A Variable Neighborhood Search for the Single Machine Total Stepwise Tardiness Problem

Chao-Tang Tseng[1], Ying-Chun Chou, and Wan-Yin Chen

Abstract

A new type of tardy cost, called stepwise tardiness, is considered in a single machine problem. The stepwise tardiness scheduling has been found in many practical situations. To the best of our knowledge, there exist few studies directly addressing the problem. In this paper, we develop a heuristic for the single machine total stepwise tardiness (SMTST) problem. In addition, a variable neighborhood search (VNS), which is a novel meta-heuristic, is proposed to improve the quality of solution. The proposed VNS algorithm includes a new encoding scheme and three new neighborhood structures. The experimental results show that the proposed VNS algorithm can obtain a good performance for the considered problem.

Keywords: Single machine, Scheduling, Total stepwise tardiness, Variable neighborhood search;

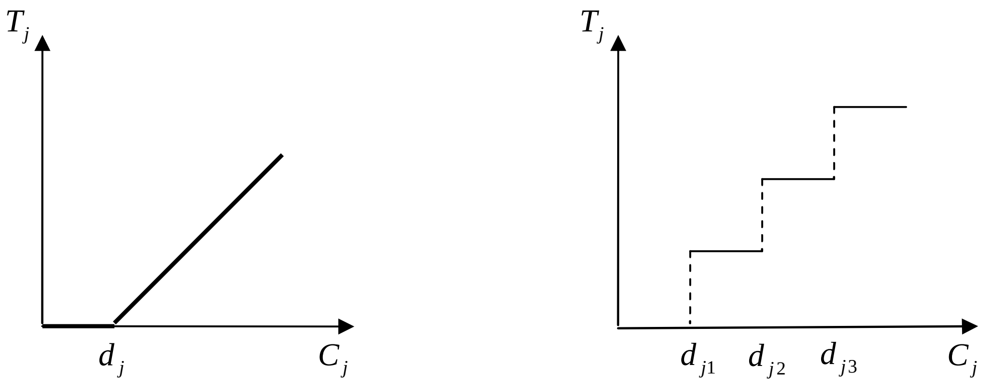
Introduction

Nowadays, scheduling problems with due date-related criteria have more practical meaning than before, because most companies are concerned with meeting customer's demand in terms of due dates (Sen and Gupta, 1984; Abdul-Razaq et al., 1990; Koulamas, 1994). A tardy job may result in customer dissatisfaction, contract penalties, loss of sale, and loss of reputation among others. Therefore, the tardiness problem is always an important issue. All the studies on this subject assume that the tardiness of jobs is either linear or two-piecewise function. For the linear type as shown in Figure 1(a), the tardiness cost is a linear function of the completion time of a job. Then, a major concern for this type is to minimize the total tardiness of all jobs. As for the two-piecewise type, it assumes a specific amount of tardy cost will be charged whenever a job is late, no matter how later its completion time is than its due date. When such a tardy cost is concerned, the problem of minimizing the total tardiness cost is transformed into that of minimizing the weighted number of tardy jobs. The two types of problems on a single machine are both NP-hard problems.

In recent years, a new type of tardy cost, called stepwise (multiple-piecewise) tardiness, has received attention. As shown in Figure 1(b), the stepwise type arises from when multiple due dates are allowed for a job. Every time a job misses one due date, its tardy cost is most likely to have a jump, which is at a bigger level with the increasing lateness of the due date. The stepwise tardiness scheduling has been found in many practical situations. In an earlier project management reference, Birge and Maddox (1995) discuss this type of tardy cost. Curry and Peters (2005) pointed out that the stepwise tardiness happens in many modes of transportation, such as less than truckload, expedited ground, and airfreight forwarding. They applied a simulation approach for the parallel machine rescheduling problem with the stepwise tardiness and machine assignment stability criteria. Later, Sahin

[1] Assistant Professor, Department of Industrial Engineering and Management, Chaoyang University of Technology, 168, Jifong E. Rd., Wufong Township, Taichung County, 41349, Taiwan Tel.+886-4-23323000 ext. 4677 Fax.+886-4-23742327 Email. cttseng@cyut.edu.tw

(2006) also examined the stepwise tardiness criterion on the yard operations problem in railroad.



(a) (b)

Figure 1. Traditional Tardiness and Stepwise Tardiness of Job *j*

In this paper we consider the single machine total stepwise tardiness (SMTST) problem. The single machine weighted number of tardy jobs (SMWNT) problem is a special case for the addressed problem if there only exists one due date for each job. Karp (1972) proved that the SMWNT problem is NP-hard, even if all jobs are constrained by a common due date. In consequence, several exact methods based on dynamic programming or branch-and-bound have been presented, such as Villarreal and Bulfin (1983), Potts and Van Wassenhove (1988), Tang (1990) and M'Hallah and Bulfin (2003). For the special case that all jobs have the same weight for their lateness, Moore (1968) developed a polynomial algorithm to minimize the number of tardy jobs on a single machine. Recently, Wu et al. (2009) developed a heuristic for SMTST problem. Yang (2009) considered the multiple common due dates for SMTST problem and developed a branch-and bound.

Because of the essential complexity of the single machine total stepwise tardiness problem, it is difficult to use an exact method to solve the problem. As an alternative, the use of metaheuristic can obtain the near optimal schedules in reasonable. Variable neighborhood search (VNS), proposed by Mladenović and Hansen (1997), is a novel metaheuristic based on the principle of systematic change of neighborhood during the search. VNS differs from most local search heuristics in that it uses two or more neighborhoods, instead of one, in its structure. It has been proved to be a simple and effective method for solving scheduling problems, including single machine scheduling problem (Liao and Cheng, 2007; Wang and Tang, 2008), parallel machine scheduling problem (De Paula et al., 2007; Chen and Chen, 2008), flow shop scheduling problem (Blazewicz et al., 2008) and job shop scheduling problem (Aydin and Sevkli, 2008). A survey of studies on VNS can be found in Hansen et al. (2009).

The remainder of this paper is organized as follows. In Section 2, we formally define the single machine total stepwise tardiness problem. A heuristic based on Moore's algorithm (Moore, 1968) is developed to provide an initial solution to the addressed problem in Section 3. To further improve the solution quality, a VNS algorithm with a new encoding scheme and a memory-based neighborhood structure is proposed in Section 4. Section 5 evaluates the performances of the proposed heuristic and VNS algorithm through extensive computational experiments. Finally, conclusions are provided in Section 6.

Problem Definition

In this section we formally define the problem considered in this paper. The following notation is used throughout the paper:

n	number of jobs
m	number of due dates
J_j	job j $(j = 1, 2, ..., n)$
p_j	processing time of job j
d_{jk}	k th due date of job j $(k = 1, 2, ..., m)$
w_{jl}	tardiness cost of job j in the l th late period $(l = 1, 2, ..., m)$
C_j	completion time of job j
T_j	tardiness of job j
π	a processing sequence of jobs
S	partial schedule
U	set of unscheduled jobs

The scheduling problem considered in this paper can be stated as follows. A set of n jobs, all available for processing at time zero, is to be processed on a continuously available single machine. The machine can process only one job at a time. Each of the n jobs has a processing time p_j and m due dates, denoted by d_{jk} ($k = 1, 2, ..., m$). The tardiness of job j is defined as

$$T_j = \begin{cases} 0 & if \;\; C_j \le d_{j1} \\ w_{j1} & if \;\; d_{j1} < C_j \le d_{j2} \\ w_{j2} & if \;\; d_{j2} < C_j \le d_{j3} \\ \vdots & \qquad \vdots \\ w_{jm} & if \;\; C_j > d_{jm} \end{cases} \tag{1}$$

where w_{jl} denotes the tardiness cost of job j in the l th late period. The objective of the problem is to determine a sequence π that minimizes the following cost function:

$$f(\pi) = \sum_{j=1}^{n} T_j \tag{2}$$

The addressed problem is NP-hard since the single machine weighted number of tardy jobs problem, which is NP-hard, is its special case.

Proposed Heuristic

In this section, we develop a heuristic based on the algorithm of Moore (1968), which efficiently solves the single machine the number of tardy jobs problem. The basic idea of the proposed can be described as follows. At the beginning of the heuristic, the jobs are ordered in non-decreasing order of their first due dates (i.e., EDD rule). Set *S* denotes the set of scheduled jobs (or the partial schedule). First, the jobs completed by their first due dates are sequentially put into *S*. Then, we use the concept of the smallest tardiness cost to determine whether a considered job should be scheduled. We consider each job, which does not belong to *S*, inset it into each position of S , and computer the tardiness cost. The job of the smallest tardiness cost will be scheduled in the selected position of *S*. The procedure is continued until all jobs are considered. We now present the steps of the heuristic as follows:

Step 1. Set $S = \phi$ and $U = \{J_1, J_2, ..., J_n\}$.

Step 2. Obtain a processing order π according to the earliest first due date (EDD) rule, put the jobs completed by their first due dates into S, and delete them from U. Do not change the relative positions the jobs with respect to each other in S.

Step 3. Pick each job of U and insert it at all possible position in the partial schedule S, respectively. Assign the job and position of the smallest tardiness cost into S and delete it from U.

Step 4. If $U = \phi$, STOP, otherwise go to Step 3.

Proposed VNS Algorithm

In this section, we design a new encoding scheme to represent a feasible solution for the addressed problem and proposed three neighborhood structures for the VNS algorithm. In particular, one of the neighborhood structures is based on memory-based mechanism. The details are given in what follows.

Encoding

Our proposed VNS algorithm designs a "Position to Late Period" representation for the feasible solution. We define a solution as $X = (x_{10}, x_{11}, ..., x_{nm})$, $x_{il} \in J_j$ where x_{il} equals J_j if job j is placed in the ith position of the in the lth late period and 0 otherwise. It is noted that there are m late period defined by a serious of due dates as given in Eq. (1) and $l = 0$ when the job is completed early. For example, suppose the job sequence is (3612574) and there are 3 due dates for each job. The completion times of jobs 3, 6, 1, and 2 are before their first due dates, and the completion times of jobs 5, 7, 4 are after their third due dates. By this definition, we have $x_{10} = 3$, $x_{20} = 6$, $x_{30} = 1$, $x_{40} = 2$, $x_{13} = 5$, $x_{23} = 7$, $x_{33} = 4$, and all other $x_{il} = 0$ (see Figure 2).

$$\text{Late period } (l)$$

$$\text{Position } (i) \quad \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 3 & 0 & 0 & 5 \\ 6 & 0 & 0 & 7 \\ 1 & 0 & 0 & 4 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Figure 2. "Position to Late Period" Representation

Neighborhood Structures

The following neighborhood structures are considered in the proposed VNS algorithm:

Interchange neighborhood

Two lateness periods are selected and the schedules are obtained by swapping each pair of jobs from two different selected periods.

Insertion neighborhood

One lateness period is selected and the schedules are obtained by removing a job in a position and inserting it in another position.

Memory base neighborhood

Use a memory array which records the total priority of each job to generate a set of neighborhoods. We define a memory array as $M = (r_1, r_2, ..., r_n), \quad r_j \in R,$ where r_j represents the total priority of job *j*. The priority of each job is obtained from the processing sequence of jobs (the move solution) at each move step of VNS. We add 30 and 10 to the total priorities of the first 30% and last 40% jobs in the obtained sequence, respectively. To generate the neighborhoods, we select the jobs of the first 30% and last 40% priorities from the memory array. The neighborhoods are obtained by removing the selected job in its position and inserting it in the first position of the candidate solution.

Local Search

We make use of a similar interchange neighborhood to implement the local search and apply the best improvement strategy to move to the next feasible solution. The neighborhood contains all of the schedules obtained by swapping all possible pairs of positions in the certain late period. The steps of the local search are given as follows:

Step 1. Find an initial schedule x and set $i = 0$.

Step 2. Consider the i th late period and obtain the best schedule x' from the neighborhood of x.

Step 3. If x' is better than x, then set $x \leftarrow x'$, $i \leftarrow i+1$ and return to Step 2, else STOP.

The Proposed VNS Algorithm

The steps of our VNS structure are described below:

Step 1. Find an initial solution s and set $\eta = 1$.

Step 2. Repeat the following steps until *N* accumulative no-improving iterations:

(i) Generate a schedule s' at random from the η th neighborhood of s.

(ii) Local search: Apply the proposed local search with s' as initial solution and obtain the local optimum s''.

Step 3. If s'' is better than the incumbent, then update $s = s''$ and set $\eta = 1$, else $\eta = \eta + 1$. If $\eta \leq 2$, return to (i); otherwise return to Step 2.

Computational Results

In this section, we evaluate the proposed heuristic and VNS for the single machine total stepwise tardiness problem. All the algorithms were coded in Dev C++ and run on a Pentium IV 3.0 GHz PC with 1GB memory.

The problem instances we generated according to Wu et al. (2007) are defined as follows. There are five different numbers of jobs ($n = 10, 20, 30, 40, 50$), where the processing time (p_i) and the first due date (d_{j1}) were drawn from discrete uniform distributions $\mathrm{U}(1, 99)$ and $\mathrm{U}(p_j, p_j + n)$, respectively. Each job has 3 due dates. The distance between due dates was randomly generated from discrete uniform distribution $\mathrm{U}(1, n)$. There are 10 instances for each problem with a total of 50 instances, each of which was tested for 10 trials. All comparative experiments were conducted on the above instances.

The relative percentage deviation (RPD) is used as the performance measure. Let A_i represent the total stepwise tardiness associated with Algorithm i . The RPD of Algorithm i is defined as:

$$\frac{A_i - \min(A_j, \forall j)}{\min(A_j, \forall j)} \times 100 \tag{3}$$

Because the objective function is to be minimized, the smaller the RPD value is the better the algorithm.

To verify the performance of the proposed heuristic and VNS algorithm, we will make a comparison with a heuristic, a VNS, and a basic tabu search, presented by Wu et al. (2008). The computational results are summarized in Table 1, which gives the mean RPD (MRPD) and Average MRPD (Avg.) for the heuristic of Wu et al. (H_W), the proposed heuristic (H_T), the proposed VNS algorithm (VNS_T), VNS algorithm of Wu et al. (VNS_W) and basic tabu search (TS_B). It is noted that, for $n = 10$ problems, the solutions obtained by VNS_T and VNS_W are equal to optimal solutions yielded by total enumeration. All computation times of two heuristic are smaller than 0.1 seconds, while the average computation time (t) of other algorithms is presented in Table 1.

It is observed from Table 1 that the proposed heuristic outperforms the heuristic of Wu et al. (2008). The average MRPD of the proposed heuristic is 5.04%. The results show that the proposed heuristic is a useful method for solving the addressed problem. In addition, it is also clear from table that the proposed VNS algorithm yield solutions of superior quality, as against the VNS algorithm of Wu et al. and the basic tabu search. Therefore, the proposed VNS algorithm is competitive.

Table 1. Comparison of Different Algorithms

Job	H_W	H_T	VNS_T	VNS_W	TS_B	t (seconds)
10	18.41	4.56	0.00	0.00	3.31	10
20	31.32	4.42	1.38	0.84	17.29	20
30	28.19	2.74	0.17	0.24	14.72	40
40	27.68	5.75	2.07	0.02	14.68	80
50	29.73	7.76	1.73	1.03	16.22	120
Avg.	26.99	5.04	1.18	0.43	13.24	54

Conclusions

The stepwise tardiness scheduling has been found in many practical situations. To the best of our knowledge, there exist few studies directly addressing the single machine with total stepwise tardiness criterion. In this paper, we have developed a heuristic, which can provide a good solution in a short computation time. To further improve the solution quality, we also have proposed a VNS algorithm, which incorporates a "Position to Late Period" representation and the interchange, insertion, and memory-based neighborhood structures, to search for a near-optimal solution. In addition, a similar interchange local search has been presented to implement the local search of VNS. The computational results have demonstrated the superiority of the proposed VNS over both VNS and TS proposed by the literature. There is a limitation of the proposed VNS. That is, although the proposed VNS is competitive, it only provides a good performance for the specific problem considered in this paper. Future research may be conducted to further investigate the applications of VNS to other scheduling problems. It is also worthwhile to design other metaheuristics to continue pursuing the best performance of VNS in solving the scheduling problems considered in this paper.

References

Abdul-Razaq, T.S., Potts, C.N. and Van Wassenhove, L.N., 1990. A survey of algorithms for the single machine total weighted tardiness scheduling problem. *Discrete Applied Mathematics*, 26, 235-253.

Aydin, M.E. and Sevkli, M., 2008. Sequential and parallel variable neighborhood search algorithms for job shop scheduling. *Metaheuristics for Scheduling in Industrial and Manufacturing Applications*, 128, 125–144.

Birge, J.R. and Maddox, M.J., 1995. Bounds on expected project tardiness. *Operations Research*, 43, 838–850.

Blazewicz, J., Pesch, E., Sterna, M. and Werner, F., 2008. Metaheuristic approaches for the two-machine flow-shop problem with weighted late work criterion and common due date. *Computers and Operations Research*, 35, 574–599.

Chen, C.L. and Chen, C.L., 2009. Hybrid metaheuristic for unrelated parallel machine scheduling with sequencedependent setup times. *International Journal of Advanced Manufacturing Technology*, 43, 161-169.

Curry, J. and Peters, B., 2005. Rescheduling parallel machines with stepwise increasing tardiness and machine assignment stability objectives. *International Journal of Production Research*, 43, 3231-3246.

De Paula, M.R., Ravetti, M.G., Mateus, G.R. and Pardalos, P.M. 2007. Solving parallel machines scheduling problems with sequence-dependent setup times using variable neighbourhood search. *IMA Journal of Management Mathematics*, 18, 101-115.

Karp, R.N., 1972. Reducibility among combinatorial problems. *In:* R.E. Miller and J. W. Thatcher, eds. *Complexity of Computer Computations*. New York: Plenum Press, 85-103.

Koulamas, K., 1994. The total tardiness problem: review and extensions. *Operations Research*, 42, 1025–1041.

Liao, C.J. and Cheng, C.C., 2007. A variable neighborhood search for minimizing single machine weighted earliness and tardiness with common due date. *Computers and Industrial Engineering*, 52, 404–413.

M'Hallah, R. and Bulfin, R.L., 2003. Minimizing the weighted number of tardy jobs on a single machine. *European Journal of Operational Research*, 145, 45-56.

Mladenović, N. and Hansen, P., 1997. Variable neighborhood search. *Computers and Operations Research*, 24, 1097-1100.

Moore, J.M., 1968. One-machine sequencing algorithm for minimizing the number of late jobs. *Management Science*, 15, 102-109.

Potts, C.N. and Van Wassenhove, L.N., 1988. Algorithm for scheduling a single machine to minimize the weighted number of late jobs. *Management Science*, 34, 843-858.

Sahin, G., 2006. *New combinatorial approaches for solving railroad planning and scheduling problems*. Thesis (PhD), University of Florida, USA.

Sen, T. and Gupta, S.K., 1984. A state-of-art survey of static scheduling research involving due dates. *OMEGA, The International Journal of Management Science*, 12. 63-76.

Tang, G., 1990. A new branch and bound algorithm for minimizing the weighted number of tardy jobs. *Annals of Operations Research*, 24, 225-232.

Villarreal, F.J. and Bulfin, R.L., 1983. Scheduling a single machine to minimize the weighted number of tardy jobs. *IIE Transactions*, 15, 337-343.

Wang, X. and Tang, L., 2009. A population-based variable neighborhood search for the single machine total weighted tardiness problem. *Computers and Operations Research*, 36, 2105-2110.

Wu, T.H., Tsai, M.L., Yang, H.W. and Tseng, C.T., 2008. Minimizing the total stepwise tardiness on a single machine. *Proceedings of the CIIE Conference, Taoyuan, Taiwan.*

Yang, H.W., 2009. Maximizing total profit of jobs with stepwise non-increasing profit under multiple common due dates. *Proceedings of the 20th Annual POMS conference, Orlando, Florida*.