Journal of Quality Engineering and Production Optimization

Vol. 3, No. 1, PP. 13-26, 2018

An Improved Tabu Search Algorithm for Job Shop Scheduling Problem Trough Hybrid Solution Representations

Parviz Fattahi*[1], Masume Messi Bidgoli[2] and Parvaneh Samouei[3]

1. Associate professor, Department of Industrial Engineering, Faculty of Engineering, Alzahra University, Tehran, Iran

2. PhD candidate, Department of Industrial Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran

3. Assistant Professor, Department of Industrial Engineering, Faculty of Engineering, Bu-Ali Sina University, Hamedan, Iran

[*]***Corresponding Author:*** *ParvizFattahi (E-mail: p.fattahi@alzahra.ac.ir)*

Abstract- *Job shop scheduling problem (JSP) is an attractive field for researchers and production managers since it is a famous problem in many industries and a complex problem for researchers. Due to NP-hardness property of this problem, many meta-heuristics are developed to solve it. Solution representation (solution seed) is an important element for any meta-heuristic algorithm. Therefore, many researchers try to present different encodings to solve this problem. Fattahi et al., and Gen & Cheng suggested two solutions for this problem that both have advantages and weaknesses in searching solution space to reach an acceptable solution. In the current paper, a cyclic algorithm based on tabu search algorithm was proposed to improve the exploration and exploitation powers of these encodings. Also, several problems of different sizes are solved by it and the obtained results were compared. Results showed the applicability and effectiveness of the proposed solution representation in comparison with the existing ones.*

Keywords: *Job Shop Scheduling Problem, Solution Representation, Tabu Search algorithm*

I. INTRODUCTION

One of the most important issues for managers, engineers, and researchers in production field is suitable scheduling. The goal of scheduling is to allocate jobs to machines that optimize some objectives. Due to diverse applications and importance of Job shop scheduling problem (JSP), many researchers developed and used methods to solve it. There are several good review papers such as those of Mellor and Jain & Meeran. This problem is NP-hard and its optimal solution is difficult to obtain even if the objective is to minimize the completion time of the job finished last. Jamili presented mathematical models, exact and heuristic algorithms for robust job shop scheduling problem. Ku & Beck had a computational analysis on mixed integer programming models for job shop scheduling. Since JSP is one of the hardest combinatorial optimization problems (Garey et al.), meta-heuristic algorithms are useful and attractive alternatives to solve it. Therefore, many meta-heuristic algorithms are used for the job shop problem by researchers. In the recent years, the applications of these algorithms for JSP are increased. Hence, it seems necessary to offer the good structures for these algorithms to obtain better solutions for this problem. Application of the suitable representation for solutions is one of the basic features of these algorithms in solving job shop scheduling problems to gain better results. As already known, the efficiency of these algorithms is essentially related to the search operators (e g, neighbourhood in SA, and TS, recombination in GA, etc.) and the evaluation function could be used to measure the applicability and efficiency of the algorithms.

As already known, many alternative representations are usually possible for a specific problem. But among them, it is necessary to use an encoding that is suitable and relevant to the studied problem. These conditions create some

Manuscript Received:02 -Feb-2016 and revised: 25-Oct-2016 *ISSN: 2423-3781*
Accepted: 12-Feb-2017

challenges to design the most efficient one. The necessary characteristics for an appropriate solution representation are as follows:

(1) Completeness: All feasible solutions associated with the related problem could be represented by this encoding.

(2) Connectivity: A search path exists between any two solutions in the search space. This implies that any solution (including the/a global optimum) could be reached no matter the initial solution.

(3) Efficiency: The representation can be efficiently manipulated by the search operators. The complexity (time and space) of the operators (e g, calculating the neighbourhood and the size of the neighbourhood/search space) should be reduced.

(4) Redundancy of the representation: It means that two encodings should not map towards the same solution. In other words, a one to one relationship between the real solution space and the coded space is intended.

(5) Based on these descriptions, some various solution representations are proposed by the researchers in the JSP literature. Some of the most well-known representations in JSP are operation-based, job-based, preference list-based, job pair relation-based, priority rule-based, disjunctive graph-based, completion time-based, machine-based, random keys, priority matrix, and that of Fattahi et al. Among all of these representations, two encodings proposed by Fattahi et al., (priority matrix) and Gen & Cheng (operation-based) were selected.

(6) By considering the fact that both of these encodings have some advantages and weaknesses in searching the solution space, in the current paper an efficient algorithm was proposed that tried to improve the quality of final solution using their advantages and overcoming their weaknesses. Therefore, the main contribution of the paper is proposing a hybrid solution representation for job shop scheduling problems, and using a tabu search algorithm as a meta-heuristic algorithm to show the effects of using the hybrid solution representation on the quality of solutions.

(7) The remaining sections of the current paper are organized as follows. In section 2, the problem description, mathematical model and both solution representations are described. Section 3 is dedicated to a review on existing meta-heuristic solution methods for job shop problem. Some improved tabu search algorithms based on the suggested structure are presented in section 4. Numerical experiments and their results are discussed in section 5. Finally, section 6 presents the conclusions and future researches.

II. PROBLEM DESCRIPTION AND TWO SOLUTION REPRESENTATIONS

In this section, at first, the mathematical model of JSP is provided and then, the representations of priority matrix and operation-based are described.

A. MATHEMATICAL MODELING

In the classical JSP, there are $"m"$ different machines and $"n"$ different jobs to be scheduled and each job has a set of operations that should be conducted on different machines. Therefore, it is necessary to find the best sequence of operations on each machine to meet the following constraints;

1. A job does not visit the same machines twice;
2. There are no precedence constraints between operations of different jobs;
3. Each operation cannot be interrupted;
4. Each machine can process only one job at a time;
5. The processing time is fixed and neither release times nor due dates are specified.

The parameters of JSP mathematical model are given below:

$Cmax$: Completion time (for all jobs),

m: The number of machines,

n: The number of jobs,

$O_{j,h}$: The operation h of the job j.

$t_{j,h}$: Start time of the processing of operation $O_{j,h}$.

$p_{j,h}$: Processing time of operation $O_{j,h}$.

$x_{i,j,h,k}$: If $O_{j,h}$ is performed on machine i in priority k is equal to 1 else it is 0.

$Tm_{i,k}$: Start of working time for machine i in priority k.

L: A positive large number.

$a_{i,j,h}$: Describe the capable machines set $M_{j,h}$ is assigned to operation $O_{j,h}$. It is 1 if $O_{j,h}$ can be performed on machine i, else it is 0.

The objective of the mathematical model shows minimization of the required time to complete all jobs.

$min\ C_{max}$

$s.t.$

$$C_{max} \geq t_{j,h_j} + p_{j,h_j} \qquad j = 1, \dots, n \tag{1}$$

$$t_{j,h} + p_{j,h} \leq t_{j,h+1} \qquad j = 1, \dots, n; h = 1, .., h_j - 1 \tag{2}$$

$$Tm_{i,k} + p_{j,h} x_{i,j,h,k} \leq Tm_{i,k+1} \qquad i = 1, \dots, m; j = 1, \dots, n; h = 1, .., h_j - 1; k = 1, \dots, k_i - 1 \tag{3}$$

$$Tm_{i,k} \leq t_{j,h} + \left(1 - x_{i,j,h,k}\right).L \qquad i = 1, \dots, m; j = 1, \dots, n; h = 1, .., h_j; k = 1, \dots, k_i \tag{4}$$

$$Tm_{i,k} + \left(1 - x_{i,j,h,k}\right).L \geq t_{j,h} \qquad i = 1, \dots, m; j = 1, \dots, n; h = 1, .., h_j; k = 1, \dots, k_i \tag{5}$$

$$\sum_j \sum_h x_{i,j,h,k} = 1 \qquad i = 1, \dots, m; \quad k = 1, \dots, k_i \tag{6}$$

$$\sum_k x_{i,j,h,k} = a_{i,j,h} \qquad i = 1, \dots, m; j = 1, \dots, n; h = 1, .., h_j \tag{7}$$

$$t_{j,h} \geq 0 \qquad j = 1, \dots, n; h = 1, .., h_j \tag{8}$$

In this model, constraint set (1) shows the completion time of all jobs. Constraint set (2) directs each job to follow a specified operation sequence. Constraint (3) shows each machine to process one operation at a time. Constraint sets (4) and (5) direct each operation $O_{j,h}$ that can start after its assigned machine is idle and previous operation $O_{j,h-1}$ is completed. Constraint set (6) forces each operation to be performed only on one machine, Constraint set (7) determines the machine for each operation, and finally, Constraint set (8) shows that processing time cannot be negative.

B. PRIORITY MATRIX REPRESENTATION

The priority matrix representation presented by Fattahi et al., used a matrix representation with four rows that show the machine number, the job number, the operation number, and the priority number, respectively. Also, this matrix has "v" columns, which "v" equals to the sum of the number of operations for all the jobs. The structure of this matrix is shown in Fig (1).

An example of this representation is presented in Fig (2). It shows that there are three machines and jobs. Also there are three operations for each job. The last row of this matrix shows that the first operation of job 1 should be performed on machine 1 in priority 3. On the other hand, machine1 should do the second operation of job 2, the third operation of job 3 and the first operation of job 1, respectively.

The privilege of this approach is that there is a one-to-one relationship between the real solution space and the coded space and the weakness of this approach is that this approach does not guarantee the feasibility of the solutions (Fattahi et al.). Infeasibility in this representation occurred when two numbers in the fourth row were replaced and this resulted in a solution that the precedence constraints were not met for this new solution. Therefore, this representation may lead to search a wide area of the infeasible part of solution space. By this description, this representation causes the capability of exploration in meta-heuristic algorithms be more than its exploitation.

$$\begin{array}{cc} & 1,\ldots,v \\ \begin{array}{c}1\\2\\3\\4\end{array} & \begin{bmatrix}\text{Machine number} \\ \text{Job number} \\ \text{Operation number} \\ \text{Priority}\end{bmatrix}\end{array}$$

Fig 1. Priority Matrix Representation

$$\begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 \\ 1 & 2 & 3 & 3 & 1 & 2 & 2 & 3 & 1 \\ 3 & 1 & 2 & 2 & 1 & 3 & 3 & 1 & 2 \end{bmatrix}$$

Fig 2. An Example of the Representation

TABLE I. A Comparison Between the Real and the Coded Space for the Operation-based Representation

Problem No.	No. of jobs	No. of operations	No. of machines	No. of real solutions	No. of coding solutions
1	3	9	3	216	1680
2	5	15	3	1.7×10^{6}	1.6×10^{8}
3	7	28	4	6.4×10^{14}	6.6×10^{19}
4	10	50	5	6.2×10^{32}	4.4×10^{43}

C. OPERATION-BASED REPRESENTATION

Another representation used in the current article was presented by Gen & Cheng. In this representation, all operations of a certain job had the same name and interpreted according to the order of occurrence in the sequence for a given representation. If "m" indicates the number of operations for a special job, the name of this job appears "m" times in the representation (Fattahi et al.). For example, [1 2 1 3 2 3 1 3] is an operation-based representation. In this case, job1 has three operations and "1" was repeated three times in this representation. Also, it shows that the first operation of job1 should be done first, then the first operation of job2, the second operation of job1, the first operation of job 3 and so on.

In this situation, any permutation in operation-based representation is feasible. But unfortunately, there is a one-to-many relationship between the real solution space and the coding space. It means that a real solution is introduced by several representations. It causes this approach be an incompetent method for large-sized problems (Fattahi et al.). By this encoding, on one hand, the exploration power of the algorithm through the solution space decreased intensively but on the other hand, the exploitation of the algorithm was improved. Table I shows the differences between the number of real solutions and the number of coding solutions.

In the current paper, both operation-based and priority matrix representations were used to solve an example with tabu search. The best obtained solutions from each iteration of tabu search algorithm under the two mentioned encodings are displayed in fig (3) and 4, respectively.

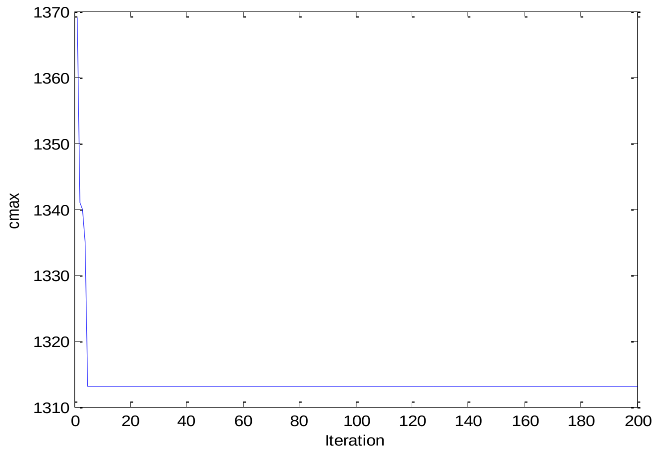

Fig 3. The best C_{max} Diagram Based on Operation-based Representation

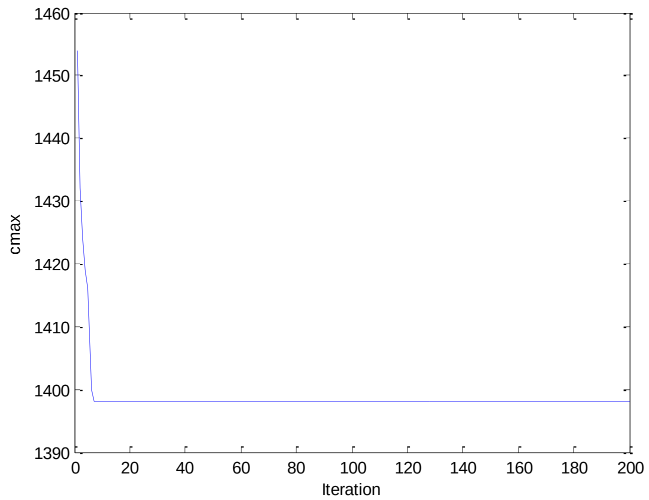

Fig 4. The Best C_{max} Diagram Based on Priority Matrix Representation

As observed in fig (3) and 4, two representations could obtain their final solutions in the first ten iterations and the algorithm could not improve them in the next iterations. Due to this weakness of both representations, reaching a better solution is not promising by these encodings.

In section 4, it was tried to combine these representations to use their advantages. For this purpose, a tabu search algorithm was presented that used these encodings in a cyclic manner to shock the algorithm in the iterations in which no more improvement occurred in the final solution and this matter resulted in enriching the exploration ability of the algorithm through the solution space. A review on meta-heuristic solution methods are presented in the next section and the reason why tabu search is selected for solving the proposed model are expressed.

III. A REVIEW ON META-HEURISTIC SOLUTION METHODS FOR JOB SHOP PROBLEM

(1) Simulated annealing: Elmi et al., used simulated annealing to minimize the make-span for job shop problems, which considered intercellular moves and non-consecutive multiple processing of parts on a machine.

(2) Ant colony optimization: Chang et al., studied a multi-stage job-shop parallel-machine-scheduling problem with an ant colony optimization system developed. Their paper also addressed the multiple-objectives scheduling in which in addition to the production (or quantitative) objectives, the marketing (strategic or qualitative) criteria were also considered. Rossi and Boschi presented a hybrid of genetic and ant colony optimization to solve the flexible manufacturing systems (FMS) scheduling in a job-shop environment with routing flexibility, where the assignment of operations to identical parallel machines had to be managed, in addition to the traditional sequencing problem.

(3) Genetic algorithm: Wang & Tang proposed an improved adaptive genetic algorithm for job shop problem inspired from hormone modulation mechanism, and then the adaptive crossover probability and adaptive mutation probability were designed. De Giovanni & Pezzella proposed an improved genetic algorithm to solve distributed and flexible job-shop scheduling problem. With respect to the solution representation for non-distributed job-shop scheduling, gene encoding was extended to include information on job-to-FMU assignment, and a greedy decoding procedure exploited flexibility and determined the job routings. Besides traditional crossover and mutation operators, a new local search based operator was used to improve available solutions by refining the most promising individuals of each generation. Zhao et al., proposed a shuffled complex evolution algorithm with sequence mapping mechanism for job shop scheduling problems to minimize the make-span. Rabiee et al., presented a non-dominated sorting genetic algorithm for a partial flexible job shop with the objective of minimizing make-span and minimizing total operation costs. Also, Asadzadeh presented an agent-based local search genetic algorithm to solve the job shop scheduling problem with intelligent agents. Furthermore, Kurdi presented a hybrid island model genetic algorithm to solve the job shop scheduling problem with the objective of make-span minimization. Recently, Kuhpfahl & Bierwirth considered the job shop scheduling problem with total weighted tardiness objective and concentrated on the neighbourhood search techniques.

(4) Particle swarm optimization: Moslehi & Mahnam applied multi-objective particle swarm algorithm to the flexible job-shop scheduling problem based on priority. Then, Gao et al., proposed a hybrid particle swarm optimization algorithm based on variable neighbourhood search to solve the job shop scheduling problem. In order to overcome the blind selection of neighbourhood structures during the hybrid algorithm design, they used a neighbourhood structure evaluation method based on logistic model to guide the neighbourhood structures selection.

(5) Artificial immune algorithm: Bagheri et al., developed an artificial immune algorithm for the flexible job-shop scheduling problem that used several strategies to generate the initial population and select the individuals for reproduction. Different mutation operators were also utilized to reproduce new individuals.

(6) Tabu Search: Bülbül studied the job shop scheduling problem to minimize the total weighted tardiness and proposed a hybrid shifting bottleneck-tabu search (SB-TS) algorithm by replacing the re-optimization step in the shifting bottleneck (SB) algorithm by a tabu search (TS). Brandimarte proposed a hybrid algorithm using some dispatching rules and the tabu search algorithm. Saidi-mehrabad and Fattahi proposed a mathematical model and tabu search algorithm with two heuristics to solve the problem with sequence-dependent setups. Fattahi et al., presented a mathematical model to obtain optimal solutions for small-sized problems and six different hybrid searching structures for realistically sized problems. It can be concluded from the above literature that the hierarchical algorithms had better performance than integrated algorithms, and hybrid algorithms combining tabu search and simulated annealing heuristics were more efficient than the other algorithms. Hurink et al., proposed a tabu search heuristic in which reassignment and rescheduling were considered as two different types of moves. The integrated approach represented by Dauzere-Peres & Paulli was defined as a neighbourhood structure for the problem where there was no distinction between reassigning and resequencing an operation. Mastrololli & Gambardella (2002) improved tabu search techniques by Dauzere-Peres & presented two neighbourhood functions.

These heuristic procedures are capable of providing high-quality solutions with reasonable computational effort. Among these heuristics, the tabu search algorithms are more often used for the job shop and flexible job shop scheduling problems. The historical dominance of tabu search algorithms for the JSP, e g, as documented by Bla˙zewicz et al., and Jain & Meeran seem to provide confirmatory evidence for the hypothesis that tabu search holds an inherent advantage over other meta-heuristics in their basic forms for the JSP. These reasons justify the selection of a tabu search approach for the job shop problem. In the next section, priority matrix, operation-based, and the proposed hybrid solution representations are applied on a tabu search algorithm and their results are compared with each other.

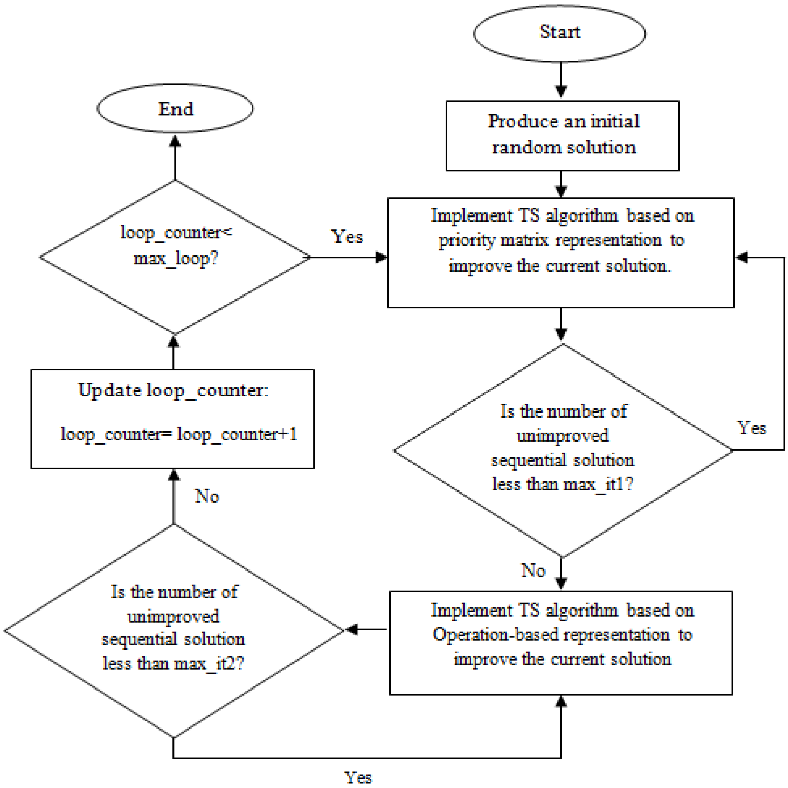

Fig 5. Flow Chart of the Proposed Algorithm

IV. THE PROPOSED ALGORITHM

In the current paper, the proposed tabu search algorithm was used a cyclic approach that the representations repeatedly changed from priority matrix to operation-based representation and inversely. For this purpose, two rules to create shocks were necessary. This matter prevented the algorithm to fall in a local optimum. When the changing rule was satisfied, the representation was transformed into the other. The changing rule defined the number of consecutive repetitions in which there were no changes in the best solution for each representation.

The algorithm proposed by the current paper began with a feasible solution. To use a better primal solution, it was better to search a larger solution space. Therefore, after gaining the first solution, the optimization process was started with priority matrix representation. If changing rule from priority matrix representation was gained, this representation was changed to operation-based one. It meant that the algorithm searched a small space to gain the better answer. If changing rule from operation-based representation was provided, the changing solution seed to priority matrix representation entered a good shock to algorithm. This cycle was repeated until a stopping condition was obtained. The flowchart of the proposed algorithm is given in Fig (5).

A. THE PROPOSED TABU SEARCH ALGORITHM

Tabu search is a meta-heuristic local search algorithm that can be used to solve combinatorial optimization problems (Glover & Mcmillan). Local search procedures often get stuck in poor-scoring areas. Tabu search enhances the performance of these techniques using memory structures that describe the visited solutions or user-provided sets of rules (Glover). During the search, some of the possible solutions were named “Tabu” and included in a list through the memories, during a defined number of iterations. These solutions were not visited in some points of the search; meanwhile better solutions were found to replace an initial proposed solution, until some stopping criterion were satisfied (Glover & Laguna).

The pseudo code displayed in Fig (6), presents the tabu search algorithm as described above for the job shop scheduling problem.

```
Step 0: Best_solution ← x0(initial solution)
        Cmax* ← Cmax(x0)
        tabuList ← null
Step 1: while stopping Condition is not satisfied, repeat the following steps (for n=1, 2,…):
            Generate Neighbor solutions of the current solution and set:
            Best_ Neighborhood solution ← xn(current solution)
            Best_ Neighborhood Cmax ← inf
            For each x ∈ Neighborhood(xn)
                If (Cmax(x) < Best_ Neighborhood Cmax)
                    If moving from x to xn is not banned or if this move is in Tabu but Cmax(x) < Cmax*
                        Best_ Neighborhood solution ← x
                        Best_ Neighborhood Cmax ← Cmax(x)
                    End
                End
            End
            Xn+1 ← Best_ Neighborhood solution
        If (Best_ Neighborhood Cmax < Cmax*)
                Best_solution ← Best_ Neighborhood solution
                Cmax* ← Best_ Neighborhood Cmax
            End
            Tabu list must be updated.
        End
Return (Best_solution)
Return (Cmax*)
```

Fig 6. Pseudo Code of Proposed Tabu Search Algorithm

$$[1\ 2\ 1\ 3\ 2\ 3\ 1\ 3] \Rightarrow [1\ 2\ 3\ 3\ 2\ 1\ 1\ 3]$$

Fig 7. An Example of Neighbour Solution for the Operation-based Representation

$$\begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 \\ 1 & 2 & 3 & 3 & 1 & 2 & 2 & 3 & 1 \\ 3 & 1 & 2 & 2 & 1 & 3 & 3 & 1 & 2 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 & 3 \\ 1 & 2 & 3 & 3 & 1 & 2 & 2 & 3 & 1 \\ 3 & 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 \end{bmatrix}$$

Fig 8. An Example of Neighbour Solution for the Priority Matrix Representation

In the phase of neighbour solution generation for operation-based representation, a neighbour solution was defined by replacing every two numbers in the current solution. One example of neighbour solution for this encoding is displayed in Fig (7).

But in the priority matrix representation, as shown in Fig (8), a neighbour solution was obtained by substituting two numbers in the priority row for a certain machine.

As mentioned previously, to improve the quality of solutions obtained from these solution seeds, an algorithm was presented that used priority matrix and operation-based representations, repeatedly. In this algorithm, one of these representations were used at some sequential iterations of algorithm, and for each of these representations tabu search were run to search the solution space and improve the time completion (make-span).These frequent reformations could improve the exploitation and exploration powers. This matter is displayed in Fig (9). Two marked parts of this figure show the results obtained by running the algorithm using the operation-based encoding. As observed in this figure, the proposed algorithm in these two parts (operation-based representation) powerfully exploited the best obtained solution to improve it. But, in the other parts of the figure related to priority matrix representations, the algorithm intensively explored the solution space.

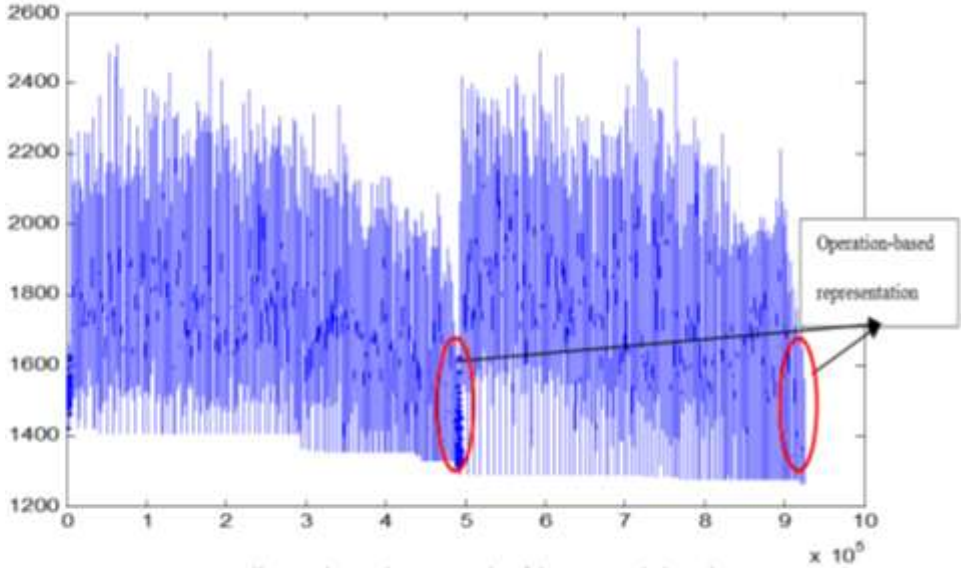

Fig 9. All C_{max} Obtained in Two Cycles of the Proposed Algorithm

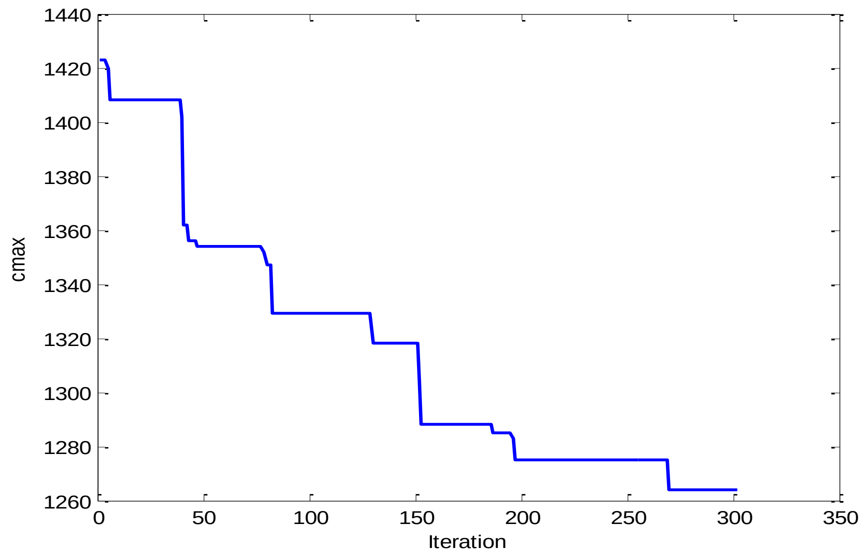

Fig 10. The Best C_{max} Obtained in Two Cycles of the Proposed Algorithm

By these descriptions, the best solution for JSP was improved in the sequential iterations and some better final solutions were obtained. Fig (10) confirms this matter for a numerical example.

As observed, this change prevented the algorithm to fall in a local optimal solution and more improvements in the final solution were possible by applying these sequential changes in encodings.

Some improved search strategies in the proposed classic algorithm could be applied to enhance the ability of algorithm to obtain the better solutions. From this point of view, two reformations were suggested in this paper:

(1) Using a tabu list by randomized length and

(2) changing the neighbourhood rule in which three components of solution seeds replaced each other in each solution to produce a neighbour solution (3-opt).

The results of these changes are discussed in the next sections.

TABLE II: The Parameters of the Tabu Search Algorithm

Parameter	Value
The length of tabu list	10
max_loop	6
max_it1	50
max_it2	40

TABLE III. The Comparative Results

Name	Dimension (n×m)	C_{max} _Optimal	The proposed algorithm			Priority matrix representations			Operation-based representation		
			C_{max}			C_{max}			C_{max}		
			C_{max} _Best	C_{max} _Avg	stDv	C_{max} _Best	C_{max} _Avg	stDv	C_{max} _Best	C_{max} _Avg	stDv
	3×3	22	22	22	0	22	22	0	22	22	0
MT06	6×6	55	57	59.1	1.19	59	59.6	0.52	55	58.6	1.43
LA01	10×5	666	678	686.7	13.05	724	739	11.74	689	706.2	14.07
LA21	15×10	1046	1135	1240	46.24	1323	1349.5	39.35	1217	1275.2	46.96
LA26	20×10	1218	1396	1453	33.87	1473	1544	58.75	1460	1506.2	36.84
LA06	15×5	926	926	928.5	5.58	927	961.9	28.76	926	932.3	6.06
LA11	20×5	1222	1222	1225	8.175	1222	1244.2	33.64	1222	1235.2	14.89
LA31	30×10	1784	1849	1890	27.7	1873	1948	49.9	1881	1967.2	56.75

V. NUMERICAL EXPERIMENTS AND DISCUSSION

Now that the performance of the proposed algorithm is presented and analysed, the effectiveness and applicability of this algorithm are discussed by comparing its results to the ones obtained from two other existing solution seeds in the special literature. The most recent and complete publication of the results from many of JSP test functions came from Liu et al. Before using these data in the analysis, it was necessary to determine the best parameters for the proposed tabu search algorithm. For this purpose, parameters were set by Minitab 16 software. The obtained parameters are displayed in Table II.

A. NUMERICAL EXPERIMENTS

Each problem was solved by this algorithm ten times with random initial solutions in each implementation.

The best, average and standard deviations of solutions obtained from these implementations for each problem are presented in Table III.

As observed in Table III, the proposed algorithm could obtain better answers in comparison with two other solution seeds in almost all problems. In other cases, such as MT06, LA06, and LA11 three algorithms could get to optimal solutions.

As well as the best solutions, the proposed algorithm had better average solutions between the other algorithms that used these representations, separately.

After computing the objective function for each test problem using the algorithm, relative percentage deviation (RPD) in percentage was calculated by the following formulas:

$$RPD\_BEST = \frac{\text{Cmax\_BEST} - \text{Cmax\_OPTIMAL}}{\text{Cmax\_OPTIMAL}} \times 100$$

$$RPD\_AVERAGE = \frac{\text{Cmax\_AVERAGE} - \text{Cmax\_OPTIMAL}}{\text{Cmax\_OPTIMAL}} \times 100$$

This criterion could be a useful and effective tool to measure the ability of the algorithm to reach a good and near optimal solution. The results from computing this criterion for the current test problem are displayed in Table IV.

TABLE IV. The RPD Quantities of the Improved Classic Tabu Search Algorithm

Name	Dimension (n×m)	$C_{max\_OPTIMAL}$	The proposed algorithm		Priority matrix representations		Operation-based representation	
			C_{max}		C_{max}		C_{max}	
			RPD_best (%)	RPD_average(%)	RPD_best (%)	RPD_average(%)	RPD_best (%)	RPD_average (%)
–	3×3	22	0	0	0	0	0	0
MT06	6×6	55	3.63	7.45	7.27	8.36	0	6.55
LA01	10×5	666	1.80	3.10	8.71	11	3.45	6.04
LA21	15×10	1046	8.50	18.55	26.5	29	16.34	21.9
LA26	20×10	1218	14.61	19.29	20.9	26.8	19.87	23.7
LA06	15×5	926	0	0.27	0.11	3.88	0	0.68
LA11	20×5	1222	0	0.24	0	1.82	0	1.08
LA31	30×10	1784	3.64	5.94	4.99	9.19	5.43	10.3

TABLE V. Computational Time of the Numerical Examples

Name	Dimension (n×m)	The proposed algorithm	Priority matrix representations	operation-based representation
		Elapsed time (sec)	Elapsed time (sec)	Elapsed time (sec)
	3×3	0.235	0.2481	0.223333
MT06	6×6	9.076	8.28	5.69
LA01	10×5	26.07	19.93	13.586
LA21	15×10	243.304	380.685	357.3625
LA26	20×10	456.54	1016.339	915.4625
LA06	15×5	71.96	61.915	48.48
LA11	20×5	118.67	192.158	141.404
LA31	30×10	1895.7	3120.55	3559.98

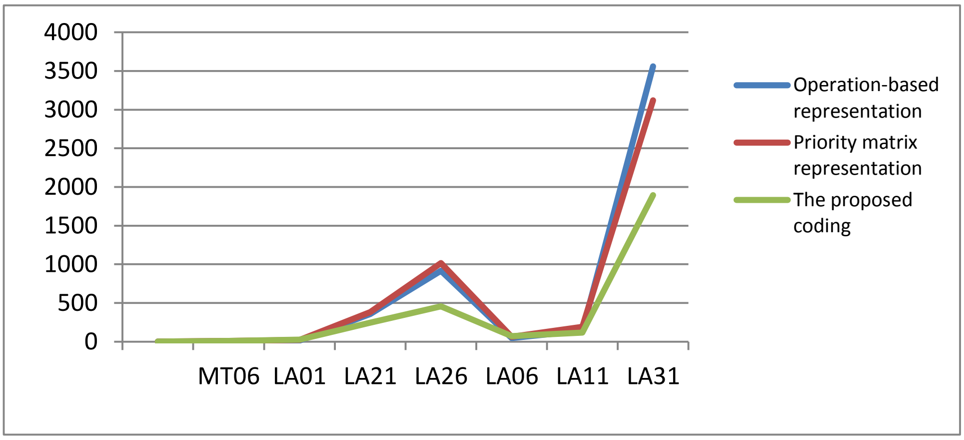

Fig 11. The Average Computational Time for Different Job Shop Problems

As observed in Table IV, the proposed algorithm had the best quantities for RPD in comparison with two other solution representations.

The computational times of the mentioned problems are presented in Table V.

To investigate the efficiency of the proposed algorithm, elapsed times of solving these problems were computed. Fig (11) showed that from this point of view, the proposed algorithm for most of the test problems had better speed among the others in reaching the final solution.

TABLE VI. The Results of the Improved Classic Tabu Search Algorithm

Name	Dimension (n×m)	C_{max} _Optimal	2-opt (constant Tabu length)			2-opt (randomized Tabu length)			3-opt (constant Tabu length)			3-opt (randomized Tabu length)		
			C_{max}			C_{max}			C_{max}			C_{max}		
			C_{max} _Best	C_{max} _Avg	stDv	C_{max} _Best	C_{max} _Avg	stDv	C_{max} _Best	C_{max} _Avg	stDv	C_{max} _Best	C_{max} _Avg	stDv
	3×3	22	22	22	0	22	22	0	22	22	0	22	22	0
MT06	6×6	55	57	59.1	1.19	58	60	1.58	55	56.8	1.8	55	57.2	2.05
LA01	10×5	666	678	686.7	13.05	678	704.8	15.8	666	675.2	8.75	666	670	7.82
LA21	15×10	1046	1135	1240	46.24	1224	1263	32.6	1170	1195.6	18.2	1132	1168	41.3
LA26	20×10	1218	1396	1453	33.87	1425	1437.4	16.7	1331	1381.4	52.4	1334	1367	35.6
LA06	15×5	926	926	928.5	5.58	926	926	0	926	926	0	926	926	0
LA11	20×5	1222	1222	1225	8.175	1222	1223.8	4.02	1222	1224.6	5.8	1222	1224	4.1
LA31	30×10	1784	1849	1890	27.7	1813	1886.3	63.9	1784	1882.8	67.6	1813	1886	63.9

TABLE VII. The RPD Quantities of the Improved Classic Tabu Search Algorithm

Name	Dimension (n×m)	$C_{max\_OPTIMAL}$	2-opt (constant Tabu length)		2-opt (randomized Tabu length)		3-opt (constant Tabu length)		3-opt (randomized Tabu length)	
			C_{max}		C_{max}		C_{max}		C_{max}	
			RPD_best(%)	RPD_average(%)	RPD_best (%)	RPD_average(%)	RPD_best (%)	RPD_average (%)	RPD_best (%)	RPD_average (%)
	3×3	22	0	0	0	0	0	0	0	0
MT06	6×6	55	3.64	7.45	5.45	9.09	0	3.27	0	4
LA01	10×5	666	1.8	3.11	1.8	5.83	0	1.38	0	0.6
LA21	15×10	1046	8.51	18.5	17	20.7	11.85	14.3	8.22	11.7
LA26	20×10	1218	14.6	19.3	17	18	9.27	13.42	9.52	12.2
LA06	15×5	926	0	0.27	0	0	0	0	0	0
LA11	20×5	1222	0	0.25	0	0.15	0	0.213	0	0.16
LA31	30×10	1784	3.64	5.94	1.63	5.73	0	5.54	1.63	5.72

B. NUMERICAL EXPERIMENTS FOR THE IMPROVED TABU SEARCH ALGORITHMS

As described in the previous section, some changes could be applied to improve the effectiveness of the tabu search algorithm. The numerical results of these changes are represented in Table VI.

As observed in Table VI, tabu search algorithm with improved neighbouring rule as moving three components of representation could reach better solutions in several implementations for six of the eight studied problems.

As observed, the suggested reformations in the neighbouring rule could result in better average answers in all studied cases. On the other hand, randomized length of tabu list could result in better solutions in several cases in 2-opt tabu search algorithm. The proposed tabu search algorithm with 3-opt neighbouring rule by randomized tabu length could reach better answers in comparison with the other algorithms in most of the problems discussed in this paper.

RPD quantities for these improved types of the proposed tabu search algorithm are shown in Table VII.

VI. CONCLUSIONS

JSP is one of the hardest combinatorial optimization problems. Therefore, effective meta-heuristic methods are necessary to obtain better solutions for it. One of the most effective elements to improve the quality of obtaining solutions from meta-heuristics is applying a better representation. It causes, the exploration and exploitation of the algorithm that can be improved, simultaneously. In the current paper, two main existing solution seeds of JSP were combined.

The proposed algorithm had good features of the two representations and improved the exploration and exploitation of the algorithm by sequential transforming from each solution seed to the other. Also, a cyclic manner was applied in a tabu search algorithm. Numerical experiments confirmed the effectiveness of this algorithm.

Furthermore, in the current paper some reformations in the neighbouring rule and tabu length were applied to improve the obtained solutions of the tabu search algorithm. The experimental results showed that 3-opt neighbouring rule could enhance the effectiveness of the proposed algorithm.

Besides using the proposed representation for other meta-heuristic algorithms such as simulated annealing, genetic algorithm, ant colony, etc., investigation of other rules for tabu length and reformations in the neighbourhood rules could be a good area to improve the results for future researches.

REFERENCES

Asadzadeh, L. (2015). A local search genetic algorithm for the job shop scheduling problem with intelligent agents.Computers & Industrial Engineering, 85, 376–383.

Bagheri, A., et al. (2010). "An artificial immune algorithm for the flexible job-shop scheduling problem." Future Generation Computer Systems, 26.4, 533-541.

Bla˙zewicz J., Domschke W. & Pesch E. (1996). The job shop scheduling problem: conventional and new solution techniques. European Journal of Operational Research, 93(1):1–33.

Brandimarte P. (1993). Routing and scheduling in a flexible job shop by tabu search. Ann Oper Res, 41:157–183.

Bülbül, K. (2011). A Hybrid Shifting Bottleneck-Tabu Search Heuristic For The Job Shop Total Weighted Tardiness Problem. Computers & Operations Research, 38, 967-983.

Chang, P., Lin, K., Pai, P., Zhong, C., Lin, C., & Hung, L. (2008). Ant Colony Optimization System For A Multi-Quantitative And Qualitative Objective Job-Shop Parallel-Machine-Scheduling Problem.International Journal of Production Research, 46(20), 5719-5759.

Cheng, R., Gen, M., & Tsujimura, Y. (1996). A Tutorial Survey of Job-Shop Scheduling Problems Using Genetic Algorithms-I Representation", Computers & Industrial Engineering, 30, 983-997.

Dauzere-Peres S Paulli J. (1997). An integrated approach for modeling and solving the general multiprocessor job shop scheduling problem using tabu search, Ann. Operat. Res, 70, 281–306.

De Giovanni, L., & Ferdinando P. (2010). "An improved genetic algorithm for the distributed and flexible job-shop scheduling problem." European journal of operational research, 200.2, 395-408.

Elmi, A., Solimanpur, M., Topaloglu, S., & Elmi, A. (2011). "A Simulated Annealing Algorithm For The Job Shop Cell Scheduling Problem With Intercellular Moves And Re-entrant Parts", Computers & Industrial Engineering, 61, 171-178.

Fattahi P, Saidi M & Jolai F. (2007). Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. J Intell Manuf, 18:331–342

Fattahi, P., Jolai, F., & Arkat, J. (2009). Flexible Job Shop Scheduling With Overlapping In Operations. Applied Mathematical Modeling, 33, 3076-3087.

Fattahi, P., ShiraziManesh, M. & &Roshani, A. (2012). A New Solution Seed For Job Shop Scheduling Problem. Applied Mechanics And Materials, 110-116, 3899-3905.

Gao, L., Li, X., Wen, X., Lu, C., & Wen, F. (2015). A hybrid algorithm based on a new neighbourhood structure evaluation method for job shop scheduling problem.Computers & Industrial Engineering, 88, 417–429.

Garey, M., Johnson, D., & Sethi, R. (1976). The Complexity Of Flow shop And Job Shop Scheduling. Mathematical Methods of Operations Research, 1, 117-129.

Gen, M., & Cheng, R. (1997). Genetic Algorithm and Engineering Design. John Wiley, New York.

Glover, F., & Laguna, M. (1997). Tabu Search. Kluwer Academic Publishers, Boston, Mass, USA.

Glover, F., & Mcmillan, C. (1986).The General Employee Scheduling Problem: An Integration of MS And AI. Computers And Operations Research, 13, 563–573.

Glover, F. (1989). Tabu Search - Part 1. ORSA Journal On Computing, 1 (2), 190–206.

Hurink E, Jurisch B, Thole M. (1994). Tabu search for the job shop scheduling problem with multi-purpose machines, Operat. Res. Spekt, 15, 205–215.

Jamili, A., (2016), “Robust job shop scheduling problem: Mathematical models, exact and heuristic algorithms”, Expert Systems with Applications 55, 341-350.

Jain, A. S., & Meeran, S. (1999). A State-Of-The-Art Review Of Job-Shop Scheduling Techniques. European Journal Of Operations Research, 113, 390-434.

Ku, W. Y., Beck, J. C., (2016), “Mixed Integer Programming models for job shop scheduling: A computational analysis”, Computers & Operations Research, 73, 165-173.

Kuhpfahl, J. & Bierwirth, C. (2016). A study on local search neighbourhoods for the job shop scheduling problem with total weighted tardiness objective.Computers & Operations Research, 66, 44-57.

Kurdi, M. (2015). A new hybrid island model genetic algorithm for job shop scheduling problem.Computers & Industrial Engineering, 88, 273-283.

Liu, J., Zhong, W., & Jiao, L. (2006).A Multiagent Evolutionary Algorithm For Constraint Satisfaction Problems. IEEE Transactions On Systems, Man, And Cybernetics-Part B: Cybernetics, 36, 54-73.

Mastrololli M & Gambardella L.M. (2002). Effective neighbourhood functions for the flexible job shop problem, J. Schedul. 3 (1) 3–20.

Mellor, P. (1966). A Review of Job Shop Scheduling. Operational Research Society, 17, 161-171.

Moslehi, Ghasem, & Mehdi Mahnam. (2011). "A Pareto approach to multi-objective flexible job-shop scheduling problem using particle swarm optimization and local search." International Journal of Production Economics, 129.1: 14-22.

Rabiee, M.,Zandieh, M., & Ramezani. P. (2012). Bi-Objective Partial Flexible Job Shop Scheduling Problem: NSGA-II, NRGA, MOGA And PAES Approaches. International Journal of Production Research, 50(24), 7327-7342.

Rossi, Andrea, & Elena Boschi. (2009). "A hybrid heuristic to solve the parallel machines job-shop scheduling problem." Advances in Engineering Software, 40.2: 118-127.

Saidi-mehrabad M & Fattahi P (2007). Flexible job shop scheduling with tabu search algorithms. Int J Adv Manuf Technol, 32:563–570

Wang, Lei, & Dun-bing Tang. (2011). "An improved adaptive genetic algorithm based on hormone modulation mechanism for job-shop scheduling problem." Expert Systems with Applications, 38.6: 7243-7250.

Zhao, F., Zhang, J., Zhang, C., & Wang, J. (2015). An improved shuffled complex evolution algorithm with sequence mapping mechanism for job shop scheduling problems. Expert Systems with Applications, 42 (8), 3953–3966.