Parts Layout Decision of Cell Production Assembly Line using Genetic Algorithm and Virtual System

Hidehiko Yamamoto[1], Tomokazu Watanabe[2], Takayoshi Yamada[3], Masahiro Nakamura[4], Raafat H. Elshaer[5]

[1,3]*Department of Human and Information Systems, Gifu University, Japan*
[1]*(yam-h@gifu-u.ac.jp),* [3]*(yamat@gifu-u.ac.jp)*
[2]*Graduate School of Engineering, Gifu University, Japan, (o3128038@edu.gifu-u.ac.jp)*
[4]*Lexer Research Inc., Japan*
[5]*Zagazig University, Egypt, (r_h_elshaer@zu.edu.eg)*

Abstract: One of the problems of cell production system is how to decide the parts layout locations. Traditionally, the problem is solved using trial and error method which takes a lot of efforts and time. In this paper, we propose a Virtual Assembly Cell-production system (VACS) for the cell production assembly line. The VACS use a genetic algorithm (GA) system to find a reasonable solution and a virtual production (VP) simulator for giving us a visibility of that solution in the production system. The validation and the efficiency of the proposed VACS system are tested on ten varieties of a product. The simulator results show that the VACS system is capable of getting good solution in a reasonable computational time when compared to the traditional one.

Keywords: Genetic algorithm, Parts layout locations, Cell production, Virtual production**.**

I. INTRODUCTION

Recently, the production system has changed from mass production of a limited variety of products to low volume production of a wide variety of products, mass customization[1,2]. It is because the consumers' individual needs are diversified. Therefore, it is important to produce products of a wide variety efficiently. One of the needs for modern production methods is the cell production system[3]. A cell production system is a production system in which a single worker or small team of production workers perform multiple production jobs in short segment lines. The cell design places a wide range of tools and equipment in close proximity to workers, enabling them not only to perform a wide range of production tasks, but to customize the products as well. One of the problems of cell production system is the parts layout locations. The current state of this problem is to be solved empirically using a trial and error method which takes a lot of effort and time. In addition, the improvement in production efficiency is going up gradually during the assembling process where there is no opportunity to make the production efficiency a peak from the beginning.

In this research, we propose a Virtual Assembly Cell-production System (VACS), a cell production simulation system, for solving the problem. The VACS integrates a Parts Layout Decision system (PLD system) and a Virtual Production simulator (VP simulator). To carry out the PLD system, we adopt a genetic algorithm (GA) system whose crossover method is the original called TTC.

The paper is further organized as follows. Section 2 introduces the VACS system and its two functions, PLD and VP systems. In section 3, the test problem and the computational results are presented. Finally, section 4 concludes the paper.

II. VACS

The VACS system consists of two collaborating functions, the PLD System including GA system and the VP simulator, as shown in Figure 1. The steps of the VACS system are described in Table 1.

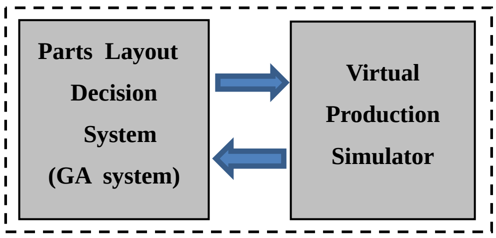


Fig.1 VACS system

Table 1. Steps of the VACS system.

Step 1.	VP simulator draws the workshop floor, and sends the coordinated data of the locations, such as shelves and worktables, to the PLD system.
Step 2.	In the PLD system, the parts layout locations are decided using the GA system and are sent back to the VP simulator.
Step 3.	VP simulator draws the parts layout and animately visualizes the working environment.

1. VP simulator

The VP simulator arranges the parts inside the shelves in workshop floor according to the received information from PLD system showing that in three dimensions. In addition, it visualizes the working environment in which an animation of the assembling process is shown (see section III.3).

2. PLD system

The PLD system uses a GA system to decide the better parts layout locations in terms of minimum total moving distances. In the GA system [4], the information of the parts layout locations encodes into feasible chromosome. In the searching process for finding a better parts layout locations, genetic operators, such as crossover and mutation …etc., are repeated, until a predefined stop criteria is verified. More details about the proposed GA in the following paragraphs.

1. Chromosome representation

To treat the information of the parts layout locations, we use a direct representation in which a part number is expressed as a gene, and the part position in the chromosome is expressed as the part location in the layout. For instance, Figure 2 is expressed with the following chromosome: <DFABEC>.

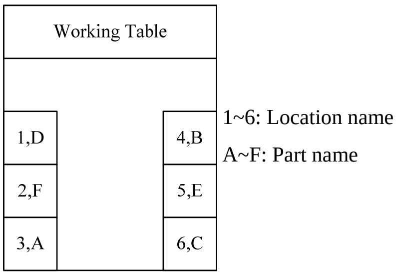


Fig.2. An example of chromosome representation

2. Initial population and fitness function

Initial population is randomly generated and the fitness function is expressed as the reciprocal of the total moved distances to achieve a certain amount of production.

3. Selection, crossover and mutation methods

In selection for crossover [4], in this paper, Roulette-wheel selection method is used. Applying conventional crossover methods in our proposed chromosome may generate a lethal chromosome. To solve this problem, we develop what we call Twice Transformation Crossover (TTC). By this method, the chromosome is transformed into the shape that can be cross over, and it is reversely transformed after crossover to its former shape. One point crossover is applied with probability CP. The TTC is as shown in Table 2. As an example, from Figure 2, consider Ch1<CEDFAB> and Ch2<AFBCDE> be two selected chromosomes for crossover. The steps of TTC for generating two offsprings, Off1 and Off2, from the two chromosomes, Ch1 and Ch2, are explained in Table 3.

Each gene in the chromosome may be mutated with probability MP. Mutation method is to swap the selected gene with randomly selected one.

Table 2: Twice transformation crossover method

1	In non-decreasing order the parts are arranged in list *L*
2	Take the first gene, *i*, in the current chromosome
3	While *L* is non-empty, Do:
	3.1 Replace gene *i* with its order in list *L*.
	3.2 Remove part *i* and update list *L*
	3.3 *i* equals next gene
4	Apply Steps from 1 to 3 to transform two selected chromosomes.
5	Apply one point crossover to generate two offsprings
6	Use the reverse method of steps 1 to 3 to transform the two offsprings into the original shape

Table 3. An example for the TTC method

First Chromosome:Ch1< CEDFAB > *L*={A, B, C, D, E, F}, and *i* = C Ch1<3EDFAB> , Update *L* *L*={A, B, D, E, F}, and *i* = E Ch1<34DFAB>, Update *L* *L*={A, B, D, F}, and *i* = D Ch1<343FAB>, Update *L* … Ch1<343311> By the same method Ch2<151111>
Assume that a one point crossover is randomly chosen between position 3 and 4 to generate Off1 and Off2.
Off1<151311> and Off2<343111>
Decode Off1 to original shape: *L*= {A, B, C, D, E, F} Off1<A51311>, Update *L* *L*= {B, C, D, E, F} Off1<AF1311>, Update *L* … Off1<AFBECD> By the same method Off2<CEDABF>

4. Stop criteria and GA parameters

The control parameter values and terminating condition used in our GA was selected based on several preliminary runs with alternate control parameters and terminating conditions on different instances of the problem. These values were then used for the test problem reported in the computational results. The final parameter values are summarized in Table 4.

Table 4. Parameter values for the proposed GA

Description	*Values*
● Pop. Size	100
● Crossover rate	0.9
● Mutation rate	0.05
● % of solutions replaced by new gen.	0.95
● Stop criteria	100*

*Stop after 100 generations without improvement

III. COMPUTATIONAL RESULTS

1. Test problem

The developed VACS system is tested on a cell-production assembly line of a personal computer. The workplace design and the required parts are as shown in Figure 3. The workshop shelves layout and the final product are as shown in Figures 4 and 5 using VP simulator. There are ten varieties of the product in which each one contains at most 18 parts. The assembly process sequence of each product type is known. The parts of the same type are arranged in one shelf. The workplace contains one worker, and the movement between the shelf and the worktable is in a straight line.

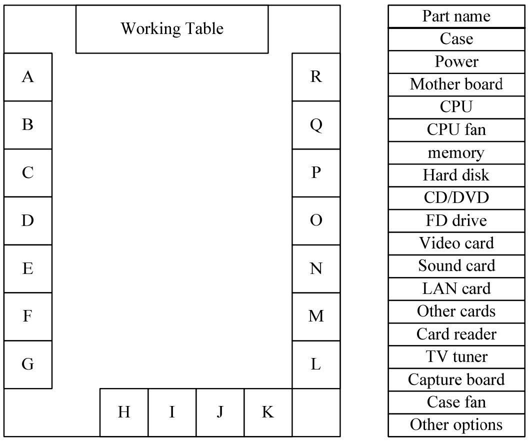


Fig.3. Test problem information

2. Results

Table 5 shows the computational results of 10 runs for the PLD system with the proposed genetic algorithm and with the random method. Columns 2 and 3 show the best total moving distances of the both methods. The percentage improvement of the GA and the computational CPU time are shown in columns 4 and 5 respectively. Using T-test at 99% confidence level, we find that our developed algorithm makes a statistically significant improvement over the random one.

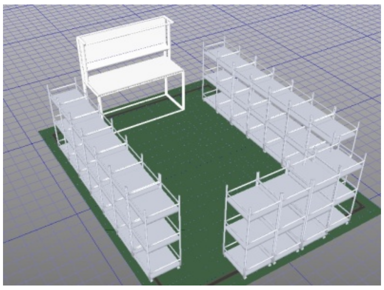


Fig.4. workshop floor shown the worktable and shelves

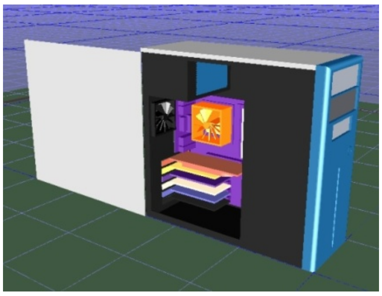


Fig.5. 3D configuration of the final product

Table 5. Comparison between PLD with GA and without GA (units in meter)

Simulation No.	Random Method	GA Method	% of Imp.	GA's CPU time (msec)
1	5087	5003	1.65	718
2	5238	5053	3.53	795
3	5251	5053	3.77	780
4	5226	5011	4.11	920
5	5179	5092	1.68	468
6	5150	5027	2.39	843
7	5157	5025	2.56	921
8	5099	4975	2.42	1104
9	5210	5190	0.39	717
10	5227	5021	3.95	655

A comparison between the fitness curve and the distance reciprocal curve of the random method for getting a best layout in the first simulation is as shown in Figure 6. It is clear from the figure that the random method makes a dramatically improvement in the beginning after that it continues without gaining any improvement. In the other side, the fitness curve is getting better. This difference between the two methods is because the GA exploits the historical information to make improvement, but the random method is not. Figure 7 shows the best parts layout location obtain using the GA from the first simulation.

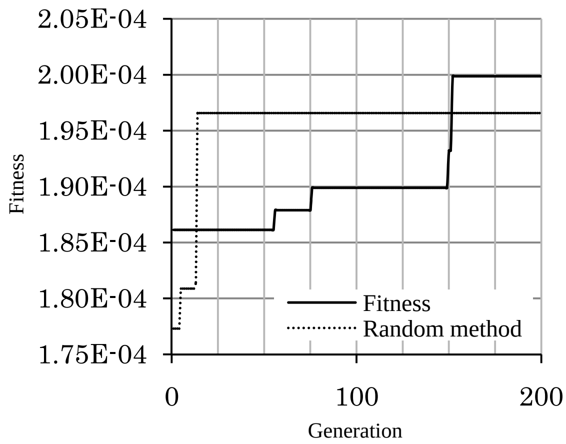
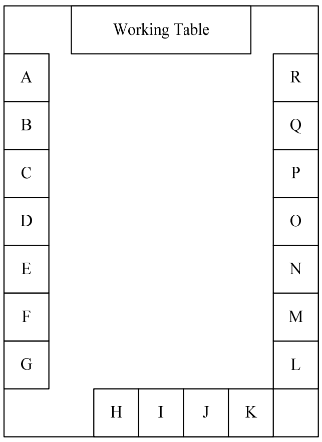


Fig.6. Best fitness curve



Location	Part name
A	Sound card
B	Case
C	Video card
D	Capture board
E	FD drive
F	CPU fan
G	Other options
H	Case fan
I	LAN card
J	Card reader
K	Other cards
L	CD/DVD
M	Mother board
N	Hard disk
O	Power
P	memory
Q	CPU
R	TV tuner

Fig.7. Best parts layout using GA from first simulation

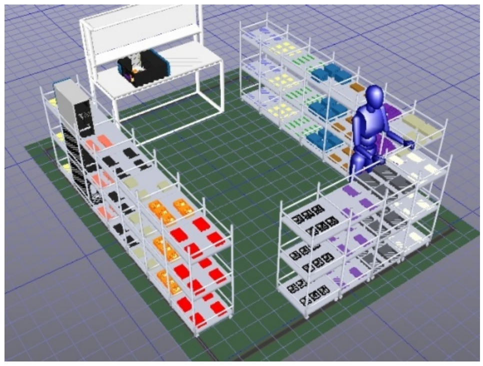
3. Visualization by VP simulator

VP simulator receives the output results from PLD system and visualizes the working environment. Figure 8(a and b) shows snapshots from the virtual production of our test problem for the best parts layout locations shown in Figure 7.

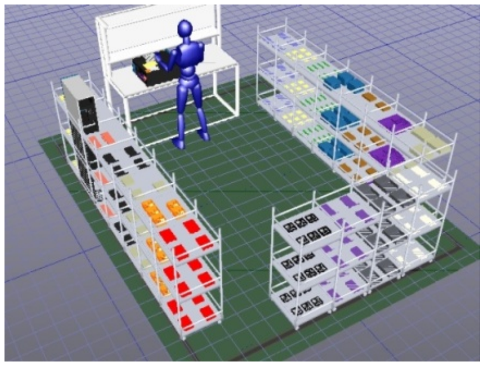
VI. CONCLUSIONS

In this paper, we developed VACS system that integrated PLD system and VP simulator. The PLD system used our developed GA system whose crossover method used the original TTC. The PLD system obtained a good layout for the parts locations in a reasonable computational time. The VP received the layout from the PLD system and visualized the working environment.

From the computational results, we found that the PLD system with the proposed GA was statistically significant impact on the results than the PLD without the GA. Moreover, the VP simulator can be used for educational purposes where the steps of assembling process are visualized with animation.



(a)



(b)

Fig.8. Virtual production for the layout shown in Fig.7.

REFERENCES

[1] Juan Diego Frutos, Denis Borenstein(2004), A framework to support customer-company interaction in mass customization environments, Computers in Industry 54 (2):115-135
[2] Suresh Kotha (1996), From mass production to mass customization: The case of the National Industrial Bicycle Company of Japan, European Management Journal 14(5):442-450
[3] Katsuhide Isa, Tsuyoshi Tsuru (2002), Cell production and workplace innovation in Japan: Toward a new model for Japanese manufacturing?, Industrial Relations: A Journal of Economy and Society 41(4):548-578.
[4] D.E. Goldberg, Genetic algorithms in search (1989), optimization, and machine learning, Addison-Wesley