



## A New Multi-objective Job Shop Scheduling with Setup Times Using a Hybrid Genetic Algorithm

M. B. Fakhrazad, A. Sadeghieh, L. Emami\*

Department of Industrial Engineering, University of Yazd, Iran

### PAPER INFO

#### Paper history:

Received 17 March 2012

Received in revised form 18 May 2012

Accepted 18 October 2012

#### Keywords:

Job Shop Scheduling  
Sequence-dependent Setup Times  
Multi-objective Optimization  
Earliness/Tardiness  
Makespan  
Hybrid Evolutionary Algorithm

### ABSTRACT

This paper presents a new multi-objective job shop scheduling with sequence-dependent setup times. The objectives are to minimize the makespan and sum of the earliness and tardiness of jobs in a time window. Since a job shop scheduling problem has been proved to be NP-hard in a strong, traditional approaches cannot reach to an optimal solution in a reasonable time. Thus, we propose an efficient multi-objective hybrid genetic algorithm (GA). We assign fitness based dominance relation and weighted aggregate in the genetic algorithm and local search, respectively. We take a variable neighborhood search algorithm as a local improving procedure in the proposed algorithm to the best individuals in the population of GA every specific number generations. To validate the efficiency of our proposed HGA, a number of test problems are solved. Its performance based on some comparison metrics is compared with a prominent multi-objective evolutionary algorithm, namely SPEA-II. The computational results show that the proposed HGA outperforms the SPEAII algorithm.

doi: 10.5829/idosi.ije.2013.26.02b.11

## 1. INTRODUCTION

Scheduling problem is the allocation of resources to perform a set of activities in a certain period of time [1]. Job shop scheduling problem (JSSP) is one of the most complicated combinatorial optimizations. JSSP may be described as follows: we have a set of  $n$  jobs that need to be done on a set of  $m$  machines [2]. Each job has its own processing route; that is, different jobs process machines in different manners. Each job might need to be performed only on some  $m$  machines, not all of them.

A lot of studies have been carried out in this case of scheduling problem in very different work conditions and criteria. Most researches have considered a single criterion whereas, in the real world, a decision maker (DM) often faces multiple criteria. Thus, we planned to formulate a job shop scheduling problem as a bi-objective one which simultaneously minimizes the maximum completion time and the sum of the earliness and tardiness (ET) of jobs in due window problems. Both of these objectives are applicable to many industries. Makespan minimization implies the

maximization of the throughput and the second objective comes from the just-in-time (JIT) manufacturing strategies. This objective, which is more realistic than ET problems, assumes that the due date of a job is an interval rather than a point in time. This model is clearly more general and practical than the first one since it includes the first model as its special case with zero due window size. In fact, the scheduling problem in a due window is a general case that includes ET results as special case in which the interval of time (due window of the job) is zero [3]. In other word, if we assume the values of left end and the right end of the window are the same, the sum of the earliness and tardiness of jobs in due window results is equal to the sum of the earliness and tardiness of jobs results [3].

In general, there are two common approaches to deal with multi-objective optimization problems: a priori and Pareto approaches. A priori approach usually converts a multiple objective function into a single objective function or define a preference order to optimize the objective functions one by one. This approach finds only a single optimal at a time. A Pareto approach, on the other hand, tries to find multiple solutions of Pareto-optimal set at a time. This approach evaluates solution quality based on the concept of

\*Corresponding Author Email: [Leila.emamy@gmail.com](mailto:Leila.emamy@gmail.com) (L. Emami)

Pareto dominance. In this paper, we used a method based on aggregation functions and Pareto dominance.

In this study, we consider multi-objective job shop scheduling with sequence-dependent setup times (SDST). In many real-life situations such as chemical, printing, pharmaceutical and automobile manufacturing, the setup operations, such as cleaning up or changing tools, are not only often required between jobs but they are also strongly dependent on the immediately preceding process on the same machine [4, 5] (sequence-dependent). The consideration of sequence-dependent setup times is gaining increasing attention among researchers in recent years. The incentive behind this assumption is to obtain tremendous savings when setup times are explicitly included in scheduling decisions [6]. With respect to the corresponding explanation, we take into account SDSTs in our problem. The JSSP is known to be a NP-hard optimization problem [7]. Thus, the bi-objective job shop scheduling with setup times is more complex and NP-hard and consequently, designing a meta heuristic algorithm to solve this problem is desirable. As such, this paper proposes an efficient multi-objective hybrid genetic algorithm. Dominance-based and weighted aggregate fitness assignment methods are used in the parts of genetic algorithm and local search, respectively.

The remainder of this paper is organized as follows: Section 2 gives a brief literature review of job shop scheduling. We describe our problem and the related mathematical model in section 3. Section 4 introduces the proposed hybrid algorithm and multi-objective technique that we used in this paper. We show the experimental results obtained by the proposed HGA and then compare these results with a multi-objective evolutionary algorithm, called SPEA-II. Finally, the conclusion is presented in section 6.

State equations are constructed by taking appropriate transition rates. In section 3, matrix geometric solution of the system is given. In section 4, various performance characteristics of the system are formulated explicitly in terms of steady state probabilities. In section 5, we perform comparative study of the system characteristics for various input parameters. In the last section, conclusions and future scopes of the work are provided.

## 2. LITERATURE REVIEW

Many studies have been done on job shop scheduling problem in both academic and industrial environments. Since this problem belongs to a NP-hard problem, we cannot find an exact solution within a reasonable computation time. The majority of papers on job shop scheduling problem is concentrated on single criterion problems. Most studies of single-objective JSSP result in a schedule to minimize the time required to complete all jobs, i.e., to minimize the makespan. Many

approximate methods are presented to overcome the limitations of exact enumeration techniques. For example, the approximate approaches include simulated annealing (SA) [8], tabu search [9-11] and genetic algorithms (GA) [12-15] proposed for JSSP to minimize the makespan.

Job shop scheduling problem is seldom considered sequence-dependent setup times because SDST scheduling problem is more complex. Some studies have been done on JSSP with SDST as follows. Gupta [16] presented a branch-and-bound algorithm for a SDST job shop. Another branch-and-bound algorithm was proposed by Brucker and Thiele [17] for the same problem. Schutten [18] addressed the job shop with some assumptions, such as release and due dates, setup times and transportation times. He then proposed an extension of the shifting bottleneck procedure for the problem. Cheung and Zhou [19] presented a hybrid algorithm based on a genetic algorithm and a well-known dispatching rule for SDST JSSP, where the setup times are anticipatory. Artigues and Roubellat [20] introduced a polynomial insertion algorithm for multi-resource SDST JSSP to minimize the maximum lateness. Roshanaei et al. [21] presented a VNS method to solve job shop scheduling with sequence-dependent setup times under minimization of the makespan. Naderi et al. [22] presented job shop scheduling with dependent setup time and preventive maintenance that minimizes the makespan. They introduced two techniques to integrate production planning and preventive maintenance problems. Naderi et al. [23] developed a hybrid approach of simulated annealing and artificial immune algorithm to solve job shop scheduling with sequence-dependent setup time and flexible machine availability constraints on the purposed problem of minimizing the total completion time. Tavakkoli-Moghaddam et al. [24] presented a hybrid method of SA and electromagnetic-like mechanism to solve job shop scheduling problems with sequence-dependent setup times and availability constraint. Minimizing of the total weighted tardiness is considered as the objective function.

Several researches addressed JIT scheduling problems with due windows under different criteria. The majority of the literature on JIT scheduling deals with single machine problems. Kramer and Lee [25], Chen and Lee [26], Wan and Yen [27] and Yeung et al. [28] are the few papers that studied a multi machine problem with a due window model. A study on job shop scheduling problem with the due window model was conducted by Huang et al. [29] who proposed an ant colony optimization to minimize the total penalty due to earliness and tardiness.

However, realworld production systems require simultaneous achievement of multiple objective requirements. This means that the academic concentration of objectives in the JSSP must be

extended from single to multiple. Recent related JSSP research with multiple objectives is summarized as below. Skawa and Kubota [30] presented a genetic algorithm for job shop scheduling problem with fuzzy processing times and fuzzy due date. The objective is to maximize the minimum agreement index, to maximize the average agreement index and to minimize the maximum fuzzy completion time. Thiagarajan and Rajendran [31] considered a dynamic assembly job shop scheduling problem with some dispatching rules. Their objective functions were to minimize the sum of the weighted tardiness/earliness and weighted flow time of all the jobs. Low et al. [32] investigated job shop scheduling problems with re-entrant operations and sequence-dependent setup time. Three practical performance measures, minimum total job flow time, minimum total job tardiness and minimum machine idle time, were considered. First, an integer programming model was developed to optimize each single objective. Then, they employed a multiple-decision-making technique to evaluate three objectives simultaneously. A Pareto archived simulated annealing (PASA) method, was suggested by Suresh et al. [33] to find non-dominated solution sets for the JSSP with the objectives of minimizing the makespan and the mean flow time of jobs. Lei [34] considered a JSSP by targeting the makespan and the total tardiness. He first converted the problem into a continuous optimization problem and then presented a Pareto archive particle swarm optimization in which the global best position selection is combined with crowding measure based archive maintenance. Qian et al. [35] suggested a memetic algorithm based on differential evolution (MODEMA) to minimize the makespan and the total tardiness, which applies simultaneously the DE-based global search and an adaptive local search to balance the exploration and exploitation. Sha et al. [36] proposed a multi-objective particle swarm optimization (MOPSO) for an elaborate multi-objective job shop scheduling problem. Their objective function was the minimization of makespan, total tardiness, and total machine idle time. Cheng et al. [37] developed a hybrid algorithm combining dispatching rules, the shifting bottleneck procedure, and the evolutionary algorithm for solving multi-objective JSSP to minimize the makespan and the total tardiness. Fang et al. [38] proposed an optimization method of many goals scheduling based on grey relation theory and ant colony algorithm for the multi-objective JSSP. Qing et al. [39] presented a hybrid genetic algorithm for an inventory based two-objective job shop scheduling model in which both the makespan and the inventory capacity were as objectives and were optimized simultaneously.

In this paper, we deal with a multi-objective job shop scheduling problem with setup times. The makespan and sum of earliness and tardiness are to be optimized simultaneously. To tackle this problem, an effective

multi-objective hybrid genetic algorithm is designed for searching locally the Pareto-optimal frontier.

### 3. PROBLEM DESCRIPTION

This section presents a MILP model for the job shop scheduling with setup times. The model characterized here is based on the following hypotheses: This problem has  $n$  jobs and  $m$  machines. Each job has a different sequence of operations that must be processed on one specific machine out of the machine set ( $m$ ). All the  $n$  jobs are independent and available for processing at the time 0. We assume infinite intermediate buffer between machines. Each machine can process only one job at a time, and each job can be processed by only one machine at any time. Setup times are sequence-dependent and they are non-anticipatory, meaning that the setup can begin as soon as the job and the machine are both available. All the jobs have a due window ( $d_{j1}$ ,  $d_{j2}$ ), where  $d_{j1}$  is the earliest due date and  $d_{j2}$  is the latest due date ( $d_{j2}, d_{j1}$ ). To present the mathematical model, the following notations are needed:

#### 3.1. Mathematical Model

$$\min z_1 = E_j + T_j \quad (1)$$

$$\min z_2 = C_{\max} \quad (2)$$

$$c_{ij} \geq c_{kj} \times R_{kj} + p_{ij} + \sum_{l=0}^n X_{ilj} \times s_{ilj} \times R_{ij} - M(1 - a_{ikj}) \quad (3)$$

$$j = 1, 2, \dots, n; \quad j \neq l; i, \quad k = 1, 2, \dots, m$$

$$c_{ij} \geq c_{il} \times R_{il} + p_{ij} + s_{ilj} - M(1 - X_{ilj}) \quad (4)$$

$$j = 1, 2, \dots, n; \quad l = 0, 1, 2, \dots, n;$$

$$i = 1, 2, \dots, m; \quad R_{ij} \geq 0$$

$$X_{ilj} \leq R_{ij} \times R_{il} ; \quad (5)$$

$$j = 1, 2, \dots, n; \quad l = 1, 2, \dots, n; \quad i = 1, 2, \dots, m$$

$$X_{ij0} \leq R_{il} ; \quad i = 1, 2, \dots, m; l = 1, 2, \dots, n \quad (6)$$

$$X_{ioj} \leq R_{ij} ; \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n \quad (7)$$

$$\sum_{\substack{j=0 \\ j \neq l}}^n X_{ilj} \times R_{il} = 1 ; \quad l = 1, 2, \dots, n; i = 1, 2, \dots, m \quad (8)$$

$$\sum_{\substack{l=0 \\ l \neq j}}^n X_{ilj} \times R_{ij} = 1 ; \quad j = 1, 2, \dots, n; i = 1, 2, \dots, m \quad (9)$$

$$c_j = \max_{1 \leq i \leq m} \{c_{ij}\} ; \quad j = 1, 2, \dots, n \quad (10)$$

$$T_j = \max \{0, c_j - d_{2j}\} ; \quad j = 1, 2, \dots, n \quad (11)$$

$$E_j = \max \{0, d_{1j} - c_j\} \quad ; \quad j=1,2,\dots,n \quad (12)$$

$$c_{\max} \geq c_j \quad j=1,2,\dots,n \quad (13)$$

$$c_{ij} \geq 0, T_j \geq 0, E_j \geq 0 \\ i=1,2,\dots,m; j=1,2,\dots,n \quad (14)$$

$$X_{ilj} \in \{0,1\} \\ i=1,2,\dots,m; j=1,2,\dots,n; l=0,1,2,\dots,n \quad (15)$$

Equations (1) and (2) determine the objective functions of our presented model, the first one is to minimize the sum of tardiness and earliness of jobs and the second one is to minimize the makespan. Equation (3) ensures that  $O_{ij}$  cannot start before the process of the previous job  $j$  operation completes. Constraint set (4) states that if  $O_{ij}$  is processed immediately after  $O_{il}$ , it cannot begin before  $O_{il}$  completes, while constraints (5),(6) and (7) ensure 0 values for variables  $X_{ij}$  that do not process on machine  $i$ . Constraint set (8) and (9) express that before each job there is only one job to be processed on the same machine and after each job there is only one job to be processed on the same machine, respectively. Constraint set (10) calculates the completion time of each job. Equations (11) and (12) reflect the tardiness and earliness for each job with respect to the defined earliest and latest due dates, respectively. Constraint (13) calculates the makespan. Finally, there are constraint sets (14) and (15) to define the decision variables.

#### 4. SOLUTION APPROACH

The pseudo-code of our hybrid genetic algorithm is as follows. Details of each step will be given in the following subsections.

---

$P$ =population, $p_i$ =parents, $c_i$ =child, $c$ =childs after mutation, $r_m$ =mutation rate,  $N_f$ =interval to do VNS,  $A$ =Pareto archive, $n$ =current iteration, $N_G$ =the maximum number of iteration

Randomly generating  $4*(pop\_size)$  chromosomes. Select  $best(pop\_size)/2$  chromosomes regarding  $Z_1$  (first objective function) and  $best(pop\_size)/2$  chromosomes regarding  $Z_2$  (second objective function) as initial population

While  $n < N_G$

Selection

$(c_1, c_2) = crossover(p_1, p_2);$

$c'_1 \leftarrow mutate(c_1, r_m)$

$c'_2 \leftarrow mutate(c_2, r_m)$

$(c'_1, c'_2) \leftarrow select(p_1, p_2, c_1, c_2);$

If  $n \bmod N_f = 0$  then

---



---

Solution improvement:

{

$S \leftarrow$  pick a solution from non-dominated solution

$K=1$

While the stopping criterion is not met do

Apply NSS type  $k$

If  $S$  is improved do

$K=1$

else

$k=k+1$

if  $k=4$  do

$k=1$

endif

endif

endwhile

}

End if

$n=n+1$

end while

---

**4. 1. Chromosome Encoding** Encoding is an important decision for designing a metaheuristic algorithm and making the solutions recognisable for the algorithm. The most frequently used representation for job shop scheduling is the operation-based representation [40]. In this representation, if a job has  $m$  operations that must be processed on  $m$  machines, the job number occurs for  $m$  times in the permutation. By scanning the permutation from left to right, one may see that the  $k$ th occurrence of a job number refers to the  $k$ th operation in the technological sequence of this job. For example, a problem with three jobs and three machines is shown as  $\{1\ 1\ 2\ 2\ 2\ 3\ 3\ 3\}$ . Job 1 has two operations, and Jobs 2 and 3 consist of three operations by repeated three times. So, for this problem, we have eight operations. In this example, assume a chromosome as  $\{2\ 1\ 2\ 3\ 2\ 1\ 3\ 3\}$ , the third position of it is 2. Here, the symbol "2" denotes the second operation of job 2 because number 2 has been repeated twice. If a job number is repeated as the number of operation, the solution is always feasible.

**4. 2. Fitness Assignment** After chromosomes are decoded and their related objectives ( $C_{\max}$  and ET) are calculated, we assign fitness values to them to show their importance in the evolutionary process. There are various good fitness assignment mechanisms for multi-objectives optimization. In this paper, we choose a mechanism based on NSGAII [41]. It assigns a fitness value based on the level of non-domination (Pareto ranking( $r$ )) and the crowding distance( $d$ ).

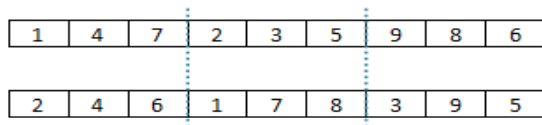
**4. 3. Selection Operator** When selecting individuals as parents to produce an offspring, we favor those with higher solution quality and diversity by the the selection mechanism shown below.

First, the mutating pool fills with non-dominated solutions from Pareto archive set. To fulfill the rest, we apply tournament selection mechanism. For this purpose, we select solutions based on three criteria: rank, crowding distance and euclidean distance (distance between the solutions and the ideal point). So, the pool is a subset of both diverse and high quality solutions with an approximation of the Pareto-optimal set.

**4. 4. Crossover and Mutation** An  $e$  percentage of selected chromosomes transfer to the next generation by reproduction. The remaining chromosomes  $(1 - e)\%$  are generated through an operator called crossover and mutation. There are a wide variety of proposed crossover operators suitable for the foregoing encoding scheme in the literature of job shop scheduling problems. In this study, a novel crossover is proposed and executed as follows: at first, each solution is decoded as an operation number given by:

|   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|
| 3 | 1 | 3 | 2 | 3 | 1 | 2 | 2 | 1 |
| 7 | 1 | 8 | 4 | 9 | 2 | 5 | 6 | 3 |

Consider the following two chromosomes after decoding:



Two crossover points are selected randomly. Let the partial chromosomes between  $P_1$  and  $P_2$  in each parent be  $C_1$  and  $C_2$ , respectively. For each operation, a random number is generated. If the value is less than 0.8, the operation from the parent 1(2) is copied to the offspring 2(1), otherwise the operation from the second parent is selected.

|      |      |      |      |      |      |   |   |   |
|------|------|------|------|------|------|---|---|---|
| 0.61 | 0.32 | 0.85 | 0.43 | 0.72 | 0.18 |   |   |   |
| 4    | 6    | 1    | 2    | 3    | 5    | 7 | 8 | 9 |
| 4    | 2    | 6    | 1    | 7    | 8    | 3 | 5 | 9 |

The inversion mutation chooses two random cut points and the elements between the cut points are then reversed. The probability of mutation ( $r_m$ ) is defined as the probability by which each offspring undergoes a mutation operation.

**4. 5. Environmental Selection** One other selection in genetic algorithm is environmental selection. In our proposed algorithm to keep the population diversity, after two offsprings are generated, we replace two

parents with two individuals with a minimum  $r/d$ . This mechanism implicitly implements the elitism; in addition, doing the replacement avoids the rapid growth of individuals with similar gene structures and the loss of population diversity.

**4. 6. Improvement of Non-dominated Solutions by VNS**

Our genetic algorithm is combined with a local search to reach its advantages and improve the search ability. The used local search procedure is Variable Neighborhood Search (VNS) algorithm [22].

In this algorithm, three different types of neighborhood search structures are used. They are based on insertion neighborhood.

We do the local search for the non-dominated solutions in the population of GA in every  $N_{LS}$  generation. The pseudo-code of three NSSs is as follows:

---

**Procedure: NSS\_type\_1**

```

Improvement=yes
While(Improvement=yes)do
    improvement=no
    i=1
    while i<=NO do (NO=max of iteration
        remove operation h at random and with out
        repetition from the current sequence S
        S'=put operation h into a new random position
        in current sequence S
        If fit_function(S')<fit_function(S) do
            S=S'
            i=NO
            improvement=yes
        endif
        i=i+1
    endwhile
endwhile
    
```

---



---

**Procedure: NSS\_type\_2**

```

i=1
While i<=NO do
    ii=ii+1
    while ii<=NO do
        remove i and ii-th operations of current sequence S
        S'=relocate this two operations in to two new
        random positions
        if fit_function(S')<fit_function(S) do
            S=S'
            ii=NO
            i=NO
            improvement=yes
        endif
        ii=ii+1
    endwhile
    i=i+1
endwhile
    
```

---

**Procedure: NSS\_type\_3**


---

```

for i=1:NO

    remove three randomly selected jobs from the current
    sequence S
    S=relocate these three jobs in to three random positions
endfor
S=find best(S) % even if sequence best(S) does not improve
fitness_function

```

---

**The pseudo-code of VNS**

```

For each input
K=1
While the stopping criterion is not met do
    Apply NSS type k
    If S is improved do
        K=1
    else
        k=k+1
        if k=4 do
            K=1
        endif
    endif
endwhile
end for

```

---

In the VNS [22], the best partial sequence is the one with the shortest makespan. In our algorithm, we propose an acceptance strategy to decide on best partial sequence(s) as follows:

$$F = \left[ \sum_{i=1}^2 w_j \left( \frac{f_i(x) - f_i^*}{f_i^*} \right)^p \right]^{\frac{1}{p}}, p = 1 \quad (16)$$

In this strategy, to compute the weights of the objectives, we first sort the individuals in the increasing order of  $C_{\max}$ . The weight vector  $w_j$  defined associated with the  $i$ th sorted individual is defined by  $w_j = (1 - (i-1) / (|A|-1)) / ((i-1) / (|A|-1))$ , where  $A$  is the set of individuals to do the local improvement [42].

**4. 7. Termination** The proposed hybrid genetic algorithm must be repeated for a pre-specified number of times before it terminates.

**5. EXPERIMENTAL RESULTS**

The performance of the proposed algorithm is compared with that of a well-known multi-objective evolutionary algorithm (EAs), namely SPEA-II. These two algorithms are coded in MatlabR2009a and executed on 2.2 GHz CPU, with a Windows 7 operating system using 4 GB of RAM. Herein, a brief discussion about the implementation of SPEA-II seems to be appropriate.

**5. 1. Strength Pareto Evolutionary Algorithm II (SPEA-II)**

Zitzler et al. [43] proposed a Pareto-based method, the strength Pareto evolutionary algorithm II (SPEA-II), which is an intelligent enhanced version of SPEA. In SPEA-II, each individual in both the main population and the elitist non-dominated archive is assigned a strength value, which incorporates both dominance and density information. On the basis of the strength value, the final rank value is determined by the summation of the strengths of the points that dominate the current individual. Meanwhile, a density estimation method is applied to obtain the density value of each individual. The final fitness is the sum of the rank and density values. Additionally, a truncation method is used to maintain a constant number of individuals in the Pareto archive [44].

**5. 2. Design of Experiment** The experiments are conducted for 13 problems. The following assumptions are held for all experiments.

- ❖ The processing times ( $p_{ij}$ ) for  $O_{ij}$  operations are integers and are generated from a uniform distribution of  $U(1,100)$ .
- ❖ Setup times are randomly generated from uniform distribution on the integer between  $[1, 50]$ .
- ❖ Another important issue is the earliest and latest due dates of the jobs. To generate the due dates of all  $n$  jobs, we proposed the following steps:
- Compute the total processing time of each job on all  $g$  stages.

$$p_j = \sum_{i=1}^m p_{ij}, \forall j \in N;$$

- Compute the average setup time for every possible subsequent job and sum it for all  $g$  stages.

$$S_j = \sum_{i=1}^m \frac{\sum_{l=1}^n s_{ilj}}{n-1}, \forall j \in N$$

Then,  $d_{j1}$  and  $d_{j2}$  are generated as follows:

$$d_{j1} = (1 + \text{random} \times 2) \times (s_j + p_j) \quad \forall j \in N$$

$$d_{j2} = d_{j1} + (\text{random} \times 2) \times (s_j + p_j) \quad \forall j \in N$$

The experimental set is illustrated in Table 1. In this table, the rows demonstrate the parameters of the experiments and the columns illustrate the number of the alternative related to each parameter and the values of each one for 13 problems.

**5. 3. Evaluation Metrics** To validate the efficiency of the proposed algorithm, the following four metrics are taken into account.

Number of non-dominated solutions (NDS): This metric shows the non-dominated solutions that are obtained by each algorithm. Based on this metric, the higher value, the better of solution quality we have [45].

MID (mean ideal distance): The closeness between Pareto solution and the ideal point (0, 0) is [3]:

$$MID = \frac{\sum_{i=1}^n c_i}{n}, \quad c_i = \sqrt{f_{1i}^2 + f_{2i}^2} \quad (17)$$

Spacing Metric (SM): The spacing metric enables us to measure the uniformity of the point spread within the solution set. The definition of this metric is given below [46]:

$$s = \frac{\sum_{i=1}^N |\bar{d} - d_i|}{(N-1)\bar{d}} \quad (18)$$

The spread of non-dominance solution (SNS): SNS, as diversity measure, can be calculated as the follows [3]:

$$SNS = \sqrt{\frac{\sum_{i=1}^n (MID - c_i)^2}{n-1}} \quad (19)$$

The higher value of SNS, the better of solution quality we have (more diversity in obtained solution).

**5. 4. Comparison of Results** The performance of the proposed HGA is compared with well-known multi-objective evolutionary algorithm namely SPEAII, Some parameters are set in advance as follows.

**5. 5. Computational Results** For each problem size, twenty random problems are generated and tested. For each random problems, all the considered genetic algorithms perform 20 runs. The proposed GA (PGA) and HGA are applied to a number of test problems and their performance is compared with SPEA-II. Tables 3 and 5 list the average values of the above mentioned comparison metrics. As shown in Table 3, the proposed genetic algorithm is superior to the SPEAII. In order to verify the statistical validity of the results we use a paired T-test for the following null hypothesis and the alternative hypothesis with respect to NDS and SNS:

$$H_0 : \mu_{SPEAII} - \mu_{PGA} \geq 0$$

$$H_1 : \mu_{SPEAII} - \mu_{PGA} < 0$$

For performance measures SM and MID, we considered the following test:

$$H_0 : \mu_{SPEAII} - \mu_{PGA} \leq 0$$

$$H_1 : \mu_{SPEAII} - \mu_{PGA} > 0$$

If the null hypothesis is rejected with a certain significance level, we can conclude that PGA performs better than SPEAII statistically. The statistical analysis has been performed by a software, MINITAB 14.

Table 4 shows the p-values of the test. The null hypothesis is rejected with  $\alpha$  if the p-value is smaller than  $\alpha$ . Table 5 shows the results is obtained from HGA

and SPEAII. The results show that the proposed algorithm can obtain a greater number of Pareto optimal solutions with higher qualities than SPEA-II. The proposed HGA provides non-dominated solutions that have lower average values of the spacing metric. The data reveal that non-dominated solutions obtained by the proposed algorithm are more uniformly distributed in comparison with SPEA-II. The average values of the SNS metric in our proposed algorithm are considerably greater than SPEA-II (i.e HGA finds non-dominated solutions that diversify more).

In order to verify the statistical validity of the results we perform a paired T-test for the following null hypothesis and the alternative hypothesis with respect to NDS and SNS:

$$H_0 : \mu_{SPEAII} - \mu_{HGA} \geq 0$$

$$H_1 : \mu_{SPEAII} - \mu_{HGA} < 0$$

For performance measure SM and MID we considered the following test:

$$H_0 : \mu_{SPEAII} - \mu_{HGA} \leq 0$$

$$H_1 : \mu_{SPEAII} - \mu_{HGA} > 0$$

Table 6 shows the p-values of the test. The computational time of our proposed HGA and SPEA-II algorithms is shown in Table 7. This table expresses that the proposed algorithm is more time consuming than SPEA-II, so the computational time of this genetic algorithm is better than HGA.

**TABLE 1.** Experimental set for the given problems

| Parameter     | Alternative | Values   |
|---------------|-------------|--|
| No.jobs       | 5           | 100-50-30-20-10  |
| No.machines   | 3           | 20-15-10   |
| Process. time | U[1,100]    | 1  |
| Setup times   | U [1, 50]   | 1  |
| Due date      |             | $d_{j1} = (1 + random \times 2) \times (s_j + p_j) \quad \forall j \in N$<br>$d_{j2} = dj1 + (random \times 2) \times (s_j + p_j) \quad \forall j \in N$ |

**TABLE 2.** Parameters tuning

| Algorithm | Parameters  |
|-----------|---|
| HGA       | Population_size:110   |
|           | Mutation rate =0.15   |
|           | e =0.15   |
|           | $N_G = 350$   |
|           | $N_L = 35$  |
|           | nom =10   |
| SPEAII    | The binary tournament selection procedure is used.            |
|           | The selection rate is set to 0.8.                             |
|           | The ratios of OX and IN are set to 0.8 and 0.4, respectively. |
|           | The number of iteration set to 350.                           |

**TABLE 3.** Comperation of PGA and SPEAII

| Problem        | NDS  |        | MID     |          | SM     |        | SNS   |        |
|----------------|------|--------|---------|----------|--------|--------|-------|--------|
|                | PGA  | SPEAII | PGA     | SPEAII   | PGA    | SPEAII | PGA   | SPEAII |
| 10 × 10        | 8.2  | 9.6    | 1189.26 | 1155.36  | 0.503  | 0.52   | 15.72 | 18.33  |
| 20 × 10        | 11.7 | 11.1   | 1313.7  | 1342.029 | 0.836  | 0.853  | 16.93 | 14.86  |
| 20 × 15        | 12.3 | 10.6   | 1529.72 | 1561.21  | 0.617  | 0.728  | 10.3  | 11.83  |
| 20 × 20        | 9.4  | 10.7   | 1890.73 | 1867.33  | 0.842  | 0.764  | 8.24  | 7.13   |
| 30 × 10        | 13.2 | 12.3   | 2412.03 | 2426.19  | 0.653  | 0.675  | 22.81 | 16.1   |
| 30 × 15        | 10.4 | 13.1   | 1659.4  | 1646.07  | 0.513  | 0.563  | 29.8  | 33.6   |
| 30 × 20        | 13.3 | 12.6   | 1803.1  | 1816.21  | 0.728  | 0.7091 | 27.6  | 27.9   |
| 50 × 10        | 15.2 | 14.1   | 4383.8  | 4410.36  | 0.814  | 0.819  | 38.92 | 39.2   |
| 50 × 15        | 16.3 | 13.6   | 4166.09 | 4242.7   | 0.938  | 1.067  | 43.31 | 47.5   |
| 50 × 20        | 17.5 | 19.8   | 4583.61 | 4660.96  | 0.917  | 0.933  | 41.29 | 40.73  |
| 100 × 10       | 15.1 | 16.6   | 6102.9  | 5981.2   | 0.531  | 0.547  | 45.47 | 46.45  |
| 100 × 15       | 15.1 | 15.3   | 6206.27 | 6373.32  | 0.726  | 0.714  | 53.6  | 56.7   |
| 100 × 20       | 16.4 | 18.6   | 6273.33 | 6488.4   | 0.961  | 0.989  | 59.7  | 58.2   |
| <b>Average</b> | 13.3 | 13.7   | 3180.3  | 3382.41  | 0.7403 | 0.76   | 33.53 | 32.094 |

**TABLE 4.** Paired T-test of PGA vs SPEAII ( $\alpha=0.05$ )

| Type of problem (No.jobs) | Evaluation metric | P-value | Result                   |
|---------------------------|-------------------|---------|--------------------------|
| 10,20                     | NDS               | 0.126   | accepted null hypothesis |
|                           | MID               | 0.369   | accepted null hypothesis |
|                           | SM                | 0.341   | accepted null hypothesis |
|                           | SNS               | 0.081   | accepted null hypothesis |
| 30                        | NDS               | 0.088   | accepted null hypothesis |
|                           | DM                | 0.501   | accepted null hypothesis |
|                           | SM                | 0.053   | accepted null hypothesis |
|                           | SNS               | 0.34    | accepted null hypothesis |
| 50                        | NDS               | 0.093   | accepted null hypothesis |
|                           | DM                | 0.129   | accepted null hypothesis |
|                           | SM                | 0.026   | rejected null hypothesis |
|                           | SNS               | 0.297   | accepted null hypothesis |
| 100                       | NDS               | 0.126   | accepted null hypothesis |
|                           | DM                | 0.093   | accepted null hypothesis |
|                           | SM                | 0.018   | rejected null hypothesis |
|                           | SNS               | 0.004   | rejected null hypothesis |
| Total                     | NDS               | 0.452   | accepted null hypothesis |
|                           | DM                | 0.21    | accepted null hypothesis |
|                           | SM                | 0.191   | accepted null hypothesis |
|                           | SNS               | 0.121   | accepted null hypothesis |



**TABLE 5.** Comperation of HGA and SPEAII

| Problem  | NDS   |        | MID      |          | SM     |        | SNS   |        |
|----------|-------|--------|----------|----------|--------|--------|-------|--------|
|          | HGA   | SPEAII | HGA      | SPEAII   | HGA    | SPEAII | HGA   | SPEAII |
| 10 × 10  | 11.3  | 9.6    | 1117.52  | 1155.36  | 0.482  | 0.52   | 17.9  | 18.33  |
| 20 × 10  | 14.2  | 11.1   | 1292.91  | 1342.029 | 0.612  | 0.853  | 19.65 | 14.86  |
| 20 × 15  | 13.7  | 10.6   | 1501.46  | 1561.21  | 0.417  | 0.728  | 9.7   | 11.83  |
| 20 × 20  | 10.9  | 10.7   | 1817.38  | 1867.33  | 0.541  | 0.764  | 10.2  | 7.13   |
| 30 × 10  | 17.1  | 12.3   | 2362.19  | 2426.19  | 0.327  | 0.675  | 22.81 | 16.1   |
| 30 × 15  | 17.7  | 13.1   | 1606.1   | 1646.07  | 0.3130 | 0.563  | 31.4  | 33.6   |
| 30 × 20  | 18.9  | 12.6   | 1731.71  | 1816.21  | 0.762  | 0.7091 | 28.2  | 27.9   |
| 50 × 10  | 20.2  | 14.1   | 3821.36  | 4410.36  | 0.611  | 0.819  | 38.46 | 39.2   |
| 50 × 15  | 25.6  | 13.6   | 3519.83  | 4242.7   | 0.861  | 1.067  | 43.66 | 47.5   |
| 50 × 20  | 21.7  | 19.8   | 3709.06  | 4660.96  | 0.713  | 0.933  | 44.6  | 40.73  |
| 100 × 10 | 27.3  | 16.6   | 5673.22  | 5981.2   | 0.531  | 0.547  | 52.1  | 46.45  |
| 100 × 15 | 31.1  | 15.3   | 5461.46  | 6373.32  | 0.663  | 0.714  | 58.9  | 56.7   |
| 100 × 20 | 22.4  | 18.6   | 5108.1   | 6488.4   | 0.951  | 0.989  | 66.3  | 58.2   |
| Average  | 19.39 | 13.7   | 2924.028 | 3382.41  | 0.726  | 0.76   | 36.29 | 32.094 |

**TABLE 6.** Paired T-test of HGA vs SPEAII( $\alpha=0.05$ )

| Type of problem (NO.jobs) | Evaluation Metric | P-value | Result                   |
|---------------------------|-------------------|---------|--------------------------|
| 10,20                     | NDS               | 0.015   | rejected null hypothesis |
|                           | MID               | 0.169   | accepted null hypothesis |
|                           | SM                | 0.301   | accepted null hypothesis |
|                           | SNS               | 0.0216  | rejected null hypothesis |
| 30                        | NDS               | 0.008   | rejected null hypothesis |
|                           | DM                | 0.60    | accepted null hypothesis |
|                           | SM                | 0.001   | rejected null hypothesis |
|                           | SNS               | 0.134   | accepted null hypothesis |
| 50                        | NDS               | 0.0348  | rejected null hypothesis |
|                           | DM                | 0.0312  | rejected null hypothesis |
|                           | SM                | 0.024   | rejected null hypothesis |
|                           | SNS               | 0.097   | accepted null hypothesis |
| 100                       | NDS               | 0.001   | rejected null hypothesis |
|                           | DM                | 0.01    | rejected null hypothesis |
|                           | SM                | 0.001   | rejected null hypothesis |
|                           | SNS               | 0.011   | rejected null hypothesis |
| Total                     | NDS               | 0.452   | accepted null hypothesis |
|                           | DM                | 0.021   | rejected null hypothesis |
|                           | SM                | 0.001   | rejected null hypothesis |
|                           | SNS               | 0.011   | rejected null hypothesis |

TABLE 7. Computational time (s) for the HGA and SPEAII

| Problem  | HGA | SPEAII |
|----------|-----|--------|
| 10 × 10  | 17  | 13     |
| 20 × 10  | 31  | 21     |
| 20 × 15  | 44  | 34     |
| 20 × 20  | 51  | 43     |
| 30 × 10  | 69  | 62     |
| 30 × 15  | 86  | 71     |
| 30 × 20  | 98  | 84     |
| 50 × 10  | 156 | 149    |
| 50 × 15  | 188 | 169    |
| 50 × 20  | 279 | 240    |
| 100 × 10 | 536 | 449    |
| 100 × 15 | 599 | 512    |
| 100 × 20 | 716 | 569    |

## 6. CONCLUSION

This paper explains how to develop a new multi-objective job shop scheduling problem with makespan and sum of the earliness and tardiness of jobs, where a sequence-dependent setup time is necessary before starting the processing of a job. To find a set of Pareto solutions, a multi-objective hybrid genetic algorithm is proposed. Dominance-based and weighted aggregate fitness assignment methods are used in the development of genetic algorithm and the local search, respectively. To validate the proposed algorithm, various test problems were designed to evaluate the performance and reliability of the algorithm in comparison with a well-known multi-objective evolutionary algorithm called SPEA-II. Also, some useful comparison metrics were applied for the act of validation. The experimental results have indicated that our proposed algorithm outperforms SPEAII and is able to improve the quality of the obtained solutions.

A clue for future research is to consider the proposed algorithm on other types of scheduling problems. As another direction, it could be interesting to work on some other meta-heuristics and compare them with our proposed algorithm.

## 7. REFERENCES

- Baker, K. R., "Introduction to sequence and scheduling", New York: John Wiley & Sons, (1974).
- Sule, D. R., "Industrial scheduling, PWS Publishing company New York, Vol. 20, (1997).
- Behnamian, J., Zandieh, M. and Fatemi Ghomi, S. M. T., "Bi-objective parallel machines scheduling with sequence-dependent setup times using hybrid metaheuristics and weighted min-max technique", *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, Vol. 15, No. 7, (2011), 1313-1331.
- Tavakkoli-Moghaddam, R., Safaei, N. and Sassani, F., "A memetic algorithm for the flexible flow line scheduling problem with processor blocking", *Computers & Operations Research*, Vol. 36, No. 2, (2009), 402-414.
- Luh, P. B., Gou, L., Zhang, Y., Nagahora, T., Tsuji, M., Yoneda, K., Hasegawa, T., Kyoya, Y., and Kano, T., "Job shop scheduling with group-dependent setups, finite buffers, and long time horizon", *Annals of Operations Research*, Vol. 76, No., (1998), 233-259.
- Allahverdi, A., Ng, C., Cheng, T. and Kovalyov, M. Y., "A survey of scheduling problems with setup times or costs", *European Journal of Operational Research*, Vol. 187, No. 3, (2008), 985-1032.
- Ombuki, B. M. and Ventresca, M., "Local search genetic algorithms for the job shop scheduling problem", *Applied Intelligence*, Vol. 21, No. 1, (2004), 99-109.
- Louren, H. R., "Local optimization and the job-shop scheduling problem", *European Journal of Operational Research*, Vol. 83, (1995), 347-364.
- Nowicki, E. and Smutnicki, C., "A fast taboo search algorithm for the job shop problem", *Management Science*, Vol. 42, No. 6, (1996), 797-813.
- Pezzella, F. and Merelli, E., "A tabu search method guided by shifting bottleneck for the job shop scheduling problem", *European Journal of Operational Research*, Vol. 120, No. 2, (2000), 297-310.
- Sun, D., Batta, R. and Lin, L., "Effective job shop scheduling through active chain manipulation", *Computers & Operations Research*, Vol. 22, No. 2, (1995), 159-172.
- Bean, J. C., "Genetic algorithms and random keys for sequencing and optimization", *Operations Research Society of America (ORSA) Journal on Computing*, Vol. 6, No. 2, (1994), 154-160.

13. Goncalves, J. F., Mendes, J. J. M. and Resende, M. G. C., "A hybrid genetic algorithm for the job shop scheduling problem", *European Journal of Operational Research*, Vol. 167, No. 1, (2005), 77-95.
14. Kobayashi, S., Ono, I. and Yamamura, M., "An efficient genetic algorithm for job shop scheduling problems", in Proceedings of the 6th International Conference on Genetic Algorithms, Morgan Kaufmann, San Francisco. CA., (1995), 506-511.
15. Wang, L. and Zheng, D. Z., "An effective hybrid optimization strategy for job-shop scheduling problems", *Computers & Operations Research*, Vol. 28, No. 6, (2001), 585-596.
16. Sushil, K. G., "N jobs and m machines job-shop problems with sequence-dependent set-up times", *The International Journal of Production Research*, Vol. 20, No. 5, (1982), 643-656.
17. Brucker, P. and Thiele, O., "A branch & bound method for the general-shop problem with sequence dependent setup-times", *OR Spectrum*, Vol. 18, No. 3, (1996), 145-161.
18. Schutten, J., "Practical job shop scheduling", *Annals of Operations Research*, Vol. 83, No., (1998), 161-178.
19. Cheung, W. and Zhou, H., "Using genetic algorithms and heuristics for job shop scheduling with sequence-dependent setup times", *Annals of Operations Research*, Vol. 107, No. 1, (2001), 65-81.
20. Artigues, C. and Roubellat, O., "An efficient algorithm for operation insertion in a multi-resource job-shop schedule with sequence-dependent setup times", *Production Planning & Control*, Vol. 13, No. 2, (2002), 175-186.
21. Roshanaei, V., Naderi, B., Jolai, F. and Khalili, M., "A variable neighborhood search for job shop scheduling with set-up times to minimize makespan", *Future Generation Computer Systems*, Vol. 25, No. 6, (2009), 654-661.
22. Naderi, B., Zandieh, M. and Fatemi Ghomi, S. M. T., "Scheduling sequence-dependent setup time job shops with preventive maintenance", *The International Journal of Advanced Manufacturing Technology*, Vol. 43, No. 1, (2009), 170-181.
23. Naderi, B., Khalili, M. and Tavakkoli-Moghaddam, R., "A hybrid artificial immune algorithm for a realistic variant of job shops to minimize the total completion time", *Computers & Industrial Engineering*, Vol. 56, No. 4, (2009), 1494-1501.
24. Tavakkoli-Moghaddam, R., Khalili, M. and Naderi, B., "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-A Fusion of Foundations, Methodologies and Applications*, Vol. 13, No. 10, (2009), 995-1006.
25. Kramer, F. J. and Lee, C. Y., "Due window scheduling for parallel machines", *Mathematical and Computer Modelling*, Vol. 20, No. 2, (1994), 69-89.
26. Chen, Z. L., "Scheduling and common due date assignment with earliness-tardiness penalties and batch delivery costs", *European Journal of Operational Research*, Vol. 93, No. 1, (1996), 49-60.
27. Wan, G. and Yen, B. P. C., "Tabu search for single machine scheduling with distinct due windows and weighted earliness/tardiness penalties", *European Journal of Operational Research*, Vol. 142, No. 2, (2002), 271-281.
28. Yeung, W., Oguz, C. and Edwin Cheng, T., "Two-stage flowshop earliness and tardiness machine scheduling involving a common due window", *International Journal of Production Economics*, Vol. 90, No. 3, (2004), 421-434.
29. Huang, R. H. and Yang, C. L., "Ant colony system for job shop scheduling with time windows", *The International Journal of Advanced Manufacturing Technology*, Vol. 39, No. 1, (2008), 151-157.
30. Sakawa, M. and Kubota, R., "Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms", *European Journal of Operational Research*, Vol. 120, No. 2, (2000), 393-407.
31. Thiagarajan, S. and Rajendran, C., "Scheduling in dynamic assembly job-shops to minimize the sum of weighted earliness, weighted tardiness and weighted flowtime of jobs", *Computers & Industrial Engineering*, Vol. 49, No. 4, (2005), 463-503.
32. Low, C., Wu, T. H. and Hsu, C. M., "Mathematical modelling of multi-objective job shop scheduling with dependent setups and re-entrant operations", *The International Journal of Advanced Manufacturing Technology*, Vol. 27, No. 1, (2005), 181-189.
33. Suresh, R. and Mohanasundaram, K., "Pareto archived simulated annealing for job shop scheduling with multiple objectives", *The International Journal of Advanced Manufacturing Technology*, Vol. 29, No. 1, (2006), 184-196.
34. Lei, D., "A Pareto archive particle swarm optimization for multi-objective job shop scheduling", *Computers & Industrial Engineering*, Vol. 54, No. 4, (2008), 960-971.
35. Qian, B., Wang, L., Huang, D. X. and Wang, X., "Scheduling multi-objective job shops using a memetic algorithm based on differential evolution", *The International Journal of Advanced Manufacturing Technology*, Vol. 35, No. 9, (2008), 1014-1027.
36. Sha, D. and Lin, H. H., "A multi-objective PSO for job-shop scheduling problems", *Expert Systems with Applications*, Vol. 37, No. 2, (2010), 1065-1070.
37. Cheng, H. C., Chiang, T. C. and Fu, L. C., "A two-stage hybrid memetic algorithm for multiobjective job shop scheduling", *Expert systems with applications*, Vol. 38, No. 9, (2011), 10983-10998.
38. Fang, Y. D., Wang, F., Wang, H., "Research of multi-objective optimization study 548 for job shop scheduling problem based on grey ant colony algorithm", *Advanced Materials Research*, Vol. 308, (2011), 1033-1036.
39. Qing, R., Wang, Y. and Wang, X., "Inventory Based Two-Objective Job Shop Scheduling Model and Its Hybrid Genetic Algorithm", *Applied Soft Computing*, (in press).
40. Cheng, R., Gen, M. and Tsujimura, Y., "A tutorial survey of job-shop scheduling problems using genetic algorithms, part II: hybrid genetic search strategies", *Computers & Industrial Engineering*, Vol. 36, No. 2, (1999), 343-364.
41. Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., "A fast and elitist multiobjective genetic algorithm: NSGA-II", *Evolutionary Computation, IEEE Transactions on*, Vol. 6, No. 2, (2002), 182-197.
42. Chiang, T. C., Cheng, H. C. and Fu, L. C., "NNMA: An effective memetic algorithm for solving multiobjective permutation flow shop scheduling problems", *Expert systems with applications*, Vol. 38, No. 5, (2011), 5986-5999.
43. Zitzler, E., Laumanns, M., & Theile, L. "SPEA II: "Improving the strength Pareto evolutionary algorithm"", *Computer Engineering and Networks Laboratory (TIK)*, (2001).
44. Rahimi-Vahed, A. and Mirghorbani, S., "A multi-objective particle swarm for a flow shop scheduling problem", *Journal of combinatorial optimization*, Vol. 13, No. 1, (2007), 79-102.
45. Schaffer, J. D., "Multiple objective optimization with vector evaluated genetic algorithms", in Proceedings of the 1st international Conference on Genetic Algorithms, L. Erlbaum Associates Inc., (1985), 93-100.
46. Srinivas, N. and Deb, K., "Multiobjective optimization using nondominated sorting in genetic algorithms", *Evolutionary computation*, Vol. 2, No. 3, (1994), 221-248.

## NOMNECLATURE

| Indices    |  |
|------------|--|
| $i, k$     | Machine index $\{1, 2, \dots, m\}$                                 |
| $j$        | Jobs index $\{0, 1, 2, \dots, n\}$                                 |
| $l$        | Jobs index $\{0, 1, 2, \dots, n\}$                                 |
| Parameters |  |
| $n$        | The number of jobs   |
| $m$        | The number of machines $m$   |
| $P_{ij}$   | The processing time of job $j$ on machine $i$                      |
| $d_{1j}$   | The earliest due date  |
| $d_{2j}$   | The latest due date  |
| $S_{lij}$  | The setup time of job $j$ on machine $i$ immediately after job $l$ |

$$a_{kij} = \begin{cases} 1 & \text{if job } j \text{ must visit machine } i \text{ immediately after machine } k \\ 0 & \text{otherwise} \end{cases}$$

$$R_{ij} = \begin{cases} 1 & \text{if job } j \text{ is processed on machine } i \\ 0 & \text{otherwise} \end{cases}$$

$M$  A large positive number

## Variables

$C_{ij}$  The completion time of job  $j$  on machine  $i$

$E_j$  Earliness of job  $j$

$T_j$  Tardiness of job  $j$

$$X_{ij} = \begin{cases} 1 & \text{if job } j \text{ is processed immediately after} \\ & \text{job } l \text{ on machine } i \\ 0 & \text{otherwise} \end{cases}$$

## A New Multi-objective Job Shop Scheduling with Setup Times Using a Hybrid Genetic Algorithm

M. B. Fakhrazad, A. Sadeghieh, L. Emami

Department of Industrial Engineering, University of Yazd, Iran.

### PAPER INFO

### چکیده

#### Paper history:

Received 17 March 2012

Received in revised form 18 May 2012

Accepted 18 October 2012

#### Keywords:

Job Shop Scheduling  
Sequence-dependent Setup Times  
Multi-objective Optimization  
Earliness/Tardiness  
Makespan  
Hybrid Evolutionary Algorithm

در این مقاله یک مساله جدید چند هدفه زمانبندی کارگاهی با زمان های آماده سازی وابسته به توالی ارائه می شود. اهداف، کمینه کردن دامنه عملیات و مجموع دیرکرد و زودکرد کارها در یک بازه زمانی است. از آنجایی که مساله زمانبندی کارگاهی به عنوان یک مساله NP-hard ثابت شده است، رویکردهای سنتی در زمانی معقول به جواب بهینه نمی توانند برسند. بنابراین، ما یک الگوریتم ژنتیک ترکیب چندهدفه ای پیشنهاد داده ایم. برازندگی بر اساس روابط غلبه و مجموع وزنی توابع به ترتیب در الگوریتم ژنتیک و جستجوی محلی استفاده شده است. ما الگوریتم جستجوی همسایگی متغیر را به عنوان رویه بهبود دهنده محلی در الگوریتم پیشنهادی روی S بهترین جواب ها در تکرارهای مشخصی به کار گرفته ایم. جهت اثبات کردن کارایی الگوریتم پیشنهادی، تعدادی مساله حل شدند. اعتبار آن بر پایه چند معیار مقایسه ای با الگوریتم تکاملی شناخته شده SPEAII سنجیده شده است. نتایج نشان داد که الگوریتم HGA پیشنهادی بر الگوریتم SPEAII برتری دارد.

doi: 10.5829/idosi.ije.2013.26.02b.11