A binary LP model to the facility layout problem

Christos Papahristodoulou[1]

Abstract

In facility layout problems, a major concern is the optimal design or remodeling of the facilities of an organization. The decision-maker's objective is to arrange the facility in an optimal way, so that the interaction among functions (i.e. machines, inventories, persons) and places (i.e. offices, work locations, depots) is efficient. A simple pure-binary LP model is developed and solved for two small layouts, (up to six functions and six locations). The model is rather flexible and can be used, with small modifications, for larger facility layouts.

JEL classification: C60, C61, C63.

Keywords: Facility Layout, Binary, Linear Programming, Pairs, Assignment.

[1] Department of Economics, Uppsala University, Box 513, S-751 20 Uppsala
Tel: +46 18 4711127, Fax: +46 18 4711478
e-mail: Christos.Papahristodoulou@nek.uu.se

1. Introduction

It is well known that, in facility layout problems one investigates where each function, in a given floor space, will be placed, when all functions interact with each other. Such locations will influence material handling or distance costs and consequently the efficiency of facility. Typical facility layout problems arise in the design or renovation of factories, distribution centers, hospitals, banks, department stores, military supply depots, universities etc., so that functions with high (low) rate of interaction will be placed close (away) to (from) each other. Thus, the distance or time cost of items or persons will be minimized and the efficiency will increase.

Nahmias [1], referring to some studies, argues that the US spent more than $ 500 billion annually on construction and modification of facilities. Effective facilities planning could reduce costs by 10 to 30 percent per year. He also believes that intelligent layout is a key factor to the Japanese production efficiency.

Mainly industrial engineers and researchers in operations research have studied the facility layout problem. Among the first who studied this problem are Armour and Buffa [2]. Francis and White [3], were the first who collected and updated the early research on this area. Two recent studies, the first by Domschke and Drexl [4] and the other by Francis, McGinnis and White [5], have updated later research.

In general, two approaches have been applied to solve facility layout problems with many functions and locations. The first approach is based on greedy pairwise exchange heuristic. It starts with an initial layout and then seeks an improved one by exchanging the locations of a pair of functions. These approaches are ”greedy” in a sense that they often exchange the pair of functions with the largest net reduction in total travel time from the locations. The pairwise exchanges[2] are repeated as long as improvements are possible. Since these heuristics consider only two-way exchanges, they do not guarantee that the optimal layout will be found, if for instance all functions need to be exchanged. The second approach is based on a binary integer quadratic objective function. Particular software packages such as CRAFT (Computerized Relative Allocation of Facilities Technique), by Armour and Buffa [1], SDPIM (Steepest Descent Pairwise Interchange Method) developed by STORM Software [6], or GRASP (Greedy Randomized Adaptive Search Procedure) by Resende, Li and Pardalos [7], claim that these algorithms are

[2] If there is an exchange in locations between functions *i* and *j,* the number of function pairs whose travel time change is *2(n-2),* where *n* is the number of locations or functions. This is because there are *2* ways to choose the member of the pair that must be either *i* or *j*, and *(n-2)* ways to choose the member of the pair that is different from *i* or *j*.

efficient and solve large problems (with more than 15 functions and locations). These algorithms are however, rather complex. For instance, the GRASP algorithm is almost fifteen pages long!

In this paper a simple pure-binary linear programming (PBLP) model has been developed to find the optimal solution for small layout problems. The model is rather flexible and could be applied to larger layouts. As in binary integer quadratic models though, the number of iterations increases dramatically and it may take a long time to find the optimal solution for larger layouts.

2. A PBLP model

2.1. Notation

ij = the pair of functions i and j, $i = 1,...,n$, and $j = 1,...,n$, $i \neq j$;

kl = the pair of locations k and l, $k = 1,...,m$, and $l = 1,...,m$, $k \neq l$;

Although layouts need not be rectangular, to simplify the model formulation, we assume $n = m$. In that case there are $n(n-1)/2$ pair of functions and as many pair of locations.

f_{ij} = mean number of trips from function i to function j;

d_{kl} = cost time from location k to location l;

$C_{ijkl} = f_{ij} \cdot d_{kl}$, facility's total cost;

$$X_{ijkl} = \begin{cases} 1, \text{ if the pairs } ij \text{ and } kl \text{ are placed correctly} \\ 0, \text{ if that assignment is false} \end{cases}$$

There are $[n(n-1)/2]^2$ binary X_{ijkl} variables. There are also $[n(n-1)/2]^2$ cost units, when all functions are reallocated and therefore multiplied by all possible times.

2.2. Model formulation

It is clear from the notation above, that out of $[n(n-1)/2]^2$ binary X_{ijkl} variables, only $n!/(n-2)!2!$ will take the value 1 (because it is equal to the number of ways to choose 2 of the n-functions, for an exchange in location).

The facility's objective function is to minimize the total cost of assigning functions to locations:

$$min \sum_{i,j,k,l=1}^{n} c_{ijkl} x_{ijkl}, \qquad i \neq j \text{ and } k \neq l \qquad (1)$$

The following two constraints ensure that each pair of functions is assigned to exactly one pair of locations, and each pair of locations will place exactly on pair of functions.

$$\sum_{k,l=1}^{n} x_{ijkl} = 1, \qquad i, j = 1,...,n,\ i \neq j \qquad (2)$$

$$\sum_{i,j=1}^{n} x_{ijkl} = 1, \qquad k, l = 1,...,n,\ k \neq l \qquad (3)$$

These constraints though, are not sufficient to exclude inconsistent pair allocations, i.e. the possibility of multiplying correct (incorrect) trips with incorrect (correct) time, when for instance a reallocated pair of functions' trips are multiplied with its initial location and not the new one.

To exclude this inconsistency, new binary variables Y_t, where $t = 1,.....,n \cdot n$, are introduced.

$$Y_t = \begin{cases} 1, \text{ if } n\text{-}1 \text{ function pairs and } n\text{-}1 \text{ location pairs are placed correctly} \\ 0, \text{ if only one function is assigned to a location} \end{cases}$$

The implication of this binary is to catch up all correct pair of locations and combine it with all correct pairs of functions, in such a way, so that wrong multiplication of trips by time will be excluded. Observe that if $Y_t = 0$, for some t, only one function pair ij is placed into a location pair kl, but incorrectly. Alternatively, only one function of that pair either i or j is placed correctly into either k or l.

To understand the function of this binary variable, let us look at the table 1 below (where n=m=5). Consider the first four rows and the first four columns. It is obvious that either one or four of these 16 variables will take the value *1*. For instance, $X_{1214} = 1$ is possible. Although constraints *(1)* and *(2)* are satisfied, we are not certain yet if *"1"* or *"2"* will be placed in *"1"* or *"4"*. That will depend upon which of the remaining X_{ijkl} will take the value *1*. Another possibility is having $X_{1214} = X_{1315} = X_{1412} = X_{1513} = 1$, satisfying of course constraints *(1)* and *(2)*. But, it is not possible to have only two or three of X_{ijkl} equal to *1*! If for instance $X_{1214} = X_{1315} = 1$, we are certain that function *"1"* will be placed in location *"1"*, function *"2"* in *"4"* and function *"3"* in *"5"*. Given the remaining two function pairs *"14"* and *"15"* and the remaining two location pairs *"12"* and *"13"*, two more variables must be equal to *1*, either (X_{1412}, X_{1513}), or (X_{1413}, X_{1512}). Now, it is easy to formulate these constraints.

[Table 1, here]

For instance, when function *"1"* will be paired with all other functions, and might be assigned to all *"1"*- location pairs, that is formulated as:

$$\sum_{j,l=2}^{5} x_{1j1l} - 3\,Y_1 = 1 \qquad (3a)$$

It is now clear that if $Y_1 = 1$, then all four function pairs are placed correctly; if $Y_1 = 0$, only one function will be located somewhere.

By the same token, when function *"2"* will be paired with all other functions, and might be assigned to all *"1"*- location pairs, that is formulated as:

$$\sum_{j=3,l=2}^{5} x_{2j1l} + \sum_{l=2}^{5} x_{121l} - 3\,Y_2 = 1 \qquad (3b)$$

Obviously, it is possible for both Y_1 and Y_2 to be zero, but not both to be equal to *1*. If for instance, $Y_1 = 1$, then four variables of the first block (such as those mentioned above) will be equal to *1*, implying that the functions will be just placed as above. All other possibilities are false, that is $Y_2 = 0$. The first four columns, i.e. constraint (2), will take care of it. On the other hand, the possibility of Y_1 being equal to *0*, does not necessarily imply that $Y_2 = 1$. If for instance, only $X_{1214} = 1$, (which exists in both *(3a)* and *(3b)*), it does not necessarily imply that three more variables in constraint *(3b)* will be equal to *1* too. There are three more functions (i.e. three more binary Y_t) left which can be paired with function *"1"*. Thus, only one of these first five binary Y_t is equal to *1*. Moreover, such a constraint is superfluous due to constraints *(2)* and *(3)*.

We proceed similarly with the remaining functions and locations (23 more constraints of the same kind for $n=m=5$).

In addition, the set of binaries:

$$X_{ijkl} = 0 \text{ or } 1, \quad i, j = 1,\dots,n, \; i \neq j \text{ and } k, l = 1,\dots,n, \; k \neq l \qquad (4)$$

$$Y_t = 0 \text{ or } 1, \quad t = 1,\dots,n \cdot n \qquad (5)$$

Altogether there are 125 binary variables and 45 constraints for $n=m=5$.

3. Modifications

Various modifications to the formulation above are possible. For instance, the rows and columns constraints [*(1)* and *(2)*] are not necessary, if instead, the following constraints are introduced (for $n = m = 5$): (see proof in Appendix)

$$\sum_{t=1}^{5} Yt = 1, \; \sum_{t=6}^{10} Yt = 1, \; \sum_{t=11}^{15} Yt = 1, \; \sum_{t=16}^{20} Yt = 1, \; \sum_{t=21}^{25} Yt = 1 \qquad (6)$$

The PBLP formulation is rather flexible and can be applied to larger layouts of rectangular and no-rectangular structure. Other structural constraints, such as a particular function must be placed at that specific place, are of course easy to formulate and save considerable computing time.

Table 2 summarizes some key points of larger rectangular layouts, consistent with our formulation. For instance, when $n=m=6$, the function-location matrix dimension is 15x15 and includes 225 pair variables. There are 36 sub-matrices of 5x5 dimension and therefore 36 binary variables and 36 constraints where each binary will operate. In addition, there are 30 structural constraints, one for every row and every column. Therefore, constraint *(3a)* where the first binary *(Y_1)* operates will be formulated as:

$$\sum_{j,l=2}^{6} x_{1j1l} - 4\, Y_1 = 1 \qquad (3a)'$$

[Table 2, here]

The same argument applies. Only if $Y_1 = 1$ are all five pairs located correctly. We do not need to examine the possibility of having two, three or four pairs placed correctly, because in these cases all five pairs are placed correctly too.

In general, the coefficient of Y_t is equal to (*n-2*), where *n* is the number of functions or locations.

4. Applications

4.1. Example one (n=m=5)

Limited resources in the public sector necessitate additional measures to increase efficiency. Hospital services is a typical example where doctors' and nurses' time is insufficient to meet the demand by patients. According to nurses' observations and experience, from a small hospital, some, or all five functions located on the same floor, are placed in wrong locations with regard to the daily interactions between each pair of functions. That leads to unnecessary long trips by both the staff and patients. Considerable travel timesavings might be derived if these particular functions are reallocated to rooms.

The following figure[3] depicts the actual allocation of these five functions to the respective location (room).

[Figure 1, here]

The symmetric figure 2 displays two data sources. The first entry shows the time (in seconds) it takes for the hospital staff (and the patients as well) to travel from room to room. For instance, it takes almost half a minute to travel from A (Examination Room) to C (Hematology Lab), or vice versa. The second entry shows the daily frequency of interaction between each pair of functions (which might be the average of an observed period). For instance, a nurse will make 180 single trips following all patients who must go from the Waiting Room (D) to the Hematology Lab (C), and another 150 single trips from A to B. Contrary to the distance observations which are easy to collect, the trips are of course not stable and change very often. That makes the problem dynamic or stochastic and would be extremely difficult to formulate, solve and above all implement all statistically significant layouts. Therefore are all these problems disregarded in this example.

[Figure 2, here]

[3] A similar example is found in Shogan [8].

In addition, there are some other idiosyncrasies with a hospital's layout. For instance, different functions might need different space and must be placed in certain locations. Even safety or aesthetic reasons should be taken into consideration when facilities are to be located. Obviously, the importance of idiosyncrasies, of fixed costs and the large fluctuations in the average number of trips, must be taken into account before the optimal layout is implemented. Such additional constraints though are easy to formulate.

It is meaningless to know the number of patients. Some of them will have to go through all the functions while others might need only an X-ray or a Hematology examination. Some of them might go straight to get their Medical Records from previous examinations, while the majority of them will have to wait at the Waiting room. Even if all of them must wait at the Waiting room before they go to the specific investigation rooms, we cannot count the number of patient-services.[4] In addition, since the examination sequence is not taken into account, the number of patient-trips shown in figure 2 is the maximum amount from both directions. For instance, 200 patient-trips between B and C should be regarded as any combination of 200 patients going between B and C, such as 120 from B to C and 80 from C to B.

If we multiply the travel time with the number of trips made daily, we compute the total travel time (cost). According to this layout, it takes 24,290 seconds (almost 6 hours and three-quarters).

We formulated the problem as a PBLP model[5], and solved it in Mathematica [9], in almost twenty minutes of computing time[6]. The optimal solution is:

---

[4] The maximum amount of patients (not patient-services) is all those who start from the Waiting room, i.e. 460, if they go straight to these different functions and never return there. A possible solution consistent with 460 patients is given in the following matrix (minus indicates inflow of patients from the respective room):

		To				
		X-ray	Hematol.	Examin.	Med. Rec.	Out
	Waiting	0	180	230	50	0
	X-ray	0	-200	150	-60	-110
From	Hematol.	200	0	-130	100	170
	Examin.	-150	130	0	70	50
	Med. Rec.	60	-100	-70	0	-110
	Out	110	10	180	160	460

[5] In our formulation we set A = 1, B = 2, C = 3, D = 4, E = 5 and Waiting room = 1, Examination room = 2, X-ray room = 3, Hematology Lab = 4 and Medical Records = 5.

[6] QSB+ [10] provided the same solution in 6 minutes (after 59 iterations).

$X_{1214} = X_{1315} = X_{1412} = X_{1513} = X_{2345} = X_{2424} = X_{2534} = X_{3425} = X_{3535} = X_{4523} = 1,$

i.e., $Y_1 = Y_9 = Y_{15} = Y_{17} = Y_{23} = 1$

Minimum objective function = 20,940 seconds (i.e. almost 5 hours and 49 minutes), that is an efficiency gain of 55 minutes per day (almost 14 %), compared with the actual layout. All five functions were placed incorrectly! Observe that although $X_{1214} = 1$, as in the initial layout, the Waiting and Examination Rooms have now changed place, so that the number of trips and the time remains unchanged for that pair. In addition, the X-ray shifted from B to E, the Medical Records from E to C and the Hematology Lab from C to B. The solution is shown in Table 3. The initial layout variables are marked in italics.

[Table 3, here]

The problem has been also solved using the modified formulation (without constraints *(2)* and *(3)*. Unfortunately, it took more than four hours to solve this modified formulation. Thus, as usual with integer LP formulations, the execution time increases if one tries to save some constraints explicitly, with the help of additional binaries.

4.2. Example two (n=m=6)

Consider a university, which plans to rebuild and reallocate its economics and business departments. The building is designed to house six departments, (economics, economic history, business administration, management, statistics and information science). Assume that all departments are of the same size, although the number of students who study the particular subject varies. In addition, some of them take courses in all departments, while others in some departments only. The average time each student needs to get to and from classes in the building depends upon the location of the department in which he or she takes courses. The distance in minutes between the centers of the six departments, is also known. The *initial* and **optimal** layout is shown below, where rows are pairs of students and columns are pairs of departments.

In the *initial* layout, the assigned pairs are: *A:2, B:1, C:3, D:4, E:6, F:5,* i. e. $Y_2 = Y_7 = Y_{15} = Y_{22} = Y_{30} = Y_{35} = 1,$ with a minimum value of 39,840. In the **optimal** layout, all students change place, since the assigned pairs are: A:1, B:2, C:6, D:5, E:3, F:4, i.e. $Y_1 = Y_8 = Y_{18} = Y_{23} = Y_{27} = Y_{34} = 1$, with a minimum value of 35,650. Observe also that there are two variables (marked with a star) that take the value *1* in both layouts, implying that students 1,2 shift locations with A,B and 3,6 shift locations with C,E.

[Table 4, here]

It took almost seven and a half-hours to solve that problem in Mathematica. Considerable time can be saved though, if the following simple steps are considered.

Step 1: Consider the binary Y_t Table 5. Start first with function ”*1*” (the first binary row) and select the location which will be placed (based on the lowest value in objective function). Take then all other rows (one at a time), to check if another function will be placed at the same location. There are *n* subproblems to solve. The minimum objective function (31,850) is obtained when $Y_1 = 1$. Thus, function ”*1*” is located at location ”*1*”.

Step 2: Set all other binaries at the same row and column equal to zero, and continue with function ”2” and all remaining (five) locations (one at a time) as before. Apply the same criterion, when the binaries are selected. There are now *(n-1)* subproblems to solve. The minimum objective function (32,325) is obtained when $Y_8 = 1$. Thus, function ”*2*” is located at location ”*2*”.

[Table 5, here]

Step 3 to 6: Set all other binaries at the same row and column equal to zero, and continue with all remaining functions and locations as before. Apply the same criterion, when the binaries are selected. The minimum objective function (36,195) is obtained (in sequence) when $Y_{15} = Y_{22} = Y_{30} = Y_{35} = 1$.

Step 7: To check if that solution is optimal or not, (because the problem was solved recursively) there are at least two options. (i) Use the minimum objective function found, as an upper bound and solve the entire problem (with all its 36 binaries). Mathematica found the optimal solution (35,650) in almost five hours.

(ii) Check instead, after each step, if the optimal solution has been found already, by taking into consideration higher objective values. In step three for instance, the minimum objective function was 32,845, when $Y_{15} = 1$. That value increased finally to 36,195 (which is suboptimal) when the remaining three functions were placed correctly. Moreover, in the same step, the third higher value in objective function (35,650), placed even all other functions correctly too, i.e., it was in fact the optimal solution. To check it, we set that value as an upper bound and solve the entire problem. Mathematica solved it, in slightly more than four hours.

5. Conclusions

Facility layout or quadratic assignment problems are challenging, interesting and easily understood by an average intelligent decision-maker. These problems are more or less hard to solve though. Therefore have many researchers devoted considerable amount of time to develop specific algorithms or heuristics to increase computation speed. Despite the fact that some of the existing programs claim that they provide optimal solution for very large layouts, they are rather complex and expensive and therefore not appropriate for many small companies, whose layout problems are relatively small.

In this paper, a pure binary LP model was developed and solved for rectangular layouts with *5* and *6* functions and locations. The main point of this model is to take into account (and model) the multiplicative (or quadratic) feature which exists in layout problems, i.e., the cost of assigning function i to location k depends also upon where the other functions are located. By using constraint *(3),* where the binary Y_t operates, the model disregards all costs, unless all function pairs ij are correctly paired with locations kl.

Like all typical quadratic assignment formulations unfortunately, it was very difficult to solve when the number of functions and locations increased to six. That is expected, since the entire model consists of hundreds of binaries. The model has not been formulated yet for $n = m > 6$. I expect that there will be an optimal solution, if my PC does not run out of memory! In that case, additional modifications or simplifications should be required.

Acknowledgements

The author is grateful to the Faculty of Social Sciences, Uppsala University, for financial support.

Appendix

Proposition 1: When constraint *(1)* is replaced by the new constraints *(6)*, more than one X_{ijkl} is not possible to appear at the same raw.

Proof: Let us assume that the first four variables of the second row equal to one. Since these variables are part of constraints *(3a)* and *(3c)* also, it would imply that $Y_1 = Y_3 = 1$, which contradicts the first, new constraint. The same argument applies to any set of four X_{ijkl} at the same row, because they always appear together with two Y_t of the same constraint, like the above. Thus it is not possible to have more than one X_{ijkl} at the same raw.

Proposition 2: When constraint *(2)* is replaced by the new constraints *(6),* more than one X_{ijkl} is not possible to appear at the same column.

Assume instead that the first four variables of the first column equal to one. Since these variables are part of constraints *(3a)* and *(3f)* also, it would imply that $Y_1 = Y_6 = 1$, thus $Y_2 = Y_3 = Y_4 = Y_5 = Y_7 = Y_8 = Y_9 = Y_{10} = 0$. Thus, the function-location matrix would be transformed to the following table A1.

[Table A1, here]

Moreover, constraints *(3k)* to *(3o),* together with $\sum_{t=11}^{15} Yt = 1$, imply that four variables from either columns 8 and/or 9 will take the value 1 and all others the value 0. Two of these variables X_{1534} and X_{1535} appear in all constraints *(3k)* to *(3o).* If they also take the value 1, it contradicts the previous result that it is not possible to have more than one X_{ijkl} at the same raw. If both take the value 0, there are three (and not four) relevant variable pairs left to take the value 1 (the thick ones). In that case, constraint $\sum_{t=11}^{15} Yt = 1$, is violated. A similar argument applies for all other combinations of variables from columns 9 and 10. Thus, it is not possible to have more than one X_{ijkl} at the same column either.

References

[1] S. Nahmias, Production and Operations Analysis, 3rd ed. Irwin/McGraw-Hill, Singapore, (1997).

[2] J.M. Armour, E.S. Buffa, A Heuristic Algorithm and Simulation Approach to Relative Allocation of Facilities, Management Science 9 (1963) 294-309.

[3] R.L. Francis, J.A. White, Facility Layout and Location: An Analytical Approach, Prentice-Hall, Englewood Cliffs, NJ, (1974).

[4] W. Domschke, A.Drexl, Location and Layout Planning: An International Bibliography, Springer Verlag, Berlin, (1985).

[5] R.L. Francis, L.F. McGinnis, J.A. White, Facility Layout and Location: An Analytical Approach, Prentice-Hall, Englewood Cliffs, NJ, (1992).

[6] H. Emmons et al., STORM, Quantitative Modeling for Decision Support, Prentice Hall, Englewood Cliffs, NJ, (1992).

[7] M.G.C. Resende, Y Li, P.M. Pardalos, A Greedy Randomized Adaptive Search Procedure (GRASP) for the Quadratic Assignement Problem (QAP), Transactions on Mathematical Software 22 (1) (1996) 104-118.

[8] A.W. Shogan, Heuristics (suppl. to Management Science), Prentice-Hall, Englewood Cliffs, NJ, (1988).

[9] S. Wolfram, The Mathematica book, 3rd ed, Cambridge University Press, (1996).

[10] Y-L. Chang, R.S. Sullivan, QSB+, Quantitative Systems for Business Plus, Prentice Hall, Englewood Cliffs, NJ, (1989).

	12	***13***	***14***	***15***	***23***	***…***	***…***	***45***
12	x_{1212}	x_{1213}	x_{1214}	x_{1215}				
13	x_{1312}	x_{1313}	x_{1314}	x_{1315}				
14	x_{1412}	x_{1413}	x_{1414}	x_{1415}				
15	x_{1512}	x_{1513}	x_{1514}	x_{1515}				
23								
…								
…								
45								

Table 1: Function 1 and location 1 paired with the rest four (n=m=5)

Functions & Locations	Number of Layouts N!	Number of X Variables $[n!/2!(n-2)!]^2$	Binary Y Variables n*n	Number of Constraints
4,4	24	36	16	2*6 + 16
5,5	120	100	25	2*10 + 25
6,6	720	225	36	2*15 + 36
7,7	5040	441	49	2*21 + 49
8,8	40320	784	64	2*28 + 64

Table 2: Key characteristics of rectangular layouts

A | | **C**
Examination Room | | Hematology Lab

B
X-ray Room

D | | **E**
Waiting Room | | Medical Records

Entrance

Figure 1: Hospital's actual layout

	(A) **Examin. Room**	(B) **X-ray**	(C) **Hematology**	(E) **Med. Records**
(D) **Waiting Room**	(10, 230)	(15, 0)	(35, 180)	(28, 50)
(A) **Examin. Room**		(18, 150)	(28, 130)	(35, 70)
(B) **X-ray Room**			(18, 200)	(15, 60)
(C) **Hematology Lab**				(10, 100)

Figure 2: Time and Patient-trips between each pair of functions

	AB	**AC**	**AD**	**AE**	**BC**	**BD**	**BE**	**CD**	**CE**	**DE**	**time**
WR-ER			**X**, *X*								2,300
WR-XR				**X**		*X*					0
WR-He	**X**							*X*			3,240
WR-MR		**X**								*X*	1,400
ER-XR	*X*									**X**	4,200
He-ER		*X*				**X**					1,950
MR-ER				*X*				**X**			2,450
He-XR					*X*		**X**				3,000
MR-XR							*X*		**X**		600
He-MR					**X**				*X*		1,800

Table 3: The **optimal** and the *initial* solution *(n=m=5)*

	AB	***AC***	***AD***	***AE***	***AF***	***BC***	***BD***	***BE***	***BF***	***CD***	***CE***	***CF***	***DE***	***DF***	***EF***
12	X*														
13				**X**		*X*									
14					**X**		*X*								
15			**X**						*X*						
16		**X**						*X*							
23		*X*						**X**							
24									**X**						
25					*X*		**X**								
26				*X*		**X**									
34										*X*					**X**
35												*X*	**X**		
36											X*				
45														***X***	
46												**X**	*X*		
56										**X**					*X*

Table 4: The **optimal** and the *initial* solution *(n=m=6)*

	1	2	3	4	5	6
1	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6
2	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}
3	Y_{13}	Y_{14}	Y_{15}	Y_{16}	Y_{17}	Y_{18}
4	Y_{19}	Y_{20}	Y_{21}	Y_{22}	Y_{23}	Y_{24}
5	Y_{25}	Y_{26}	Y_{27}	Y_{28}	Y_{29}	Y_{30}
6	Y_{31}	Y_{32}	Y_{33}	Y_{34}	Y_{35}	Y_{36}

Table 5: The binary Y_t table *(n=m=6)*

	12	*13*	*14*	*15*	*23*	*24*	*25*	*34*	*35*	*45*
12	1	0	0	0	0	0	0	x_{1234}	x_{1235}	x_{1245}
13	1	0	0	0	0	0	0	x_{1334}	x_{1335}	x_{1345}
14	1	0	0	0	0	0	0	x_{1434}	x_{1435}	x_{1445}
15	1	0	0	0	0	0	0	x_{1534}	x_{1535}	x_{1545}
23	0	0	0	0	0	0	0	x_{2334}	x_{2335}	x_{2345}
24	0	0	0	0	0	0	0	x_{2434}	x_{2435}	x_{2445}
25	0	0	0	0	0	0	0	$\mathbf{x_{2534}}$	$\mathbf{x_{2535}}$	x_{2545}
34	0	0	0	0	0	0	0	x_{3434}	x_{3435}	x_{3445}
35	0	0	0	0	0	0	0	$\mathbf{x_{3534}}$	$\mathbf{x_{3535}}$	x_{3545}
45	0	0	0	0	0	0	0	$\mathbf{x_{4534}}$	$\mathbf{x_{4535}}$	x_{4545}

Table A1: The transformed function-location table when $X_{1j12} = 1$, $j = 2,...,5$