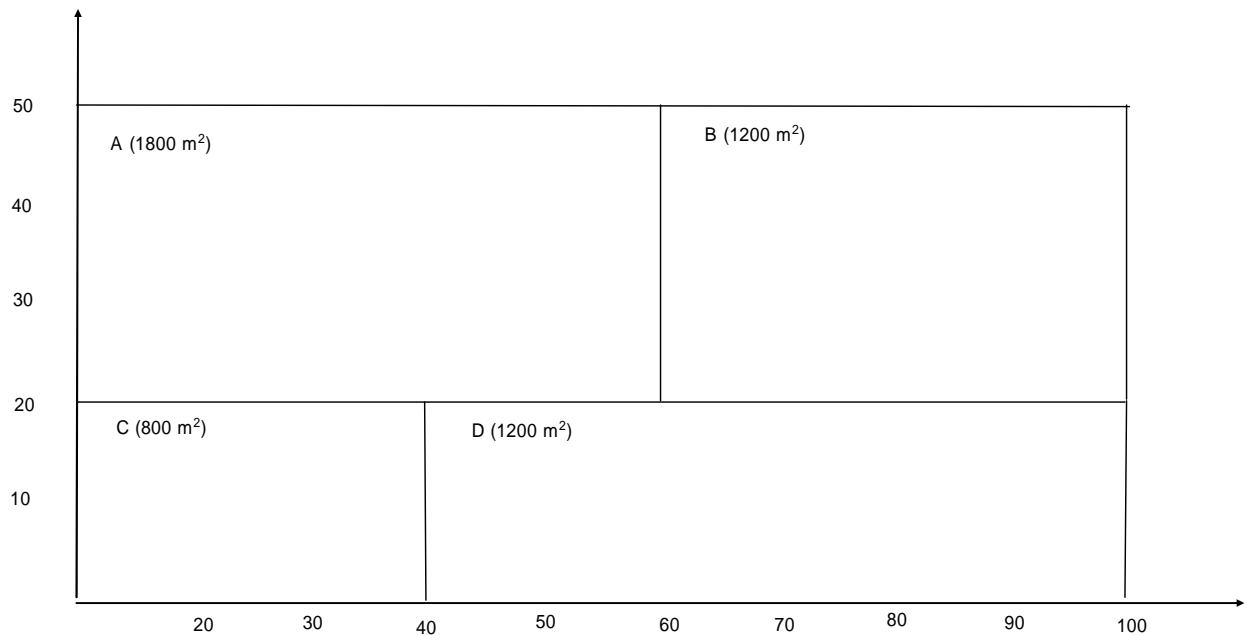


Figure 2-3: Initial plant layout for example problem

The following table contains the material flows between all departments.

Material Flow	A	B	C	D
A	0	2	7	4
B	3	0	5	5
C	6	7	0	3
D	8	2	3	0

The distance between 2 departments is to be assumed as the rectilinear distance between the centroid locations of the corresponding departments. Try to improve the initial layout by applying the CRAFT algorithm (pairwise exchanges).

1. Determination of distances referring to the initial layout: The centroid of a rectangular R is defined by 2 points \bar{x} , \bar{y} .

$$\bar{x} = \frac{(x_1 + x_2)}{2} \quad \bar{y} = \frac{(y_1 + y_2)}{2}$$

The rectilinear distance between 2 centroid locations is

$$d_{12} = |\bar{x}_1 - \bar{x}_2| + |\bar{y}_1 - \bar{y}_2|$$

For the initial layout we obtain the following centroid locations

	Centroid	
	x	y
A	30	35
B	80	35
C	20	10
D	70	10

and the following distances

Distance	A	B	C	D
A	0	50	35	65
B	50	0	85	35
C	35	85	0	50
D	65	35	50	0

- Determination of current total costs:

We multiply the distances with the material flows -> **Total transportation cost: 3050**

- Now we list all possible pairwise exchanges:

A-B -> possible

A-C -> possible

A-D -> possible

B-D -> possible

C-D -> possible

All other exchanges are not possible, because the departments are neither neighboured nor of same size.

- Estimate total costs for each valid exchange (assuming that the centroid locations stay the same):

A-B: estimated transportation cost = 2950

A-C: estimated transportation cost = 2715

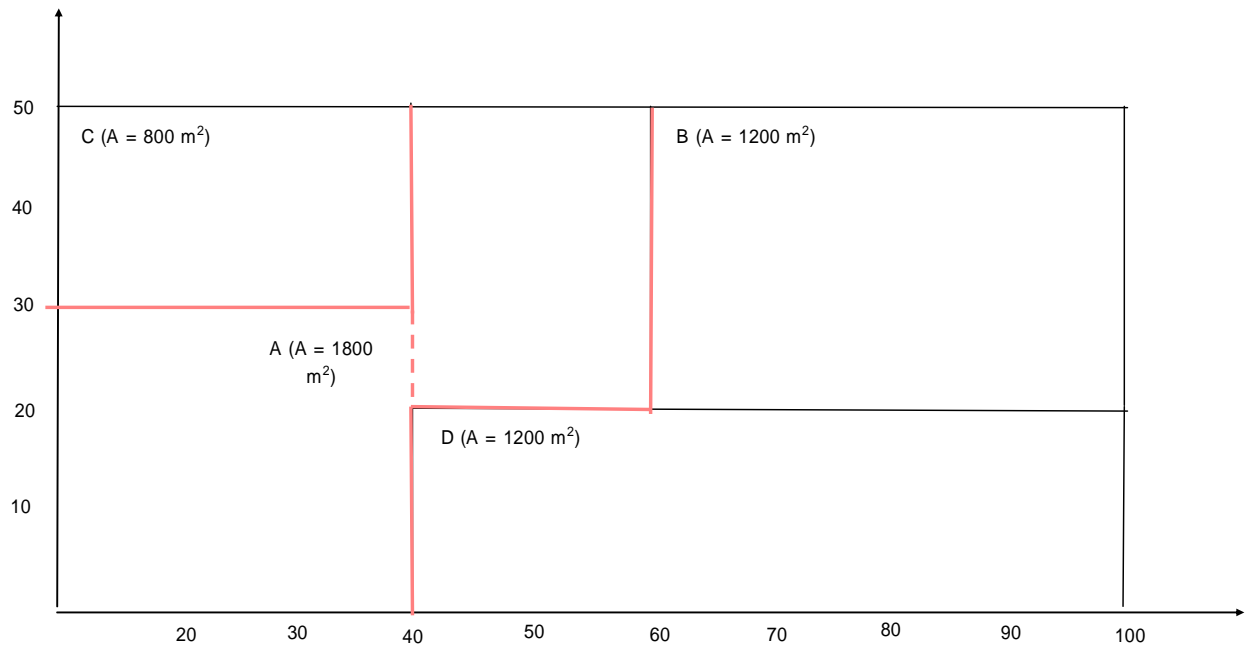
A-D: estimated transportation cost = 3185

B-D: estimated transportation cost = 2735

C-D: estimated transportation cost = 2830

-> Exchange A-C is supposed to be the best one

- Perform the selected exchange and calculate the new total costs.



If a plane R consists of a collection of rectangles R_1, R_2, \dots, R_k the respective boundaries are $[(x_{1i}, x_{2i}, (y_{1i}, y_{2i}))]$ for $1 \leq i \leq k$.

In order to find \bar{x}, \bar{y} (which describe the centroid location), we first have to obtain the moments of R :

$$M_x = \sum_{i=1}^k \frac{x_{2i}^2 - x_{1i}^2}{2} (y_{2i} - y_{1i})$$

$$M_y = \sum_{i=1}^k \frac{y_{2i}^2 - y_{1i}^2}{2} (x_{2i} - x_{1i})$$

Let $A(R)$ be the area of R . Then the centroid of R is given by

$$\bar{x} = \frac{M_x}{A(R)}$$

$$\bar{y} = \frac{M_y}{A(R)}$$

The new centroids are

	Centroid	
	x	y
A	30	21,667
B	80	35
C	20	40
D	70	10

leading to the following new distances

Distance	A	B	C	D
A	0	63,333	28,333	51,667
B	63,333	0	65	35
C	28,333	65	0	80
D	51,667	35	80	0

and total costs of 2809 (we see that we do not achieve the estimated costs for this scenario (2715), but compared to the initial layout's total cost we have a cost reduction anyway).

6. Now we list all possible pairwise exchanges:

A-B -> possible

A-D -> possible

B-D -> possible

All other exchanges are not possible, because the departments are neither neighboured nor of same size.

7. Estimate total costs for each valid exchange (assuming that the centroid locations stay the same):

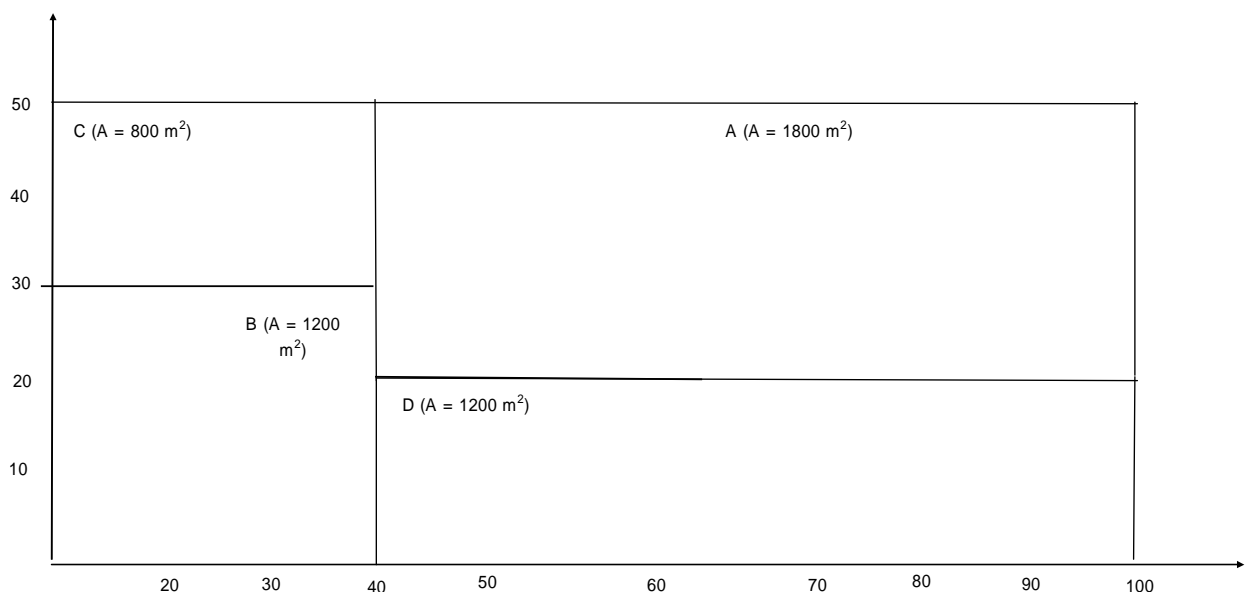
A-B: estimated transportation cost = 2763

A-D: estimated transportation cost = 3228

B-D: estimated transportation cost = 2982

-> Exchange A-C is supposed to be the best one

8. Perform the selected exchange and calculate the new total costs.



The new centroids are

	Centroid	
	x	y
A	70	35
B	20	15
C	20	40
D	70	10

leading to the following distances

Distance	A	B	C	D	
A	0	70	55	25	
B	70	0	25	55	
C	55	25	0	80	
D	25	55	80	0	

and total costs of **2530** (this time we had underestimated the cost reduction).

9. Now we list all possible pairwise exchanges:

A-C -> possible

A-D -> possible

B-C -> possible

B-D -> possible

All other exchanges are not possible, because the departments are neither neighbored nor of same size.

10. Estimate total costs for each valid exchange (assuming that the centroid locations stay the same):

A-C: estimated transportation cost = 3175

A-D: estimated transportation cost = 2735

B-C: estimated transportation cost = 2675

B-D: estimated transportation cost = 3325

-> no further cost reduction is expected! We stop with total costs of **2530!**

1. Appendix A: Methodological Basics

Complexity

Almost all optimization problems occurring in production and logistics can be solved either exactly or by applying heuristic methods. The selection of a solution method may depend on:

- Software availability
- Cost-benefit
- Problem complexity

Even if we know adequate (time consuming) exact methods we are going to apply heuristic methods if we do not have adequate software available or costs (installation, personnel instruction, etc.) exceed the expected benefit.

On the other hand we know a number of combinatorial problems, which are classified to be „NP-*hard*“, which indicates the assumption that the computational effort for solving the problem will not increase polynomial with the problem dimension. In case of real-world applications with the according problem size we face unacceptable computational times, even for high performance IT-systems, regularly.

LP-Problems (average case) are to be solved with polynomial effort, since the number of simplex-iterations increases linearly with the number of constraints (and each iteration causes quadratic effort).

LP-Problems with integer variables usually are solved by applying a Branch and Bound (B&B) method, where a common LP-model is solved in each iteration. Here the number of iterations increases exponentially with the number of integer variables. Thus, these problems cannot be solved with polynomial effort.

For some problem classes (e.g. transportation problems, (linear) assignment) due to their problem structure integer/binary property of the decision variables is guaranteed automatically leading to a low problem complexity.

Some problems with integer/binary variables can (by using special exact methods) be solved with polynomial effort, anyway.

Referring to heuristic methods we usually distinguish between:

- Starting heuristics (quick generation of a feasible solution)
- Improvement heuristics (start with a feasible solution and try to find a better one)
- Combinations of starting and improvement heuristics

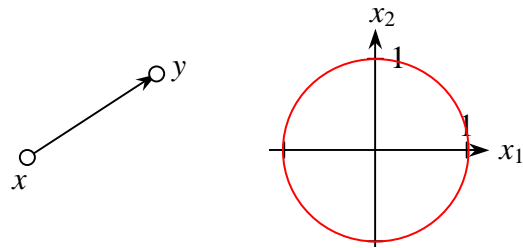
We use “general purpose”-heuristics or metaheuristics (e.g. Simulated Annealing, Tabu Search or Genetic Algorithms) in order to leave local optima during improvement steps.

Costs and distances

The majority of problems being dealt with during this course will be solved based on costs and/or distances c_{ij} . In most cases costs are determined based on given technical parameters (machine setup,..) or distances (e.g. distance between object i and j). In the following we are going to introduce three common distances:

Euclidean distance:
$$d(x,y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

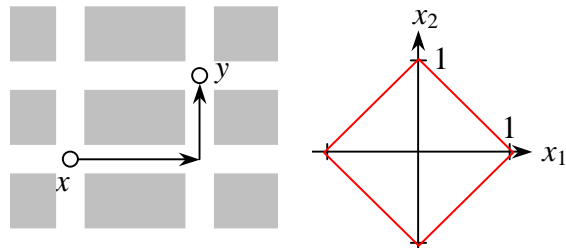
(Straight line distance between two points x and y)



Manhattan distance:

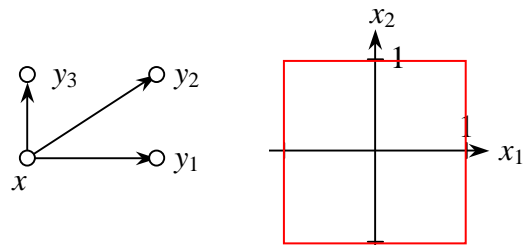
$$d(x,y) = \sum_{i=1}^n |x_i - y_i|$$

(The distance between two points measured along axes at right angle)



Maximum distance:
$$d(x,y) = \max_{i=1,\dots,n} |x_i - y_i|$$

(Drilling plates, movement of cranes,...)

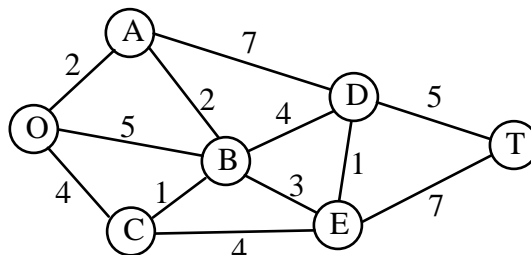


In most cases we know the distance between every couple of adjacent nodes (locations, customers,...). For determining the distance between any two nodes within the network, we have to solve a shortest path problem.

Basics on Graph Theory

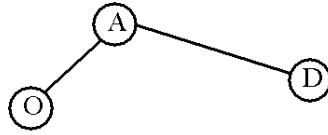
A **graph** consists of points known as **nodes** (*vertices*) which are connected with each other using **lines** (edges, arcs).

Graph:



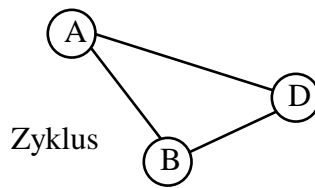
A **chain** between nodes i and j is a sequence of edges connecting these two nodes. A **path** is a chain where the direction is clear (oriented); oriented edges are usually called **arrows** (or arcs).

Chain from A to D:



A **cycle** is a chain that connects a node with itself, while no edge is traversed more than once.

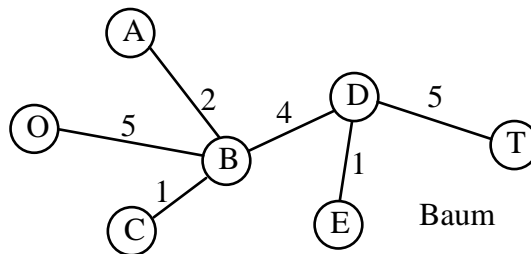
Cycle:



A graph is **connected** if for each pair of nodes there exists a path connecting these two.

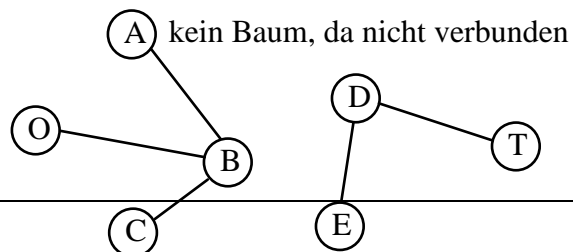
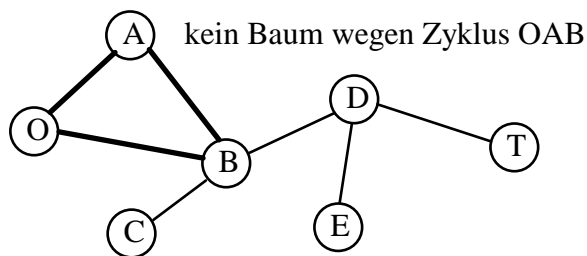
A **tree** is a connected graph without cycles.

Tree:



One of the theorems in graph theory indicates that a graph with n nodes is connected if it contains $(n-1)$ edges, but no cycles.

No tree (due to cycle OAB):



No tree (because not connected):

The edge of graph is *directed* or is an *arrow* if an orientation is given (one way street). A *directed graph* contains only directed arcs. An *undirected graph* contains only undirected edges. A *mixed graph* contains both directed and undirected edges.