

# Optimization and Prediction Approaches for Flexible Routing in Dynamic Environment FMS

Jaber Abu Qudeiri<sup>1\*</sup>, Usama Umer<sup>1</sup>, Fayiz Abu Khadra<sup>2</sup>, H.M.A. Hussein<sup>3</sup>

<sup>1</sup>Advanced Manufacturing Institute, King Saud University, Saudi Arabia

<sup>2</sup>Faculty of Engineering, King Abdulaziz University, Saudi Arabia

<sup>3</sup>Department of Mechanical Engineering, Faculty of Engineering, Helwan University, Egypt

Copyright © 2015 by authors, all rights reserved. Authors agree that this article remains permanently open access under the terms of the Creative Commons Attribution License 4.0 International License

**Abstract** Recently, many production systems that work in dynamic environment are widely used in high volume industries. Increasing the product varieties in dynamic environment Flexible Manufacturing Systems (FMS) have gained more importance due to growing production lines' complexity. Increasing the customers demand to diversified products and rapid changing of the need to certain types of products require an FMS with high ability to adapt any change in production ratio during production. This paper introduces a Two Stages Approach (TSA) to increase the flexibility of the FMS to any change of the production ratios during production. The first stage of TSA is to propose a Genetic Algorithm (GA) based Production Simulator (GA-PS) to maximize the FMS throughput by optimizing the FLXible Routes (FLXR) for all products type at a given production ratio. The second stage is to propose a Neural Network (NN) approach to predict the routes of all products in FMS that can be used efficiently instead of a long, time-consuming GA-PS. TSA can improve the response of FMS to any change of the production ratio by finding a new FLXR that achieves the optimal throughput of the FMS within the new production ratios. Numerical examples will be applied to demonstrate applicability of the proposed approach. As a result, it could be ascertained that the proposed TSA is useful.

**Keywords** FMS, Flexible Routing, Production Simulator, Genetic Algorithms, Neural Network

production with entire range of parts. FMS typically operate at 70–80% utilization, which is more than three times higher than the utilization of traditional machines that can operate with as low as 20% utilization [2]. In order to achieve the above characteristics of FMSs, the FMS design, production planning, scheduling, and controlling stages should be optimized. Production scheduling is still one of the most important factors that can improve the productivity of FMS because the FMS is complicated system and have a many unpredictable conditions, especially in dynamic environments. These characteristics of FMS required an artificial intelligence and heuristic-based approaches to optimize the FMS scheduling. The ability of an FMS to produce a family of parts simultaneously can results in reduced inventories and faster responses to changes in demand requirements [3]. Many decision problems related to FMS must be answered by researchers [4]. Furthermore, most studies in the current literature have investigated the concept of flexibility in relation to a particular domain and a specific objective [5]. In FMS, scheduling problems become extremely complex, even for simple breakdowns, when dynamic uncertainties such as machine breakdowns and the uncertain arrival of jobs are taken into consideration [6]. One of the methods for studying the FMS is genetic algorithms (GA) [7, 8]. Increasing the customers demand to diversified products increases the use and the importance of FMS. Rapid change in the mood of consumers and change the need of the markets to certain types of products required an FMS with high ability adapt to any required change in production ratio.

In this paper, a Two Stages Approach (TSA) is proposed to increase the flexibility of the FMS to any change of the production ratio. TSA applies through two stages algorithms. In the first stage, the well-known Genetic Algorithms based on Production Simulator (GA-PS) is proposed. GA-PS generates a number of routes groups for all product types in FMS and the fitness corresponding to each group. The results of GA-PS are fed into the Neural Network (NN) and are used to train and validate the NN. Then, the well-trained NN can be applied promptly to predict the flexible routes in FMS. Numerical examples are given to demonstrate applicability of the proposed approach. The notations used in this paper are listed below.

## 1. Introduction

Variety of products produce by Flexible Manufacturing Systems (FMS) have gained more and more importance because of growing production lines' complexity and production costs and because an increasing demands for customized products. Conventionally, the product variety has been produced in a job shop or small batch operation. Job shop facilities are usually not suitable for large-volume mass production, because of low productivity and high production cost [1]. In contrast, FMSs achieve high-volume mass

**Notations**

|                             |  |
|-----------------------------|--|
| $i$                         | index of part type, $i=1, 2, \dots, l$   |
| $j$                         | operation index, $j= 1, 2, \dots, O$   |
| $k$                         | machine index, $k= 1, 2, \dots, M$   |
| $l$                         | No. of different part types  |
| $M_i$                       | Machine tool $i$   |
| $O_{ij}$                    | operation $j$ of part type $i$   |
| $PT_{ijk}$                  | machining time of operation $O_{ij}$ in machine $k$  |
| $ST_{ijk}$                  | setup time of operation $O_{ij}$ in machine $k$  |
| $WT_{ijk}$                  | waiting time of the product $i$ in front of the bay of machine $k$ to perform operation $O_{ij}$ |
| $TT_{ir}$                   | travel time of part time $i$ through route $R_i$   |
| $T_{load\backslash unload}$ | time required to load\unload product   |
| TH                          | throughput of the FMS  |
| $R_i$                       | route of part type $i$ .   |
| $n_i$                       | number of operations of product type $i$   |
| $T_s$                       | total time that product $s$ spends to complete its tour through the FMS.                         |
| $O_p$                       | No. of operations of part $p$ .  |
| $TT_{0 \rightarrow 1}$      | time to move from load\unload station to machine of first operation                              |
| $TT_{O_s \rightarrow 0}$    | time to move from last operation to load\unload station  |
| $TT_{a \rightarrow b}$      | time to move from location a to location b   |
| $F(i)$                      | fitness of individual number $i$   |
| NI                          | Number of individuals in each population   |
| $IN(i)$                     | individual $i$   |
| $PR(i)$                     | selection probability of individual $i$  |
| $P(k)$                      | period between fitness $F(k-1)$ and $F(k)$   |
| $POP(k)$                    | population $k$   |
| $s$                         | Number of elitist individuals  |
| $C_r$                       | Crossover rate   |
| $M_r$                       | Mutation rate  |
| $P$                         | number of parts to be produced   |

**2. Modeling FMS and Model Assumptions**

FMS is the major form of manufacturing system uses in high-volume mass production industries. FMS consider in this research consists of  $k$  different machine tools (CNC machining centers). Machine tools have finite local buffer capacity and uniform part delivery and holding pallets. All machine tools capable to process a required operation ( $O_{ij}$ ) on any part type, with different processing time ( $PT_{ijk}$ ). In addition, the products inter and leave the FMS through load\unload station and handling system. The part travels between the FMS component by a number of automated guided vehicles (AGVs) with one pallet at a time. A typical structure of an FMS is shown in Figure 1. The FMS in this study is defined by a number of assumptions. These assumptions are summarized below.

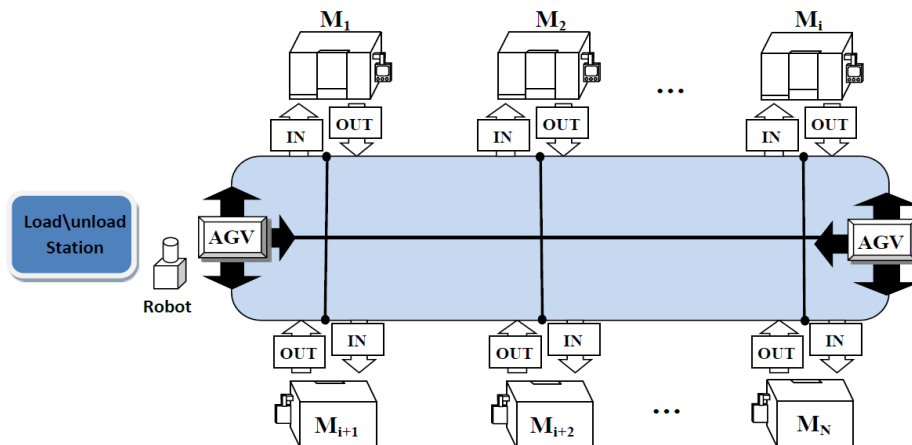


Figure 1. FMS model

**FMS model assumptions**

1. Large finite products variety can be produced by the FMS.
2. Handling of parts and materials in the FMS is done in single units (on unit-load pallets).
3. The products enter and leave the FMS through load/unload station
4. The time required to load and unload a part on/from the pallet is assumed to be 3 minutes. It needs one minute to move the pallet onto or from the AGV.
5. Each part is transferred on the unit-load pallet while in the system.
6. The layout of the FMS components is well defined, these components are fixed and the distance between each pair of them is known.
7. Buffer allocations are included in the system.
8. One operation can be performed in each machine tool of FMS at a time.
9. Tool change times are included in processing times.
10. All necessary components are available and fit in the system to execute any work order.
11. Processing time and setup time for each part type in each machine tool is known.
12. Some randomly selected machines can be considered as out of order.
13. Each machine is capable to perform different operations, but only one part can be processed at a time.
14. Once any operation is started it cannot be interrupted or divided.
15. Each machine tool stops  $N$  times an hour, and  $N < 5$ .  $N$  is randomly selected.
16. Each machine tool stops for a quality check every  $P$  parts.
17. It takes one minutes to travel between two neighbors machine tools in vertical and horizontal directions.

### 3. Problem Definition

The FMS considered in this paper can produce  $l$  different part types; each type requires  $O_s$  operations. Each operation can be processed by any machine tool ( $M_i \in M$ ) without any priority in machine tools. Parts can follow different routing among the machine tools to perform a job for  $P$  parts. The FMS scheduling problem is to find the optimal or near optimal route for each part type to maximize the throughput of the FMS. In this paper, GA is utilized to find efficient route that achieves the maximum throughput. The problem can be described mathematically as described below:

**Given:**

A set of machine tools  $M = \{M_1, M_2, \dots, M_K\}$ , and a set of jobs  $J = \{J_1, J_2, \dots, J_K\}$ , each job

$J_i$ ,  $\forall i = 1, 2, \dots, N$ , consists of  $n_i$  operations, each operation  $O_{ij}$ ,  $\forall (i = 1, 2, \dots, N \text{ and } j = 1, 2, \dots, n_i)$  can be processed in a machine tool  $M_i$ ,  $M_i \in M$  and a production ratio for all product types :  $P_1 : P_2 : \dots : P_l$

**Determine:** a route for each part type.

**So that,** TH ( $R_1, R_2, \dots, R_l$ ) is maximum.

In order to find the number of part that can be produced by a given time, the total time,  $T_p$ , which the product type  $p$  spends during its trip through the FMS from the moment it leaves the load/unload station until it returns back to load/unload station is needed. The time required to produce

product of type  $p$  when it travels through route  $R_p$  can be calculated as follows:

$$\begin{aligned}
 T_p = & T_{load/unload} + TT_{0 \rightarrow 1} + TT_{O_p \rightarrow 0} + TT_{(ir)} \\
 & + \sum_{j=1, k \in R_p}^{O_p} PT_{pjk} + \sum_{j=1, k \in R_p}^{O_p} ST_{pjk} \\
 & + \sum_{j=1, k \in R_p}^{O_p} WT_{ipk}
 \end{aligned}$$

### 4. Genetic Algorithms for the FMS Optimal Design

GA is a heuristic optimization method which is based on principles of natural selection and genetics Abu Qudeiri [9] and Goldberg [10]. GA starts with an initial population include a number of randomly generated solutions to the search problem. The set of solutions in the population is called individual. Each of the individuals represents one of the possible solutions for the optimization problem. The fitness is measured for all individual of the initial population. The next population is generated by selecting a number of individuals of higher fitness from the current population and by applying GA operations including, crossover and mutation. This iteration will repeat until the fitness value optimized. The best overall solution becomes the candidate solution to the problem. The flow chart of the GA is shown in Figure 2. The characteristics of the GA are introduced below.

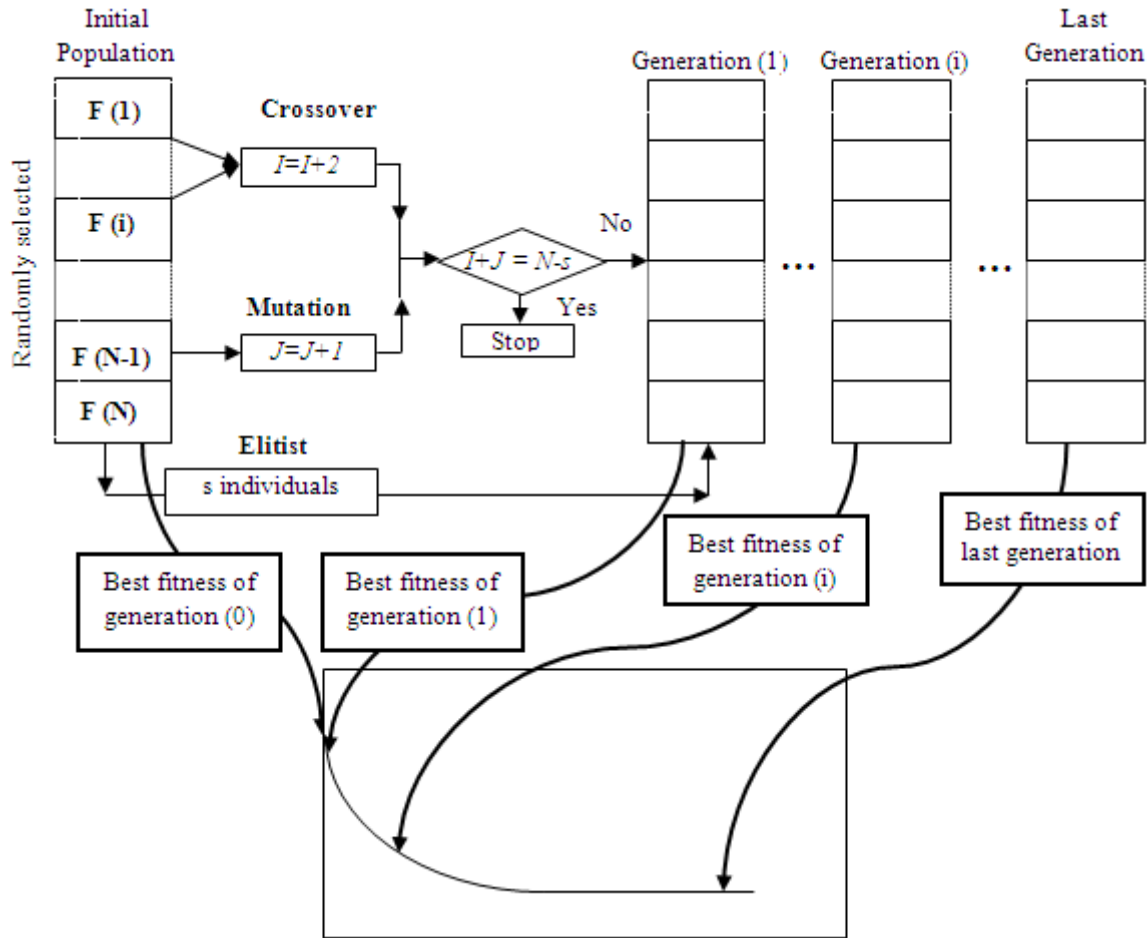


Figure 2 The GA outline

4.1. Matrix Encoding Method

The first step of GA is determination of the chromosome encoding method. The conventional linear genes encoding method is difficult to be used encode the complex arrangements of the route for all part products types. This is because there are many part types and each part type has different route. In this research, a Matrix Encoding Method (MEM) is proposed to express genes in individual. MEM encode the genes in chromosome as  $M \times N$  matrix, where  $M$  is the number of the part types (part variety) and  $N$  is the number of operations required to produce each type. MEM expresses the individuals according to the number of part types and the number of operation required to produce each type. The individual using MEM is defined as follows.

$$Individual = \begin{bmatrix} M_1^1 & M_2^1 & L & M_i^1 & L & M_o^1 \\ M_1^2 & M_2^2 & L & M_i^2 & L & M_o^2 \\ M & M & M & M & M & M \\ M_1^j & M_2^j & L & M_i^j & L & M_o^j \\ M & M & M & M & M & M \\ M_1^l & M_2^l & L & M_i^l & L & M_o^l \end{bmatrix} \quad (1-a)$$

Where  $M_i^j$  is the machine that will operate operation  $i$  of the part type  $j$ , and  $l$  is number of part types and  $O_s$  is number of operations required for part  $s$ . Equation (1-b) shows an example of the MEM encoding method

$$\begin{bmatrix} M_3 & M_5 & M_1 & M_2 & M_7 & M_4 & M_6 \\ M_7 & M_1 & M_4 & M_2 & M_5 & M_6 & M_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ M_2 & M_1 & M_5 & M_7 & M_3 & M_6 & M_4 \end{bmatrix} \quad (1-b)$$

The expression matrix is not limited and it can be defined with any number of rows and columns, thus, any FMS structure and with any part varieties can be dealt with.

4.2. Initial Population and Selection of the Next Generations

The first group generation,  $POP(0)$ , is selected randomly. Each group contains  $N$  individuals, and each individual contains  $l$  vectors and each vector contains  $O$  elements. Based on current population, GA generates the next generation, by applying the following algorithm.

| <b>Algorithm 1</b> generation of new population   |  |
|---|--|
| <b>Step 1:</b> Calculate $F(i) \forall i = 1, 2, \dots, IN$ for current population  |  |
| <b>Step 2:</b> Select $IN(i) \forall i = 1 \rightarrow s$ by applying elitist strategy  |  |
| <b>Step 3:</b> For each individual $i \in$ current population and $i = (1, \dots, NI)$<br>Calculate the probability of selecting $PR(i)$ as follows |  |
|   | $PR(i) = \frac{F(i)^2}{\sum_{i=1}^{NI} F(i)^2} \quad (2)$  |
| <b>Step 4:</b> Calculate accumulation $A(i) \forall i = 1, 2, \dots, IN$ by using Equation (3)  |  |
|   | $A(i) = \sum_{i=1}^i PR(i) = \frac{\sum_{i=1}^i F(i)^2}{\sum_{i=1}^{NI} F(i)^2} \quad (3)$   |
| <b>Step 5:</b> Calculate period $P(i) \forall i = 1, 2, \dots, IN$ as follows:  |  |
|   | $P(0) = [0, A(1)] \text{ and } P(i) = [A(i-1), A(i)], \forall i = 1, 2, \dots, IN$   |
| <b>Step 6:</b> Carry out crossover operation on the $IN(i)$ and $IN(j)$   |  |
| <b>Step 7:</b> Define new endpoints of the two selected periods by a constant value $n$ as follows:   |  |
|   | $\text{Set } P(i) = [A(i-1), A(i) - n] \text{ for } P(i) \subset N1 \text{ and}$<br>$P(j) = [A(j-1), A(j) - n] \text{ for } P(j) \subset N2$ |
| <b>Step 8:</b> Select an individual $IN(i) \in P(k)$ and carry out mutation operation.  |  |
| <b>Step 9:</b> Reduce the endpoint of the selected period by a constant value $n$ as follows:   |  |
|   | $\text{Set } P(i) = [A(i-1), A(i) - n] \text{ for } P(i) \subset N1.$  |
| <b>Step 10:</b> Repeat steps 6 to 9 to generate $N - s$ individuals of the new population based on $C_r$ and $M_r$ .                                |  |
| <b>Step 11:</b> Repeat step 1 to step 10.   |  |
| <b>Step 12:</b> Repeat step 11 to reach the optimum fitness value and set the individual of this fitness as the optimal individual.                 |  |

### 4.3. Crossover by MEM

MEM crossover is different from conventional crossover. MEM is applied using crossover line that passes through crossover points in the vectors, as shown in Figure 3. The MEM crossover is carried out using the following steps.

| <b>Algorithm 2:</b> Crossover for MEM   |  |
|---|--|
| <b>Step 1:</b> Select two numbers between 0 and $A(NI)$ as follows:<br>$N_1 \leftarrow \text{Random} [0, \dots, A(NI)]$ and $N_2 \leftarrow \text{Random} [0, \dots, A(NI)]$<br>If $N_1$ and $N_2 \in P(i), \forall i = 1, 2, \dots, NI$ Then, reselect $N_2$ |  |
| <b>Step 2:</b> Find $IN(i) \in P(i) \subset N1$ and $IN(j) \in P(j) \subset N2, \forall i, j = 1, 2, \dots, IN$   |  |
| <b>Step 3:</b> Select $l$ number of crossover points, $CP_i, \forall i = 1, 2, \dots, l$ as follows:<br>$CP_j \leftarrow \text{Random} [1, \dots, i, \dots, O - 1]$   |  |
| <b>Step 4:</b> Exchange the genes after crossover line between the two individual.  |  |

Figure 3 illustrate MEM crossover

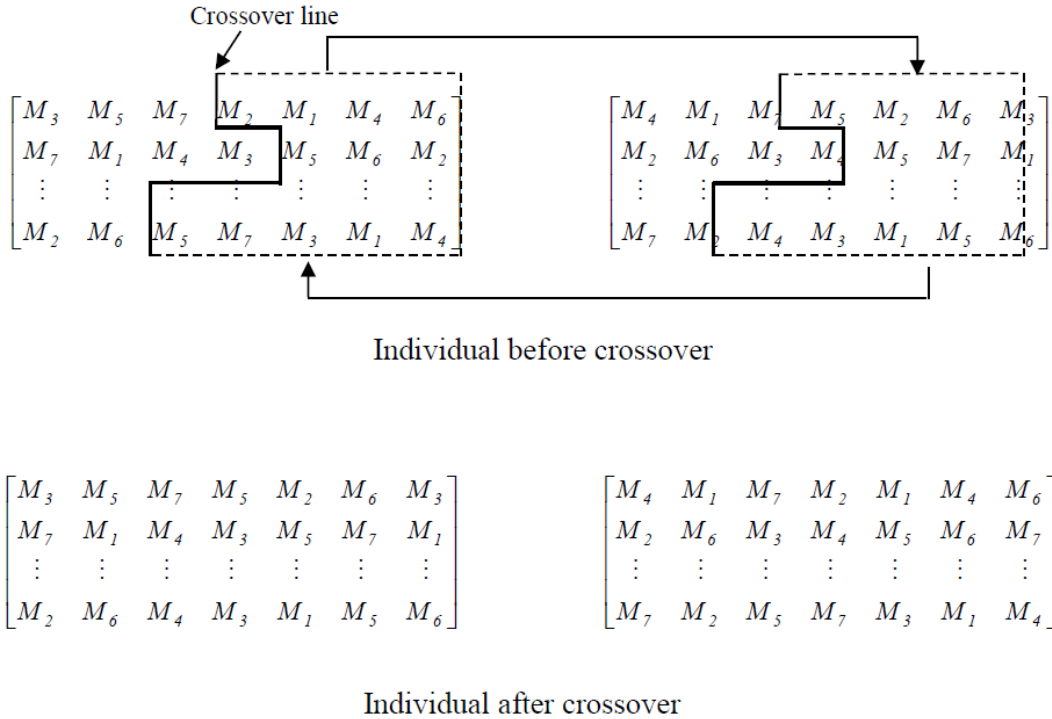


Figure 3. MEM crossover operations

4.4. Mutation by MEM

Mutation operation for the MEM is applied as follows.

**Algorithm 3:** Mutation for MEM

**Step 1:** Select one number between 0 and  $A(NI)$  as follows:  
 $N \leftarrow \text{Random} [0, \dots, A(NI)]$

**Step 2:** Find  $IN(i) \in P(i) \subset N1$

**Step 3:** Select two genes in each of the individual vectors as follows  
 $a, b \leftarrow \text{Random} [1, \dots, NI]$

**Step 4:** Swap the location of the two genes in each vector.

Figure 4 shows the mutation graphically.

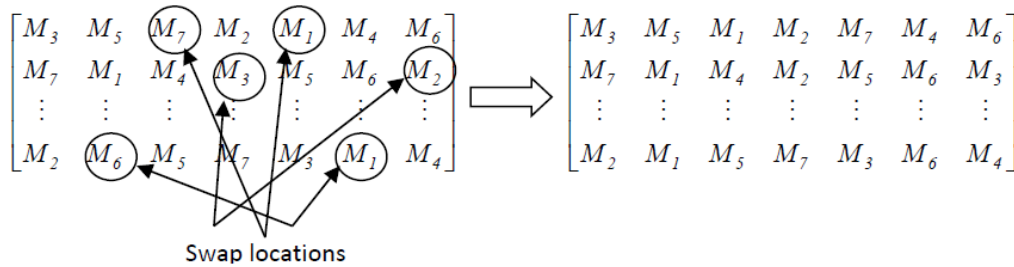


Figure 4. Swap mutation

5. GA based Production Simulator

To find the route of each part type the FMS that maximizes the throughput of the FMS, PS and GA combination is proposed. The GA-PS structure is shown in Figure 5. GA-PS searches the route repeat the cycle including the discrete simulation and apply the GA in each run of simulation. The characteristics of the FMS model are entered into the PSS. Based on this input, the simulator gives the route for given input. The PSS algorithm is applied using following steps.

**Algorithm 4: PSS algorithm**

**Step 1:** Generate randomly the initial population with  $N$  individuals.  
**Step 1:** Generate POP(0) as follows:

$$\text{Select } IN(i) \leftarrow \text{Random} \begin{bmatrix} M_1^1 & \dots & M_{Os}^1 \\ \vdots & \vdots & \vdots \\ M_1^p & \dots & M_{Os}^p \end{bmatrix}, \forall i = 1 \rightarrow NI$$

**Step 2:** Read all characteristics of the desired FMS.  
**Step 3:** Calculate  $F(i) \forall i = 1, 2, \dots, NI$  for current population  
**Step 4:** Select  $IN(i) \forall i = 1 \rightarrow s$  by applying elitist strategy.  
**Step 5:** Calculate  $PR_i, \forall i = 1, 2, \dots, NI$   
**Step 6:** Carry out GA operations.  
**Step 7:** If  $(C_r \times NI + M_r \times NI) = NI - s$ , then, continue to step 8, if not, return to step 6.  
**Step 8:** Regard the generated individuals as the next generation population.  
**Step 9:** Repeat until fitness reaches its maximum value.

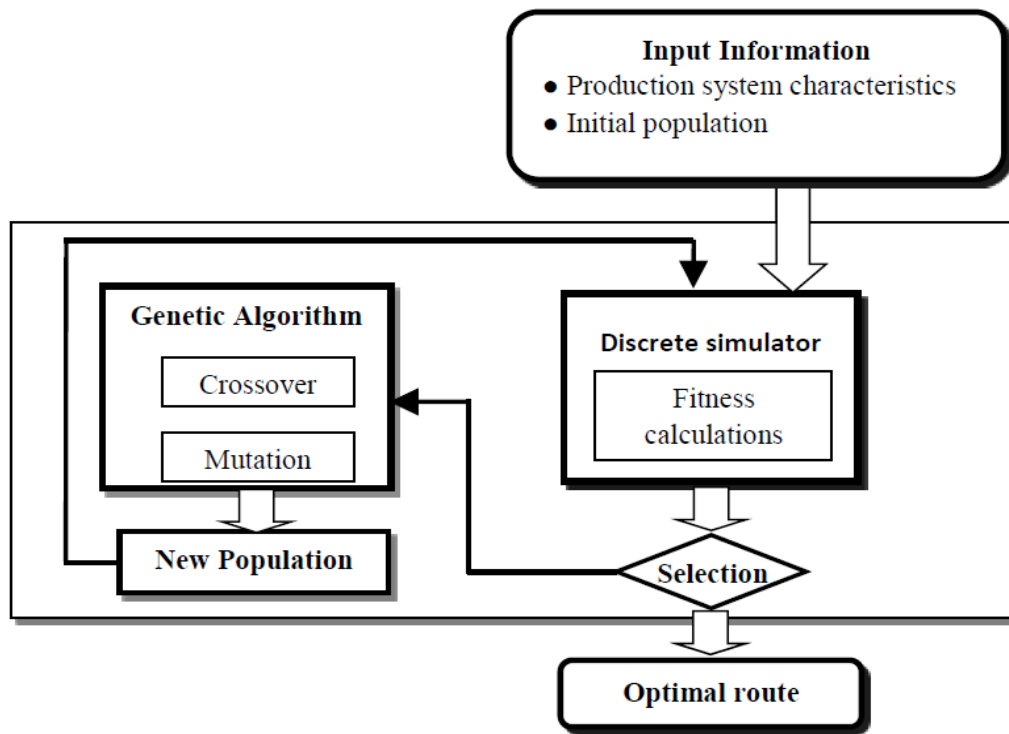


Figure 5. Production simulator system

## 6. Prediction the Flexible Routing

To reduce the computational time for the routes determination required by the GA-PS, NN is introduced. NN learns the relationships between input data and output data and generalizes those relationships to unseen data. In this stage the NN is trained and introduced to predict the routes of product produced by a certain FMS. The algorithm of this stage is carried out using the following steps.

**Algorithm 5: NN procedure**

**Step1:** Construct the NN structure  
**Step2:** Train the NN with some the routes resulted by GA-PS in the first stage.  
**Step3:** Validate the NN by the rest of routes.

## 7. Case Studies

### 7.1. Simulation Example

To show efficacy of the proposed simulation and prediction models, a simulation has been performed. The simulation is run to produce a specific number of products. The FMS consists of

8 machine tools and it is capable to produce four different types of products with the following production ratios: P1:P2:P3:P4 = 1:2:4:5. Each of the four types requires ten operations. The processing times and the setup times are randomly generated as follows: Production times between four and ten minutes, and setup times between two and four minutes.

**7.2. GA and NN Parameters**

The following GA parameters are selected after run the simulation several trials:

Population size of 100, probability of crossover of 0.9 and probability of mutation of 0.05 are used. The simulation is run to 500 generations. To train the NN, the three layers: input, hidden and output are selected; the three layers are 24, 23, and 1 respectively.

**7.3. Simulations Results**

Based on the GA-PS results, the routes of all products type are optimized, the routes and the maximum fitness obtained from GA-PS and those predicted by NN are given in Tables 1.

**8. Conclusions**

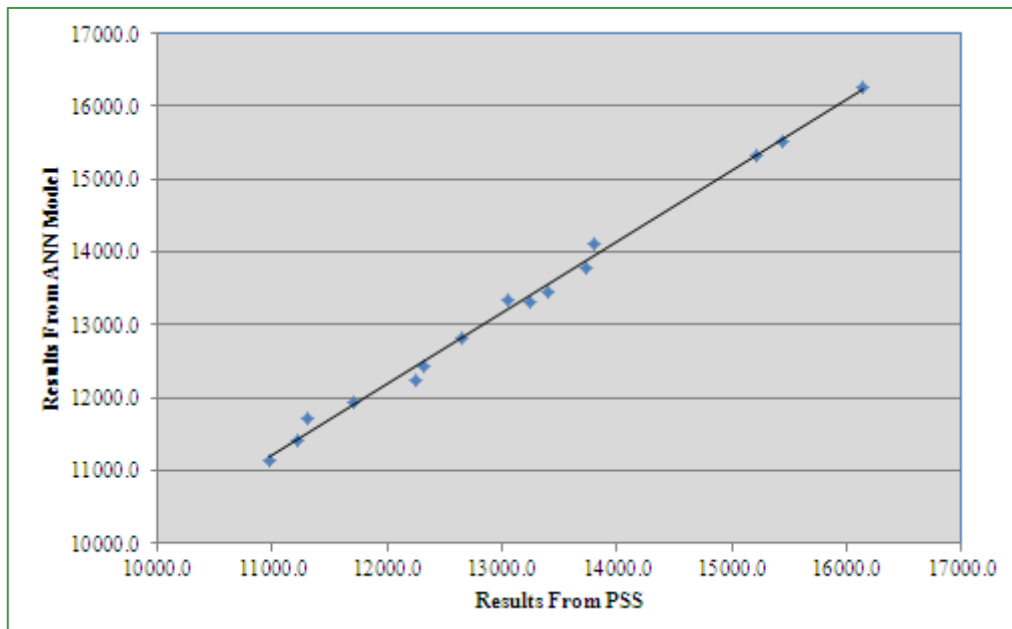
In this paper, we presented TSA to increase the flexibility of the FMS to any change of the production ratios during production to find proper routes for all product types in FMS. TSA used PSS generated a set of routes groups. The generated routes groups were used to train ANN utilized GA-PS to find optimal Flexible routes for all products type at a given production ratio. Unique genetic operations were applied for efficient used of GA.

The GA-PS results were used to train the NN. The trained NN were then used instead of GA-PS to predict the routes of all product types. By this way, a long simulation time required by the simulation can be eliminated. Numerical simulation showed that the results of the simulation and the trained NN approach are closed to each other and prediction approach can be used efficiently instead of the simulation.

**Table 1.** Comparison fitness obtained from GA-PS and those predicted by NN

| Production Ratio | Number of Machines | Routes   | Best Fitness    |                  |
|------------------|--------------------|--|-----------------|------------------|
|                  |                    |  | Acquired by PSS | Predicted by ANN |
| 1:2:4:5          | 8                  | Type 1 route: 2→5→4→5→3→2→1→2→5→3<br>Type 2 route: 4→8→8→7→7→2→3→7→1→8<br>Type 3 route: 6→3→4→1→7→8→3→3→4→7<br>Type 4 route: 4→5→6→3→7→8→7→8→6→6 | 10.978E003      | 11.1368E003      |

The regression analyses of outputs from the GA-PS and NN approach for the simulation is shown in Figure 6.



**Figure 6.** Regression analyses of outputs from the ANN and PSS methods



## Acknowledgements

The authors would like to thank Fatima Alnijris Research Chair for Advanced Manufacturing Technology for the financial support provided for this research.

---

## REFERENCES

- [1] Groover, M. P. 1980. Automation, production systems, and computer-aided manufacturing. Englewood Cliffs, N.J.: Prentice-Hall.
- [2] Mahdavi, I., A. H. F. M. Azar, and M. Bagherpour. 2009. Applying fuzzy rule based to flexible routing problem in a flexible manufacturing system. Paper read at Industrial Engineering and Engineering Management, 2009. IEEM 2009. IEEE International Conference on, 8-11 Dec. 2009.
- [3] Gershwin, S. B. 1994. Manufacturing systems engineering. Englewood Cliffs, N.J.: PTR Prentice Hall.
- [4] Sabuncuoğlu, I., and M. Lahmar. 2003. An Evaluative Study of Operation Grouping Policies in an FMS. International Journal of Flexible Manufacturing Systems 15 (3):217-239.
- [5] Shuiabi, E., V. Thomson, and N. Bhuiyan. 2005. Entropy as a measure of operational flexibility. European Journal of Operational Research European Journal of Operational Research 165 (3):696-707.
- [6] Bilkay, O., O. Anlagan, and S. E. Kilic. 2004. Job shop scheduling using fuzzy logic. The International Journal of Advanced Manufacturing Technology The International Journal of Advanced Manufacturing Technology 23 (7-8):606-619.
- [7] Qudeiri, J. A., H. Yamamoto, and R. Ramli. 2009. Buffer size decision for Flexible Transfer Line with Rework Paths using Genetic Algorithm. IJISTA International Journal of Intelligent Systems Technologies and Applications 7 (2).
- [8] Ficko, M., M. Kovacic, and M. Brezocnik. 2004. Genetic algorithms : a useful optimization method for manufacturing problems. *Academic Journal of Manufacturing Engineering* 2:21-26.
- [9] Abu Qudeiri, J. Y. H. R. R. J. A. 2008. Genetic algorithm for buffer size and work station capacity in serial-parallel production lines. *Artificial Life and Robotics* 12 (1-2):1-2.
- [10] Goldberg, D. E. 1989. *Genetic algorithms in search, optimization, and machine learning*. Reading, Mass.: Addison-Wesley Pub. Co.
- [11] Altiok, T. 1997. *Performance analysis of manufacturing systems*. New York: Springer.
- [12] Askin, R. G., and C. R. Standridge. 1993. *Modeling and analysis of manufacturing systems*. New York: Wiley.
- [13] Buzacott, J. A., and J. G. Shanthikumar. 1993. *Stochastic models of manufacturing systems*. Englewood Cliffs, N.J.: Prentice Hall.
- [14] Dallery, Y., and S. B. Gershwin. 1992. Manufacturing flow line systems: a review of models and analytical results. *Queueing Syst Queueing Systems* 12 (1-2):3-94.
- [15] Heavey, C., H. T. Papadopoulos, and J. Browne. 1993. The throughput rate of multistation unreliable production lines. *European Journal of Operational Research European Journal of Operational Research* 68 (1):69-89.
- [16] Helber, S. 1999. Performance Analysis of Flow Lines with Non-Linear Flow of Material. *Lecture notes in economics and mathematical systems*. (473):ALL.
- [17] Papadopoulos, C. T., C. Heavey, and J. Browne. 1993. *Queueing theory in manufacturing systems analysis and design*. London; New York: Chapman & Hall.
- [18] Papadopoulos, H. T., and C. Heavey. 1996. Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines. *EOR European Journal of Operational Research* 92 (1):1-27.
- [19] Perros, H. G. 1994. *Queueing networks with blocking : exact and approximate solutions*. New York: Oxford University Press.
- [20] Seong, D., S. Y. Chang, and Y. Hong. 1995. Heuristic algorithms for buffer allocation in a production line with unreliable machines. *International journal of production research*. 33 (7):1989.