Volume 1, Number 3, 2012

International Journal of Research in Industrial Engineering

journal homepage: www.nvlscience.com/index.php/ijrie

A Novel Discrete Electromagnetism-Like for Fuzzy Open Shop Scheduling Problem with Parallel Machines to Minimize Makespan

A. Jafari[1], **P. Chiniforooshan**[2,*], **F. Mousavinejad**[1], **Sh. Shahparvari**[3]

[1]Department of Industrial Engineering, Science and Culture University, Tehran, Iran
[2]Department of Industrial Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
[3]Department of Industrial Engineering, Kish International Campus, University of Tehran, Kish Island, Iran

A R T I C L E   I N F O

Article history :
Received:
August 23, 2012
Revised:
November 3, 2012
Accepted:
December 2, 2012

Keywords :
Fuzzy open shop scheduling problem,
Mathematical Model,
Discrete electromagnetism-like algorithm.

A B S T R A C T

In this paper, we consider the fuzzy open shop scheduling problem with parallel machines in each working stage where processing times are vague and are represented by fuzzy numbers. An open shop scheduling problem with parallel machines in each working stage under this condition is close to the real production scheduling conditions. A mixed-integer fuzzy programming (MIFP) model is presented to formulate this problem with the objective of minimizing makespan. To solve small-sized instances, an interactive fuzzy satisfying solution procedure is applied. Since this problem is known as a class of NP-hard, a novel discrete electromagnetism-like (DEM) is proposed to solve medium to large size examples. The DEM algorithm employs a completely difference approach. It makes use the crossover operators to calculate force and move particle is used. We employ Taguchi method to evaluate the effects of different operators and parameters on the performance of DEM algorithm. Finally to assess the performance of the algorithm, the results are compared with an existing EM algorithm from the literature and benchmark problems. The result exhibited the ability of the proposed DEM algorithm to converge to the efficient solutions.

1. Introduction

Scheduling includes the allocation and sequencing of activities that need to be performed in a set of limited available resources [1]. Generally, these problems can be defined by a set of n jobs that need to be processed by a set of m working stages. Several production workshops are defined by their different attributes that belong to the processing rout of their jobs. One of them is open shop scheduling problem (OSSP) in which each job has to be processed on each one of the m stages. However, some of these processing times may be zero. There are no restrictions on the routings of each job. So in this case, the scheduler is allowed to determine a route for each job and different jobs may have different routes [2]. OSSP can be appeared in a service environment such as a network of diagnostic testing facilities in hospital where patients can do their tests at various test centers in an arbitrary order [3]. Other applications that can do pointed out to them are as follows: automobile repair, satellite communications,

[*]Corresponding author
E-mail address: p.chiniforooshan@yahoo.com

teacher–class assignments, semiconductor manufacturing, quality control centers, and flight timetabling for airlines [4, 5]. However real production floors rarely use a single machine for each operation and sometimes processing times due to measurement errors and human impact on production processes are vague, but usually the open shop scheduling problems in which it is commonly assumed that each stage has one machine and processing time is deterministic. Indeed the purpose of replicating machines in parallel is to leveling the speed of the stages, to increase the revenue and capacity of the shop floor, or to reduce the impact of bottleneck stages on the overall shop efficiency.

In open shop scheduling problems, the processing order of operations is arbitrary. Thus, the solution space of an open shop problem is larger than that of job shop and flow shop problems. If the number of machines is more than two, then OSSP is NP-hard [6]. In open shop scheduling problem with parallel identical machines, three decisions must be taken following: (1) Processing route determination (2) Job sequence determination (3)Job assignment to machines inside each stage. Therefore, this problem is at least as hard the hard OSSP and is a class of NP-hard problems. To solve such a problem in medium and large size, use of exact methods is often unpractical and requires the use of efficient metaheuristic methods.

Naderi er al. [7] formulated an open shop scheduling problem to minimizing total tardiness. They presented four mixed integer linear problems for $OSSP_s$ and then investigated the complexity of these models. They designed Genetic Algorithm (GA) and Variable Neighborhood Search (VNS) and investigated the effect of various operators on the GA using Taguchi method. Liaw[8] considered the problem of scheduling preemptive open shop to minimize total tardiness. He developed an efficient constructive heuristic for solving large-sized problems. He also presented a branch-and-bound algorithm for solving medium-sized problems. Mosheiov& Oron[9] addressed batch scheduling problems on an *m*-machine open shop to minimizing makespan and flow time. They assumed identical processing time jobs, machine and sequence-independent setup times and batch availability. Su et al. [10] studied two models of two-stage processing with flow shop at the first stage followed by open shop at the second stage to minimize the makespan. They proposed an integer programming model and a branch and bound algorithm for model 1 and a lower bound developed for model 2 as a benchmarks for the heuristic algorithms. Naderi et al. [11] investigated an open shop that each stage consists of a set of parallel machines to minimize total completion times. They suggested a mixed integer linear programming model for this problem. Also, they presented a memetic algorithm (MA) for solving the problem.

Panahi&Tavakoli-Moghadam [12] offered an efficient method based on Multi-Objective Simulated Annealing (MOSA) and ant colony optimization (ACO) for an open shop scheduling problem with minimizing makespan and total tardiness. They also applied a decoding operator to improve the quality of produced schedules. Sha et al. [13] proposed a Multi-objective particle swarm optimization (MOPSO) algorithm for $OSSP_s$ with multi-objective. Due to the discrete of scheduling problems and PSO application in continuous optimization problems modified the particle position representation, particle velocity, and particle movement.

Electromagnetism-like (EM) is one of new methods in the field of optimization based on the swarm intelligence. This optimization method works inspired by the existing rules of electrostatic systems and inherently used for continuous minimization problems [14]. This algorithm makes use of a small number of parameters and appropriate values of the parameters can be easily determined by performing several simulation. The EM has been used for solving various problems such as routing problems [15, 16], Fuzzy solving equations [14], Neural Network analysis [17], Multifunctional control [18] and scheduling problems [19-21]. Naderi et al. [22] presented EM for OSSP with sequence-dependent setup times. They incorporated a fast search engine and a simple simulated annealing to improve algorithm performance.

Following a brief review of the literature is provided in the Table1. The table contains type of the objective function and solution methods.

Table 1. Category articles in the term of the objective function and solution methods.

Reference	**Objective function**		**Solution Methods(metaheuristic)**
	Single objective	Multi objective	
Noori-Darvish et.al [23]		total weighted tardiness , total weighted completion times	MOPSO
Hashemi Doulabi et.al[24]	the sum of weighted earliness/tardiness penalties		HSA
Yu et al. [25]	makespan		SA
Naderi et al. [7]	makespan		GA, VNS
Roshanaei et al. [26]	makespan		SA
Sha et.al [13]		makespan, total flow time and Machine idle time	MOPSO
Naderi et al. [22]	Total completion times		EM
Gonzalez et.al [27]	Expected makespan		GA
Matta [3]	makespan		GA
Low& Yeh [28]	Total tardiness		GA
Seraj& Tavakkoli-Moghadam[29]		total weighted tardiness , total weighted completion times	TS
Zhang& Wu [30]	total weighted tardiness		SA & GA
Sha& Hsub[31]	makespan		PSO
Huang& Lin [32]	total weighted tardy jobs		TS
Senthilkumar&Shahabudeen[33]	makespan		GA
Blum& Sampels [34]	makespan		ACO
Liaw [35]	total tardiness		TS
Blum [36]	makespan		Beam-ACO
Liaw [37]	makespan		TS

The inherent uncertainty in the parameters of models is increasingly being taken into account in various fields. Moreover, there are several factors involved in real-world scheduling problems that are often vague or uncertain in nature. This is especially true when the factors that made human, considered into the problems. Thus, parameters are often faced with uncertainties.

Accordingly, production scheduling problems can be divided into two general categories: deterministic scheduling and uncertain scheduling problems [38, 39]. There are basically two approaches to deal with uncertainties [40], such as the stochastic-probabilistic theory and possibility theory or fuzzy set theory [41, 42].

In this practice, fuzzy set theory is applied for dealing with the uncertainties in scheduling problems. It provides an appropriate alternative framework for the mathematical modeling for real-world systems and offers several advantages associated with the use of heuristic approaches:

- Probability theory needs considerable knowledge about the statistical distribution of the unknown parameters. Vs, fuzzy theory provide an effective way to model uncertainty even when no historical information is available [43].
- Using stochastic-probabilistic theory includes comprehensive computation and requires thorough knowledge on the statistical distribution of the uncertain time parameters [44].
- The use of fuzzy set theory reduces the computational complexity of the scheduling problem compared with the stochastic probabilistic theory [45].
- One of the capabilities of fuzzy theory, the use of fuzzy rules in heuristic algorithms [39].

Konno& Ishii [46] presented a model for a preemptive open shop scheduling problem with fuzzy resource and allowable time. Their problem had bi-criteria to be maximized, i.e., minimum degree of satisfaction with respect to the intervals of processing jobs and, minimum satisfaction degree of resource amounts applied in the processing intervals. Palacios et al. [27] investigated the $OSSP_s$ where processing times were fuzzy. They suggested a GA algorithm to minimize average maximum completion time of jobs. Noori-Darvish et al. [23] addressed a $OSSP_s$ with Sequence-dependent setup times, fuzzy processing times and fuzzy due dates. They presented a new bi-objective possibilistic mixed-integer linear programming model to minimize total weighted tardiness and total weighted completion times. For solving small-sized instances, an interactive fuzzy multi-objective decision making (FMODM) approach, called TH method proposed by Torabi and Hassini[47], is applied.

In this study, we present a mixed-integer fuzzy programming (MIFP) model for OSSP with a set of parallel machines at each stage. Furthermore, we devise a novel discrete electromagnetism-like algorithm to solve the considered problem and use the benchmark of the Taillard[48] as lower bound to evaluate the performance of the algorithm. The rest of the paper is as follows. A MIFP formulation of the problem under study is set out in Section 2. In Section 3, we suggest an interactive fuzzy satisfying solution procedure to the proposed model. Computational results indicate that the MIFP model can be solved in reasonable CPU time to run, for only limited number of jobs. For problems with larger number of jobs, we describe a Discrete Electromagnetism-Like algorithm (DEM) in Section 4. We describe the experimental design to evaluate the posed method in section 5. Finally, concluding remarks are given in Section 6.

2. A MIFLP formulation of the problem

In formulating scheduling models, parameters such as job processing, ready and setup times are generally considered as deterministic values. However, in real-world situations, these parameters are often uncertain values. Time required to process parts on machines cannot be determined exactly due to measurement errors and the involvement of human activities in the manufacturing process. Due to the inconsistency in the performance of operators and machines at the shop floor, repeated measurement of the system's parameters provides a certain range of values. Therefore, the information that we have about the model parameters is often vague and imprecise [49, 38]. In a situation where we lack enough information to define the parameters, qualitative expression described by linguistic variables like 'too short' or 'about 100' are often used based on ambiguous data. In fact, fuzzy set theory provides the tools to deal with uncertain model parameters, which are not as deterministic values but rather as interval values representing estimates [50].

In this section, we formulated a mixed-integer fuzzy linear programming (MIFLP) model for open shop scheduling problem with a set of parallel machines at each stage that presented by Yimer & Demirli [51] . The parameter that is related to uncertain time (processing) is offered by triangular fuzzy sets $\tilde{p}_{j,i} = (p^{o}_{j,i}, p^{m}_{j,i}, p^{o}_{j,i})$. Here, the objective will be minimizing the makespan C_{max}, that is, the time lag from the start of the first operation until the end of the last one. A problem often denoted $FuzzPO_m||C_{max}$ in the literature.

2.1. Nomenclature

We need to introduce the notations including parameters, indices and variables used in the model. The parameters and indices are defined in Tables 2, 3 and 4.

Table 2. Indices used in the models

Index	For	Scale
j, k	Jobs	$\{1,2,....n\}$
i, l	Stages	$\{1,2,....m\}$
r	Machines	$\{1,2,....m_i\}$

Table 3. Deterministic Parameters used in the models

Deterministic Parameters	Description
n	The number of jobs
m	The number of stages
m_i	The number of identical machines in stage i
$o_{j,i}$	The operation of job j in stage i
M	A large positive number

Table 4. Fuzzy parameters used in the models

Fuzzy parameters	Description
$\tilde{p}_{j,i}$	The processing time of $o_{j,i}$
$\tilde{c}_{j,i}$	The completion time of $o_{j,i}$
$\tilde{z}(\vec{x})$	imprecise makespan

Marketing above the symbol indicates that these variables represent vague values or fuzzy numbers.

Binary integer

$X_{j,\,i,\,l}$ 1 if O_{ji} is processed after O_{jl}

or 0 otherwise. $i \in \{1, 2 \dots m-1\}, l > i.$

$Y_{j,i,k}$ 1 if O_{ji} is processed after O_{ki}

or 0 otherwise. $j \in \{1, 2, \dots, n-1\}, k > j.$

$Z_{j,i,r}$ 1 if O_{ji} is processed on rth machine in stage i

or 0 otherwise. $r \in \{1, 2, \dots m_i\}.$

General variables

$\chi_f(\vec{x})$ fuzzy solution space

$\chi_c(\vec{x})$ crisp solution space

$\vec{x}$ a feasible solution vector of decision variables $\vec{x} \in \chi_f(\vec{x}) \cup \chi_c(\vec{x})$

λ fuzzy goal satisfying level $(0 < \lambda < 1)$

2.2.The proposed model

Fuzzy goal function: The objective is to minimize the completion time of the last delivery among the n jobs, commonly referred to as the makespan. It is related to the *throughput* of the schedule. Because throughput is defined as the amount of work completed per unit time, and because the amount of work in the n-job model is fixed, we maximize throughput by minimizing the makespan[52].

The fuzzy objective function (1) gives the imprecise makespan of all jobs:

$$\tilde{Z}(\vec{x}) = \tilde{C}_{\max} \tag{1}$$

Crisp solution space: The constraint related to the each job is processed by only one machine at each stage does not depend on the fuzzy time variables. So, it is considered to be crisp.

$$\chi_c(\vec{x}) \equiv \sum_{r=1}^{m_i} Z_{j,i,r} = 1 \qquad \forall j, i \tag{2}$$

$$Z_{j,i,r} \in \{0,1\} \qquad \forall i, j, r \tag{11}$$

Fuzzy solution space: constrains related to the time imprecise parameter, belong to the space of fuzzy solution. The fuzzy constraints include:

$$\chi_f(\vec{x}) \cong \tilde{C}_{j,i} \geq \tilde{P}_{j,i} \qquad \forall j,i \qquad (3)$$

$$\tilde{C}_{j,i} \geq \tilde{C}_{j,l} + \tilde{P}_{j,i} - M(1 - x_{j,i,l}) \qquad \forall j,i \in \{1,2,...,m-1\}, i < l \qquad (4)$$

$$C_{j,l} \geq C_{j,i} + \tilde{P}_{j,i} - M \times (x_{j,i,l}) \qquad \forall j,i \in \{1,2,...,m-1\}, i < l \qquad (5)$$

$$C_{j,i} \geq C_{k,i} + \tilde{P}_{j,i} - M \times (1 - Y_{j,i,k}) - M \times (2 - Z_{j,i,r} - Z_{k,i,r}) \qquad (6)$$

$$\forall i,r,\ j \in \{1,2,...,n-1\},\ j < k$$

$$\tilde{C}_{k,i} \geq \tilde{C}_{j,i} + \tilde{P}_{j,i} - M \times (Y_{j,i,k}) - M \times (2 - Z_{j,i,r} - Z_{k,i,r}) \qquad (7)$$

$$\forall i,r,\ j \in \{1,2,...,n-1\},\ j < k$$

$$X_{j,i,l} \in \{0,1\} \qquad \forall j,i \in \{1,2,....m-1\},\ i < l \qquad (9)$$

$$Y_{j,i,k} \in \{0,1\} \qquad \forall i,\ j \in \{1,2,....n-1\},\ j < k \qquad (10)$$

Constraint set (3) assures that the completion time of each operation must be greater than its processing time. Constraint sets (4) and (5) specify the relation between each pair of operations of a job. For example, the completion time of $O_{j,i}$ must be greater than that of $O_{j,l}$ if job *j* visits stage *i* after stage l. Similarly, constraint sets (6) and (7) define the relation between the completion times of each pair of jobs in each stage. For example, the completion time of $O_{j,i}$ must be greater than that of $O_{k,i}$ if job k proceeds job j in stage *i* if they are processed by the same machine. Constraint sets (9)–(10) define the decision variables.

2.3.Fuzzy goal programming

The imprecise and vague time-dependent parameters are expressed by fuzzy sets. The degrees of membership functions for the fuzzy numbers parameters are defined based on psychic judgments. Symmetric triangular fuzzy number is the simplest form function of fuzzy numbers, which is made of two basic estimations, the most possible value, and the maximum deviation from it [53]. For example, a symmetric triangular membership function for a fuzzy processing time $\tilde{p}_{j,i}$ can be defined by:

$$\tilde{P}_{j,i} \cong P^m_{j,i} \pm P^\delta_{j,i} = (P^{m-\delta}_{j,i}, P^m_{j,i}, P^{m+\delta}_{j,i}) = (P^l_{j,i}, P^m_{j,i}, P^r_{j,i})$$

Values of the left and right of the center have the lowest likely to belong to the set of possible values, so, their membership degree is zero. The most likelihood value, which is in the middle of the bound, has the highest degree of membership $\lfloor \mu_{\tilde{a}}(P^m_{j,i}) = 1 \rfloor$.Other values in the span of $\tilde{P}_{j,i}$, will assume to be a linearly varying membership function in the interval [0, 1]. Figure1 shows a symmetric triangular membership function for $\tilde{P}_{j,i}$. Also, the fuzzy objective function can be defined in terms of two deterministic objective functions for makespan:

$$\tilde{Z}(\vec{x}) \simeq Z^{m}(\vec{x}) \pm Z^{\delta}(\vec{x})$$

where

$$Z^{m}(\vec{x}) = C_{\max}^{m} \qquad Z^{\delta}(\vec{x}) = C_{\max}^{\delta} \tag{13}$$

Similarly, the fuzzy solution space $\chi_f(\vec{x})$ given by Eq. (3)-(7) can be defined as a combination of two sets of crisp constraints which are as follows:

$$\chi_f(\vec{x}) \cong \chi_m(\vec{x}) \pm \chi_\delta(\vec{x}) \tag{14}$$

where

$$\begin{aligned}
&\chi_m(\vec{x}) \cong C^{m}{}_{j,i} \ge P^{m}{}_{j,i} && \forall j, i \\
&C^{m}{}_{j,i} \ge C^{m}{}_{j,l} + P^{m}{}_{j,i} - M(1 - x_{j,i,l}) && \forall j, i \in \{1,2,\dots,m-1\}, i < l \\
&C^{m}{}_{j,l} \ge C^{m}{}_{j,i} + P^{m}{}_{j,i} - M \times (x_{j,i,l}) && \forall j, i \in \{1,2,\dots,m-1\}, i < l \\
&C^{m}{}_{j,i} \ge C^{m}{}_{k,i} + P^{m}{}_{j,i} - M \times (1 - Y_{j,i,k}) - M \times (2 - Z_{j,i,r} - Z_{k,i,r}) \\
&\forall i, r,\ j \in \{1,2,\dots,n-1\}, j < k \\
&C^{m}{}_{k,i} \ge C^{m}{}_{j,i} + P^{m}{}_{j,i} - M \times (Y_{j,i,k}) - M \times (2 - Z_{j,i,r} - Z_{k,i,r}) \\
&\forall i, r,\ j \in \{1,2,\dots,n-1\}, j < k \\
&X_{j,i,l} \in \{0,1\} && \forall j, i \in \{1,2,\dots m-1\}, i < l \\
&Y_{j,i,k} \in \{0,1\} && \forall i, j \in \{1,2,\dots n-1\}, j < k
\end{aligned}$$

and

$$\begin{aligned}
&\chi_\delta(\vec{x}) \cong C^{\delta}{}_{j,i} \ge P^{\delta}{}_{j,i} && \forall j, i \\
&C^{\delta}{}_{j,i} \ge C^{\delta}{}_{j,l} + P^{\delta}{}_{j,i} - M(1 - x_{j,i,l}) && \forall j, i \in \{1,2,\dots,m-1\}, i < l \\
&C^{\delta}{}_{j,l} \ge C^{\delta}{}_{j,i} + P^{\delta}{}_{j,i} - M \times (x_{j,i,l}) && \forall j, i \in \{1,2,\dots,m-1\}, i < l \\
&C^{\delta}{}_{k,i} \ge C^{\delta}{}_{j,i} + P^{\delta}{}_{j,i} - M \times (Y_{j,i,k}) - M \times (2 - Z_{j,i,r} - Z_{k,i,r}) \\
&\forall i, r,\ j \in \{1,2,\dots,n-1\}, j < k \\
&C^{\delta}{}_{j,i} \ge C^{\delta}{}_{k,i} + P^{\delta}{}_{j,i} - M \times (1 - Y_{j,i,k}) - M \times (2 - Z_{j,i,r} - Z_{k,i,r}) \\
&\forall i, r,\ j \in \{1,2,\dots,n-1\}, j < k \\
&X_{j,i,l} \in \{0,1\} && \forall j, i \in \{1,2,\dots m-1\}, i < l \\
&Y_{j,i,k} \in \{0,1\} && \forall i, j \in \{1,2,\dots n-1\}, j < k
\end{aligned} \tag{15}$$

A fuzzy decision is obtained by considering the intersection of the fuzzy objective and the whole space solution. When information related to the objective function and constrains sets is vague, the problem can be formulated as a goal fuzzy programming problem which is described below:

Find: $\vec{x}$ (16)

To satisfy: $\tilde{Z}(\vec{x}) \cong Z^m(\vec{x})$ *and* $\vec{x} \in \chi_c(\vec{x}) \cup \chi_f(\vec{x})$

where

$\vec{x}$ is a solution vector of decision variables in feasible solution space $\chi_c(\vec{x}) \cup \chi_f(\vec{x})$, and $Z^m(\vec{x})$ related to the goal fuzzy objective. The symbol "≅" in the constrain indicates that the resulting makespan $(\tilde{Z}(\vec{x}))$ should be around expected value $Z^m(\vec{x})$ with some symmetric deviation $Z^\delta(\vec{x})$ on both sides.

3. Solution approach

For the problem is presented in previous section, the objective function will be a triangle symmetric possibility distribution. This function can be defined by three vertices $\tilde{Z}(\vec{x}) = (Z^l(\vec{x}), Z^m(\vec{x}), Z^r(\vec{x}))$.

In fact minimization $\tilde{Z}(\vec{x})$ is obtained by moving the three vertices towards origin; under this condition, the problem becomes a certain multi-objective linear programming by converting $\tilde{Z}(\vec{x})$ into three interdependent crisp objectives [53].

Indeed, three objective functions: Minimizing the most possible value $Z_1(\vec{x})$, maximize the possibility to obtain lower objective function $Z_2(\vec{x})$ and to minimize the risk of getting high objective function $Z_3(\vec{x})$:

$$
\begin{aligned}
&Min \quad Z_1(\vec{x}) = Z^m(\vec{x}) \\
&Max \quad Z_2(\vec{x}) = Z^{m-l}(\vec{x}) = Z^\delta(\vec{x}) \\
&Min \quad Z_3(\vec{x}) = Z^{r-m}(\vec{x}) = Z^\delta(\vec{x}) \\
&Subject: \quad \vec{x} \in \chi_c(\vec{x}) \cup \chi_\delta(\vec{x})
\end{aligned} \tag{17}
$$

$Z^\delta(\vec{x})$, Represents the symmetric deviation from the fuzzy number.

By using fuzzy decision making of Bellman and Zadeh[54] and fuzzy programming method of Zimmermann[55], MOLP problem can be transformed into single goal linear programming problem. The initial values are obtained for the positive and negative ideal solutions by solving each of the above functions separately:

$$Z_1^{PIS}(\vec{x}) = Min \quad Z^m(\vec{x})$$
$$Z_1^{NIS}(\vec{x}) = Max \quad Z^m(\vec{x})$$
$$Z_2^{PIS}(\vec{x}) = Z_3^{NIS} = Max \quad Z^\delta(\vec{x})$$
$$Z_2^{NIS}(\vec{x}) = Z_3^{PIS} = Min \quad Z^\delta(\vec{x}) \tag{18}$$

By using membership functions outlined below, the objective functions are converted into fuzzy goals.

$$\mu_1(Z_1) = \frac{Z_1^{NIS} - Z_1}{Z_1^{NIS} - Z_1^{PIS}}$$
$$\mu_2(Z_2) = \frac{Z_2 - Z_2^{NIS}}{Z_2^{PIS} - Z_2^{NIS}}$$
$$\mu_3(Z_3) = \frac{Z_3^{NIS} - Z_3}{Z_3^{NIS} - Z_3^{PIS}} \tag{19}$$

Applying membership functions expressed and the fuzzy decision of Bellman and Zadeh[54],the MOLP problem can be represented:

$$Maxi\min e: \quad \min\{\mu_1(Z_1), \mu_2(Z_2), \mu_3(Z_3)\}$$
$$Subject\ to: \quad \vec{x} \in \chi_c(\vec{x}) \cup \chi_f(\vec{x}) \tag{20}$$

Finally, by introducing an auxiliary fuzzy goals satisfying level $\lambda\,(0 \le \lambda \le 1)$, the MOLP problem can be reduced to single objective formal LP problem of Zimmermann [55]:

$$Maxi\min e: \quad \lambda$$
$$Subject\ to: \quad \lambda \le \mu_i(Z_i) \quad for\ i = 1,2,3$$
$$\vec{x} \in \chi_c(\vec{x}) \cup \chi_f(\vec{x}) \tag{21}$$

In Eq. (21) high value of λ indicates that the objective functions are optimized with a high degree of confidence.

4. Proposed discrete Electromagnetism-like algorithm

As mentioned in section1, the problem considered in our study belongs to class of NP-hard problems. So, for solving medium to large size problems, we suggest an efficient DEM algorithm.

4.1.Classic EM

Electromagnetism-like (EM) is one of new methods in the field of optimization based on the swarm intelligence. It was introduced by Birbil and Fang [14]. The main idea of EM is based on the attraction-repulsion mechanism of electromagnetism theory (Coulomb’s law). In this algorithm each solution is considered as a charged particle and the charge of particle is belonged to its objective function value. The scale of absorption or desorption on candidate solutions in the population is determined by this charge. The route of this charge for particle *i* is determined by adding the exerted pressure of the other particles on particle *i*. In this mechanism, a particle with superior objective function value attracts the others ones, while a particle with inferior objective function value excretes the others ones. The charge for each particle is calculated by the following formula:

$$q^i = \exp\left(-n \frac{f(y^i) - f(y^{best})}{\sum_{k=1}^{popsize} \left(f(y^k) - f(y^{best}) \right)} \right) \quad (22)$$

In Eq. (22), $f(y^i)$ and $f(y^{best})$ denote the objective function value of particle *i* and the best solution. The force of particle *i* is calculated as follows:

$$F^i = \sum_{j \neq i}^{popzize} \begin{cases} (y^j - y^i) \dfrac{q^i q^j}{\| y^i - y^j \|^2} & f(y^j) > f(y^i) \\ (y^i - y^j) \dfrac{q^i q^j}{\| y^i - y^j \|^2} & f(y^j) \leq f(y^i) \end{cases}, \qquad \forall i \quad (23)$$

The general scheme of EM is shown in Fig. 1. It includes four phases: initialize, computing of total force exerted over each particle, moving particles in the direction of the force and, local search.

```
Procedure Electromagnetism algorithm
Initialize ()
While (hasn't met stop criterion) do
    Local Search ()
   Calculate total force F ()
   Move particle by F ()
   Evaluate particles ()
End while
```

Figure 1. The fundamental procedures of EM

4.2.Proposed DEM

Although the results of applying the EM was very satisfactory for continuous space problems but these results was not enough for discrete space problems [14].the main reason that EM cannot be used for discrete problems is that its operators (force calculation and movement) are not compatible with this type of spaces.

Since the scheduling problems are in the category of discrete problems, in this research, we have developed the classical EM to DEM that is described in:

4.2.1. Encoding and decoding outlines, initialization

Coding scheme is a procedure which makes an algorithm is able to identify a solution. One of these schemes is permutation list. In this method a string that contains n×m array is designed. In fact, we produce a random permutation of the elements of the set $A = \{1,2 \ldots n \times m\}$ (*n*=jobs number, *m*= stages number). Suppose, n=2 and m=3, the string is generated by the following:

o_{11}	o_{12}	o_{13}	o_{21}	o_{22}	o_{23}
5	4	1	6	2	3

Figure 2. Illustrates a permutation list.

Fig.2. indicates that an operation is placed in sequence according to its corresponding number in the second string. According to the Fig. 2, at first job3 is processed over stage1, and then job2 is processed over stage2, etc.

Non delay schedule is applied to decode the permutation list.

Non delay schedule: this schedule is investigated under the terms makespan, the search space is reduced by this decoding so that the optimal solution does not disappear from it. We apply procedure proposed in [56] and later used in [31]: all operations are placed in a set (*U*) including unscheduled operations. We calculate *y* which equals the minimum of the earliest possible starting times (s_{ij}) of operations in *U*. All the operations whose starting time is equal to *y* are assigned to a set called *R*. Among the operations in *R*, the operation O^* with the earliest relative position in permutation θ is scheduled and extracted from *U*. In this decoding, we assign the jobs to the first available machine at every stage. Fig. 3 illustrates the decoding scheme.

Procedure non-delay schedule

U=all operation in given permutation θ

While $U \neq \phi$

$y = \min\{s_{ij} \text{ of } O_{ij} \mid O_{ij} \in U\}$

$R = \{O_{ij} \mid s_{ij} = y, O_{ij} \in U\}$ %*R* is a set of operations whose starting times are equal to *y*

Choose O^* from the set of *R* with the earliest relative position in permutation θ

Extract O^* from *U*

Figure 3. The procedure of decoding scheme by the principal of non-delay schedule

4.2.2. Calculating total force and particles movement

This study applies the modified EM that proposed by Debels et.al. [57], to obtain the total force exerted on the particle. In this procedure dose not determined the force exerted on

particle i from particle j by using the fixed charge of q_i and q_j. In place of, q_{ij} is related to the relative difference of $f(x^i)$ and $f(x^j)$.
In the proposed algorithm, the roulette-wheel is used to select particle *i* and particle *j*. After selecting two particles the particle charge is computed as follows:

$$q_{ij} = \frac{f(x_i) - f(x_j)}{f(x^{worst}) - f(x^{best})} \tag{24}$$

If the objective value $f(x^i)$ is larger than $f(x^j)$, particle *j* will attract particle *i*. from the other point of view, when $f(x^i) < f(x^j)$, particle *i* will attract particle *j* and there is no action when $f(x^i) = f(x^j)$. More, the force exerted on particle *i* by particle *j* is calculated as follows:

$$F_{ij} = (x_j \ominus x_i) \odot q_{ij} \tag{25}$$

Now, the particle move from solution x_i to $x_i \oplus F_{ij}$ in the direction of x_j .The definitions of the operator $\ominus$and operator $\oplus$ are as follow.
The subtract operator $\ominus$. This is applied as Position-based Crossover and Linear Order Crossover that following in:
Before ruining operator$\ominus$, operato$\odot$ is used to determine the number of dimensions and uses the following expression:

$$L = | q_{ij} \times n |$$

In the above equation n indicates the number of dimension.
Position-based Crossover: If $q_{ij} > 0$, value *L* is rounded to up, then to the size of *L*, is randomly selected dimension from particle *i* and moved to new particle and the rest of numbers chosen from particle *j*. if $q_{ij} < 0$ above procedure is reversed (place two particles are reversed).Fig.4 is shown the implementation steps of the operator. Suppose the permutation of the particle *i* and *j* is the following and $q_{ij} = 0.23$, so the number dimensions of each particle are 6. Then *L*=6×0.23=1.38. Because $q_{ij} > 0$ we randomly select 2 dimensions of particle *j*, that dimensions 1 and 5 are selected and transferred to new particle. We remove the numbers of particle *j* that are selected from particle *i* and place the rest into the new particle according to their same order in particle *i*.
Liner order crossover (LOX): at first introduced by Falkenauer & Bouffouix [58], works as follows:
A subsequence of operations from a parent is randomly selected, and then is created the initial part of the offspring by copying the subsequence into the corresponding position of it. The operations that are currently in the subsequence from the second parent are deleted and finally the operations are placed into the unfixed positions of the offspring from left to right according to the order of the sequence. This procedure is shown in Fig. 5.

In fact, the difference of two crossover operators is that in position-based crossover *L* dimension is randomly selected but in Linear Order crossover part of the parent chromosomes are selected length *L* and are copied into the offspring.

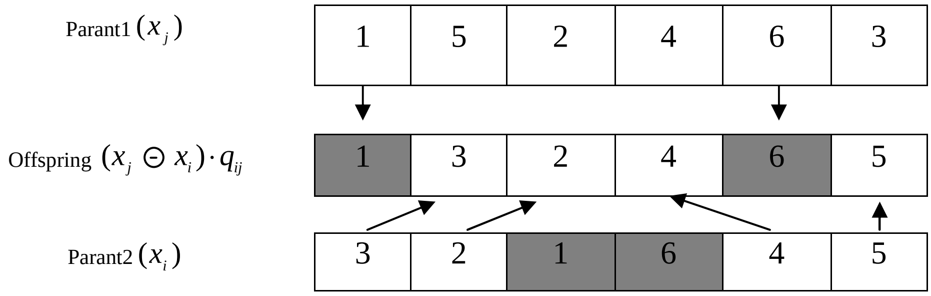

Figure 4. Illustration of the Position-based Crossover operator

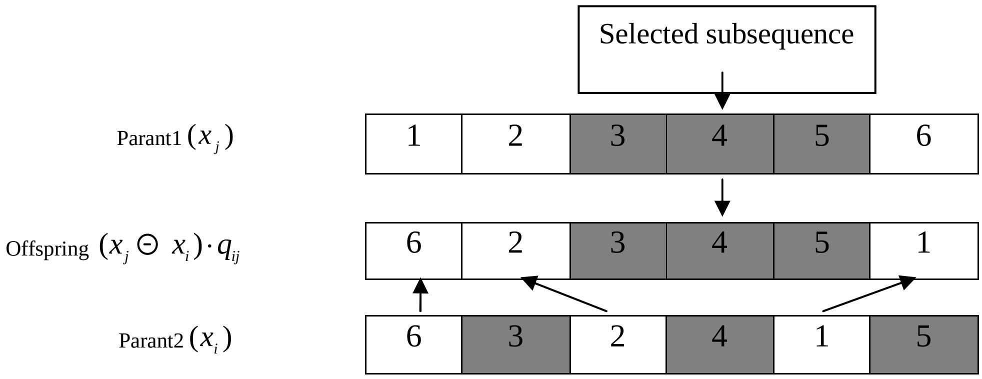

Figure 5. Illustration of the LOX crossover operator

The add operator $\oplus$. This operator can be considered as Extension of Precedence Preservative Crossover [59] and called EPPX. EPPX is shown as follows: a string of equal length as the particle is produced then all of its elements are filled with random number at [0, 1]. This string defines the order in which elements are successively drawn from x_i and F_{ij}. The offspring is initially empty. Start with the first element of x_i and F_{ij}, when the *k*th element is selected, if $q_{ij} < R(q_{ij} > R)$ corresponding number in $x_i, (F_{ij})$ is transferred to offspring, if selected element comes from $x_i, (F_{ij})$ and *d*th $(d \geq k)$ element in $F_{ij}, (x_i)$, then delete element from x_i, F_{ij} and shift the elements of $F_{ij}, (x_i)$ between point k and d right once. The step is repeated until x_i, F_{ij} are empty and offspring is obtained. Fig.6 describes an illustration of EPPX. Suppose $q_{ij} = 0.46$.

F_{ij}	1	5	2	4	6	3
string	0.2	0.1	0.5	0.56	0.43	0.82
x_i	3	2	1	6	4	5
offspring	3					
		1	5	2	4	6
		2	1	6	4	5
	3	2				
			1	5	4	6
			1	6	4	5
			⋮			
offspring	3	2	1	5	6	4

Figure 6. An illustration of EPPX

4.2.3. Local procedure

This algorithm selects the best solution in the each iteration and perturbs the solution by moving the two points at random, and then finds its objective value. If the objective value of the new solution is better than the best solution, the new solution will replace it. Otherwise If the objective value of the new solution is worse than the best solution, and is better than the worst solution, it will replace the worst solution. So the worst solution is found and this new solution will replace it. Therefore, it attempts to improve average solution iteratively.

5. Algorithm’s calibration

Parameter setting is an important part of the designing algorithms because we can adapt algorithm to the problem. So in this section, the behavior of DEM with different operator and parameters are appraised. Several DEM_s can be obtained with different combinations of parameters and operators.

Between the alternative experimental examinations the Taguchi method is more efficient for calibrating the algorithm because it can survey generous decision variables with a small number of experiments [60]. In the Taguchi method, factors are categorized into two main groups: controllable and noise factors. Noise factors are those that we have no direct control over them. Since the removal of these factors is often impossible, the Taguchi method seeks to minimize the impact of these factors and to determine the optimal level of controllable factors [61]. Taguchi studies the impact of factors on the response variable variance and then based on the mean response variable determines the impact of the factors that are not effective on the variance. The main reason why Taguchi method is regarded as the design is that it tries to adjust the stability of the algorithm so that uses the ratio S/N which in fact determines ratio Signal to Noise. Taguchi classifies all objective functions into three groups: the smaller-the-better type, the larger-the-better type, and nominal-is-best type. Considering that almost all functions in scheduling are the smaller-the-better type, their corresponding S/N ratio [62] is.

$$S/N \; ratio = -10\log_{10}(objective\;\; function)^2 \tag{26}$$

Table 5 shows the factors that need to be tuned with their levels.

From standard table of orthogonal arrays, the L_{18} is chosen as an orthogonal array for the algorithm. We generate a set of 25 instances as follows: we have 5 combinations (4×4, 5×5, 7×7, 10×10, and 15×15). There exist five replicates with different m_i (number of parallel machine in each stage), that is generated from a uniform distribution over (2, 4) for each combination thus summing up to 25 instances. The processing times are randomly generated from a uniform distribution over (1, 99). In order to conduct the experiments, we implement DEM in C# and run on a PC with 2.0 GHz Intel Core 2 Duo and 2 GB of RAM memory. We use relative percentage deviation (RPD) as a common performance measure to compare the methods. RPD is calculated as such:

$$RPD = \frac{A\lg_{sol} - Min_{sol}}{Min_{sol}} \cdot 100 \tag{27}$$

where Alg_{sol} is C_{max} obtained for a given algorithm and instance and Min_{sol} is the lowest C_{max} for a given instance obtained by any of the algorithms.

Table 5. Factors and their Levels

Factors	Level	Type
Crossover Operator	2	(1) Position-based Crossover (2) Linear Order Crossover
Population Size	3	10 20 40
Number of Local Search	3	15 25 50

We run DEM for each trail of Taguchi experiment. Table 6 shows the results that are transformed into S/N ratio. Fig.7 shows the mean ratio obtained for each level of the factors. The optimal level of factors becomes: Crossover: Position-based, Population-Size= 20, Local Search number = 50.

Table 6. The results are transformed into S/N ratio

Cross_Type	Pop_Size	Local_No	Trail 1	Trail 2	Trail 3	Trail 4	Trail 5	S/N
1	10	15	20.33	25.61	28.82	26.69	21.90	-27.91
1	10	25	21.06	22.07	27.80	18.72	26.41	-27.41
1	10	50	19.51	17.63	16.34	18.00	18.37	-25.11
1	20	15	10.84	14.13	11.58	12.69	14.08	-22.10
1	20	25	10.51	10.74	12.70	10.59	13.95	-21.42
1	20	50	9.91	8.65	6.72	7.92	5.54	-17.95
1	40	15	17.06	16.80	16.18	10.61	13.94	-23.59
1	40	25	15.11	17.13	16.56	15.81	15.89	-24.14
1	40	50	18.54	16.57	19.59	18.83	20.18	-25.47
2	10	15	25.60	25.11	20.27	20.55	22.56	-27.21
2	10	25	24.19	24.93	23.45	25.92	25.31	-27.88
2	10	50	23.78	20.72	19.28	17.74	21.28	-26.30
2	20	15	18.67	16.54	14.54	15.60	14.92	-24.15
2	20	25	13.69	14.64	15.16	14.74	17.25	-23.60
2	20	50	12.07	13.77	12.55	13.83	10.96	-22.07
2	40	15	16.70	15.16	16.69	16.78	14.27	-24.06
2	40	25	17.76	17.67	19.00	19.53	18.03	-25.30
2	40	50	21.12	20.70	18.03	20.41	19.55	-26.02

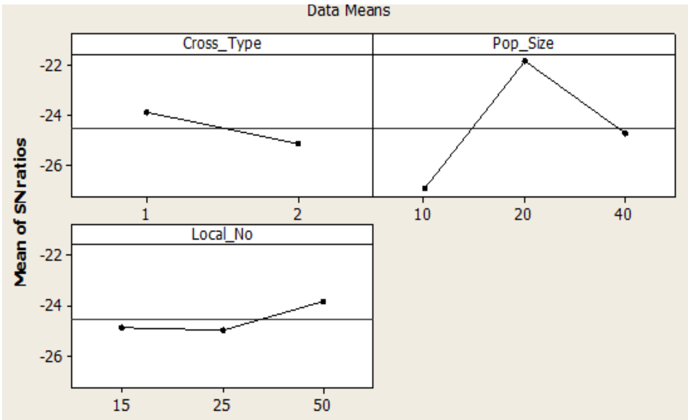

Figure 7. The mean S/N ratio plot for each level of the factors

6. Experimental result

In this section, we intend to appraise the FMILP model and the proposed DEM algorithm. At first small-sized problems are solved to evaluate the mathematical model and also DEM algorithm against the results obtained from the model. We implement FMILP model in CPLEX 10.1 and the algorithms in MATLAB 7.6 and run on a PC with 2.0 GHz Intel Core 2 Duo and 2 GB of RAM memory. In this paper, the stopping criterion used when testing all instances with the algorithms is n × m × 0.4 s.

For the experimental study we use [63] and generate a set of fuzzy problem instances from well-known benchmark problems from Taillard [48]. In fact each crisp processing time *t* is converted into a symmetric fuzzy processing time *p(t)* so that a certain value is $p^2 = t$ and p^1 , p^3 are random values, symmetric w.r.t and generate so the TFN's maximum range of fuzziness is 30% of p^2.under these conditions, the optimal solution of the crisp problem provides a lower bound for expected fuzzy makespan [63]. 10 fuzzy instances were generated from each crisp problem instance. So in total there are 250 problem instances.

Computational results for Small to medium size and for large size Table 7, 8 and 9 are respectively collected. In these tables Lower Bound, MILP Model, DEM and ME respectively indicate to solve crisp instances by Taillard [48], solve fuzzy instances with CPLEX10.1, solve fuzzy instances using the proposed algorithm and solve fuzzy instances using the suggested algorithm by Chang et al. [64].The mathematical model is allowed a maximum of 1000 s of computational time.

Since the investigated problem is the fuzzy open shop scheduling problem with parallel machines at the stage so at first we suppose m_i=1 to evaluate the model and algorithm. In this case, first the small size problems have been studied with Tillard's benchmark which is a lower bound for our problem. As shown in the Table 8, by taking that we have added fuzzy assumption, the difference of the MILP model with lower bound is negligible. Also according to the obtained values by algorithm DEM can be realize its performance.

Table 7. Small Size Experiments (Fuzzy without Parallel, , m_i=1)

Problem	Lower Bound	MILP Model		DEM		MEM	
		C_{max}	CPU Time	C_{max}	CPU Limit Time	C_{max}	CPU Limit Time
Tail_4×4_1	193	205.28	20.10	206.79	6.40	208.87	6.40
Tail_4×4_2	236	251.10	19.58	251.129	6.40	254.65	6.40
Tail_4×4_3	271	284.29	18.59	287.01	6.40	287.81	6.40
Tail_4×4_4	250	266.92	22.29	264.246	6.40	269.85	6.40
Tail_4×4_5	295	311.74	20.25	312.322	6.40	318.15	6.40
Tail_4×4_6	189	200.79	12.48	198.511	6.40	198.91	6.40
Tail_4×4_7	201	212.95	15.52	213.995	6.40	211.29	6.40
Tail_4×4_8	217	228.96	12.80	229.576	6.40	230.60	6.40
Tail_4×4_9	261	276.91	24.77	275.302	6.40	282.77	6.40
Tail_4×4_10	217	229.46	20.33	232.099	6.40	235.61	6.40
Tail_5×5_1	300	321.90	54.91	325.22	10.00	325.48	10.00
Tail_5×5_2	262	281.12	47.09	284.135	10.00	284.12	10.00
Tail_5×5_3	323	344.89	74.99	345.025	10.00	345.15	10.00
Tail_5×5_4	310	328.72	45.08	326.903	10.00	333.96	10.00
Tail_5×5_5	326	350.29	88.53	350.54	10.00	349.91	10.00
Tail_5×5_6	312	334.65	88.55	339.991	10.00	341.41	10.00
Tail_5×5_7	303	322.06	37.23	323.352	10.00	328.40	10.00
Tail_5×5_8	300	318.956	81.23	323.346	10.00	321.983	10.00
Tail_5×5_9	353	373.912	95.11	380.006	10.00	377.009	10.00
Tail_5×5_10	326	347.477	81.83	350.934	10.00	352.562	10.00
Tail_7×7_1	435	514.438	1000	466.589	19.6	470.552	19.6
Tail_7×7_2	443	512.912	1000	482.432	19.6	487.322	19.6
Tail_7×7_3	468	550.592	1000	510.785	19.6	510.356	19.6
Tail_7×7_4	463	522.257	1000	509.027	19.6	499.813	19.6
Tail_7×7_5	416	468.926	1000	459.151	19.6	465.336	19.6
Tail_7×7_6	451	537.998	1000	483.587	19.6	504.25	19.6
Tail_7×7_7	422	495.524	1000	456.472	19.6	462.28	19.6
Tail_7×7_8	424	507.255	1000	469.284	19.6	469.137	19.6
Tail_7×7_9	458	520.755	1000	502.55	19.6	502.515	19.6
Tail_7×7_10	398	461.154	1000	424.128	19.6	438.36	19.6

For large-size problem we have evaluated our algorithm and the Modified EM (MEM) of the Change et al. [64] with the lower bound of Tillard's benchmark. We have used RPD as a common performance measure to compare the methods. The RPD of DEM is between 3.19846 and 12.36. The mean RPD of DEM is 9.218% and the mean RPD of MEM is 12.79%. According to the mean RPD and consider the fuzziness of the OSSP can be realized the effectiveness of DEM. Since that the mean RPD of the DEM algorithm is better than MEM algorithm then can be realized that DEM is more efficient than MEM. These results are shown in Table 8. As previously described, we'll investigate the fuzzy open shop scheduling problem with parallel machines so in the here investigate states that $m_i \neq 1$. In this case we have considered the performance of the DEM by lower bond that is obtained of solving model in deterministic mode. We performed each of the examples 5 times for different examples. As shown in the Table 9. RPD of the DEM is between 7.00928 and13.3854, and the mean RPD of DEM is 9.12073%., As regards some of this difference is due to the fuzzy nature of the problem, because the lower bound is obtained in crisp condition, can understand that the DEM algorithm has good performance. Also the mean RPD of MEM indicates that

DEM algorithm is better than MEM algorithm (the mean RPD of MEM is smaller than MEM algorithm). These results are shown in Table 9.

Table 8. Large Size Experiments (Fuzzy without Parallel , m_i=1)

Problem	Lower Bound	DEM		MEM	
		Cmax	RPD%	Cmax	RPD%
Tail_10×10_1	637	701.191	10.077	683.105	7.238
Tail_10×10_2	588	660.677	12.36	645.313	9.747
Tail_10×10_3	598	662.027	10.7069	677.996	13.38
Tail_10×10_4	577	635.445	10.1291	647.966	12.3
Tail_10×10_5	640	660.47	3.19846	680.993	6.405
Tail_10×10_6	538	599.37	11.407	631.998	17.47
Tail_10×10_7	616	659.793	7.10933	699.251	13.51
Tail_10×10_8	595	636.899	7.04183	635.84	6.864
Tail_10×10_9	595	631.478	6.1307	642.759	8.027
Tail_10×10_10	596	635.514	6.62982	656.426	10.14
Tail_15×15_1	937	1054.57	12.5472	1067.52	13.93
Tail_15×15_2	918	998.459	8.76455	1032.65	12.49
Tail_15×15_3	871	936.259	7.49239	1000.52	14.87
Tail_15×15_4	934	1019.01	9.10169	1070.66	14.63
Tail_15×15_5	946	1023.21	8.16196	1102.73	16.57
Tail_15×15_6	933	1010.85	8.34393	1047.81	12.31
Tail_15×15_7	891	971.593	9.04521	1046.32	17.43
Tail_15×15_8	893	967.635	8.3578	967.39	8.33
Tail_15×15_9	899	1009.21	12.2594	958.456	6.614
Tail_15×15_10	902	1010.7	12.0505	1040.85	15.39
Tail_20×20_1	1155	1267.75	9.76153	1370.99	18.7
Tail_20×20_2	1241	1386.03	11.6863	1424.07	14.75
Tail_20×20_3	1257	1424.63	13.3355	1406.3	11.88
Tail_20×20_4	1248	1319.4	5.72081	1423.62	14.07
Tail_20×20_5	1256	1397.88	11.2966	1485.69	18.29
Tail_20×20_6	1204	1281.91	6.47074	1355.98	12.62
Tail_20×20_7	1294	1442.75	11.4954	1542.63	19.21
Tail_20×20_8	1169	1293.97	10.6902	1281.93	9.661
Tail_20×20_9	1289	1380.41	7.09122	1485.19	15.22
Tail_20×20_10	1241	1341.23	8.07672	1386.62	11.73
Average RPD %			**9.218**		**12.79**

Table 9. Experiments in Fuzzy with Parallel Form ($m_{i\neq}$ 1)

Problem	Lower Bound	DEM		MEM		CPU Limit Time
		Cmax	RPD%	Cmax	RPD%	
5×5×2_1	119	127.979	7.54516	133.241	11.97	10
5×5×2_2	164	177.593	8.28822	187.071	14.07	10
5×5×2_3	123	135.179	9.90168	138.202	12.36	10
5×5×2_4	135	147.577	9.31617	149.58	10.8	10
5×5×2_5	225	246.379	9.50176	258.334	14.82	10
10×10×3_1	406	439.241	8.18741	457.75	12.75	40
10×10×3_2	385	417.308	8.39163	432.755	12.4	40
10×10×3_3	308	338.336	9.84942	351.4	14.09	40
10×10×3_4	220	241.075	9.57977	244.08	10.95	40
10×10×3_5	315	344.734	9.4393	357.659	13.54	40
15×15×4_1	376	413.1	9.86691	422.447	12.35	90
15×15×4_2	359	403.436	12.3777	412.047	14.78	90
15×15×4_3	408	452.204	10.8343	477.862	17.12	90
15×15×4_4	463	524.975	13.3854	527.437	13.92	90
15×15×4_5	425	467.509	10.002	479.794	12.89	90
20×20×5_1	625	682.046	9.12741	720.544	15.29	160
20×20×5_2	535	579.805	8.3747	585.372	9.415	160

Table 9. Continued

20×20×5_3	598	659.51	10.2859	657.675	9.979	160
20×20×5_4	559	605.285	8.27991	622.233	11.31	160
20×20×5_5	620	667.136	7.60256	685.706	10.6	160
25×25×5_1	900	967.28	7.47555	996.243	10.69	250
25×25×5_2	935	1011.99	8.23374	1096.86	17.31	250
25×25×5_3	1035	1107.55	7.00928	1144.27	10.56	250
25×25×5_4	795	870.295	9.47106	930.986	17.11	250
25×25×5_5	1072	1161.76	8.37359	1231.95	14.92	250
30×30×5_1	1359	1457.62	7.25683	1524.04	12.14	360
30×30×5_2	1109	1200.04	8.20957	1251.86	12.88	360
30×30×5_3	1273	1375.01	8.01362	1452.01	14.06	360
30×30×5_4	1236	1363.56	10.3205	1400.87	13.34	360
30×30×5_5	1085	1192.99	9.95301	1245.07	14.75	360
Average RPD %			**9.12073**		**13.05**	

Also we carry out an analysis of variance (ANOVA) test to investigate performance two algorithms. Table 10 shows the results of ANOVA. Since *p-value* <0.05 so can be said that there are significant difference between the two algorithms. Also by Fig.8 can be realized the efficiency of the algorithm DEM.

Table 10: ANOVA: Results versus Algorithms

Source	*df*	*SS*	*MS*	*F*	*P*-value
Algorithms	1	234.85	234.85	71.78	0.00
Error	58	189.75	3.27		
Total	59	424.60			

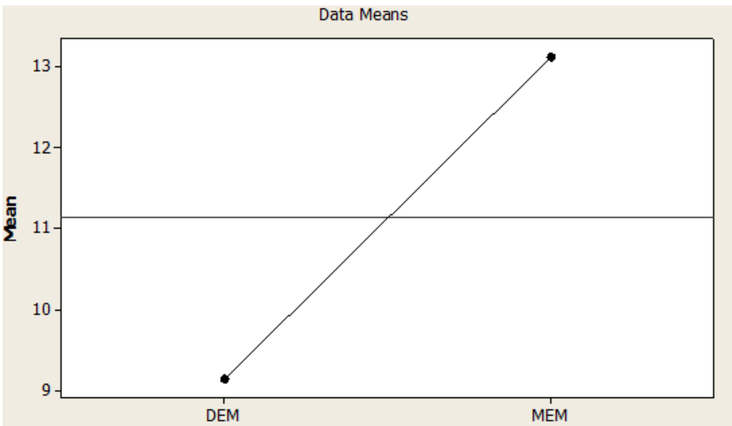

Figure 8: mean effect plot for algorithms

7. Conclusion and Future Research

In this paper, we presented a mixed-integer fuzzy programming (MIFP) approach for open shop scheduling problem with parallel machines at each stage to minimize makespan. Taking fuzzy assumption the desired problem was more practical and closer to the real world. We assumed that processing times is uncertainty and represented with triangular fuzzy number. Since it is known as NP-hard, to solve several medium to large-sized, we proposed a novel discrete electromagnetism-like algorithm (DEM). This algorithm made use of a decoding procedure using a permutation list and for calculating force and moving particle used crossover operators. The proposed DEM algorithm was tested on a set of benchmark problems from the literature under the circumstance m_i=1 and then $m_i \neq 1$. We applied the mean RPD for evaluating the performance of the DEM algorithm. Since the value mean RPD of the DEM under the terms of m_i=1 and $m_i \neq 1$ is 9.218 and 9.12073 respectively (As regards some of this difference is due to the fuzzy nature of the problem, because the lower bound is obtained in crisp condition), can find out that the DEM algorithm has good performance. Also we carried out an extensive comparison of the proposed DEM against MEM for same problem under a comprehensive benchmark of instances. The stopping criterion is set to a maximum elapsed CPU time for all the evaluated algorithms. After several statistical analyses, we can conclude that proposed method provides the best results for small instances and especially for large instances.

An interesting future research direction is to study the fuzzy open shop with non-identical parallel machine, and consider the problem studied here with the addition of some other assumption like no-wait or sequence dependent setup times, use our discrete EM to solve other scheduling problem and think over the multiple functions simultaneously.

References

[1] Low, Ch.Y.Y. (2009), Genetic algorithm based heuristics for an open shop scheduling problem with setup, processing, and removal times separated, *Robotics and Computer-Integrated Manufacturing,*Vol. 25, pp. 314-322.

[2] Pinedo, M. (2008), *Scheduling: Theory, Algorithms, and Systems (Third ed.),* Stern School of Business New York University: original edition published by Prentice Hall.

[3] Matta, M.E. (2009), A genetic algorithm for the proportionate multiprocessor open shop, *Computers & Operations Research*, Vol. 36, pp. 2601 -2618.

[4] Liu, C.Y. and Bulfin, R.L. (1987), Scheduling ordered open shops, *Comput. Oper. Res.* Vol. 14, pp. 257-264.

[5] Prins, C. (1994), An overview of scheduling problems arising in satellite communications, *Journal Operation Research Society,*Vol. 40, pp. 611-623.

[6] Gonzalez, T. (1976), Open shop scheduling to minimize finish time, *Journal of the ACM* , Vol. 4, pp. 665-679.

[7] Naderi, B., Fatemi Ghomi, S.M.T., nayeri, A. and Zandieh. M. (2010), A study on open shop scheduling to minimise total tardiness, *International Journal of Production Research*, pp. 1-22.

[8] Liaw, C.F. (2005), Scheduling preemptive open shop to minimize total tardiness, *European Journal of Operational Research*, Vol. 162, pp. 173-183.

[9] Mosheiov, G. and Oron, D. (2008), Open shop batch scheduling with identical jobs, *European Journal of Operational Research*, Vol. 187, pp. 1282-1292.

[10] Su, L.H., Chou, F.D. and Ting, W.CH. (2005), Minimizing makespan in a two stage system with flow shop an open shop, *Computers and Industrial Engineering,* Vol. 49, pp. 520-536.

[11] Naderi, B., Ghomi, S.M.T., nayeri, A. and Zandieh, M. (2011), Scheduling open shops with parallel machines to minimize total completion time, *Journal of Computational and Applied Mathematics*, pp. 1275-1287.

[12] Panahi, H. and Tavakkoli-Moghadam, R. (2011), Solving a multi-objective open shop scheduling problem by a novel hybrid ant colony optimization, *Expert Systems with Applications*, Vol. 38, pp. 2817-2822.

[13] Sha, D.Y., Lin, H.H. and Hsu, C.Y. (2010), A Modified Particle Swarm Optimization for Multi-objective Open Shop Scheduling, *Proceeding of the International MultiConference of Engineers and Comuter Scientists*, Vol. 3, Hong Kong.

[14] Birbil, S.I. and Fang, SH.Ch. (2003), An electromagnetism-like mechanism for global optimization, *Journal of Global Optimization*, Vol. 25, pp. 263-282.

[15] Yurtkuran, A. and Emel, E. (2010), A new hybrid electromagnetism-like algorithm for capacitated vehicle routing problems, *Expert Syst. Appl.* Vol.37, pp. 3427–3433.

[16] Wu, P., Yang, K. and Fang, H. (2006), A revised EM-like algorithm + K-OPT method for solving the traveling salesman problem, *Proceedings of the first international conference on innovative computing, information and control.*

[17] Wu, P., Yang, W.H. and Wei, N.Ch. (2004), AN ELECTROMAGNETISM ALGORITHM OF NEURAL NETWORK ANALYSIS—AN APPLICATION TO TEXTILE RETAIL OPERATION, *Journal of the Chinese Institute of Industrial Engineers*, Vol. 21, pp. 59-67.

[18] Tsou, C.S. and Kao, C.H. (2008), Multi-objective inventory control using electromagnetism-like meta-heuristic, *International Journal of Production Research,* Vol. 1, pp. 1-16.

[19] Maenhout, B. and Vanhoucke, M. (2007), An electromagnetic meta-heuristic for the nurse scheduling problem, *J. Heur., Springer, Netherlands*, pp. 359-385.

[20] Debels, D. and Vanhoucke, M. (2004), An electromagnetism meta-heuristic for the resource-constrained project scheduling problem, *Published in Lecture notes on Computer Science*, Vol. 3871, pp. 259-270.

[21] Chang, P.C., Chen, Sh. and Fan, C.Y. (2009), A hybrid electromagnetism-like algorithm for single machine scheduling problem, *Expert Systems with Applications,* Vol. 36, pp. 1259-1267.

[22] Naderi, B., Ghomi, S.M.T., nayeri, A. and Zandieh, M. (2010), Modeling and scheduling open shops with sequence-dependent setup times to minimize total completion time, *Int. J. Adv. Manuf. Technol.*

[23] Noori-Darvisha, S., Mahdavi, I. and Mahdavi-Amiri, N. (2012), A bi-objective possibilistic programming model for open shop scheduling problems with sequence-dependent setup times, fuzzy processing times, and fuzzy due dates, *Applied Soft Computing,*Vol. 12, pp. 1399-1416.

[24] Hashemi Doulabi, M.A. (2011), An effective hybrid simulated annealing and two mixed integer linear formulations for just-in-time open shop scheduling problem, *Int. J. Adv. Manuf. Technol*.

[25] Yu,V.F., Lin, S.W. and Ch, Sh.Y. (2010), The museum visitor routing problem, *Applied Mathematics and Computation*, Vol. 216, pp. 719-29.

[26] Roshanaei, Seyed Esfehani, M.M. and Zandie, M. (2010), Integrating non-preemptive open shops scheduling with sequence-dependent setup times using advanced metaheuristics, *Expert Systems with Applications*, Vol. 37, pp. 259-266.

[27] Palacios, J.J., Puentel, J., Camino, R. and Gonzalez-R, I. (2009), A Genetic Algorithm for the Open Shop Problem with Uncertain Durations, *Springer-Verlag Berlin Heidelberg*, pp. 255-264.

[28] Low, C. and Yeh, Y. (2009), Genetic algorithm-based heuristics for an open shop scheduling problem with setup, processing, and removal times separated, *Robotics and Computer-Integrated Manufacturing,* Vol. 25, pp. 314–322.

[29] Seraj, O. and Tavakkoli-Moghadam, R. (2009), A tabu search method for a new bi-objective open shop scheduling problem by a fuzzy multi objective decision making approach, *International Journal of Engineering,* Vol. 22, pp. 1-14.

[30] Zhang, R. and Wu, Ch. (2008), A Simulated Annealing Algorithm Based on Bottleneck Jobs for the Open Shop Scheduling Problem, *Proceedings of the 7th World Congress on Intelligent Control and Automation, IEEE.*

[31] Sha, D.Y. and Hsu, C.Y. (2008), A new particle swarm optimization for the open shop scheduling problem, *Computers and Operations Research,* Vol. 35, pp. 3243-3261.

[32] Huang, and Lin. (2007), Concurrent Open shop Problem to Minimize the Weighted Number of Late Jobs, *Book Chapter: Multiprocessor Scheduling: Theory and Applications.*

[33] Senthilkumar, P. and Shahabudeen, P. (2006), based heuristic for the open job shop scheduling problem, *Int. J. Adv. Manuf. Technol.* Vol. 30, pp. 297-301.

[34] Blum, CH. and Michael, S. (2004), Ant Colony Optimization Algorithm for Shop Scheduling Problems, *Journal of Mathematical Modelling and Algorithms,* Vol. 3, pp. 285-308.

[35] Liaw, Ch.F. (2003), An effcient tabu search approach for the two-machine preemptive open shop scheduling problem, *Computers & Operations Research,* Vol. 30, pp. 2081-2095.

[36] Blum, Ch. (2003), Beam-ACO—hybridizing ant colony optimization with beam search: an application to open shop scheduling, *Computers & Operations Research.*

[37] Liaw, Ch.F. (1999), A tabu search algorithm for the open shop scheduling problem, *Computers & Operations Research,* Vol. 26, pp. 109-126.

[38] Peng, J. and Liu, B. (2004), Parallel machine scheduling models with fuzzy processing times, Information Sciences-Informatics and Computer Science, *An International Journal,* Vol. 166, pp. 49-66.

[39] Balin, S. (2011), Parallel machine scheduling with fuzzy processing times using a robust genetic algorithm and simulation, *Information Sciences,* Vol. 181, pp. 3551-3569.

[40] Mok, P.Y., Kwong, C.K. and Wong, W.K. (2007), Optimization of fault-tolerant fabric-cutting schedules using genetic algorithms and fuzzy set theory, *European Journal of Operations Research,* Vol. 177, pp. 1876-1893.

[41] Zadeh, L.A. (1987), Fuzzy set as a basis for a theory of possibility, *Fuzzy Sets and Systems,* Vol. 1, pp. 3-28.

[42] Dubois, D. and Prade, H. (1998), *Possibility Theory: An Approach to Computerized Processing of Uncertainty*, Plenum Press, New York

[43] Anglani, A., Grieco, A., Guerriero, E. and Musmanno, R. (2005), Robust scheduling of parallel machines with sequence-dependent set-up costs, *European Journal of Operational Research,* Vol. 161, pp. 704-720.

[44] Balasubramanian, J. and Grossmann, I.E. (2003), Scheduling optimization under uncertainty – an alternative approach, *Computers and Chemical Engineering,* Vol. 27, pp. 469-490.

[45] Slowinski, R. and Hapke, M. (2000), *Scheduling Under Fuzziness*, Physical-Verlag Editions, New York.

[46] Konno, T. and Ishii, H. (2000), An open shop scheduling problem with fuzzy allowable time and fuzzy resource constraint, *Fuzzy Sets and Systems*, Vol. 109, pp. 141-147.

[47] Torabi, S.A. and Hassini, E. (2008), An interactive possibilistic programming approach for multiple objective supply chain master planning, *Fuzzy Sets and Systems,* Vol. 159, pp. 193 - 214.

[48] Taillard, E. (1993), Benchmarks for basic scheduling problems, *European Journal of Operational Research*, Vol. 64, pp. 278-285.

[49] Itoh, T. and Ishii, H. (1999), Fuzzy due-date scheduling problem with fuzzy processing time, *Int. Trans. Op. Res*. Vol. *6*, pp. 639-647.

[50] Adamopoulos, G.I., Pappis, C.P. and Karacapilidi, N.I. (1999), Job sequencing with uncertain and controllable processing times, *Int. Trans. Op. Res,* Vol. 6, pp. 483-493.

[51] Yimer, A.D. and Demirli, K. (2009), Fuzzy scheduling of job orders in a two-stage flow shop with batch-processing machines, *International Journal of Approximate Reasoning*, Vol. 50, pp. 117-137.

[52] Baker, K.R. and Trietsch, D. (2009), *Principles of sequencing and Scheduling*, Published by John Wiley & Sons, Inc, Hoboken, New Jersey, Published simultaneously in Canada.

[53] Yimer, A.D. and Demirli, K. (2006), Minimizing weighted flow time in a two-stage flow shop with fuzzy setup and processing times, *IEEE Conference Proceeding, in: North America Fuzzy Information Processing Society*, pp. 708-713.

[54] Bellman, R.E. and Zadeh, L.A. (1970), Decision making in a fuzzy environment, *Manage. Sci*. Vol. 17, pp. 141–164.

[55] Zimmermann, H.J. (1978), Fuzzy programming and linear programming with several objective functions, *Fuzzy Sets and Syst.*, Vol. 1, pp. 45-56.

[56] Giffler, J. and Thompson, G.L. (1960), Algorithms for solving production scheduling problems, *Operations Research*, Vol. 8, pp. 487-503.

[57] Debels, D., Reyck, B.D., Leus, R. and Vanhoucke, M. (2006), A hybrid scatters search/electromagnetism meta-heuristic for project scheduling, *European Journal of Operational Research*, Vol. 169, pp. 638–653.

[58] Falkenauer, E. and Bouffouix, S. (1991), A genetic algorithm for job shop, in*: Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 824-829.

[59] Bierwirth, C., Mattfeld, D. and Kopfer, H. (1996), On permutation representations for scheduling problems, in: H.M. Voigt (Ed.), *Proceedings of Parallel Problem Solving from Nature IV, Springer, Berlin*, pp. 310-318.

[60] Phadke, M.S. (1989), *Quality Engineering Using Robust Design*, Prentice-Hall, Upper Saddle River.

[61] Al-Aomar, R. (2006), Incorporating robustness into genetic algorithm search of stochastic simulation outputs, *Simulation Modeling Practice and Theory*. Vol. 14, pp. 201-223.

[62] Taguchi, G. (1986), Introduction to quality engineering, White Plains: Asian Productivity Organization/UNIPUB.

[63] Fortemps, P. (1997), Job shop scheduling with imprecise durations: a fuzzy approach, *IEEE Transactions of Fuzzy Systems*, Vol. 7, pp. 557-569.

[64] Chang, Pei-Ch., Chen, Shi-Hs. and Fan, Ch-Yu. (2009), A hybrid electromagnetism-like algorithm for single machine scheduling problem, *Expert Systems with Applications*, Vol. 36, pp. 1259-1267.