Available online at www.sciencedirect.com
ScienceDirect
Computers & Industrial Engineering 55 (2008) 606–619
computers & industrial engineering
www.elsevier.com/locate/caie

Optimal solution for the two-dimensional facility layout problem using a branch-and-bound algorithm

Maghsud Solimanpur [a,*], Amir Jafari [b]

[a] *Faculty of Engineering, Urmia University, Urmia, Beheshti Avenue, Urmia, West Azerbaijan Province 57153, Iran*
[b] *Doctoral School on Robotics, Neuroscience and Nanotechnology, University of Genova, Genova 16145, Italy*

Received 28 February 2007; received in revised form 27 January 2008; accepted 30 January 2008
Available online 7 February 2008

Abstract

Facilities layout problem is one of the important issues affecting the productivity of manufacturing systems. This problem deals with the determination of optimum arrangement of manufacturing facilities with respect to different layout patterns. A two-dimensional layout is an arrangement fashion in which the manufacturing facilities are laid in a planar area. In this paper, a mixed-integer nonlinear mathematical programming model is proposed for determining the optimum layout of machines in a two-dimensional area. The parameters considered by the proposed model are (a) production capacity of machines, (b) multiple machines of each type (machine redundancy), (c) processing route of parts, (d) dimensions of machines. A technique is used to linearize the formulated nonlinear model. An algorithm based on branch-and-bound approach is proposed to obtain the optimal solution of the proposed mathematical programming model. A simple illustrative example is discussed to demonstrate the technique, and then small-, medium-, and large-sized problems are solved. Comparison of the layout obtained from the proposed model indicates that the proposed model considerably reduces the total distance traveled by products as compared to an optimum process layout configuration for small- and medium-sized problems. The paper concludes that the proposed branch-and-bound approach performs inefficient for large-sized problems. For large-sized problems, the proposed mathematical programming model should be solved through meta-heuristics like genetic algorithms, tabu search, etc.

Keywords: Facility planning; Layout problem; Mathematical programming; Branch-and-bound approach

1. Introduction

Some design factors such as machines dimensions, capacity of machines, production volumes, processing routes, etc. are to be considered to achieve a good facility layout in a manufacturing environment. Regarding these factors, a layout planner seeks to obtain a layout pattern which is effective according to some measures.

* Corresponding author. Tel.: +98 441 2777040; fax: +98 441 2777022.
E-mail address: m.solimanpur@mail.urmia.ac.ir (M. Solimanpur).

0360-8352/$ - see front matter
doi:10.1016/j.cie.2008.01.018

One of the most important measures to be minimized in developing a facility layout pattern is the total material handling distance/cost (Heragu & Kusiak, 1988).

In a manufacturing system, the material handling function includes transportation of work-in-process (WIP), finished parts, materials and tools between machines or workstations. While developing a layout, it is essential that the location of machines/workstations be such that the total distance traveled by personnel or material handling devices throughout the shop floor is minimized. When minimizing this measure, some design constraints are to be taken into consideration. For example, appropriate space should be considered between machines to facilitate temporary storage of WIP. It is notable that in facility layout planning the word 'machine' has a general meaning and may be meant as machine tools, departments, etc.

There are some layout types which are used in manufacturing systems. These layouts can be classified as process layout, product layout, cellular layout (GT), fixed-position layout, hybrid layout, etc. In these layout types, facilities can be arranged in different shapes like single-row, multi-row, circular, U-shape, etc (Solimanpur, Vrat, & Shankar, 2005). However, enforcement of facilities to get arranged in a pre-specified layout shape may increase the total distance traveled by the materials. In fact, the layout shapes mentioned above are specific forms of a general layout problem called as two-dimensional layout (TDL) (Heragu & Kusiak, 1991).

In TDL, the arrangement of facilities is not dictated by the layout shape. Instead, in TDL, the attempt is to minimize the total distance so that the clearance required between the machines is maintained. A nonlinear mathematical programming model is proposed in this paper for a two-dimensional layout problem.

2. Literature review

Love and Wong (1976a) presented a linear mixed-integer programming model for the two-dimensional layout problem. Kusiak and Heragu (1987) surveyed the optimal and heuristic models and algorithms for the two-dimensional facility layout problem. Two classes of optimal algorithms have been used to solve the facility layout problem: branch-and-bound algorithms and cutting-plane algorithms. The algorithms developed by Lawer (1963) and Kaku and Thompson (1986) are examples of optimal branch-and-bound algorithms and those developed by Bazzara and Sherali (1980) and Bukard and Bonniger (1983) are examples of cutting plane algorithms that have been developed for the linear transformations of Quadratic Assignment Problem (QAP) model. Some recent researches on layout problem are reviewed in the following.

Montreuil (1990) introduced a mixed-integer programming (MIP) model for facilities planning that has been used as the basis for several rounding heuristics.

Meller, Narayanan, and Vance (1998) reformulated Montreuil's model by redefining his binary variables and tightening the department area constraints. They proposed some general classes of valid inequalities. Using these inequalities in a branch-and-bound algorithm, they were able to moderately increase the range of solvable problems.

Mir and Imam (2001) presented a hybrid optimization approach for the layout design of unequal-area facilities. Simulated annealing was used to optimize a randomly generated initial placement on an "extended plane" considering the unequal area facilities enclosed in magnified envelops blocks. An analytical method was then applied to obtain the optimum placement of each envelop block. Stepwise reduction of sizes of the envelop blocks allowed controlled convergence in a multi-phase optimization process.

Heragu and Alfa (2003) experimentally analyzed performance of simulated annealing-based algorithms for the layout problem. In their research, results from an experimental analysis involving the two-way and three-way exchange algorithms, a modified penalty algorithm, the simulated annealing algorithm and a hybrid simulated annealing algorithm were presented. The hybrid simulated annealing algorithm uses the modified penalty algorithm to generate an initial solution and then improves it using simulated annealing. It was tested on one-dimensional layout problem with facilities of unequal area and two-dimensional layout problem with facilities of equal area.

Sherali, Fraticelli, and Meller (2003) presented an improved mixed-integer programming (MIP) model with effective solution strategies for the facility layout problem motivated by the work of Meller et al. (1998).

Konak, Kulturel-Konak, and Bryan (2004) developed a mixed-integer programming formulation to find optimal solutions for the block layout problem with unequal departmental areas arranged in flexible bays. The nonlinear department area constraints were modeled in a continuous plane without using any surrogate constraints.

Amaral (2006) considered the layout of arranging a number of departments on a line and presented an exact solution methodology. In this research, a new mixed-integer linear programming model was proposed for the problem.

Tavakkoli-Moghaddam, Javadian, Javadi, and Safaei (2007) presented a new mathematical model to solve a facility layout problem in cellular manufacturing systems (CMSs) with stochastic demand. The objective was to minimize the total costs of inter and intra-cell movements in both machine and cell layout problem simultaneously.

Urban, Chiang, and Russel (2000) proposed a model that did not require the machines to be placed in a functional layout or in a cellular arrangement, but allowed the material flow requirements to dictate the machine placement. They assumed: (i) there is more than one product each with a deterministic demand rate, (ii) there are various types of machines to produce each product, (iii) the time required to process each product on a given machine and the capacity of each machine are known, and (iv) the processing route of each product is known. We have also considered these factors in the model presented in this paper. However, the model formulated in Urban et al. (2000) assumes that the location of sites is known a priori. The formulation presented in this paper is more general because the locations of sites are not required to be known in advance.

3. Problem formulation

3.1. Assumptions

The mathematical model proposed in this paper has been developed with respect to the following assumptions:

(i) machines are to be arranged in a planar area,
(ii) machines are to be oriented in only one given direction,
(iii) each machine has a predetermined rectangular shape,
(iv) machines are to be located in a restricted area,
(v) the capacity of machines is limited,
(vi) the processing route of each product is known in advance,
(vii) the demand for each product is known.

3.2. Notation

The following notation is used in this paper to describe the formulated mathematical programming model:

I number of machine types,
N_i number of machines of type i,
M total number of machines; $M = \sum_{i=1}^{I} N_i$,
P number of products to be manufactured,
D_p demand of product p,
C_i capacity of each machine of type i,
b_{ip} machine requirement parameter, which is 1 if product p needs machine type i, and zero, otherwise.
a_{ij}^{p} processing route parameter, which is 1 if machine type j is needed just after machine type i in the processing route of product p; and zero, otherwise.
Z_{inp} a decision variable, which is 1 if product p should be produced by the nth machine of type i; and zero, otherwise.
t_{ip} processing time of product p on machine type i,

l_i length of machines of type i,
b_i width of machines of type i,
x_{in} horizontal coordinates of the centroid of nth machine of type i,
y_{in} vertical coordinates of the centroid of nth machine of type i,
d_{ij} minimum centroid-to-centroid distance required between machines of type i and j,
h horizontal length of available area,
v vertical length of available area.

3.3. Proposed mathematical models

3.3.1. Model M1

This model is a two-dimensional extension of the ABSMODEL presented by Heragu and Kusiak (1991). However, in the proposed model, we use a pair (i,n) of indices (rather than a single index) to address the nth machine of type i and that the flow between each pair of machines is not fixed.

The objective function for this problem is to minimize the total distance traveled by the materials in the shop floor. This objective can be mathematically represented as follows:

$$\sum_{p=1}^{P}\sum_{i=1}^{I}\sum_{j=1(j\neq i)}^{I}\sum_{n=1}^{N_i}\sum_{r=1}^{N_j} a_{ij}^{p} D_p Z_{inp} Z_{jrp}(|x_{in}-x_{jr}|+|y_{in}-y_{jr}|) \tag{1}$$

The constraints set includes the following:

C1. Ensure the location of each machine is feasible:

$$0 \leqslant |x_{in}-x_{jr}| \leqslant h \tag{2}$$
$$0 \leqslant |y_{in}-y_{jr}| \leqslant v \tag{3}$$

C2. Ensure that no two facilities in the layout overlap:

$$|x_{in}-x_{jr}| + M\alpha_{injr} \geqslant .5*(l_i+l_j)+d_{ij} \tag{4}$$
$$|y_{in}-y_{jr}| + M(1-\alpha_{injr}) \geqslant .5*(b_i+b_j)+d_{ij} \tag{5}$$
$$\alpha_{injr}(1-\alpha_{injr}) = 0 \quad i=1,\ldots I \ \&\ j=i+1,\ldots,I \ \&\ n=1,\ldots,N_i \ \&\ r=1,\ldots,N_j \tag{6}$$

where M is an arbitrary large positive number.

Constraint (6) ensures that only one of the constraints (4) and (5) holds. It should be noted that the solution of this model may result in a layout in which there is empty space between machines, which is unavoidable in practical machine layout.

C3. Ensure that product p visits all the machine types required in its route:

$$\sum_{n=1}^{N_i} Z_{inp} = b_{ip} \quad i=1,\ldots,I \ \&\ p=1,\ldots,P \tag{7}$$

C4. Ensure that the capacity of each machine is maintained:

$$\sum_{p=1}^{P} Z_{inp} t_{ip} D_p \leqslant C_i \quad i=1,\ldots,I \ \&\ n=1,\ldots,N_i \tag{8}$$

3.3.2. Model M2

The model M1 is a nonlinear mixed-integer programming model with an absolute operator in the objective function and constraints (2)–(5). Presence of absolute function in the model makes it difficult to optimally solve the problem. The technique applied in Kusiak (1990) is used here to linearize the absolute terms in the objective function and constraints. The modified model is named as model M2. Let us define

$$x^{+}_{injr} = \begin{cases} x_{in} - x_{jr} & \text{if } x_{in} \geqslant x_{jr} \\ 0 & \text{if } x_{in} < x_{jr} \end{cases} \tag{9}$$

$$x^{-}_{injr} = \begin{cases} -(x_{in} - x_{jr}) & \text{if } x_{in} \leqslant x_{jr} \\ 0 & \text{if } x_{in} > x_{jr} \end{cases} \tag{10}$$

$$\beta_{injr} = \begin{cases} 1 & \text{if } x_{in} < x_{jr} \\ 0 & \text{if } x_{in} \geqslant x_{jr} \end{cases} \tag{11}$$

$$y^{+}_{injr} = \begin{cases} y_{in} - y_{jr} & \text{if } y_{in} \geqslant y_{jr} \\ 0 & \text{if } y_{in} < y_{jr} \end{cases} \tag{12}$$

$$y^{-}_{injr} = \begin{cases} -(y_{in} - y_{jr}) & \text{if } y_{in} \leqslant y_{jr} \\ 0 & \text{if } y_{in} > y_{jr} \end{cases} \tag{13}$$

$$\gamma_{injr} = \begin{cases} 1 & \text{if } y_{in} < y_{jr} \\ 0 & \text{if } y_{in} \geqslant y_{jr} \end{cases} \tag{14}$$

By substituting the variables defined above in the model M1, the following mixed-integer nonlinear programming model is derived:

$$\text{Min}: \sum_{p=1}^{P} \sum_{i=1}^{I} \sum_{j=1(j \neq i)}^{I} \sum_{n=1}^{N_i} \sum_{r=1}^{N_j} a^{p}_{ij} D_p Z_{inp} Z_{jrp} (x^{+}_{injr} + x^{-}_{injr} + y^{+}_{injr} + y^{-}_{injr}) \tag{15}$$

Subject to:

$$\sum_{n=1}^{N_i} Z_{inp} = b_{ip}; \qquad i = 1, \ldots, I, \quad p = 1, \ldots, P \tag{16}$$

$$\sum_{p=1}^{P} Z_{inp} t_{ip} D_p \leqslant C_i; \qquad i = 1, \ldots, I, \quad n = 1, \ldots, N_i \tag{17}$$

$$(x_{in} - x_{jr}) + M\alpha_{injr} + M\beta_{injr} \geqslant .5 * (l_i + l_j) + d_{ij} \tag{18}$$

$$-(x_{in} - x_{jr}) + M\alpha_{injr} + M(1 - \beta_{injr}) \geqslant .5 * (l_i + l_j) + d_{ij} \tag{19}$$

$$x_{in} - x_{jr} = x^{+}_{injr} - x^{-}_{injr} \tag{20}$$

$$(y_{in} - y_{jr}) + M(1 - \alpha_{injr}) + M\gamma_{injr} \geqslant .5 * (b_i + b_j) + d_{ij} \tag{21}$$

$$-(y_{in} - y_{jr}) + M(1 - \alpha_{injr}) + M(1 - \gamma_{injr}) \geqslant .5 * (b_i + b_j) + d_{ij} \tag{22}$$

$$y_{in} - y_{jr} = y^{+}_{injr} - y^{-}_{injr} \tag{23}$$

$$0 \leqslant -(x_{in} - x_{jr}) \leqslant h + M\beta_{injr} \tag{24}$$

$$0 \leqslant -(x_{in} - x_{jr}) \leqslant h + M(1 - \beta_{injr}) \tag{25}$$

$$0 \leqslant (y_{in} - y_{jr}) \leqslant v + M\gamma_{injr} \tag{26}$$

$$0 \leqslant -(x_{in} - x_{jr}) \leqslant h + M(1 - \beta_{injr}) \tag{27}$$

$$\alpha_{injr}, \beta_{injr}, \gamma_{injr} \in \{0, 1\} \quad i = 1, \ldots, I, \quad j = i + 1, \ldots, I, \quad n = 1, \ldots, N_i, \quad r = 1, \ldots, N_j \tag{28}$$

There are two difficulties to obtain the optimum solution of the model M2. These difficulties are: (i) nonlinearity of the objective function, (ii) the large number of binary variables for a real-sized problem.

The nonlinearity of model M2 is due to the multiplication of binary variables Z in the objective function. To rectify the nonlinear term, all the possible combinations of values of these variables are enumerated. Since these values only affect constraints (16) and (17), those combinations which satisfy these constraints are considered and the rest are eliminated from further consideration.

The constraints (16) and (17) largely reduce the number of feasible combination of values. Therefore, for each feasible combination of values of variables Z, a mixed-integer linear programming model is resulted. The branch-and-bound technique is then applied to the resulted linear models to obtain the optimal solution of model M2.

4. Branch-and-bound approach

An algorithm based on branch-and-bound approach is proposed to solve the linear mixed-integer models. Elements of the proposed branch-and-bound approach are discussed in the following sections.

A simple example called as Example 1 is considered to describe the proposed approach. Let us consider layout of three machine types to be arranged in a two-dimensional area. It is assumed that there is only one product with a demand of five parts per day. There is only a single machine of each type. The dimensions and capacity of machines and the processing times are given in Table 1. The minimum clearance required between machines 1–2 is 4 m and it is $2m$ between machines 1–3 and 2–3. The product initially visits machine 1, then goes to machines 2 and 3, respectively.

4.1. Initialization

The procedure is initiated by relaxing constraint (28), and substituting them with the following constraints:

$$0 \leqslant \alpha_{injr} \leqslant 1 \tag{29}$$

$$0 \leqslant \beta_{injr} \leqslant 1 \tag{30}$$

$$0 \leqslant \gamma_{injr} \leqslant 1 \quad i = 1, \ldots, I; \quad j = 1, \ldots, I; \quad n = 1, \ldots, N_i; \quad r = 1, \ldots, N_j \tag{31}$$

$UB = \infty$ and the lower bound of new model, $LB_0 = 0$.

4.2. Branching

As one can see in constraints 18, 19, 21 and 22, there are two integer variables in each constraint. Thus, every node must be branched to four new nodes. In each step, the node with minimum LB is selected. This node is then branched to four nodes. Out of four nodes, two nodes are with:

$$\alpha_{injr} \geqslant 1 \tag{32}$$

and other two nodes with:

$$\alpha_{injr} \leqslant 0 \tag{33}$$

The first two branches satisfy constraints (21) and (22) and ignore constraints (18) and (19). Therefore, there is no need to consider variable β_{injr} since it appears in constraints (18) and (19). These two branches are distinguished by second new constraints as:

$$\gamma_{injr} \leqslant 0 \tag{34}$$

and

$$\gamma_{injr} \geqslant 1 \tag{35}$$

The second two branches satisfy constraints (18) and (19) and ignore constraints (21) and (22). Thus, there is no need to consider variable γ_{injr} since it appears in constraints (21) and (22). These two branches are distinguished by second new constraints as:

$$\beta_{injr} \leqslant 0 \tag{36}$$

Table 1
The problem data of Example 1

Machine type	Length (m)	Width (m)	Processing time (min)	Capacity (min/day)
1	6	6	5	300
2	4	6	5	300
3	8	4	3	400

and

$$\beta_{injr} \geqslant 1 \tag{37}$$

In the example, the branching is initiated with $(\alpha_{1121} \leqslant 0, \beta_{1121} \leqslant 0)$, $(\alpha_{1121} \leqslant 0, \beta_{1121} \geqslant 1)$, $(\alpha_{1121} \leqslant 0, \gamma_{1121} \leqslant 0)$ and $(\alpha_{1121} \leqslant 0, \gamma_{1121} \geqslant 1)$. As shown in Fig. 1, the initial branching corresponds to branch $k = 1(\alpha_{1121} = 0, \beta_{1121} = 0)$, branch $k = 2(\alpha_{1121} = 0, \beta_{1121} = 1)$, branch $k = 3(\alpha_{1121} = 1, \gamma_{1121} = 0)$ and branch $k = 4(\alpha_{1121} = 1, \gamma_{1121} = 1)$. Each branch is a simple linear model which can be solved optimally by the Simplex method.

4.3. Labeling

For each of the four new branches the label (lower bound) LB_k is determined by solving the model with the new constraints associated with each branch. As the branching goes deeper, the model is further restricted and thus lower bound for each node increases. This fact is obviously clear from Fig. 1. The node with $k = 0$ has $LB_k = 0$. In this case, all variables α,β,γ are continuous variables within zero and one. While branches go deeper in the search tree, the lower bound LB_k increases accordingly.

4.4. Setting bounds

Branch k is bounded if:

(1) lower bound LB_k is greater than UB (e.g. node $k = 6$ is bounded since UB has already been updated to 85 at node $k = 13$),
(2) there is no feasible solution (e.g. at node $k = 3$),

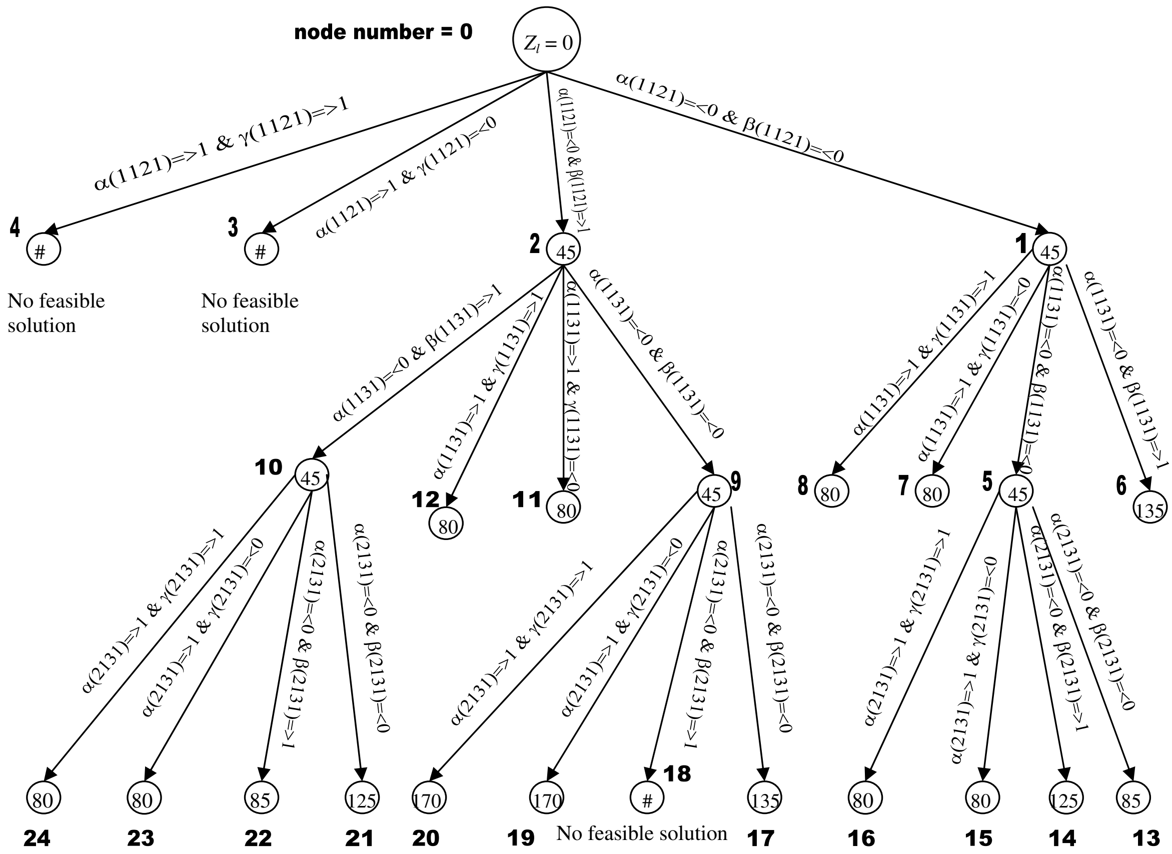

Fig. 1. Branch-and-bound tree search of example problem.

(3) a feasible (integer) solution is obtained (e.g. at node $k = 13$). In this case, it is set as $LB_k = UB$.
The proposed branch and bound algorithm terminates when all nodes are fathomed. Obviously, the final solution of current sub-problem will be the solution corresponding to the last UB.

4.5. Branch selection

Considering all nodes in each step, the one with smallest LB is selected. The information for other branches is recorded. If a branch is bounded, the jump tracking procedure is launched to find the next node with the lowest LB.

The lower bound of branch $k = 1$ is $LB_1 = 45$, for $k = 2$ is $LB_2 = 45$, for $k = 3$ is $LB_3 = \infty$ and for $k = 4$ is $LB_4 = \infty$. Thus, branch $k = 1$ and $k = 2$ are selected (Fig. 1). The branching process is continued with $LB_1 = 45$.

4.6. Updating of the upper bound

When all variables α,β,γ take integer values (0 or 1) the solution obtained is a feasible complete solution to the facility layout problem. If the value of the objective function for this solution, LB_k, is less than UB (the current best value of the objective function), then UB is updated, i.e. $UB = LB_k$.

For the example (at node $k = 13$), $LB = 85$, all variables α,β,γ have integer values and hence, this is a feasible solution. Up to this point, the UB was equal to ∞, and hereafter, the UB is updated to be equal to 85.

4.7. Jump tracking

This procedure is used when a node is branched and a new lower bound value is obtained. The jump tracking procedure sorts the LB for all obtained nodes and selects the lowest of them. If there are two or more nodes with lowest value of LB, then the deepest node would be selected. If there are more than one branch with lowest LB in the same level the one with smaller node number is selected. Referring to Fig. 1, the nodes 2 and 3 have the same LB, and hence node 2 is branched prior to node $k = 3$.

4.8. Final solution

When the algorithm based on branch-and-bound approach is applied to all feasible sets of variables Z the solution with lowest UB is selected as the final solution. There might be several solutions with lowest UB. In this case, the layout designer can consider the real aspects not captured in the model, and then select the most practically applicable solution. The optimal layout of facilities for the example problem is shown in Fig. 2, which corresponds to node 15 in Fig. 1.

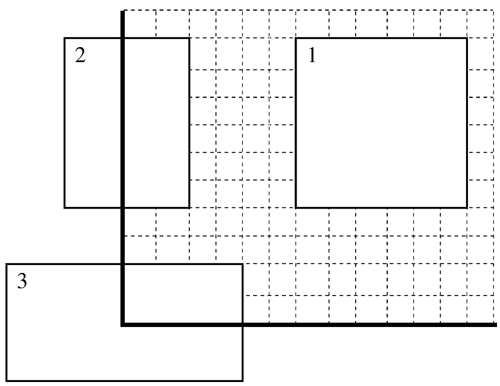

Fig. 2. The final layout obtained by the proposed method for the illustrative example.

5. Flowchart

The steps of the proposed approach are illustrated in Fig. 3. The first step is to find all possible sets of variables Z, satisfying constraints (16) and (17). For the illustrative example, the only feasible values for variables Z are $Z_{111} = 1$, $Z_{121} = 1$, $Z_{131} = 1$. By fixing variables Z, the problem becomes a linear mixed-integer model instead of a nonlinear one and can be solved with the proposed branch-and-bound algorithm. Therefore, corresponding to each set of feasible values for variables Z, there will be a linear mixed-integer model. The next step is to determine the optimum location of each machine (variables x_{in}, y_{in}). The optimum location of centroids of machines in Example 1 is $(x_1, y_1) = (9, 7)$ for machine 1, $(x_2, y_2) = (0, 7)$ for machine 2, and $(x_3, y_3) = (0, 0)$ for machine 3 as shown in Fig. 2.

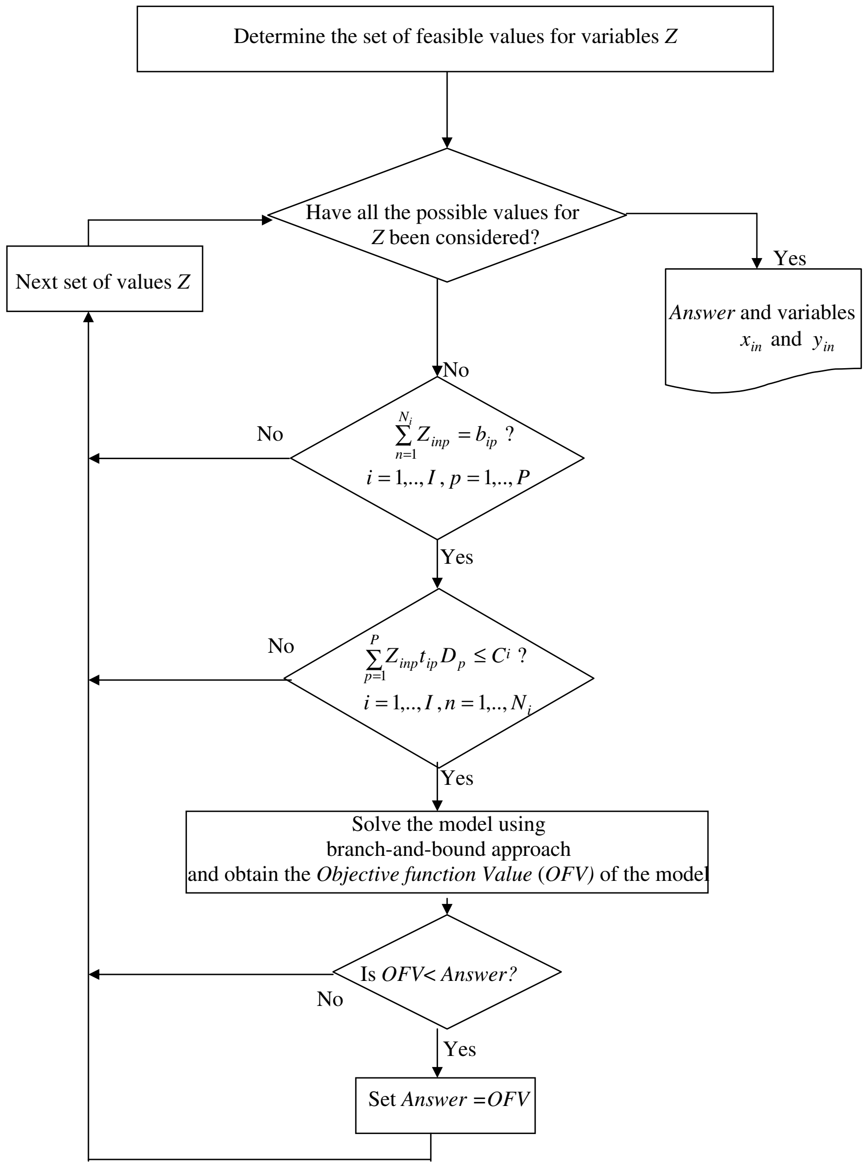

Fig. 3. Flowchart of the proposed algorithm.

6. Computational results

The proposed algorithm has been coded in C++ and run on a PC with a Pentium IV, 2.4 GHz processor. In programming the model, all feasible sets of variables Z are found. Each feasible set of variables Z is then exported to a program coded in LINGO Software to solve the associated linear programming model. The solution obtained from LINGO is then exported to the original program coded in C++ to proceed the branching and bounding process discussed in Section 4.

In this section, small-, medium-, and large-sized problems are solved using the proposed approach to evaluate the effectiveness and efficiency of the proposed approach. The results are compared with the arrangement of machines based on process layouts.

6.1. Small problem

Suppose four types of machines are to be laid out in a manufacturing system; the type L which includes two lathe machines, the type M which includes three milling machines, the type G which includes one grinding machine, and finally the type A which includes three assembling machines. Three products A, B, and C must be processed in this system. Each product has its own process route. The product A must be processed on the lathe machine, milling machine, grinding machine and assembly machine, respectively. The route of product B includes lathe machine, grinding machine and assembly machine, respectively. The product C must be processed on the milling machine followed by the lathe machine and then is processed on assembly machine. The daily demand for products A, B, and C is 15, 15, and 17, respectively.

Dimensions of each type of machines is presented in Table 2. The processing time of each product on each machine type is presented in Table 3. The minimum distance required between different machine types is presented in Table 4.

This problem has been solved by the proposed approach and the optimum coordinates of centroids of machines are shown in Table 5. Fig. 4 shows the layout of machines based on the optimum solution obtained

Table 2
Dimensions of machines in Example 2

Machine type	Length (m)	Width (m)
Lathe machine	6	2
Milling machine	2	2
Grinding machine	5	2
Assembly machine	5	4

Table 3
Processing time of each product on each machine type in Example 2 (min)

Product machine type	A	B	C
Lathe machine	5	5	5
Milling machine	5	5	2
Grinding machine	3	5	3
Assembly machine	3	3	3

Table 4
The minimum distance needed between the machines in the Example 2 (m)

	Lathe	Milling	Grinding	Assembly
Lathe	0	4	2	2
Milling	4	0	2	2
Grinding	2	2	0	2
Assembly	2	2	2	0

Table 5
Coordinates of the centroids of machines of Example 2 obtained by the proposed method

Machine	X (m)	Y (m)
First assembly machine	0	4
Second assembly machine	13.5	9
Third assembly machine	21	1
First lathe machine	5.5	10
Second lathe machine	13.5	0
First milling machine	5.5	4
Second milling machine	21	6
Third milling machine	21	10
Grinding machine	13.5	4

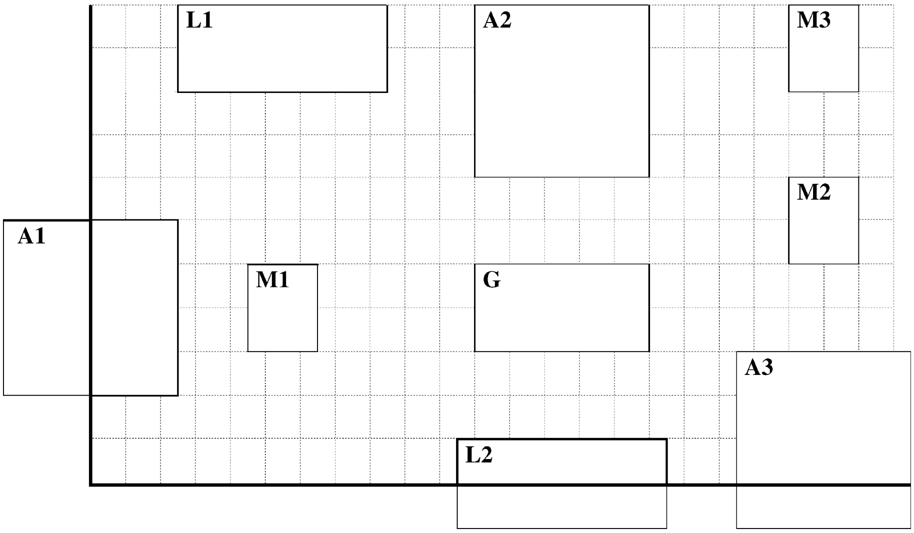

Fig. 4. The global optimum layout obtained by the proposed method for Example 2.

by the proposed algorithm. The solution (optimum values obtained for variables Z) obtained for this example indicates that one of the assembly machines and two milling machines are not visited by any product. This is because of the fact that these machines are redundant. Since the capacity of milling machine and two assembly machines is sufficient to produce the amount of interest for each product, the model has decided to put the rest of these machines in the rightmost side of Fig. 4. Therefore, the proposed model has the advantage to identify the minimum number of machines of each type required.

Fig. 5 shows the optimum arrangement of these machines based on a process layout configuration. The total distance traveled by the products in the layout obtained by the proposed model (Fig. 4) is 615.5 m per day and it is 1057.5 m per day for the process layout (Fig. 5). Therefore, the proposed model has reduced the total distance traveled by the products about 41.8%.

6.2. Medium- and large-sized problems

The raw data for a medium-sized problem are presented in Tables 6 and 7. This problem was solved by the proposed approach. The solution process was interrupted several times to study the reduction in the objective function. The objective function value after 1, 4, 8, 16, and 24 h, was recorded as 3421, 3421, 2829, 1834, and 1012, respectively. The proposed algorithm obtained the optimum solution after 24 h. The objective function value (total distance) for a process layout was 1834. Therefore, the proposed algorithm has reduced the total distance traveled daily by the materials by 44.8% within 24 h computation. Although this time is too much, it

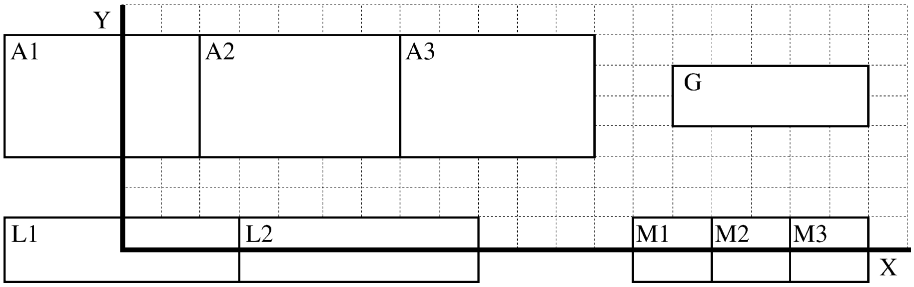

Fig. 5. The process layout of Example 2.

Table 6
Medium-sized problem data

Machine type	Number of machines	Dimension (X)	Dimension (Y)
1	2	2	3
2	1	2	3
3	1	2	3
4	2	2	3
5	3	2	2
6	1	2	1
7	1	2	2
8	2	1	3
9	1	2	1
10	1	2	1

Table 7
The demand and process route of each product type for medium-sized problem

Product type	Demand	Required sequence
1	15	1–2–4–8–10
2	13	5–1–4
3	12	3–6–7–9
4	11	2–1–3
5	14	4–5–10

is notable that the layout of machines is not a frequent problem and it may arise for a company once at the beginning or when reconfiguration of manufacturing facilities is needed. On the other hand, the 44.8% reduction in total distance is attained everyday and the company can be benefited from this reduction daily for its life time. Hence, it can be judged that the performance of the proposed approach for medium-sized problems is acceptable.

Tables 8 and 9 show the data for a large-sized problem. The solution process for this problem was interrupted after 1, 4, 8, 16, and 24 h. The solution process was terminated after 24 h before its completion. Even after four hours, the proposed approach could not obtain a feasible solution for this problem. After 8, 16, and 24 h, the objective function value yet obtained by the proposed approach was 11345, 11345, and 9187, respectively. This value for a process layout, however, was 3467. Hence, it may be stated that the proposed approach can not be used for large-sized problems.

7. Conclusion

In this paper, a new nonlinear mixed-integer programming model is presented for the facility layout problem in a two-dimensional area to minimize the total distance traveled by the material in the shop floor. A technique is used to linearize the proposed model. An algorithm based on branch-and-bound approach is

Table 8
Large-sized problem data

Machine type	Number of machines	Dimension (X)	Dimension (Y)
1	2	2	3
2	1	2	3
3	1	2	3
4	2	2	3
5	3	2	2
6	1	2	1
7	1	2	2
8	2	1	3
9	1	2	1
10	1	2	1
11	1	2	1
12	1	2	2
13	1	2	1
14	2	3	1
15	2	2	1
16	3	2	2
17	1	1	1
18	1	1	1
19	2	2	1
20	1	1	2

Table 9
The demand and process route of each product type for large-sized problem

Product type	Demand	Required sequence
1	15	11–2–4–8–10
2	13	5–12–4
3	12	3–16–7–19
4	11	2–1–3
5	14	4–5–10
6	11	13–14–15
7	12	20–18
8	14	7–11–17

developed to optimally solve the proposed mathematical programming model. An illustrative example is solved and then the obtained result is compared to the configuration obtained from the process layout with respect to the total material handling distance. It is shown that the presented model decreases the total distance traveled by the products about 41.8% for small-sized and about 44.8% for medium-sized problems as compared to the process layouts for the example problems. Since the process layouts are special cases of the general two-dimensional layout (TDL) modeled in this paper and with respect to the fact that the proposed branch-and-bound based algorithm determines the optimal solution of TDL, obviously the solutions obtained by the proposed algorithm will be better than or at least as good as the configurations arranged based on process layouts.

The computational results reveal that the proposed approach can not be used for large-sized problems. Since the solution approach is an exact method, this finding however was already expected. The exact algorithms can not be applied for solving large-sized combinatorial optimization problems in a reasonable time. It is therefore a scope for future researches to develop heuristic and meta-heuristic methods for solving the proposed mathematical programming model. In the case of single row layout, the proposed branch-and-bound algorithm will solve the equivalent of Love and Wong's (1976b) model, which was shown to be inferior to the model presented in Amaral (2006).

The proposed model can also be applied to determine the processing route of products in an existing layout. When the location of each machine in an existing manufacturing system is known, then the variables x_{in} and

y_{in} will be fixed in Eq. (1) and the model can be solved to obtain variables Z. These variables determine the processing routes of each product so that the total distance is minimized.

Acknowledgements

The authors express their sincere thanks to the anonymous referees for their valuable time and constructive comments which improved the quality and scientific level of the paper.

References

Amaral, A. R. S. (2006). On the exact solution of a facility layout problem. *European Journal of Operational Research, 173*(2), 508–518.

Bazzara, M. S., & Sherali, M. D. (1980). Benders' partitioning scheme applied to a new formulation of quadratic assignment problem. *Naval Research Logistics Quartery, 27*(1), 29–41.

Bukard, R. E., & Bonniger, T. (1983). A heuristic for quadratic Boolean program with application to quadratic assignment problems. *European Journal of operational Research, 13*, 347–386.

Heragu, S. S., & Alfa, A. S. (2003). Experimental analysis of simulated annealing based algorithms for the layout problem. *European Journal of Operational Research, 57*(2), 190–202.

Heragu, S. S., & Kusiak, A. (1988). Machine layout problem in flexible manufacturing systems. *Operations Research, 36*(2), 258–268.

Heragu, S. S., & Kusiak, A. (1991). Efficient models for the facility layout problem. *Operations Research, 53*, 1–13.

Kaku, B. K., & Thompson, G. L. (1986). An exact algorithm for the general quadratic assignment problem. *European Journal of Operational Research, 23*, 382–390.

Konak, A., Kulturel-Konak, S., & Bryan, A. (2004). A new mixed-integer programming formulation for facility layout design using flexible bays. *Operations Research Letters, 34*(6), 660–672.

Kusiak, A., & Heragu, S. S. (1987). The facility layout problem. *European Journal of Operational Research, 29*(3), 229–253.

Kusiak, A. (1990). *Intelligent manufacturing systems*. New Jersey: Prentice-Hall International Inc..

Lawer, E. L. (1963). The quadratic assignment problem. *Management Science, 9*(4), 586–599.

Love, R. F., & Wong, J. Y. (1976a). Solving quadratic assignment problems with rectilinear distances and integer programming. *Naval Research Logistics Quarterly, 23*, 623–627.

Love, R. F., & Wong, J. Y. (1976b). On solving a one-dimensional space allocation problem with integer programming. *INFOR, 14*(2), 139–143.

Meller, R. D., Narayanan, V., & Vance, P. H. (1998). Optimal facility layout design. *Operations Research Letters, 23*, 117–127.

Mir, M., & Imam, M. H. (2001). A hybrid optimization approach for layout design of unequal-area facilities. *Computers & Industrial Engineering, 39*(1-2), 49–63.

Montreuil, B. (1990). A modeling framework for integrating layout design and flow network design. In *Material handling research colloquium*. Hebron, KY.

Sherali, H. D., Fraticelli, B. M. P., & Meller, R. D. (2003). Enhanced model formulations for optimal facility layout. *Operations Research, 51*, 629–644.

Solimanpur, M., Vrat, P., & Shankar, R. (2005). An ant algorithm for the single row layout problem in flexible manufacturing systems. *Computers & Operations Research, 3*(32), 583–598.

Tavakkoli-Moghaddam, R., Javadian, N., Javadi, B., & Safaei, N. (2007). Design of a facility layout problem in cellular manufacturing systems with stochastic demands. *Applied Mathematics and Computation, 184*(2), 721–728.

Urban, T. L., Chiang, W. C., & Russel, R. A. (2000). The integrated machine allocation and layout problem. *International Journal of Production Research, 38*(13), 2911–2930.