

GENETIC AND TABU SEARCH ALGORITHMS FOR THE SINGLE MACHINE SCHEDULING PROBLEM WITH SEQUENCE-DEPENDENT SET-UP TIMES AND DETERIORATING JOBS

U. Bahalke*

University of Science and Technology, Department of Industrial Engineering, Tehran, Iran
Khoddami@cc.iut.ac.ir

A. M. Yolmeh

Sharif University of Technology, Department of Industrial Engineering, Tehran, Iran

A. N. Hajizade

University of Science and Technology, Department of Railway Engineering, Tehran, Iran

A. Bahalke

University of Allameh Tabatabaee, Department of Economic, Tehran Iran

* Corresponding Author

(Received: December 6, 2009 – Accepted in Revised Form: July 15, 2010)

Abstract This paper introduces the effects of job deterioration and sequence dependent set-up time in a single machine scheduling problem. The considered optimization criterion is the minimization of the makespan (C_{max}). For this purpose, after formulating the mathematical model, genetic and tabu search algorithms were developed for the problem. Since population diversity is a very important issue in preventing from being trapped in a local optimum, two methods were used to overcome this problem. Because of the novelty of the considered problem, there was no standard benchmark available to test the developed algorithms, so the results were checked by a lower bound which was calculated by a procedure that emphasizes on job deterioration effects. The tests and evaluations illustrate the effectiveness of the presented algorithms.

Keywords Single machine; Deterioration; Sequence-dependent set-up time; Genetic algorithm; Tabu search.

چکیده این مقاله تاثیر کارهای با ویژگی زوال و زمان های آماده سازی وابسته به توالی در مسایل زمان بندی تک ماشین معرفی می شود. تابع هدف در نظر گرفته شده به این منظور، زمان اتمام آخرین کار می باشد. بدین منظور بعد از مدل کردن مسئله، دو الگوریتم ژنتیک و جست و جوی ممنوع نیز برای مسئله توسعه داده شده اند. از آنجا که دامنه و گستردگی جمعیت در الگوریتم ژنتیک یکی مسئله های مهم در جلوگیری از گیر افتادن در بهینه موضعی می باشد، دو روش برای غلبه بر این مشکل مورد استفاده قرار گرفته اند. به دلیل جدید بودن مسئله مورد مطالعه مسائل حل شده ی استاندارد موجود نمی باشد تا الگوریتم های بکاررفته مورد مقایسه قرار گیرند، به همین دلیل حد پایین هایی تهیه شده اند که تاکید بیشتری بر ویژگی زوال کارها دارند. تست ها و ارزیابی های انجام شده کارا بودن الگوریتم های ارائه شده را نشان می دهند.

1. INTRODUCTION

Job deterioration and sequence-dependent set-up times are both very important elements in increasing makespan or C_{max} . This issue (job deterioration) was first introduced by Browne and Yechiali [1]. They defined the processing time of these tasks as a function of their starting time

($P_j = a_j + b_j \times t_j$), where a_j is the fixed part of the processing time, b_j is a positive coefficient and t_j is the starting time of the job j .

There are two types of set-up times: sequence-independent and sequence dependent. In the first one, the set-up time depends only on the task to be processed, regardless of its preceding task. While in the second type, the set-up time depends on

both the task and its preceding task. This paper considers the sequence dependent one. In the field of scheduling and set-up time, Jeng and Lin [2] studied a scheduling problem of minimizing the total completion time on a single machine where the processing time of a job is a function of its starting time and a due date that is common to all jobs. They showed that this problem is NP-hard and developed a lower bound and two elimination rules to design a branch-and-bound algorithm to derive optimal solutions from a practical aspect. Stecco et al. [3] developed a tabu search heuristic for sequence dependent and time-dependent scheduling problem on a single machine. Shin et al. [4] presented a heuristic and tabu search algorithm for the single machine scheduling problem with sequence-dependent set-up times, release times and due dates, with the objective function of minimizing the maximum lateness. Ji and Cheng [5] considered the parallel machine scheduling by deteriorating jobs with the objective function of minimization of the total completion time. Koulamas and Kyparisis [6] first introduced a scheduling problem with past-sequence-dependent set-up time. By considering of the past sequence-dependent set-up time, the set-up times are proportional to the length of the already processed jobs and the processing time of the job is a function of its starting time. There are numerous papers in the field of scheduling with considering set-up times or deteriorating tasks separately. The paper of Tavakkoli-moghaddam and Mehdizade [7] could be mentioned. They proposed a new ILP model for the identical parallel machine scheduling problem with the family set-up times, the considered objective function is the total weighted flow time. Also, Tavakkoli-moghaddam et al [8] considered identical parallel machine scheduling problem under the sequence dependent set-up time and precedence relationships. Tavakkoli-moghaddam and Aroman-Bajestani [9] developed a new B and B algorithm for the above problem. In the case of meta-heuristic methods and scheduling problems Tavakkoli-Moghaddam et al. [10] applied genetic algorithms to parallel machine scheduling problems. Seraj and Tavakkoli-moghaddam [11] presents Tabu search algorithm for open shop scheduling problem by a fuzzy multi objective decision making approach.

So to the best of our knowledge the effect of deterioration and sequence dependent set-up time simultaneously has not been considered in scheduling problems. For this aim a 0-1 integer programming model is developed in section (2). The remainder of the paper is as follows. In section (3) proposed genetic algorithm is presented and section (4) includes the tabu search algorithm. At the end the test problems are solved in section (5).

2. PROBLEM DESCRIPTION AND MATHEMATICAL FORMULATION

The single machine scheduling problem with only considering the sequence dependent set-up times are classified in NP-hard problems [3] and the considered problem with additional deterioration parameter fall into NP-hard problems, too. So, the developed mathematical model will be effective for small and medium size problems. Thus, in this section mathematical model will be developed and in the next section meta-heuristic methods are applied to deal with the larger size problems. Problems can be described as follows: there are N independent jobs $\{J_1, J_2 \dots J_N\}$ available at time zero, which should be processed on a single machine. The preemption is not allowed and the processing time of jobs depends on their starting time and their sequence position, because of the job deterioration and set-up times, respectively. The additional conditions of this problem should be noted as follows:

- Job processing time is described by functions of the starting time and fixed part of the processing time of the job ($p_j = a_j + b_j \times t_j$).
- The growth rate of the processing time (b_j) is independent of machine.
- The jobs are independent of each other.
- No job preemption is allowed.
- Set-up time is sequence-dependent.
- Each job j should be processed on machine only for once.
- The machine can handle one job at a time.

2.1 Known parameters and variables:

- N Total number of jobs to be scheduled
- $i, j \in I = \{0, 1, \dots, N+1\}$ Designate the job, where job 0 and job $N+1$ are dummy jobs, job 0 is always at the first sequence position and job $N+1$ is the last sequence.
- S_{ij} Set-up time of switching from job i to j , $i, j \in I, i \neq j$
- St_j Starting time of job j
- p_j Processing time of job j
- C_j Completion time of job j
- a_j Fixed part of the processing time of job j
- b_j The growth rate of the processing time of job j
- $X_{ij} \begin{cases} 1 & \text{if task } j \text{ positioned after task } i \\ 0 & \text{otherwise} \end{cases}$
- M A large positive number

$$j = 1, 2, \dots, N+1$$

$$s_{n+1j} = 0, St(0) = 0, C(0) = 0,$$

$$C(i), St(j), P(j) \geq 0$$

Equation (1) shows the objective function. Equation (2) ensures that always one job, job i , is assigned before job j . Equation (3) states that each job (except the last dummy job), immediately is followed by one job, job j . Number (4) ensures that a dummy ($i=0$), job is positioned in the first sequence position. Equation (5) declares that the starting time of job in a sequence position, job j , is equal to the sum of the completion time of preceding job. Equation (6) expresses that the processing time of job in a sequence position is described by functions of the starting time and fixed part of the processing time. Equation (7) correlates the processing time of a job to its starting time and completion time. Since mathematical models are computationally expensive for large scale problems; therefore some metaheuristic algorithms has been developed to solve the problems in a reasonable computational time.

2.2 Mathematical model

$$\text{Min } Z = C_{N+1} \quad (1)$$

$$\sum_{i=0}^{N+1} x_{ij} = 1 \quad (2)$$

$j = 1, 2, \dots, N+1, i \neq j$

$$\sum_{j=0}^{N+1} x_{i-1j} = 1 \quad (3)$$

$i = 1, 2, \dots, N+1, i \neq j+1$

$$\sum_{i=1}^{N+1} x_{i0} = 0 \quad (4)$$

$$St_j + M(1 - x_{ij}) \geq C_j + S_{ij} \quad (5)$$

$i, j = 1, 2, \dots, N+1, i \neq j$

$$P_j = a_j + b_j \times St_j \quad (6)$$

$j = 1, 2, \dots, N+1$

$$C_j \geq St_j + P_j \quad (7)$$

3. PROPOSED GA

Genetic Algorithm is a robust algorithm in optimization problems. In recent researches, advanced methods have been presented in GA field for achieving better results through the algorithm. In a pure GA, first a population of the feasible solutions is produced through the random process, afterwards a fitness value is described for each of the solutions in the population, and solutions are ranked due to the fitness values. Next, on the basis of a selection mechanism, a set of chromosomes are chosen for exerting crossover and mutation operators, next the produced individuals are transferred to the next pool. This process is called Generation, the generation is iterated until the stopping conditions are met.

3.1. Initial solution By improving genetic algorithms some heuristic and metaheuristic rules are considered for the initial solution of the GA.

Generally, by using these methods, genetic algorithms could give good results in a shorter time and with better convergence speed. It should be noted that if the whole or most members of the population are generated by the heuristic or metaheuristic rules, diversity of the population will be low and this will result in being trapped in a local optimum. This paper considers two forms of initializations. The first one is to generate by a random process and the second one is to produce one of the population members by the solution that Browne and Yechiali [1] developed. They showed that in a single machine scheduling problem with job deterioration, sequencing the jobs in non-decreasing order of (a_j/b_j) minimizes the makespan. Since by increasing the number of jobs in a single machine the effect of deterioration is really more than sequence-dependent set-up time on increasing makespan (C_{max}), this method will be efficient. Two types of initial solutions are described below:

1. Produce all members of the population randomly (R_GA).
2. Produce one member of the population by sorting jobs in non decreasing order of (a_j/b_j) and generate the rest of the pool randomly (H-GA).

3.2. Selection Mechanism and Replacement

In this paper the classic roulette wheel selection method [12] is implemented. After doing crossover and mutation, the best solution of former population can be missed, and this factor may lead to cycling or late convergence. In this paper a replacement method is used for preventing this problem. By using this application, detrimental effect of crossover and mutation is annihilated. The replacement mechanism is described as follow:

1. Sort the individuals of the population in ascending order of fitness values.
2. Transfer best 25% of the current population chromosomes to the next pool.

Substantially, by doing this mechanism current best solutions would not be worse than former best solutions.

3.3. Crossover Crossover is the process of taking two parents and producing a child. After the reproduction process, the population is enriched with better chromosomes. Crossover operator is

applied to the mating pool with the hope of creating a better child. In this paper, the considered operator is called “OX or Ordered Crossover” proposed by Davis [13].

3.4. Mutation Mutation process guarantees the diversity in the population. In this paper, during the mutation process, the mutation probability increases as the diversity of the population decreases. For this purpose, a function, which is same as the function of Zhou et al [14], is described for the diversity as follows:

$$\text{Diversity} = (f_{\max} - f_{\min}) / \bar{f}$$

where f_{\max} and f_{\min} are the largest and smallest fitness values (C_{\max}), among the chromosomes of the population, respectively, and \bar{f} is the average fitness value. Nearchou [15] used an application for controlling the convergence in GA and

proposed a formula as $\min D = f_{\min} / \bar{f}$, where the mutation probability decreases as the value of D exceeds a threshold 0.95. When this condition is hold, the chromosomes in entire population are almost the same, thus the population convergence is met. But in our proposed GA, the application of Nearchou is done exactly inverse for controlling the diversity in the population i.e. mutation probability (P_m) increases as the value of D exceeds a threshold 0.95. Because of being replacement method, without affecting in the case of convergence, it is possible to increase the P_m to reach a larger solution area (bigger diversity). The application is as follows:

- If $D \geq 0.95$ then $P_m = 1.5 P_m$;

Well-known Swap mutation is used for the proposed GA.

3.5. Restart scheme In GAs, it is common to achieve low diversity and trapping in a local optimum. To overcome this problem, a procedure which is called “Restart scheme” is used. Restart scheme is a really effective technique in controlling diversity and escaping from local optimality without negative effects of convergence problem. This method is used in Ruiz et al. [16 and 17] for flow shop scheduling problems. The mechanism of restart is described as follow:

1. At each iteration i , store the minimum value of makespan, mak_i .
 2. If $mak_i = mak_{i-1}$, then $count = count + 1$. Otherwise, $count = 0$.
 3. If $count > Gr$, then apply the following method:
 - Sort the chromosomes of the population in ascending order of makespan.
 - Fill the first 20% of the next pool from the sorted list (the best chromosomes).
 - From the remaining 80% population, 50% are produced by doing mutation on the first best 20%.
 - From the remaining 30% population, one chromosome is produced by applying the Browne and Yechiali's sorting method.
 - The rest of population is produced randomly.
 - $Count = 0$;
- GA's searching process stops after doing $100 \times n$ (n is the number of all jobs) iterations.

4. TABU SEARCH

Tabu search is a meta-heuristic neighborhood search methodology, which is introduced and developed by Glover [18]. Tabu search starts from an initial solution and moves at each iterations from the current solutions to the best solution in its neighborhood $N(s)$, even if this movement deteriorates the objective function value. To avoid the issue of local optimality, attributes of recently visited solutions are declared tabu for a certain number of iterations; however this tabu restriction can be ignored if a forbidden movement meets aspiration criteria. The structure of proposed TS is described in the next sections.

4.1. Representation and initial solution

Structure of TS representation is same as the used chromosomes in GA. Every permutation of jobs in a chromosome with the length of n (number of all jobs) shows a feasible solution for TS. Initial solution, starts from the Browne and Yechialie's sorting method i.e. sequencing job in non-decreasing order of (a_j/b_j) .

4.2. Neighborhood generation

A neighbor of a solution is obtained by exchanging each job with the immediately following job. In proposed TS, for each solution all neighbors are produced.

4.3. Tabu status and tabu tenure After exchanging job $_i$ in position j with the following job in position $j+1$, locating job $_i$ in position j is considered to be tabu. In this study, fixed tabu tenure β is considered, which follows the considered formulation of Stecco et al [3] which is expressed in Equation (1).

$$b = \frac{(2\sqrt{n} + n)}{2} \quad (8)$$

4.4. Aspiration and stopping criteria Tabu status will be over ruled if a tabu solution is better than the best solution so far. The tabu search stops when the value of objective function do not improve for k ($k=1000$) consecutive iterations.

5. DIVERSITY

Maintaining diversity is a very important element in increasing solution area and escaping from the local optimum. One method to overcome this problem is increasing population size, but simply increasing the population size may not be enough to avoid the problem, while any increase in population size will incur the twofold cost of both extra computation time and more generations to converge on an optimal solution.

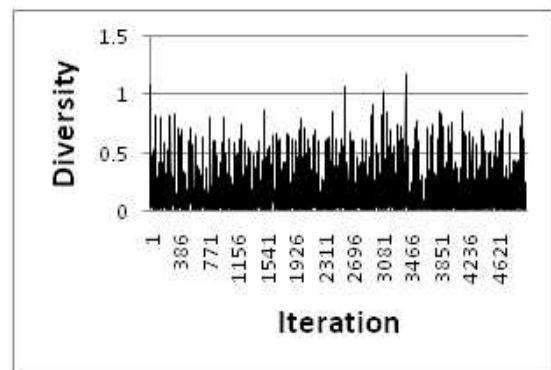


Figure 1. Diversity chart (using diversity maintaining schemes)

Sivanandam and Deepa [19] have stated that the genetic algorithm faces a difficult problem, which

is “How can a population be encouraged to converge on a solution while still maintaining the diversity?” In this paper, two methods are applied simultaneously for maintaining diversity and escaping from the local optimum (Restart and mutation probability increasing schemes), for overcoming the negative effects of diversity maintaining method on convergence problem, replacement is considered. Figures (1) and (2) show the diversity of genetic search in situations using diversity maintaining schemes and without using them, respectively. Also, it is obviously seen that there are significant differences between the output results of these two manners in part 7.

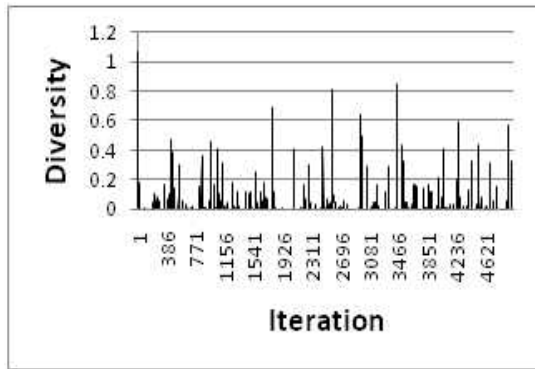


Figure 2. Diversity chart (non-executing diversity maintaining schemes)

6. COMPUTATIONAL RESULTS

At first, the results of proposed algorithms are compared with exact solution of mathematical model for the three small size problems, which all of the three algorithms could reach to optimal solution for these problems (Table 1).

TABLE 1. Small-size results

Instances	R-GA	TS	GA	Optimum	Offset
8 Jobs	61.44	61.44	61.44	61.44	0
10 Jobs	71.50	71.50	71.50	71.50	0
12 Jobs	117.23	117.23	117.23	117.23	0

Afterwards presented algorithms are tested for

some large size problems which are created by a procedure which is described in the following statements. Instances for the problem are created through a random process, which the fixed part of the processing times are chosen randomly from the set of {1, 2, 3, 4}, set-up times are a randomly produced matrix from a set of {0.1, 0.2, 0.3,0.4} and deterioration rates are created from the set of {0.2, 0.3, 0.4, 0.5}.

TABLE 2. Larger Sized Results

Instances	TS	R_GA	pure_GA	HGA
	offset(%)	offset(%)	offset(%)	offset(%)
20	5.90	3.38	3.60	3.36
30	2.39	2.46	3.87	2.00
40	5.19	3.59	12.88	2.67
50	2.11	1.44	19.19	1.43
60	2.90	1.96	14.63	1.59

Table 2 shows the results of proposed algorithms for some sample examples, which are created by the described procedure, it contains the results of TS, H_GA and R_GA. In addition, it shows the results of pure-GA, which has no diversity maintaining scheme and hybrid initial solution. Results of examples are checked by the lower bound, which is calculated on the basis of the Browne and Yechiali’s (1990) sorting method, that achieves global optimum solution for the single machine problem with job deterioration, but it would not be efficient for the problems with sequence-dependent set-up times. For overcoming this problem, the set up times are considered the lowest value from the set of values of the set-up time (0.1) for every job-switching, thus by means of this procedure the Lower Bound is calculated for all of instances. Results are shown on the basis of following expression (Equation 2). The best result of each instance is bolded. Where, as seen in results, worst results are relevant to pure-GA and the best results are related to H-GA.

$$offset = \frac{Algorithm_{sol} - LB}{LB} \times 100. \quad (9)$$

where $Algorithm_{sol}$ is the result of proposed algorithms and LB exhibits the lower bound.

Figure.3 shows the comparisons of convergence speed between R-GA and pure-GA and it can be seen that the proposed R-GA is significantly better than the pure genetic algorithm as its result and convergence speed are lower than pure-GA, respectively.

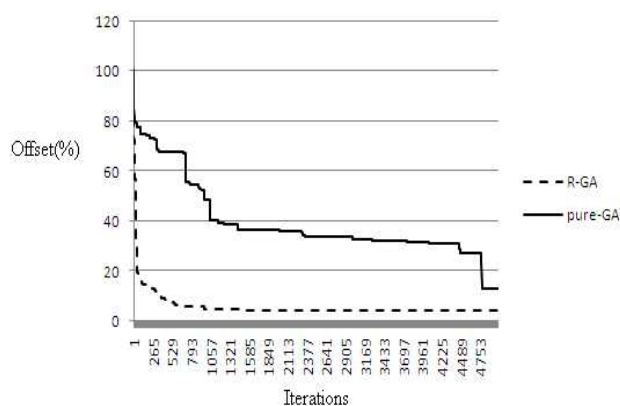


Figure 3. Convergence speed of R-GA and pure-GA

7. REFERENCES

1. Browne, S. and Yechiali, U., "Scheduling deteriorating jobs on a single processor". *Operations Research*, Vol. 38, (1990), 495–8.
2. Jeng, A. and Lin, B., "Minimizing the total completion time in single-machine scheduling with stepdeteriorating jobs". *Computers & Operations Research*, Vol. 32 (2005), 521–536.
3. Gabriella Stecco, Jean-François Cordeau and Elena Moretti. "A tabu search heuristic for a sequence dependent and time-dependent scheduling problem on a single machine". *J. of Sched.*, Vol. 12, (2009), 3–16. DOI 10.1007/s10951-008-0068-6.
4. Shin, H.J., Kim, C.O. and Kim, S.S. "A Tabu Search Algorithm for Single Machine Scheduling with Release Times, Due Dates, and Sequence-Dependent Set-up Times." *Int. J. Adv. Manuf. Technol.*, Vol. 19, (2002), 859–866.
5. Min, Ji and Cheng T.C.E., "Parallel-machine scheduling with simple linear deterioration to minimize total completion time". *European Journal of Operational Research*, Vol. 188, (2008), 342–347.

6. Koulamas C. and Kyparisis G.J, "Single-machine scheduling problems with past-sequence-dependent setup times". *European Journal of Operational Research*, Vol. 187, (2008), 68–72.
7. Tavakkoli-Moghaddam, R. and Mehdizade, E., "A new ilp model for identical parallel-machine scheduling with family setup times minimizing the total weighted flow time by a genetic algorithm". *International Journal of Engineering*, Vol. 20, (2007), 183-194.
8. Tavakkoli-Moghaddam, R., Taheri, F. and Bazzazi, M., "multi-objective unrelated parallel machines scheduling with sequence-dependent setup times and precedence constraints". *International Journal of Engineering*, Vol. 21, (2008), 269-278.
9. Tavakkoli-Moghaddam, R. and Aramon-Bajestani M., "A novel B and B algorithm for a unrelated parallel machine scheduling problem to minimize the total weighted tardiness". *International Journal of Engineering*, Vol. 22, (2009), 269-286.
10. Tavakkoli-Moghaddam, R., Jolai, F., Khodadadeghan, Y. and Haghnevis M., "a mathematical model of a multi-criteria parallel machine scheduling problem: a genetic algorithm (research note)". *International Journal of Engineering*, Vol. 19, (2006), 79-86.
11. Seraj O. and Tavakkoli-Moghaddam, R., "A tabu search method for a new bi-objective open shop scheduling problem by a fuzzy multi-objective decision making approach (research note)". *International Journal of Engineering*, Vol. 22, (2009), 269-282
12. Sivanandam, S.N. and Deepa S.N., "Introduction to Genetic Algorithms." ISBN 978-3-540-73189-4. Springer Berlin Heidelberg, New York, (2008), 47-48.
13. Davis, L., "Applying adaptative algorithms to epistatic domains", in: *Proceedings of the International joint conference on artificial intelligence*, (1985), 162–164.
14. Hong Zhou, Waiman Cheung and Lawrence C. Leung, "Minimizing weighted tardiness of job-shop scheduling using a hybrid genetic algorithm". *European Journal of Operational Research*, Vol. 194, (2009), 637–649.
15. Andreas C. Nearchou, "The effect of various operators on the genetic search for large scheduling problems". *Int. J. Production Economics*, Vol. 88, (2004), 191- 203.
16. Roben Ruiz, Concepcion Maroto and Javier Alcaraz. "Solving the flowshop scheduling problem with sequence dependent setup times using advanced metaheuristics". *European Journal of Operational Research*, Vol. 165, (2005), 34-54.
17. Roben Ruiz, Concepcion Maroto, Javier Alcaraz. Two new robust genetic algorithms for the flowshop scheduling problem. *Omega*, Vol. 34, (2006), 461-476.
18. Glover, F. and Laguna, M "Tabu search", Kluwer Academic, Boston, 1997.
19. Sivanandam, S.N. and Deepa, S.N., "Introduction to Genetic Algorithms". ISBN 978-3-540-73189-4. Springer Berlin Heidelberg New York, (2008), 89-90.