PERGAMON

Computers & Operations Research 28 (2001) 1223–1230

computers & operations research

www.elsevier.com/locate/dsw

Tabu search for a class of single-machine scheduling problems

U. Al-Turki, C. Fedjki, A. Andijani*

Department of Systems Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Received 1 September 1998; received in revised form 1 July 1999

Abstract

In this paper we develop a tabu search-based solution procedure designed specifically for a certain class of single-machine scheduling problems with a non-regular performance measure. The performance of the developed algorithm is tested for solving the variance minimization problem. Problems from the literature are used to test the performance of the algorithm. This algorithm can be used for solving other problems such as minimizing completion time deviation from a common due date.

Scope and purpose

Scheduling problems with non-regular performance measures has gained a great importance in modern manufacturing systems. These problems are found to be hard to solve and analyze. The purpose of this paper is to present a tabu search approach for solving a certain class of single-machine scheduling problems with non-regular performance measure. Minimizing the variance of completion times and the total deviation from a common due date are two examples of such problems. The proposed approach is found to perform better than the simulated annealing approach for the variance minimization problem.

Keywords: Tabu search; Scheduling; Variance minimization; V-shaped schedules

1. Introduction

The scheduling of a certain number of jobs on a single machine has gained importance for theoretical and practical reasons. There are many real-world situations involving a single resource used for accomplishing several tasks. Environments involving multiple machines can sometimes be

---

* Corresponding author. Tel.: + 966-3-860-2968; fax: + 966-3-860-2965.
E-mail address: chawki@ccse.kfupm.edu.sa (A. Andijani).

PII: S0305-0548(00)00036-8

considered as a single-machine problem by considering the bottleneck machine. Studying the single-machine problem theoretically is the building block for understanding larger scheduling problems of a more complex nature with multiple machines.

In this paper we develop a tabu search-based solution procedure designed specifically for certain class of scheduling problems having V-shaped optimum schedules. By limiting the solution procedure to a certain class of problems, we have the advantage of making use of the properties of that class to obtain good solutions that require less computation time than previously.

Many single-machine scheduling problems are known to have V-shaped optimum schedules. A V-shaped schedule is a schedule that satisfies the V-shaped property, i.e. jobs before the shortest job are in longest processing time (LPT) order and jobs after the shortest job are in shortest processing time (SPT) order. This property arises in problems with non-regular performance measures in Al-Turki [1].

The variance minimization problem is one of the earliest problems that are known to have the V-shaped property. Scharge [2] showed that the first job in the optimum schedule is the job with the longest processing time and later Eilon and Chowdhury [3] proved the V-shapeness of the optimal schedule. Another class of problems having the V-shaped property is the class of early-tardy problems. See Baker and Scudder [4] and Raghavachari [5] for such problems. Furthermore, there are certain stochastic single-machine problems that can be reduced to equivalent deterministic problems with optimum V-shaped schedules. Examples of such problems include minimizing the expected square deviation from a common due date when the machine is subject to random breakdowns [6] and minimizing the expected variance of job completion times when job processing times are random [7]. In general, the importance of the V-shaped property increases as the research of single-machine scheduling advances to stochastic environments and multiobjective performance measures.

Different techniques are used to search for the optimal solution among the 2^{n-1} possible V-shaped schedules. Among these techniques is the class of random search techniques such as simulated annealing and tabu search. Mittenthal et al. [8] developed a simulated annealing-based procedure to search for a near optimal solution to problems that have a V-shaped optimum schedule. They have reported their computational experience for the variance minimization problem. Their results motivated us to use the tabu search approach as an alternative approach that has shown superiority in many cases in finding optimal and near optimal solutions for a variety of problem settings.

In the next sections, we briefly describe the tabu search approach and its parameters. The parameter setting is given in Section 3. In Section 4 the algorithm is described as used in this paper. Computational experience is reported in Section 5 and the results are compared to those obtained by Mittenthal et al. [8]. We conclude with some remarks and observations in the last section.

2. The tabu search technique

A tabu search is a meta-heuristic that guides local heuristic search procedures to explore the solution space beyond local optimality. It was introduced by Glover [9–11] specifically for combinatorial problems. Its basic ideas have also been proposed by Hansen [12] and Hansen and Jaumard [13] the latter using the phrase *steepest ascent mildest descent*. Since then, tabu search has successfully been applied to flow-shop scheduling [14,15], generalized assignment problems [16], architectural design [17], time-tabling problems [18], interacting hub facility locations [19], among others.

The tabu search requires the following basic elements to be defined:

- *Configuration* is an assignment of values to variables (i.e. a solution).
- *A move* is a specific procedure for getting a trial solution which is a feasible solution to the optimization problem that is related to the current solution (i.e. a move is a procedure by which a new solution (a neighbor) is generated from the current one by some minor random perturbation of the current solution). This procedure is usually straightforward for the case of combinatorial optimization, but a lot harder to design for the case of continuous optimization.
- *A set of candidate moves* is the set of all possible moves out of a current configuration. This set is usually countable for the case of combinatorial optimization, but can be large, in which case one could operate with a subset of this set. For continuous optimization problems, the set of possible moves is uncountable and one has to be more creative in defining them.
- *Tabu restrictions*: This is the most distinctive feature of tabu search. It is what differentiates tabu search from simulated annealing algorithms. The tabu restrictions play a memory role for the search in the sense that recent moves are not allowed to be reversed by making them forbidden (*tabu*). This is done by forming a list called *tabu list*, of all of these tabu moves.
- *Aspiration criteria*: These are rules that override tabu restrictions, i.e. if a certain move is forbidden by tabu restrictions then the aspiration criteria, when satisfied, can make this move allowable (e.g. if a tabu move is better than the best obtained so far by the search, then one can take this move even though it is tabu, or override the tabu restriction).

Given the above basic elements, the tabu search scheme can be described as follows: start with an initial configuration, label this configuration as the current one, evaluate the objective (criterion) function for that configuration. Then, using the current configuration and a procedure for obtaining trial solutions, generate a certain set of candidate moves and evaluate their corresponding objective function values. If the best of these moves (in terms of the objective function) is not tabu or if the best is tabu, but satisfies the aspiration criteria, then select that move and consider it to be the new current configuration. Otherwise pick the first non-tabu move and consider it to be the new current configuration. Repeat the above procedure for a certain number of iterations. On termination, the best solution obtained so far is the solution obtained by the algorithm.

Note that the move that is chosen at a certain iteration is put in the tabu list so that it cannot be reversed in the next few iterations. The tabu list has a certain size (which is treated as a parameter), and when its length reaches that size and a new move enters that list, then the first move (i.e. the oldest tabu move) in the tabu list becomes non-tabu and the process continues (i.e. the tabu list is circular). The tabu list size controls the tabu search to either emphasize *exploration* or *itensification*. If the tabu list is small, then intensification is emphasized, i.e. a local search around the current point is intensified. If the tabu list is large, then exploration (different regions of the solution space are explored) is emphasized, i.e. points that are far from the current point are examined. The *aspiration criteria* can reflect the value of the criterion (objective) function, or, if the tabu move results in a value of the criterion function that is better than the best known so far, then the *aspiration criteria* is satisfied and this overrides the tabu restriction.

3. Parameters of the algorithm

In this section, we discuss the parameters of the algorithm, and their expected effects on performance. These parameters are described below.

Neighbor generation scheme: To generate a neighbor, a job is selected randomly and moved to the opposite arm. The moved job is inserted in the opposite arm in such a way that the V-shaped property is not violated.

Tabu list size (*NTLM*): The tabu list contains the history of the search, and it is the most distinctive feature of the tabu search technique. It assumes that the last NTLM moves will not be reversed, and hence the solutions in the immediate past will not be revisited. NTLM is the maximum number of moves that one can store in the list; therefore the larger the value of NTLM, the stronger the memory of the search which, in turn, encourages more *diversification* by the algorithm. On the other hand, the smaller the value of NTLM, the less memory the search has, and therefore the algorithm emphasizes more *intensification* in the search.

Probability threshold (*P*): The probability threshold P is used to allow moves that are tabu and better than the present current solution (although worse than the best solution found so far) to be examined as this may lead to a better solution.

Number of trial solutions (*NH*): In tabu search, one investigates NH trial solutions for deciding on the next move to take. The larger the value of NH, the more neighbors are examined and the more the search is intensified, while smaller values of NH imply fewer neighbors to examine and hence the search emphasizes diversification.

The maximum number of non-improving moves (*IMAX*): This parameter decides on how many non-improving moves are allowed.

4. The algorithm

In this section we give an outline of the proposed algorithm.

Initialization: Choose the maximum number of iterations allowed IMAX, the probability threshold P, the tabu list size NTLM and the number of neighbors to be generated at every iteration NH. Obtain a V-shaped schedule. Set the initial schedule as the current schedule CS and initial objective function as the current value JC. Set the best schedule BS = CS and the best objective function JB = JC, let I = 1 and go to step 1.

Step 1: Generate NH neighbors of the current schedule CS. Compute the objective function for each neighbor generated. Let JT, TS be the objective function value and the corresponding schedule of the best neighbor, respectively. Go to step 2.

Step 2: If the neighbor is not tabu (see tabu restriction) go to step 3. Otherwise, if JT < JC set JC = JT, CS = TS and go to step 5. Otherwise, go to step 4.

Step 3: If JT < JB, set JB = JT, JC = JT, BS = TS, CS = TS and go to step 5. Otherwise Set JC = JT, CS = TS and go to step 5.

Step 4: Generate a random number. If the number generated is greater than the probability threshold P, set JC = JT, CS = TS and go to step 5. Otherwise, if all neighbors have been checked go to step 6. Otherwise, let TS be the next best schedule and JT be the corresponding objective function value and go to step 2.

Step 5: If the tabu list is not full, add the move to the first free position in the list. Otherwise delete the oldest move (top of the list), shift up the list and add the move at the bottom of the list. Go to step 6.

Step 6: If $I >$ IMAX stop, the best schedule is BS. Otherwise set $I = I + 1$ and go to step 1.

5. Computational experience

The first stage in our computational experience involved many trial experiments to choose values for the parameters that performed well for various problem sizes. As a result, the parameters were set as follows: NTLM $= 6, P = 0.95$, NH $= 5$, IMAX $= 5000$.
These values were fixed for subsequent stages of the computational experience.

To test the performance of the algorithm, we solved the same set of problems that Mittenthal et al. [8] used for testing their simulated annealing based algorithm. These problems were also used by Elion and Chowdhury [3], Vani and Raghavachari [7], Gupta, Gupta and Bector [20]. The set consists of 21 problems of different sizes ranging from 6 to 20 jobs. The solutions to the problems are reported by Mittenthal et al. [8] along with the relative errors of the simulated annealing algorithm using three different initial schedules; LPT, V-shape, SPT. The V-shaped schedule is in the form $(J_1, J_3, J_5, J_7, \ldots, J_6, J_4, J_2)$ where job J_1 is the job with the largest processing time.

In this paper all the problem are solved with three different initial solutions similar to Mittenthal et al. [8] except for the V-shaped schedule. The initial V-shaped schedule chosen in this paper is in the form $(J_1, J_3, J_4, J_7, \ldots, J_6, J_5, J_2)$, where job J_1 is the job with the largest processing time. This sequence was suggested by Al-Turki [21] as a good and fast initial sequence for the variance minimization problem.

Table 1 shows the processing time of 21 problems given in optimal sequence and their corresponding minimum variances. The relative error, (Optimal − Tabu)/Optimal × 100, for the three initial solutions (LPT, V-shape, SPT) obtained by Mittenthal et al. [8] are reported in columns 4, 5 and 6, respectively. The last three columns of Table 1, give the relative error obtained using tabu search. The total relative error for each method of solution is given in the last row of the table.

The results show that almost all problems are solved optimally using the tabu search algorithm. Even for the few problems that are not solved optimally, the relative error is very small. Compared with the simulated annealing of Mittenthal et al. [8], a large improvement is clearly obtained by tabu search. There is no single problem solved by simulated annealing that is not solved by tabu search with the initial V-shaped schedule. As problem size increases both methods become less accurate in obtaining optimal solutions.

Among all methods of solution, the tabu search method with a V-shaped initial solution is better by far than the other methods. All problems, except three, are solved with a total error of 0.001%. The LPT initial solution is the next best with a total error of 0.009%. The SPT initial solution is the worst for both simulated annealing and tabu search methods. For the simulated annealing method, using a V-shaped starting solution produces better results in terms of the number of problems solved optimally, while the LPT initial solution is better in terms of total error. For the tabu search algorithm, the V-shaped starting solution produces better results in terms of total error.

Table 1
Percentage deviation from minimum

Pb no.	Optimal sequence	Minimum	Mittenthal results			Tabu search results		
			LPT	V	SPT	LPT	V	SPT
1	21 19 6 2 9 12	218.47	0	0	0	0	0	0
2	82 21 9 6 3 2 65	918.28	0	0	0	0	0	0
3	16 9 8 4 2 6 7 10	187.23	0	0	0	0	0	0
4	16 12 9 8 1 2 10 13	254.23	0	0	0	0	0	0
5	19 18 10 8 5 1 2 9 13 16	486.40	0.0082	0	0.1727	0	0	0
6	92 82 23 21 2 3 6 9 34 65	3584.0	0	0	0	0	0	0
7	88 61 52 22 17 6 12 33 40 70	7027.96	0	0.0973	0	0	0	0
8	99 98 50 37 22 12 14 26 78 79	12269.76	0	0	0	0	0	0
9	94 79 72 68 34 11 37 48 73 82	20903.36	0	0	0	0	0	0
10	96 76 48 46 34 10 39 41 52 70	14094.01	0.0028	0	0	0	0	0
11	98 77 62 52 42 10 44 48 65 75	18884.80	0	0	0	0	0	0
12	98 83 80 32 26 24 14 11 3 20 25 29 39 47 90	16052.20	0.0302	0.0066	0.0475	0	0	0
13	97 77 72 59 45 27 24 16 6 20 35 41 63 64 91	27082.80	0	0	0	0	0	0.0625
14	97 77 63 58 55 43 18 13 5 26 40 50 57 62 91	29805.39	0.0002	0	0.0053	0	0	0.1098
15	98 95 92 79 68 61 42 7 15 30 64 73 82 83 97	57713.76	0.0033	0.0024	0.0036	0	0	0.1662
16	96 85 77 67 51 40 26 16 7 32 42 52 59 82 84	36005.18	0	0.0002	0.0000	0	0	0.1570
17	99 97 77 73 71 58 49 46 32 9 8 17 26 44 57 59 61 75 86 88	70843.38	0.0222	0.0192	0.0031	0.0080	0	0.0009
18	97 90 88 84 75 70 56 36 30 9 2 13 21 43 47 74 77 83 87 92	76050.63	0.0002	0.0001	0.0006	0	0.0006	0.0034
19	97 94 77 71 69 48 46 29 27 20 7 9 28 38 40 53 63 74 79 89	58912.59	0.0016	0	0.0038	0.0010	0	0.0036
20	82 74 73 69 60 55 46 44 22 4 2 14 25 35 47 56 63 66 72 77	54589.11	0.0025	0.0028	0.0027	0	0.0001	0.0019
21	94 92 78 65 63 41 40 31 27 10 6 14 24 36 37 47 48 70 83 89	50015.14	0.0052	0.0016	0.0028	0.0001	0.0003	0.0071
	Total		0.0764	0.1302	0.2421	0.0091	0.0010	0.5124

6. Conclusion

In this paper, we have described and developed a tabu search approach that can be used for solving single machine scheduling problems that are known to have the V-shaped property. The developed algorithm was used to solve the single machine variance minimization problem. Computational results show great success in solving problems reported in the literature. Most of the problems were solved optimally. Even for problems that were not solved optimally the solution was only 0.001% away from the optimal solution. The algorithm was compared to the simulated annealing method and showed superiority in terms of the number of problems solved optimally, and also in terms of the total relative deviation from the optimal solution. Further research may be conducted for improving the performance of the algorithm by selecting the parameters optimally.

Acknowledgements

The authors gratefully acknowledge the support of King Fahd University of Petroleum and Minerals.

References

[1] Al-Turki UM, Mittenthal J, Ragavachari M. A dominant subset of V-shaped sequences for a class of single machine sequencing problems. European Journal of Operational Research 1996;88:345–7.
[2] Schrage L. Minimizing the time-in-system variance for a finite job set. Management Science 1975;31:540–3.
[3] Elion S, Chowdhury IC. Minimizing the waiting time variance in the single machine problem. Management Science 1977;23:567–75.
[4] Baker KR, Scudder GD. Sequencing with earliness and tardiness penalties: a review. Operations Research 1990;38:22–36.
[5] Raghavachari M. Scheduling problems with non-regular penalty functions: a review. Operations Research 1988;25:144–64.
[6] Mittenthal J, Ragavachari M. Stochastic single machine scheduling with quadratic Early-Tardy penalties. Operations Research 1993;41:786–96.
[7] Vani V, Raghavachari M. Deterministic and random single machine sequencing with variance minimization. Operations Research 1987;35:111–20.
[8] Mittenthal I, Ragavachari M, Rana AI. A hybrid simulated annealing approach for single machine scheduling problems with nonregular penalty functions. Computers and Operations Research 1993;20:103–11.
[9] Glover F. Future paths for integer programming and linkage to artificial intelligence. Computers and Operations Research 1986;13:533–49.
[10] Glover F. Tabu Search – Part I. ORSA Journal on Computing 1989;1(3):190–206.
[11] Glover F. Tabu Search – Part II. ORSA Journal on Computing 1990;2(1):4–32.
[12] Hansen P. The steepest ascent mildest descent heuristic for combinatorial programming. Congress on Numerical Methods in Combinatorial Optimization, Capri, Italy, 1986.
[13] Hansen P, Jumard B. Algorithm for the maximum satisfiability problem. RUTCOR Research Report RR No. 43–87, Rutgers, New Brunswick, NJ, 1987.
[14] Nowicki E, Smutnicki C. A fast taboo search algorithm for the flow shop. Report ICT PRE 8/94, Technical University of Wroclaw, 1996.
[15] Taillard E. Some efficient heuristic methods for the flow shop sequencing problem. European Journal of Operations Research 1990;47:65–74.

[16] Laguna M, Kelly JP, Gonzalez-Velarde JL, Glover F. Tabu search for the multilevel generalized assignment problem. European Journal of Operations Research 1995;82:176–89.
[17] Bland JA, Dawson JP. Tabu search and design optimization. Computer-aided Design 1991;23(3):195–201.
[18] Hertz A. Tabu search for large scale time tabling problems. European Journal of Operations Research 1991;51:39–47.
[19] Skorin-Kapov D, Skorin-Kapov J. On tabu search for the location of interacting hub facilities. European Journal of Operations Research 1994;73:502–9.
[20] Gupta MC, Gupta YP, Bector CR. Minimizing the flow time variance in single machine systems. Journal of the operational Research Society 1990;41(8):767–79.
[21] Al-Turki UM. Minimizing flow time variance in scheduling operations. The Fifth Saudi Engineering Conference, Umm Al-Qurah University Mekkah Al-Mukarama, Saudi Arabia, vol. 4, 1999. p. 207–11.

Umar Al-Turki is an assistant professor in the Department of Systems Engineering, King Fahd University of Petroleum and Minerals (KFUPM), in Saudi Arabia. He received his B.Sc. and M.Sc. degrees in systems engineering from KFUPM in 1982 and 1986, respectively. He was awarded the M.Sc. degree in industrial engineering and management from Oklahoma State University and a Ph.D. in decision sciences and engineering systems from Rensselaer Polytechnic Institute. His research interests include production and manufacturing systems, scheduling, and quality control.

Chawki Fedjki received an ‘Ingeniorat d'etat” in computer science from University of Constantine, Algeria, an M.Sc. degree in operations research from Carnegie-Mellon University. He worked as a lecturer at the University of Setif (Algeria) and at the University of Biskra (Algeria). Since September 1994, he has been at the Department of Systems Engineering at KFUPM as a Ph.D. student. Mr Fedjki's interests are in the areas of mathematical programming and combinatorial optimization.

Abdulbasit Andijani is an associate professor and the chairman of the Department of Systems Engineering, King Fahd University of Petroleum and Minerals (KFUPM), in Saudi Arabia. He received the Ph.D. degree in industrial and systems engineering from Ohio State University. His research interests include design and analysis of inventory and production systems, quality control, simulation models, and stochastic systems.