◈ **Application Paper**

A Tabu Search Methods for Minimizing Mean Tardiness in Parallel Machines Scheduling

Chun Tai-Woong*
Park Hai-Chun*

Abstract

In this paper we consider to parallel machines scheduling problems for minimizing mean tardiness that is known NP-complete. This problems is classified into two cases, one of which is the case which processing time are identical and the other, nonidentical. A Tabu Search method is applied to the problems considered in this paper to get an improved solution. To this end, we design move attribute, Tabu attribute and Tabu tenure, and thereafter perform the experiments to the problems.

1. Introduction

In this paper we consider to parallel machines scheduling problems that occurs the bottleneck in the manufacturing system, parallel processing in the computer system. This problems is classified into two cases, one of which is the case which processing time are identical and the other, nonidentical. The performance of scheduling in this paper is mean tardiness. This problem, which has been studied for a long time[1,3,6,8], that is known NP-complete. Therefore the use of heuristic algorithm is the resonable better than optimal algorithm in application of real problem. Ho-Chan[7] proposes TPI(Traffic Priority Index) priority that compose the EDD(Earliest Due Date) rule and SPT(Shortest Processing Time) rule that is applied to minimizing mean tardiness scheduling in identical parallel machines problems.

A Tabu Search method[2, 11] is applied to the problems considered in this paper to get an improved solution. To this end, we design move attribute, Tabu attribute and Tabu tenure, and thereafter perform the experiments to the problems. Section 2 gives the representation of this problem in mathematical model, section 3 presents an implementation of Tabu Search for this problem. In section 4, we discuss the effectiveness of this method.

2. Mathematical Model

To develope the mathematical model necessary for this research, we make the following assumptions:

a) there are n jobs and m machines

b) the processing time (P_{ki} : when job k is processed to i th machine) and due date D_{ki} $(k=1,\dots, n)$ are fixed

*Dept. of Industrial Eng., Chosun Uni.,

c) no two job processed at same time on the same machine.
d) a start of all jobs could be time zero.

Mathematical Model

$$\text{minimize} \quad \overline{T} = \frac{1}{n}\sum_{k=1}^{n} T_k \tag{1}$$

subject to

$$T_k = \max(0,\ C_k - D_k) \quad 1 \le k \le n \tag{2}$$

$$C_k = \sum_{i=1}^{m}\sum_{j=1}^{n} C_{ij} \cdot X_{kij} \quad 1 \le k \le n \tag{3}$$

$$C_{ij} = \sum_{i=1}^{m}\sum_{k=1}^{n} P_{ki} \cdot X_{kij} \quad 1 \le i \le m, 1 \le j \le n \tag{4}$$

$$\sum_{k=1}^{n} X_{kij} \le 1 \quad 1 \le i \le m, 1 \le j \le n \tag{5}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{n} X_{kij} = 1 \quad 1 \le k \le n \tag{6}$$

$$X_{kij} = \begin{cases} 1, \text{ if } \textit{job k is assigned to the jth position of machine i} \\ 0, \textit{elsewhere.} \end{cases}$$

where,

C_k: *the completion time of job k.*

Eq.(1) is object function, *Eq.*(2) is a tardiness of job k that is presented to nonlinear regarding C_k. *Eq.*(3)and *Eq.*(4) is that the completion time of job k. *Eq.*(5) is restriction for job k is not job processed at same time on the same machine, *Eq.*(6) is restriction condition for job k is must be assigned one machine to j th position.

Branch & bound methods have the advantage of providing an optimal solution to this problem. Unfortunately, the computing times becomes prohibitive when then number of operations exceeds few hundred. Therefore the use of heuristic algorithm is the resonable better than optimal algorithm in application of real problem. The minimum job load assignment method (MJLA) as heuristic method of parallel machines scheduling in generally has been known is that: first step is that schedule to all job n like as one machine scheduling, and second step is the first order job by the results of one step assign the minimum load machine, and third step is replicated second step until all job has been assigned.

3. Tabu Search Method for Parallel Machines Scheduling

3.1 Move

Move is the generation of new solution s' from current solution s. In this study, the move is two type, one is swap move and the other is transfer move. The swap move is mutually exchange to the jobs that is assigned to two machine(Fig. 1). Fig. 1 is swap move example that show J_{11} and change of J_{22}. If a job of n is exchanged, the neighborhood $N(s)$ is $n^2 - \sum_{n=1}^{m} n_i^2$. The transfer move is that one job in one machine transfer the order machine, therefore the current job distribution structure is changed(Fig. 2). If a number of job n is transfer, the neighborhood is $n(m-1)$.

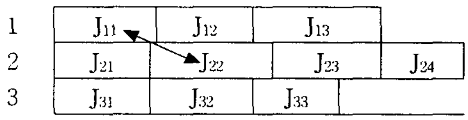

Fig. 1 swap move

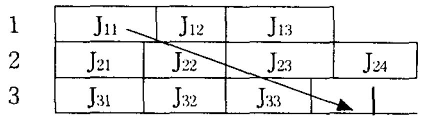

Fig. 2 transfer move

The cardinal of neighborhood by the swap and transfer move in this paper is Eq.(7).

$$N(s) = n^2 - \sum_{n=1}^{m} n_i^2 + n(m-1) \tag{7}$$

3.2 Restricted Neighborhood

The cardinal of neighborhoods is larger, the possibility of involving the best solution in neighborhoods is extended. But computational time is rapidly increases. Therefore we use to restricted neighborhood for reducing computational time. In this paper, the restricted neighborhood is that: when the job J_{yj} $(y=1,\dots, m,\ y \neq x)$ and job $J_{xj}(x=1,\dots, m)$ is swap, considering to position of job to be swapped in x, y machine Therefore the cardinal of restricted neighborhoods $N_{r_e}(s)$ are Eq.(8)

$$N_{r_e}(s) = 3n(m-1)/2 \tag{8}$$

3.3 One Machine Scheduling

The real value of current solution after move(s->s') is determined when all machine's job order has been optimized. But the problem for minimizing mean tardiness in single machine is NP-complete. Therefore EDD, SPT, Wilkerson and Irwin[12], Ho and Chang[7] methods could used for one machine scheduling methods. We proposes to insert method for one machine scheduling that is: at first step make n different schedule by inserting job J_{xj} in front of job($J_{x1},\dots, J_{xj},\dots, J_{xn}$. $j=1,\dots,n$) and compute to solution value to each schedule. Second step is select to the best solution among n solution.

3.4 Tabu Tenure

Dynamic Tabu tenure that is varied the size of tabu list according to algorithm execution results is used in this study. In order to the determination of tabu tenure parameter with maximum TT_{max} and minimum TT_{min}, we experiments to example problems which is m=3, 4 ,5 and n=30, 50, 60, 80, 100. And so the problems has been replicated the number of twenty times, TT_{max} and TT_{min} is determined according to the average value and N(s). Moreover considering to N(s), Tabu tenure is determined to proportion to

$\sqrt{n \times m}$($\alpha \sqrt{n \times m}$, $\beta \sqrt{n \times m}$, where α, β is the parameter for represent to tabu tenure range). In this paper, Tabu attribute is job number, and the upgrade of Tabu list be used to cyclical method. Under condition if current solution is tabu solution, and if the value of current solution is less than the value of best solution, aspiration level is satisfied. Maximum replication times for stopping rule is $3 \times n$.

3.5 Experiments

The data in this paper is generated by Eq. (9) and Eq. (10)[7]. Because we compare to Ho-Chang study. A type of problems are the case of m=3, n=30, 60, the case of m=4, n=60, 80, the case of m=5, n=50, 100.

$$P_{k_i} = Uniform[1, x] \quad (9)$$

$$D_i = Uniform[1, 2np/4.5q] \quad (10)$$

Where, x is the maximum value of processing, we set to 25. And q is the tightness of due date(1,2,3,4,5).

The Tabu Search Algorithm for Parallel Machines Scheduling

Step 0. Read(n, m, T_{max}=3×n, α, β), and T_{iter} = 0.
Step 1. Generate to current solution s take place of initial solution .
Step 2. Generate to neighborhood N(s), s->s'.
Step 3. Carry out one machine scheduling(section 3.3).
Step 4. if s' is not Tabu solution, go to step 6.
Step 5. if s' is not satisfied to aspiration level, go to 7.
Step 6. To determine wether best candidate solution.
Step 7. if all neighbor solutions are examinate, then go to 8,else go to 2.
Step 8. Current solution take place of best candidate solution. Upgrade to tabu list, and $T_{iter}=T_{iter}+1$.
Step 9. if T_{iter} less than T_{max}, go to 2.
*Step*10. Stop.

4. Experiment and Results

4.1 Tabu Tenure

As the results of pilot experiments in section 3.4, the Tabu tenure range is determined that is shown in table 1.

Table 1. Tabu Tenure

the type of problem	Tabu tenure range
identical machines	$1.0\sqrt{n \times m}$ ~ $1.5\sqrt{n \times m}$
nonidentical machines	$1.0\sqrt{n \times m}$ ~ $1.5\sqrt{n \times m}$

4.2 Identical Parallel Machines Scheduling Problem

The Tabu search method in this paper is compared to such heuristic algorithm as EDD-P, WI-P, EDD-Tabu and WI-TABU, HC-TABU. EDD-P is EDD rule for one machine scheduling and MJAL method for parallel machine scheduling, WI-P is Wilkerson and Iron[12] for one machine scheduling and MJAL method for parallel machine scheduling, EDD-Tabu is EDD rule for one machine scheduling and Tabu search for parallel machine scheduling, WI-TABU is Wilkerson and Iron for initial solution and Tabu search for parallel machine scheduling, HC-TABU is Ho-Chang for initial solution and Tabu search for parallel machine scheduling. The results of experiments is presented in table 2.

Table 2. The Results of Experiments in Identical Parallel Machines

m	n	EDD-P	WI-P	Ho-Chang	EDD-TABU	WI-TABU	HC-TABU
3	30	26.28	22.99	22.58	22.50	22.43	22.43
3	60	70.56	49.21	47.05	46.27	46.84	46.84
4	60	43.48	31.53	29.28	28.89	29.03	29.03
4	80	72.37	51.07	47.61	47.66	48.51	48.51
5	50	29.06	23.08	22.61	22.59	22.45	22.45
5	100	71.15	50.15	46.60	46.47	46.97	46.97
*solution		52.63	36.45	35.95	35.73	35.85	35.85
**improvement		0.000	0.270	0.296	0.302	0.299	0.299

* solution is the value of average solution.

** the rate of improvement =1-heuristic algorithm/EDD-P

As shown table 2, the value of average solution derived by the Tabu Search method in this paper better than that EDD-P and HO-Chang by 30.2% and 1.62%, respectably.

4.3 Nonidentical Parallel Machines Scheduling Problem

The use of algorithm for comparing to heuristic algorithm in nonidentical parallel machines scheduling problem is like as section 4.2. The results of experiments is presented in table 3.

Table 3. The Results of Experiments in Nonidentical Parallel Machines

m	n	EDD-P	WI-P	EDD-TABU	WI-TABU
3	30	9.27	6.56	4.49	4.58
3	60	23.38	14.49	8.64	8.78
4	60	7.25	4.98	2.96	2.55
4	80	10.55	7.86	3.60	3.83
5	50	2.09	2.17	0.82	0.83
5	100	3.57	3.11	1.47	1.44
*solution		9.35	6.53	3.66	3.66
**improvement		0.00	0.30	0.60	0.60

* solution is the value of average solution.

** the rate of improvement =1-heuristic algorithm/EDD-P

As shown in table 3, the value of average solution derived by the Tabu Search method in this paper better than that EDD-P and Wilkerson and Iron by 60% and 30%, respectably.

5 Conclusion

The parallel machines scheduling problems with mean tardiness is one of the combinatorial optimization problems that often occurs in the real world. This study proposes Tabu Search methods for solving parallel machines scheduling problems related to minimizing mean tardiness and experimented. The results of experiments, in the case of identical parallel machines, Tabu Search method in this paper better than EDD-P and HO-Chang by 30.2% and 1.62%, respectably. Tabu Search method in the case of nonidentical parallel machines better than that EDD-P and Wilkerson and Iron by 60% and 30%, respectably.

References

[1] Baker, K. R., *Introduction to Sequenceing and Scheduling*, John Wiley & Sons, Inc., New York, 1974.

[2] Barnes, J. W. and M. Laguna, "A Tabu Search Experience in Production Scheduling," *Annals of Operations Research*, Vol. 41, pp. 141-156, 1993.

[3] Chen, T. C. E. and C. C. S. Sin, "A State of the Art Review of Parallel Machine Scheduling Research," *European Journal of Operational Research*, Vol. 47, pp. 271-292, 1990.

[4] De, P. and T. E. Morton, "Scheduling to Minimize Makespan on Unequal Parallel Processors," *Decision Sciences*, Vol. 11, pp. 586-601, 1980.

[5] Dileepan, P. and T. Sen, "Job Tardiness in Unequal Parallel Processor Systems," *Journal of Information & Optimization Sciences*, Vol. 10, No. 3, pp. 579-590, 1989.

[6] Hariri, A. M. A. and C. N. Potts, "Heuristics for Scheduling Unrelated Parallel Machines," *Computers & Operations Research*, Vol. 18, No. 3, pp. 323-331, 1991.

[7] Ho, J. C. and Yih-Long Chang, "Heuristic for Minimizing Mean Tardiness for m Parallel Machines," *Naval Research Logistics Quartely*, Vol. 38, pp. 367-381, 1991.

[8] Ho, J. C. and Yih-Long Chang, "Minimizing the Number of Tardy Jobs for m Parallel Machines," *European Journal of Operational Research*, Vol. 84, pp. 343-355, 1995.

[9] Horowitz, E. and S. Sahni, "Exact and Approximate Algorithms for Scheduling Nonidentical Processors," *Journal of the Association for Computing Machinery*, Vol. 23, No. 2, pp. 317-327, 1976.

[10] Ibarra, O. H. and C. E. Kim, "Heuristic Algorithm for Scheduling Independent Tasks on Nonidentical Processors," *Journal of the Association for Computing Machinery*, Vol. 24, No. 2, pp. 280-289, 1977.

[11] Reeves, C. R., *Mordern Heuristic Techniques for Combinatorial Problems*, John Wiley & Sons, Inc., New York, 1993.

[12] Wilkerson, J. and J. D. Irwin, "An Improved Algorithm for Scheduling Independence Tasks," *IIE Transactions*, Vol. 3, pp. 239-245, 1971.