ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 9, March 2017

APPLICATION OF FIREFLY ALGORITHM AND ITS PARAMETER SETTING FOR JOB SHOP SCHEDULING

S. K. Rajesh Kanna[1]

Professor, Rajalakshmi Institute of Technology, Chennai[1].

***Abstract*— Job shop scheduling problem (JSSP) is one of the most famous scheduling problems, most of which are categorised into Non-deterministic Polynomial (NP) hard problem. The objectives of this paper are to i) present the application of a recent developed metaheuristic called Firefly Algorithm (FA) for solving JSSP; ii) investigate the parameter setting of the proposed algorithm; and iii) compare the FA performance using various parameter settings. The computational experiment was designed and conducted using five benchmarking JSSP datasets from a classical OR-Library. The analysis of the experimental results on the FA performance comparison between with and without using optimised parameter settings was carried out. The FA with appropriate parameters setting that got from the experiment analysis produced the best-so-far schedule better than the FA without adopting parameter settings.**

***Keywords:* Scheduling, Job shop, Metaheuristics, Firefly Algorithm, Experimental design, Parameter setting**

I. INTRODUCTION

Scheduling problem is a decision making process involving the allocation of resources over time to perform a set of activities or tasks. Scheduling problems in their static and deterministic forms are simple to describe and formulate, but are difficult to solve as it involves complex combinatorial optimisation. For example, if there are m machines, each of which is required to perform n independent operations. The combination can be potentially exploded up to (n!)m operational sequences. Job shop scheduling is one of the most famous scheduling problems, most of which are categorised into NP hard problem. This means that due to the combinatorial explosion, even a computer can take unacceptably large amount of time to seek a satisfied solution on even moderately large scheduling problem. Another potential issue of complexity is the assembly relationship [1-2]. Job shop scheduling problem (JSSP) is comprised of a set of independent jobs or tasks (J), each of which consists of a sequence of operations (O). Each operation is performed on machine (M) without interruption during processing time. The main purpose of JSSP is usually to find the best machine schedule for servicing all jobs in order to optimise either single criterion or multiple scheduling objectives (measures of performance) such as the minimisation of the makespan (Cmax) or the penalty costs of tardiness and/or earliness. Various optimisation approaches have been widely applied to solve the JSSP. Conventional methods based on mathematical model and/or full numerical search (for example, Branch and Bound [3-4] and Lagrangian Relaxation [5-6]) can guarantee the optimum solution. They have been successfully used to solve JSSP. However, these methods may highly consume computational time and resources even for solving a moderate-large problem size and therefore impractical if the computational limitation is exist. Approximation optimisation methods or metaheuristics (e.g. Tabu Search [7] and Simulated Annealing [8]), that usually conduct stochastic steps in their search process, have therefore been recently received more attention for solving a large-size problem in the last few decades. However, it does not guarantee the optimum solution. Firefly Algorithm (FA) was recently introduced by Yang [9], who was inspired by firefly behaviours. FA has been widely applied to solve continuous mathematical functions [9, 10]. FA seems promising for dealing with combinatorial optimisation problem, but has been rarely reported. FA is a type of metaheuristic algorithm therefore quality of problem solutions depends on setting parameters in the algorithm.

There is however no report on international scientific databases related to the investigation of the FA parameters' setting and its application on the JSSP. The popular Job shop scheduling problems that were used to test metaheuristic algorithm regularly are the datasets from OR-Library [11]. The objectives of this paper are to: i) presents the application of a recent developed metaheuristic called Firefly Algorithm for solving JSSP; ii) explore the parameters of the proposed algorithm; and iii) investigate the performance of the FA with different parameter setting and compare with best known solution from literature review. A job shop scheduling tool was written in modular style using Tcl/Tk programming language. The computational experiment was designed and conducted using five benchmarking datasets of the JSSP instance from the well-recognised OR-Library published by Beasley [11]. The remaining sections in this paper are organised as follows. Section 2 reviews the literature relating to job shop scheduling problems. Section 3 describes the procedures of the Firefly Algorithm (FA) and its pseudo code for solving the JSSP. Section 4 presents the experimental design and analyses results. Finally, Section 5 draws the conclusions of the research and suggests possible further work.

Comparison of Performance with Tracking Collector" European Journal of Scientific Research, ISSN 1450-216X Vol.40 No.1, pp.144 -155, Euro Journals Publishing, Inc. 2010.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 9, March 2017

5. Wattana Ratismith (2012), "A Novel Non-Tracking Solar Collector for High Temperature Application.", proceedings of ecos 2012 - the 25th international conference on efficiency, cost, optimization, simulation and environmental impact of energy systems, Perugia, Italy.
6. Kristina Uzuneanu, Alexandrina Teodoru, Tanase Panait, "Optimum Tilt Angle for Solar Collectors with Low Concentration Ratio". Vol. 32 No.1, pp. 134-168.

II. JOB SHOP SHEDULING PROBLEMS (JSSP)

Shop floor scheduling problems can be classified into four main categories [12]: (i) single-machine scheduling (ii) flow-shop scheduling (iii) job-shop scheduling and (iv) open-shop scheduling. Single-machine scheduling is the simplest shop scheduling problem, in which there is only one available machine for servicing the arriving jobs. In flowshop scheduling, jobs are processed on multiple machines in an identical sequence. Job-shop scheduling is a general case of flow-shop scheduling, in which the sequencing of each job through the machines is not necessarily identical. The open-shop scheduling is similar to the job-shop scheduling except that a job may be processed on the machines in any sequence the job needs. Since the job-shop scheduling is commonly found in many real-world businesses and/or manufacturing industry, this problem was proposed in this paper. Although JSSP is common found but it is very difficult work to construct the best schedule within limited resources. The complexity of JSSP is increasing with the number of constraints defined and the size of search space operated. JSSP is known as one of the most difficult Non-deterministic Polynomial (NP) hard problems [13-14], in which the amount of computation required increases exponentially with problem size. The JSSP can be described as follow: a classical n-job m-machine (n×m) JSSP consists of a finite set [Jj] $1 \le j \le n$ of n independent jobs or tasks that must be processed in a finite set [Mk] $1 \le k \le m$ of m machines. The problem can be characterised as follows [15]: each job $j \in J$ must be processed by every machines $k \in M$; the processing of job Jj on machine Mk is called the operation Ojk; operation Ojk requires the exclusive use of machine Mk for an uninterrupted duration tjk, its processing time; each job consists of a sequence of xj operations; Ojk can be processed by only one machine k at a time (disjunctive constraint); each operation, which has started, runs to completion (nonpreemption condition); and each machine performs operations one after another (resource/capacity constraint). An example of two jobs to be performed three machines (2×3) job shop scheduling problem is illustrated in Table 1. In this problem, each job requires three operations to be processed on a pre-defined machine sequence. The first job (J1) need to be initially operated on the machine M1 for 5 time units and then sequentially processed on M2 and M3 for 4 and 9 time units, respectively. Likewise, the second job (J2) has to be initially performed on M3 for 5 time units and sequentially followed by M1 and M2 for 6 and 7 time units, respectively. The design task for solving JSSP is to search for the best schedule(s) for operating all pre-defined jobs in order to optimise either single or multiple scheduling objectives, which is used for identifying a goodness of schedule such as the minimisation of the makespan (Cmax).

Figure 1 shows another example on machine routings of a larger size of (3×6) job shop scheduling problem.

Table 1 An example of 2-jobs 3-machines scheduling problem with processing times.

Job (*Jj*)	**Operation (*Ojk*)**	**Time (*tjk*)**	**Machine (*Mk*)**		
			M_1	M_2	M_3
J1	*O11*	5	5	-	-
	O12	4	-	4	-
	O13	9	-	-	9
J2	*O23*	5	-	-	5
	O21	6	6	-	-
	O22	7	-	7	-

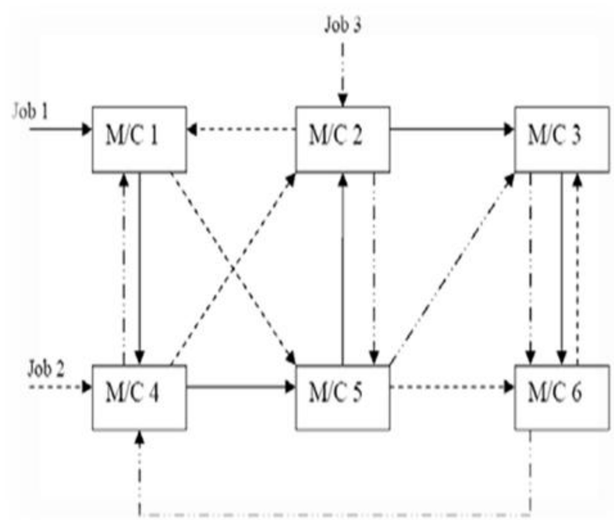

Fig. 1. Another example of the machine routings for 3-jobs 6-machines scheduling problem.

There has been a number of research works focused on the JSSP reported in the literature. The recent published research works related to JSSP by classifying those articles into four categories [16]: heuristic rules (for example, dispatching heuristics/priority rule [17-18]), mathematical programming techniques (e.g. branch and bound method, Lagrangian relaxation based approaches, queuing network model and etc. [3-6]), neighbourhood search methods (for instances, Tabu Search, Simulated Annealing [7-8]), and artificial intelligence techniques (such as, expert/knowledge based systems, artificial neural network, fuzzy logic, petri net based approaches [19-22]).

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 9, March 2017

III. FIREFLY ALGORITHM FOR SOLVING JOB SHOP SCHEDULING PROBLEMS

Firefly Algorithm (FA) is a nature inspired algorithms, which is based on the flashing light of fireflies. The flashing light helps fireflies for finding mates, attracting their potential prey and protecting themselves from their predators. The swarm of fireflies will move to brighter and more attractive locations by the flashing light intensity that associated with the objective function of problem considered in order to obtain efficient optimal solutions.

The development of firefly-inspired algorithm was based on three idealised rules [9]: i) artificial fireflies are unisex so that sex is not an issue for attraction; ii) attractiveness is proportional to their flashing brightness which decreases as the distance from the other firefly increases due to the fact that the air absorbs light. Since the most attractive firefly is the brightest one, to which it convinces neighbours moving toward. In case of no brighter one, it freely moves any direction; and iii) the brightness of the flashing light can be considered as objective function to be optimised. The main steps of the FA start from initialising a swarm of fireflies, each of which is determined the flashing light intensity. During the loop of pairwise comparison of light intensity, the firefly with lower light intensity will move toward the higher one. The moving distance depends on the attractiveness. After moving, the new firefly is evaluated and updated for the light intensity. During pairwise comparison loop, the best-so-far solution is iteratively updated. The pairwise comparison process is repeated until termination criteria are satisfied. Finally, the best-so-far solution is visualised. The pseudo code of FA applied to solve the JSSP is shown in Fig. 2. The main processes are described in the following subsections.

3.1 Population initialisation

Figs. 3-5 illustrate a typical firefly representation for the scheduling with nine operations. All operations required to produce jobs are encoded using alphanumeric strings. Each encoded operation is randomly selected and sequenced until all operations are drawn in order to create a firefly, which represents a candidate solution. This random selection is repeated to generate a swarm of fireflies with the required size. The length of slots in a firefly is equal to the total number of operations to be performed. The size of the firefly population determines the number of candidate solutions or the amount of search in the solution space.

Objective function $f(x)$, $x = (x1, ..., xd)T$
Generate initial population of fireflies xi (i = 1, 2, ..., n)
Light intensity Ii at xi is determined by $f(xi)$
Define light absorption coefficient γ
While t < Max Generation (G)
For $i = 1 : n$ all n fireflies
For $j = 1 : i$ all n fireflies
If ($Ij > Ii$), Move firefly i towards j in d-dimension;
Attractiveness varies with distance rij via $exp[-\gamma rij]$
Evaluate new solutions and update light intensity
End if
End for j
End for i
Rank the fireflies and find the current best
End while
Post process on the best-so-far results and visualisation

Fig. 2.The pseudo code of the FA procedure adopted from [9].

3.2 Firefly evaluation

The next stage is to measure the flashing light intensity of the firefly, which depends on the problem considered. In this work, the evaluation on the goodness of schedules is measured by the makespan, which can be calculated using equation (1), where Ck is completed time of job k.

Minimises $C_{max} = \max(c_1, c_2, \ldots, c_k)$ (1)

3.3 Distance

The distance between any two fireflies i and j at xi and xj, respectively, can be defined as a Cartesian distance (rij) using equation (2), where xi,k is the kth component of the spatial coordinate xi of the ith firefly and d is the number of dimensions [9, 23].

$$r_{ij} = \|x_i - x_j\| \sqrt{\sum_{k=1}^{d} \left(x_{i,k} - x_{j,k}\right)^2}$$

J1:1	J1:2	J1:3	J2:1	J2:2	J2:3	J3:1	J3:2	J3:3
01	02	03	04	05	06	07	08	09
0.45	0.52	0.99	0.27	0.09	0.01	0.33	0.85	0.78

Fig.3. Random interval 0-1 for each job operation.

J2:3	J2:2	J2:1	J3:1	J1:1	J1:2	J3:3	J3:2	J1:3
06	05	04	07	01	02	09	08	03
0.01	0.09	0.27	0.33	0.45	0.52	0.78	0.85	0.99

Fig.4. Sort random number and follow with job operation.

J2:3	J2:2	J2:1	J3:1	J1:1	J1:2	J3:3	J3:2	J1:3
06	05	04	07	01	02	09	08	03
0.01	0.09	0.27	0.33	0.45	0.52	0.78	0.85	0.99

J2:1	J2:2	J2:3	J3:1	J1:1	J1:2	J3:2	J3:3	J1:3
04	05	06	07	01	02	08	09	03
0.27	0.09	0.01	0.33	0.45	0.52	0.85	0.78	0.99

Fig.5. Check and repair the sequence of each job.

3.4 Attractiveness

The calculation of attractiveness function of a firefly are shown in equations (3), where r is the distance between any

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 9, March 2017

two fireflies, β0 is the initial attractiveness at r = 0, and γ is an absorption coefficient which controls the decrease of the light intensity [9, 23].

$$\beta_0 - \beta_0 * exp(-\gamma r^m), with\ m \geq 1$$

3.5 Movement

The movement of a firefly i which is attracted by a more attractive (i.e., brighter) firefly j is given by the following equation (4), where xi is the current position or solution of a firefly, the $\beta_0 * exp(-\gamma r_{ij}^2) * (x_j - x_i)$ is attractiveness of a firefly to seen by adjacent fireflies. The α (rand – 1/2) is a firefly's random movement. The coefficient α is a randomisation parameter determined by the problem of interest with α ϵ [0-1], while rand is a random number obtained from the uniform distribution in the space [0,1] [9][23].

$X_i = x_i + \beta_0 \times exp(-\gamma r_{ij}^2) * (x_j - x_i) + \alpha(rand - 1/2)$ (4)

Because the FA was recently developed, there have been a few research works applied the FA for solving optimisation problems, most of which has been formulated into mathematical equations. In the previous works, the settings of FA parameters, including the amount of fireflies (n), the number of generations (G), the light absorption coefficient (γ), the randomisation parameter (α) and the attractiveness value (β0), have been defined in an ad hoc fashion. Table 2 summarises the FA parameter settings used in previous researches for solving various optimisation problems.

Unfortunately, most of the work has not reported on the investigation of the appropriate setting of FA parameters via a proper statistical design and analysis. The computational experiments described in the next section were therefore proposed to identify the appropriate setting of FA parameters for solving scheduling problem.

Table 2 Examples of FA parameters' setting used in previous researches.

Authors	Problems	FA parameters			
		nG	*γ*	*α*	*β₀*
Apostolopoulos and Vlachos [24]	Economic emissions load dispatch	12*50	1.0	0.2	1.0
Lukasik and Zak [10]	Continuous equation	40*250	1.0	0.01	1.0
Horng and Jiang [25]	Image vector quantisation	50*200	1.0	0.01	1.0

IV. EXPERIMENTAL DESIGN AND ANALYSIS

Firefly Algorithm (FA) is a nature inspired algorithms, which is based on the flashing light of fireflies. The flashing light helps fireflies for finding mates, attracting their potential prey and protecting themselves from their predators. The swarm of fireflies will move to brighter and more attractive locations by the flashing light intensity that associated with the objective function of problem considered in order to obtain efficient optimal solutions.

The development of firefly-inspired algorithm was based on three idealised rules [9]: i) artificial fireflies are unisex so that sex is not an issue for attraction; ii) attractiveness is proportional to their flashing brightness which decreases as the distance from the other firefly increases due to the fact that the air absorbs light. Since the most attractive firefly is the brightest one, to which it convinces neighbours moving toward. In case of no brighter one, it freely moves any direction; and iii) the brightness of the flashing light can be considered as objective function to be optimised. The main steps of the FA start from initialising a swarm of fireflies, each of which is determined the flashing light intensity. During the loop of pairwise comparison of light intensity, the firefly with lower light intensity will move toward the higher one. The moving distance depends on the attractiveness. After moving, the new firefly is evaluated and updated for the light intensity. During pairwise comparison loop, the best-so-far solution is iteratively updated. The pairwise comparison process is repeated until termination criteria are satisfied. Finally, the best-so-far solution is visualised. The pseudo code of FA applied to solve the JSSP is shown in Fig. 2. The main processes are described in the following subsections.

3.1 Population initialisation

Figs. 3-5 illustrate a typical firefly representation for the scheduling with nine operations. All operations required to produce jobs are encoded using alphanumeric strings. Each encoded operation is randomly selected and sequenced until all operations are drawn in order to create a firefly, which represents a candidate solution. This random selection is repeated to generate a swarm of fireflies with the required size. The length of slots in a firefly is equal to the total number of operations to be performed. The size of the firefly population determines the number of candidate solutions or the amount of search in the solution space.

Objective function $f(x)$, $x = (x1, ..., xd)T$
Generate initial population of fireflies *xi* (i = 1, 2, ..., *n*)
Light intensity *Ii* at *xi* is determined by $f(xi)$
Define light absorption coefficient *γ*
While *t* < Max Generation (G)
For *i* = 1 : *n* all *n* fireflies
For *j* = 1 : *i* all *n* fireflies
If (*Ij* > *Ii*), Move firefly *i* towards *j* in *d*-dimension;

Attractiveness varies with distance *rij* via *exp*[−*γrij*]
Evaluate new solutions and update light intensity
End if
End for *j*
End for *i*
Rank the fireflies and find the current best
End while
Post process on the best-so-far results and visualisation

Fig. 2.The pseudo code of the FA procedure adopted from [9].

3.2 Firefly evaluation

The next stage is to measure the flashing light intensity of the firefly, which depends on the problem considered. In this work, the evaluation on the goodness of schedules is measured by the makespan, which can be calculated using equation (1), where Ck is completed time of job k.

Minimises $C_{max} = \max(c_1, c_2, \ldots, c_k)$ (1)

3.3 Distance

The distance between any two fireflies i and j at xi and xj, respectively, can be defined as a Cartesian distance (rij) using equation (2), where xi,k is the kth component of the spatial coordinate xi of the ith firefly and d is the number of dimensions [9, 23].

$r_{ij}=\|x_i-x_j\|\sqrt{\sum_{k=1}^{d}(x_{i,k}-x_{j,k})^2}$

J1:1	J1:2	J1:3	J2:1	J2:2	J2:3	J3:1	J3:2	J3:3
01	02	03	04	05	06	07	08	09
0.45	0.52	0.99	0.27	0.09	0.01	0.33	0.85	0.78

Fig.3. Random interval 0-1 for each job operation.

J2:3	J2:2	J2:1	J3:1	J1:1	J1:2	J3:3	J3:2	J1:3
06	05	04	07	01	02	09	08	03
0.01	0.09	0.27	0.33	0.45	0.52	0.78	0.85	0.99

Fig.4. Sort random number and follow with job operation.

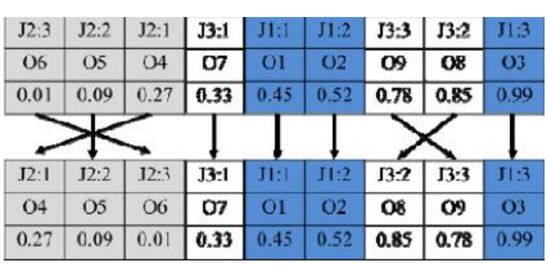

Fig.5. Check and repair the sequence of each job.

3.4 Attractiveness

The calculation of attractiveness function of a firefly are shown in equations (3), where r is the distance between any two fireflies, β0 is the initial attractiveness at r = 0, and γ is an absorption coefficient which controls the decrease of the light intensity [9, 23].

$$\beta_0 - \beta_0 * exp(-\gamma r^m), with\ m \geq 1$$

3.5 Movement

The movement of a firefly i which is attracted by a more attractive (i.e., brighter) firefly j is given by the following equation (4), where xi is the current position or solution of a firefly, the $\beta_0 * exp(-\gamma r_{ij}^2) * (x_j - x_i)$ is attractiveness of a firefly to seen by adjacent fireflies. The α (rand – 1/2) is a firefly's random movement. The coefficient α is a randomisation parameter determined by the problem of interest with α ϵ [0-1], while rand is a random number obtained from the uniform distribution in the space [0,1] [9][23].

$X_i = x_i + \beta_0 \times exp(-\gamma r_{ij}^2) * (x_j - x_i) + \alpha(rand - 1/2)$ (4)

Because the FA was recently developed, there have been a few research works applied the FA for solving optimisation problems, most of which has been formulated into mathematical equations. In the previous works, the settings of FA parameters, including the amount of fireflies (n), the number of generations (G), the light absorption coefficient (γ), the randomisation parameter (α) and the attractiveness value (β0), have been defined in an ad hoc fashion. Table 2 summarises the FA parameter settings used in previous researches for solving various optimisation problems.

Unfortunately, most of the work has not reported on the investigation of the appropriate setting of FA parameters via a proper statistical design and analysis. The computational experiments described in the next section were therefore proposed to identify the appropriate setting of FA parameters for solving scheduling problem.

Table 2 Examples of FA parameters' setting used in previous researches.

Authors	Problems	FA parameters			
		nG	*γ*	*α*	*β₀*
Apostolopoulos and Vlachos [24]	Economic emissions load dispatch	12*50	1.0	0.2	1.0
Lukasik and Zak [10]	Continuous equation	40*250	1.0	0.01	1.0
Horng and Jiang [25]	Image vector quantisation	50*200	1.0	0.01	1.0

V. APPLICATIONS

i. Digital Image Compression and Image Processing

Very recently, an FF-LBG algorithm for vector quantization of digital image compression was based on the firefly algorithm, which proves to be faster than other algorithms such as PSO-LBG and HBMO-LBG (particle swarm optimization and honey-bee mating optimization; variations on the Linde-Buzo-Gray algorithm for minimum cross entropy thresholding, firefly-based algorithm uses the least computation time. Also, for gel electrophoresis images, FA-based method is very efficient.

ii. Eigenvalue optimization

Eigenvalue optimization of isospectral systems has solved by FA and multiple optimum points have been found efficiently.

iii. Feature selection and fault detection

Feature selection can be also carried out successfully using firefly algorithm. Real-time fault identification in large systems becomes viable, based on the recent work on fault identification with binary adaptive firefly optimization.

iv. Antenna Design

Firefly algorithms outperforms ABC for optimal design of linear array of isotropic sources and digital controllable array antenna.

v. Structural Design

For mixed-variable problems, many optimization algorithms may struggle. However, firefly algorithm can efficiently solve optimization problems with mixed variables.

vi. Scheduling and TSP

Firefly-based algorithms for scheduling task graphs and job shop scheduling requires less computing than all other metaheuristics. A binary firefly algorithm has been developed to tackle the knapsack cryptosystem efficiently Recently, an evolutionary discrete FA has been developed for solving travelling salesman problems Further improvement in performance can be obtained by using preferential directions in firefly movements.

vii. Semantic Web Composition

A hybrid FA has been developed by Pop et al. for selecting optimal solution in semantic web service composition.

viii. Chemical Phase equilibrium

For phase equilibrium calculations and stability analysis, FA was found to be the most reliable compared with other techniques.

ix. Clustering

Performance study for clustering also suggested that firefly algorithm is very efficient.

VI. CONCLUSION

Firefly Algorithm (FA) was applied to find the lowest makespan (Cmax) of five benchmarking JSSP datasets adopted from the OR-Library. Experimental design and analysis were carried out to investigate the appropriate parameters setting of the FA. The one-third fractional factorial experimental design can reduce the number of experimental runs by 66.67% compared with the conventional full factorial design. Ranges of FA parameters used by previous research were reviewed and investigated. The investigation was aimed to study the effect of the FA parameter setting on its performance before comparing the FA results between using and not using optimised parameter settings. In this research, the optimised setting of the FA parameters of nG, α, β0, and γ parameters was suggested at 100*25, 0.5, 1, and 0.1, respectively.

Moreover, the proposed algorithm with appropriate parameters setting produced the best-so-far schedule better than the FA without adopting parameter settings. It also found the best known solution in some cases. It should be noted that the appropriate parameter settings of the proposed algorithms may be case specific based on the nature and complexity of the problem domain

REFERENCE

[1]. P.Pongcharoen, C. Hicks, P. M.Braiden and D. J. Stewardson, "Determining optimum Genetic Algorithm parameters for scheduling the manufacturing and assembly of complex products", International Journal of Production Economics, 78, 2002, pp. 311-322.

[2]. Pongcharoen, C. Hicks and P. M. Braiden, "The development of genetic algorithms for the finite capacity scheduling of complex products, with multiple levels of product structure", European Journal of Operational Research, 152, 2004, pp. 215-225.

[3]. Artigues and D. Feillet, "A branch and bound method for the job-shop problem with sequence dependent setup times", Annals of Operations Research, 159, 2007, pp.135-159.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762
Available online at www.ijarmate.com

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE) Vol. 3, Special Issue 9, March 2017

[4].C. Artigues, M.-J. Huguetand P. Lopez, "Generalized disjunctive constraint propagation for solving the job shop problem with time lags", Engineering Applications of Artificial Intelligence, 24, 2007, pp. 220-231.

[5]. P. Baptiste, M. Flamini and F. Sourd, "Lagrangian bounds for just-in-time job-shop scheduling", Computers & Operations Research, 35, 2008, pp. 906- 915.

[6]. H. X. Chen and P. B. Luh, "An alternative framework to Lagrangian relaxation approach for job shop scheduling", European Journal of Operational Research, 149, 2003, pp. 499-512.

[7]. H. Gröflin and A. Klinkert. "A new neighborhood and tabu search for the Blocking Job Shop", Discrete Applied Mathematics, 157, 2009, pp. 3643-3655.

[8]. R. Zhang and C. Wu, "A simulated annealing algorithm based on block properties for the job shop scheduling problem with total weighted tardiness objective", Computers & Operations Research, 38, 2011, pp. 854- 867.

[9].X.-S. Yang, "Firefly Algorithms for multimodal optimization", Lecture Notes in Computer Science, 5792, 2009,pp. 169-178.

[10]. S. Lukasik and S. Zak, "Firefly Algorithm for continuous constrained optimization Tasks", Lecture Notes in Computer Science, 5796, 2009, pp. 97-106. [11] J. E. Beasley, "OR-library: distributing test problems by electronic mail", Journal of the Operational Research Society, 41, 1990, pp. 1069-1072.

[12]. X.-J. Wang, C.-Y. Zhang, L. Gao and P.-G. Li, "A survey and future trend of study on multi-objective scheduling", Proceedings of the 4th International Conference on Natural Computation, 2008., pp. 382-391.

[13]. G. Moslehi and M. Mahnam, "A Pareto approach to multiobjective flexible job-shop scheduling problem using particle swarm optimization and local search", International Journal of Production Economics, 129, 2011, pp. 14-22.

[14].L. Asadzadeh and K. Zamanifar, "An agent-based parallel approach for the job shop scheduling problem with genetic algorithms", Mathematical and ComputerModelling, 52, 2010, pp. 1957-1965.

[15].K. Ripon, C.-H. Tsang and S. Kwong, "An Evolutionary approach for solving the multi-objective job-shop scheduling problem", Studies in Computational Intelligence, 49, 2007, pp.165-195.

[16]. http://en.wikipedia.org

[17]. O. Holthaus and C. Rajendran, "Efficient dispatching rules for scheduling in a job shop", International Journal of Production Economics, 48, 1997, pp. 87-105.

[18] O. Bilkay, O. Anlagan and S. E. Kilic, "Job shop scheduling using fuzzy logic", International Journal of Advanced Manufacturing Technology, 23, 2004, pp. 606-619.

[19] M. Nakamura, K. Hachiman, H. Tohme, T. Okazaki and S. Tamaki, "Evolutionary computing of Petri Net structure for cyclic job shop scheduling", IEICE Transactions on Fundamentals of Electronics,

[20](N. Kawasaki, "Parametric study of thermal and chemical nonequilibrium nozzle flow," M.S. thesis, Dept. Electron. Eng., Osaka Univ., Osaka, Japan, 1993.
(Patent style)