Proceedings of the 15th International Conference on Manufacturing Systems – ICMaS
Published by Editura Academiei Române, ISSN 1842-3183

"Politehnica" University of Bucharest, Machine and Manufacturing Systems Department
Bucharest, Romania, 26–27 October, 2006

THE EVALUATION AND IMPLEMENTATION OF THE MODEL OF THE OPTIMIZATION OF A FLEXIBLE ASSEMBLY SYSTEM

Adriana FOTA

Abstract: *Many reconfigurable conveyor-components have been developed for the construction assembly line systems. The components have different transporting paths, shapes, sizes, etc. This paper describes a genetic algorithm to configure those reconfigurable conveyor-components forming a flexible assembly line system to meet the ever-changing production requirements. The transporting paths, shapes and sizes of reconfigurable conveyor-components are coded into binary string as chromosome to represent an assembly line layout for analysis and evaluation. The three evolutionary processes generate the layouts: selection, crossover, and mutation. The process of updating control parameters is integrated into the genetic algorithm to improve the performance and efficiency of the evolutionary processes. The reconfiguration of a flexible assembly line system to meet the requirements of minimization of the number of reconfigurable conveyor-components and the provision of alternative processes paths are discussed in detail in the paper.*

Key words: *flexible assembly system, optimisation, reconfiguration.*

1. INTRODUCTION

Nowadays, the customer-driven products are very diversified. Facing of this manufacturing situation, many factories have attempted to introduce flexible assembly line systems as the strategy to produce the diversified products. Recently, many reconfigurable conveyor-components have been developed.

Typical reconfigurable conveyor-components are linear conveyor, rotating conveyor, conveyor-bend, s-shape conveyor, U-shape conveyor, and lift conveyor. They have different transporting paths, shape, sizes, etc. Those reconfigurable conveyor-components can be formed into various assembly line configurations. As the result, a number of design alternatives may exit and many possible system configurations can be formed to meet the production needs. This is very difficult to reconfigure an assembly line system among all possible configurations.

2. EVALUATION OF FLEXIBLE ASSEMBLY LINE SYSTEM RECONFIGURATION

The evaluation phase contains two major steps: first is to calculate objective values for each assembly line layout; second is to convert objective values into fitness values. In the example, two requirements have been defined in separately for reconfiguring different properties of the assembly line system. Requirement 1 is to minimize the number of the reconfigurable conveyor-components. Requirements 2 are to provide three alternative processing paths in the flexible assembly line system. For requirement 1, four objective functions (obj_fun) have been considered for evaluating each candidate assembly line layout as follow:

1. to evaluate the layout whether the work-part flows from loading to unloading stations;
2. to evaluate the layout whether the work-part passes all processing workstations in sequence;
3. to evaluate the layout whether within the provided space;
4. to evaluate the shortest processing time of the production.

For requirement 2, four objective functions (*obj_fun*) have been considered for evaluating each candidate assembly line layout as follow:

5. to evaluate the layout whether the work-part flows from loading to unloading stations;
6. to evaluate the layout whether the work-part passes all processing workstations in sequence;
7. to evaluate the layout whether within the provided space;
8. to evaluate the maximum numbers of alternative flow paths in the layout.

Each objective function has its own score for representing its objective value. If the layout meets an objective function, the layout will get score '1', otherwise it will get score '0'.

Finally, the total objective value will be converted into the fitness value as flow:

$$\text{eval}(v_k) = \sum_{k=1}^{obj\_fun} s_0, \qquad (1)$$

where: s_o is each objective value for each obj_fun and eval (v_k) is fitness value for each chromosome v_k.

The evaluation method and fitness calculation of each objective function are described detail in section follows.

2.1. Selection

The selection operator is used to select parent for generation. A roulette wheel approach has been adopted as the selection procedures. It belongs to the fitness-proportional selection and can select a new population with respect to the probability distribution based on fitness values. The roulette wheel can be constructed as follows:

Step 1: Sum up the fitness value eval (v_k) for each chromosome v_k; named as total fitness (F) is:

$$F = \sum_{k=1}^{\text{pop\_size}} \text{eval}(v_k). \quad (2)$$

Step 2: Generate a random number (r) from the range [0, F].

Step 3: Return the first population member whose fitness, added to the fitness of the preceding population member, is greater than or equal to r. For example, there are four chromosomes in the population pool, v_1, v_2, v_3, v_4 with fitness values 1, 2, 2, 3, respectively.

Then calculate cumulative probability (q_k):

$$q_1 = 1,\ q_2 = 1 + 2 = 3,\ q_3 = 1 + 2 + 2 = 5,$$
$$q_4 = 1 + 2 + 2 + 3 = 8.$$

Now ready to spin the roulette wheel four times, and each time a single chromosome is selected for parent pool. Let us assume that a random sequence of 4 numbers from the range [0, F] is 4, 6, 2, and 7.

The first number $r_1 = 4$ is greater than q_2 and smaller then q_3, that means the chromosome v_3 is selected for forming first parent (P_1).

The second number $r_2 = 6$ is greater than q_3 and smaller then q_4, that means the chromosome v_4 is selected for forming second parent (P_2)< and so on. Finally, the parent pool consists of v_3, v_4, v_2, and v_4 for crossover.

2.2. Crossover

Crossover is the main genetic operator. It transfers a portion of genetic codes between two-selected parents and then generates offspring by combining both chromosomes' feature.

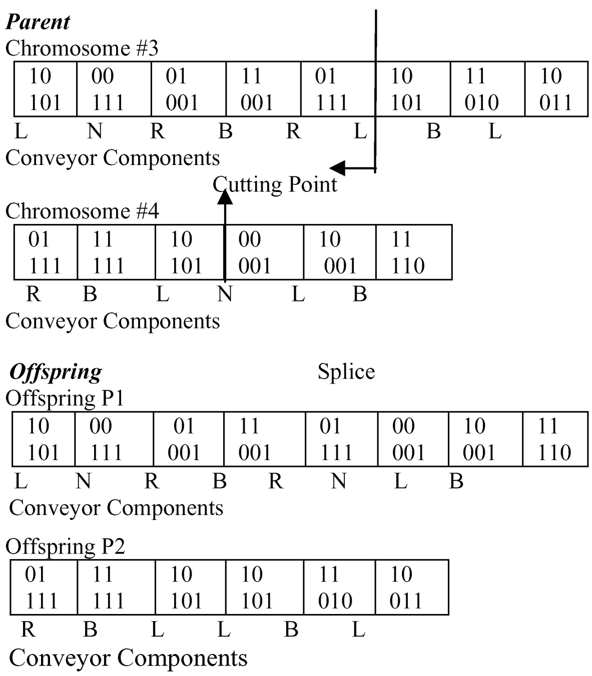

Fig. 1. Schematics of cut and splice operators.

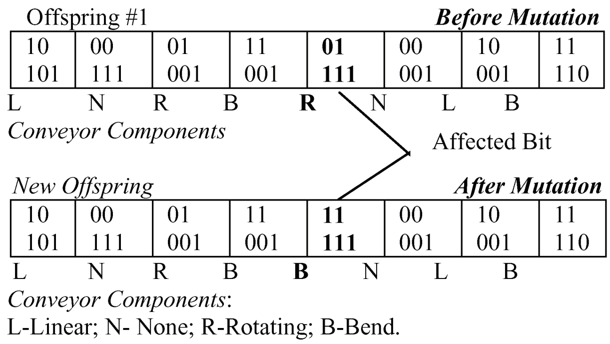

Fig. 2. Schematics of mutation operator.

The position of cuts can be chosen independently for both parents. After the cut operation partial strings are spliced in a random order as show in Fig. 1, where: L –Linear; N – None; R – Rotating; B – Bend.

2.3. Mutation

Mutation is a background operator, which produced spontaneous random changes in various chromosomes. This operator is randomly applied with probability P_m during evolution and helps to ensure that no point in the search space has a zero probability of being examined. For each gene in a chromosome, an arbitrary choice is made to decide whether the mutation operation is performed or not. If the decision is not to perform the mutation operation, the gene will be kept unchanged. Otherwise, the affected bit may change value from 1 to 0 from 0 to 1 as shown in Fig. 2.

2.4. Updating control parameters

The crossover rate is defined as the ratio of the number of offspring produced in each generation to the population size [4]. This ratio controls the expected number of chromosomes to undergo the crossover operation. In order to improve efficiency of the evolutionary process, the P_c should be updated during the fitness average (f_{av}) of population >= Max. Fitness Value ($F_{\max}$) – 1, that means when the f_{av} is greater or equal to $F - 1$, the P_c will be changed from 0.9 to 0.65. Because a high crossover rate allows exploration of more of the solution space and reduces the chances of settling for a false optimum; but if this rate is too high, it results in the wastage of a lot of computation time in exploring unpromising regions of the solution space.

3. INMPLEMENTATION OF THE MGA APPROACH

Let pop_size denotes the size of population, let P_c denotes the crossover rate, let P_m denotes the mutation rate, and let max_gen denotes the maximal generation for a run.

The data flow chart of the messy genetic algorithm (MGA) cycle is illustrated in Fig. 3, and the procedures of operating MGA are summarized as follows:

Step 0 (Parameter setting):

Set evolutionary environment: workstations, loading and unloading positions, pop_size = 60, P_c = 0.9 for f_{av} <

$< F_{max} - 1$ and $P_c = 0.65$ for $f_{av} >= F_{max-1}$, $P_m = 0.001$, and max_gen = 200.

Step 1 (Initialization):

Randomly generate initial population containing pop_size chromosomes.

Step 2 (Evaluation): Decode the chromosomes and calculate fitness value of each layout.

Step 3 (Selection): Make a roulette wheel selection to select fitter chromosomes from the current population for generation.

Step 4 (Crossover): Make pop_size * P_c offspring using the cut and splice operator.

Step 5 (Mutation): Make off_size * P_m offspring using the proposed mutation operator.

Step 6 (Evaluation): The offspring have been generated and evaluate their fitness values. If generation equal to the max_gen or the fittest layout has been generated, stop the evolutionary process; otherwise go to step 7.

Step 7 (Replacement): Insert the new chromosomes into a new population and go back to step 3.

3.1. Fulfillment of requirement 1

The proposed method has been applied for the minimization of the number of reconfigurable conveyor-components. The possible layout for this case is given in Fig. 4.

It has fulfilled the four objective functions. The fitness value of this layout is 4.

The calculation of evaluation phase (evaluation of the objective functions for requirement 1) is listed in follows:

Decoded Chromosome (Requirement 1):

(150, 220) (182, 220) (214, 220) (246, 180) (246, 180) (246, 156) (214, 124) (182, 124) (150, 124)

- **Objective function no 1:**

Does the work-part flow from LS to US?

Evaluation Criteria: In the first and the last coordinates of decoded chromosomes of the layout are equal to the given coordinates as shown in anterior paper (Fig. 3), the s_1 of the objective function will get 1, otherwise will get 0.

Objective value: $s_o = 1$.

- **Objective function no. 2:**

Does the work-part pass three WS in sequence?

Evaluation Criteria: If the distance between the conveyor and a WS is smaller than 30 mm, that means the conveyor has passed the WS, the s_2 of the objective function will get 1, otherwise will get 0.

Objective value: $s_o = 1$.

- **Objective function no. 3:**

Is the layout smaller than the given space as shown in Table 1?

Evaluation Criteria: If the overall size of the layout is smaller than the given space as shown Table 1, the s_3 of the objective function will get 1, otherwise will get 0.

Objective value: $s_o = 1$.

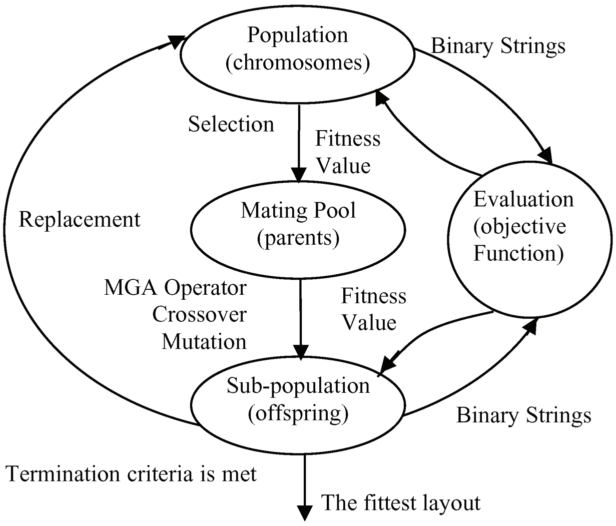

Fig. 3. Messy genetic algorithm cycle.

- **Objective function no. 4:**

To minimize the processing time.

Evaluation Criteria: The processing time is calculated by:

$$[(N_R \times t_R) + (N_L \times t_L) + (N_B \times t_B)] / P_{val}, \tag{3}$$

where, N_R, N_L, N_B is the number of rotating, linear, bend conveyors.

Objective value: $s_o = 1$.

- **Fitness Value:**

$$eval(v_k) = \sum_{o=1}^{obj\_fun} s_o. \tag{4}$$

Objective value: $s_o = 4$.

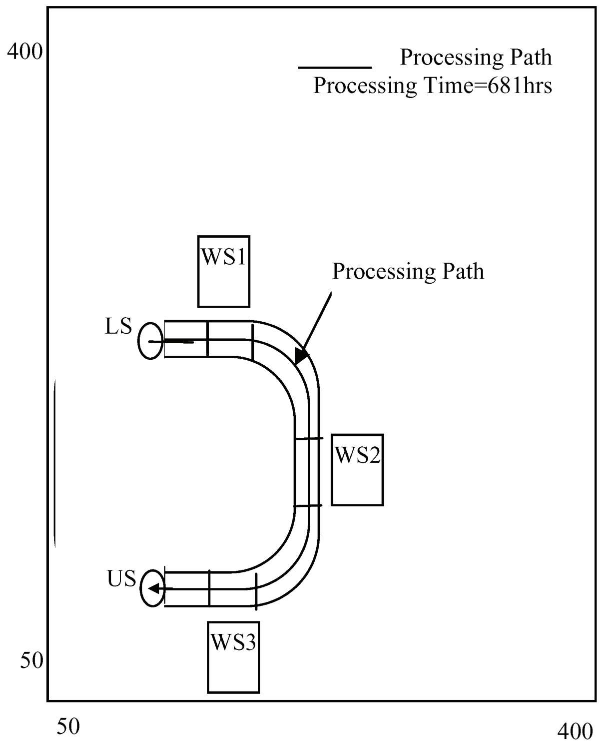

Fig. 4. The best of assembly line layout of meeting requirement 1.

3.2. Fulfillment of requirement 2

The proposed method has also been applied to provide three alternative processing paths. The possible layout for this case is given in Fig. 5. It has fulfilled the four objective functions. The fitness value of this layout is 4. The evaluation methods for the first three objective functions same as requirement 1.

The calculation of evaluation phase (evaluation of the objective functions for requirement 2) is listed in follows:

Decoded Chromosomes (Requirement 2):
(150, 220) (150, 252) (182, 284) (214, 284) (246, 252) (246, 220) (214, 220) (182, 220) (214, 220) (246, 2200 (278, 188) (246, 188) (246, 220) (246, 188) (246, 156) (246, 124) (214, 124) (182, 124) (150, 124).

- **Objective functions:**
  Objective no. 1 to no. 3:
  Evaluation Criteria: Same as requirement 1.
  Objective value (s_0): $s_0 = 3$

Objective no. 4: To evaluate the number of alternative paths.

Evaluation Criteria: If the layout has three alternative flow paths, the s_4 of the objective function will get 1, otherwise will get 0.

Objective value (s_0): $s_0 = 1$

- **Fitness Value:**

$$\text{eval}(v_k) = \sum_{o=1}^{\text{obj\_fun}} s_0 \tag{5}$$

Objective value: $s_0 = 4$.

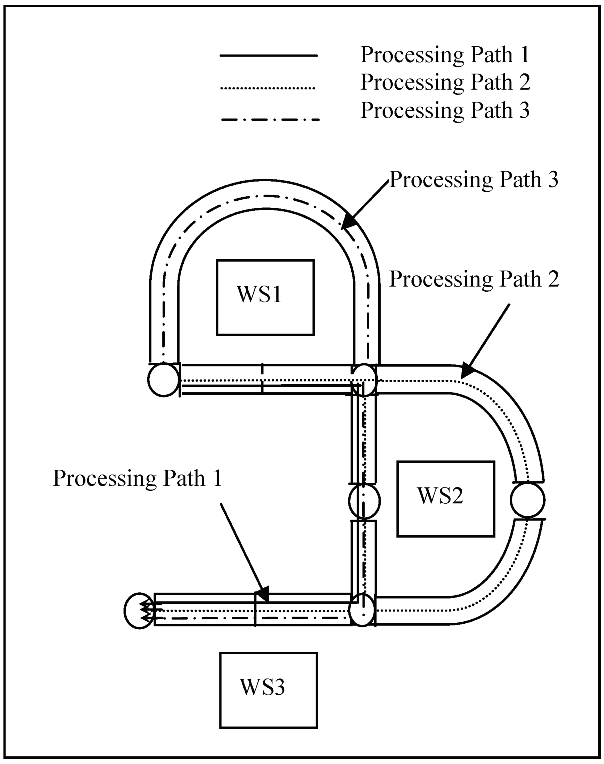

Fig. 5. The best of assembly line layout of meeting requirement 2.

4. DISCUSSION

The proposed genetic algorithm has been tackled two different requirements. The flexible assembly line system of fulfilling the requirements 1 is inflexible to cope with unplanned events occurred during the operation such as system component breakdown or suddenly call for product change. The operation has to stop. As the result, the production cost and time may be increased. The flexible assembly line systems of fulfilling the requirement 2, three alternative processing paths have been generated in the layout. The alternative paths pass through the each process workstations in a given sequence. Although the duplication of conveyor-components has increased the production cost, the flexible assembly line systems are able to deal with the change of production requirements and unplanned events occurred during assembly.

5. CONCLUSION

The proposed genetic algorithm is able to reconfigure a flexible assembly line system to meet the desire production requirements. The proposed approach offers a method for the selection amount of all possible flexible assembly line system configurations of satisfying given production requirements. The three evolutionary processes generate the layouts: selection, crossover, and mutation. The process of updating control parameters is integrated into the genetic algorithm to improve the performance and efficiency of the evolutionary processes. The reconfiguration of a flexible assembly line system to meet the requirements of minimization of the number of reconfigurable conveyor-components and the provision of alternative processes paths. The advantage of the proposed genetic algorithm doesn't rely past experience to reconfigure the flexible assembly line system.

REFERENCES

[1] Boncoi, Gh., Calefariu, G., Fota, A. (2000, 2001). *Sisteme de producţie*, vol. 2, Edit. "Transilvania" University, Braşov.

[2] Fota, A. (2003). *Cercetări privind dimensionarea şi configurarea sistemelor flexibile de fabricaţie pentru prelucrarea arborilor circulari*, Ph.D. Thesis, "Transilvania" University, Braşov.

[3] Fota, A. (2004). *Proiectarea sistemelor de maşini. Modelare şi simulare*, Edit. "Transilvania" University, Braşov.

[4] Goldberg, D. E. & Korb, B. Deb, K. (1989). *Messy genetic algorithms: motivation, analatsis, and first results*, Complex Systems, *3*, pp. 349–530.

[5] Ho, John K. L., Ranky, Paul G. (1997). *Object oriented modeling and design of reconfigurable conveyors in flexible assembly systems*, International Journal of Computer Integrated Manufacturing, **10**, *5*, September, pp. 360–379.

Author:

Dr. Eng. Adriana FOTA, Assoc. Prof., "Transilvania" University of Braşov, Economic Engineering and Manufacturing Systems Department, Romania, E-mail: fota.a@unitbv.ro