

A new mathematical model for the job shop scheduling problem with uncertain processing times

M. A. Shafia^a, M. Pourseyed Aghae^b and A. Jamili^{a*}

^aDepartment of Industrial Engineering, Iran University of Science and Technology, Tehran Iran

^bDepartment of Railroad Engineering, Iran University of Science and Technology, Tehran, Iran

ARTICLE INFO

Article history:

Received 1 Feb 2010

Received in revised form

10 June 2010

Accepted 20 June 2010

Available online 23 June 2010

Keywords:

Job Shop Scheduling

Robust optimization

Simulated annealing

Heuristic

Mixed Integer Programming

Uncertainty

Interruption

ABSTRACT

Job shop scheduling (JSS) problem has been one of the most interesting research issues in the literature during the recent years. JSS problem has been studied in different forms of deterministic, fuzzy, and stochastic at different depths. The idea of robust optimization (ROP), on the other hand, has earned a particular value to become a popular subject of the breakthrough for problem solving affairs amongst the researchers. Based on the emerged opportunity for illustrating a new area of search, a robust JSS problem is proposed as a challenge to this boundary of knowledge. The proposed method is capable of handling the perturbation which exists amongst the processing times. In fact, in many real world job scheduling problems, a small change in the processing times, not only causes a non-optimal solution, but also the infeasibility of the final solution may also occur. The proposed robust method could guarantee that, a small deviation of the processing times does not affect the feasibility. The implementation of the proposed method is illustrated using some numerical examples and the outcomes of the investigation are discussed

© 2010 Growing Science Ltd. All rights reserved.

1. Introduction

The job shop-scheduling (JSS) problem normally deals with manufacturing industry where there are many practical applications on this sector. A simple definition of JSS problem can be defined as follows. Given n jobs which have to be processed on m machines, each job consists of a sequence of task operations. Each of which requires to be processed without interruption for a given period of time on a given machine. The sequence of machines for each job is prescribed, and each of them can process at most one job at a time. Tasks of the same job cannot be processed simultaneously & each of them needs to meet each machine only once and the aim is to generate a time schedule to perform the operations on machines. One of the most attractive objectives is to find a schedule that minimizes the makespan, that is, the time needed to complete all the jobs. The JSS problem is classified as one of the most difficult NP-complete problems (Garey, et al., 1976). Presently, there are literally lots of optimization and approximation algorithms such as meta-heuristics and hybrid methods proposed to solve the JSS problem. In addition to the classical JSS problem, with some induced alteration in the basic model, other kinds of JSS problems also appear such as flexible JSS (Kacem, et al., 2002), fuzzy JSS (Allet, 2003), and stochastic JSS (Ginzburg, & Gonik, 2002). The main contribution of this paper is to implement robust optimization technique to handle uncertainty in JSS problem. As the first attempt, Soyster (1973) proposed a model which admits the highest protection. Therefore in his

* Corresponding author. fax: +98-21-77240482

E-mail addresses: A_Jamili@iust.ac.ir (A. Jamili)

case, the objective function is determined in the worst-case condition. Ben-Tal and Nemirovski (2000) presented a more popular robust approach which addressed a more conservative solution where a non-linear robust optimization model is presented. Bertsimas and Sim (2004) introduced a new robust approach in which the robust counterpart is of the same class as the nominal problem and therefore the proposed robust approach remains linear/mixed integer, if the nominal problem is linear/mixed integer.

Consider a linear constraint $\sum_j a_{ij} \times x_{ij} \leq b_i$ and let J_i represent the set of coefficients which are subject to perturbation ($a_{ij}, j \in J_i$). The true value \tilde{a}_{ij} equals $(1 + \varepsilon \xi_{ij}) a_{ij}$ where $\varepsilon > 0$ is a given uncertainty level and ξ_{ij} are random variables distributed symmetrically in the interval $[-1, 1]$. Therefore $\tilde{a}_{ij}, j \in J_i$ takes values in $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$. Bertsimas and Sim (2004) proposed parameter $0 \leq \Gamma_i \leq |J_i|$, to adjust the desired conservatism level of the final solution. It means that at last only $\lfloor \Gamma_i \rfloor$ are allowed to change, and one coefficient a_{i_i} changes by $(\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{i_i}$.

Note that, if $\Gamma_i = |J_i|$ is considered, the method is reduced to Soyster's approach. The idea of using robust optimization has become popular (Farhang Moghadam & Seyed Hosseini, 2010; Gharakhani et al., 2010; Roghanian & Foroughi, 2010). For more details on the implementation of robust optimization, interested readers are referred to (Bertsimas & Sim, 2004). Shafia et al., (2010) adapted the latter approach to train routing and makeup problem.

2. Robust Job Shop Problem

As we explained, the perturbation is normally occurs in JSS problem, especially in the processing times of jobs in each machine. Suppose one is about to schedule the job shop problem introduced by Baker (1974). The tabular representations of the data for this prototype problem are shown in Table 1.a and Table 1.b.

Table 1.a
Processing Times

		Operation		
		1	2	3
Job	1	4	3	2
	2	1	4	4
	3	3	2	3
	4	3	3	1

Table1.b
Routing

		Operation		
		1	2	3
Job	1	1	2	3
	2	2	1	3
	3	3	2	1
	4	2	3	1

Consider minimization of total weighted tardiness (TWT) as the objective function, and let the starting time of schedule for each job, release time, to be equal to 1. The optimal solution for this example is shown as Fig. 1. Furthermore it is noticeable that this solution is also optimal in the case of makespan objective.

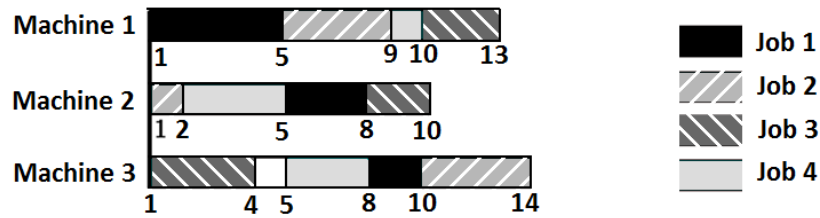


Fig. 1. Optimal Job Shop Schedule

As we can observe from Fig. 1, the optimum schedule value is 10. One can assume that the operation of job 1 in machine 1 is subjected to perturbation and the processing time is increased by one. In other words it changes from 4 to 5. This alteration makes the solution shown in Fig. 1, non-optimal. The perturbed schedule is depicted in Fig. 2.

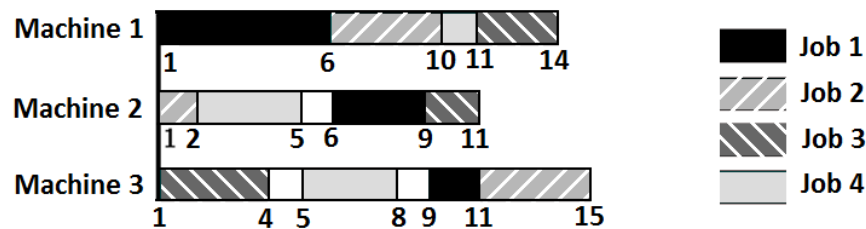


Fig. 2. Perturbed Job Shop Schedule

As it can be found in Fig 2, this interruption worsens the TWT by 3 minutes (10→13). However, for this case the optimum schedule on the basis of TWT is calculated as Fig. 3.

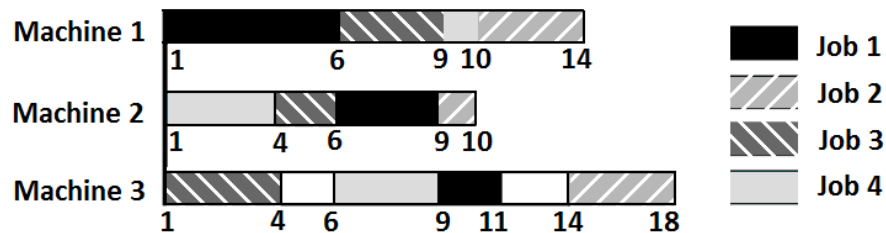


Fig. 3. Optimal Job Shop Schedule-perturbation case

The optimum schedule value in Fig. 3 for the TWT objective function is 10. It can be easily verified that just a little perturbation can highly change the optimum solution and this could be regarded as a good motivation to the robust optimization concept.

In the remaining part of this section, the robust JSS problem is introduced after proposing an appropriate mathematical model for the classical JSS problem. At the first step, the reader can consider the integer programming model specified by Baker. The employed notations are as follows:

Notations

- A: Set of machines, $\{1,2,\dots,m\}$
- P: Set of jobs, $\{1,2,\dots,n\}$
- t_{pjm} : Required time to process task j of job p on machine m
- e_i : The machine which performs the last operation on job i
- M: A big number

a_{pm} : A variable which denotes the completion time of job p on machine m

b_{ipm} : A binary variable which is 1, if job i precedes job p on machine m and 0, otherwise

$$\text{Model 1: } \min \sum_{i \in P} a_{ie_i}$$

subject to:

$$a_{ik} - a_{ih} \geq t_{ijk}, \quad (i, j-1, h) \ll (i, j, k) \quad (1)$$

$$a_{pk} - a_{ik} + M(1 - b_{ipk}) \geq t_{pqk}, \quad i, p \in P, k \in A \quad (2)$$

$$a_{ik} - a_{pk} + M \times b_{ipk} \geq t_{ijk}, \quad i, p \in P, k \in A \quad (3)$$

$$a_{ik} \geq 0, \quad i \in P, k \in A$$

$$b_{ipk} \in \{0, 1\}, \quad i, p \in P, k \in A$$

See (Baker, 1974) for more details.

Since, the processing times, t_{ipm} , is subject to perturbation, the introduced robust method must be applied on inequalities 1, 2 and 3. It can be observed that for all these equations, there is only one perturbed parameter in each constraint. As a result by applying the introduced robust approach to these inequalities, the achieved results are the same as solving the model in the worst-case, where a fix supplementary time is added to all processing times. On the other hand, it is clear that model 1 is only capable to cover the effects of possible perturbation amongst the consecutive jobs. Therefore, in this case, the effects of perturbation propagation amongst jobs are ignored. In order to overcome the mentioned weak characteristics, another formulation is proposed in which the sequencing of jobs on machines is appeared in the model. Following notations are used for the proposed model 2.

Notations

s : Sequence index defined for each job in every machine

O_p : The last operation associated with job p .

x_{jpms} : A binary variable which is 1, if task j of job p is allocated on machine m in sequence s , and 0, otherwise.

y_{jpms} : Denotes start-processing time of task j of job p on machine m

L_{ms} : Denotes the idle time of machine m in sequence s . i.e. the gap between (s) -th and $(s-1)$ -th operations performed on machine m .

$$\text{Model 2: } \min Z = \sum_{p \in P} \sum_{m \in A} \sum_{s=1}^{|P|} (y_{O_p, pms}) \quad (4)$$

subject to:

$$\sum_{s=1}^{|P|} \sum_{m \in A} x_{jpms} = 1, \quad \forall j, p \quad (5)$$

$$\sum_{p \in P} \sum_{j=1}^{O_p} x_{jpms} \leq 1, \quad \forall m, s \quad (6)$$

$$y_{jpms} \leq M \times x_{jpms}, \quad \forall j, p, m, s \quad (7)$$

$$\sum_{s=1}^{|P|} \sum_{m \in A} (y_{jpms} + x_{jpms} \times t_{ipm}) \leq \sum_{s=1}^{|P|} \sum_{m \in A} y_{(j+1)pms}, \quad \forall j, p \quad (8)$$

$$\sum_{p \in P} \sum_{j=1}^{O_p} (y_{jpms} + x_{jpms} \times t_{ipm}) \leq \sum_{p \in P} \sum_{j=1}^{O_p} y_{jpms+1}, \quad \forall m, s \quad (9)$$

The objective function shown in equation 4 minimizes the sum of finishing time of the last operation of each job. Eq. 5 indicates that task j of job p must be assigned to just one sequence of one machine. Inequality 6 guarantees that only one task of each job can be executed in sequence s of machine m . Inequality 7 establishes the relation of x_{jpms} and y_{jpms} variables. In other word, y_{jpms} must be zero if the binary variable x_{jpms} is equal to zero. Inequality 8 guarantees the task sequences for each job. Inequality 9 guarantees that each task's processing time does not have any overlap with any other task.

The desired job sequencing on machines appears using the new presented problem formulation. As explained previously, in order to have a robust job shop schedule, the possible effects of all operations on each other must be considered. For this purpose, we suggest to apply constraints 10 and 11, instead of constraint 9.

$$\sum_{s''=s'}^{s-1} \sum_{j=1}^{O_p} \sum_{p \in P} (x_{jpms''} \times t_{jpm}) + \sum_{s''=s'}^{s-1} L_{ms''} \leq \sum_{p \in P} \sum_{j=1}^{O_p} y_{jpms} \quad \forall m, s, s' < s \quad (10)$$

$$L_{ms} = \sum_{p \in P} \sum_{j=1}^{O_p} y_{jpms} - \sum_{p \in P} \sum_{j=1}^{O_p} (y_{jpms-1} + x_{jpms-1} \times t_{jpms-1}) \quad \forall j, p, m, s \quad (11)$$

Applying the robust approach to constraint 10, leads to a situation in which there will be a confident gap amongst jobs in a specified machine, but there would be no such gaps amongst the operations associated with a specific job. In other words in order to have a robust job shop schedule, not only we need to consider the possible interruptions in each machine, but also it requires to embed some buffer times amongst the operations corresponds to each job, so that the possible delay in one machine do not affects the other machines, as well. For this reason, the inequality 12 is proposed instead of inequality 8.

$$\sum_{s=1}^{|P|} \sum_{m \in A} y_{j'pms} + \sum_{j''=j'}^{j-1} \sum_{m \in A} \sum_{s=1}^{|P|} (x_{j''pms} \times t_{j''pm}) \leq \sum_{s=1}^{|P|} \sum_{m \in A} y_{jpms} \quad \forall j, p, j' < j \quad (12)$$

Both inequalities 10 and 12 consist of some surplus constraints compared with inequalities 8 and 9. To illustrate the case, one could consider job 1 in figure 1. Inequality 12 represents three constraints for job 1 in this specific instant as follows:

- 1- Second task of job 1 must be initiated after the first task.
- 2- Third task of job 1 must be started after the second one.
- 3- Third task of job 1 must be initiated after the finishing time of the first task added with the processing time of the second task of job 1.

It is obvious that item 3 will be surplus under the condition that items 1 and 2 are fulfilled. It can be easily shown that the number of constraints resulted by equation 12 for the specific job p that contains O_p individual tasks will be equal to:

$$N_p^{O_p} = \binom{O_p}{2} + \binom{O_p}{3} + \dots + \binom{O_p}{O_p} = 2^n - \binom{O_p}{0} - \binom{O_p}{1} = 2^n - O_p - 1$$

As inequality 8, represents $\binom{O_p}{2}$ constraints, the number of surplus constraints is equal to:

$$S_p^{O_p} = 2^n - \binom{O_p}{0} - \binom{O_p}{1} - \binom{O_p}{2} = 2^n - \frac{O_p \times (O_p - 1)}{2} - O_p - 1 = 2^n - \frac{O_p^2}{2} - 0.5 \times O_p - 1$$

The number of surplus constraints associated with inequality 10 can be also computed in a similar way. The $S_p^{O_p}$ number of extra constraints, in addition to those resulted by inequality 10, are defined to formulate the robust JSS problem. The Bertsimas and Sim approach is briefly reviewed as follows.

Consider the constraint $\sum_{j=1}^n a_{ij}x_j \leq b_i$ and suppose that H_i represents the set of coefficients defined in constraint i which are subject to perturbation (\tilde{a}_{ij} , $j \in H_i$). It is assumed that each uncertain coefficient \tilde{a}_{ij} , $j \in H_i$ belongs to the interval $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$.

It is supposed that at last only $\lfloor \Gamma_i \rfloor$ of parameters subjected to disruptions will be interrupted in constraint i , and one coefficient a_{ik_i} alters by $(\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{ik_i}$. In the Bertsimas and Sim approach, the robust formulation of the specified constraint is formulated as follows,

$$\sum_j a_{ij}x_j + \max_{\{K_i \cup \{k_i\} | K_i \subseteq H_i, |K_i| = \lfloor \Gamma_i \rfloor, k_i \in H_i, k_i \notin K_i\}} \left\{ \sum_{j \in K_i} \hat{a}_{ij} |x_j| + (\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{ik_i} |x_{k_i}| \right\} \leq b_i, \quad (13)$$

where, K_i is a subset which has $\lfloor \Gamma_i \rfloor$ parameters subjected to perturbation. The added statement to the left hand-side of constraint 13, is called protection function.

In order to reformulate the proposed model as a robust job shop one, at first the protection functions must be added to the left side of the equations 10 and 12. Based on inequality 13, the protection function for inequality 12 is proposed as Eq. 14. The other one could be generated as well.

$$pf = \max_{\left\{ \begin{array}{l} K_{jj'p} \cup \{k_{jj'p}\} | K_{jj'p} \subseteq H_{jj'p}, |K_{jj'p}| = \lfloor \Gamma_{jj'p} \rfloor \\ k_{jj'p} \in H_{jj'p}, k_{jj'p} \notin K_{jj'p} \end{array} \right\}} \left\{ \sum_{j''=j'}^{j-1} \sum_{m \in A} \left(\hat{t}_{j''pm} \times \sum_{s=1}^{|P|} x_{j''pms} \right) + (\Gamma_{jj'p} - \lfloor \Gamma_{jj'p} \rfloor) \times \sum_{m \in A} \left(\hat{t}_{k_{jj'p}pm} \times \sum_{s=1}^{|P|} x_{k_{jj'p}pms} \right) \right\} \cdot \forall j, p, j' < j \quad (14)$$

The above equation equals the objective function of the following linear optimization problem. For the details the interested reader may refer to (Bertsimas, & Sim, 2004).

$$\text{Model 3: } \max \sum_{j''=j'}^{j-1} \sum_{m \in A} \left[\left(\hat{t}_{j''pm} \times \sum_{s=1}^{|P|} x_{j''pms} \right) \times z_{j''pm} \right]$$

subject to:

$$\sum_{j''=j'}^{j-1} \sum_{m \in A} z_{j''pm} \leq \Gamma_{jj'p},$$

$$0 \leq \sum_{m=1}^M z_{j''pm} \leq 1. \quad \forall j' \leq j'' \leq j-1$$

The dual of model 3 is as follows:

$$\text{Model 4: } \min \sum_{j''=j'}^{j-1} P_{jj''p} + \Gamma_{jj'p} \times z_{jj'p}$$

subject to:

$$z_{jj'p} + P_{jj'p} \geq \sum_{m=1}^M t_{j'pm} \times \left(\sum_{s=1}^S x_{j'pms} \right), \quad \forall j' \leq j'' < j$$

$$P_{jj'p} \geq 0, \quad \forall j' \leq j'' < j$$

$$z_{jj'p} \geq 0.$$

Note that the above problem must be defined as $\forall j, p, (j' < j)$.

Substituting the optimization function of the model 4 with the proposed protection function, equation 14, and embedding the remaining constraints of model 4 to the main body of the model 2 will lead to the Robust JSS Problem. Note that, the robust approach must also be applied to inequality 10. The procedure is similar to that of inequality 12 and therefore is not repeated here.

The value of Γ level

As it is clarified, the role of parameter Γ is to adjust the robustness of the proposed model against the level of conservatism of the solution. The proposed $\Gamma_{jj'p}$ level specified for the model 4 is as follows:

$$\Gamma_{jj'p} = (j - j' - \alpha_p) \times \beta_p \quad 0 \leq \alpha \leq 1 \ \& \ 0 < \beta \leq 1 \quad (15)$$

In Eq. 15, the value $j - j'$ is equal to the number of parameters which are subjected to the perturbation. Parameters α_p and β_p are used to adjust the conservatism level. The parameters determine the trade-off between the desired robustness and the optimality. In the case $\alpha_p = 0$ and $\beta_p = 1$ the level $\Gamma_{jj'p}$ makes the problem to work in the worst condition exactly the same as Soyster's model.

Note that by applying the robust approach to inequality 10, a similar level to Eq. 15 would be resulted: $\Gamma_{ss'm} = (s - s' - \lambda_m) \times \gamma_m$. The role of parameters λ_m and γ_m are alike α_p, β_p . To illustrate the effects of the proposed robust JSS mathematical model, one can consider the Baker's example which was investigated earlier in this section. It is assumed that the maximum deviation for each processing time from the nominal value, \hat{t} , is equal to 10%. In other word, the true value, \tilde{t}_{ipm} , takes values in $[0.9 \times t_{ipm}, 1.1 \times t_{ipm}]$. Furthermore suppose that: $\alpha_p = \lambda_m = 0.5, \beta_p = \gamma_m = 0.8, \forall p, m$.

The robust optimum solution for this example is shown in Fig. 4.

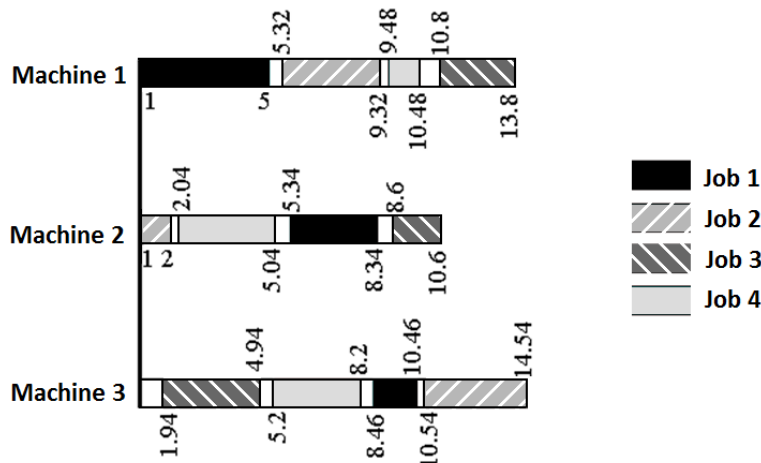


Fig. 4. Optimal Robust Job Shop Schedule

The optimal value of the objective function is equal to 13.56. Through applying this method, it is not only guaranteed that the generated schedule is robust against any perturbation less than the defined

robustness level, but also it remains robust under the condition of heavier disruptions with a high probability. For more details see Bertsimas & Sim, 2004.

3. Proposed heuristic algorithm

Job shop scheduling problem is known NP-Hard in its classical type. Considering the number of used binary variables involved in a real-world problem formulation, we need to look for a near-optimal solution as an alternative using a heuristic approach. In the remaining part of this section, a heuristic algorithm based on the simulating annealing (SA) algorithm is proposed. SA is known to be one of the most successful algorithms amongst the other applied meta-heuristic ones and its acceptable performance is demonstrated by experience in the conducted researches relevant to scheduling problems, for example (Krishna, 1995), (Steinhofel, 1999), (Suresh, & Mohanasundaram, 2006), (Ponnambalam, 1999), (Bozejko, 2009).

The following algorithm is proposed to find a near-optimum schedule for the introduced robust JSS problem.

Sub *Simulated annealing*

$k \leftarrow 1$

Call Initial schedule generation

$s_{best} \leftarrow s$

While $k < K$ **do**

Call neighborhood Schedule generation (s)

If $f(s') \leq f(s)$ **or** $\text{random}[0, 1] < P_{accept}(s, s', T_k)$ **then**

$s \leftarrow s'$.

Endif

If $f(s') < f(s_{best})$ **then**

$s_{best} \leftarrow s'$

Endif

$k \leftarrow k + 1$

$T_k = \alpha \times T_{k-1}$

Endwhile

Endsub

Sub *Initial schedule generation*

Do

 Select one of the non-scheduled operations that its all predecessors is already scheduled, randomly.

 Compute the amount of protection function based on the desired robustness level, i.e. the required buffer times.

 Compute the start and the completion time of the selected operation considering other scheduled operations in addition to the required buffer time.

 If all operations are scheduled then **Exit Do**.

Loop

Endsub

Sub neighborhood Schedule generation (s)

Select one of the operations which are delayed in schedule s , e.g. op_i .

Find the operation which causes the delay of op_i in schedule s , e.g. op_j .

Generate the schedule s' , by this new assumption that op_i must be scheduled earlier than op_j .

Note that all other operations must be scheduled in the same order of schedule s .

Endsub

In this part, s is a schedule, s_{best} is the best available solution, $f(s)$ is the objective function value for schedule s , k is the counter, K is the maximum iteration number which specifies the termination criteria, T_k is the temperature in k -th iteration, α is the cooling factor which belongs to the interval (0, 1), and $P_{accept}(s, s', T_k)$ is the probability function to accept non-improving solution s' , Eq. 16.

$$P_{accept}(s, s', T_k) = \begin{cases} 1 & \text{if } f(s') < f(s) \\ \exp\left(\frac{f(s) - f(s')}{T_k}\right) & \text{Otherwise} \end{cases} \quad (16)$$

Parameter tuning

In order to use the best combination of parameters defined in the proposed SA algorithm in this section, a full factorial design of experiment (DOE) approach is applied. As it is explained in the proposed algorithm, two parameters need to be tuned. These include T_0 and α . In Table 2, three levels for the parameters are considered, and therefore a 3^2 design is applied. The experiments contain 15 different instances randomly generated and solved by assuming each of 9 different combinations of (T_0, α) .

Table 2

Three levels of SA parameters

	Level 1	Level 2	Level 3
α	0.97	0.98	0.99
T_0	50	100	150

In order to compare the instances, the relative deviation index (RDI) is used. This index is obtained by

$$RDI_k = \frac{F_k - Min_k}{Max_k - Min_k} \times 100, \quad (17)$$

where, F_k is the objective function value in the k -th instance. Min_k and Max_k are the best and the worst solutions obtained for each instance, respectively.

We have solved 15 different instances, each of which is associated with one of the combinations of (T_0, α) .

The results are analyzed by the means of analysis of variance (ANOVA) technique. The performed tests for analyzing the normality, homogeneity of variance and independence of residuals do not show any particular pattern in the experiments. Fig. 5 depicts the interaction plot for parameters T_0 and α . The combination of $T_0=150$ and $\alpha=0.98$ are calculated and produce better results compared to the other evaluated combinations.

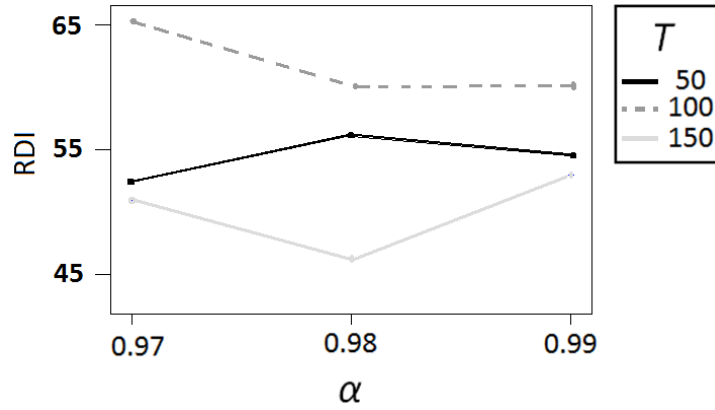


Fig 5. The achieved results for tuning the parameters of SA

The proposed SA algorithm is implemented in VB on a Laptop with Pentium IV Core 2 Duo 2.53 GHz CPU. The outputs of the algorithm are compared with optimum solutions achieved by Branch and Bound algorithm (B&B).

Each instance can be characterized by a number of parameters, such as number of jobs, number of machines, job routes, processing times, and the desired robustness parameters. The generated instances are based on the following assumptions:

- Processing times are all integer numbers between the interval [10, 20] generated at random.
- The objective is to minimize the total completion times of all jobs.
- The robustness parameters are set as follows: $\alpha_p = \lambda_m = 0.5$, $\beta_p = \gamma_m = 0.8$, $\forall p, m$.

The outputs are shown in Table 3.

Table 3

The comparison results between the proposed hybrid algorithm and B&B algorithm

Item	Number of jobs	Number of Machines	B&B algorithm		SA algorithm	
			TWT	Duration	TWT	Duration
1	4	3	116	00:00:01	116	00:00:00
2	4	4	181	00:00:02	181	00:00:00
3	4	5	247	00:00:09	249	00:00:01
4	6	6	448	00:03:59	460	00:00:01
5	6	6	418	00:03:15	422	00:00:01
6	6	7	496	00:06:37	501	00:00:01
7	6	7	487	00:09:53	488	00:00:05
8	6	8	553	00:05:31	553	00:00:05
9	6	8	598	00:08:49	598	00:00:11
10	6	9	631	00:47:23	662	00:00:40

The results show that the SA algorithm is capable to find near optimum solutions in a considerably reduced amount of time in comparison with the B&B algorithm. Moreover, the sensitivity of the objective function to the alteration of interruption values is experienced for a randomly generated instance with 8 jobs and 10 machines. Fig. 6 shows the best found solutions found by SA algorithm for the different cases of $\tilde{t} \in [(1-x)t, (1+x)t]$.

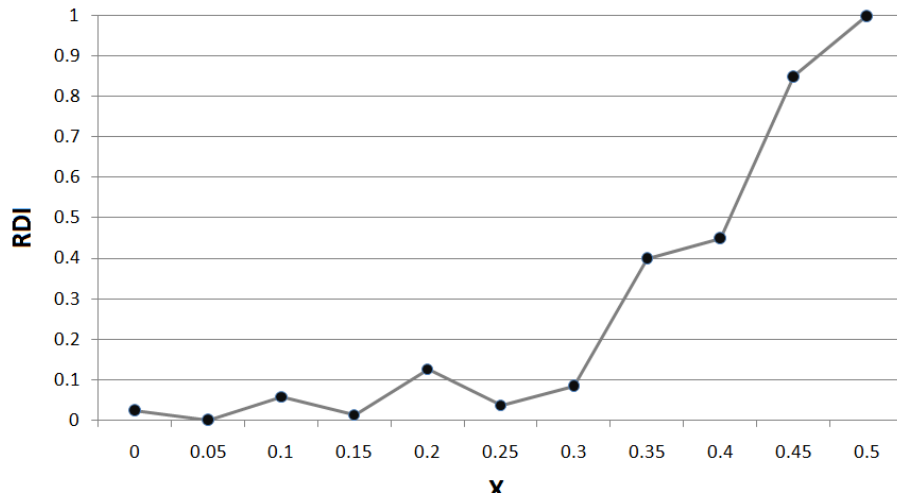


Fig 6. The achieved results for different interruption intervals

Fig. 6 shows that as the interruption intervals increases, in order to reach the same level of robustness, the TWT of the solutions are worsen.

4. Conclusion

We have presented a robust JSS problem which has the capability of handling the perturbation which exists amongst almost all input parameters. It is illustrated that, for many real world job scheduling problems, a small change in input parameters may lead us to some final solutions which are neither feasible nor optimal. The proposed method of this paper could guarantee that a small alteration in input parameters does not have any effect in the feasibility and the optimality of the robust one and does not alter from the crisp one. Besides, it is known that the common software packages are not able to find optimum solutions for real world application in a reasonable amount of time. Therefore, a SA algorithm has been proposed to find near optimal solutions in a reasonable amount of time. The effectiveness of the algorithm has been demonstrated, using some numerical examples and the results of SA have been compared with optimal solutions.

Acknowledgment

The author would like to thank the anonymous referees for their constructive comments on earlier version of this work.

References

- Allet, S. (2003). Handling flexibility in a generalized job shop with a fuzzy approach. *European Journal of Operational Research*, 147, 312–333.
- Baker, K. R. (1974). Introduction to sequencing and scheduling. New York: Wiley.
- Ben-Tal, A., & Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88, 411–424.
- Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research*, 52, 35–53.
- Bozejko, W., Pempera, J., & Smutnicki, C. (2009). Parallel Simulated Annealing for the Job Shop Scheduling Problem. In Proceedings of the 9th International Conference on Computational Science: Part I, 631 – 640.
- Farhang Moghadam, B. & Seyedhosseini, S. M. (2010). A particle swarm approach to solve vehicle routing problem with uncertain demand: A drug distribution case study, *International Journal of Industrial Engineering Computations*, 1(1), 55–64.

- Gharakhani, M., Taghipour, T. & Jalali Farahani, K. (2010). A robust multi-objective production planning, *International Journal of Industrial Engineering Computations*, 1(1), 73-78.
- Garey, E.L., Johnson, D.S., & Sethi, R. (1976). The complexity of flowshop and job-shop scheduling. *Mathematics of Operations Research*, 1, 117-129.
- Ginzburg, D.G., & Gonik, A. (2002). Optimal job-shop scheduling with random operations and cost objectives. *International Journal of Production Economics*, 76, 147-157.
- Kacem, I., Hammadi, S., & Borne, P. (2002). Pareto-optimality Approach for Flexible Job-shop Scheduling Problems: Hybridization of Evolutionary Algorithms and Fuzzy Logic. *Journal of Mathematics and Computers in Simulation*, 60, 245-276.
- Krishna, K., Ganeshan, K., & Janaki Ram D. (1995). Distributed Simulated Annealing Algorithms for Job Shop Scheduling, *IEEE Transactions on systems. MAN. & Cybernetics*, 25 (7), 1102-1109.
- Ponnambalam, S. G., Jawahar, N., & Aravindan, P. (1999). A simulated annealing algorithm for job shop scheduling. *Production Planning & Control*, 10 (8), 767-777.
- Shafia, M. A., Pourseyed Aghae, M., Sadjadi, S. J., & Jamili, A. (2010). Robust Train Timetabling problem: Mathematical model and Branch and bound algorithm. Working paper
- Shafia, M. A., Sadjadi, S. J., & Jamili, A. (2010). Robust Train Formation Planning. *Proceedings of the IMechE Part F: Journal of Rail and rapid transit*, 224(F2), 75-90.
- Roghanian, E. & Foroughi, A. (2010). An empirical study of Iranian regional airports using robust data envelopment analysis, *International Journal of Industrial Engineering Computations*, 1(1), 65-72.
- Soyster, A.L. (1973). Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21, 1154-1157.
- Steinhofel, K., Albrecht, A., & Wong, C. K. (1999). Two simulated annealing-based heuristics for the job shop scheduling problem. *European Journal of Operational Research*, 118 (3) 524-548.
- Suresh, R.K. , & Mohanasundaram, K.M. (2006). Pareto archived simulated annealing for job shop scheduling with multiple objectives. *The International Journal of Advanced Manufacturing Technology*, 29, 184-196.