Optimization of FTL Layout Design
Through an Asymmetrical and Restricted Plant Using GA*

Jaber ABU QUDEIRI** and Hidehiko YAMAMOTO**

**Intelligent Manufacturing Systems Laboratory, Gifu University
1-1 Yanagido, Gifu Shi, 501-1193, Japan
k3812203@guedu.cc.gifu-u.ac.jp

Abstract
One of the problems encountered in the design and implementation of a flexible transfer line (FTL) is the layout of the FTL in a restricted area. The layout of the FTL has an important impact on production cost. In this paper we propose efficient FTL layout design procedures for a FTL layout in an asymmetrical and restricted plant area. In order to find the layout of the FTL including the buffer size between each pair of FTL machines, an efficient FTL layout design procedure called a One by One Layout Method (OOLM) in conjunction with genetic algorithm (GA) is proposed. The OOLM generates an efficient solution for a set of irregularly shaped machines through a restricted plant area. A CAD system is linked to the OOLM to draw the FTL layout. The OOLM is not limited to a single static environment, but is highly flexible within the plant structure. An application example was developed, and after a number of operations based on OOLM, an efficient FTL layout design could be found.

Key words: Flexible Transfer Line, Layout Design, Genetic Algorithm, Buffer Size, CAD.

1. Introduction

A Flexible transfer line (FTL) is a production system consisting of a series of machines separated by buffers. One of the problems encountered in the design and implementation of a FTL is the layout of the FTL in a restricted area. The layout of the FTL has an important impact on material handling. The efficient layout design of the FTL can reduce the cost of material handling by at least 10-30%. Depending on the production system, between 15% - 70% of the total production cost can be attributed to material handling(1). Thus, the FTL layout is important to reduce production cost. The FTL layout problem is identified as a NP-complete problem(2), and many heuristic approaches have been developed to solve this problem for near optimum solution(3), (4). A simulated annealing algorithm to solve the single row layout problem was proposed by Heragu and Alfi(5). Solimanpur(6) developed an ant algorithm to solve the single row machine layout problem. Kumar and Hadjinicola(7) developed a constructive greedy heuristic that assigns facilities with the largest number of moves between them to adjacent locations on a single row line, and the dimension of the machines is not considered in the construction of the line. Bragila(8) proposed a combination of genetic algorithms and simulate annealing to minimize the total backtracking in the linear ordering of the machines. Lee and Kim(9) have proposed several procedures for transforming an irregular grid-based block layout into a regular one. Yang and Peters(10) developed a sophisticated approach assuming a continuous layout by mapping the area to be

*Received 23 Aug., 2006 (No. 06-0095)
[DOI: 10.1299/jmmp.1.81]
arranged as rectangularly shaped elements. Most approaches proposed in the literature consider machines as equal blocks and neglect operational details. A detailed layout planning was proposed by Yang et al. (11), but they considered an open plan without restrictions and regular machine shape. Bock and Hoberg (12) developed detailed layout planning for irregularly-shaped machines with a transportation path design. However, despite the many existing methodologies regarding the FTL layout problem, almost all approaches studying this problem neglect the important operational details such as the buffer size between the machines and restricted areas in the plant area.

In this paper operational details including the movement of the products between machines are considered, in addition to the restricted areas in the site. In order to find a FTL layout including machines layout and a material handling path between each pair of machines, an efficient FTL layout design procedure called a One by One Layout Method (OOLM) in conjunction with genetic algorithm (GA) is proposed. The OOLM generates an efficient solution for a set of irregularly shaped machines through a restricted plant area. The OOLM is not limited to a single static environment, but is highly flexible, within the plant structure. To efficiently solve the FTL layout problem using the proposed OOLM, a new GA mutation and crossover operators are proposed. A CAD system is linked to the proposed OOLM to draw the FTL layout.

2. Problem Description

Some problems involving a new FTL in any plant are restrictions and area limitations. The problem can be described mathematically as follows:

**Given:**
- A set of $K$ irregular-shapes machines and their dimensions;
- Spaces and machine allocation limitations;
- The plant restrictions, such as plant columns, walls, and any other restrictions in the plant area;
- The buffer size vector $[B_1, B_2, \ldots, B_{K-1}]$.

**Determine:** FTL layout design. So as to minimize $A_F$

With the following condition

$A_F \leq A_P$ and $D_k$ is located at or is close to the FTL end point.

Where

- $A_F$: Required area needed to create the new FTL layout.
- $A_P$: Available plant area.
- $D_k$: Drop point of the last machine.

3. OOLM Techniques

The OOLM is used to layout the FTL components between a given start and end points of a FTL. The location of the next component of FTL depends on the current component location; the next component can be located either up, down, right, or left of the current component. For example, if $i$ and $i+1$ are two consecutive components in the FTL path, the OOLM works as follows: The component $i+1$ is located either up, down, right, or left of component $i$. Choose one of the four possible locations to locate component $i+1$, for example, up is selected, now, If the location up to the component $i$ is empty Then, component $i+1$ is located up from the component $i$; Else choose one of the other three possible locations (down, right, or left) and apply the If-Then-Else rule again, once Then of the If-Then-Else rule is satisfied, move to the next component. If Then of the If-Then-Else rule is not satisfied for the four possible locations then return back to component $i$ and change its current location up to one of the other three possible locations. Continue the If-Then-Else rule until the FTL path ends at the last machine drop point.

4. FTL Model Assumptions

The FTL model has the following constraints.
1. The machines in the FTL have irregular shapes and different dimensions following current industrial practices. The machine is defined by its overall length \( (L_i) \) and overall width \( (W_i) \). The distance between each pair of machines is calculated with respect to their pick and drop points.

2. Each machine has a pick point, \( P_i \), and a machine drop point, \( D_i \), as shown in Fig. 1. The \( P_i \) is the point where the parts enter machine \( i \). \( D_i \) is the point where the parts leave machine \( i \).

3. The clearance between each pair of machines is not fixed but depends on the buffer size, \( B_{ij} \), between each pair of machines, where \( B_{ij} \) is the buffer size between machines \( M_i \) and \( M_{i+1} \). This assumption is important in real life manufacturing to temporarily store the parts waiting for processing for the next machine. The main factors effecting the clearance spaces required between the machines are the machining times of operations on machines and the shape and size of the buffer spaces path between each pair of machines.

4. Each machine, except the last one in the FTL, is followed by buffer spaces; each buffer space is represented by a square with constant dimensions.

5. The pick point of the first machine is located at the FTL start point.

6. The drop point of the last machine is located at the FTL end point. The drop point of the last machine means the point that the products leave the last machine in FTL.

5. **Plant Layout Model**

Following Bock and Hoberg\(^{(12)}\), the plant in this study is defined as uniform squares connected together and denoted in this paper as cells as shown in Fig. 2. The restricted areas (obstacles) including plant columns and any other obstacles are indicated on the plant layout. The start point and the end point of the FTL locations are given and are represented on the plant layout. The plant model can be defined by using the following steps.

**Step 1:** Draw the smallest rectangle that passes through the outside walls to include all plant...
facilities and sub walls. The area covered by this rectangle is denoted in this paper as the extended plant area.

**Step 2:** Divide the extended plant area into uniform cells that are equal to each other and connected to each other.

**Step 3:** Define a matrix that represents the extended plant layout area as shown in Eq. (1),

\[
\begin{bmatrix}
C(i,1) & C(i,2) & \cdots & C(i,j) & \cdots & C(i,n-1) & C(i,n) \\
C(2,1) & C(2,2) & \cdots & \cdots & \cdots & C(2,n-1) & C(2,n) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
C(i,1) & C(i,2) & \cdots & C(i,j) & \cdots & C(i,n-1) & C(i,n) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
C(m-1,1) & C(m-1,2) & \cdots & C(m-1,j) & \cdots & C(m,n-1) & C(m,n) \\
C(m,1) & C(m,2) & \cdots & C(m,j) & \cdots & C(m,n-1) & C(m,n) \\
\end{bmatrix}
\]

(1)

where \(C(i,j)\) represents the cell located in row \(i\) and column \(j\).

### 6. OOLM Procedure

In order to find an efficient layout of the FTL components through the available plant area, a GA is included into the OOLM. The OOLM is conducted using the following algorithm. Before describing the algorithm, the notations and terms used in the algorithm are defined.

**Notations**

- \(K\): Number of machines in the FTL.
- \(B_i\): Buffer allocation between machines \(i\) and \(i+1\).
- \(S_i^c\): Side length of each cell.
- \(L_i\): Overall machine length.
- \(W_i\): Overall machine width.
- \(L\): Overall plant length.
- \(W\): Overall plant width.
- \(L_i\): Side length of the buffer space.
- \(Blocked-cells(i)\): Number of cells included in blocked area \(i\).
- \(Machine-cells(i)\): Number of cells included in machine layout \(i\).
- \(Blocked-areas\): Number of blocked areas throughout the extended area.
- \(i_p^c\): Cell through the plant area that represents the FTL start point.
- \(i_d^c\): Cell through the plant area that represents the FTL end point.
- \(c_j^i\): Location of cell \(j\) in machine \(i\) cell map relative to the machine pick point.
- \(c_b^i\): Location of cell \(k\) in blocked area \(i\).
- \(c_p^i\): Cell location of the machine \(i\) pick point through the plant area.
- \(c_d^i\): Cell location of the machine \(i\) drop point through the plant area.

**Definitions**

[Definition 1]: Machine pick point, \(P_i^c\): The machine pick point is the point where the parts input into the machine \(i\).

[Definition 2]: Machine drop point, \(D_i^c\): The machine drop point is the point where the parts leave machine \(i\).

[Definition 3]: Machine cells group, \(M_{cells-group}^i\): The machine cells group is the set of cells that has been mapped to machine \(i\) relative to the \(P_i\) cell.

[Definition 4]: Buffer-cell ratio, \(B_{m-c}\): The buffer–cell ratio is the smallest integer number greater than or equal to the division of the buffer side length to the cell length; \(B_{m-c}\) can be calculated by using Eq. (2).

\[
B_{m-c} = \frac{l}{S_i^c}
\]

(2)

[Definition 5]: FTL Components: FTL components include machines and buffer spaces.

[Definition 6]: FTL Component location, \(COM_{location}^i\): The component location is the location of the next component with respect to the current component in the FTL. The component can be located in one of four locations with respect to the current component (up, down, right or left).
[Definition 7]: Extended plant area, $A_{plant-ext}$: The extended plant area is the area of the rectangle that passes through some of the plant outside wall and includes all plant facilities and sub walls.

[Definition 8]: Blocked area: The blocked area is the area in the plant that has been restricted by some obstacles such as plant columns, partitions, and sub areas used by built stations, etc. The blocked areas are not usable for allocation of machines or buffers. In addition to the blocked areas in the plant area, the areas in the extended area that are not included within the plant area are assumed to be blocked areas.

[Definition 9]: Block area cells group, $B_{cells-group}^i$: The blocked area cells group is the set of cells that have been mapped to the blocked area $i$.

[Definition 10]: Machine cells map, $M_{map}^i$: The machine cells map is the set of cells that have been mapped to the machine layout. The location of each cell in the machine map is defined relative to the machine pick point.

[Definition 11]: Machine drop point possible locations, $C_{drop}^i(j)$: The machine drop point possible locations are the locations of the machine drop points with respect to a given machine pick point. Follows(13), the machine can be located at $0, 90, 180$ or $270$, thus $1 \leq j \leq 4$. To describe the machine drop point’s possible locations, the following example is introduced.

Example: Based on machine $i$ with its pick and drop points as shown in Fig. 3.

![Fig. 3  Machine model](image)

If the machine pick point is located at cell $(I,J)$ then the location of the drop point possible locations are as follows:

- $(I,J + 3)$ if $M_{map}^i \not\in B_{cells-group}^k \forall k = 1 \ldots$ blocked - areas ,

- $(I,J - 3)$ if $M_{map}^i \not\in B_{cells-group}^k \forall k = 1 \ldots$ blocked - areas ,

- $(I + 1,J)$ if $M_{map}^i \not\in B_{cells-group}^k \forall k = 1 \ldots$ blocked - areas and

- $(I - 1,J)$ if $M_{map}^i \not\in B_{cells-group}^k \forall k = 1 \ldots$ blocked - areas

The possible locations of the drop point are cells 1, 2, 3 and 4 in Fig. 4.

![Fig. 4  Possible locations of machine drop point](image)

[Definition 12]: Buffer space possible locations, $B_{drop}^i(t)$: The buffer space $i$ possible locations are the locations of the buffer space $i$ with respect to the location of the component $i-1$.

To describe the buffer space possible locations, the following example is introduced.

Example: Consider machine $i$ in Fig. 3 and assume that the buffer size next to this machine is 5. The possible locations of all buffers are shown in Fig. 5 below. The possible locations of the first buffer space are the cells coded by the number 1, and the possible locations of the second buffer space are the cells that are coded by the number 2, and so on.

![Fig. 5. The possible locations of the buffers.](image)
[Definition 13] FTL area, $A_F$: The FTL area is the area used by the FTL components.

[Definition 14] Plant area, $A_P$: The plant area is the available factory area.

[OOLM algorithm]

Step 1: Read the matrix that represents the extended plant layout area as described in §5.

Step 2: Set $C(k) = C(i, j)$ for $k = 1, 2, ..., (m \times n)$, $i = 1, 2, ..., m$ and $j = 1, 2, ..., n$.

Step 3: Read the FTL start point. Set $C_{start} = c(i, j)$, where $i, j \in [1, ..., n]$ and $c(i, j) \in B_{\text{cells-groups}}^d \forall k = 1 \ldots \text{blocked - areas}$.

Step 4: Read the FTL end point. Set $C_{end} = c(i, j)$, where $i, j \in [1, ..., n]$ and $c(i, j) \in B_{\text{cells-groups}}^d \forall k = 1 \ldots \text{blocked - areas}$.

Step 5: Define the cells map for all machines, $M_{\text{cells-group}}^i$, relative to the $P_i$ cell.

Step 6: Define the drop cell for machine $i$, relative to the $P_i$ cell.

Step 7: Define all blocked areas cells group through the extended plant area, $B_{\text{cells-group}}^d$.

Step 8: Set the first machine pick point at the FTL start point. Set $P_1 = C_{start}$.

Step 9: Find all possible locations of the first machine drop points, $c_{d_1}$, as described in definition 11.

Step 10: Randomly select one of the possible locations of the machine drop points.

Step 11: Find all possible locations of the first buffer space with respect to the machine drop point as described in definition 12.

Step 12: Randomly select one of the possible locations of the first buffer space.

Step 13: Find all possible locations of the next buffer space with respect to the current buffer space location and randomly select one of these locations.

Step 14: Repeat step 13 to find one location for each space of buffer size.

Step 15: Find all possible locations of the next machine pick points, $c_{p_1}$, with respect to the last buffer space and randomly select one of these locations.

Step 16: Repeat steps 13 - 15 to find one of the possible locations of all machines and spaces of all buffers.

Step 17: Apply the GA operations to find the best locations for all of the FTL components.

Step 18: Calculate the FTL area by using Eq. (3).

$$A_F = \sum_{i=1}^{K} (L_i \times W_i) + \sum_{i=1}^{K-1} (B_i \times L_i)$$  \hspace{1cm} (3)

Step 19: Carry out the following rule.

If: the evaluate Eq. (4) is satisfied,

$$A_F - A_T \geq 0$$  \hspace{1cm} (4)

Then: draw the FTL.

Else: read a new buffer size vector $[B_1, B_2, ..., B_{K-1}]$.

7 Genetic Algorithms

A GA paradigm has been proposed to solve a wide range of problems \cite{14, 15, 16}. GA has been successfully applied to optimization problems in diverse fields. It differs from other search techniques which depend on a natural genetic evaluation process. GA starts with an initial set of solutions randomly selected and called population. A suitable encoding for each solution in the population is used to allow computation of fitness. The solution set in the population, called a chromosome or individual, represents a solution to the optimization problem. Each individual contains a number of genes. The individuals in the initial population are evaluated to measure its fitness. To create the next population, new individuals are formed by either merging two individuals from the current population using a crossover operator or modifying an individual solution using a mutation operator. Based on the individuals’ fitness, the individuals to be included in the next generation are then probabilistically selected from the set of individuals in
the current population. The iteration, called a generation will continue until fitness reaches its maximum value. The best overall solution becomes the candidate solution to the problem. The operations for our GA are described in §§ 7-1 ~ 7-5.

7.1 Encoding

One of the important jobs to use GA is how to express a chromosome. One of the main difficulties in encoding this problem is that each pair of components in the FTL is located relative to each other, which means that each component’s location should be specifically identified in the individual. This research adopts each component’s location in the FTL as the gene, and each machine in the FTL is represented by two genes: one for the machine pick point location and the other for the machine drop point location. The other component’s locations are represented by one gene for each component. Conventional GA operations are generally based on an individual with a similar gene size. In the case that we are studying here, i.e., an FTL layout, it is difficult to use a similar gene size in individuals. This is because the number of possible component locations is not equal for all components. The number of possible location for the next component is decided according to the state of locations up, down, left and right of the current component. For example, if component i is located at cell (I, J), then the number of possible locations, NPL, of component i+1 is as follows:

$$NPL = \begin{cases} 
0 & \text{if the constraints 1, 2, 3, and 4 are satisfied} \\
1 & \text{if three constraints from constraints 1, 2, 3, and 4 are satisfied} \\
2 & \text{if two constraints from constraints 1, 2, 3, and 4 are satisfied} \\
3 & \text{if one constraint from constraints 1, 2, 3, and 4 is satisfied} \\
4 & \text{if the constraints 1, 2, 3, and 4 are not satisfied} 
\end{cases}$$

(5)

Constrain 1: \((I, J + 1) \in \{ c_{b_i1}, c_{b_i2}, \ldots, c_{b_{i+1}} \}\) \(\forall i = 1,2,\ldots, \text{Blocked} - \text{area}\)

Constrain 2: \((I, J - 1) \in \{ c_{b_i1}, c_{b_i2}, \ldots, c_{b_{i+1}} \}\) \(\forall i = 1,2,\ldots, \text{Blocked} - \text{area}\)

Constrain 3: \((I + 1, J) \in \{ c_{b_i1}, c_{b_i2}, \ldots, c_{b_{i+1}} \}\) \(\forall i = 1,2,\ldots, \text{Blocked} - \text{area}\)

Constrain 4: \((I - 1, J) \in \{ c_{b_i1}, c_{b_i2}, \ldots, c_{b_{i+1}} \}\) \(\forall i = 1,2,\ldots, \text{Blocked} - \text{area}\)

In this research, we propose a new individual encoding method to express each individual. The new encode method is called a one to one encoding method (OOEM). The following paragraph describes how to use OOEM to encode components in the FTL. The first element in the individual represents the location of the drop point of the first machine in the FTL. The location of the first machine drop point can be determined with respect to the location of the first machine pick point as described in definition 11. Elements \(2, 3, \ldots, G_{B_{i+1}}\) in the individual represent the location of the spaces of the first buffer. The location of the first buffer space is determined with respect to the location of the machine drop point and the location of the next buffer space is determined with respect to the current buffer space location and so on. Figure 6 shows the encoding using the OOEM.

The number of genes in each individual, \(T\), is calculated using Eq. (6).

$$T = 2K - 1 + \sum_{i=1}^{K-1} B_i$$

(6)

The number of items in the individuals is not limited, which means that any production
line with any number of machines and any buffer size between each pair of machines can be dealt with.

7.2 Initial population

The initial population is randomly selected. The initial population contains \( N \) number of individuals. Each individual expresses a buffer size as shown in Fig. 6. The location of components should be specifically identified in the individual because each pair of components in the FTL is located relative to each other. The initial population is determined as follows:

Set \( \text{individual} (i) = [G_1, G_2, \ldots, G_{T-1}, G_T] \quad \forall i = 1, 2, \ldots, N \)

Each individual in the initial population is determined by using steps 8 ~ 16 in §6.

7.3 Crossover

The traditional crossover operation is not suitable for this type of representation because the genes are selected one by one. The encoding method to express each individual using OOEM is different from that which is obtained using conventional encoding methods. The crossover operations for our GA system are also different. The main difference between the OOEM crossover and the conventional methods crossover is that in the conventional method the genes after the crossover point are swapped between the two individuals without any constrains. In contrast, the genes after the crossover point are selected one by one to avoid restricted cells.

OOEM crossover selects the first cell location of one parent, and checks the cell location leaving that cell location in the second parent. If the cell location is empty, choose the cell of the second parent, otherwise choose the cell of the first parent. For example, consider the two parents as shown in Fig. 7. If the first individual is chosen as the template, child 1 can be generated using the following steps:

**Step 1:** Select cell \( U \) (the first cell location in the individual) as the first cell location of child 1.

**Step 2:** Find the edges after gene 1 in both parents with respect to the first cell location of child 1: (first cell location, \( U \)), \( \text{read as the cell up to the first cell location} \) and (first cell location, \( L \)), \( \text{read as the cell left to the first cell location} \); then, find the cell state of these two edges. If the cell located left to the first cell location is empty (neither restricted nor used by another component) then it will be chosen as the second gene for child 1, otherwise, the cell up to the first cell location will be chosen as the second gene for child 1. Assuming that the cell up to the first cell location is empty, we select \( U \) as the next gene of child 1.

**Step 3:** Find the edges after gene 2: (second cell location, \( L \)), and (second cell location, \( L \)). If the edges after gene 2 are the same, we select \( L \) as the next gene of child 1.

**Step 4:** Find the edges after gene 3: (third cell location, \( D \)), and (third cell location, \( U \)). Assuming that the cell up to the third cell location is not empty, we select \( D \) as the next gene of child 1.

Fig. 7  OOEM crossover operation
Step 5: Find the edges after gene 4: (fourth cell location, $R$), and (fourth cell location, $L$). Assuming that the cells to the right and left of the fourth cell location are not empty, we select $D$ as the next gene of child $l$. Note: that if the cell down from gene 4 is also not empty in this case, the previous gene (gene 3 in this example) is changed to the third possible choice which is $L$, and continues from this point again.

Step 6: Repeat the same rule in the previous step to generate all genes of child $l$.

We can use the same procedure to generate child 2 as shown in Fig. 7. After crossover, both offspring encode legal path.

7.3 Mutation

The mutation of our GA system is different from a traditional mutation operator because the gene expression adopts OOEM. Instead of using a traditional mutation operator, we randomly select two mutation points (two genes in one individual) and swap their values. After swapping the two genes, the cell locations of the two mutation points and of the genes between the two mutation points are tested, and if any of these cells belong to a block area cells group, the two mutation points are reselected. The new mutation points are selected between the two old mutation points. This process is repeated until the cells belonging to the genes between the two mutation points are not restricted. Thus, we still have a legal path after swap mutation. The mutation is carried out using the following steps.

Step 1: Select one individual randomly from the current population.

Step 2: Select two mutation points, $MP_1$ and $MP_2$, randomly.

Step 3: Swap the two mutation genes values.

Step 4: Check the cells belonging to the two mutation points and all cells belonging to the genes between the two mutation points, and if the cells of the two mutation points and the cells between them after being swapped are empty (neither restricted nor used by another component) then go to Step 7, otherwise go to Step 5.

Step 5: Select two new mutation points, but this time between the current mutation points.

Step 6: Go to Step 3.

Step 7: Accept the mutation.

The follow chart in Fig. 8 shows the mutation algorithm.

![Mutation Algorithm Chart]

7.4 Fitness

Our goal is to find the FTL layout, so that the FTL starts at a given point and ends at another point. As the end point of the FTL for one solution is closed to the given end point, the layout becomes more accurate. Thus, the fitness for each solution is the distance, $D$, between the last machine drop point of that solution and the FTL end point. The distance is given by using Eq. (7).

$$D = \sqrt{(x - x_e)^2 + (y - y_e)^2}$$

(7)
Where:
$(x_i, y_i)$ is the coordinate of the drop point of the last machine in an individual, and $(x_e, y_e)$ is the coordinate of the FTL end point.

8. Numerical Experiments

We applied the developed OOEM to some FTL examples as follows:

8.1 Applied FTL example 1: 4 machines and 3 buffers

The FTL we adopted in this example has 4 machines and 3 buffers. Each buffer space is assumed to be equal to the plant cells, $B_{m-c} = 1$. The machine shapes and specifications and the buffer size between each pair of machines are as shown in Fig. 9.

<table>
<thead>
<tr>
<th>Machine 1</th>
<th>Machine 2</th>
<th>Machine 3</th>
<th>Machine 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>p</td>
<td>p</td>
<td>d</td>
</tr>
<tr>
<td>d</td>
<td>d</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Buffer 1</td>
<td>Buffer 2</td>
<td>Buffer 3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 9  Machines shapes and specifications and buffer size

The FTL layout resulted by the OOLM is as shown in Fig. 10.

8.2 Applied FTL example 2: 10 machines and 9 buffers

The FTL we adopted in this example has 10 machines and 9 buffers. Each buffer space is assumed to be equal to the plant cells, $B_{m-c} = 1$. The machine shapes and specifications are as shown in Fig. 11. Table 1 gives the buffer size between each pair of machines in the FTL.
Table 1  The buffer size of the FTL

<table>
<thead>
<tr>
<th></th>
<th>B1</th>
<th>B2</th>
<th>B3</th>
<th>B4</th>
<th>B5</th>
<th>B6</th>
<th>B7</th>
<th>B8</th>
<th>B9</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>10</td>
<td>9</td>
<td>4</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Machines 1, 3, 5, 7 and 10</th>
<th>Machines 2 and 8</th>
<th>Machines 4, 6 and 9</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Machine 1-5" /></td>
<td><img src="image2" alt="Machine 2-8" /></td>
<td><img src="image3" alt="Machine 4-9" /></td>
</tr>
</tbody>
</table>

The available plant area and plant restricted locations are shown in Fig. 12. Figure 13 shows the FTL layout resulting from OOLM.

![Plant area model and restrictions](image4)

A 3D drawing of the FTL layout is drawn by a CAD system as shown in Fig. 14.

![A 3D drawing of FTL layout](image5)
9. Conclusions

This paper has found an efficient put of the FTL components including all machines and the material handling path between each pair of machines in a restricted area by proposing OOLM in conjunction with a genetic algorithm. The combination of OOLM and GA generates an efficient solution for a set of irregularly shaped machines throughout a restricted plant area. This combination can efficiently solve the FTL layout. In order to utilize the GA most efficiently, a new techniques of GA mutation and crossover are proposed. A 3D drawing of the FTL is drawn by linking a CAD system to the proposed OOLM. We used our developed OOLM to determine some FTL layout designs in different restricted areas. As a result, the FTL layout design achieving the best utilization of the plant area could be determined. The combination of OOLM and GA can be applied to obtain the layout of any FTL in any plant restricted area.

References