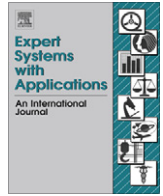




Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A hybridization of simulated annealing and electromagnetism-like mechanism for a periodic job shop scheduling problem

A. Jamili^{a,*}, M.A. Shafia^a, R. Tavakkoli-Moghaddam^b

^a Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

^b Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

ARTICLE INFO

Keywords:

Periodic job shop scheduling
Periodic event scheduling problem
Simulated annealing
Electromagnetism-like mechanism

ABSTRACT

In this paper, we present on a periodic job shop scheduling problem (PJSSP) based on the periodic event scheduling problem (PESP), which is different from cyclic scheduling. The PESP schedules a number of recurring events, such that each pair of events fulfills certain constraints during a given time period. To solve such a hard PJSS problem, we propose a hybrid algorithm, namely EM-SA, which is based on electromagnetism-like mechanism (EM) and simulated annealing (SA). To evaluate this proposed EM-SA, some randomly constructed instances are solved, and the related results are compared with SA and optimum solutions obtained by the branch-and-bound (B&B) algorithm. The results demonstrate the efficiency and effectiveness of the proposed hybrid EM-SA algorithm to solve the PJSSP.

© 2010 Published by Elsevier Ltd.

1. Introduction

The job shop scheduling problem (JSSP) is known to be one of the most interesting combinatorial optimization problem that has been the subject of many research studies during the last three decades. A periodic job shop consists of a set of n jobs that should be processed on a set of m machines each job consists of a predetermined sequence of task operations where each of which requires to be processed without preemption on a single machine that can handle at most one job at a time. A periodic schedule is an assignment of operations to time slots with the maximum length of T , on corresponding machines. The objective is to find the sequence of all the jobs in such a way that total (weighted) tardiness is minimized. For instance, Naderi, Khalili, and Tavakkoli-Moghaddam (2009a) solved an extended JSSP with a specific case of machine availability constraints by the artificial immune algorithm (AIA) hybridized with a SA algorithm. They considered sequence-dependent setup times and preventive maintenance (PM) operations minimizing the total completion time. Sha and Lin (2009) constructed a particle swarm optimization (PSO) for a multi-objective JSSP, called MOPSO. The objectives include the minimization of makespan, total tardiness, and total machine idle times. Pan and Huang (2009) proposed a hybrid genetic algorithm to solve the no-wait job shop problems based on an asymmetric traveling salesman problem formulation followed by a local search that explore a better solution region. It is worthy to note that the JSSP belongs to NP-complete problems (Garey, Johnson, & Sethi, 1976).

In this paper, we study the periodic JSSP that is formulated based on the periodic event scheduling problem (PESP) introduced by (Serafini & Ukovich, 1989). Before introducing the PESP, the cyclic scheduling (CS) problem, which is different from the PJSSP, is shortly reviewed. In this problem, a set of activities is to be repeated an indefinite number of times, where the primary objective function is to minimize the period length. CS problems arise in domains that the sequence of jobs are repeated, such as synthesis of digital signal processing, course scheduling, railway timetabling, robotics, automated manufacturing systems, time-sharing of processors in embedded systems and in compilers for scheduling loop operations for parallel or pipelined architectures. The primary objective is to minimize the period length. The CS problem is generally studied in two different ways. The first one does not address resource constraints. The basic cyclic scheduling (BCS) problem falls in this category. Brucker and Kampmeyer (2008) studied a general basic cyclic scheduling problem (GBCS) where the objective is to minimize the cycle time. In contrast with the first category, the cyclic job shop scheduling problem (CJSSP) contains the resource constraints. Kimbrel and Sviridenko (2008) introduced a bi-objective CJSSP, in which the jobs are all the same, and the schedule is processed in a cyclic mode, where the objectives are minimizing the cycle time and the flow time. Cavory, Dupas, and Goncalves (2005) studied the CJSSP with linear precedence constraints and presented a general approach based on the coupling of a genetic algorithm and a scheduler that utilizes a Petri-net modeling the linear precedence constraints between cyclic tasks. Song and Lee (1996) presented a tabu search procedure for the periodic JSSP. Song and Lee (1998) also investigated the scheduling problem for a general CJSSP with blocking where each machine has

* Corresponding author. Tel.: +98 21 22830153.

E-mail address: a_jamili@iust.ac.ir (A. Jamili).

an input buffer with the finite capacity. Moreover, they proposed a MIP model to find an optimal sequence and schedule that is deadlock-free and has the minimum cycle time. On the other hand, the PESP approach is used where the period length is constant, and the objective is to minimize the completion time of jobs. This approach has a special application in constructing different types of timetables.

This paper is organized as follows. Section 2 deals with presenting a summary of the PESP and the PJSSP. Section 3 presents the proposed electromagnetism-like mechanism (EM) and simulated annealing (SA) algorithms, as well as the hybridization of these algorithms. The computational results are presented in Section 4. The conclusion remarks are given at the end to summarize the contribution of this paper.

2. Periodic job shop scheduling problem (PJSSP)

The PJSSP discussed here is based on the periodic event scheduling problem (PESP) firstly introduced by Serafini and Ukovich (1989) and is defined as the problem of scheduling some periodically repeating events subject to certain constraints. Serafini and Ukovich (1989) showed that the PESP is NP-complete. Furthermore, Liebchen and Peeters (2001), Odijk (1994), Nachtigall (1996) are also investigated the complexity of the PESP. Given a time period T , a set of events V , and a set of periodic interval constraints C , every constraint (e.g., a) is associated with a pair of events (i, j) as well as a lower bound l_a and an upper bound u_a .

It is intended to find a schedule through encoding the set of periodic interval constraints imposed on the timetable in an event-activity graph $D = (V, C)$ with node set V and arc set C .

By the above definitions, in PESP each pair of events must satisfy the corresponding constraints. In other words, a solution of a PESP instance is a node assignment $\pi: V \rightarrow [0, T)$ that satisfies Eq. (1) by

$$(\pi_j - \pi_i - l_a) \bmod T \leq u_a - l_a, \quad \forall a \in C \tag{1}$$

Where mod stands for the modulo operator, and π_i is the occurrence time of event i correspondence with the completion time of each operation in a PJSSP environment. It is worth noting that one can scale an instance such that $0 \leq l_a < T$, and $u_a - l_a < T$. In order to be able to use, with $d_a < T$. To apply integer programming techniques, the following reformulation of inequality (1) is used.

$$l_a \leq \pi_j - \pi_i - T \times z_a \leq u_a \tag{2}$$

In other words, by the assumption of $z_a \in Z$ inequality 2 can be replaced with inequality 1. The newly defined variable z_a is called periodical offset of the activity a . To further illustrate the issue, consider an example of a PESP, with three events, four constraints and time period $T = 60$, and the constraint graph shown in Fig. 1. The problem is to find π_i and $z_i, \forall i = 0, 1, 2$, where the following first two constraints are satisfied and one of the two last constraints are fulfilled.

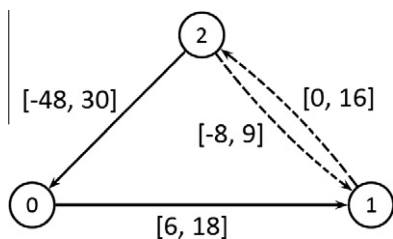


Fig. 1. A small PESP instance.

$$6 \leq \pi_1 - \pi_0 - 60z_0 \leq 18, \text{ and } -8 \leq \pi_1 - \pi_2 - 60z_1 \leq 9, \text{ and } (-48 \leq \pi_0 - \pi_2 - 60z_2 \leq 30 \text{ or } 0 \leq \pi_2 - \pi_1 - 60z_1 \leq 16)$$

For more details, the interested reader may refer to Kinder (2008), Liebchen and Peeters (2001), Odijk (1994), Nachtigall (1996) and Serafini and Ukovich (1989) showed that the PESP is NP-complete for fixed $T \geq 3$.

Operations in the PJSSP have the role of events in the PESP. Therefore, to reach the formulation of the PJSSP, the constraints involved with the scheduling of operations must be updated based on the introduced PESP approach. Following notations are employed in the mathematical model of the PJSSP.

- M set of machines
- J set of jobs
- w_i the weight (priority) of job i
- d_i due date of job i
- r_i release date of job i
- o_i the first machine of the route correspondence with job i
- e_i the last machine of the route correspondence with job i
- P_{ijm} processing time of the j th operation of job i that is on machine m
- T time period length
- x_{ik} A positive variable indicates the completion time of job i on machine k

The mathematical formulation of the PJSSP, called Model P1, is as follows:

$$\min \sum_{i \in P} w_i \times \max\{(x_{ie_i} - d), 0\}$$

$$p_{ivk} \leq x_{ik} - x_{ih} \leq T - 1, (i, v - 1, h) \ll (i, v, k), \forall i \in J, \forall k, h \in M \tag{3}$$

$$p_{ivk} \leq x_{ik} - x_{jk} - z_{ijk} \times T \leq T - p_{jv'k}, \forall i, j \in J, \forall k \in M \tag{4}$$

$$x_{io_1} - p_{io_1} \geq r_i, \quad \forall i \in J \tag{5}$$

$$d_i = 0 \text{ and } w_i = 1$$

where (i, v, k) represents the v th operation of job i which must be processed on machine k , and $z_{ijk} \in Z (\forall i, j \in J \text{ and } \forall k \in M)$.

The objective function of the above model (P1) is to minimize the total weighed tardiness, and in a special case, where for all jobs $d_i = 0$, and $w_i = 1$ can be interpreted as the minimization of completion time of jobs. Constraint 3 represents the precedence of each job, where the v th operation of job i is performed on machine k . Periodic interval constraint 4 ensures that no two operations are processed simultaneously by the same machine constraint 5 specifies that the first operation related to each job must be started after the associated release date. Note that operation u corresponds with the process of job i on machine o_i .

3. Proposed EM-SA algorithm

To solve the defined PJSSP, and specially large-scale problems, in a rational amount of time, this section represents the proposed hybrid algorithm consisting of electromagnetism-like mechanism

Table 1
Use of random keys as the encoding scheme.

Operations	1	1	2	2	3	3
Assigned Random key	0.4532	0.6728	0.9185	0.1319	0.8861	0.2099
Relative order operations	3	4	6	1	5	2

(EM) and simulated annealing (SA) algorithms for the PJSSP. To that end, it is intended to present an encoding scheme in order to be applied as a useful method in order to generate schedules. Then, SA and EM algorithms are separately reviewed and adjusted for the PJSSP introduced in the Section 2. Finally, the hybridization procedure of these algorithms is discussed to present the EM-SA algorithm.

3.1. Encoding scheme

An appropriate representation of the candidate schedules plays a key role in the effectiveness of any algorithm. Using random keys (RKs) as the encoding scheme is the most frequently used, and is easy to adjust to the EM and SA algorithms (Chang, Chen, & Fan, 2009; Naderi, Tavakkoli-Moghaddam, & Khalili, 2009b; and Tavakkoli-Moghaddam, Khalili, & Naderi, 2009; Lin & et al., 2009).

The searching space of the PJSSP contains permutation of n jobs on m machines, and therefore each schedule is represented with the relative order of $n \times m$ operations. In this paper, the permutation of jobs is shown through random keys. To that end, a random number between 0 and 1, called random key, is assigned to each operation and these random keys show the relative order of the jobs. To illustrate the issue, consider a problem with three jobs each of which must be performed on two machines. Each job consists of two operations and therefore six operations can be defined: {1 1 2 2 3 3}. In the next step in order to define the permutation of operations, for each of them, a random number is randomly generated from a uniform distribution between 0 and 1. These RKs are then sorted to find the relative order of operations, as shown in Table 1.

3.2. Simulated annealing (SA) algorithm for PJSSP

Simulated annealing (SA) is one of the most popular meta-heuristics algorithms in scheduling optimization research area that has its origin in the fields of material science and physics. In annealing, a metal is first heated to a high temperature and then cooled with a slow rate to the room temperature. In SA to escape from local optima, even moves resulting in solutions of the worse quality than the current solution may be accepted. At the first steps of the procedure, SA is started at a high temperature ($Temp_0$), so most of the moves are accepted, on the opposite side, when the temperature, or probability of accepting a worse solution, is nearly zero, this simply seeks the bottom of the local optima. The chance of finding a good solution can be traded off with the computational time by slowing down the defined cooling factor procedure.

The employed notations of the SA algorithm are as follows.

k	the iteration counter
$Temp_k$	the temperature in the k th iteration
S_k	the k th schedule
S'	the neighborhood schedule of s
$Temp_0$	the initial temperature in the first iteration
S_{best}	the best found solution
$TWT(s)$	the total weighted tardiness value of schedule s
α	the cooling factor

The general outline of the SA algorithm for the PJSSP is summarized as follows.

- Step 1. *Initial conditions:* Set $Temp_0$ and α based on the desired running time and Section 3.2.4. Let $k \leftarrow 0$.
- Step 2. *Generating the initial solution:* Select an initial solution, s based on the desired running time and the Section 3.2.4. Let $S_{best} \leftarrow s$.

- Step 3. *Checking the infeasibility:* If s is infeasible, go to Step 2; otherwise, go to Step 4. The infeasibility checking is based on Section 3.2.1.
- Step 4. *Generating the neighborhood solution:* Generate a neighborhood solution, s' , using schedule s based on Section 3.2.3.
- Step 5. *Checking the infeasibility:* If s' is infeasible go to Step 4; otherwise, go to Step 6. The infeasibility checking is based on the Section 3.2.1.
- Step 6. *Checking the acceptance condition:* $TWT(s') \leq TWT(s)$ or random $[0, 1] < \exp\{(-TWT(s)-TWT(s'))/Temp_k\}$ then $s \leftarrow s'$. Note that random $[0, 1]$ is a function that generates random values between 0 and 1. If $TWT(s') \leq TWT(S_{best})$ then $S_{best} \leftarrow s$.
- Step 7. that the sequence of jobs are repeated, such as synthesis of digital signal processing, course scheduling, railway timetabling, robotics, If the termination criterion, in this paper is to perform a fixed number of iteration, is not satisfied then $Temp_k = \alpha \times Temp_{k-1}$, $k \leftarrow k + 1$ and go to Step 4; pre-emption on a single machine that can handle at most one job at a time. A periodic and return S_{best} .

3.2.1. Characteristics of a feasible solution

In non-periodic JSSP (i.e., classic JSSP), schedules are generated based on the orders that are given to the operations that always leads to a feasible solution. However, as specified in proposition 1, this method may result in infeasibility in the PJSSP.

Proposition 1: *There is no feasible solution for the instances, in which the following condition is held:*

Condition: $\exists k \in M$, such that $\sum_{i \in J} p_{ijk} > T$, where the j th operation of job i should be performed on machine k .

Proof: The result immediately follows from the definition of the PJSSP, which schedules some repeating operations in a time interval with period length T_0 .

In general, a feasible schedule is the one that for each machine, all assigned operations must be scheduled in a time interval less than the pre-specified period length. To clarify the issue, consider Fig. 2 which depicts two time slot allocation scheme. It is intended to schedule six operations in the period length of T in a machine. As shown in this figure, the first time allocation is infeasible, because there is no interval to locate job 6 in the period length, while the second one is feasible, where all six operations are scheduled in a time interval with period length of T .

3.2.2. Method of Initial solution generation

As explained in Section 3.2.1, finding an initial feasible schedule may not be so easy. For this reason, we experimentally observe that following algorithm outperforms others.

- 1. *FLFS rule:* Find an initial schedule based on the first leave first served (FLFS) rule. In this procedure when two or more jobs

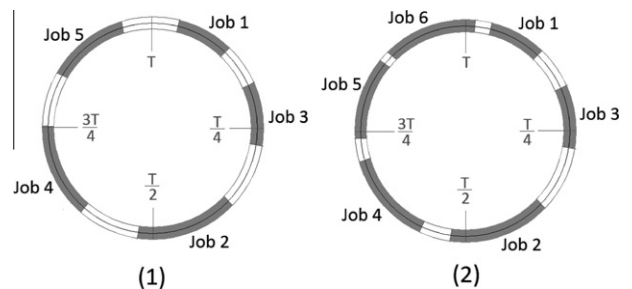


Fig. 2. Infeasibility of PJSSP.

- compete for the same machine, the precedence is given to the one which leaves the machine first.
2. *Feasibility checking #1*: Check the feasibility of the schedule. If the schedule is feasible terminate the algorithm; otherwise, go to Step 3.
 3. *FIFO rule*: Find an initial schedule based on first in first out (FIFO) rule. In this procedure, when two or more jobs compete for the same machine, the precedence is given to the one which waits more.
 4. *Feasibility checking #2*: Check the feasibility of the schedule. If the schedule is feasible terminate the algorithm; otherwise, go to Step 5.
 5. *RKs method*: Find an initial schedule based on generating random keys. Go to Step 4.

3.2.3. Method of Neighborhood solution generation

The following algorithm known as single point operator is utilized in order to generate a neighborhood candidate:

1. *Job selection*: Select one of the jobs randomly.
2. *RKs alteration*: Regenerate the random key associated with the selected job.
3. *Neighborhood checking*: If the order of operations is changed, terminate the algorithm; otherwise, go to Step 1.

3.2.4. SA parameter tuning

To improve the performance of the introduced SA algorithm, parameters T_0 and α , are tuned in this subsection. To that end, as shown in Table 2, for each parameter, three levels are tested, and therefore, 3^2 designs are applied. Moreover, 15 different instances are randomly generated and solved by assuming each of $3^2 = 9$ different combinations of (T_0, α) . The termination condition is to perform the algorithm by 3000 iterations.

To compare the instances, the relative deviation index (RDI) is used for the TWT as a common performance measure. This index is obtained by:

$$RDI_k^l = \frac{F_k^l - \text{Min}_k}{\text{Max}_k - \text{Min}_k} \times 100 \tag{6}$$

where F_k^l is the TWT value of the k th instance by using l th combination of parameters. Min_k and Max_k are the best and worst TWT values obtained for each instance.

Fifteen instances are randomly generated in different combinations of the number of jobs and the number of machines. Each instance is solved considering one of the combinations of T_0 and α . Therefore $(15 \times 3 \times 3=)$, 135 instances are totally solved. The related results are analyzed by the means of the analysis of variance (ANOVA) technique. The normality and homogeneity of variance and independence of residuals do not show any particular pattern in experiments. Fig. 3 depicts the interaction plot for parameters T_0 and α . It is concluded that the combination $T_0 = 150$ and $\alpha = 0.97$ results in statistically better output than other evaluated combinations.

3.3. Electromagnetism-like mechanism (EM)

Electromagnetism-like mechanism (EM) is a population based meta-heuristic that has been proposed to solve continuous prob-

Table 2
Three levels of SA parameters.

	Level 1	Level 2	Level 3
T_0	100	200	300
α	0.9	0.94	0.98

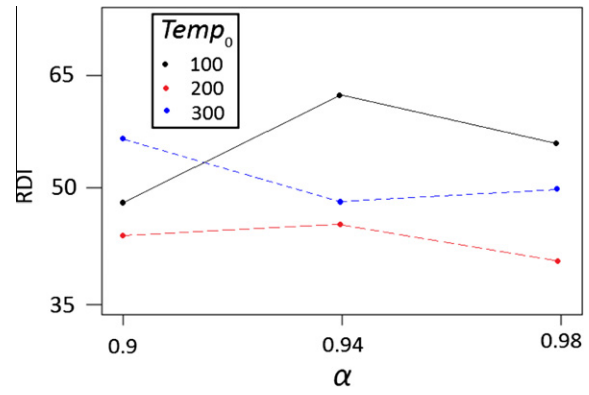


Fig. 3. Interaction between SA parameters.

lems effectively. Gol-Alikhani, Javadian, and Tavakkoli-Moghadam (2009) presented a novel hybrid approach based on EM embedded with a well-known local search, called Solis and Wets, for continuous optimization problems. They compared related results with two algorithms known as the original and revised EM. In general, EM simulates the attraction–repulsion mechanism of electromagnetism theory that is based on Coulomb’s law. Each particle represents a solution, and the charge of each particle relates to its objective function value (OFV). The better OFV of the particle, the higher charge the particle has. To compute the force between two points, a charge, such as value q_i , is assigned to each point. The charge of the point is calculated according to the relative efficiency of the objective function values in the current population, as given in Eq. (7).

$$q_i = \exp \left(-n \times \frac{f(x_i) - f(x_{best})}{\sum_{j=1}^{popsize} (f(x_j) - f(x_{best}))} \right), \tag{7}$$

$\forall i = 1, 2, \dots, popsize$

where q_i is the charge of particle i . In addition, $f(x_i)$, $f(x_{best})$, and $f(x_j)$ denote the objective values of particle i , the best solution, and particle j , respectively. Finally, $popsize$ is the population size. In this way, the points that have better objective function values possess higher charges. Note that, unlike electrical charges, no signs are attached to the charge of an individual point in the Eq. (1); instead, the direction of a particular force between two points is determined after comparing their objective function values. The total force, F_i , exerted on candidate solution i is also calculated by:

$$F_i = \begin{cases} \sum_{j \neq i}^{popsize} (x_j - x_i) \frac{q_j \times q_i}{\|x_j - x_i\|^2}; & f(x_j) < f(x_i) \\ \sum_{j \neq i}^{popsize} (x_j - x_i) \frac{q_j \times q_i}{\|x_j - x_i\|^2}; & f(x_j) \geq f(x_i) \end{cases} \tag{8}$$

A two-dimensional example total force vector, F_i , exerted on candidate solutions is shown in Fig. 4. The force exerted by x_1 on x_3 is F_{13} (repulsion: the objective function of x_1 is worse than that of x_3), and the force exerted by x_2 on x_3 is F_{23} (attraction: the objective function of x_2 is better than that of x_3). F_3 is the total force exerted on x_3 by x_1 and x_2 .

The next procedure is to move the candidate solutions based on the total force calculated by Eq. (8). All the candidate solutions are moved with the exception of the current best solution. The move for each candidate solution is in direction of total force exerted on it by a random step length. This length is generated from uniform distribution between (0, 1), see Eq. (9). We can guarantee that candidate solutions have a nonzero probability to move to the unvisited solution along this direction by selecting random length.

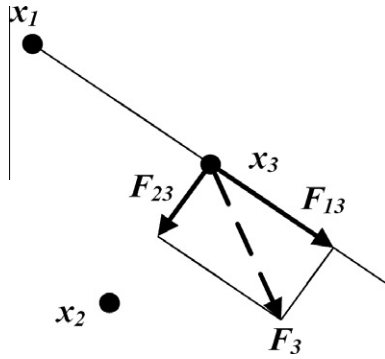


Fig. 4. Example of exertion of forces.

$$\begin{cases} x_{ik} = x_{ik} + \text{Random}(0, 1) \times F_{ik}(1 - x_{ik}); & F_{ik} > 0 \\ x_{ik} = x_{ik} + \text{Random}(0, 1) \times F_{ik}(x_{ik}); & F_{ik} \leq 0 \end{cases} \quad (9)$$

The proposed EM algorithm is applied for the given PJSSP.

- Step 1. *Generating initial schedules*: Generate *popsiz*e initial schedules: $s_1^r, r=1, \dots, \text{popsiz}$ e. If each of the *popsiz*e generated schedules is infeasible, re-generate it until all the schedules are feasible. Find the TWT value for each particle. Set $P_1^r \leftarrow s_1^r$.
- Step 2. *Schedule updating*: Calculate q_r and F_r for each schedule. Update the schedules based on Eq. (9).
- Step 3. *Infeasibility checking*: If each of the updated schedules is infeasible set $s_1^r \leftarrow P_1^r$. If the OFV of each of the schedules improved let $s_1^r \leftarrow P_1^r$. Update P_g .
- Step 4. *Termination condition*: If the termination criterion is not met, go to Step 2; otherwise, return P_g as the best found schedule.

where P_1^r is the best local solution that the *r*th particle has achieved. P_g is the best solution obtained in the whole population. The procedure to find initial and neighborhood solutions are similar to the SA algorithm. The parameters of the proposed EM are tuned in the following subsection.

The population size of the proposed EM is only required to be tuned. Similar to the proposed SA, a full factorial design in the DOE approach is applied. We consider three levels (e.g., 5, 10 and 20) for this parameter. To that end, 15 different instances are randomly generated. The stopping criterion is to perform $\frac{10000}{\text{popsiz}}$ iterations.

The RDI also used for the TWT as the performance measure to compare the instances. The means plot and least significant differences (LSD) intervals (at the 95% confidence level) for the levels of the *popsiz*e parameter are shown in Fig. 5. It is concluded that the *popsiz*e, which is equal to 15, results the best among other investigated levels.

3.4. Hybrid EM-SA algorithm

The idea of the proposed hybrid algorithm, called EM-SA, based on SA and EM algorithms for job shop problems has been widely exploited in the literature (Naderi et al., 2009b; Zhang, Shao, Li, & Gao, 2009; and Tavakkoli-Moghaddam et al., 2009). EM is a population-based meta-heuristic that is to solve continuous problems effectively. Moreover, SA is a meta-heuristic that is designed for finding a near-optimal solution of combinatorial optimization problems. Therefore, the EM and SA algorithms are combined which can omit the concrete velocity-displacement updating method in the traditional EM for the PJSS problem. On the other hand, the solved initial instances demonstrate that the performance of EM strongly depends on the initial solution. To that

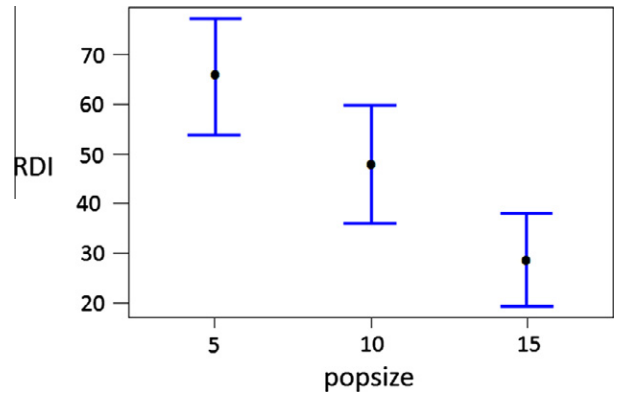


Fig. 5. Means plot and LSD intervals for popsize.

end, the final SA solution is used as one of the initial solutions of the EM algorithm.

The proposed EM-SA hybrid algorithm includes two phases. In the first phase, SA obtains a good initial solution. In the second phase further to the final solution of the SA algorithm, the other remained initial solutions are randomly generated. Then, the EM algorithm combined with the SA algorithm is run. The main steps of the proposed hybrid algorithm for the PJSSP are summarized as follows.

- Step 1. *Generating the initial schedules*: $k \leftarrow 1$. Run the SA algorithm, and consider the found solution as the first initial schedule. Generate *popsiz*e-1 initial random schedules. If any of the generated schedules is infeasible, reconstruct them until all the initial solutions are feasible. Set $P_1^r \leftarrow s_k^r$ and update P_g .
- Step 2. *Schedule updating #1*: For each of the particles, run the SA algorithm. If each of the updated schedules is infeasible, set $s_k^r \leftarrow P_1^r$. Furthermore, in the case of improvement in the TWT value of each of the schedules, let $P_1^r \leftarrow s_k^r$. Update P_g .
- Step 3. *Schedule updating #2*: Update the particles by exerting the repulsion and attraction forces of the particles to each other using Eqs. (7)–(9).
- Step 4. *Infeasibility checking*: In the case of infeasibility of the updated schedules set $s_k^r \leftarrow P_1^r$.
- Step 5. *Termination condition*: Find the objective value for each particle. Update P_1^r and If termination criterion is not satisfied, then $k \leftarrow k + 1$ and go to Step 2; otherwise, return the P_g as the best found schedule.

The general outline of the proposed EM-SA hybrid algorithm is summarized in Fig. 6.

3.4.1. Parameter tuning

Further to the parameters that are separately tuned for the SA and EM algorithms, we need to tune the main parameters engaged in the hybrid algorithm to find any possible interaction between parameters. To that end, a full factorial design in the DOE is

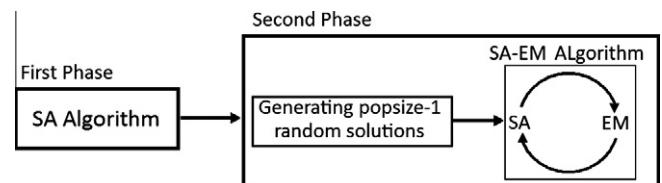


Fig. 6. General outline of the hybrid algorithm.

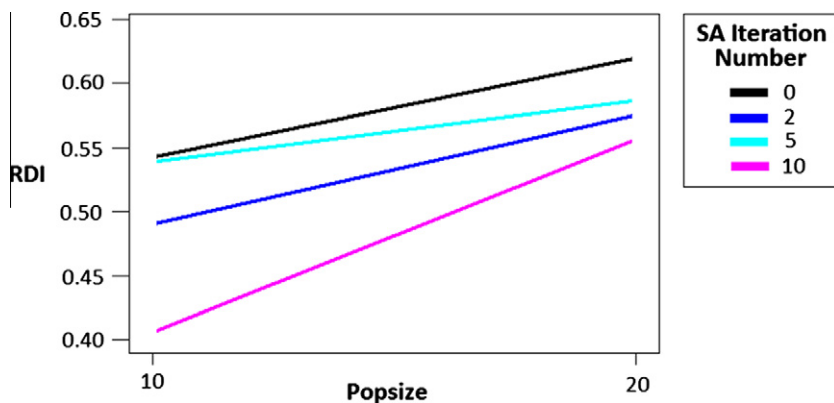


Fig. 7. Interaction plot for the SA-EM parameters.

applied. We consider two levels (i.e., 10 and 20) for the *popsiz* and four levels for the number of iterations (i.e., 0, 2, 5, 10) defined in the SA algorithm. If the latter parameter is considered 0, it means that we consider the traditional EM without benefiting the SA algorithm. For other levels of the SA iteration parameter, we set a fixed initial temperature to 150 and the cooling factor to 0.32, 0.63, and 0.79 for levels 2, 5, and 10, respectively. The related results are shown in Fig. 7. According to this figure, it is concluded that the best pair is 10 for both population size and SA iteration number, which is accompanied with 0.7 for the cooling factor parameter.

4. Experimental results

To illustrate the effectiveness and performance of the hybrid algorithm (i.e., EM-SA) proposed in this paper, it is coded in VB on a Laptop with Pentium IV Core 2 Duo 2.53 GHz CPU. The outputs of this hybrid algorithm are compared with that achieved by the SA algorithm alone and also the optimum solutions achieved in an hour running time. Some instances are randomly generated. Each instance is characterized by a number of parameters, such as number of jobs, number of machines, and operation routes of jobs, release dates of jobs, due dates, processing times, and the period length. All the generated instances are based on the following assumptions specified below:

1. All jobs meet all machines.
2. The routes correspondence with the jobs are randomly generated.
3. The release dates corresponding to each job are equal to zero.
4. Processing times are all randomly generated integer numbers between the interval [10, 20].
5. The objective is to minimize the TWT, where the weights are equal to 1, and the related due dates are zero.

The related results of the given instances are shown in Table 3. These results show that the hybrid EM-SA can improve the solutions obtained by using SA alone. The improvement in the solved instances is equal to 3.5 % in average. It is possible to find the optimum solutions of the instances smaller than 8 jobs and 8 machines in less than one hour running time. By comparing the results of EM-SA with the optimum solutions, we conclude that the EM-SA algorithm can achieve the optimum solutions of six instances among 13 solved ones.

5. Conclusion

This paper has first investigated the simulated annealing (SA) and electromagnetism-like mechanism (EM) algorithms in order

Table 3
Outputs of the proposed algorithms.

Job	Machine	Period	Algorithms		
			SA	EM-SA	Optimum
6	6	105	Inf.	Inf.	Inf.
6	6	120	353	331	275
6	6	150	193	193	167
6	6	300	181	169	167
6	7	120	387	355	325
6	7	150	257	216	196
6	7	180	189	180	180
6	7	300	189	180	180
6	8	180	177	177	177
6	8	360	177	177	177
6	10	180	Inf.	Inf.	Inf.
6	10	210	183	183	165
6	10	300	183	183	165
8	8	180	Inf.	Inf.	Inf.
8	8	210	323	323	323
8	8	300	323	323	323

Inf. stands for Infeasible.

to solve the periodic job shop scheduling problems (PJSSP). Then, an efficient hybrid EM-SA algorithm based on electromagnetism-like mechanism (EM) and simulated annealing (SA) has been proposed in order to solve periodic job shop scheduling problems (PJSSP). The parameters associated with the algorithms are tuned so that it is ensured that the algorithms perform in a high efficiency. The performance of the proposed hybrid EM-SA algorithm has been evaluated in comparison with the results obtained by the SA algorithm alone as well as the optimum solutions achieved by the branch-and-bound (B&B) algorithm after one hour running of the Lingo software. The achieved results demonstrate the effectiveness of the proposed hybrid EM-SA algorithm. Future research directions can be as follows: (1) defining some other well-known methods to solve the problem, such as branch-and-bound algorithm; (2) defining the lower bounds in order to have a better comparison amongst the algorithms; (3) solving the PJSSP under an uncertain environment; (4) testing other well-known meta-heuristics; (5) considering other objective functions; and (6) extending the proposed algorithm to other similar scheduling problems as well as train scheduling one.

Up to now, a periodic scheduling problem has not been addressed by any meta-heuristic method and is an open field for these near-optimal methods. For this reason for the future research, there is a need to define specific benchmarks for measuring the performance of the proposed meta-heuristics. The data regarding the randomly generated instances are accessible by writing to the authors.

References

- Brucker, P., & Kampmeyer, T. (2008). A general model for cyclic machine scheduling problems. *Discrete Applied Mathematics*, 156, 2561–2572.
- Cavory, G., Dupas, R., & Goncalves, G. (2005). A genetic approach to solving the problem of cyclic job shop scheduling with linear constraints. *European Journal of Operational Research*, 161, 73–85.
- Chang, P. C., Chen, S. H., & Fan, C. Y. (2009). A hybrid electromagnetism-like algorithm for single machine scheduling problem. *Expert Systems with Applications*, 36, 1259–1267.
- Garey, M. R., Johnson, D. S., & Sethi, R. (1976). The complexity of flowshop and job-shop scheduling. *Mathematics of Operations Research*, 1, 117–129.
- Gol-Alikhani, M., Javadian, N., & Tavakkoli-Moghaddam, R. (2009). A novel hybrid approach combining electromagnetism-like method with Solis and Wets local search for continuous optimization problems. *Journal of Global Optimization*, 44(2), 227–234.
- Kimbrel, T., & Sviridenko, M. (2008). High-multiplicity cyclic job shop scheduling. *Operations Research Letters*, 36, 574–578.
- Kinder, M. (2008). *Models for periodic timetabling*. Berlin: Technische Universität.
- Liebchen, C., Peeters, L. 2001. Some Practical Aspects of Periodic Timetabling. In: Chameni, P., et al, (Ed.), *Operations Research*, Springer, 2002.
- Lin, T.-L. et al. (2009). An efficient job-shop scheduling algorithm based on particle swarm optimization. *Expert Systems with Applications*. doi:10.1016/j.eswa.2009.08.015.
- Nachtigall, K. (1996). Cutting planes for a polyhedron associated with a periodic network, DLR Interner Bericht, 112-96/17.
- Naderi, B., Khalili, M., & Tavakkoli-Moghaddam, R. (2009a). A hybrid artificial immune algorithm for a realistic variant of job shops to minimize the total completion time. *Computers & Industrial Engineering*, 56(4), 1494–1501.
- Naderi, B., Tavakkoli-Moghaddam, R., Khalili, M. (2009b). Electromagnetism-like mechanism and simulated annealing algorithms for flowshop scheduling problems minimizing the total weighted tardiness and makespan, *Knowledge-Based Systems*, doi: 10.1016/j.knosys.2009.06.002.
- Odiijk, M. A. (1994). Construction of periodic timetables; part I: A cutting plane algorithm, Technical Report, DUT-TWI-94-61, Delft, The Netherlands.
- Pan, C.-H., & Huang, H.-C. (2009). A hybrid genetic algorithm for no-wait job shop scheduling problems. *Expert Systems with Applications*, 36, 5800–5806.
- Serafini, P., & Ukovich, W. (1989). A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4), 550–581.
- Sha, D. Y., & Lin, H. H. (2009). A multi-objective PSO for job-shop scheduling problems. *Expert Systems with Applications*. doi:10.1016/j.eswa.2009.06.041.
- Song, J. S., & Lee, T. E. (1996). A tabu search procedure for periodic job shop scheduling. *Computers & Industrial Engineering*, 30, 433–447.
- Song, J. S., & Lee, T. E. (1998). Petri net modeling and scheduling for cyclic job shops with blocking. *Computers & Industrial Engineering*, 34(2), 281–295.
- Tavakkoli-Moghaddam, R., Khalili, M., & Naderi, B. (2009). A hybridization of simulated annealing and electromagnetic-like mechanism for job shop problems with machine availability and sequence-dependent setup times to minimize total weighted tardiness. *Soft Computing*, 13, 995–1006.
- Zhang, G., Shao, X., Li, P., & Gao, L. (2009). An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem. *Computers & Industrial Engineering*, 56, 1309–1318.