IMPACT: International Journal of Research in
Engineering & Technology (IMPACT: IJRET)
ISSN(P): 2347-4599; ISSN(E): 2321-8843
Vol. 5, Issue 3, Mar 2017, 27-42

MULTI– OBJECTIVE FLEXIBLE OPEN SHOP SCHEDULING PROBLEM USING MODIFIED DISCRETE FIREFLY ALGORITHM (MFOSSP)

N. KAMATCHI[1] & D. SARAVANAN[2]

[1]Research Scholar, Bharathiyar University, Coimbatore, Tamil Nadu, India

[2]Professor of Mathematics, Karpaga Vinayaga College of Engineering and Technology, Chennai, Tamil Nadu, India

ABSTRACT

In this paper, flexible open shop scheduling problem using modified discrete firefly algorithm(MFOSSP-DFA) is studied in the case of optimizing different contradictory objectives consisting of (i) make span (ii) maximal machine workload (iii) total workload (iv) machine idle time (v) total tardiness. The main constraints of this scheduling problem are that each operation has to be processed without preemptive by exactly only one machine at one stage (i.e) no ordering constraints on operations. It is very difficult to adopt the situations where the undertakings making up an occupation can be performed in any request. Despite the fact that it is unrealistic to convey out more than one task at any particular time. It is non polynomial- hard problem. So its complexity is more. Because of its high complexity many researchers found difficult to solve using classical optimization methods. In this study, firefly algorithm is embraced to take care of the issue in which the machine task and operation arrangement are handled by constructing a correct conversion of the continuous functions as attractiveness, distance and movement into new discrete functions. Benchmark problems are used to evaluate and study the performance of the firefly algorithm. The final result shows that the firefly algorithm produced better results than other author’s algorithm

KEYWORDS: Flexible Open Shop Scheduling Problem, Modified Discrete Firefly Algorithm, Multi-Objective Optimization

INTRODUCTION

Booking is worried with the allotment of uncommon assets to exercises with the reason of optimizing one or more performance measures. The study of scheduling dates back to the 1950s. Investigators in operation research industrial designing and assembling building were defied with the issue of overseeing diverse activities taking place in a workshop. Efficient scheduling algorithms can lower the production cost in an assembling procedure and empower the organization to stay in aggressive markets. Among the problems subject to scheduling, Flexible open-shop scheduling problem has been highly concentrated on by investigators in recent years. The problem was first studied by Teo filo F, Gonzalez and Sartaj sahi in 1976. It is well known that this problem is non deterministic polynomial-time (NP) hard. The classical OSP consists of scheduling a set of jobs on a set of machines with the objective to minimize a certain criterion, subject to the constraints that each job has a specified processing order through all machines where are fixed and known in advance. In theoretical computer science and operation research, The open-shop scheduling issue (OSSP) is a booking issue in which a given arrangement of employments should each be prepared for given amount of time at each of a given set of machines in an arbitrary order, and the goal is to determine the time at which each job is to be processed at each machines (1).

More precisely, the input to the open –shop scheduling problem consists of a set of n jobs, another set of m machines, and a two-dimensional table of the amount of time each job should spend at each work station (possibly zero). Each job may be processed only at one machine at a time, and each machine can process only one job at a time. However, unlike the job shop problem in order in which the processing steps happen can very freely. The goal is to assign a time for each job to be processed by each machine, so that two jobs are assigned to the same machines at the same time, no job is assigned to two machines at the same time, and every job is assigned to each machine for the desired amount of time.

The flexible open shop scheduling problem (FOSSP) is an extension of the classical OSSP that allows an operation to be processed on any machine from a given set of alternative machines. This kind of scheduling problem reduces machine constraints, and enlarges searching scope of practicable solutions. It is closer to the real manufacturing situation (2).

FOSSP is more complex than classical OSSP because of the additional need to determine the assignment of machines for each operation.

A meta-heuristic approach for solving the flexible open shop scheduling problem (FOSSP) is more complex NP-hard problem. This problem consists of two sub problems the routing problem and the sequencing problem and is among the hardest combinatorial optimization problems. The routing sub problem assigns each operation to a machine among a set of capable machines. The sequencing sub problem involves sequencing the operation assigned to the machines in order to obtain a feasible schedule that minimizes predefined objectives

LITERATURE SURVEY

Most researches have been done on FJSP and several methods that include integrated approach and hierarchical, have been developed to solve FJSP. Most of the research FJSP has been concentrated on single objective alone. However, more than one objective must be considered simultaneously in the real life situation and these objectives often conflict with each other. In this paper, flexible open shop scheduling problem with five objectives (21) are taken. The open shop scheduling is similar to the job shop scheduling expect that a job may be processed on the machines in any sequence the job needs. FOSSP is the hardest combinatorial optimization problems in the branch of production scheduling. Now a day's biologically inspired algorithms are becoming powerful in modern numerical optimization especially for the NP hard problems. This paper aims to introduce the new firefly algorithm.

Adil you if, (9) considered the scheduling jobs on grid computing using firefly algorithm. He used efficient solution than min-max and max-min heuristics in many scheduling scenarios. Saeed yaghoubi,(12) addressed the multi objective project scheduling under resource constraints using under resource constraints using firefly algorithm. He solved with limited resources (RCPSP) modeling and Meta heuristic algorithm of firefly worm and the results were compared with NSGA II algorithm. The results give bets performance in solving proposed problems of RCPSP and multi objective RCPSP. Iztok fister,(19) considered a comprehensive review of firefly algorithm in order to use the algorithm to solve diverse problems. He modified or hybridized the original firefly algorithm. Xin-she yang (1) investigated about firefly algorithms for multi model optimization. He compared the FA with other Meta heuristic algorithms such as particle swarm optimization (PSO).

Haomiao Li, presented firefly algorithm on multi objective optimization of production scheduling system. He proposed FA to solve the production scheduling firefly such complex combinatorial optimization problems. Xin-she yang, Xingshitte(14), explained about firefly algorithm recent advances and applications. They concluded that Meta heuristics such as firefly algorithm are better than the optimal intermittent search strategy. They also analyses FA and their implications for higher-dimensional optimization problems. Alireza khatami,(10) addressed an efficient firefly algorithm for the flexible job shop scheduling problem. They adopted two optimization techniques including harmony search (HS), algorithm and firefly algorithm for flexible job shop scheduling problem.

Aphirak khadwilard, (9) investigated an application of firefly algorithm and its parameter setting for job shop scheduling. He designed and conducted computational experiment using five bench marking JSSP datasets from a classical OR-library. Dinakara Prasad reddy. P,(7) considered an application of firefly algorithm for combined economic load and emission dispatch. They proved that FA is easily to implement and much superior to other algorithms in terms of accuracy and efficiency. K.C.Udaiyakumar (11) investigated an application of firefly algorithm in job shop scheduling problem for minimization of make span. He used FA to find the make span minimization using 1-25 Lawrence problems as a bench marking from a classical OR-library.

Surafel Luleseged Tilahun, considered firefly algorithm for optimization problems with non continuous variables. S.Karthikeyan, (16) explained a hybrid discrete firefly algorithm for multiobjective flexible job shop scheduling problem with limited resource constraints. Song Huang, addressed multi objective flexible job shop scheduling problem using modified discrete PSO. They evaluated using Kacem instances and Brdata instances. S.V.Kamble, (21) considered hybrid multi objective particle swarm optimization for flexible job shop scheduling problem.

FOSSP PROBLEM FORMULATION

Problem Description

The open shop scheduling problem (OSSP) in which a given set of jobs must each be processed for given set of jobs must each be processed for given amounts of time at each of a given set of workstations in a self-assertive request and the objective is to decide the time at which each occupation is to be processed at each work station. The problem was first studied by Teofilo F Gonzalez and Sartaj Sahni in 1976. (29)

In flexible open-shop scheduling problem a set of n jobs J_1, J_2… Jn has to be processed on a set of m machines M_1, M_2 Mm. Each job J_i ($1 \leq i \leq n$) consists of a sequence of n_i operations. Each operation O_{ij} (i=1,2,…n ; j=1,2…n_i) of job (J_i) can be processed by one machine M_{ij} in the set of correct machines M_{ij}. X_{ijk} denotes the processing time of operation O_{ij} on machine $k \in M_{ij}$ (16).

The processing of a job on a machine is denoted as an operation and order in which the operation of a job or processed on the machines is immaterial that is no ordering constraints on operations. It is very difficult to adopt the situations where the tasks making up a job can be performed in any order. Even though it is not possible to carry out more than one task at any particular time. It is non-polynomial-hard problem so its complexity is more. It is one of the important method of scheduling, the routing of each job is up to the scheduler (i.e. it is open) that means if there is a job waiting for processing when a machine is free, then that machine is not allowed to remain idle. The open-shop scheduling is flexible so this bound is typically attained.

The main purpose of FOSSP is commonly used to find the best machine schedule for serving all jobs in order to optimize single criterion / objective or multi-scheduling objectives they are also known as open shop performance measures such as the make span minimization or mean flow time or the mean tardiness or earliness etc. [6]

ASSUMPTIONS

The following assumptions and Constraints are also considered [11]

- Jobs are released at time 0 and machines are available at time 0
- Job has the same priority.
- Jobs descriptions are known in advance.
- There is no priority restriction among operations of different jobs.
- No preemption is possible.
- Process time includes set up time as well.
- Operations cannot be disturbed.
- Machines cannot process the parallel job at a time.
- A job can be processed only once at a stage.
- Every job can be processed by not more than a single machine at a time.

CONSTRAINTS

In this model, no wait constraints is assumed and it is assumed that enough buffer space is made available just in case

In general, the constraints used in flexible open-shop scheduling are [7]

- Each machine can perform only one operation at a time.
- Completing time for the job.
- Every machine can process only one job at a time.
- A job can be moved in any order.
- No ordering constraints on operations.

Most of optimization techniques have been applied to solve the FOSSP. Classical methods based on mathematical model or numerical search such as branch and bound [10], [11]. And lagrangian relaxation which can assure the optimum solution these methods have been effectively and efficiently used to solve FOSSP even though these methods are used for moderating-large problem size (10x10) and to solve FOSSP but it may consume high computational time resources and therefore there is a computational limitation exist [19,20,21]. Recently a large size of FOSSP have been solved by an approximation optimization methods or Meta heuristics (for example Taboo search [7] and simulated annealing [8]) these methods usually follow stochastic steps in their iterative or search process.

For example, FOSSP with three jobs and four machines is shown in table 1, the numbers in the table present the processing time of operations and symbol '-' means the operation cannot be processed on the corresponding machine.

Table 1: An Example of 3 –Jobs 4-Machines Scheduling Problems Processing Times [4]

Job	Operation	Machine (M_k)			
	(O_{ik})	M_1	M_2	M_3	M_4
J_1	O_{11}	3	-	-	5
	O_{12}	4	8	-	3
	O_{13}	-	5	-	7
J_2	O_{21}	-	-	8	8
	O_{22}	7	-	-	-
	O_{23}	3	6	11	4
J_3	O_{31}	5	-	9	9
	O_{32}	7	4	5	1
	O_{33}	4	10	-	6

In this paper, the following objectives are to be minimized

- **Mc:** Make span (ie) the maximal completion time of machines or jobs
- **$M_{w:}$:** Maximal workload (ie) the maximum working time spent on any machine.
- **M_T:** Total work load of machines, which represents as total processing time over all Machines.
- **$M_{i:}$:** Total idle time, which is defined idle time of Machines.
- **Mt:** Total tardiness which is defined as lateness of jobs.

The notations used in this study are listed as follows

i,p - denotes of jobs i,p = 1,2,…n

j,q - denotes of operation sequence

j,q=1,2,…n

k - denotes of machines k=1,2,…m

n - Total number of jobs

m - Total number of machines

O_{ij} - the j th operation of job i

M_{ij} - the set of available machines for the

Operation O_{ij}

- X_{ijk} - processing time of operation O_{ij} on

  Machine k

- T_{ijk} - start time of operation O_{ij} on

Machine k

- I k - is the idle time of the machine Mk
- T_k - is the lateness of the machine M_k
- P_{ij} - completion time of the operation O_{ij}
- P_k - is the complete time of M_k
- Wk - is the workload of Mk.

Decision Variable

$$Q_{ijk} = \begin{cases} 1, & \text{if machine k is selected for the operation } O_{ij} \\ 0, & \text{otherwise} \end{cases}$$

Our model is presented as below:

- **M_c:** Make span (ie) the maximal completion time of

machines or jobs

$$M_C = \max_{1 \le k \le m}(P_k)$$

- **M_w :** Maximal workload (ie) the maximum working time

spent on any machine

$$M_W = \sum_{i=1}^{n} \sum_{j=1}^{ni} \sum_{k=1}^{m} X_{ijk} Y_{ijk}$$

- **M_T :** Total work load of machines, which represents as

total processing time over all Machines.

$$M_T = \max_{1 \le k \le m} \sum_{i=1}^{n} \sum_{k=1}^{ni} X_{ijk} Y_{ijk}$$

- **M_i :** Total idle time, which is defined idle

time of machines

$$M_i = \max_{1 \le k \le m}(I_k)$$

- **M_t :** Total tardiness which is defined as

lateness of jobs.

$$M_t = \max_{1 \le k \le m}(T_k)$$

Subject to:

$P_{ij} - P_{i\,(j-1)} \le X_{ijk}\, Q_{ijk},$ j=1, 2….n_i

$[(P_{pq} - P_{ij} - T_{pqk})Q_{pqk} X_{ijk} \geq 0] \vee [(P_{ij} - P_{pq} - T_{ijk})Q_{pji} Q_{ijk} \geq 0]$,

$\forall$ (i,j), (p,q), k

$\sum_{k \in Mij} Q_{ijk} = 1$, for all i

Equation (1) represents the minimization of maximal completion time of the machines. Equation (2) represents the minimization of maximal machine work load of all the machines. Equation (3) ensures the minimization of total work load of machines. Equation (4) represents minimization of total idle time of machines. Equation (5) represents total tardiness which is defined as lateness of jobs. Inequality (6) represents the operation precedent constraint. Inequality (7) represents that jobs can process of available machines. Equation (8) ensures that one machine could be selected from the set of machines for each operation. Many methods have been formulated to solve the multi objective optimization.

FIRE FLY ALGORITHM

Over the last 20 years new Meta heuristic algorithm has been introduced almost every year [1]. The nature inspired ones have become very interesting and distinguished. FA is one of the new swarm intelligence methods which was proposed by Xin- She Yang in 2008 and it is a kind of stochastic, nature inspired meta heuristic calculation that can be connected for taking care of the hardest streamlining issues (also NP-hard problem). Now a day's FA and its variants have been applied for solving many optimization and classification problems. It is used in almost all the branches of engineering areas: image processing, industrial optimization, wireless sensor networks, antenna design, business optimization, robotics, semantic web, chemistry and civil engineering.

Fire fly algorithm (FA) [1, 2] is a Meta heuristic algorithm, inspired by the flashing behavior of fireflies. The Fire fly Algorithm (FA) is a populace –based method to locate the worldwide ideal arrangement in view of swarm insight, investigating the foraging behavior of fireflies. Fire flies, which belong to the family of lampyridae, are small winged insects having ability of creating light with almost no heat and it is called a cold light. It flashes the light in order to attract mates. They are whispered to have a capacitor-like instrument, that bit by bit charges until the clear edge is reached, at which they discharge the energy in the form of light, subsequent to which the cycle repeats.

Fire fly was introduced by Xing –She Yang (2008). It is based on the firefly bugs behavior, including the light emission, light absorption and the mutual attraction, which was developed to solve the continuous optimization problems.

The flashing light of fireflies is a unique to the kind of species they belong to and varies from one type of species to the other. There are about 2000 firefly spices and most fireflies produce short and rhythmic flash. The model of flashes is often unique for a particular species. The flashing light is produced by a procedure of bioluminescence and the genuine elements of such flagging frameworks are still debating. However two fundamental functions of such flashes are to attract mating partners and to attract potential prey. In addition, flashing may be serving as a protective warning mechanism. The rhythmic flash, the rate of flashing and the amount of time from part of the flag framework that unites both genders. Females react to a male's interesting example of flashing in the same spices, while in some species such as photuris, female fireflies can mimic the mating flashing pattern of other species so as the lure and eat the male fireflies who may mistake the flashes as a potential suitable mate. The flashing light can be formulated in a manner that it is related with the target capacity to be upgraded, which makes, it possible to formulate new optimization algorithms.

When nature inspires algorithm such as particles and swarm optimization (PSO) [17] as firefly algorithm are the most power fly algorithm for optimization.

This paper focuses on yang's [2] implementation of the FA. This algorithm is based on a physical formula of light intensity L that decreases with the increase in the square of the distance $d^{2.}$ However as the distance from the light source increases, the light absorption causes that light becomes weaker and weaker. These phenomena can be associated with the objective function to be optimized. As a result the base FA can be formulated as illustrated in the following algorithm

The development of firefly-inspired algorithm was based on following idealized rules [14]

- The firefly S attracts all other fireflies and is attracted to all other fireflies
- The less bright firefly is attracted and moved to the brighter one
- The brightness decreases when the distance between fireflies is increased
- The brightest firefly moves randomly (no other fireflies can attract it)
- The firefly particles are randomly distributed in the search space.

As indicated by above tenets there are two primary focuses in firefly calculation, the engaging quality of the firefly and the movement towards the attractive firefly [9]

Using These Rules, As Pseudo –Code of the Modified Discrete Firefly Algorithm

- Define an initialize parameters f(S), $S= (S_1, S_2,\dots S_d)^T$
- Generate initial population of fireflies S_i (i=1,2….m)
- Determine light intensity for S_i by calculating f (S_i)
- Define light absorption coefficient ϒ
- **While** t<maximum Generation
- Make a copy of the generated firefly population for move function
- **For** i=1 : m all m fireflies
- **For** j=1 : I all m fireflies
- **If** ($L_j > L_i$)
- Move fireflies i and j according to attractiveness evaluating new solutions and updating light intensity for next emphasis.
- **End** if
- **End** for j
- **End** for i
- Sorting the fireflies to discover the present best

- **End while**
- Begin post process on best results obtained
  - All fireflies are unisex
  - Their attractiveness is proportional to their intensity
  - The light intensity of a fire fly is affected and determined by the landscape of the fitness function.

Characteristics of the Modified Discrete Firefly Algorithm

There are two important issues in firefly algorithm, the first one is the variation of light intensity and the second one is the formulation of the attractiveness.

We can assume always that the attractiveness of a firefly is determined by its brightness which in turn is associated with the encoded objective function [10].

In the simplest case for maximum optimization problems, the brightness L of a firefly at a particular location S can be chosen as

$L(S) \alpha f(S)$

However, the attractiveness β is relative; it should be found according to the spectator or judged by alternate fireflies. Hence, it will shift with the distance d_{ij} between firefly I and firefly j. In addition, light intensity (L) decreases with the separation from its source, and light is additionally caught up in the media, so we ought to permit the attractiveness to vary with the degree of absorption. In the simplest form, the light intensity (L_d) varies according to the inverse square law $\mathbf{L(d) = L_s/d^2}$ where L_s is the intensity at the source. For a given medium with a fixed light absorption coefficient ϒ, the light intensity L varies with the distance d. i.e. $\mathbf{L = L_0\, e^{-\Upsilon d}}$, where $\mathbf{L_0}$ is the original light intensity. In order to avoid the singularity at d=0 in the expression $\mathbf{L_s/d^2}$, the combined effect of both the backwards square law and assimilation can be approximated utilizing the accompanying Gaussian shape

$$\mathbf{L(d) = L_0\, e^{-\Upsilon d^2}}$$

Where L_0 is the original light intensity at the distance d=0 and ϒ is the the rules we known that in our simulation we suppose the attractiveness of firefly is proportional to the light intensity L. So we can define the firefly's light attractive coefficient β in the similar way as the light coefficient L. i.e

$$\beta(d) = \beta_0\, \mathbf{e}^{-\Upsilon d^2}$$

Where β_0 is the attractiveness at d =0.

In the implementation, the actual form of attractiveness function β (d) can be any monotonically diminishing capacities, for example, the accompanying summed up shape

$$\beta(d) = \beta_0 e^{-\gamma d^{n}}, (n \geq 1)$$

For a fixed ϒ, the characteristic length becomes

$$r = \Upsilon^{-1/n} \rightarrow 1 \text{ as } n \rightarrow \infty.$$

Conversely, for a given length scale r in an optimization problem, the parameter ϒ can be used as a typical initial value that is $\Upsilon = 1 / r^{n.}$

Distance

The Cartesian distance between any two firefly i and j at S_i and S_j respectively [8,13]

$$d_{ij} = \| S_{i,h} - S_{j,h} \| = \sqrt{\sum_{h=1}^{d} (S_i - S_j)^2}$$

Where $S_{i,h}$ is the h^{th} component of the spatial co ordinate, S_i of the i^{th} firefly, in 2D case we have

$$d_{ij} = \sqrt{(S_i - S_j)^2 + (P_i - P_j)^2}$$

The movement towards attractive firefly

The amount of movement of firefly I to another more attractive firefly j is determined by

$$S_i(t+1) = S_i(t) + \beta_0 \mathbf{e}^{-\Upsilon d^2} (S_i(t) - S_j(t)) + \alpha\, \varepsilon_i$$

Where $S_i(t)$ is the current location of firefly i, $\beta_0 \mathbf{e}^{-\Upsilon d^2} (S_i(t) - S_j(t))$ is due to the attraction of the firefly S_j and $\alpha\, \varepsilon_i$ a randomization with the vector of random variables

ε_i being drawn ; so if $\beta_0 = 0$ then it turns out to be a simple random from different distributions such as the uniform distribution, Gaussian distribution and Levy flight. Here α is the scaling parameter that controls the step size and it should be linked with the interests of the problems [8, 13].

The algorithm compares the attractiveness of the new firefly position with old one. If the new position gives higher attractiveness value, the firefly is moved to the new position; generally the firefly will stay in the present position. The end criterion of the DFA is based on an arbitrary predefined number of iterations or predefined fitness value

The brightest firefly moves randomly based on the following equation

$$S_i(t+1) = S_i(t) + \alpha\, \varepsilon_i$$

This new Meta heuristics DFA algorithm has been applied by many of the researchers for tackling enhancement issues, dominant part of them have been defined into numerical equations.

In this paper, the parameters of firefly algorithm are number of fireflies (m), number of generations/iterations (G), the light absorption coefficient (ϒ), randomization parameter (α), and attractiveness value (β_0) have been defined.

The Frame Work of Modified DFA

The modified discrete firefly algorithm (11) is expected to obtain good performance in solving multi-objective flexible open shop scheduling problem. The frame work of the modified DFA is illustrated in the following figure. During the operation process, the initialization is done then fitness of all fireflies from the objective function is evaluated. If the stop criterion is met, the non dominated solution is the optimal results. Otherwise, update light intensity of fireflies. The algorithm is repeated until a termination criterion is met.

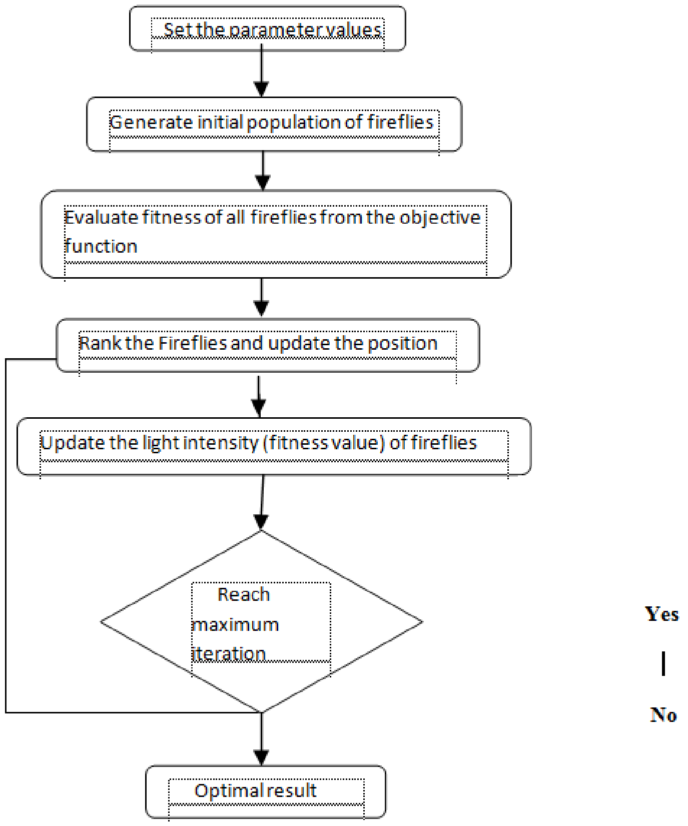

Figure 1

Computational Results

In this paper, MFOSSP (DFA) is compared with some other algorithm. This algorithm I coded in java and implemented on a personal computer with 3.2 GHz and 4GB RAM. The parameter of the discrete algorithm is as follows

- Population size is set as 100 for 4X5, 200 for 8x8, 300 for 10x10, and 400 for 15x15.
- Maximum number of generation =150
- Maximum local search iteration =100
- Attractiveness of fireflies β_0=1.0
- Light absorption coefficient ϒ=0.1
- Randomization parameter α = 1.0

In this section multiple runs on the same problem is carried out to obtain meaningful results. The MFOSSP (DFA) is compared with some well known famous algorithms in literature. These algorithms include “AIA” of Baheri et al(10),”SM” of Xing et al., ”HDPSO” of Zhang et al., ”HMPSO” of S.V.Kamble et al., ”MOPSO” of Song Huang et al., “MOGA OF Wang et al., In these issues the amount of jobs spans from 10 to 20, the number of machines ranges from 4 to 20 are taken to compare other results. Which are frequently tried on as of late distributed literary works, are utilized to assess the legitimacy and execution?

Table 3 lists non dominated solutions obtained by the modified discrete firefly algorithm and several recently published algorithms. For the 8 jobs x 8 machines instance, the 10 jobs x 10 machines instance and 15 jobs x 15 machines instance, all the solutions obtained by eight algorithms are non dominated solutions.

Table 2: Comparison of MFOSSP Modified DFA to Other Algorithms

Size	Objective	Xing	AIA(1)	HGA	HDPSO	HMPSO	MOGA	MOPSO	MFOSSP (DFA)
8 X 8	M_c	14	14	14	14	14	15	16	14
	M_w	12	12	12	12	12	11	13	11
	M_T	17	77	77	77	77	81	73	73
	M_i	-	-	-	-	21	-	-	20
	M_r	-	-	-	-	19	-	-	18
10 X 10	M_c	7	7	7	7	7	8	8	7
	M_w	6	6	5	5	6	5	7	5
	M_T	42	43	43	43	43	42	41	41
	M_i	-	-	-	-	11	-	-	9
	M_r	-	-	-	-	20	-	-	17
15 X 15	M_c	-	12	12	11	11	-	-	11
	M_w	-	11	11	11	11	-	-	10
	M_T	-	91	91	91	91	-	-	89
	M_i	-	-	-	-	20	-	-	19
	M_r	-	-	-	-	38	-	-	37

Table 3 lists non dominated solutions obtained by the modified algorithm and several recently published algorithms. For 10 jobs x6 machines instance, the 15 jobs x 8 machines instance, the 15 jobs x 4 machines instance, the 10 jobs x 15 machines instance, the 20 jobs x 5 machines instance, all the solutions obtained by five algorithms are non dominated solutions.

Table 3: Performance Comparison between MFOSSP (DFA) with Other Algorithms

	Objective	AIA	Xing	HSFLA	MOGA	HDPSO	HMPSO	MFOSSP (DFA)
MK01	M_c	40	42	40	40	40	40	40
	M_w	36	42	37	36	36	36	34
	M_T	171	162	165	169	167	163	161
	M_i	-		-	-	-	33	31
	M_r	-	-	-	-	-	72	70
MK02	M_c	26	28	26	26	27	27	26
	M_w	26	28	26	26	27	26	24
	M_T	154	155	157	151	145	144	143
	M_i	-	-	-	-	-	35	34
	M_r	-	-	-	-	-	85	83
MK03	M_c	204	204	204	204	210	204	204
	M_w	204	204	204	199	210	210	202
	M_T	1207	854	852	855	848	852	848
	M_i	-	-	-	-	-	90	90
	M_r	-	-	-	-	-	145	143
MK04	M_c	60	68	62	66	61	61	60
	M_w	60	67	61	63	60	60	60
	M_T	403	372	364	345	366	365	342
	M_i	-	-	-	-	-	85	84
	M_r	-	-	-	-	-	121	121
MK05	M_c	173	177	173	173	173	175	173
	M_w	173	177	173	173	173	173	172

	M_T	686	702	685	683	683	682	681
	M_i	-	-	-	-	-	80	80
	M_r	-	-	-	-	-	111	110
MK06	M_c	63	75	64	62	62	62	61
	M_w	56	67	55	55	58	58	53
	M_T	470	431	403	424	412	412	412
	M_i	-	-	-	-	-	75	73
	M_r	-	-	-	-	-	94	90
MK07	M_c	140	150	141	139	141	140	138
	M_w	140	150	141	139	141	141	140
	M_T	695	717	696	693	692	693	691
	M_i	-	-	-	-	-	45	45
	M_r	-	-	-	-	-	52	51
MK08	M_c	214	227	215	214	211	211	210
	M_w	203	221	210	204	207	207	203
	M_T	2121	1989	1957	2082	1998	1995	1990
	M_i	-	-	-	-	-	98	97
	M_r	-	-	-	-	-	143	141

CONCLUSIONS

In this paper multi objective FOSSP using modified discrete firefly algorithm with five objectives is investigated to meet the real world production situation. Most of the researchers focused with single and three objectives. But there exist other objectives in the real world such as minimization of total idle time and total tardiness to decrease production costs and lateness of the job. The novelty modified discrete firefly algorithm is used to solve the FOSSP with multiple objectives. For applying ordinary firefly algorithm, here we deal with discrete version of the continuous function such as movement, distance, attractiveness to find correct position of firefly. Experimental results show that MFOSSP using modified DFA gives better results as compared to the algorithms given in literature. In future, researchers can focus on multi objectives with some other objectives (more than 5) of FOSSP (DFA) or the fuzzy version of that.

REFERENCES

1. X. S. Yang,” Firefly algorithms for multi model optimization, stochastic Algorithms; Foundations and Applications”, Proceedings of the 5 th international conference Stochastic algorithms: foundations and applications, Sapporon, 2009, pp.169-178.

2. A. Bagheri, Zandiesh,” An artificial immune algorithm for the flexible job shop scheduling problem,” Future Generation Computer Systems, Vol 26(4), 2010, pp.533-541.

3. L. N. Xing, Y. Chen, K. W. Yang,” An efficient search method for multi-objective flexible job shop scheduling problems,”Journal of Intell.Manuf., 2009, pp: 283-293.

4. X. S. Yang, Nature inspired metaheuristic algorithms, Luniver press, London, Vol 1, 2008, pp. 36-50.

5. Kratika Chandra, Sudhir singh, ”Firefly Algorithm to tackle two dimensional been pressing Problem “, International Journal of Computer Science and Information Technologies, Vol 5(4), 2014, pp. 5368-5373.

6. Dinakara Prasad Reddy P, J N Chandra Sekar “Utilization of firefly Algorithm for joined Economic Load and Emission Dispatch ”International journal on Recent and Innovation Trends in Computing and Communication

Vol 2(8),2014, pp. 2448-2452.

7. 7.M.K. Sayadi, R.Ramezanin, N. Ghaffari- Nassab, “A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems”, International Journals of Industrial Engineering Computations, Vol 1(1), 2010, pp. 1-10

8. Aphirak Khadwilard, Sirikarn Chansombat, Thatchai Thepphakorn,”Application of firefly Algorithm and Its parameter setting for job shop scheduling”, The journal of Industrial Technology Vol 8(1),2012, pp.2555-2564.

9. Alireza khatami, Seyed Habib A.Rahmati, Abbas Ahamdi, “An efficient firefly algorithm for the flexible job shop scheduling problem”, International conference on Industrial Engineering and Operations Management Dubai, United Arab Emirates (UAE) Vol 5, 2015, pp. 2144-2146.

10. Sharik Farook, P.sangameswara Raju,” Evolutionary hybrid genetic –firefly algorithm for global optimization”, International journal of computational engineering & management, Vol 16(3),2013.pp. 37-45.

11. K.C. Udaiyakumar, M. Chandrasekaran,” Application of firefly algorithm in job shop scheduling problem for minimization of makespan” Procedia Engineering - Elesvier Vol 97,2014, pp.1798-1807.

12. J. Gao, L. Sun, and M. Gen,” A hybrid genetic and variable neighbourhood descent algorithm for flexible job shop scheduling problem,” Computers operation Research, Vol 35(9),2008, pp.2892-2907.

13. H. Groflin, A. Klinkert, “A new neighborhood and tabu search for the Blocking job shop”, Discrete applied mathematics Vol 157, 2009, pp. 3643-3655.

14. R. Zhang, C. Cuv, A simulated annealing algorithm based on block properties for the job shop problem with total weighted tardiness objective, computers & Operations Research, Vol 38,2011, pp. 854-867.

15. S. Karthikeyan, P. Asokan, S. Nickloas, ”A hybrid discrete firefly algorithm for multi objective flexible job shop scheduling problem with limited resource constraints”, International Journal of Advanced Manuf Technol”, Vol 72, 2014, pp.1567 – 1579.

16. X-S. Yang, “Nature inspired metaheuristic Algorithms, Luniver Press”, Vol 1, 2008, pp. 60-78.

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

18. Iztok Fister, Iztok Fister Jr, Xin –She Yang,Janez Bress, “A comprehensive review of firefly algorithm.”,”Swarm &Evolutional computation”, Elsevier Vol 13,2013, pp. 34-46.

19. C. Artigues M-J. Huguet and P. Lopez,” Generalized disjunctive constraint propagation for solving the job shop problem with time lags”, Engineering Application of Artificial Intelligence, Vol 24,2007, pp. 220-231.

20. Kamble SV, Mane SU, Umbarkar AJ “Hybrid multi- objective particle swarm optimization for flexible job shop scheduling problem”, International Journal of Intelligence Systems and Applications, Vol 7(4),2015, pp. 54-61.

21. Balaraju G, venkatesh S, Reddy BSP “Multi – objective flexible job shop scheduling using hybrid differential evolution algorithm”, International Journal of Internet. Manuf. Serv, Vol3(3),2014, pp. 226-243.

22. Kacem I, hammadi S, Borne P,” Pareto – optimality approach for flexible job shop scheduling problems, Hybridization of evolutionary algorithms and fuzzy logic “,.Math Comput Simul, Vol 60(3),2002,pp. 245-276.

23. M. Chandirasekaran, P. Ashokan, S. Kumanan, S. Arunnachalam, ”Application of selective breeding Algorithm for solving jobshop scheduling problems”, International Journals of Manufacturing Science and Technology USA Vol 9(12), 2007, pp. 103-118.

24. S. Yang, D.W Wang, T. Chai, G. Kendall,” An improved constraint satisfaction adaptive neural network for job shop scheduling”, journals of scheduling Vol 13,2010, pp.17-38

25. S.X.Yang, D.W. Wang, “A new adaptive neural network and heuristics hybrid approach for job shop scheduling”, Computers & Operations Research, Vol 28,2001, pp. 955-971.

26. Saced Yaghoubi, Meisam Jafari Eskandari, Meysam Farahmand Nazer,” Multi Objective Project scheduling under Resource Constraints using Algorithm of firefly”, Vol (3),2015, pp. 347-358.

27. T-L. Lin, S-J. Harng, T-W. Kao, Y-H. Chin R-S Run, R-J.Chen, and J-L. Lail, I.H. Kua, “An efficient job shop scheduling algorithm based on particle swarm optimization”, expert systems with applications, vol 37,2010, pp. 2629-2636.

28. G. Moslehi and M. Mahnam, “A pareto approach to multi objective flexiable job shop scheduling problem using particle swarm optimization and local search”, International journals of production Economics Vol 129, 2011, pp.14-22.

29. L. Asadzadeh and K. Zamanifar, “An agent-based parallel approach for the job shop scheduling problem with genetic algorithms”, Mathematical and computer modeling, Vol 52, 2010, pp. 1957-1965.

30. M. Nagamani, & E. Chandrasekaran, ”Single Objective For Partial flexible Open shop scheduling Problem Using Hybrid particle Swarm Opyimization Algorithms”, International Journal of Scienticfic Technology, Vol 35, 2015, pp. 5645-5652.

31. Xue H, Zhang P, Wei S, Yang L, “An improved immune algorithm for multi objective flexible job-shop scheduling.” J. Netw, Vol 9(10), 2014, pp. 2843 -2850.

32. Zhang G,Gao L,Shi Y “An efficient genetic algorithm for the flexible job shop scheduling problem Expert syst Appl, Vol 38 s (4),2011, pp.3563 -3563.