The Thirteenth International Symposium on Artificial Life and Robotics 2008(AROB 13th '08),
B-Con Plaza, Beppu, Oita, Japan, January 31-February 2, 2008

Production Layout Design System by GA with One by One Encoding Method

J. Abu Qudeiri, H. Yamamoto and R. Rizauddin

Intelligent Manufacturing Systems Laboratory, Gifu University,
1-1 Yanagido, Gifu Shi, 501-1193, Japan
(Tel : 81-58-293-2550; Fax : 81-58-293-2550)
(yam-h@gifu-u.ac.jp)

Abstract: One of the problems encountered in the design and implementation of a flexible transfer line (FTL) is the layout design of the FTL. The layout design of the FTL has an important impact on material handling. In this paper, we develop a LAyout Design system (LAD). LAD can find FTL layout design including the buffer size between each pair of FTL machine tools. LAD divides the goal plant into cells and uses GA to find the efficient FTL layout design. In order to carry out GA, we propose a new encoding method to express GA individual. The new encoding method is called a One by One Encoding Method (OOEM). OOEM generates the elements of the individual one by one with the sequence of relative direction of the FTL components. The sequence of individual elements indicates the sequence of machine tools and conveyer buffer spaces of FTL. The developed LAD based on the proposed OOEM is not limited to a single static environment plant, but is highly flexible within the plant structure. An application example was developed, and after a number of generations based on LAD, an efficient FTL layout design was able to be found.

Keywords: Flexible transfer line, Layout design, Genetic algorithm, Buffer size.

I. INTRODUCTION

One of the problems encountered in the design and implementation of a flexible transfer line (FTL) is the layout design of the FTL. The layout design of the FTL has an important impact on material handling. The efficient FTL layout design 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]. The FTL layout problem is identified as a NP-complete problem [2], many heuristic approaches have been developed to solve this problem for near optimum solution [3, 4], such as a simulated annealing [5], an ant algorithm [6] and constructive greedy heuristic [7]. A detailed layout planning was proposed by Yang *el al.* [8], but they considered an open plan without restrictions and regular machine tool shape. 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 machine tools.

In this paper, we develop a LAyout Design system (LAD). LAD can find FTL layout design including the buffer size between each pair of FTL machine tools. LAD divides the goal plant into cells and uses GA to find the efficient FTL layout design. For the efficient use of GA to carry out LAD, we propose a new encoding method to express GA individual. The new encoding method is called a One by One Encoding Method (OOEM). OOEM generates the elements one by one with the sequence of relative direction of the FTL components. LAD based on the OOEM 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 LAD, an efficient FTL layout design could be found.

II. LAYOUT DESIGN SYSTEM OUTLINE

To find the efficient FTL layout design, we develop LAyout Design system (LAD). The difficulty of finding FTL layout design is how to find the best location for each of the FTL components (machine tools and conveyer buffer spaces). LAD solves the problem by using two processes. First, LAD divides the goal plant and each of the FTL machine tools into cells. Second, LAD uses GA to find the efficient layout design. Before describing the two processes, the FTL model and the plant model assumptions are introduced

III. PLANT AND FTL ASSUMTIONS

The following assumptions are introduced for the plant model and FTL model.

1. FTL start point and end point are indicated through the plant layout.
2. Each machine tool is defined by its pick point and drop point. The pick point means the point where the parts enter the machine tool. The drop point means the point where the parts leave the machine tool.
3. The distance and the direction from the pick point to the drop point for each machine tool are known.
4. Machine drop point can be located by *0°, 90°, 180° or 270°* relative to its pick point.
5. Conveyer buffer spaces are divided into small sections. The section is called as space. Each consecutive spaces are neighbors
6. The style of FTL has a set of irregularly shaped machine tools.

IV. DIVIDING GOAL PLANT AND MACHINE TOOLS INTO CELLS

The first process of LAD is to divide the goal plant and the machine tools into cells. The divisions of the goal plant are carried out by using the following steps.

Step1: Draw the smallest rectangle that surrounds the goal plant layout, as shown in Figure 1.

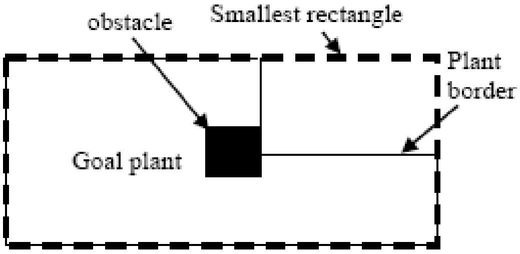


Fig. 1. Model of goal plant layout

Step 2: Divide the rectangle as a lattice as shown in Figure 2 and call the lattice element as a cell.

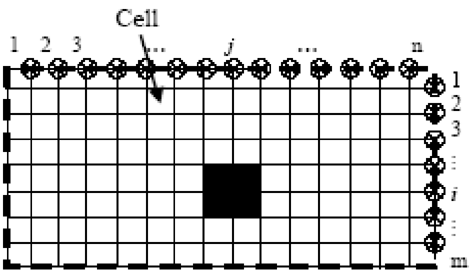


Fig. 2. Creation of the cells

Similarly, the machine tool plane is divided into cells by using the same steps described above. The set of cells of machine k is called as M^k_{cell}.

After the division of the goal plant is finished, the generated each cell state is decided whether the cell is empty or restricted. The decision is carried out by applying the following rule.

If the cell $(i,j) \in$ obstacle.
Then cell (i,j) state is called as restricted.
Else cell (i,j) state is called as empty.

V. USAGE OF GA FOR FTL LAYOUT

The second process of LAD is to use GA. The characteristics of our GA are described through the following sections.

1. Encoding: Express Individual by OOEM

In order to use GA, we need to express the individual. In order to do this, we propose OOEM. OOEM has two characteristics. The first characteristic is OOEM expresses the individual with a sequence of angles one by one. The sequence of angles has two types of combinations. One is the combination of two elements that indicate the directions of pick point and drop point of a machine tool. The other is the combination of one or more elements side by side and the elements indicate the directions of the conveyer buffer spaces. Hereafter, edge corresponds to pick point, drop point and all spaces. The element k in the sequence is defined by an angle, θ, which indicates the direction from the edge k-*1* to edge k, as shown in Figure 3.

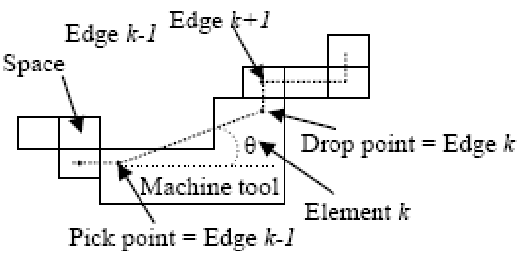


Fig. 3. Definition of element direction

The second characteristic is OOEM checks the cells state in order not to generate a lethal individual. That is, OOEM checks whether the FTL components can be located or not by judging empty or restricted cell state. In order to achieve these characteristics, OOEM has 4 functions enable the expression of the individual. The 4 functions are ① find FTL edges possible locations, ② check the component cells state, ③ randomly select one of the possible locations, ④ express the individual one by one.

By utilizing the 4 functions, OOEM generates the individuals. The following is the algorithm of the 4 functions. The notations used in the algorithm are defined first.

L_i The distance between edge *i-1* and edge *i*.

β_i The direction of the line that connects edge *i-1* and edge *i* from horizontal.

L The length of cell side.

B_k The size of conveyer buffer spaces between machine tool *k* and machine tool *k+1*.

N Number of machine tools in the FTL.

Step 1: Set R = FTL start point.

Step 2: Find the possible locations of edge *i*=1, P_i^1, P_i^2, P_i^3 and P_i^4 as follows.

$P_i^1 = (d_i, \varphi_i^1)$, $\varphi_i^1 = \theta_i$. $P_i^2 = (d_i, \varphi_i^2)$, $\varphi_i^2 = \theta_i + 90$

$P_i^3 = (d_i, \varphi_i^3)$, $\varphi_i^3 = \theta_i + 180$. $P_i^4 = (d_i, \varphi_i^4)$, $\varphi_i^4 = \theta_i + 270$

where

$$d_i = \begin{cases} L_i & \text{if edge } i = \text{drop point} \\ L & \text{if edge } i = \text{pick point or space} \end{cases}$$

$$\theta_i = \begin{cases} \beta_i & \text{if edge } i = \text{drop point} \\ 0 & \text{if edge } i = \text{pick point or space} \end{cases}$$

Step 3: Check the component cells state by using Steps 3.1 and 3.2.

Step 3.1: Set j=0

Step 3.2: Applying the following rule.

If($P_i^1 \in restricted$). *Then* set $P_i^1 = 0$.
Else if(*edge* $i = drop\ point$ and $M_{cell}^k \in empty$).
Then set j=j+1. *Else* set $P_i^1 = 0$
If($P_i^2 \in restricted$). *Then* set $P_i^2 = 0$.
Else if (*edge* $i = drop\ point$ and $M_{cell}^k \in empty$).
Then set j=j+1. *Else* set $P_i^2 = 0$
If($P_i^3 \in restricted$). *Then* set $P_i^3 = 0$.
Else if (*edge* $i = drop\ point$ and $M_{cell}^k \in empty$).
Then set j=j+1. *Else* set $P_i^3 = 0$
If($P_i^4 \in restricted$). *Then* set $P_i^4 = 0$.
Else if (*edge* $i = drop\ point$ and $M_{cell}^k \in empty$).
Then set j=j+1. *Else* set $P_i^4 = 0$

Step 4: Randomly select one of the enoty possible locations as follows. *If*($j \geq 1$). *Then* set R = *edge i*.
Else set i=i-1, set R = *edge i-1* and go to Step 3.

Step 5: Encode element i as φ_i^j.

Step 6: Repeat Step 2, 3, 4 and 5 to encode edge i, $\forall i = 2,3,\ldots,T$

$$T = \left(2N + -1 + \sum_{i=1}^{N-1} B_i \right)$$

The Steps 1 and 2 correspond to function ①, Step 3 corresponds to function ②, Step 4 corresponds to function ③ and Steps 5 and 6 correspond to function ④. By using the above algorithm, the individual with any size can be expressed.

2. OOEM Crossover

The traditional crossover operation is not suitable for the individuals coded by OOEM because the elements are expressed one by one. The main difference between traditional crossover and OOEM crossover is that in the conventional crossover the elements after the crossover point are swapped between the two individuals without any constrains. In the contrary, in OOEM crossover, the elements after the crossover point are selected one by one to avoid the generation of a lethal gene. The crossover of the individuals by OOEM is carried out as follows.

Step 1: Randomly select two individuals, individual 1 and individual 2.

Step 2: Set the first element of individual 1 as the first element of the generated individual, child 1.

Step 3: Set i=2.

Step 4: Check cell states that are located at (d_i, β_i) and (d_i, λ_i).

where

β_i is the i^{th} element of individual 2.

λ_i *is the* i^{th} element of individual 1.

Step 5: Apply the following rule.

If (d_i, β_i) is empty.
Then set β_i as the i^{th} element of child 1,
Else If (d_i, λ_i) is empty.
Then set λ_i as the i^{th} element of child 1,
Else If (d_i, λ_i+90) is empty.
Then set λ_i +90 as the i^{th} element of child 1,
Else If (d_i, λ_i+180) is empty.
Then set λ_i +180 as the i^{th} element of child 1,
Else If (d_i, λ_i+270) is empty.
Then set λ_i +270 as the i^{th} element of child 1,
Else go to Step 1.

Step 6: *If* $i<T$. Set i=i+1, *Then* go to Step 4,
Else end the algorithm.

To display the above algorithm, the following example for child 1 is introduced.

[1]: Assume that individual 1 and individual 2 of Step 1 are selected randomly as shown in Figure 4.

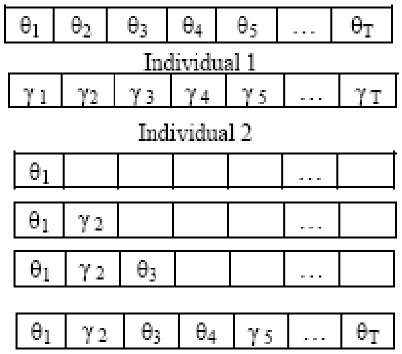
[2]: The first element of child 1 becomes θ_1 because the first element of individual 1 is θ_1.

[3]: i=2.

[4]: (d_i, β_i) and (d_i, λ_i) become (d_2, γ_2) and (d_2, θ_2) respectively.

[5]: Assume that (d_2, γ_2) is empty. The second element of child 1 becomes γ_2.

[6]: Child 1 of Figure 4 is acquired.



Generated individual, Child 1
Fig. 4. OOEM crossover operation

3. OOEM Mutation

The mutation operation is also different from a traditional mutation because the individual expression adopts OOEM. The main difference between the traditional mutation and the mutation of individuals coded by OOEM is that the cell state of the edge locations for all elements after mutation should not be restricted. The mutation is carried out using the following algorithm.

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

Step 2: Select two mutation locations randomly.

Step 3: Replace the elements of the two mutation location by each other.

Step 4: Check the cells state after mutation

Step 5: *If* any cell state after mutation is *Restricted*. *Then* go to Step 2. *Else* accept the mutation.

VI. NUMERICAL EXPERIMENT

The LAD based on OOEM was applied for FTL example. The FTL we adopted has 10 machine tools and 9 buffers. Each buffer space is assumed to be equal to the cell. Machine tools 1, 4, 8 and 10 have rectangular shape. Machine tools 3, 5, and 7 have L shape. Machine tools 2, 6 and 9 have U shape. The buffer sizes between each pair of machine tools are 7, 10, 6, 3, 13, 4, 5, 9 and 6, where 7 is the buffer size between machine tool 1 and machine tool 2, and so on. The acquired 3D drawing of the FTL layout is shown in Figure 5

VII. CONCLUSIONS

The paper has found an efficient put of the FTL components including all machine tools and conveyer buffer spaces path between each pair of machine tools in a restricted area by developing LAD in conjunction with GA.

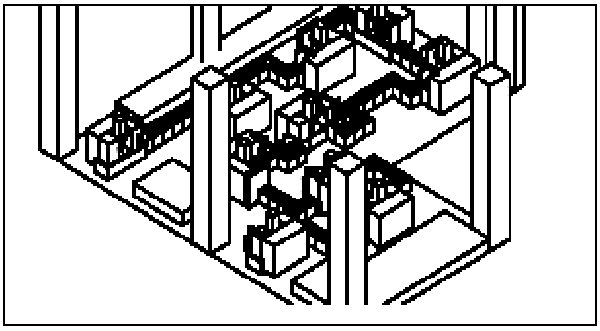


Fig. 5. FTL layout design for the application example

The combination of LAD and GA generates an efficient solution for a set of irregularly shaped machine tools throughout a restricted plant area. This combination can efficiently solve the FTL layout. In order to utilize the GA most efficiently, OOEM is proposed as a new encoding method. We used our developed LAD 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.

REFERENCES

[1] Tompkins J, White J, Bozer Y, Frazella E and Tanchoco J (1996), Facilities Planning, 2nd edition, Wiley, New York.

[2] Suresh G and Sahu S (1993), Multiobjective facility layout using simulated annealing, International Journal of Production Economics, 32: 39-54.

[3] Conway D and Venkataramanan M, Genetic search and dynamic facility layout problem (1994), Computers and operations research 21(8): 955-960.

[4] Raoot, A. D. and Rakshit, A., 'Fuzzy' heuristic for the quadratic assignment formulation to the facility layout problem, International Journal of Production research, Vol. 32, No. 3 (1994), pp. 563-581.

[5] Heragu, S. S. and Alfa A. S., Experimental analysis of simulated annealing based algorithm for the layout problem, European Journal of Operational Research, Vol. 57, No. 2 (1992), pp. 190-202.

[6] Solimanpur, M., Vrat, P. and Shankar, R., An ant algorithm for the single row layout problem in flexible manufacturing system, Computers & Operations Research, Vol. 32 (2005), p. 583-598.

[7] Kumar, K. R. and Hadjinicola, G. C., A heuristic procedure for the single row facilities layout problem, European Journal of Operational Research, Vol. 87 (1995), pp. 65-73.

[8] Yang, T., Peters, B. and Tu, M., Layout design for flexible manufacturing systems considering single loop directional flow patterns, European Journal of Operational Research, Vol. 146 (2005), pp. 440-455.