Journal of the Society of Korea Industrial and Systems Engineering
Vol, 26, No. 4, pp.42－49, December 2003.

형태제약을 가지는 부서의 다층빌딩 설비배치*

김 채 복

경북대학교 경영학부

Multi-Level Building Layout With Dimension Constraints On Departments

Chae-Bogk Kim

School of Business Administration, Kyungpook National University

The branch and bound techniques based on cut tree and eigenvector have been introduced in the literature[1, 2, 3, 6, 9, 12]. These techniques are used as a basis to allocate departments to floors and then to fit departments with unchangeable dimensions into floors. Grouping algorithms to allocate departments to each floor are developed and branch and bound forms the basis of optimizing using the criteria of rectilinear distance.

The proposed branch and bound technique, in theory, will provide the optimal solution on two dimensional layout. If the runs are time and/or node limited, the proposed method is a strong heuristic. The technique is made further practical by the fact that the solution is constrained such that the rectangular shape dimensions length and width are fixed and a perfect fit is generated if a fit is possible. Computational results obtained by cut tree-based algorithm and eigenvector-based algorithm are shown when the number of floors are two or three and there is an elevator.

Keywords : building layout, eigenvector, cut tree, grouping

1. INTRODUCTION

1.1. Literature Review

The facility layout problem has had an explosion research in interest in the past 20 years. This has been due to the appearance of new techniques such as expert systems analysis and the rapid increase in computing power. In this paper, an interpretation of the cut tree concept of Montreuil and Ratliff[12] and the cluster analysis plot of Drezner[6] based on eigenvector is used with integer programming and a branch and bound approach based on the grid of facility to attack the problem of multi-dimensional constrained layout with a chance of proving optimality with a small number of departments. The constraint is that the department rectangular shape cannot be changed and the fit into the factory rectangle must be perfect.

Since the three dimensional layout problem is an extension of the two dimensional layout problem, the first step to attack the three dimensional layout problem is grouping departments floor by floor. If we know which department will be placed on which floor, we can solve the three dimensional layout problem by any method for the two dimensional layout problem with a little modification. There are three important techniques which handle a multilevel factory : SPACECRAFT[8], MULTI-HOPE[12], BLOCPLAN[5] and MULTIPLE[4].

SPACECRAFT is the first algorithm (improvement type) for multiple floor layout in the literature. It is an extension of CRAFT for multiple floor layout. There are two ex-

* This Research was supported by Kyungpook National University Research Fund, 2003

tensions from CRAFT : vertical travel (nonlinear function) and facility transformation. To analyze vertical travel, SPACECRAFT requires interactive input from the user. For the improvement of the interchanging procedure, SPACECRAFT transforms the actual layout into a long narrow layout in order to allow exchanges much more freely, and changes the size of departments. Like CRAFT, it is also a heuristic and it requires several initial layouts to obtain a better solution. It can handle multiple floors and elevators. It provides distorted shapes of departments in its final solution as CRAFT does.

BLOCPLAN is a layout design system developed by Donaghey and Pire[5]. It is a PC-based system using the concept of zone defense of basketball. It requires a fixed department area but has no constraint on dimensions (i.e., both squares and rectangles are possible). However, the ratio of department length to width must be within a prescribed range. Using several scoring methods and algorithms, it generates a final layout on the screen graphically. BLOCPLAN can handle a multi-level factory as well as generate a final layout with shape distortion without necessarily preserving the rectangle dimensions. It does not consider elevators between floors and flows between departments on the different floors.

Bozer, Meller and Erlebacher[4] extended CRAFT to the multiple floors in order to generate block layout. They defined an upper and lower limit on department sizes. MULTIPLE exchanges the departments in order to improve the objective function (distance based measure) or can squeeze the department size if necessary. For an initial layout, they employed a space filling curve technique and defined two measures for better department shape in the final layout. It can consider elevators and handle multiple floors. It may provide distorted shapes for departments in the final solution.

MULTI-HOPE is a technique to solve multi-floor layout problems based on genetic algorithm. In order to obtain the solution, Kochhar and Heragu[10] employed string representation, initial population generation, crossover, mutation and other operations.

1.2. A Mathematical/Analytical Based Approach

Since the problem in this paper has dimension constraints (no distortion), we cannot use the above approaches directly. In order to resolve the problem, we base our work on the eigenvector[2, 6] and cut tree approach[1, 3, 9]. The solution procedure in this paper consists of two stages like Meller[11] : assignment of departments to the floors and layout determination of each floor in the building. The assumptions used in this paper are as follows.

(1) The total number of floors in the building is given.
(2) Each floor has a rectangular or square shape.
(3) Width of each floor is given and cannot be changed.
(4) Length of each floor is given and cannot be changed.
(5) From-to chart is given.
(6) The total number of departments is given.
(7) The area of one department does not exceed the floor area.
(8) Not all departments have the same size.
(9) All departments have the square or rectangular shape.
(10) The location of the elevator is given a priori.
(11) Each department's area is specified.
(12) The choice of area A of a department has a utility value given by U(A), $0 \leq U(A) \leq 1$.

The above assumptions are not strong since assumptions (1), (6) and (9) are generally true and assumptions (2), (3), (4), (8) and (11) are common. Because of mechanical and architecture problems, we can use assumptions (9) and (10) and satisfy the architectural design. Note that assumption (5) is used also in CRAFT. Assumption (12) may be used to find the exact department area for the floor area when we divide departments into groups for floors and the division does not generate total areas which match floor areas.

Using the above assumptions, methods for layouts of departments for a multi-level factory will be presented. Also, rectilinear distance measure is employed to calculate the total cost. When the rectilinear distance measure is used, we compute the centroid of each department and define distance between department using centroid coordinate.

2. Grouping Departments

In order to group the departments floor by floor, Donaghey and Pire[5] used an interactive method. They ask the decision maker to specify the area difference factor, the maximum difference between the total sum of department areas on a certain floor and the actual floor size, and group the departments according to the area difference factor. For example, if the actual area of a certain floor is 50 square feet and the area difference factor is 10%, then the total sum of department areas on that floor should be in the range [45, 55] square feet.

Another approach for grouping is to provide some flexibility in department size in Bozer et al.[4]. Suppose there is a department whose required area is between A^l, the minimum area and A^u, the maximum area. If a department exchange is required between floors, they squeeze or expand the department area, if necessary, in order to obtain feasibility.

In this paper, we de not use an area difference factor because it is difficult to pick a specific value for each department area and floor area. Instead, the departments are grouped with the area difference factor in the range [0%, 30%] since the run time of grouping departments with several area difference factors is negligible compared to the run time for layout. With an increment of the area difference factor by 5% from 0%, we group the departments floor by floor with the area difference factor (increment step 5% can vary depending on the user). Let F^n, N and F be the total number of floors, the total number of departments and the floors areas, respectively.

The algorithm proposed in this paper is based on the cut tree generated from the cost flow from-to matrix. Details of the computation of this tree are given in [1, 3, 9]. The idea of the proposed algorithm is that departments that are close to each other are grouped on the same floor. If we take the standard definition of distance between nodes i and j (the minimum number of arcs traversed to move from node i to node j), then we can group based on closeness by starting from a node on the perimeter of the tree called a "tip node". If we define the center node as a node for which distances $d(c, j) \forall j$ is a minimum, then a tip node is a node such that $d(c, j)$ is a maximum. This maximum value is called the radius. Proposed algorithm for grouping departments can then be stated as follows.

Algorithm 1. (grouping departments)

Step 1. Calculate a matrix, $S[i],[j]$, which indicates the department of j^{th} relationship with department i from the node-arc distance and flow matrix (see details in [5]). The S matrix has entries s_{ij} where s_{ij} is defined as the department with the j^{th} strongest relation with i as measured by the product of node-arc distance and f_{ij} (flow between i and j) ranked from smallest to largest. Ties are broken automatically if ($s_{ij} \times f_{ij}$) do not determine rank uniquely. Pick any tip node, say x, with radius distance in a cut tree. Set i =0.00, j =1, and k =1.

Step 2. If department $S[x][j]$ is not labeled, sum the area of department and go to *Step* 3.
If department $S[x][j]$ is labeled, j=j+1.
If j>N, go to *Step* 5.
Otherwise, repeat *Step* 2.

Step 3. Let D_l be the department assigned in *Step* 2.
If the total sum of area up to now is less than F, label department D_l. Set j=2 and x=D_l . Go to *Step* 2.
If the total sum area is in the range [F, (1+i)F], label department D_l. Set j=2, x=D_l and k=k+1.
If k>(F^n-1), then go to *Step* 4.
Otherwise, go to *Step* 2.
If the total sum of area is greater than (1+i)F, j=j+1.
If j>N, go to *Step* 5.
Otherwise, go to *Step* 2.

Step 4. Sum all areas of departments unlabeled for F^n group (final floor).
If the total sum of the areas is in the range of [(1-i)F, (1+i)F], save solution and unlabel all departments. ε=ε+0.05.
If ε>0.30, go to *Step* 6.
Otherwise, set k=1 and j=1.
The tip node picked in *Step* 2 is set to x. Go to *Step* 2.
If the total sum of the areas is not in the range of [(1-ε)F, (1+ε)F], Go to *Step* 5.

Step 5. Infeasible solution. i=ε+0.05.
If i>0.30, go to *Step* 6.
Otherwise, set k=1 and j=1. Go to *Step* 2.

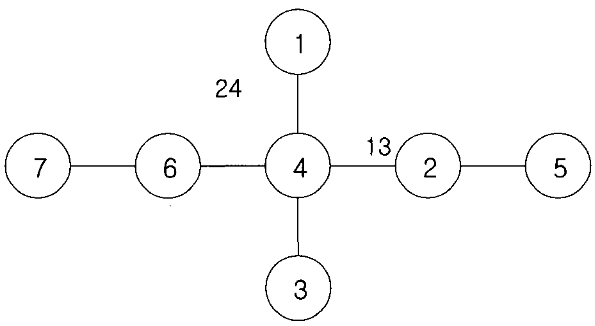
Step 6. Stop.

Example. Consider a 7 department problem. The flow matrix, cut tree figure and node-arc distance matrix are given in Table 1, Figure 1 and Table 2, respectively. If the floor area is 10 and the plant has two floors, we need to group the 7 departments into two sets. The required department sized are in Table 3. First, we construct the S matrix (Table 4) using the node-arc distance and flow matrix. Department 4 is a center node and department 7 is tip node (D_{47}=2 $\geq \forall D_{ij}$), department 7 is a starting node. Add department areas until the sum is equal to ten (assume the area difference factor is

0%). Then, departments 7, 6, 4, 3, and 2 are in one group and departments 1 and 5 are in one group. If the difference factor is 10%, the department 7, 6, 4, 3, 2 and 5 are in one group, and the other group has only department 1.

<Table 1> Flow Matrix (from-to-matrix)

D	1	2	3	4	5	6	7
1	0	8	9	7	0	0	0
2	8	0	0	5	7	0	0
3	9	0	0	4	0	9	0
4	7	5	4	0	4	6	8
5	0	7	0	4	0	0	2
6	0	0	9	6	0	0	11
7	0	0	0	8	2	11	0



<Figure 1> Cut Tree

<Table 2> Distance matrix

D_{ij}	1	2	3	4	5	6	7
1	0	2	2	1	3	2	3
2	2	0	2	1	1	2	3
3	2	2	0	1	3	2	3
4	1	1	1	0	2	1	2
5	3	1	3	2	0	3	4
6	2	2	2	1	3	0	3
7	3	3	3	2	4	1	0

<Table 3> Department sizes

Dept.	1	2	3	4	5	6	7
Size	3×3	2×2	1×2	1×2	1×1	1×1	1×1

<Table 4> S matrix

N	1	2	3	4	5	6	7
1	1	4	3	6	2	7	5
2	2	5	4	6	3	1	7
3	3	4	6	1	2	5	7
4	4	6	3	2	1	5	7
5	5	2	4	3	6	1	7
6	6	7	4	3	2	1	5
7	7	6	4	3	2	1	5

After applying the Algorithm 1 proposed, we can select the groups for each floor and pick the partitioning with the minimum area difference factor. When the sum of department areas in each group is different from the area of floor, we can use a zero-one integer programming to alter department size for an exact area sum equal to the floor size (one can determine the department areas by observation if the optimal department areas are not sought). Since each the department's area has its own utility, the problem for the k^{th} floor is formulated as follows :

$$[P] \quad Maximize \quad \sum_{i=1}^{n_k}\sum_{j=1}^{i_a} U_{ij}\, x_{ij}$$

$$S.T. \quad \sum_{i=1}^{n_k}\sum_{j=1}^{i_a} A_{ij}\, x_{ij} = F$$

$$\sum_{j=1}^{i_a} x_{ij} = 1 \quad \text{for } i=1,2,\ldots,n$$

$$x_{ij}=0 \text{ or } 1$$

,where U_{ij} ≡ utility value of department i when the j^{th} department area is chosen

A_{ij} ≡ area of department i when the j^{th} department area is chosen

x_{ij} ≡ if the j^{th} department area for department i is chosen, $x_{ij} = 1$. Otherwise, $x_{ij} = 0$.

F ≡ floor area

i_a ≡ number of area alternatives for department i

n_k ≡ number of departments for the k^{th} floor

n ≡ total number of departments

In order to increase the possibility that problem [P] has feasible solutions, we need to make the minimum and maximum area of each department more sophisticated. Since the shape of each department is a square or rectangle, consider the length of department i, d_i^l, and the width of department i, d_i^w. Suppose $d_i^l < d_i^w$. Then, the minimum area for department i is set to $A_i^l = d_i^l(d_i^w-1)$, and the maximum area for department i is set to $A_i^u = d_i^l(d_i^w+1)$. If department i requires (1×1) size, there is no minimum area but two area alternatives, (1×2) and (1×3). Another method which makes sure that problem [P] has feasible solutions is to determine the A_{ij} values according to the difference of floor area and total area of departments for the corresponding floor and the number of departments.

The idea is that we can make any number less than or equal to 10 with 4 numbers, 1, 2, 3 and 4. For example, 7 is the summation of 1, 2 and 4. Suppose that we have 800 square feet to make up and 5 departments, first consider the smallest 10^n area required. Then, divide the number of departments by 4 and check the quotient and the remainder. Arrange the departments by decreasing order of department size. Assign the variation of A_{ij} to the multiple of $10^{(n-1)}$. In this example, if the department ordering is (1, 2, 3, 4, 5), then the variation of A_{ij} is (400, 300, 200, 100, 100) from the original area. After solving the integer programming problem, we can obtain the information of which department will be placed on which floor. This problem is NP-complete but the number of variables will be limited since most problems will have N for a floor less than 30. The specialization of the A_{ij} allows for simple heuristics based on the number theory given.

3. Construction of the Three Dimensional Layout

Since we have department sets assigned to the floors, we can change the three dimensional layout problem to the two dimensional layout problem with a little modification. First, consider the elevator in order to calculate the flow between departments placed on different floors. Because we assumed the location of the elevator is known a priori, we can know the distance measure from a department to elevator if we have a layout. There are two points for vertical movement with the elevator : waiting time and moving time. Moving time with the elevator is linear in terms of the differences of the floors. Average waiting time for the elevator can be considered as constant regardless of the floor. The vertical distance with the elevator can be stated as

$$d^e_{ij} = a + b(f^d_{ij})$$

,where a ≡ distance accounting for waiting time

b ≡ distance accounting for vertical movement time

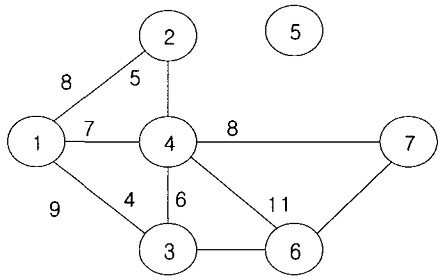
f^d_{ij} ≡ difference between floor with department i and floor with department j

d^e_{ij} ≡ distance accounting for vertical moving time between department i and j

If we know the value of a and b, there is no problem computing the objective function. Another difficulty lies on the connectivity of the network for each floor since we want to group the departments using the node-arc distance and flow matrix. If a node is not connected, it is impossible to apply the Gomory-Hu algorithm[7] to obtain the cut tree information. See Table 5, Figure 2, Table 6 and Figure 3 for two disconnectivity cases.

<Table 5> Illustration of disconnectivity with a node

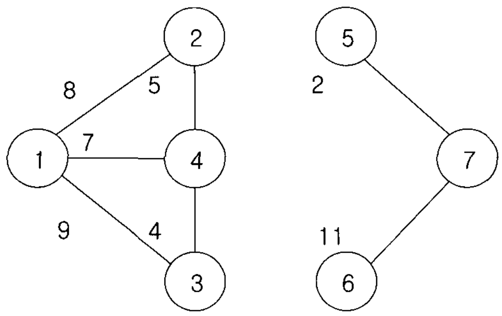
N	1	2	3	4	5	6	7
1	0	8	9	7	0	0	0
2	8	0	0	5	0	0	0
3	9	0	0	4	0	9	0
4	7	5	4	0	0	6	8
5	0	0	0	0	0	0	0
6	0	0	9	6	0	0	11
7	0	0	0	8	0	11	0



<Figure 2> Network with a node disconnected

<Table 6> Illustration of disconnectivity with a node set

N	1	2	3	4	5	6	7
1	0	8	9	7	0	0	0
2	8	0	0	5	0	0	0
3	9	0	0	4	0	9	0
4	7	5	4	0	0	0	0
5	0	0	0	0	0	0	2
6	0	0	0	0	0	0	11
7	0	0	0	0	2	11	0



<Figure 3> Network with a node set disconnected

If a network is divided into more than two sets, the Gomory-Hu algorithm cannot generate a cut tree graph. In order to resolve this problem, we add ε(very small positive number) to all elements except diagonal elements in flow matrix. Then, if the original network is connected it provides the same cut tree. Even though the original network is disconnected a connected network is generated. Different cut values and maximum flow between departments are presented whether the original network is connected or not. Since node-arc distance is not affected by adding ε units of flow, we can use the algorithm in the cut tree approach. Even though the maximum flow values are changed by adding ε, the maximum flow distance order is not affected. That means if the maximum flow between node i and j is greater than the distance between node k and l before adding ε unit, it is true that the maximum flow distance between node i and j is greater than the distance between node k and l after adding ε unit flow. The algorithm for the three dimensional layout problem can be stated as follows.

Algorithm 2. (three dimensional layout)

Step 1. Group departments according to the floors by algorithm 1.

Step 2. Obtain all information about departments on each floor.

Step 3. Layout the first floor. Set i =2.

Step 4. Determine the layout for floor i with the layouts from floor 1 to floor (i-1).

Step 5. i=i+1. If i>F^n, go to *Step* 6. Otherwise, go to *Step* 4.

Step 6. Stop.

4. COMPUTATIONAL RESULTS

We generated 9 test problems for a two floor building and 9 test problems for a three floor building. With Algorithm 1 and Algorithm 2, we solved the 18 test problems by the eigenvector approach and cut tree approach for a particular floor fit preserving dimensional shape. Table 7 and 8 show the departments groups according to the number of floors.

In order to compare the eigenvector approach with the cut tree approach, we rotated the point layout three times. Both approaches limit the number of department candidates for corner location to five. The location of the elevator, waiting time and moving time for the elevator are given by users. Table 9 shows the layout problem with two floors and Table 10 shows the layout problem with three floors when rectilinear distance measure is considered.

From the computational results, the cut tree approach performs better than the eigenvector approach regardless of the distance measures. Even though the eigenvector approach generates a better layout than the cut tree approach in a two dimensional layout (first floor), it provides a worse solution because of the vertical movements. Since Algorithm 2 determines the layout of the first floor without considering the vertical flows, it is possible for the eigenvector approach to generate a worse objective.

<Table 7> Grouping of the departments (two floors)

Test Problem	department set for each group
1	(1, 4, 5, 9, 12, 13, 15, 16, 18, 19, 20) (2, 3, 6, 7, 8, 10, 11, 14, 17)
2	(1, 2, 3, 10, 11, 13, 18) (4, 5, 6, 7, 8, 9, 12, 14, 15, 16, 17, 19, 20)
3	(2, 6, 7, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20) (1, 3, 4, 5, 8, 9, 10)
4	(1, 2, 5, 6, 8, 18, 19) (3, 4, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 20)
5	(2, 5, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20) (1, 3, 4, 6, 7, 8, 9)
6	(3, 7, 8, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20) (1, 2, 4, 5, 6, 9, 10)
7	(2, 3, 5, 6, 8, 12, 14, 15, 18) (1, 4, 7, 9, 10, 11, 13, 16, 17, 19, 20)
8	(1, 3, 5, 8, 9, 13, 15, 16, 19) (2, 4, 6, 7, 10, 11, 12, 14, 17, 18, 20)
9	(1, 4, 6, 8, 10, 11, 12, 14, 16, 17) (2, 3, 5, 7, 9, 13, 15, 18, 19, 20)

<Table 8> Grouping of the departments (three floors)

Test Problem	department set for each group
1	(2, 4, 5, 8, 20, 23, 27, 29, 30) (1, 6, 9, 12, 13, 18, 22, 25, 26, 28) (3, 7, 10, 11, 14, 15, 16, 17, 19, 21, 24)
2	(1, 8, 13, 14, 15, 17, 18, 23, 24, 27, 29, 30) (2, 4, 5, 6, 12, 21, 22) (3, 7, 9, 10, 11, 16, 19, 20, 25, 26, 28)
3	(1, 4, 8, 12, 15, 19, 27, 28, 29, 30) (5, 6, 7, 9, 11, 22, 24, 25, 26) (2, 3, 10, 13, 14, 16, 17, 18, 20, 21, 23)
4	(1, 5, 11, 13, 17, 19, 25, 26, 27, 29) (6, 7, 9, 12, 18, 20, 21, 22, 23, 24, 28, 30) (2, 3, 4, 8, 10, 14, 15, 16)
5	(4, 5, 7, 13, 15, 17, 18, 22, 25, 27, 29) (1, 3, 6, 8, 11, 20) (2, 9, 10, 12, 14, 16, 19, 21, 23, 24, 26, 28, 30)
6	(4, 5, 6, 11, 16, 20, 24, 25, 26, 29) (2, 9, 12, 14, 15, 17, 18, 19, 21, 22, 23, 27, 28, 30) (1, 3, 7, 8, 10, 13)
7	(1, 7, 8, 13, 16, 17, 21, 26, 27, 30) (3, 5, 10, 11, 12, 18, 19, 20, 22, 23, 25) (2, 4, 6, 9, 14, 15, 24, 28, 29)
8	(5, 6, 7, 10, 12, 19, 20, 22, 26, 27, 28) (1, 8, 9, 15, 17, 18, 21, 23, 25, 29, 30) (2, 3, 4, 11, 13, 14, 16, 24)
9	(6, 7, 10, 14, 16, 17, 18, 19, 20, 22, 24, 25, 27, 29) (1, 4, 11, 12, 13, 15, 28, 30) (2, 3, 5, 8, 9, 21, 23, 26)

<Table 9> Computational results (rectilinear distance and two floors)

test problem	floor number	Eigenvector (-,-,-)[a]	cut tree (-,-,-)[b]
1^d	1	(5752, 5752, 5752)	(5847, 6029, 5765)
	2	(38394, 37744, 37954)	(39691, 38125, 37689)
2^d	1	(***, ***, ***)	(***, 3315, 4157)
	2		(35179, 35925, 35485)
3^d	1	(4844, 4796, 4796)	(4882, 4890, 4768)
	2	(***, ***, ***)	(***, 35694, ***)
4	1	(***, ***, ***)	(***, ***, ***)
	2		
5	1	(12816, 13976, 13976)	(13685, ***, 12527)
	2	(***, ***, ***)	(***, ***, ***)
6^d	1	(9312, 9176, 9176)	(9591, 9807, 8843)
	2	(***, ***, ***)	(69991, ***, ***)
7^c	1	(8642, 8624, 8624)	(8976, 9310, 8634)
	2	(99800, 95632, 95456)	(100827, 97461, 96880)
8^d	1	(8702, 8702, 8098)	(***, 8232, 7698)
	2	(95166, 95166, 95166)	(97044, 96104, 94812)
9^c	1	(9520, 9642, 9642)	(9562, 9410, 9564)
	2	(98488, 98080, 94728)	(98545, 95355, 96501)

<Table 10> Computational results (rectilinear distance and three floors)

test problem	floor number	Eigenvector (-,-,-)[a]	cut tree (-,-,-)[b]
1^d	1	(4658, 4406, 4534)	(5142, 4534, 4406)
	2	(36714, 36574, 36574)	(37433, 37433, 36853)
	3	(104970, 104882, 104548)	(104195, 101252, 102903)
2^d	1	(4652, 4476, 4476)	(4465, 4477, 4573)
	2	(30552, 30552, 30552)	(30772, 29652, 30772)
	3	(108256, 108672, 108672)	(108946, 103298, 107678)
3^d	1	(4244, 4244, 4128)	(4495, 4719, 4209)
	2	(46844, 46992, 46992)	(50630, 48720, 48706)
	3	(109100, 108644, 108232)	(111858, 107694, 110746)
4	1	(7220, 7220, 7061)	(7337, 7559, 7061)
	2	(74009, 77241, 77241)	(82970, 75972, 84062)
	3	(169629, 169605, 169605)	(180815, 165426, 180523)
5	1	(10112, 9966, 9966)	(10112, 10514, 10046)
	2	(***, ***, ***)	(***, ***, ***)
	3		
6	1	(8632, 8632, 8184)	(8642, 9442, 8184)
	2	(100732, 100068, 99428)	(100141, 99131, 99355)
	3	(***, ***, ***)	(***, ***, ***)
7^d	1	(9732, 10200, 9552)	(***, 10037, 9605)
	2	(104790, 104072, 104722)	(106073, 104103, 103673)
	3	(258080, 255792, 255800)	(261936, 245470, 257456)
8^d	1	(11212, 11336, 11336)	(11689, 11683, 10867)
	2	(119760, 119400, 119400)	(119791, 116991, 89881)
	3	(267252, 267040, 266948)	(238350, 217626, 236004)
9^d	1	(19900, 19900, 19900)	(19603, 19929, 19819)
	2	(121964, 119536, 119440)	(117840, 120538, 119720)
	3	(261172, 258444, 258444)	(258033, 257239, 257239)

a : best solution of (0 degree, 120 degree, 240 degree) rotations
b : best solution of (node-arc distance, maximum flow distance, size) criteria
c : eigenvector approach provides better solution
d : cut tree approach provides better solution
*** : no feasible fit found

Another reason that the cut tree approach performs better than the eigenvector approach is that we grouped the departments using the information of the cut tree. As the number of floors increase, the cut tree approach provides a better solution since there are more vertical movements among floors. For the three dimensional layout problem, both approaches should be applied because of the vertical movements, especially when the number of elevator locations is less than four. The cut tree models path total flow better than cluster approach based on eigenvector. The arc between nodes on different floors directly represents the flow of the elevator.

5. Conclusion and Suggestions for Further Research

This paper addressed the problem of assigning *n* departments without shape distortion to the multiple floor building. In order to construct automated three dimensional facility layout system, the information of cut tree (distance matrix, maximum flow matrix) is obtained when we run the Gomory-Hu algorithm.

For the three dimensional layout problem, we grouped the departments according to the floors using the node-arc distance and flow matrix. Then, we formulated a zero-one integer programming problem to obtain the exact total area of departments equal to the floor area on a given floor. In order to increase the feasibility, we constructed the minimum and maximum department area. For vertical movement of flow, we considered the elevator. Both the eigenvector approach and cut tree approach were used to solve the problem when the rectilinear distance measure was considered. According to our computational results, we can recommend, when the rectilinear measure is considered, the cut tree approach are recommended because of vertical movements.

The system for the facility layout developed in this paper works when the total sum of the department area is equal to the plant area. Sometimes, it is difficult to obtain a feasible solution even though the total sum of department areas is equal to the plant area without allowing shape distortion. Suppose the plan size is 100 (10×10) and there are two departments whose required sizes are 64 (8×8) and 36 (6×6), respectively. In this case, it is impossible to obtain a feasible solution. Further research for the layout problem without shape distortion will be in how to resolve this type of problem.

REFERENCES

[1] 김채복, "Cut Tree Approach for the Facility Layout Problem", 경영과학, 제11권, 제3호(1994). pp.55-65.

[2] 김채복, 김영식, 이동훈, "Branch and Bound Algorithm for the Facility Layout Problem without Shape Distortion", 한국경영과학회지, 제21권, 제2호(1996). pp.59-70.

[3] 김채복, 박세진, 김철중, "감성 측정평가 시뮬레이터의 설비배치", 대한인간공학회지, 제18권, 제2호 (1999). pp.121-132.

[4] Bozer, Yavuz A., Meller, Russel D. and Erlebacher, Steven J., "An improvement type layout algorithm for multiple floor facilities", *Management Science*, Vol.40, No.7(1994), pp. 918-932.

[5] Donaghey, Charles E. and Pire, Vanina F., "Facility layouts with BLOCPLAN", Industrial Engineering, University of Houston, Houston, TX, 1990.

[6] Drezner, Zvi, "A heuristic procedure for the layout of a large number of facilities", *Management Science*, Vol.33, No.7(1987), pp.901-915.

[7] Gomory, R. E. and Hu, T. C., "Multi-terminal network flows", *Journal of SIAM*, Vol.9, No.4(1961), pp.551-570.

[8] Johnson, Roger V., "SPACECRAFT for multi-floor layout planning", *Management Science*, Vol.28, No.4(1982), pp.407-417.

[9] Kim, Chae-Bogk, Foote, B. L. and Pulat, P. S., "Cut-tree construction for facility layout", *Computers and Industrial Engineering*, Vol.28, No.4(1995), pp.721-730.

[10] Kochhar, J. S. and Heragu, S. S., "MULTI-HOPE : a tool for multiple floor layout problems", *International Journal of Production Research*, Vol.36, No.12(1998), pp.3421-3435.

[11] Meller, R. D., "The nulti-bay manyfacturing floor layout problem", *International Journal of Production Research*, Vol.35, No.5(1997), pp.1229-1237.

[12] Montreuil, Benoit and Ratliff, H. D., "Utilizing cut trees as design skeletons for facility layout", *IIE Transactions*, Vol.21, No.2(1989), pp.136-143.