Layout decision system for multiple production lines using work-flow-line and GA

Masato Noda
Department of Intelligence Science and Engineering, Gifu University, Yanagido 1-1,
Gifu, Gifu 501-1194, JAPAN

Hidehiko Yamamoto
Department of Mechanical Engineering, Gifu University, Yanagido 1-1,
Gifu, Gifu 501-1194, JAPAN

Hirohumi Tsuji, Yasuhisa Terawa, Yoshinori Nakamura, Masayuki Tsuchida
InfoFarm co.,Ltd, Yanaizu-cho Distribution Center 1-8-4
Gifu, Gifu 501-6123, JAPAN

Katsuaki Yamada, Yukiyasu Kuriyama
Kai industries co., ltd, Oyana 1110
Seki, 501-3992, JAPAN

E-mail: y4525062@edu.gifu-u.ac.jp, yam-h@gifu-u.ac.jp

Abstract

We develop the system to decide the efficient layout of assembly production line by using Genetic Algorithm (GA). We call the system as System of Production-line-layout Decision by Chameleon-code and GA (PDCG). PDCG decides the efficient layout of production line by using GA, work-flow-line acquired by Chameleon code and the machine breakdown data. PDCG evaluates the layout efficiency by calculating the operator's walking time in order to fix the machine breakdown occurred on the production line. By the evaluation.

Keywords: Genetic Algorithm, Chameleon-code, Assembly production, Work-flow-line, Machine breakdown

1. Introduction

In order to improve the production efficiency, many Japanese companies adopt Kaizen. The line manually performed by several operators is needed to be also improved. In this study, we develop the system of Production-line-layout Decision by Chameleon-code and Genetic Algorithm (PDCG). PDCG determines the layout of multiple production lines that some operators work. The determination is given by using Genetic Algorithm (GA), and the work flow lines obtained from Chameleon-code. PDCG is applied to the razor assembly line to verify whether it is effective in improving work efficiency.

2. Work-flow-line with Chameleon code

Chameleon code is the new color barcode whose function has a high speed and precision recognitions as shown in Figure1.This study puts Chameleon code on the top of an operator's hat and obtains the operator's work flow line data by overhead cameras. The acquired work flow line data includes the name of the operator, the camera names, the dates and times, and the X and Y coordinates of the chameleon code.

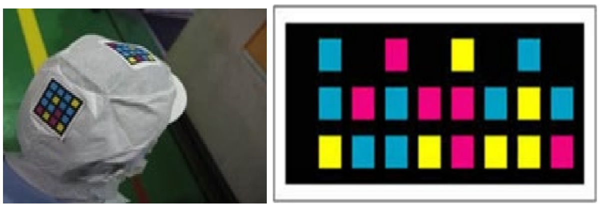

Fig. 1. Chameleon-code

3. PDCG outline

PDCG is the system to determine the layout of production lines using GA, based on the work flow line database acquired from the chameleon code and the machine breakdown database. Specifically, PDCG calculates the walking time of operator moves to fix machine breakdowns, evaluates the operator's walking time and determines the better layout where one operator manages some production lines. As shown in Figure 2, PDCG consists of the condition module and the GA module. Condition module reads the acquired work flow line database and sets various parameters for GA. GA module determines the better production line layout by using GA.

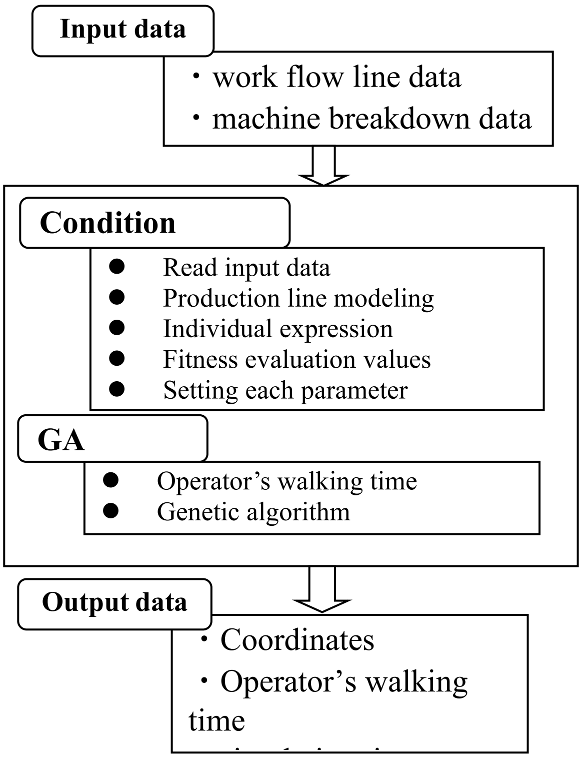

Fig. 2. PDCG outline

4. Individual representation

The individual representation of GA used in PDCG adopts the structural gene, as shown in Figure 3. We express the x-coordinate, the y-coordinate and the angle indicating the position of one production line as the three consecutive loci. When representing multiple production lines, these three loci are connected in series. Figure 3 shows the individuals in the case of three production lines. The x-coordinates, y-coordinates and angles are shown in Figure 4. In order to secure the movement path of the operator, we set the constraint that the width between each production line is an arbitrary value Z mm. Individuals that don't meet the constraint are deleted as lethal genes.

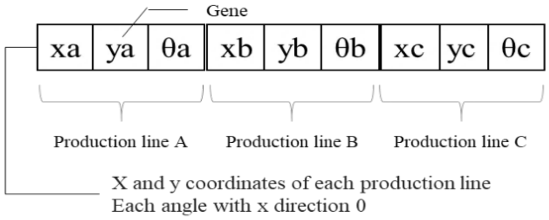

Fig. 3. Individuals expression

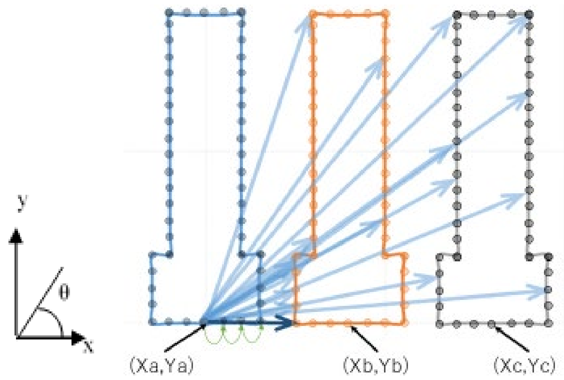

Fig. 4. How to judge lethal chromosome

5. Fitness evaluation values

PDCG adopts the walking time of the operator during the operating time as the evaluation values of fitness. If the walking time of individual i is t_i and the slowest walking time among the population of a generation is t_w, the fitness f_i of individual i is expressed by the following equation.

$$f_i = (t_w - t_i + 1)^2 \qquad (i = 1,2,3, \dots) \tag{1}$$

If the time difference between each individual and the slowest individual is taken as the fitness, the difference in fitness between an excellent individual and another individual becomes small. To make large difference, we give the difference a square. [1].

6. GA module algorithm

The GA module is the module to improve the efficiency of production line layout by GA. The algorithm of the GA module is shown below.

Step1: Generate the initial population.
Step2: Calculate the machine breakdown coordinates for each production line.
Step3: Calculate the walking time of the operator.
Step4: Calculate fitness.
Step5: Judge whether the end condition is satisfied or not, and if it is satisfied, go to Step 8. If not, go back to Step6.
Step6: Carry out selection, crossover and mutation operations.
Step7: Carry out Step 2 on the new generation's individuals.
Step8: The production line layout of the individual with the shortest walking time is adopted as the satisfying layout and is output as a solution.

We describe the details of the above algorithm.

Step1: Generate the initial population.

The individuals of the initial population are randomly selected from a range of coordinates that can be placed in the factory layout where the x- and y-coordinates and angles of the production line are applied. When there are multiple target production lines, the layout is randomly generated. There is a possibility that the production line layout may overlap or be too close. In this case, these layouts correspond to lethal genes. Not to include the lethal genes in the population, the following operations are carried out.
⑴ Generate individuals. ⑵ As shown in Fig. 4, calculate the distance between the contour of a production line and the contours of other production lines. ⑶ If the calculated distance is less than Z mm, the individuals are deleted as a lethal gene. ⑷Return to ⑴ and regenerate the individual.

Step2: Calculate the machine breakdown coordinates for each production line.

The coordinates whose machine gets breakdown in the factory layout change according to the coordinates of the production line layout. That is, since the coordinates of a machine breakdown occurrence for every generated individual are different, it is necessary to calculate each walking time. In PDCG, the coordinates of a machine breakdown occurrence of all individuals are calculated by the following operations.
(1) Search the work flow line data corresponding to the machine breakdown occurrence time in the machine breakdown data and find the coordinates of the machine breakdown that occurred in one production line. (2) Based on the found coordinates, calculate the machine breakdown occurrence coordinates for other all individuals. For example, if the coordinates and angles of one production line A of an individual generated in Step 1 are (x_a, y_a, θ_a) and the machine breakdown coordinates found in (1) are (x_e,y_e),the machine breakdown coordinates (X_e, Y_e) in production line A are expressed as the following equations.

$$X_e = x_a + x_e * sin\theta_a \qquad (e = 1,2,3, \dots) \qquad (2)$$

$$Y_e = y_a + y_e * cos\theta_a \qquad (e = 1,2,3, \dots) \qquad (3)$$

Step3: Calculate the walking time of the operator.

When a machine breakdown occurs, the operator moves from the waiting place to the place where the machine breakdown occurred to fix it. PDCG calculates the total walking time to get there and back.

Step4: Calculate fitness by using the equation (1).

Table.1 Mechanical breakdown example

Time	X coordinate	Y coordinate	Breakdown name A
411	3,513	4,046	Whether the blade is abnormal
565	3,858	2,046	Whether the blade is abnormal
890	4,128	881	On standby
1,156	2,651	3,468	Abnormal presence or absence of frame

Step5: Judge whether the end condition is satisfied or not. If it is satisfied, go to Step 8. If not, go back to Step6.

After the walking time of the operator is updated, if no better individual appears after certain generations, GA is finished.

Step6: Carry out selection, crossover and mutation operations.

Using the fitness calculated in Step 4 and using the fitness proportional method, select a set of the individuals as a crossover target. As the crossover, we adopt the two-point crossover. Mutations occur with a certain probability within the population.

Step7: Carry out Step 2 on the new generation's individuals.

Repeat steps 2 to 6 for the next generation population.

Step8: The production line layout of the individual with the shortest walking time is adopted as the satisfying layout and PDCG outputs it as a solution.

The coordinate data of the satisfying individual is output, and the layout is represented as an image based on it.

7. Simulations applications and results

PDCG was applied to the razor assembly line in the Kai Industries Co., Ltd. Since the shape of the razor assembly line is extremely complicated, we simplified it to the graphic model as shown in Fig. 5 and applied it to PDCG.

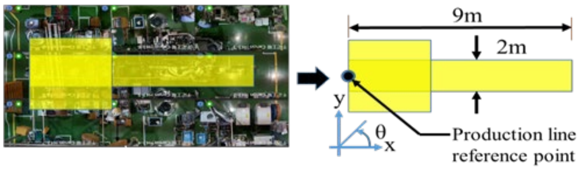

Fig. 5. Production line model

In this case, we acquired the work flow line data for 8 hours per one shift for 3 days with the chameleon code and also used the machine breakdown data of the same time. We adopt the GA parameters as follows. A mutation rate is 10%, a population size is 200 individuals, one individual is preserved as an elite. Selection and crossover methods were carried out by the fitness proportional method and two-point crossover. We also adopted b = 500mm and Z = 800mm.

The simulation results are shown below. The simulations were performed 20 times, and the layout obtained by PDCG is the solution. The layout is shown in Figure 6. The solution corresponds to the individual

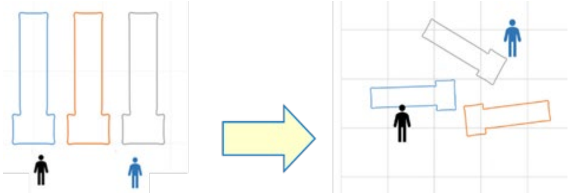

Fig. 6. Parallel layout and PDCG layout

with the minimum walking time. As a comparison, the walking time of the current layout of Kai Industries Co., Ltd. which is arranged in parallel was also calculated. The layout determined by PDCG shortened the operator walking time by about 63 minutes compared to the parallel layout. The layout determined by PDCG was effective in improving work efficiency.

Table. 2. Comparison of simulation results

	Walking time[sec]
PDCG layout	2,405
Parallel layout	6,164

8. Conclusions

After application simulations, it is ascertained that PDCG using the chameleon code of Infofarm Co., Ltd. is effective for the decision to find better layouts with multiple lines that one operator works.

9. Reference

(1) K. Honda, H. Yamamoto and T. Yamada, "Design System of Cell Type Assembly Machine with Dual Arms Robot by GA", The 2018 International Conference on Artificial Life and Robotics (ICAROB2018), Feb. 1-4, Beppu International Convention Center, Oita, Japan