

Solving machine loading problem of flexible manufacturing systems using a modified discrete firefly algorithm

Eleonora Bottani^a, Piera Centobelli^{b*}, Roberto Cerchione^c, Lucia Del Gaudio^d and Teresa Murino^e

^aDepartment of Industrial Engineering, University of Parma, viale G.P.Usberti 181/A, 43124 – Parma, Italy

^{b,c}Department of Industrial Engineering, University of Naples Federico II, P.le Tecchio 80, 80125 – Naples, Italy

^{d,e}Department of Chemical, Materials and Industrial Production Engineering, P.le Tecchio 80, 80125 – Naples, Italy

CHRONICLE

Article history:

Received October 2 2016

Received in Revised Format

October 28 2016

Accepted December 2 2016

Available online

December 2 2016

Keywords:

Discrete Firefly Algorithm

Flexible Manufacturing System

Machine Allocation Problem

Swarm-based Optimization

ABSTRACT

This paper proposes a modified discrete firefly algorithm (DFA) applied to the machine loading problem of the flexible manufacturing systems (FMSs) starting from the mathematical formulation adopted by Swarnkar & Tiwari (2004). The aim of the problem is to identify the optimal jobs sequence that simultaneously maximizes the throughput and minimizes the system unbalance according to given technological constraints (e.g. available tool slots and machining time). The results of the algorithm proposed have been compared with the existing and most recent swarm-based approaches available in the open literature using as benchmark the set of ten problems proposed by Mukhopadhyay et al. (1992). The algorithm shows results that are comparable and sometimes even better than most of the other approaches considering both the quality of the results provided and the computational times obtained.

© 2017 Growing Science Ltd. All rights reserved

1. Introduction

Manufacturing systems are changed through the years from high variety and low volume production, typical of job-shop production systems, to dedicated manufacturing lines characterized by low variety and high volume supporting the economies of scale. The FMS paradigm was developed in the 1980s, as a primary effort to include at the shop-floor level the ability to tackle with mass customization and to face the problem of mid-volume and mid-variety of the production requests (ElMaraghy, 2005).

Operatively in FMS pre- and post- release decisions have to be faced. Pre-release decisions consist of the FMS operational planning decision to be carried out before the process starts, e.g. the pre-allocation of jobs to machines, as well as tools, whereas post-release decisions refer to scheduling problems (Kumar, Murthy, & Chandrashekara, 2012). Pre-release decision problems may be very complex and according to Stecke (1983) they may include the following five objectives: machine grouping, selection of the part

* Corresponding author Tel. (+39) 0815795988. Fax: +39 0817682154
E-mail: piera.centobelli@unina.it (P. Centobelli)

type, calculation of the production rate, allocation of resource, and machine loading. Among them, the last two problems, i.e. allocation of resource and machine loading, are considered crucially important (Kumar et al., 2006). More specifically, Biswas and Mahapatra (2008) state that the overall performance of the FMS is mainly influenced by machine loading (ML) problem.

Machine loading problem covers many objectives. However, the goal is always to enhance the overall system performance (e.g. minimize the idle time of the machines, maximize the machine utilization) by defining jobs allocation of each part type to be produced to a given number of machines also satisfying technological constraints. Therefore, due to the different and simultaneous factors to be set, the problem of machine loading belongs to the NP-hard problems (Srinivas et al., 2004).

The ML problem was first formulated with the aim to maximize the throughput of the system over a given set of jobs using a branch and backtrack strategy (Stecke & Solberg, 1981). Since then, several approaches based on mathematical models (Mgwatu, 2011), heuristics (Tiwari et al., 2007), meta-heuristics (Sandhyarani & Mahapatra, 2009), simulation approaches (Lee & Jung, 1989) as well as a combination of them (Abazari et al., 2012), have been formulated to solve this optimization problem and reduce computational burden.

In recent years among metaheuristics, particularly interesting are the results obtained using the swarm-based approaches (Pandey, 2011; Yusof et al., 2014; Santuka et al., 2015). In summary it emerges that the ML problem in FMSs is a topic deeply analysed nevertheless currently under investigation.

With these premises, we propose a modified discrete firefly algorithm to be applied in the context of ML problem of FMSs in order to simultaneously reduce the system unbalance, that is the union of the under- and over-utilized times with regard to the set of machines of the system, and to maximize the throughput, which is the maximization of the production rate and the system efficiency, according to given technological constraints (i.e. tool slots available and total machining time available).

The paper is structured as follows. After this introduction, the mathematical formulation of ML in FMS is provided in Section 2. Section 3 illustrates the DFA adapted to the ML problem. In Section 4 the computational results are reported. At the end, conclusions and the future directions of research are discussed in Section 5.

2. Machine loading problem: mathematical formulation

In this article we adopt the mathematical formulation of ML problem developed by Swarnkar and Tiwari (2004) and the systematic review approach defined by Cerchione & Esposito (2016) to identify the main contributions on the topic.

The machine loading problem consists in a set of J jobs to be assigned to a set of M machines characterized by a positive processing time and fixed tool slots. Each job is considered as a set of operations with a predefinite processing time and fixed tool slot, and it can be realized on a specific set of machines. The aim is to identify an individual jobs sequence that simultaneously maximizes the throughput (TP) and minimizes the system unbalance (SU) according to given technological constraints.

The first objective to be minimized (F_1) is the system unbalance, formulated as follows:

$$\min F_1 \sum_{j=1}^M (UT_j - OT_j) = \max F_1 \left[- \sum_{j=1}^M (UT_j - OT_j) \right] = \frac{MXH - \sum_{j=1}^M (UT_j - OT_j)}{MXH} \quad (1)$$

where UT and OT are respectively the under- and over- utilized time.

The second objective to be maximized (F_2) is throughput expressed as:

$$\max F_2 = \frac{\sum_{i=1}^N (BS_i \times x_i)}{\sum_{i=1}^N BS_i} \quad (2)$$

Where BS_i is the batch size and x_i is a binary variable equal to 1 if the i -th part is selected and 0 otherwise.

Thus, the weighted sum criteria has been adopted (Moncayo-Martinez & Zhang, 2011) to convert the multi-objective problem into a single-objective one assigning a weight (w) to each target function. Therefore the overall objective function is formulated as:

$$\max F = \frac{w_1 \times F_1 + w_2 \times F_2}{w_1 + w_2} = \left[w_1 \frac{M \times H - \sum_{j=1}^M (UT_j - OT_j)}{M \times H} + w_2 \frac{\sum_{i=1}^N (BS_i \times x_i)}{\sum_{i=1}^N BS_i} \right] / (w_1 + w_2) \quad (3)$$

The problem is subjected to the following assumptions and constraints (Yogeswaran et al., 2009):

1. System unbalance. It is equal to the union of the UT and OT on the set of available machines after the complete allocation of jobs. Therefore, this value should be higher or ideally equal to 0 in case of complete utilization of the system.
2. Unique job routing. Each operation has to start and finish on the same machine.
3. Tool slots. The tools available on the selected machine should be higher - at least equal - than the number of tools necessary to process the operation of the jobs on that machine.
4. Non-splitting of job. The operations on a job should be completed before processing another one.
5. Sharing of tool slots is not allowed, i.e. the number of tool slots after any job assignment should be positive or zero.
6. Parts and tools necessary to process the jobs are readily available in proximity of the machine stations. Therefore the setup times as well as the material handling operations are negligible.

In addition each operation may be essential or optional. This latter may be executed on a certain group of machines with equal or unequal processing time and tool slots whereas the first type may be executed exclusively on a single machine which requires a given number of tool slots. Therefore, the flexibility of the system, as well as its optimization, consists in the allocation of operations to the available machines in order to enhance the overall system performance.

3. Modelling framework

3.1 Firefly algorithm

Firefly Algorithm was firstly formulated at Cambridge University by Yang (2009) and it is categorised into the class of the nature-inspired metaheuristics. It is inspired by the behaviour of fireflies, as well as their flashing feature, and can be applied to different engineering optimization problems. The flashing lights are the result of a biochemical process which allows fireflies to communicate with each other. In fact, in order to live in harmony, individuals within a group need to interact and adapt their behaviour to the whole aim of the groups. Group members cannot behave as if they are solitary. The collective decisions of fireflies represent the main biological basis for developing the firefly algorithm.

The main fireflies characteristics are formalized as follows:

- All fireflies are unisex hence every firefly will be attracted from other fireflies regardless of the sex.
- Their attractiveness is proportional to the brightness, and they both decrease as their distance increases.
- The light intensity of a firefly is affected or determined by the landscape of the fitness function.

The FA is mainly founded on four basic concepts: distance, intensity, attractiveness, and movement.

Distance (r): The distance between two generic fireflies (e.g. i and j) is expressed by the Cartesian distance as follows:

$$r_{ij} = \|p_i - p_j\| = \sqrt{\sum_{k=1}^n (p_{ik} - p_{jk})^2} \quad (4)$$

Intensity I(r): In the traditional FA the light intensity perceived of each firefly depends on the distance r and it is proportional to the value of the objective function. In line with the formulation provided by Yang (2009), it can be expressed as:

$$I = I_0 e^{-\gamma r^2} \quad (5)$$

where:

I_0 is the original light intensity

γ is the coefficient of light absorption.

Attractiveness (β): This represents a relative measure of the light perceived by beholders and other fireflies. Therefore the formulation of the attractiveness is expressed as:

$$\beta = \beta_0 e^{-\gamma r^2} \quad (6)$$

where β_0 represents the attractiveness when the distance r is equal to zero.

Movement: Finally, the movement of the generic firefly i , that is attracted by a brighter one, is expressed as:

$$p_i(t+1) = p_i(t) + \beta_0 e^{-\gamma r_{ij}^2} (p_i(t) - p_j(t)) + \alpha \epsilon_i \quad (7)$$

In this equation $p_i(t)$ represents the position of the firefly at the time t ; $\beta_0 e^{-\gamma r_{ij}^2} (p_i(t) - p_j(t))$ represents the attraction between fireflies; $\alpha \epsilon_i$ represents the randomness of the process where the vector ϵ_i includes random numbers extracted from a normal distribution and α is a random parameter.

Summarising, the firefly algorithm is governed by three different parameters: the first (α) concerns the randomness; the second (β) regards the attractiveness, and the third (γ) is the absorption coefficient. This latter influences the attractiveness and affects the convergence process of the firefly algorithm as well as its evolution.

3.2. Modified Discrete Firefly algorithm

Several discrete formulations of the FA have been applied to the discrete optimization problems in the context of operations management (Sayadi et al., 2010; Lu & Wang, 2016; Osaba et al., 2016; Zhou et al., 2015).

In this paragraph a firefly algorithm has been adopted for the ML problem of the FMS. In the FMS considered several jobs, characterized by one or more operations, must be executed with different processing requirements (i.e. processing time and tool slots necessary to execute such operations) on a given number of multifunctional machines. More specifically, in each problem jobs and operations have to be loaded on four machines. Furthermore it is assumed that each machine has 5 tool slots and 480 minutes of available time.

In our approach we defined two vectors for each firefly and they represent a candidate solution to the problem respecting the problem constraints discussed in section 2: the machine assignment vector (M_string) containing the machines assigned for each operations and, the operation scheduling vector

(O_string) reporting the operations sequencing on the machines. The methodology developed is based on the following steps:

Step 1. The population is randomly initialized, i.e. each firefly of the initial population has been generated using the casual permutation of both the part types to be allocated and the available machine on which these part types can be allocated (M_string and O_string) (Karthikeyan, 2014). With reference to the first problem formulated by Mukopadhyay, Midha, & Krishna (1992), an example of firefly representation is reported in Table 1.

Table 1

M_string and O_string representation

<i>Machine assignment vector</i>																							
<i>Position</i>																							
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19					
<i>Operations (O_{ik})</i>																							
O ₁₁	O ₂₁	O ₂₂	O ₂₃	O ₃₁	O ₃₂	O ₄₁	O ₄₂	O ₅₁	O ₅₂	O ₆₁	O ₆₂	O ₆₃	O ₇₁	O ₇₂	O ₇₃	O ₈₁	O ₈₂	O ₈₃					
<i>M_string</i>																							
3	4	4	2	1	3	3	4	2	2	4	4	2	2	2	4	2	2	1					
<i>Operation scheduling vector</i>																							
<i>Sequence</i>																							
6			4			1			3			2			8			7			5		
<i>Operation (O_{ik})</i>																							
O ₆₁	O ₆₂	O ₆₃	O ₄₁	O ₄₂	O ₁₁	O ₃₁	O ₃₂	O ₂₁	O ₂₂	O ₂₃	O ₈₁	O ₈₂	O ₈₃	O ₇₁	O ₇₂	O ₇₃	O ₅₁	O ₅₂					
<i>O_string</i>																							
11	12	13	7	8	1	5	6	2	3	4	17	18	19	14	15	16	9	10					

After that, the objective function of each firefly that belongs to the population is calculated in terms of SU and TP according to (3).

Step 2. After evaluating all the objective function value of each firefly of the population, the best one is selected in order to update the solution.

The updating is given by the firefly movement towards the brighter one, i.e. the best firefly of the population (Pbest). In order to calculate this movement, we must first measure the distance between the best firefly and the any other element of the population. Two methods has been adopted to evaluate the distance between two fireflies, i.e. the hamming distance and the position distance. The hamming distance, used in M_string, is expressed as the set of non-corresponding elements in two vectors. The position distance, used in O_string, is the number of minimal switches necessary to convert one string into another one. These distances allow to update the job sequences of the initial population, according to the attractiveness equation, and consequently the objective function values.

Considering as an example the best firefly and a generic firefly reported in table 2, the position distance for O_string is computed as follows:

Job 1. In both fireflies it is in the third position therefore in this case the position distance is 0.

Job 2. In the best firefly it occupies the fourth position while in the generic firefly the second one. The position distance for job 2 is 2.

Repeating the same process for the eight jobs, the position distance r is equal to 24.

Table 2

Position distance illustration

	<i>Jobs</i>							
Best firefly	4	3	1	2	8	5	6	7
Generic firefly	8	2	1	7	6	3	4	5

The attractiveness is expressed using the following equation:

$$\beta = \frac{\beta_0}{1 + \gamma r^2} \quad (8)$$

Once computed the β values, a randomized sequences uniformly distributed in $[0,1]$ is generated. If the random value results lower or equal to β , the corresponding insertion in the machine assignment vector and pair-wise exchange in operation scheduling vector is performed on the elements of the current firefly.

Thereafter a random integer number is generated between 1 and the dimension of M_string . The corresponding position of the machine in M_string is replaced with the one with the shortest processing time. As regards O_string , two integer randomly generated select the two operations to be switched.

Step 3. For each firefly of the population an updated value of the objective function is calculated. In order to explore the solution space each firefly replaces the previous one not only if it is better than the previous solution. However the best solution of the updating replaces the previous global best firefly only if it is better than the previous one.

Step 4. Repeat the previous steps until the maximum number of generation is reached, i.e. 50.

4. Computational experiments

The algorithm was coded in MATLAB R2015b and was run on a personal computer with an Intel® Core™ i5 processor and 4GB RAM in order to evaluate the performance of the proposed modified DFA algorithm.

In this section our computational results are presented to illustrate the benefits of the DFA developed. Several runs and trials were conducted to define the final set of the parameters to be used:

Population size: 20
 Attractiveness of firefly β_0 : 1.0
 Light absorption coefficient γ : 0.1
 Randomization parameter α : 1.0
 Maximum number of generations Gen_max : 50

In the following section the obtained results are reported.

The algorithm is tested on the benchmark problems available in Mukopadhyay, Midha, & Krishna (1992) and the comparative results between the proposed approach and the existing swarm-based ones reported in the literature are summarized in table 3. The comparative study was performed considering the heuristics with the same hypothesis of the proposed approach and developed by Yusof et al. (2014), Biswas and Mahapatra (2009), Biswas and Mahapatra (2008), Prakash et al. (2008), Prakash et al. (2008), Ponnambalam and Kiat (2008), Prakash et al. (2007), Tiwari et al. (1997), and Mukopadhyay et al. (1992).

The outcomes reported in Table 3 show that the algorithm outperforms the heuristics developed by Tiwari et al. (2007) and Mukopadhyay et al. (1992), and, except for the problem number 2 in which the SU and TP values obtained are respectively 9 and 46, reach the same results of the others swarm approaches.

In conclusion, the algorithm shows results that are comparable, and sometimes even better than, most of the other approaches in terms of quality of solutions obtained and computational times. Indeed for the most complex problem, i.e. the first one, the algorithm reaches the best solution in 8.7 seconds.

Table 3

Comparison of the proposed modified DFA with other-heuristic solutions available in the open literature for the problem set

Pr. N°	Proposed approach		Yusof et al., 2014		Biswas & Mahapatra, 2009		Biswas & Mahapatra, 2008		Prakash et al., 2008		Prakash et al., 2008		Ponnambalam & Kiat, 2008		Prakash et al., 2007		Tiwari et al., 1997		Mukopadhyay et al., 1992			
	SU	TP	SU	TP	SU	TP	SU	TP	SU	TP	SU	TP	SU	TP	SU	TP	SU	TP	SU	TP		
1	14	48	0	52	3	49	3	49	14	48	14	48	14	48	14	48	14	48	76	42	122	42
2	9	46	15	64	69	50	69	50	22	63	124	63	22	63	22	63	124	63	234	63	202	63
3	28	73	28	73	32	48	32	48	25	73	72	69	28	73	25	73	72	69	152	69	286	79
4	819	51	819	51	819	51	819	51	819	51	819	51	819	51	819	51	819	51	819	51	819	51
5	264	61	69	61	9	64	9	64	264	61	264	61	264	61	264	61	264	61	264	61	364	76
6	7	64	7	64	35	62	35	62	37	64	28	61	37	64	37	64	28	61	314	63	365	62
7	231	63	21	63	39	66	39	66	147	66	231	63	231	63	147	66	231	63	996	48	147	66
8	63	48	56	54	52	50	52	50	63	48	13	44	63	48	63	48	13	44	158	43	459	36
9	309	88	56	88	309	88	309	88	309	88	309	88	309	88	309	88	309	88	309	88	309	88
10	122	56	205	67	5	58	5	58	122	56	122	56	122	56	122	56	122	56	76	42	320	56

5. Conclusions

This paper proposes to apply a discrete firefly algorithm to the machine loading problem of the flexible manufacturing systems. Despite the machine loading problem in the context of flexible manufacturing is a topic deeply investigated in the literature and already tackled by researchers with several approaches, nevertheless the potentiality of the swarm-based approach makes it interesting and challenging.

Among them, the recently developed firefly algorithm has been selected due to its interesting adaptability features (i.e. for the calculation of distance and attractiveness between fireflies) and therefore suitable for our problem.

The present work has successfully developed an efficient heuristic based on firefly algorithm with the aim to formulate an objective function which takes into account both the throughput and the system unbalance. Iterative experiments had been conducted in order to achieve optimal or semi-optimal solutions and the findings have been analyzed and compared with the previous contributions on the topic.

The analysis of the machine loading problem concerns the challenge of locating the available resources (machines) to load the part types; the constraints of the problem are concurrently considered such as the number of operations for each part type, the allowable machines that may be allocated for the operation, and the machine time available.

The results highlight that the proposed approach offers results that are comparable to the best results of the other swarm algorithms, and outperforms the other heuristics considered for evaluation.

This paper addresses an interesting topics and it will be still the basis of new studies and research works.

Indeed, the proposed approach has been applied on datasets acquired in literature, so an extension of this work may be to apply it in a real life industrial problem; in addition, as future development the DFA procedure proposed may be tested on various similar optimization problems, particularly to the problems that regard the sequencing or the allocation of resources. In addition, this research can be also exploited in multi-objective loading and scheduling problem with the addition of more flexible attributes. Finally further experiments may be conducted using a more complex procedure to generate the initial population as well as implement a strategy to avoid trapping into local optima.

References

- Abazari, A., Solimanpur, M., & Sattari, H. (2012). Optimum loading of machines in a flexible manufacturing system using a mixed-integer linear mathematical programming model and genetic algorithm. *Computers and Industrial Engineering*, 62(2), 469-478.
- Biswas, S., & Mahapatra, S. S. (2008). Modified particle swarm optimization for solving machine-loading problems in flexible manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, 39(9-10), 931-942.
- Biswas, S., & Mahapatra, S. S. (2009). An improved metaheuristic approach for solving the machine loading problem in flexible manufacturing systems. *International Journal of Services and Operations Management*, 5(1), 76-93.
- Cerchione, R., & Esposito, E. (2016). A Systematic Review of Supply Chain Knowledge Management Research: State of the Art and Research Opportunities. *International Journal of Production Economics*, 182, 276-292.
- ElMaraghy, H. (2005). Flexible and reconfigurable manufacturing systems paradigms. *International Journal of Flexible Manufacturing Systems*, 17(4), 261–276.
- Karthikeyan, S., Asokan, P., & Nickolas, S. (2014). A hybrid discrete firefly algorithm for multi-objective flexible job shop scheduling problem with limited resource constraints. *The International Journal of Advanced Manufacturing Technology*, 72(9-12), 1567-1579.
- Kumar, A., Prakash, Tiwari, M., Shankar, R., & Baveja, A. (2006). Solving machine-loading problem of a flexible manufacturing system with constraint-based genetic algorithm. *European Journal of Operational Research*, 175(2), 1043-1069.
- Kumar, V. M., Murthy, A. N. N., & Chandrashekar, K. (2012). A hybrid algorithm optimization approach for machine loading problem in flexible manufacturing system. *Journal of Industrial Engineering International*, 8(1), 1-10.
- Lee, S. M., & Jung, H. J. (1989). A multi-objective production planning model in a flexible manufacturing environment. *The International Journal of Production Research*, 27(11), 1981-1992.
- Lu, S., & Wang, X. (2016). Modeling the fuzzy cold storage problem and its solution by a discrete firefly algorithm. *Journal of Intelligent and Fuzzy Systems*, 31(4), 2431-2440.
- Mgwatu, M. (2011). Integration of part selection, machine loading and machining optimisation decisions for balanced workload in flexible manufacturing system. *International Journal of Industrial Engineering Computations*, 2(4), 913-930.
- Moncayo-Martinez, L., & Zhang, D. (2011). Multi objective ant colony optimisation: a meta-heuristic approach to Supply Chain design. *International Journal of Production Economics*, 131(1), 407-420.
- Mukopadhyay, S., Midha, S., & Krishna, V. (1992). A heuristic procedure for loading problem in flexible manufacturing system. *International Journal of Production Research*, 30(9), 2213-2228.
- Osaba, E., Yang, X.-S., Diaz, F., Onieva, E., Masegosa, A., & Perallos, A. (2016). A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy. *Soft Computing*, 1-14.

- Pandey, M. (2011). Operation allocation in flexible manufacturing system using immune algorithm. In M. T. Harding, *Evolutionary Computing in Advanced Manufacturing* (p. 95-121). Hoboken, NJ, USA.
- Ponnambalam, S. G., & Kiat, L. S. (2008). Solving machine loading problem in flexible manufacturing systems using particle swarm optimization. *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering*, 2(3), 242-247.
- Prakash, A., Khilwani, N., Tiwari, M., & Cohen, Y. (2008). Modified immune algorithm for job selection and operation allocation problem in flexible manufacturing system. *Advances in Engineering Software*, 39(3), 219-232.
- Prakash, A., Shankar, R., Shukla, N., & Tiwari, M. (2007). Solving machine loading problem of FMS: An artificial intelligence (AI) based random search optimization approach. In *Handbook of Computational Intelligence in Manufacturing and Production Management* (p. 19-43).
- Prakash, A., Tiwari, M., & Shankar, R. (2008). Optimal job sequence determination and operation machine allocation in flexible manufacturing systems: An approach using adaptive hierarchical ant colony algorithm. *Journal of Intelligent Manufacturing*, 19(2), 161-173.
- Sandhyarani, B., & Mahapatra, S. (2009). An improved metaheuristic approach for solving the machine loading problem in flexible manufacturing systems. *International Journal of Services and Operations Management*, 5(1), 76-93.
- Santuka, R., Mahapatra, S., Dhal, P., & Mishra, A. (2015). An improved particle swarm optimization approach for solving machine loading problem in flexible manufacturing system. *Journal of Advanced Manufacturing Systems*, 14(3), 167-187.
- Sayadi, M., Ramezani, R., & Ghaffari-Nasab, N. (2010). A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. *International Journal of Industrial Engineering Computations*, 1(1), 1-10.
- Srinivas, Tiwari, M., & Allada, V. (2004). Solving the machine loading problem in a flexible manufacturing system using a combinatorial auction-based approach. *International Journal of Production Research*, 42(9), 1879-1893.
- Stecke, K. (1983). Formulation and solution of nonlinear integer production planning problems for flexible manufacturing systems. *Management Science*, 29, 273-288.
- Stecke, K. E., & Solberg, J. J. (1981). Loading and control policies for a flexible manufacturing system. *The International Journal of Production Research*, 19(5), 481-490.
- Swarnkar, R., & Tiwari, M. K. (2004). 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*, 20(3), 199-209.
- Tiwari, M., Hazarika, B., Vidyarthi, N., Jaggi, P., & Mukhopadhyay, S. (1997). A heuristic solution approach to the machine loading problem of FMS and its petri net model. *International Journal of Production Research*, 35(8), 2269-2284.
- Tiwari, M., Saha, J., & Mukhopadhyay, S. (2007). Heuristic solution approaches for combined-job sequencing and machine loading problem in flexible manufacturing systems. *The International Journal of Advanced Manufacturing Technology*, 31(7-8), 716-730.
- Yang, X. (2009). Firefly algorithms for multimodal optimization. In Stochastic algorithms: In *Stochastic Algorithms: Foundations and Applications* (p. 169-178). Springer Berlin Heidelberg.
- Yogeswaran, M., Ponnambalam, S., & Tiwari, M. (2009). An efficient hybrid evolutionary heuristic using genetic algorithm and simulated annealing algorithm to solve machine loading problem in FMS. *International Journal of Production Research*, 47(19), 5421-5448.
- Yusof, U., Budiarto, R., & Deris, S. (2014). A hybrid of bio-inspired and musical-harmony approach for machine loading optimization in flexible manufacturing system. *International Journal of Innovative Computing, Information and Control*, 10(6), 2325-2344.
- Yusof, U., Khalid, M., & Khader, A. (2014). Artificial immune system for flexible manufacturing system machine loading problem. *ICIC Express Letters*, 8(3), 709-716.
- Zhou, L., Ding, L., Qiang, X., & Luo, Y. (2015). An improved discrete firefly algorithm for the traveling salesman problem. *Journal of Computational and Theoretical Nanoscience*, 12(7), 1184-1189.



© 2016 by the authors; licensee Growing Science, Canada. This is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).