International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:6, No:9, 2012

# Solving Part Type Selection and Loading Problem in Flexible Manufacturing System using Real Coded Genetic Algorithms – Part II: Optimization

Wayan F. Mahmudy, Romeo M. Marian, and Lee H. S. Luong

***Abstract*****—**This paper presents modeling and optimization of two NP-hard problems in flexible manufacturing system (FMS), part type selection problem and loading problem. Due to the complexity and extent of the problems, the paper was split into two parts. The first part of the papers has discussed the modeling of the problems and showed how the real coded genetic algorithms (RCGA) can be applied to solve the problems. This second part discusses the effectiveness of the RCGA which uses an array of real numbers as chromosome representation. The novel proposed chromosome representation produces only feasible solutions which minimize a computational time needed by GA to push its population toward feasible search space or repair infeasible chromosomes. The proposed RCGA improves the FMS performance by considering two objectives, maximizing system throughput and maintaining the balance of the system (minimizing system unbalance). The resulted objective values are compared to the optimum values produced by branch-and-bound method. The experiments show that the proposed RCGA could reach near optimum solutions in a reasonable amount of time.

***Keywords*****—**Flexible manufacturing system, production planning, part type selection problem, loading problem, real-coded genetic algorithm

## I. Introduction

FMS is integrating hardware and software elements and defined as 'a collection of production equipment logically organized under a host computer and physically connected by a central transport system' [1]. The hardware elements are made up of computer numerically controlled (CNC) machines equipped with tool magazine, pallet, loading and unloading station, buffer for processing parts, material transport and handling equipment such as automated guide vehicle (AGV) and conveyor [2][3]. The software elements consist of standard FMS software from supplier such as CNC program and traffic management software and may be enhanced by specific software required by user [1][4][5].

FMS have emerged as the response of a rapid change of consumer's demand on a wide variety of products in low to medium volumes, shortening of product lifetimes and the increasing competition in national and global market [6][7].

W. F. Mahmudy is a PhD student at School of Advanced Manufacturing & Mechanical Engineering, University of South Australia and a lecturer at Department of Computer Science, Brawijaya University (UB), Indonesia (e-mail: wayan_firdaus.mahmudy@mymail.unisa.edu.au, wayanfm@ub.ac.id).

R. M. Marian, PhD is Senior Lecturer and Program Director at School of Advanced Manufacturing & Mechanical Engineering, University of South Australia, Australia (e-mail: romeo.marian@unisa.edu.au).

L. H. S. Luong is a professor at School of Advanced Manufacturing & Mechanical Engineering, University of South Australia, Australia (email: lee.luong@unisa.edu.au).

These objectives may be achieved by the flexibility of FMS which lead to higher utilization of resources such as machines and tools. Generally, the flexibility of FMS can be divided into two categories which may be divided into several sub categories. The first is *machine flexibility*. By using high technologies, machines configuration can be changed easily by attaching different tools to produce new type of products for different market segments [7][8]. The second is *routing flexibility*. Higher productivity and profitability can be achieved by enabling flexibility of production routes. It means that one product may be produced by a number of alternative machines and as the result is the increase of a machines utilization and the decrease of a processing time [9][10].The FMS has a potential as a strategic tool in manufacturing industries and its successful implementation depends to the quality of its production planning. Therefore, an appropriate production planning for the FMS must be established to adapt with the increasing automation and complexity of manufacturing systems [10]. The production planning is conducted to ensure an efficient production process. Its important role in determining the responsiveness and the efficiency of the FMS make production planning a promising research area [6][11].There are several issues in the production planning stage such as part type selection problem, machine grouping problem, production ratio problem, resource allocation problem, and loading problem [12][13]. Definition of the part type selection and loading problem has been discussed in the first part of this paper. The machine grouping problem deals with arrangement of similar machines into identical machines groups so each machine on the same group could perform the same operations. The production ratio problem determines the ratio of selected set of part type should be produced over time. Resource allocation problem deals with allocation of the limited number of pallets and fixtures to the part types. Depend on the specific characteristic of manufacturing environments, various combination of some production planning problems have been considered in the literatures. For example, Bilgin & Azizoglu [14], Chan & Swarnkar [15], and Chen & Ho [16] solved the machine loading problem. Swarnkar & Tiwari [17], Choudhary, Tiwari & Harding [18], Biswas & Mahapatra [7], Ponnambalam & Kiat [19], and Prakash et al. [20] solved the part type selection and loading problem simultaneously. Tabucanon, Batanov & Basu [21] solved the part type selection and loading problem simultaneously in the first stage and used the result on this stage to determine the production ratio in the next stage. However, the routing flexibility was not considered. Kim et al. [22] solved the loading problem and the partial machine grouping while considering tool life constraints.

TABLE I
TEST-BED PROBLEMS

| problem | num. of part types | num. of machines | num. of tool types | scheduling period |
|---|---|---|---|---|
| 1 | 8 | 4 | 20 | 4000 |
| 2 | 8 | 5 | 25 | 4000 |
| 3 | 10 | 4 | 20 | 4000 |
| 4 | 10 | 5 | 25 | 4000 |
| 5 | 16 | 4 | 20 | 7000 |
| 6 | 16 | 5 | 25 | 7000 |
| 7 | 18 | 4 | 20 | 7000 |
| 8 | 18 | 5 | 25 | 7000 |
| 9 | 24 | 4 | 20 | 10000 |
| 10 | 24 | 5 | 25 | 10000 |
| 11 | 26 | 4 | 20 | 10000 |
| 12 | 26 | 5 | 25 | 10000 |

Seok Shin, Park & Keun Kim [23] solved the loading problem while considering a various flexibility such as machine, sequence, tool and process routing. This paper focuses on the part type selection and loading problem with machine and tool flexibility.

A various approaches have been proposed to solve the optimization of the production planning problems such as mathematical programming [21], Lagrangean relaxation approach [14], genetic algorithms [16][18], particle swarm optimization [7][19], ant colony optimization [15], immune algorithm [20], two-stage heuristics based on a bin-packing algorithms and a simple search technique [22], multi-agent system [24], and symbiotic evolutionary algorithm [23]. A combination of two methods was also used such as hybridizing genetic algorithm with simulated annealing [25][26], and hybrid tabu search and simulated annealing-based [17]. Here, heuristic methods is widely used since direct methods which are based on mathematical programming and smart enumeration are not practical to solve these complex problems [27]. This paper proposes a GA which uses an array of real numbers as chromosome representation so the GA can be called the real coded GA (RCGA).

## II. REVIEW OF PART I

In part I, we considered a FMS which consists several machines. The machines can perform different operations when they are equipped with different tools. Each part type has a production requirement in form of sequence of operations. Each operation can be processed on several alternative machines with several alternative tools. Time needed for parts' operations depend on the assigned machine. Here, the FMS has machine and tool flexibility. Two common objectives considered in literatures, maximizing system throughput and maintaining the balance of the system (minimizing system unbalance), were explained.

TABLE II
RANDOMLY GENERATED PARAMETERS

| Parameters | Range |
|---|---|
| tool slot capacity of each machine | 40-60 |
| number of copies of each tool type | 2-(nMac-1) |
| number of slots required for each tool type | 3-7 |
| number of operations of each part type | 2-(nMac) |
| batch size of each part type | 40-60 |
| value of each part type (dollar) | 5-10 |
| number of possible machines for each operation | 1-3 |
| processing time of each operation | 20-40 |
| number of tool types required for each operation | 2-5 |

nMac: number of machines

A chromosome construction for the real-coded GA (RCGA) was also explained. The chromosome is a vector of real number whose size is same with the number of part types. Two crossover methods (*flat-crossover* and *extended-intermediate-crossover*) and two mutation methods (*random exchange mutation* and *simple-random-mutation*) were used to produce new generations. A fitness function which is used to measure the goodness of the solution was constructed by using two objective functions of the optimization of the part type selection and machine loading problem, minimizing system unbalance and maximizing system throughput. A simple problem set was given to demonstrate how the proposed RCGA solved the problem and produced an optimum solution in reasonable amount of time.

## III. RESULT AND DISCUSSION

To evaluate the performance of the RCGA, we generate 12 test-bed problems as shown in Table 1. Problems 1 to 4 are considered as small size problems, problems 5 to 8 are medium size problems and problems 5 to 8 are large size problems. Lengths of scheduling period for all machines are same within each problem. The other randomly generated parameters are shown in Table 2.

The RCGA is implemented in Java and experiment is carried out on personal computer equipped with Intel® Core™ i3-380 processor working at speed 2.53 GHz. The first step in our experiment is determining the most suitable selection method for the RCGA. Four common selection methods (roulette wheel, binary tournament, elitist, and replacement) are examined.

*The other parameters are set as follows:*

- Crossover rate is 0.25.
- Mutation rate is 0.05
- Population size is 500, 1000 and 1500 for small size problems, medium size problems and large size problems respectively.
- The weighted parameters are $\alpha_1$=3 and $\alpha_2$=1.
- GA iterations will be stopped after 5,000 successive generations no longer produces better results.

TABLE III
COMPARISON AMONG SELECTION METHODS

| selection | $F_{min}$ | $F_{max}$ | $F_{avg}$ | time | itr best |
|---|---|---|---|---|---|
| roulette wheel | 2.1264 | 2.3104 | 2.1875 | 58.5 | 3593 |
| binary tournament | 2.2250 | 2.4175 | 2.3488 | 24.7 | 1980 |
| elitist | 2.1962 | 2.4101 | 2.3344 | 26.7 | 1284 |
| replacement | 2.3331 | 2.4178 | 2.3659 | 38.9 | 7181 |

TABLE IV
COMPARISON AMONG CROSSOVER RATES

| crossover rate | mutation rate | average of fitness value |
|---|---|---|
| 0.00 | 0.40 | 2.377 |
| 0.05 | 0.35 | 2.376 |
| 0.10 | 0.30 | 2.392 |
| 0.15 | 0.25 | 2.388 |
| 0.20 | 0.20 | 2.393 |
| 0.25 | 0.15 | 2.384 |
| 0.30 | 0.10 | 2.400 |
| 0.35 | 0.05 | 2.366 |
| 0.40 | 0.00 | 2.294 |

By using these parameters and data from problem 7, we run the RCGA 10 times and obtain results of minimum ($F_{min}$), maximum ($F_{max}$), and average ($F_{avg}$) of fitness values as shown in Table 3. The average computation time (*seconds*) and number of iterations to obtain the best solution (*itr best*) are also presented. Here, the replacement selection method produces a higher of average of fitness value than other methods. Therefore, we use this selection method in the next step of the experiment.

By using the replacement selection, the RCGA can maintain the population diversity and explore the search space better. It is indicated by its significantly higher number of iterations to obtain the best solution. In contrast, the other selection methods achieve their convergence faster which may indicate that they are trapped in local optimum areas and cannot obtain a better solution. Figure 2 depicts a one run from each selection method. It shows the improvement of the best fitness value along generations. While all other selection methods achieve their convergence in less than 2000 generations, the replacement selection gradually improve its chromosomes to obtain higher fitness value.

The second step in our experiment is determining the most suitable crossover rate and mutation rate for the RCGA. Appropriate crossover rate and mutation rate will help the RCGA to balance its exploration and exploitation ability and avoid the premature convergence [28]. In order to get a fair result, we vary the crossover rate (*cr*) from 0 to 0.4 and set the mutation rate (*mr*) in such way that *cr*+*mr*=0.4. Here, all runs produce 0.4x1000 offspring in each generation. Again, we run the RCGA 10 times using problem 7.

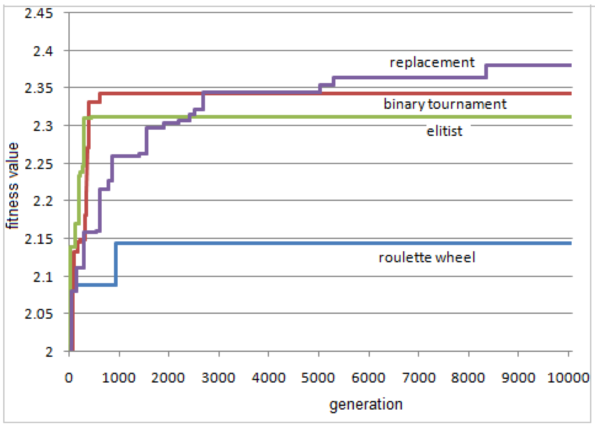


Fig. 1 The best fitness value for each selection method

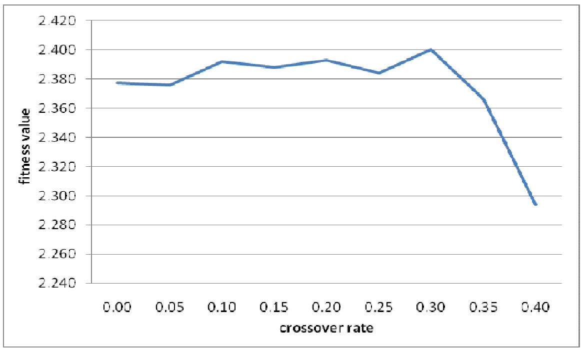


Fig. 2 Fitness values for various crossover rates

The result is presented in Table 4 and Figure 2. Apparently, the best result is produced by using crossover rate of 0.3 and mutation rate of 0.1. Here, by using a low value of crossover rate the RCGA will greatly depend on its mutation rate and tend acting as a random search method. In other hand, the RCGA will lose its ability to maintain population diversity if using a high crossover rate and a low mutation rate. Inability to maintain population diversity means that the RCGA cannot explore the search space effectively and will likely be trapped in local optimum area.

After determining the most suitable crossover rate and mutation rate for the RCGA, we run the RCGA for all test-bed problems. To measure the performance of the RCGA we use frequency of achieving optimum solution (FOS) and deviation of objective values resulted by GA to its optimum values. The optimum solutions are obtained by using branch-and-bound method. It is should be noted that branch-and-bound method required computational time more than 10 hours to solve particular test bed problems which cannot be accepted on daily operation of the FMS. Equation (1) shows the deviation of average fitness values from 20 runs of GA to optimum fitness value. $F_{opt}$ is fitness value obtained by branch-and-bound method. $FGA_r$ is fitness value obtained by GA in run *r*.

$$F_{dev} = \left| \frac{\left( F_{opt} - \left( \sum_{r=1}^{20} FGA_r \right) / 20 \right)}{F_{opt}} \right| \times 100\,\% \qquad (1)$$

TABLE V
COMPUTATIONAL RESULTS

| problem | RCGA | | | | | Optimum values | | | $F_{dev}$(%) |
|---|---|---|---|---|---|---|---|---|---|
| | *FOS* | *time (seconds)* | *F* | *TH* | *SU* | *F* | *TH* | *SU* | |
| 1 | 20 | 4.29 | 2.545 | 1,616.0 | 803.0 | 2.545 | 1616 | 803 | 0.00 |
| 2 | 20 | 5.09 | 2.926 | 2,591.0 | 9,838.0 | 2.926 | 2591 | 9838 | 0.00 |
| 3 | 20 | 5.24 | 2.972 | 3,058.0 | 6,858.0 | 2.972 | 3058 | 6858 | 0.00 |
| 4 | 20 | 6.26 | 2.531 | 2,196.0 | 3,233.0 | 2.531 | 2196 | 3233 | 0.00 |
| 5 | 12 | 16.47 | 2.133 | 2,604.0 | **3405.6** | 2.156 | 2,676 | 3,738 | 1.06 |
| 6 | 13 | 23.63 | 1.936 | 2,587.9 | 7,951.6 | 1.968 | 2,605 | 7,126 | 1.59 |
| 7 | 2 | 22.21 | 2.404 | 3,321.9 | **3,543.1** | 2.458 | 3,595 | 5,529 | 2.23 |
| 8 | 17 | 19.86 | 2.077 | 2,861.4 | 4,997.4 | 2.088 | 2,871 | 4,768 | 0.51 |
| 9 | 3 | 84.09 | 2.260 | 3,940.9 | 4,832.4 | 2.349 | 4,150 | 4,204 | 3.79 |
| 10 | 8 | 59.06 | 1.803 | 3,179.6 | **10,666.6** | 1.809 | 3,212 | 10,879 | 0.34 |
| 11 | 0 | 84.29 | 2.248 | 4,286.7 | 6,077.9 | 2.305 | 4,417 | 5,519 | 2.45 |
| 12 | 1 | 92.36 | 1.971 | 3,893.3 | 10,956.0 | 2.018 | 3,937 | 9,291 | 2.31 |
| average | | 35.25 | | | | | | | 1.19 |

The computational results are presented in Table 5. Column 'time' shows average of computation time (seconds) from 20 runs of the RCGA. Columns 'F', 'TH' and 'SU' below column 'RCGA' show the average of fitness value, throughput and system unbalance obtained from 20 runs of the RCGA.

Apparently for a small number of part types (8 and 10), the proposed real coded GA could achieve optimum solution in all runs (problems 1 to 4). These results are obtained in less than 7 seconds. In the medium size problems (problems 5 to 8), the best result is obtained in problem 8 with $F_{dev}$ of 0.51% and the worst solution is occurred in problem 6 with $F_{dev}$ of 1.59%. Except for problem 7, the RCGA could produce optimum solutions in more than 10 runs for all problems.

The RCGA also obtains optimum solutions in several runs in the large size problems (problems 9, 10, and 12), the best result is obtained in problem 10 with $F_{dev}$ of 0.34% and the worst solution is occurred in problem 9 with $F_{dev}$ of 3.79%. Overall, in larger problems, $F_{dev}$ values tend to increase as the search space becomes very wide. Increasing the population size, crossover rate and mutation rate will reduce $F_{dev}$ values but the computation time will rise.

It should be noted that lower throughputs achieved by the RCGA is compensated by better (lower) system unbalances on problems 5, 7 and 10. All $F_{dev}$ values are below 4% which may be regarded as good results considering these results are achieved in average of 35.25 seconds.

Note that these promising results are achieved by using only simple genetic operators. The novel proposed chromosome representation produces feasible solutions which minimize a computational time needed by GA to explore the feasible search space efficiently [29][30]. Other approaches may require sophisticated strategies to achieve good results which may require excessive computation time such as hybridizing tabu search with simulated annealing [17], hybridizing genetic algorithm with simulated annealing [25]-[26] and equipping particle swarm optimization with local search methods [7].

## IV. CONCLUSION

The part type selection and loading problem with flexibilities of operations have been modeled in this paper. These NP-hard problems were solved by using real coded GA. Combination of proper representation and simple genetic operators could produce promising results in reasonable amount of time. By using 12 test bed problems, the proposed RCGA improves the FMS performance by considering two objectives, maximizing system throughput and maintaining the balance of the system (minimizing system unbalance). The resulted objective values are compared to the optimum values produced by branch-and-bound method. The experiments show that the proposed RCGA could reach near optimum solutions in reasonable amount of time.

Further work will address more complex problem which considers alternative production plans which refer to possibility of producing part on alternative operation sequence. Resource allocation problem which refers to allocation of limited number of pallets and fixtures to the part types is also integrated to the existing problems. Therefore, a more powerful of GA is required.

Hybridizing the RCGA with other heuristics methods and developing new crossover and mutation methods will be considered.

## REFERENCES

[1] D.J. Parrish, *Flexible Manufacturing*, Oxford: Butterworth-Heinemann, 1993.
[2] S. Turgay, "Agent-Based Fms Control," *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 2, pp. 470-480, 2009.
[3] H.W. Kim, J.S. Kim, J.M. Yu, H.H. Doh, D.H. Lee, and S.H. Nam. "Loading Algorithms for Flexible Manufacturing Systems with Partially Grouped Unrelated Machines and Tooling Constraints," in *Computer Communication Control and Automation (3CA), 2010 International Symposium on*. 326-329, 2010.
[4] W.W. Luggen, *Flexible Manufacturing Cells and Systems*, Englewood

Cliffs, N.J: Prentice Hall, 1991.

[5] V. Malhotra, T. Raj, and A. Arora, "Excellent Techniques of Manufacturing Systems: Rms and Fms," *International Journal of Engineering Science and Technology*, vol. 2, no. 3, pp. 137-142, 2010.

[6] G. Abaza, I. Badr, P. Goehner, and S. Jeschke. "Extending an Agent-Based Fms Scheduling Approach with Parallel Genetic Algorithms," in *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*. 2689-2694, 2010.

[7] S. Biswas and S. Mahapatra, "Modified Particle Swarm Optimization for Solving Machine-Loading Problems in Flexible Manufacturing Systems," *The International Journal of Advanced Manufacturing Technology*, vol. 39, no. 9, pp. 931-942, 2008.

[8] F.T.S. Chan and H.K. Chan, "A Comprehensive Survey and Future Trend of Simulation Study on Fms Scheduling," *Journal of Intelligent Manufacturing*, vol. 15, no. 1, pp. 87-102, 2004.

[9] H.T.N.I.K. Nejad, N. Sugimura, K. Iwamura, and Y. Tanimizu, "Integrated Dynamic Process Planning and Scheduling in Flexible Manufacturing Systems Via Autonomous Agents," *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 2, no. 4, pp. 719-734, 2008.

[10] A.H.R. Zaied, "Quantitative Models for Planning and Scheduling of Flexible Manufacturing System," *Emirates Journal for Engineering Research*, vol. 13, no. 2, pp. 11-19, 2008.

[11] A. Madureira and J. Santos. "Proposal of Multi-Agent Based Model for Dynamic Scheduling in Manufacturing," in *The 6th WSEAS Int. Conf. on EVOLUTIONARY COMPUTING*. Lisbon, Portugal. 193-198, 2005.

[12] K.E. Stecke, "Design, Planning, Scheduling, and Control Problems of Flexible Manufacturing Systems," *Annals of Operations Research*, vol. 3, no. 1, pp. 1-12, 1985.

[13] S. Özpeynirci and M. Azizoglu, "Bounding Approaches for Operation Assignment and Capacity Allocation Problem in Flexible Manufacturing Systems," *Computers & Operations Research*, vol. 36, no. 9, pp. 2531-2540, 2009.

[14] S. Bilgin and M. Azizoglu, "Capacity and Tool Allocation Problem in Flexible Manufacturing Systems," *The Journal of the Operational Research Society*, vol. 57, no. 6, pp. 670-681, 2006.

[15] F.T.S. Chan and R. Swarnkar, "Ant Colony Optimization Approach to a Fuzzy Goal Programming Model for a Machine Tool Selection and Operation Allocation Problem in an Fms," *Robotics and Computer-Integrated Manufacturing*, vol. 22, no. 4, pp. 353-362, 2006.

[16] J.-H. Chen and S.-Y. Ho, "A Novel Approach to Production Planning of Flexible Manufacturing Systems Using an Efficient Multi-Objective Genetic Algorithm," *International Journal of Machine Tools and Manufacture*, vol. 45, no. 7–8, pp. 949-957, 2005.

[17] R. Swarnkar and M.K. Tiwari, "Modeling Machine Loading Problem of Fmss and Its Solution Methodology Using a Hybrid Tabu Search and Simulated Annealing-Based Heuristic Approach," *Robotics and Computer-Integrated Manufacturing*, vol. 20, no. 3, pp. 199-209, 2004.

[18] A.K. Choudhary, M.K. Tiwari, and J.A. Harding, "Part Selection and Operation-Machine Assignment in a Flexible Manufacturing System Environment: A Genetic Algorithm with Chromosome Differentiation-Based Methodology," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 220, no. 5, pp. 677-694, 2006.

[19] S.G. Ponnambalam and L.S. Kiat, "Solving Machine Loading Problem in Flexible Manufacturing Systems Using Particle Swarm Optimization," *World Academy of Science, Engineering and Technology*, vol. 39, 2008.

[20] A. Prakash, N. Khilwani, M.K. Tiwari, and Y. Cohen, "Modified Immune Algorithm for Job Selection and Operation Allocation Problem in Flexible Manufacturing Systems," *Adv. Eng. Softw.*, vol. 39, no. 3, pp. 219-232, 2008.

[21] M.T. Tabucanon, D.N. Batanov, and S. Basu, "Using Simulation to Evaluate the Batching Approach to Part Type Selection in Flexible Manufacturing Systems," *Integrated Manufacturing Systems*, vol. 9, no. 1, pp. 5-14, 1998.

[22] H.-W. Kim, J.-M. Yu, J.-S. Kim, H.-H. Doh, D.-H. Lee, and S.-H. Nam, "Loading Algorithms for Flexible Manufacturing Systems with Partially Grouped Unrelated Machines and Additional Tooling Constraints," *The International Journal of Advanced Manufacturing Technology*, vol. 58, no. 5, pp. 683-691, 2012.

[23] K. Seok Shin, J.O. Park, and Y. Keun Kim, "Multi-Objective Fms Process Planning with Various Flexibilities Using a Symbiotic Evolutionary Algorithm," *Computers and Operations Research*, vol. 38, no. 3, pp. 702-712, 2011.

[24] M.K. Tiwari, S. Kumar Jha, and R. Bardhan Anand, "Operation Allocation and Part Type Selection in E-Manufacturing: An Auction Based Heuristic Supported by Agent Technology," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 4, pp. 312-324, 2010.

[25] M.K. Tiwari, S. Kumar, S. Kumar, Prakash, and R. Shankar, "Solving Part-Type Selection and Operation Allocation Problems in an Fms: An Approach Using Constraints-Based Fast Simulated Annealing Algorithm," *IEEE Transaction on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 36, no. 6, pp. 1170-1184, 2006.

[26] M. Yogeswaran, S.G. Ponnambalam, and M.K. Tiwari, "An Efficient Hybrid Evolutionary Heuristic Using Genetic Algorithm and Simulated Annealing Algorithm to Solve Machine Loading Problem in Fms," *International Journal of Production Research*, vol. 47, no. 19, pp. 5421-5448, 2009.

[27] W. Shen, "Genetic Algorithms in Agent-Based Manufacturing Scheduling Systems," *Integr. Comput.-Aided Eng.*, vol. 9, no. 3, pp. 207-217, 2002.

[28] M. Lozano and F. Herrera, "Fuzzy Adaptive Genetic Algorithms: Design, Taxonomy," *Soft Computing*, vol. 7, pp. 545–562, 2003.

[29] R.M. Marian, L.H.S. Luong, and R. Akararungruangkul, "Optimisation of Distribution Networks Using Genetic Algorithms. Part 2: The Genetic Algorithm and Genetic Operators," *International Journal of Manufacturing Technology and Management*, vol. 15, no. 1, pp. 84-101, 2008.

[30] R.M. Marian, L.H.S. Luong, and K. Abhary, "A Genetic Algorithm for the Optimisation of Assembly Sequences," *Comput. Ind. Eng.*, vol. 50, no. 4, pp. 503-527, 2006.