Available online at www.sciencedirect.com

ScienceDirect

Computers & Industrial Engineering 52 (2007) 41–56

computers & industrial engineering

www.elsevier.com/locate/dsw

Job scheduling to minimize the weighted waiting time variance of jobs ☆

Xueping Li [a,*], Nong Ye [b], Tieming Liu [c], Yang Sun [b]

[a] Department of Industrial and Information Engineering, University of Tennessee, Knoxville, TN 37996-0700, USA
[b] Department of Industrial Engineering, Arizona State University, Tempe, AZ 85287-5906, USA
[c] School of Industrial Engineering and Management, Oklahoma State University, Stillwater, OK 74078, USA

Received 3 February 2006; received in revised form 13 August 2006; accepted 13 October 2006
Available online 22 December 2006

Abstract

This study considers the job scheduling problem of minimizing the weighted waiting time variance (WWTV) of jobs. It is an extension of WTV minimization problems in which we schedule a batch of *n* jobs, for servicing on a single resource, in such a way that the variance of their waiting times is minimized. WWTV minimization finds its applications for job scheduling in manufacturing systems with earliness and tardiness (E/T) penalties, in computer and networks systems for the stabilized QoS, and in other fields where it is desirable to minimize WWTV of jobs with different weights for priorities. We formulate a WWTV problem as an integer programming problem, prove the V-shape property for agreeably weighted WWTV problems and the nondelay property for general WWTV problems, and discover the strong V-Shape tendency of the optimal job sequences for this problem. Two job scheduling algorithms, Weighted Verified Spiral (WVS) and Weighted Simplified Spiral (WSS), are developed for the WWTV problems. Numerical testing shows that WVS and WSS significantly outperform existing WWTV algorithms.

Keywords: Weighted waiting time variance (WWTV); Quality of service (QoS); Just-in-time (JIT); Job scheduling

1. Introduction

The single machine weighted waiting time variance (WWTV) minimization problem, denoting by $1\|WWTV$, is to schedule the jobs in a batch on a single machine so as to minimize the weighted waiting time variance of the jobs as follows:

$$\min_{\lambda\in\Pi} \mathrm{WWTV} = \frac{1}{n-1}\sum_{i=1}^{n} v_i(\lambda)[W_i(\lambda) - \overline{W}(\lambda)]^2 \tag{1}$$

☆ This manuscript was processed by Area Editor Maged M. Dessouky.

[*] Corresponding author. Tel.: +1 865 974 7648; fax: +1 865 974 0588.

E-mail addresses: Xueping.Li@utk.edu (X. Li), nongye@asu.edu (N. Ye), Tieming.Liu@okstate.edu (T. Liu), Yang.Sun@asu.edu (Y. Sun).

0360-8352/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cie.2006.10.010

where Π is the set of all the permutations of the n jobs, $v_i(\lambda)$ is the weight of the job on position i to indicate its priority, $W_i(\lambda)$ is the waiting time of the job on position i and $\overline{W}(\lambda)$ is the weighted average waiting time of the jobs for a given sequence λ. A greater value of v_i means a higher priority. It is an extension of the WTV problem in which jobs are considered to have the same weight (Merten & Muller, 1972; Ye, Li, Xu, & Farley, 2006).

WWTV minimization has many applications in different fields. WWTV problems are closely related to weighted common due date problems. A weighted common due date problem can be modeled as finding an optimal common due date, d, to minimize the mean squared deviation (MSD), $\sum v_i(W_i - d)^2$. It can be obtained as $(\sum v_i^* W_i^*)/\sum v_i^*$, where v_i^* and W_i^* are the weight and waiting time of the jobs in the optimal WWTV sequence, respectively, since MSD is the second moment about d. The connection between the WWTV minimization problems and the weighted common due date problems suggests the applicability of putting WWTV minimization for scheduling problems with earliness and tardiness (E/T) penalties in manufacturing systems (Baker & Scudder, 1990; Cheng & Gupta, 1989; Verma & Dessouky, 1998). This is an extension of the relationship between the WTV problem and common due date problem or mean squared deviation (MSD) problem (Bagchi, Sullivan, & Chang, 1987; Cheng & Gupta, 1989; Cheng & Kovalyov, 1996; Cheng, Chen, & Shakhlevich, 2002; Panwalker, Smith, & Seidmann, 1982). It is shown that the CTV problem is equivalent to the unconstrained version of the mean squared deviation minimization problem (Bagchi et al., 1987).

In addition to applications to job scheduling for E/T minimization in manufacturing systems, WWTV is in connection with providing stabilized Quality of Service (QoS) in computers and networks (Ye et al., 2006). For instance, the data packets from WTV-sensitive applications (audio or video players) should have higher priorities of data transmission against other data packets (emailing or web browsing) arriving at a router. It is desirable to minimize the weighted WTV of the data packets to stabilize the router so that higher priority data packets get more consistent service performance or stable QoS. WWTV minimization may also find application in other service facilities where it is desirable to serve the jobs with different weights in stability.

The authors (Merten & Muller, 1972) first introduce the weighted variance minimization for file organization problems in which it is desirable to treat user's requests to data file in a stable manner. They propose a problem model that involves arbitrary job processing times and weights. One special case of the model, in which the jobs have equal weights, has received extensive studies since then. The authors (Merten & Muller, 1972) find that a sequence minimizing the variance of flow-time or completion time is antithetical to a sequence minimizing the variance of waiting time if the jobs have the same weight. An optimality condition of a WTV problem is shown in (Cai, 1996; Eilon & Chowdhury, 1977; Mittenthal, Raghavachari, & Rana, 1995) that the optimal sequence must be V-shaped, which means that the jobs before the job with the shortest processing time are sorted in a descending order while the jobs after the smallest job are sorted in an ascending order. The author (Schrage, 1975) considered the optimal sequences for problems with up to five jobs and showed that the longest job must be the first to be processed to minimize the completion time variance (CTV). The author (Schrage, 1975) conjectured the positions of the three longest jobs which were proven in (Hall & Kubiak, 1991). A counter example shows that the conjecture about the fourth longest job is incorrect in (Kanet, 1981). The author in (Kubiak, 1993) proves that the CTV problem on a single machine is NP-hard. The bounds for the position of the smallest job in the CTV problem are established in (Manna & Prasad, 1999). The author in (Kubiak, 1995) formulates the CTV problem as a problem of maximizing a zero-one quadratic function which is a sub-modular function with a special cost structure. The variance of job completion time with bi-criteria extension is investigated in (De, Ghosh, & Wells, 1992, 1996). A branch and bound algorithm to minimize CTV is given in (Viswanathkumar & Srinivasan, 2003) and a tabu search-based solution is developed in (Al-Turki, 2001). The pseudo polynomial algorithms and fast polynomial approximation schemes for CTV minimization problems are given in (Cheng & Kovalyov, 1996; Kubiak, Cheng, & Kovalyov, 2002; Manna & Prasad, 1997). A sufficient optimality condition for stochastic CTV is discussed in (Cai, 1996). More heuristic methods are developed for CTV/WTV problems in (Al-Turki, Fediki, & Andijani, 2001; Eilon & Chowdhury, 1977; Gowrishankar, Rajendran, & Srinivasan, 2001; Kanet, 1981; Sharma, 2002; Vani & Raghavachari, 1987; Ye et al., 2006).

We can see that there are extensive studies in the literature about WTV problems. However, studies on the WWTV problem are limited. The author in (Cai, 1995) considers the minimization of "agreeably weighted" completion time variance on a single machine. The "agreeably weighted" setting means that $p_i < p_j$ implies $v_i \geqslant v_j$, where p_i, v_i, p_j and v_j are the processing time and the weight of job i and j, respectively. This indicates that a smaller job has a greater weight value for a higher priority. The agreeably weighed condition is a strong assumption. In many cases, the weights of the jobs are independent of the processing times of the jobs.

In this paper, we will address how to reduce WWTV under different weighted scenarios including the special case studied in (Cai, 1995). The rest of the paper is organized as follows. In Section 2, we give a mathematical formulation of the WWTV problem. In Section 3, we prove the V-shape property for agreeably weighted WWTV problems and nondelay property for general WWTV problems and analyze the optimal sequences of general WWTV problems and find the strong V-Shape tendency of the optimal sequences in a general sense. Two heuristic methods are developed to construct job sequences for a WWTV problem. The testing results are given in Section 5. In Section 6, we conclude this paper.

2. Mathematical formulation

Suppose that there are n jobs in a batch ready to be released to a single machine, and the weight and processing time of each job are given. The jobs will be processed on a single machine which can process one job at a time. No preemption is allowed, and there is no setup time for each job. There are $n!$ possible permutations of job sequences. The problem is to determine the optimal job sequence, which minimizes the weighted waiting time variance.

We formulate the problem as an integer programming problem. The decision variables are binary variable, x_{ij}. x_{ij} takes the value of 1 if job j is processed at position i of a job sequence, and 0 otherwise. There are n positions in a job sequence. The job to be processed first is located at position 1, the job to be processed second at position 2, and so on. The processing time of job i is p_i and its weight is v_i. We don not use w_i notation for the weight of a job to avoid confusion with W_i which is used for the waiting time a job. The weights of the jobs are positive numbers. The waiting time of the job to process at position i is W_i.

Objective Function:

$$\text{Minimize}: \frac{1}{n-1}\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}v_j(W_i-\overline{W})^2 \tag{2}$$

Subject to:

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 1,2,\ldots,n \tag{3}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 1,2,\ldots,n \tag{4}$$

$$x_{ij} \in \{0,1\}, \quad i,j = 1,2,\ldots,n \tag{5}$$

$$W_i = W_{i-1} + \sum_{j=1}^{n}(x_{i-1,j} * p_j), \quad i = 2,3,\ldots,n \tag{6}$$

$$\overline{W} = \frac{1}{\sum_{k=1}^{n} v_k}\sum_{i=2}^{n}\sum_{j=1}^{n} x_{ij}v_jW_i \tag{7}$$

The objective is to minimize the variance of weighted waiting times of the jobs. Eq. (3) describes the constraint that each position can be taken by only one job. Eq. (4) indicates that each job can be assigned to only one position. Eq. (5) is the binary constraint. Eq. (6) defines the waiting time of the jobs at position i ($i \geqslant 2$), which is the waiting time of the job at position $i-1$ plus the processing time of the job at position $i-1$. Note that the waiting time of the first job to be processed is 0. Eq. (7) gives the weighted mean waiting times of the jobs.

3. V-shape property and tendency

A V-shaped structure in the optimal sequence of equally weighted WTV problems is discovered in (Eilon & Chowdhury, 1977). A V-shaped policy means that in terms of job processing time, p_i, the jobs before the shortest job are in descending order while the jobs after it are in ascending order. Cai (1995) shows that an optimal sequence of weighted completion time variance problems is V-shaped in terms of weighted processing time if the job-dependent weights are agreeable, in the sense that jobs with smaller processing times possess larger weights.

In the following, we will first prove that the optimal sequence for $1\|WWTV$ is also V-shaped in terms of weighted processing time p_i/v_i if the jobs are agreeably weighted. Then we will prove the nondelay property and investigate the V-shape tendency for general WWTV problems.

3.1. V-shape property for agreeably weighted WWTV problems

Theorem 1. *The optimal sequence for* $1\|WWTV$ *is V-shaped in terms of weighted processing times if* $p_i < p_j$ *implies* $v_i \geqslant v_j$, $\forall$, *i,j.*

Proof. By contradiction. Suppose a sequence λ that is not V-shaped is optimal. There must be at least three adjacent jobs, say *i*, *j*, and *k*, such that $p_j/v_j > p_k/v_k$ and $p_j/v_j > p_i/v_i$ (see Fig. 1).

Let λ' be the sequence obtained by performing a pairwise interchange on jobs *i* and *j*, and let W'_q, $q = 1, 2, \ldots, n$, be the waiting times under λ'. We have $W'_i = W_i + p_j$, $W'_j = W_j - p_i$, and $W'_q = W_q$ for $q \neq i,j$. Thus we get $\overline{W'} = (1/V)\sum_{q=1}^{n} v_q W_q = \overline{W} + \Delta$, where $\Delta = (1/V)(v_i p_j - v_j p_i)$ and $V = \sum_{q=1}^{n} v_q$. We calculate the difference of the WWTVs under the sequences λ' and λ as follows.

$$\begin{aligned}(n-1)(\mathrm{WWTV}' - \mathrm{WWTV}) &= \sum_{q=1}^{n} v_q[(W'_q - \overline{W'})^2 - (W_q - \overline{W}^2)] \\ &= \sum_{q=1}^{n} v_q[(W'_q - \overline{W'} + W_q - \overline{W})(W'_q - W_q + \overline{W} - \overline{W'})] \\ &= v_i p_j(2W_i + p_j - 2\overline{W} - \Delta) - v_j p_i(2W_j - p_i - 2\overline{W} - \Delta) \\ &= 2v_i(W_i - W_j) + v_j p_i(p_i - p_j) - \Delta(v_i p_j - v_j p_i)d \\ &\quad + (v_i p_j - v_j p_i)(2W_j + p_j - 2\overline{W}) \end{aligned} \tag{8}$$

Note that $W_i < W_j$, $p_i < p_j$, and $v_i p_j > v_j p_i$ since it is agreeably weighted. It is easy to verify that if $p_j \leqslant 2(\overline{W} - 2W_j)$ we have $\mathrm{WWTV}' - \mathrm{WWTV} \leqslant 0$.

Similarly, one may prove that if $p_j > 2(\overline{W} - 2W_j)$ a pairwise interchange of jobs *j* and *k* will produce less WWTV than sequence λ.

Fig. 1. An illustration of V-shape violation.

In summary, if $p_j \leqslant 2(\overline{W} - 2W_j)$, swapping jobs i and j will reduce WWTV(λ); if $p_j > 2(\overline{W} - 2W_j)$, swapping jobs j and k will reduce WWTV(λ). This contradicts the optimality of λ and completes the proof of the theorem. □

Corollary 1. *If $v_i = v_j$, $\forall i$ and j, any optimal sequence of $1\|WWTV$ must be V-shaped in terms of processing time p_i.*

Corollary 2. *If $p_i = p_j$, $\forall i$ and j, any optimal sequence of $1\|WWTV$ must be V-shaped in terms of weight v_i.*

Corollery 2 is the V-shape property proved in (Eilon & Chowdhury, 1977) for equally weighted WTV problem. For general $1\|$WWTV problem, meaning that the weight and the processing time of the jobs are not correlated, the V-shape property of the optimal sequence does not hold universally which can be easily shown by an counter example. For agreeably weighted WWTV problems, thanks to the V-shape property we can find out one optimal sequence in 2^{n-1} steps since job 2 must be either immediately prior to job 1 or after job 1; job 3 must be either immediately prior to jobs 1 and 2, and so on provided that $p_i/v_i < p_{i+1}/v_{i+1}$. It is computational challenging to construct an optimal sequence for a WWTV problem with the increase of job size n. Actually a WWTV problem is NP-hard since it can reduce to a WTV problem which has been already proven to be NP-hard in (Kubiak, 1993). More properties of the optimal sequence of general WWTV problems are in need. We will analyze the structure of the optimal sequence for general $1\|$WWTV problem in the next sub-section.

3.2. Nondelay property and V-shape tendency for general WWTV problems

Lemma 1. *Any idleness between jobs in a job sequence of $1\|WWTV$ deteriorates the WWTV.*

That is, an insertion of an idle time between the processing of any two jobs will lead to larger WWTV than that of the sequence without idleness between jobs. In practice, the idle time can be the release date r_j of a job or the time when the machine is kept idle.

Proof. Suppose a sequence λ as the job sequence for a WWTV problem. There is no idle time between the processing of any consecutive jobs under λ. An idle time η is inserted after job i in λ and we name the new sequence as λ', $1 \leqslant i < n$. n is the number of jobs. Let W'_q, $q = 1, 2, \ldots, n$, be the waiting times under λ'. We have $W'_q = W_q$ for $q = 1, 2, \ldots, i$ and $W'_q = W_q + \eta$ for $q = i+1, i+2, \ldots, n$. Hence, we have $\overline{W'} = (\sum_{q=1}^{n} v_q W'_q)/\sum_{q=1}^{n} v_q = \overline{W} + \varDelta, \varDelta = (\eta \sum_{q=i+1}^{n} v_q)/\sum_{q=1}^{n} v_q$. Thus, we can calculate the difference of WWTV′ and WWTV as follows:

$$\begin{aligned}(n-1)(\mathrm{WWTV}' - \mathrm{WWTV}) &= \sum_{q=1}^{n} v_q[(W'_q - \overline{W'})^2 - (W_q - \overline{W}^2)] \\ &= \sum_{q=i+1}^{n} [2v_q \varDelta(W_q - \overline{W})] + \sum_{q=i+1}^{n} v_q \varDelta(\eta - \varDelta) \qquad (9)\end{aligned}$$

We have $\eta \geqslant \varDelta$ since $\varDelta = \eta(\sum_{q=i+1}^{n} v_q)/\sum_{q=1}^{n} v_q$. So, in Eq. (9) $\sum_{q=i+1}^{n} v_q \varDelta(\eta - \varDelta) \geqslant 0$. Now we show $\sum_{q=i+1}^{n}[2v_q \varDelta(W_q - \overline{W})] \geqslant 0$. For λ, there must exist k such that $W_k < \overline{W}$ and $W_{k+1} \geqslant \overline{W}$. If $i > k$, $W_q > \overline{W}$ and apparently $\sum_{q=i+1}^{n}[2v_q \varDelta(W_q - \overline{W})] > 0$ for $q > i$. If $i \leqslant k$, $\sum_{q=i+1}^{n}[2v_q \varDelta(W_q - \overline{W})] \geqslant \sum_{q=i}^{n}[2v_q \varDelta(W_q - \overline{W})] \geqslant \cdots \geqslant \sum_{q=1}^{n}[2v_q \varDelta(W_q - \overline{W})] = 0$. This completes the proof. □

Theorem 2. *The optimal sequence for $1\|WWTV$ is a nondelay schedule.*

A schedule is called nondelay if no machine is kept idle while an operation is waiting for processing (Pinedo, 2002). Theorem 2 shows that there is no idleness between the processing of the jobs in any optimal sequence for $1\|$WWTV. The proof follows from Lemma 1 and is omitted.

The V-shape property is of great help to construct a promising sequence for agreeably weighted WWTV problem. However, obviously agreeably weighted setting is a very strong assumption which can be easily violated in the real world. Here, we will consider another case of the weight situation where $p_i < p_j$ implies

$v_i \leqslant v_j$ and a generalized weight situation in which the processing times and weights are not related. Hence, we have three scenarios:

- Positively correlated weight (*PW*): $p_i < p_j$ implies $v_i \leqslant v_j$
- Negatively correlated weight (*NW*): $p_i < p_j$ implies $v_i \geqslant v_j$
- Random weight (*RW*)

The *NW* scenario is the agreeably weighted settings in (Cai, 1995) for weighted CTV problems and previous sub section.

We define nine small problems shown in Table 1. Table 1 shows the nine instances we used to investigate the WWTV problem. These problems are originally investigated in our studies on WTV problems (Ye et al., 2006). The processing times of the jobs in problems 1 through 4 are integers. There are 5, 6, 9 and 10 jobs in problems 1 through 4, respectively. The processing times of the problems 1–4 are arbitrarily chosen. The processing times of the jobs in problem 5 are uniformly distributed between 1 and 10. The processing times of the jobs in problem 6 are exponentially distributed with $\lambda = 5$. The processing times of the jobs in problem 7 follow a normal distribution with mean 5 and standard deviation 1. Problems 8 and 9 are originally from (Eilon & Chowdhury, 1977).

We use a uniform distribution to generate random numbers between 1 and 11 as the weights of the jobs in Table 2. The range of weight between 1 and 11 is arbitrarily chosen. The assignments of the weight to the problems in the *RW* scenario are presented in the Table 2. For the *PW* and *NW* scenarios, we use the same weights for each problem but the assignment of the weights is changed accordingly. In *PW* scenarios, $p_i > p_j$ implies $v_i > v_j$ $\forall i$ and j. In *NW* scenarios, $p_i > p_j$ implies $v_i < v_j$ $\forall i$ and j. For example, the weight of 10.12 is assigned to the job with processing time of 2 in problem 1 under the *RW* scenario while it is assigned to the job with processing time of 4 under the *PW* scenario.

Table 1
Small-size problems for WWTV minimization

Job number and processing times										
P#	1	2	3	4	5	6	7	8	9	10
1	2	5	3	6	4					
2	5	2	6	7	4	3				
3	7	3	6	4	2	10	8	9	5	
4	5	3	6	2	7	10	8	4	9	11
5	4.67	8.96	9.09	1.91	8.77	4.44	1.13	6.37	2.25	9.63
6	1.12	0.09	0.68	1.84	0.06	5	0.25	3.03	0.15	0.41
7	5.24	6.2	4.77	3.72	6.73	3.91	4.7	2.82	6.1	6.28
8	9	8	25	21	100	7	13	41	5	10
9	8	13	1	5	19	10	2	18	9	16

Table 2
Weight assignment for small-size problems (RW scenario)

Job positions and weights										
P#	1	2	3	4	5	6	7	8	9	10
1	10.12	3.57	4.79	10.85	10.34					
2	7.41	4.2	2.23	9.85	1.29	5.91				
3	3.61	10.78	5.75	3.28	7.8	9.73	6.85	8.71	7.81	
4	1.15	9.9	10.04	4.85	1.6	7.18	4.83	3.11	1.42	6.38
5	3.85	4.97	1.59	1.13	6.17	4.45	6.28	10.2	2.11	9.7
6	5.34	7.25	10.72	7.13	4.3	2.96	8.37	2.31	6.87	3.49
7	1.97	8.76	8.27	9.42	8.84	6.35	3.37	3.05	3.64	7.4
8	10.04	1.55	5.11	4.56	2.69	8.27	3.54	6.06	10.38	1.1
9	3.25	3.57	10.38	5.99	2.95	5.07	6.6	7.71	9.14	4.93

To gain insights into the properties of WWTV problems, we obtain the optimal sequences and weight assignment of *NW*, *PW*, and *RW* scenarios by enumerating all the possible job sequences for each problem. The plots of the weighted processing times of the optimal sequences are shown in Figs. 2 and 3 for *PW* and *RW* scenario, respectively. Note that we only show the problems with V-shape violations. From this point on, we use the term "processing time" to denote the original processing time of a job and use "weighted processing time" of a job to represent the ratio of its processing time over its weight. In Figs. 2 and 3, the horizontal dimension represents the position of a job in an optimal job sequence and the vertical dimension represents the weighted processing time of the job. The following observations are made from the optimal job sequences.

- The largest job is not necessary placed at the last position in the WWTV problems, although in optimal sequences of WTV problems the largest job is always placed at the last position of the optimal job sequence (Schrage, 1975; Ye et al., 2006). For example, we find that for problem 5 in the *NW* scenario, the largest job is placed at the first position, not the last position.
- The job with largest weighted processing time is not always placed at the last position of the optimal job sequence as well. In the optimal sequences of problem 1 in *PW* scenario, the job with the largest weighted processing time is placed at the first position instead of the last position.
- We notice that the WTV "dual" property (Merten & Muller, 1972) does not hold when different weights are involved. The dual sequences of a WWTV problem generally yield a different WWTV value. The dual sequence of a WTV sequence is to reverse the original sequence except the last job. For instance, sequence $(1, 2, \ldots, n-1, n)$ has a dual sequence $(n-1, n-2, \ldots, 1, n)$. Dual sequences yield the same WTV value.

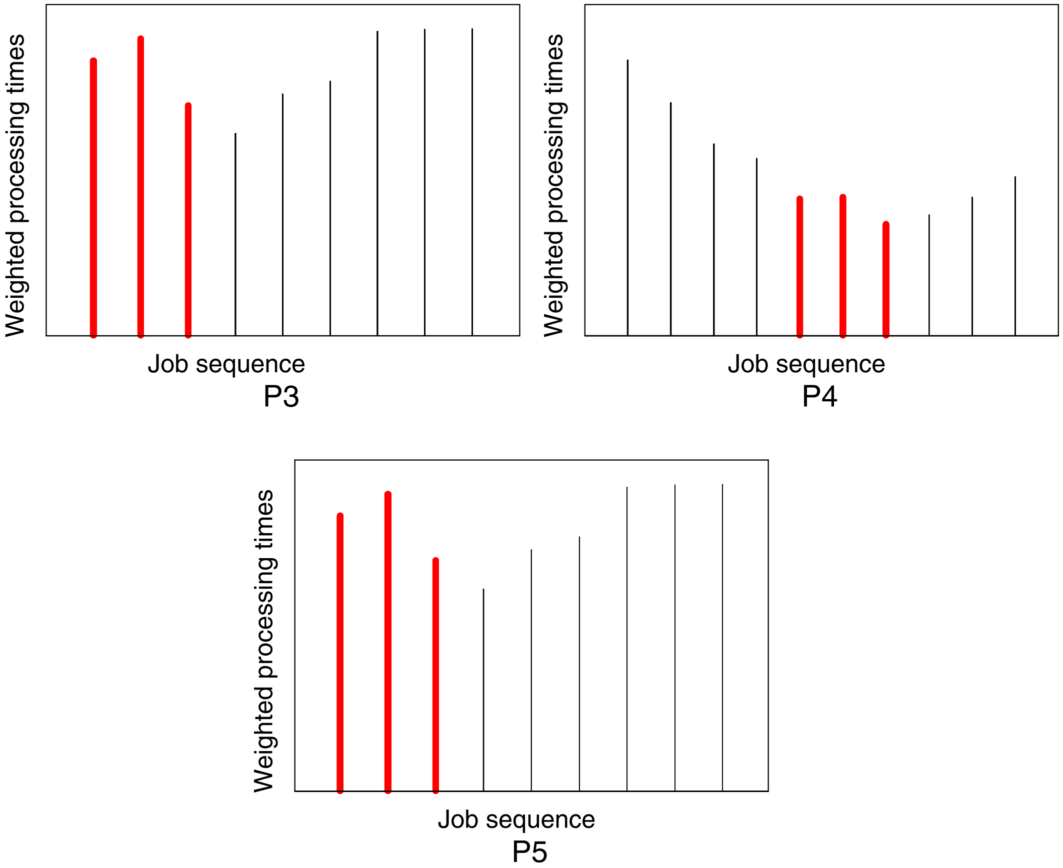


Fig. 2. Weighted processing times column plots of optimal sequences for the problems in the PW scenario. Problems with V-shape violations are depicted.

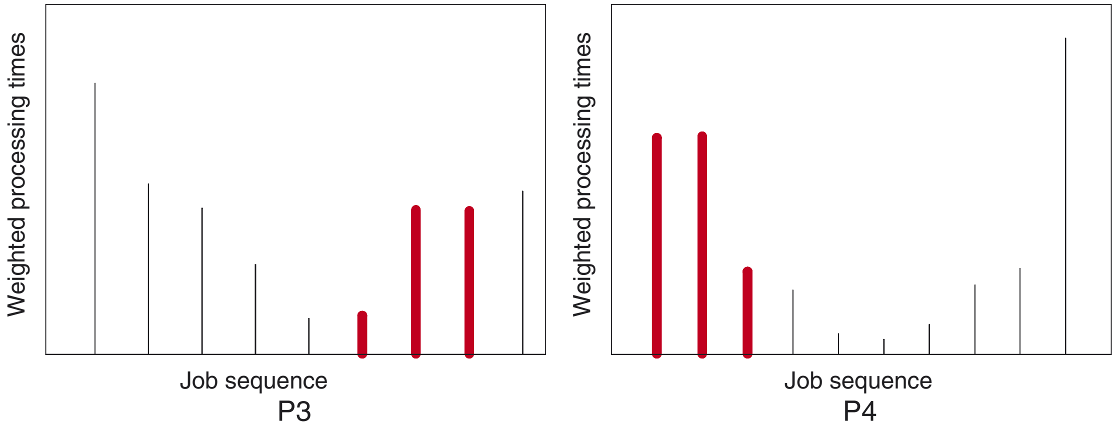


Fig. 3. Weighted processing times column plots of optimal sequences for the problems in the RW scenario. Problems with V-shape violations are presented.

For example, the dual sequences of a WTV problem (5, 4, 2, 3, 6) and (3, 2, 4, 5, 6) both yield WTV of 29.7. However, in *PW* scenario sequence (5, 4, 2, 3,6) yields WWTV of 292.76 which is different from the WWTV value of 209.74 by sequence (3, 2, 4, 5, 6).

- The V-shape property does not hold in the optimal sequences of WWTV problem in terms of weighted processing time in the *PW* and *RW* scenario. For instance, the sequences of problem 3 in the *PW* scenario and problem 4 in the *RW* scenario are not V-shaped.
- The optimal sequences in the *NW* scenario still have the V-shape property in terms of weighted processing times. The plots of the weighed processing times in the optimal sequences in Fig. 3 show strong V-shape tendencies. The optimal sequences of problem 1, 2, 5, 6, 7, 8, and 9 are V-shaped in the *RW* scenario. For problems 3, 4, and 5 in the *PW* scenario whose optimal sequences are not V-Shaped, only a few jobs violate the V-shape property while the other jobs hold the V-shaped tendencies and their neighbors have very close weighted processing times.

From the above observations we can see that most of the properties of the optimal sequences of WTV problems do not hold any more for the optimal sequences of WWTV problems. However, we still find the strong V-shape tendency of the optimal sequences in the *PW* and *RW* scenarios of the WWTV problems. These findings are also observed from more testing problems which we do not include here. Hence, based on the V-shape tendency, we develop two job scheduling methods to generate good sequences for general WWTV problems.

4. Development of heuristic scheduling methods for WWTV problems

4.1. Algorithms

The V-shape property of the agreeably weighted WWTV problems and the strong V-shape tendency of the optimal sequences of general WWTV problems gives us some insights into how to construct a promising sequence. If we take the weighted processing times of a WWTV problem as the processing times in a WTV problem, somehow we can transform a WWTV problems into a WTV problems. In (Ye et al., 2006), the authors propose highly efficient heuristic methods called the Balanced Spiral (BS) and Verified Spiral (VS) to reduce the WTV of equally weighted jobs on a single machine. Inspired by these methods, we develop two heuristic methods for WWTV problems: the Weighted Verified Spiral (WVS) algorithm and the Weighted Simplified Spiral (WSS) algorithm.

Given a set of jobs $J = \{J_1, J_2, \dots, J_n\}$ to be processed on a single resource, the processing time of job J_i is p_i and its weight is v_i correspondingly, $i = 1$ to n. Without the loss of generality, we assume that $\frac{p_i}{v_i} \leqslant \frac{p_j}{v_j}$, $\forall i < j$. The steps in the WVS algorithm is described below:

1. The initial job sequence Ω has no jobs in it, and there are n jobs in the job pool J to be scheduled.
2. Remove jobs J_1, J_{n-1}, and J_n from the job pool J and put to the job sequence, Ω, so that $\Omega = \{J_{n-1}, J_1, J_n\}$. Define two sub-sequences R and L such that $R = L = \Omega$. Job pool J becomes $\{J_2, J_3, \ldots, J_{n-2}\}$.
3. Remove the job with the largest weighted processing time from the right side of the job pool J. Try to place the job exactly before job J'_1 in sub sequence L and calculate the weighted waiting time variance WWTV_L. Try to place the job exactly after job J_1 in R and calculate the weighted waiting time variance WWTV_R. If WWTV_L is less than WWTV_R, let job sequence $\Omega = L$; otherwise $\Omega = R$. Update sub sequences L and R such that $L = R = \Omega$.
4. Repeat Step 3 until the job pool is empty and get the job sequence, Ω.

The WVS algorithm produces a job sequence with the V-shape property for a WWTV problem. Note the sorting criteria used in WVS is on weighted processing times instead of processing times in the original VS algorithm. Another difference of WVS from VS is that WVS does not start from a job sequence of $\{n-2, 1, n-1, n\}$. WVS starts from a job sequence of $\{n-1, 1, n\}$. The reason for this modification is from the observations of the optimal sequences which reveal that the second largest job in weighted processing time does not have to be the last but one. For WTV problems, there always exists an optimal sequence with $\{n-2, \ldots, n-1, n\}$ (Hall & Kubiak, 1991). However, this does not generally hold for WWTV problems.

We notice that the WVS algorithm needs to calculate WWTV of sub sequences R and L to decide the insertion position of each job in step 3, which adds the computational cost. Hence, we develop a simplified algorithm called Weighted Simplified Spiral (WSS) method which needs less computation as follows:

1. The initial job sequence, Ω, has no jobs in it, and there are n jobs in the job pool J to be scheduled. Define the empty sub-sequences L and R.
2. Remove the job with the largest weighted processing time from the right side of the job pool, J, insert it to the head of the sub-sequence, R.
3. Remove the job with the largest weighted processing time from the right side of the job pool J, append it to the tail of the sub-sequence, L.
4. Repeat step 2 and 3 till the job pool is empty. The final job sequence is the union of sub-sequences L and R as $\Omega = L + R$.

Job sequence Ω by WSS produces a job sequence of $\{J_{n-1}, J_{n-3}, \ldots, J_{n-2}, J_n\}$ with the V-shape property in a symmetric manner. We can see that WSS is more efficient than WVS in terms of the computational cost since it does not need to compute WWTV in each step. It can be easily seen that the time complexity of WSS algorithm takes $O(n \log n)$ while WVS algorithms takes $O(n^2)$.

To test the performance of the WVS and WSS algorithms, we will compare them with the First-In-First-Out (FIFO) and Weighed Shortest Processing Time first (WSPT) (Pinedo, 2002) scheduling methods in the next section. FIFO, often referred to as "a random sequence" in the single machine deterministic scheduling literature, is probably the most commonly used scheduling method in many application fields of the real world. FIFO dispatching rule is usually used in a dynamic environment. We implement FIFO in such a way that the order to process the jobs is the same as the order that the jobs are generated. That is, the jobs are processed in an "as is" sequence. The reason we choose WSPT is that WSPT gives the optimal weighted mean waiting time or WMWT and we may want to see how WSPT performs for WWTV problems. We will compare WVS and WSS with WSPT to see how much WMWT is sacrificed when we attempt to reduce WWTV.

4.2. Test problems

The nine small-size problems in Table 1 are used to test the performance of the FIFO, WSPT, WVS, and WSS for job scheduling. We also generate larger-scale problems in which the job processing times follow different probability distributions. We use the normal, exponential, uniform, and Pareto distributions since normal, exponential, and uniform distributions are commonly found in job scheduling problems (von Seggern, 1993). The authors in (Arlitt & Williamson, 1996) argue that file sizes on the web follow the Pareto distribution. We set the mean parameter of normal distribution to 576 and standard deviation parameter to 100. When generating the value of a job processing time from the normal distribution, we check whether or not a job's processing time is a positive value and replace it with a small value of 0.01 if a generated value is not positive since we require a job processing time must be greater than zero. However, while generating job processing times using the normal distribution, all values turn out to be positive. The mean of job processing times is set to 576 for an problem whose jobs have exponentially distributed processing times. We set the interval of the uniform distribution to (64, 1088). The shape parameter of the Pareto distribution is set to 1.4 and the scale parameter is set to 164.57 for problems with Pareto distributed job size. We choose 576 as the mean processing time because it is the default IP datagram size on the Internet (Postel, 1983). For each distribution, we generate 1000 problems and there are 100 jobs in each problem.

5. Testing results and discussions

For the small-size problems, we compare the job sequences produced by WVS, WSS, FIFO, and WSPT with the optimal sequences analyzed in the last section. For the larger-size problems, we compare WWTVs of the job sequences produced by the four job scheduling methods among themselves since it is highly computationally challenging to enumerate all the sequences of many jobs in a large-scale WWTV problem and obtain the optimal job sequence.

5.1. Performance results for small-size problems

For a given small-size problem, let $\mathrm{WWTV}_{\mathrm{OPT}}$ denote WWTV of the optimal sequence, and let WWTV_s be WWTV of the job sequence produced by job scheduling method s, where s represents FIFO, WSPT, WSS, or WVS. We define the percentage value of Weighted Waiting Time Variance Difference (WWTVD) of WWTV_s from $\mathrm{WWTV}_{\mathrm{OPT}}$ as follows:

$$\mathrm{WWTVD}_s = \frac{\mathrm{WWTV}_s - \mathrm{WWTV}_{\mathrm{OPT}}}{\mathrm{WWTV}_{\mathrm{OPT}}} * 100\% \tag{10}$$

Hence, WWTVD_s indicates how close WWTV_s is to $\mathrm{WWTV}_{\mathrm{OPT}}$. A smaller value of WWTVD indicates a better performance of a job scheduling method for a WWTV problem. Similarly, we define the weighted mean waiting times (WMWT) of the jobs and the percentage of Weighted Mean Waiting Time Difference (WMWTD) of WMWT_s from $\mathrm{WMWT}_{\mathrm{WSPT}}$. It can be easily verified that WSPT rule produces optimal WMWT. WMWTD indicates how much WMWT is sacrificed to achieve WWTV minimization.

$$\mathrm{WMWT} = \frac{\sum_{i=1}^{n} v_i W_i}{\sum_{i=1}^{n} v_i}$$

$$\mathrm{WMWTD}_s = \frac{\mathrm{WMWT}_s - \mathrm{WMWT}_{\mathrm{WSPT}}}{\mathrm{WMWT}_{\mathrm{WSPT}}} * 100\%$$

Table 3 shows the performance results of FIFO, WSPT, WSS, and WVS for the nine small-size problems. In the *PW* scenario, WSS and WVS outperform FIFO and WSPT for each small-size problem by producing a smaller WWTVD. For WSS, the WWTVD of the nine problems are less than 20% except problem 1 and 8 while WWTVDs of five problems are smaller than 10%. For WVS, the WWTVD for six of the nine problems are less than 5%.

In the *NW* scenario, WVS outperforms other scheduling methods in that it gives optimal WWTV for four problems and its WWTVD for eight out of the nine problems are less than 1%, followed by WSS. FIFO yields WWTV values of more than 100% for eight out of the nine problems. For problem 6, FIFO produces WWTV values that are 14 times larger than WWTV of WVS.

In the *RW* scenario, WVS produces WWTV closer to the optimal WWTV for each testing problem than the other scheduling methods, followed by WSS and WSPT. WVS gives optimal WWTV for problem one and nine. For the small-size testing problems, the WMWTDs of WVS and WSS are comparable to that of FIFO.

Table 3
Performance results of small-size problems

P#	FIFO		WSPT	WSS		WVS	
	WMWTD (%)	WWTVD (%)	WWTVD (%)	WMWTD (%)	WWTVD (%)	WMWTD (%)	WWTVD (%)
PW							
1	18.54	46.41	51.39	6.72	22.12	6.72	22.12
2	13.37	24.50	50.47	14.91	16.56	20.87	4.74
3	6.77	14.81	6.70	7.09	7.15	5.58	4.73
4	21.45	15.58	34.31	12.97	5.64	15.23	2.34
5	15.91	33.75	27.79	9.55	11.03	16.27	4.86
6	80.68	254.75	43.17	61.71	2.24	60.63	1.97
7	11.11	41.82	19.03	15.44	6.40	21.87	3.02
8	48.13	248.76	23.50	14.16	23.26	7.10	10.87
9	10.28	14.13	23.50	13.72	9.17	12.39	8.35
NW							
1	70.16	176.44	28.81	76.29	0.96	69.07	0
2	171.29	349.53	46.55	123.94	0.89	123.94	0.89
3	59.49	116.40	53.43	81.50	0.25	75.26	0.05
4	60.41	150.44	76.00	157.56	0.48	147.43	0.95
5	235.36	186.90	88.18	182.00	3.35	173.62	3.08
6	448.83	1147.85	85.98	337.67	5.60	289.51	0
7	71.16	68.05	44.22	64.12	0.07	63.63	0
8	409.49	1581.34	65.13	229.15	2.49	194.00	0.14
9	99.65	134.67	87.53	131.39	1.98	113.03	0
RW							
1	59.93	126.42	44.45	28.61	0	28.61	0
2	54.58	175.11	26.83	65.08	11.72	58.80	6.35
3	43.53	54.53	61.26	34.29	6.07	34.11	5.85
4	45.17	122.57	77.22	92.65	31.51	79.04	12.01
5	98.67	80.55	59.90	57.54	28.84	43.92	16.10
6	271.86	1024.77	95.85	266.05	4.32	230.70	2.90
7	32.95	33.14	35.26	31.29	4.42	31.29	4.42
8	230.48	859.20	96.08	61.41	7.28	81.38	1.50
9	103.57	108.46	92.27	74.39	0.08	75.49	0

We conduct computational cost analysis for the above methods. The time complexity is $O(n!)$ for exhaustive enumeration to obtain the optimal solution in the *PW* and *RW* scenarios. In the *NW* scenario, thanks to the V-shape property, we need to search only V-shaped sequences for the optimum and thus the time complexity of the algorithm can be reduced to $O(2^n)$. The time complexity of WVS and WSS algorithms are $O(n^2)$ and $O(n \log n)$, respectively. WSPT takes time $O(n \log n)$ and FIFO takes time $O(1)$. Thus, there is a trade-off between the performance and computational cost.

5.2. Performance results of larger-size problems

We compare each pair of the four scheduling methods in their performance by computing the percentage of the 1000 problems for a given distribution for which one method in the pair produces better results (smaller WWTV) than another. For example, in Table 4, the last column of the *RW* scenario for the Pareto data shows 100%, 100%, and 82.3%, which means WVS produce smaller WWTV than FIFO and WSPT for each of the 1000 problems and for 82.3% of the problems WVS produces smaller WWTV than WSS.

From Table 4 we can see that WVS outperforms the other scheduling methods in the *PW* and *RW* scenarios regardless of the distribution of the job processing times. In the *NW* scenario, WSS outperforms other scheduling methods regardless of the distribution of the job processing times. WVS and WSS produce smaller WWTV than FIFO for every test problem regardless of the weight scenario and the distribution of the job processing times.

Table 4
Percentage performance comparison for larger-size problems with the normal distribution of job processing times

	Normal			Exponential		
	WSPT (%)	WSS (%)	WVS (%)	WSPT (%)	WSS (%)	WVS (%)
PW						
FIFO	100	100	100	100	100	100
WSPT		100	100		100	100
WSS			98.9			71
NW						
FIFO	100	100	100	100	100	100
WSPT		100	100		100	100
WSS			78.1			47
RW						
FIFO	100	100	100	99.80	100	100
WSPT		100	100		100	100
WSS			86.50			72.60

	Uniform			Pareto		
	WSPT (%)	WSS (%)	WVS (%)	WSPT (%)	WSS (%)	WVS (%)
PW						
FIFO	37.6	100	100	100	100	100
WSPT		100	100		100	100
WSS			53.1			84.7
NW						
FIFO	100	100	100	100	100	100
WSPT		100	100		100	100
WSS			50.4			38.2
RW						
FIFO	100	100	100	100	100	100
WSPT		100	100		99.9	100
WSS			73.0			82.30

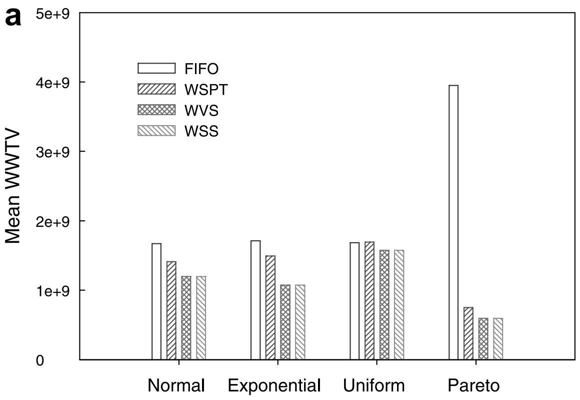
We also compare the mean of WWTV by FIFO, WSPT, WSS, and WVS for the problems with different distributions in different weight scenarios. The mean WWTV is calculated as the average WWTV of the 1000 problems for each weight scenario, each scheduling rule, and each processing time distribution.

Fig. 4(a) shows the comparison of mean WWTV for the larger problems in the *PW* scenario. For the problems with uniformly distributed job processing times, FIFO and WSPT produce about the same mean WWTV. Regardless of the distribution, the mean WWTV of WVS and WSS are smaller than those of FIFO and WSPT.

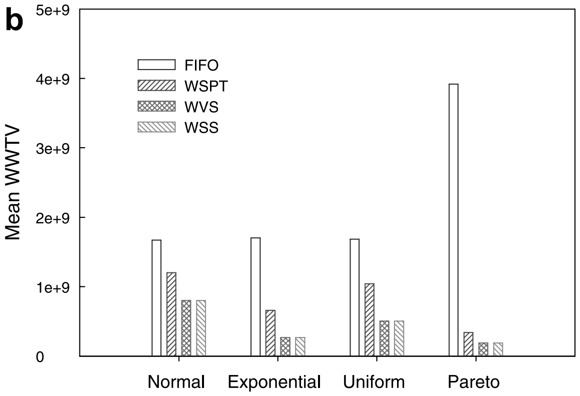
Fig. 4(b) shows the mean WWTV of the scheduling methods in the *NW* scenario. FIFO produces about the same mean WWTV for the problems with the normal, exponential, and uniform distributions. WSS and WVS produce smaller mean WWTV than WSPT and FIFO.

Fig. 4(c) shows the performance comparison results in the RW scenario. FIFO produces about the same mean WWTV in the normal, exponential, and uniform distributions. WSPT produces less mean WWTV for the problems with the Pareto distribution. The mean WWTV of WSS are about 57%, 32%, 47%, and 8% smaller than those of FIFO for the problems with the normal, exponential, uniform, and Pareto distributions, respectively.

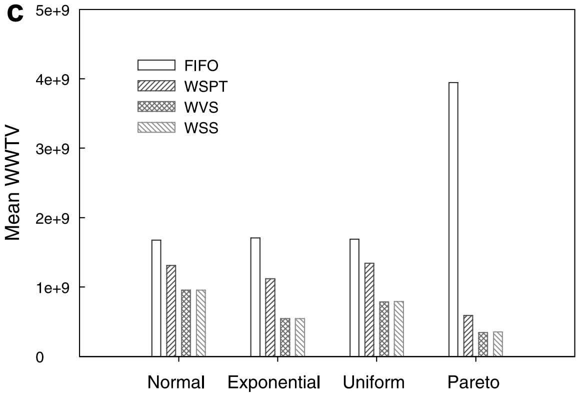
In the *PW*, *NW*, and *RW* scenarios, we can see that WVS and WSS produce smaller WWTV than those of FIFO and WSPT. The improvement made by WSS and WVS is especially greater for the Pareto distribution. In overall, FIFO produces larger WWTV than those of WVS and WSS. WSPT yields smaller WWTV than FIFO. For all the problems, WVS and WSS have similar performances. WVS produces less WWTV in the *PW* and *RW* scenarios than WSS. WSS produces smaller WWTV in the *NW* scenarios. However, the differ-



PW scenario



NW scenario



RW scenario

Fig. 4. Mean WWTV of larger-scale problems.

ence of the WWTV between WVS and WSS is small. Similar to the small-size problems, there also exists a trade-off between the performance and computational time. WVS takes more computational time with better performance than other methods. WSS takes less computational time than WVS with a comparable performance.

We also compare the standard deviation of WWTV for the larger-scale problems as shown in Table 5. WSS and WVS produce the smaller standard deviation of WWTV than FIFO and WSPT for the problems in the *PW*, *NW*, and *RW* scenarios. WSPT produces the smaller standard deviation of WWTV than FIFO except for

Table 5
Standard deviation of WWTV for the larger-scale problems

	Normal	Exponential	Uniform	Pareto
PW				
FIFO	1.17*E*+08	3.61*E*+08	1.91*E*+08	2.81*E*+10
WSPT	9.94*E*+07	3.08*E*+08	1.92*E*+08	1.97*E*+08
WVS	1.05*E*+08	2.25*E*+08	1.90*E*+08	1.23*E*+08
WSS	1.05*E*+08	2.25*E*+08	1.90*E*+08	1.24*E*+08
NW				
FIFO	1.40*E*+08	4.18*E*+08	2.34*E*+08	2.63*E*+10
WSPT	9.25*E*+07	1.52*E*+08	1.62*E*+08	5.98*E*+07
WVS	7.61*E*+07	6.49*E*+07	8.82*E*+07	2.91*E*+07
WSS	7.61*E*+07	6.49*E*+07	8.82*E*+07	2.91*E*+07
RW				
FIFO	1.26*E*+08	3.95*E*+08	2.15*E*+08	2.83*E*+10
WSPT	9.80*E*+07	2.50*E*+08	1.78*E*+08	1.44*E*+08
WVS	8.78*E*+07	1.26*E*+08	1.21*E*+08	7.37*E*+07
WSS	8.79*E*+07	1.27*E*+08	1.21*E*+08	7.61*E*+07

Table 6
Computational times of the scheduling methods (in milliseconds)

Number of jobs	FIFO	WSPT	WSS	WVS
10	0.00	0.00	0.05	0.31
50	0.00	0.00	0.05	1.35
100	0.00	0.21	0.1	3.07
500	0.00	2.5	3.39	45
1000	0.1	12.66	13.59	163.28
5000	0.42	336.41	334.06	3859.58
Computational complexity	$O(1)$	$O(n\log n)$	$O(n\log n)$	$O(n^2)$

the uniform distribution in the *PW* scenario in which they have similar standard deviation of WWTV. We notice that FIFO produces large standard deviation of WWTV for the Pareto distribution. For instance, the standard deviation of FIFO is about 371 times larger than that of WSS in RW scenario for Pareto data set. In Figs. 4(a–c) and Table 5, we observe that WSS and WVS consistently produce smaller WWTV than FIFO and WSPT.

5.3. Computational cost of job scheduling methods

Table 6 shows the computational time of running the algorithms of the scheduling methods. We implement the scheduling algorithms on a PC with a 1.9 GHZ CPU and 1 G RAM. Problems with different numbers of jobs from 10 to 5000 are measured. We run 100 problems for each level of the number of jobs and show the average computational time in Table 6. We can see that FIFO takes less computational time than the other scheduling methods, followed by WSPT and WSS. WVS takes more computational time than the others. Since the time taken to sort the jobs in their weighted processing times are included in the total computational time, the result confirms the above computational complexity analysis. For a problem with the number of jobs less than 100, WVS uses about less than 3 milliseconds to generate a good schedule. This shows that our algorithms are practically useful and can be easily implemented.

6. Conclusion

In this study we investigate the weighted waiting time variance minimization problem for a single resource on computers and networks. We formulate this problem as an integer programming problem, prove the

V-shape property for agreeably weighted WWTV problems and nondelay property for general WWTV problems, find the V-Shape tendency of the optimal sequences of general WWTV problems, and develop Weighted Verified Spiral (WVS) and Weighted Simplified Spiral (WSS) heuristic methods. We test small-size and larger-size problems in which the processing times of the jobs follow different distributions. We test three weight scenarios: the weights and processing times of the jobs are positively, negatively, and randomly correlated. For all the testing problems, WVS and WSS methods are able to produce smaller WWTV than FIFO and WSPT. WVS outperforms WSS in the PW and RW scenarios while WSS can further reduce WWTV than WVS under NW weight scenario for Pareto and exponentially distributed testing problems. WSS can be applied to job scheduling on computer and network resources due to its computational efficiency and comparable performance to that of WVS.

There are wide applications of WSS and WVS to schedule jobs to receive services from computer and network resources for providing QoS stability and performance dependability for jobs requesting services. For instance, WSS and WVS algorithms can be applied to a router to reduce the weighted waiting time variance of the data packets, a web server to schedule the web requests, and so on. Although we develop WVS and WSS in the context of Quality of Service on computers and networks, these job scheduling methods for service stability are also applicable to job scheduling problems in many other application fields that demand for service stability and performance dependability. For example, WVS and WSS can be applied to manufacturing production planning for scheduling jobs in a batch on a manufacturing machine.

Further research can be undertaken to investigate $P_m||\text{WWTV}$ and $Q_m||\text{WWTV}$ problems in which multiple machines are involved. It is of interest to find out whether the nondelay property of the optimal sequence of $1||\text{WWTV}$ still holds under multiple machine environment and to develop fast algorithms for $P_m||\text{WWTV}$ and $Q_m||\text{WWTV}$.

References

Al-Turki, U. (2001). Simple heuristics for the single machine minimum variance scheduling problem. *Arabian Journal for Science and Engineering, 26*, 127–132.

Al-Turki, U., Fediki, C., & Andijani, A. (2001). Tabu search for a class of single-machine scheduling problems. *Computers and Operations Research, 28*, 1223–1230.

Arlitt, M. F., Williamson, C. L. (1996). Web server workload characterization: the search for invariants. In: *Measurement and Modeling of Computer Systems*. ACM SIGMETRICS, pp. 126–137.

Bagchi, U., Sullivan, R. S., & Chang, Y. L. (1987). Minimizing mean squared deviation of completion times about a common due date. *Management Science, 33*(7), 894–906.

Baker, K. R., & Scudder, G. D. (1990). Sequencing with earliness and tardiness penalties: a review. *Operations Research, 38*(1), 22–36.

Cai, X. (1995). Minimization of agreeably weighted variance in single machine systems. *European Journal of Operational Research*, 576–592.

Cai, X. (1996). V-shape property for job sequences that minimize the expected completion time variance. *European Journal of Operational Research, 91*(1), 118–123.

Cheng, T., Chen, Z. L., & Shakhlevich, N. V. (2002). Common due date assignment and scheduling with ready times. *Computers and Operations Research, 29*(14), 1957–1967.

Cheng, T., & Gupta, M. (1989). Survey of scheduling research involving due-date determination decisions. *European Journal of Operational Research, 38*, 156–166.

Cheng, T., & Kovalyov, M. Y. (1996). Batch scheduling and common due-date assignment on a single machine. *Discrete Applied Mathematics, 70*, 231–245.

De, P., Ghosh, J. B., & Wells, C. E. (1992). On the minimization of completion time variance with a bicriteria extension. *Operations Research, 40*(6), 1148–1155.

De, P., Ghosh, J. B., & Wells, C. E. (1996). Scheduling to minimize the coefficient of variation. *International Journal of Production Economics, 44*, 249–253.

Eilon, S., & Chowdhury, I. G. (1977). Minimizing waiting time variance in the single machine problem. *Management Science, 23*(6), 567–575.

Gowrishankar, K., Rajendran, C., & Srinivasan, G. (2001). Flow shop scheduling algorithms for minimizing the completion time variance and the sum of squares of completion time deviations from a common due date. *European Journal of Operational Research, 132*, 643–665.

Hall, N. G., & Kubiak, W. (1991). Proof of a conjecture of schrage about the completion time variance problem. *Operations Research Letters, 10*, 467–472.

Kanet, J. J. (1981). Minimizing variation of flow time in single machine systems. *Management Science, 27*, 1453–1459.

Kubiak, W. (1993). Completion time variance on a single machine is difficult. *Operations Research Letter, 12*, 49–59.

Kubiak, W. (1995). New result on the completion time variance minimization. *Discrete Applied Mathematics, 58*, 157–168.

Kubiak, W., Cheng, J., & Kovalyov, M. Y. (2002). Fast fully polynomial approximation schemes for minimizing completion time variance. *European Journal of Operational Research, 137*, 303–309.

Manna, D. K., & Prasad, V. R. (1997). Pseudopolynomial algorithms for ctv minimization in single machine scheduling. *Computers and Operations Research, 24*(12), 1119–1128.

Manna, D. K., & Prasad, V. R. (1999). Bounds for the position of the smallest job in completion time variance minimization. *European Journal of Operational Research, 114*(2), 411–419.

Merten, A., & Muller, M. (1972). Variance minimization in single machine sequencing problems. *Management Science, 18*(9), 518–528.

Mittenthal, J., Raghavachari, M., & Rana, A. (1995). V- and a-shaped properties for optimal single machine schedules for a class of non-separable penalty functions. *European Journal of Operational Research, 86*(2), 262–269.

Panwalker, S. S., Smith, M. L., & Seidmann, A. (1982). Common due date assignment to minimize total penalty for the one machine scheduling problem. *Operations Research, 30*(2), 391–399.

Pinedo, M. (2002). *Scheduling Theory, Algorithms, and Systems*. Prentice-Hall.

Postel, J. (1983). The tcp maximum segment size and related topics. RFC 879. <http://rfc.net/rfc879.html/>.

Schrage, L. (1975). Minimizing the time-in-system variance for a finite jobset. *Management Science, 21*(5), 540–543.

Sharma, P. (2002). Permutation polyhedra and minimisation of the mari-ance of completion times on a single machine. *Journal of Heuristics, 8*(4), 467–485.

Vani, V., & Raghavachari, M. (1987). Deterministic and random single machine sequencing with variance minimization. *Operations Reserach, 35*(1), 111–120.

Verma, S., & Dessouky, M. (1998). Single-machine scheduling of unit-time jobs with earliness and tardiness penalties. *Mathematics of Operations Research, 23*, 930–943.

Viswanathkumar, C., & Srinivasan, C. (2003). A branch and bound algorithm to minimize completion time variance on a single processor. *Computers and Operations Research, 30*(8), 1135–1150.

von Seggern, D. (1993). *CRC Standard Curves and Surfaces*. CRC Press, pp. 252–253.

Ye, N., Li, X., Farley, T., Xu, X. (2006). Job scheduling methods for reducing waiting time variance. *Computers and Operations Research* (in press).