International Journal of Operations Research

C International Journal of Operations Research Vol. 4, No. 1, 11−17 (2007)

Minimizing Class-based Completion Time Variance on a Single Machine

Xueping Li[1,*], Yuerong Chen[1], Yang Sun[2], and Rapinder Sawhney[1]

[1]Department of Industrial and Information Engineering, University of Tennessee-Knoxville, 408 East Stadium Hall, Knoxville, TN 37996-0700, USA

[2]Department of Industrial Engineering, Arizona State University, Tempe, AZ 85287, USA

Received May 2006; *Revised July 2006*; *Accepted September 2006*

Abstract—We consider the problem of scheduling a set of simultaneously available jobs on a single machine. The objective is to determine a schedule that minimizes the class-based completion time variance (CB-CTV) of the jobs while reducing the overall CTV is taken as the secondary objective. This non-regular performance measure is closely related to service stability and of practical significance in many areas. We prove that a CB-CTV problem can be transformed into a series of CTV minimization problems, which allows us to apply the existing well developed properties and scheduling methods of CTV. Computational results are presented to show the trade-off between the overall CTV and CB-CTV and indicate that it is desirable to minimize CB-CTV with regard to service stability and consistency from customers' point of view.

Keywords—Completion time variance (CTV), Class-based completion time variance (CB-CTV), Job scheduling, Optimization

1. INTRODUCTION

In the scheduling field, there are two types of performance measures: regular and non-regular measures (Pinedo, 2002). Regular measures are defined as objective functions that are nondecreasing in job completion times. Numerous earlier works have focused on regular performance measures, such as makespan, mean lateness, and total weighted completion time. However, the emphasis has changed to non-regular performance measures with the increasing interest in just-in-time (JIT) product, which espouses the notion that both earliness and tardiness should be penalized (Baker and Scudder, 1990). Examples of non-regular performance measures include mean squared deviation (MSD) of completion times, completion time variance (CTV), and waiting time variance (WTV). This study addresses the class-based completion time variance (CB-CTV) minimization problem on a single machine, denoted by $1||$CB-CTV, using the well-known $\alpha|\beta|\gamma$ notation introduced by Graham et al. (1979).

CTV minimization problems have been extensively investigated since the early 1970s. Minimizing CTV of jobs means giving jobs a uniform treatment. That is, each job is kept in system for nearly the same time (Merten and Muller, 1972). The CTV minimization problem was first proposed for the computer file organization problem and it applies to many other practical problems that involve offering uniform services. Many favorable properties of CTV minimization problems have been discovered in the literature. For example, Merten and Muller (1972) showed that the optimal scheduling sequence that minimizes CTV is antithetical to the optimal scheduling sequence that minimizes WTV and the minimum values of these two variance measures are equal. Moveover, they showed that leaving the first job unmoved and reversing the order of the last $n-1$ jobs will not change CTV. It follows that there exist at least two optimal sequences, i.e., if $(n, n-1, ..., 2, 1)$ is an optimal sequence, then $(n, 1, 2, ..., n-1)$ is also optimal. Schrage (1975) conjectured that for CTV minimization problems, the largest job should be scheduled first, the second largest one should be scheduled last, and the third and fourth largest ones should be scheduled in the second and third positions respectively. Eilon and Chowdhury (1977) presented that the optimal scheduling sequence is of V-shape. In other words, jobs before the smallest job are scheduled in the decreasing order of processing times and jobs after the smallest job are scheduled in the increasing order of processing times. Kanet (1981) gave a counterexample of Schrage's conjecture about the scheduling position of the fourth largest job. On the other hand, Hall and Kubiak (1991) verified Schrage's conjecture of the placement of the first three largest jobs. Manna and Prasad (1999) further contributed to this field and proposed the bounds for the position of the smallest job in an optimal sequence. In spite of so many dominant properties, it is unlikely that there exists a polynomial time algorithm for the derivation of an optimal sequence. Kubiak (1993) proved that the CTV minimization problem is NP-hard. Many heuristic

* Corresponding author's email: Xueping.Li@utk.edu

methods (Eilon and Chowdhury, 1977; Kanet, 1981; Vani and Raghavachari, 1987; Manna and Prasad, 1997, 1999) have been developed to obtain near optimal solutions. Meanwhile, some algorithms have also been proposed to minimize CTV, including a dynamic programming algorithm (De et al., 1992), a genetic algorithm (Gupta et al., 1993), a simulated annealing method (Mittenthal et al., 1993), a tabu search method (Al-Turki et al., 2001), a branch and bound method (Viswanathkumar and Srinivasan, 2003), and an ant-colony optimization algorithm (Gajpal and Rajendran, 2006).

However, previous research on CTV minimization deals with problems mainly from the viewpoint of the system. In this point of view, jobs are assumed to be independent of each other, which is often not practical in the real world. In general, some jobs are related to some other jobs. For instance, jobs requested by the same user, such as multiple requests to a web server from the same client, often need to be considered together as a class. Thus, the system CTV performance measure may result in the dissatisfaction of a certain user with the service, because CTV of the jobs belonging to this user may be very large even though the overall CTV of all jobs in the system has reached the minimum. Consequently, the dissatisfaction with the service may cause the user to leave a system and turn to its rival. Such a result is undesirable to service providers. It is necessary, therefore, to investigate CTV minimization problems from the viewpoint of users. The CB-CTV minimization problem arises accordingly. CB-CTV is closely related to service stability since it penalizes both earliness and tardiness, and it is further related to customer satisfaction because it takes into account customer preferences. CB-CTV minimization has wide applications in many areas such as packet scheduling for Internet communications and reservation systems, modern manufacturing systems, supply chain management, and others where it is desirable to achieve service stability while considering customer preference. Since CTV is important from the perspective of the system, reducing the overall CTV is taken as the secondary objective in this paper.

In section 2 of this paper, the formulation of the problem is presented and a small example is illustrated. In section 3, we present several dominant properties for CTV problems and prove that CB-CTV problems can be transformed into a series of CTV problems on a single machine. In section 4, computational results are presented for both small and large problem instances and a trade-off relationship between CB-CTV and the overall CTV is revealed. In section 5, we summarize our results.

2. PROBLEM FORMULATION

In this paper, we consider the problem of scheduling L-class jobs on a single machine. All jobs are released at time zero and their processing times are known deterministically. Preemption is not allowed, i.e., jobs cannot be interrupted during their processing. Also, we assume that there is no setup time between two consecutive jobs. These assumptions are the same as those adopted by Merten and Muller (1972), Eilon and Chowdhury (1977) and Kanet (1981). Our objective is to find an optimal scheduling sequence that minimizes

$$CB\text{-}CTV = \sum_{i=1}^{L} \frac{n_i}{n} CTV_i \tag{1}$$

where L is the number of classes, n is the total number of all jobs, n_i is the number of jobs in the i^{th} class, and CTV_i is the CTV of jobs in the i^{th} class. CTV_i is computed as follows:

$$CTV_i = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} (C_{ij} - \overline{C}_i)^2 \tag{2}$$

where C_{ij} is the completion time of the j^{th} job in the i^{th} class and $\overline{C}_i$ is the mean completion time of the jobs in the i^{th} class.

For illustration, we give an example as follows. Suppose that there are three classes of jobs required to be scheduled on a single machine. These jobs are as follows:

Class I: 20, 5; Class II: 14, 2, 12; Class III: 8, 4, 1, 16

Here and throughout the paper, we denote jobs by their processing times. Assume they are scheduled in the following way: 12, 1, 4, 20, 14, 8, 5, 16, 2. Then the completion times of the jobs in these three classes are 37, 64; 51, 82, 12; and 59, 17, 13, 80 respectively. Hence, the CTVs of the three classes are 364.5, 1230.3, and 1066.3, respectively. Thus, the CB-CTV of this scheduling sequence is $\frac{2}{9} \times 364.5 + \frac{3}{9} \times 1230.3 + \frac{4}{9} \times 1066.3 = 965$.

Using exhaustive enumeration, we can obtain an optimal sequence of the above example that minimizes CB-CTV and takes reducing the overall CTV as the secondary objective. This optimal schedule is 20, 5, 16, 4, 1, 8, 14, 2, 12. The obtained minimum CB-CTV is 35.07 and the respective CTVs of the three classes are 12.5, 57.33, and 29.67. On the other hand, the overall CTV of this optimal sequence is 426.36. If we schedule these jobs without the consideration of the classes, we can obtain an optimal sequence which has the minimum overall CTV through enumeration. This sequence is 20, 16, 8, 5, 2, 1, 4, 12, 14 and the corresponding overall CTV is 314.36, while the

Table 1. An example of a small problem instance for *CB-CTV* and overall *CTV* minimization

	optimal sequence	CTV_1	CTV_2	CTV_3	*CB-CTV*	*CTV*
Class-based	20, 5, 16, 4, 1, 8, 14, 2, 12	12.5*	57.33*	29.67*	35.07*	426.36
Non-class-based	20, 16, 8, 5, 2, 1, 4, 12, 14	420.5	241	78.67	208.74	314.36*

CB-CTV of this sequence is 208.74 with the respective inner-class CTVs of 420.5, 241, and 78.67. We summarize these results into Table 1, where * denotes the optimal value.

From Table 1, we observe that when the overall minimum CTV is desired, CB-CTV is not minimized. The corresponding CB-CTV (208.74) has a large deviation from the possible minimum CB-CTV (35.07). The jobs from the same class receive the greatly different treatment, which are represented by their large inner-class CTVs (420.5, 241, 78.67) compared with (12.5, 57.33, 29.67). This implies that the jobs of the same class under CB-CTV minimization gain stabler services than under the overall CTV minimization without the consideration of classes. This inner-class CTV reduction leads to user satisfaction in the viewpoint of users with regard to service stability. It is the difference between the overall CTV and CB-CTV that motivates our research on the CB-CTV minimization problem.

3. DOMINANT PROPERTIES FOR CTV AND CB-CTV PROBLEMS

The CTV problem has been discovered to have a number of dominant properties. The following properties are summarized from the literature.

Property 1. For any scheduling sequence *R*, *CTV* of *R* is equal to *WTV* of *R*′, where *R*′ is the antithetical schedule of *R* (Theorem H in Merten and Muller (1972)).

Property 2. The scheduling sequence that minimizes *WTV* is antithetical to the scheduling sequence that minimizes *CTV* (Corollary H.1 in Merten and Muller (1972)).

Property 3. *CTV* remains unchanged when reversing the order of the last $n-1$ jobs (Theorem K in Merten and Muller (1972)).

Property 4. For *CTV* minimization problems, an optimal scheduling sequence is of the form of (n, $n-2$, ..., $n-1$). That is, the largest job is arranged at the first position, the second longest job is arranged at the last position, and the third longest job is arranged at the second position (Theorem 1 in Hall and Kubiak (1991)).

Property 5. The optimal sequence for a WTV minimization problem is V-shaped (Theorem B in Eilon and Chowdhury (1977)).

Property 6. The optimal sequence for a CTV minimization problem is V-shaped. (the combination of Property 2 and Property 5)

In view of these properties, it will be very desirable if a CB-CTV minimization problem can be transformed into CTV minimization problems. If so, we can apply these properties to solve the CB-CTV minimization problem. We will prove this property later.

We use the following notation to represent job scheduling sequences:

p_{ij} = the processing time of the j^{th} processed job in the i^{th} class;
X_i = the i^{th} job block that separates the jobs in a certain class.

To illustrate our notation, consider only the q^{th} class. A possible scheduling sequence may be of the following form

$p_{21}, p_{31}, p_{q1}, p_{q2}, p_{22}, p_{11}, p_{32}, p_{q3}, p_{L1}, p_{23}, p_{q4}, p_{q5}, p_{q6}, ..., p_{Ln_L}, p_{qn_q}, p_{36}, p_{2n2}, ...$

Then we can denote this scheduling sequence by the following:

$$\boxed{X_0}, p_{q1}, p_{q2}, \boxed{X_1}, p_{q3}, \boxed{X_2}, p_{q4}, p_{q5}, p_{q6}, \boxed{X_3}, ..., \boxed{X_{s-1}}, p_{qn_q}, \boxed{X_s} \quad (3)$$

where s is an appropriate integer.

Lemma 1. CTV_q is smaller in the following schedule than in Schedule (3):

$$\boxed{X_0}, p_{q1}, p_{q2}, ..., p_{qn_q}, \boxed{X_1}, \boxed{X_2}, \boxed{X_3}, ..., \boxed{X_{s-1}}, \boxed{X_s} \quad (4)$$

Proof. First, we prove a special case: there is only one block among the q^{th} class in Schedule (3). That is, CTV_q is smaller in the schedule

$$\boxed{X_0}, p_{q1}, p_{q2}, ..., p_{q(m-1)}, p_{qm}, ..., p_{qn_q}, \boxed{X_1}, \boxed{X_2} \quad (5)$$

than in the schedule

$$\boxed{X_0}, p_{q1}, p_{q2}, ..., p_{q(m-1)}, \boxed{X_1}, p_{qm}, ..., p_{qn_q}, \boxed{X_2} \quad (6)$$

where $2 \le m \le n_q$.

The following notation is used in the proof of this special case:

- k : the sum of processing times of jobs in the block X_1;
- V : CTV_q by Schedule (5);
- V' : CTV_q by Schedule (6);
- C_i : completion time of job p_{qi} in Schedule (5), $i = 1, 2, ..., n_q$;
- C_i' : completion time of job p_{qi} in Schedule (6), $i = 1, 2, ..., n_q$.

It is easy to show that

$C_i' = C_i$, $i = 1, 2, ..., m-1$
$C_i' = C_i + k$, $i = m, m+1, ..., n_q$

According to (Kanet, 1981), *CTV* has an alternative form: $V = \sum_{i=1}^{n_q} \sum_{j=i+1}^{n_q} (C_j - C_i)^2$. Hence,

$$
\begin{aligned}
V' - V &= \sum_{i=1}^{n_q}\sum_{j=i+1}^{n_q}(C_j' - C_i')^2 - \sum_{i=1}^{n_q}\sum_{j=i+1}^{n_q}(C_j - C_i)^2 \\
&= \sum_{i=1}^{m-1}\sum_{j=m}^{n_q}(C_j + k - C_i)^2 - \sum_{i=1}^{m-1}\sum_{j=m}^{n_q}(C_j - C_i)^2 \\
&= \sum_{i=1}^{m-1}\sum_{j=m}^{n_q}[(C_j + k - C_i)^2 - (C_j - C_i)^2] \\
&= \sum_{i=1}^{m-1}\sum_{j=m}^{n_q}[k(2C_j - 2C_i + k)] \\
&> 0 \text{ (since } k > 0 \text{ and } C_j > C_i)
\end{aligned}
$$

So, CTV_q is smaller in Schedule (5) than in Schedule (6).

Next we prove the lemma. First we move the block X_{s-1} in Schedule (3) to the exact back of p_{qnq}. According to the above special case, the new schedule has a smaller CTV_q. Again, by moving the block X_{s-2} to the exact back of p_{qnq}, we obtain a schedule in which CTV_q is smaller than in the last schedule. Keep moving blocks in this fashion until the block X_1 is moved and Schedule (4) is obtained. Since the schedule after each move produces a smaller CTV_q than in the former schedule, Schedule (4) has a smaller CTV_q than Schedule (3).

Lemma 2. CTV_q keeps a constant, as long as the scheduling form satisfies: *i*) No jobs from other classes are scheduled among the q^{th} class, i.e., no blocks exist among the q^{th} class; and *ii*) The inner-class scheduling order of the q^{th} class keeps unchanged.

Proof. Let S_1 and S_2 be any two schedules that satisfy the above two conditions. Denote job completion times of the q^{th} class in S_1 by $\{C_1, C_2, ..., C_{nq}\}$. Then job completion times of the q^{th} class in S_2 will be $\{C_1 + h, C_2 + h, ..., C_{nq} + h\}$, where h is an appropriate real number. Let $\overline{C}'$, CTV' and $\overline{C}$, CTV be mean completion times and CTVs of the q^{th} class in S_1 and S_2 respectively. Then

$$
\overline{C}' = \frac{1}{n_q}\sum_{i=1}^{n_q}(C_i + h) = \frac{1}{n_q}\sum_{i=1}^{n_q}C_i + h = \overline{C} + h
$$

$$
CTV' = \frac{1}{n_q - 1}\sum_{i=1}^{n_q}(C_i + h - \overline{C}')^2 = \frac{1}{n_q - 1}\sum_{i=1}^{n_q}(C_i - \overline{C})^2 = CTV
$$

This completes the proof.

Theorem 1. Regardless of the intra-class scheduling order, the scheduling form

$$
\boxed{p_{11}, ..., p_{1n1}},\ \boxed{p_{21}, ..., p_{2n2}},\ ...,\ \boxed{p_{L1}, ..., p_{LnL}} \qquad (7)
$$

has a smaller CB-CTV than any scheduling form that has the same inner-class scheduling order as Schedule (7) and in which there exists at least one class whose jobs are not scheduled consecutively.

Proof. Consider a schedule in which there is at least one class whose jobs are not scheduled consecutively. Gather the jobs of the same class at the position where that class first appears, for scheduling consecutively and without changing inner-class scheduling order. Then the scheduling form will become Schedule (7) or a similar schedule that only changes intra-class scheduling order, compared with Schedule (7). Lemma 2 guarantees that the change of intra-class order does not change every class's CTV. Thus, Schedule (7) and similar schedules have the same CB-CTV. On the other hand, Lemma 1 indicates that, every class's CTV in Schedule (7) or a similar schedule is smaller than in the original schedule. Hence, by definition, CB-CTV of Schedule (7) or a similar schedule is smaller.

Corollary 1: A CB-CTV minimization problem can be transformed into a series of CTV minimization problems. That is, the following equation holds:

$$
\underset{\lambda \in \Lambda}{Min}\left(\sum_{i=1}^{L}\frac{n_i}{n}CTV_i(\lambda)\right) = \sum_{i=1}^{L}\left(\frac{n_i}{n}\underset{\lambda_i \in \Lambda_i}{Min}(CTV_i(\lambda_i))\right) \qquad (8)
$$

where λ, Λ, and $CTV_i(\lambda)$ (i = 1, ..., L) are respectively a schedule of all jobs of L classes, the schedule set composed of all possible λ, and the CTV of the i^{th} class under the schedule λ, while λ_i (i = 1, ..., L), Λ_i (i = 1, ..., L), and $CTV_i(\lambda_i)$ (i = 1, ..., L) are respectively a schedule of all jobs of the i^{th} class, the schedule set composed of all possible λ_i, and the CTV of the i^{th} class under the schedule λ_i.

Proof. Since Schedule (7) or a similar schedule that only changes intra-class scheduling order has a smaller CB-CTV, as long as every class's jobs are further scheduled in the way such that the class's CTV is at a minimum, a minimal CB-CTV is achieved.

According to Corollary 1, to obtain the optimal scheduling sequence with the minimum CB-CTV, we only need to schedule jobs by their classes and schedule jobs of each class in the way that their innerclass CTVs are the minimum. This transformation dramatically simplifies the problem since there have been a lot of heuristics that can be used for CTV minimization problems, such as those in Eilon and Chowdhury (1977), Kanet (1981), Vani and Raghavachari (1987), Manna and Prasad (1997, 1999).

4. COMPUTATIONAL RESULTS

We have bridged CB-CTV minimization problems with CTV minimization problems. Nevertheless, it is necessary to investigate the relationship between the overall CTV and CB-CTV. In this section, we compute scheduling sequences for the overall CTV and CB-CTV minimization respectively for the same small or large program instances. Assume that the processing times of jobs follow a uniform distribution, taking values from the integers between 1 and 20 for small problem instances and the integers between 1 to 150 for large problem instances.

4.1 Small problem instances

For small instances, we consider two instances from n = 9 jobs, L = 3 classes and two instances from n = 10 jobs, L = 5 classes respectively. In Table 2 there are four small problem instances and job classes are distinguished by semicolons.

Table 2. Four small problem instances

Instances	Job processing times
1	3, 13; 6, 5, 15; 11, 19, 7, 12
2	18, 15; 8, 4, 20; 16, 13, 1, 6
3	6, 10; 2, 20; 12, 9; 11, 7; 5, 16
4	20, 5; 13, 10; 18, 16; 1, 17; 9, 19

Similar to the example in section 2, the overall CTV and CB-CTV of these problem instances cannot be minimized at the same time. Because of their small size, we can use exhaustive enumeration to obtain optimal sequences with CTV minimization. The computation of the minimum CB-CTV is based on Corollary 1 and realized by using enumeration to obtain the minimum CTV of each class and combining them. Since there is a total of 2^L optimal sequences for CB-CTV minimization, we choose the one with the smallest CTV. The computational output is shown in Table 3, where * denotes the optimal value, "N/A" stands for "not applicable", CB means that the optimal sequences are obtained under CB-CTV minimization, and NCB means that the optimal sequences are obtained under the overall CTV minimization (i.e., non-class-based CTV minimization).

From Table 3, we can observe that there is a trade-off between CB-CTV and the overall CTV. That is, if the overall CTV is minimized, inner-class CTVs may be large. Namely, CB-CTV is large. Conversely, if CB-CTV is minimized, inner-class CTVs will be optimal, while the overall CTV deviates from its optimum. In other words, the improvement of CB-CTV performance is obtained at the cost of sacrificing the overall CTV performance. Let CTV_{Im}^{i} (i = 1, 2, ..., 5), $CB\text{-}CTV_{Im}$, and CTV_S denote the performance improvement of inner-class CTVs, CB-CTV and the performance sacrifice of the overall CTV respectively. Then they can be measured by respective decrease or increase percents as follows:

$$CTV_{Im}^{i} = \frac{CTV_{NCB}^{i} - CTV_{CB}^{i}}{CTV_{NCB}^{i}} \times 100\%,\ i = 1, 2, ..., 5 \tag{9}$$

$$CB\text{-}CTV_{Im} = \frac{CB\text{-}CTV_{NCB} - CB\text{-}CTV_{CB}}{CB\text{-}CTV_{NCB}} \times 100\% \tag{10}$$

$$CTV_S = \frac{CTV_{CB} - CTV_{NCB}}{CTV_{NCB}} \times 100\% \tag{11}$$

where CTV_{CB}^{i} (i = 1, 2, ..., 5), $CB\text{-}CTV_{CB}$, CTV_{CB}, CTV_{NCB}^{i} (i = 1, 2, ..., 5), $CB\text{-}CTV_{NCB}$, CTV_{NCB} denote inner-class CTVs, CB-CTV and the overall CTV of optimal sequences under the class-based and the non-class-based situations respectively. Through the calculation with regard to the above four small problem instances, the values of these performance indices are listed in Table 4.

According to Table 4, the overall CTV performance sacrifices when the objective is to minimize CB-CTV. However, the CTV performance of an individual class is improved dramatically, which is more significant from a user's perspective. For example, in Table 3, the CTV of class 4 of instance 4 under CB-CTV minimization is equal to 1/400 of that under the overall CTV minimization. Since users are independent of each other, they receive a better service under CB-CTV minimization than under the overall CTV minimization. Also, the rate of the overall CTV performance sacrifice is much smaller than that of CB-CTV performance improvement. It indicates that the objective defined by us is desirable.

Table 3. CB-CTV vs. CTV for four small problem instances. It shows consistently smaller CTV for individual class under CB-CTV minimization than under the overall CTV minimization

No.	Optimal sequences	CTV_1	CTV_2	CTV_3	CTV_4	CTV_5	CB-CTV	CTV
1	CB: 19, 12, 7, 11, 13, 3, 15, 6, 5	4.5*	30.33*	158.25*	N/A	N/A	81.44*	644.11
	NCB: 19, 15, 11, 7, 3, 5, 6, 12, 13	648	289.33	588.33	N/A	N/A	501.93	476.78*
2	CB: 20, 4, 8, 16, 13, 1, 6, 18, 15	112.5*	37.33*	70.92*	N/A	N/A	68.96*	761.19
	NCB: 20, 16, 15, 6, 4, 1, 8, 13, 18	1250	710.33	372.33	N/A	N/A	680.04	572.61*
3	CB: 20, 2, 16, 5, 10, 6, 11, 7, 12, 9	18*	2*	40.5*	24.5*	12.5*	19.5*	716.1
	NCB: 20, 12, 11, 9, 5, 2, 6, 7, 10, 16	144.5	760.5	200	420.5	840.5	473.2	533.78*
4	CB: 19, 9, 20, 5, 17, 1, 13, 10, 18, 16	12.5*	50*	128*	0.5*	40.5*	46.3*	1226.01
	NCB: 20, 18, 17, 10, 9, 1, 5, 13, 16, 19	1800	392	2520.5	200	1458	1274.1	1021.34*

Table 4. Performance indices comparison for four small problem instances

No.	CTV_{Im}^{1}	CTV_{Im}^{2}	CTV_{Im}^{3}	CTV_{Im}^{4}	CTV_{Im}^{5}	$CB\text{-}CTV_{Im}$	CTV_S
1	99.31%	89.52%	73.1%	N/A	N/A	83.77%	35.1%
2	91%	94.74%	80.95%	N/A	N/A	89.86%	32.93%
3	87.54%	99.74%	79.75%	94.17%	98.51%	95.88%	34.16%
4	99.31%	87.24%	94.92%	99.75%	97.22%	96.37%	20.04%

Table 5. Performance comparison of CB-CTV and overall CTV for eight large problem instances

No.	CTV_{1m}^1	CTV_{1m}^2	CTV_{1m}^3	CTV_{1m}^4	CTV_{1m}^5	$CB\text{-}CTV_{1m}$	CTV_S
1 (VS)	95.31%	92.39%	73.3%	N/A	N/A	81.7%	47.35%
2 (VS)	96.07%	89.1%	75.66%	N/A	N/A	85.05%	46.03%
3 (BS)	95.12%	90.64%	77.12%	N/A	N/A	85.2%	50.41%
4 (BS)	95.85%	90.64%	74.83%	N/A	N/A	85.61%	43.54%
5 (VS)	95.18%	96.67%	96.23%	94.09%	95.97%	95.64%	48.72%
6 (VS)	96.07%	95.71%	95.61%	95.63%	95.06%	95.69%	51.8%
7 (BS)	94.8%	96.58%	94.91%	95.38%	96.15%	95.53%	45.7%
8 (BS)	95.63%	94.72%	97.04%	95.51%	94.24%	95.51%	48.45%

4.2 Large problem instances

For large instances, we consider 4 instances from $L = 3$ classes and $n = 100$ jobs and 4 instances from $L = 5$ classes and $n = 100$ jobs. For the first 4 instances, the job number of each class is 20, 30, and 50 respectively. For the last 4 instances, the job number of each class is the same 20. Because of their large size, these instances can not be listed. For the same reason, it is extremely computationally costly if not impossible to use exhaustive enumeration to obtain optimal sequences. Hence, two recently developed algorithms, Verified Spiral (VS) and Balanced Spiral (BS), are used in this paper to approximately solve the problem (Ye et al., 2006). These two algorithms show better performance than some existing algorithms such as FIFO (first-in-first-out), SPT (shortest processing time), and EC1.2 (Method 1.2 in Eilon and Chowdhury (1977)). Note that these two algorithms are developed for WTV minimization problems, but since optimal sequences of CTV and WTV minimization problems are antithetical, they can be modified and applied to CTV minimization problems. We simply describe these two modified algorithms as follows.

Assume a single machine needs to process a job set p_1, p_2, ..., p_n, where $p_1 \leq p_2 \leq ... \leq p_n$. VS method is as follows:

1. According to Schrage's conjecture, place the job p_n in the first position, the job p_{n-1} in the second position, and the job p_{n-2} in the last position. The shortest job p_1 is placed in the position between p_{n-1} and p_{n-2}.
2. Select the longest job from the unscheduled jobs. Place it either exactly before the job p_1 or exactly after the job p_1, depending on which way produces a smaller CTV of the job sequence so far.
3. Repeat *Step* 2 until all the jobs are scheduled.

BS method is as follows:

1. Place the job p_n in the first position, the job p_{n-1} in the second position, and the job p_{n-2} in the last position. Let sequence $Lt = \{p_{n-1}\}$ and sequence $Rt = \{p_{n-2}\}$. Denote by SUM_{Lt} and SUM_{Rt} respectively the sums of the processing times of the jobs in Lt and Rt.
2. If $SUM_{Lt} < SUM_{Rt}$, append the largest job from the unscheduled jobs to sequence Lt, and update SUM_{Lt}; If $SUM_{Lt} \geq SUM_{Rt}$, prepend the largest job from the unscheduled jobs to sequence Rt, and update SUM_{Rt}.
3. Repeat *Step* 2 until all the jobs are scheduled.

The computational output for the eight instances is shown in Table 5, where * denote the optimal value and "VS" or "BS" means the algorithm used for that corresponding instance.

Table 5 demonstrates that, for large problem instances, there is still a trade-off between CB-CTV and the overall CTV minimization. In addition, although the overall CTV becomes larger when pursuing the minimum CB-CTV, each class's CTV is reduced significantly. The performance improvement of each class's CTV aligns with users' needs and such class-based service stability will lead to user satisfaction with regard to service stability and consistency. For large instances, the rate of the overall CTV performance sacrifice is also much smaller than that of CB-CTV performance improvement, which indicates the desirability of our objective. The relatively larger overall CTV performance sacrifice rates in large instances than in small ones are caused by the size of job set and the values of processing times.

5. CONCLUSION

In this paper we consider CB-CTV minimization problems on a single machine, a generalized case of CTV minimization in which jobs are assumed to come from one class. CTV is a non-regular performance measure which penalizes both earliness and tardiness. It conforms to the Just-In-Time philosophy in manufacturing systems. CB-CTV minimization further takes into account the variability reduction from the customers' point of view to achieve service stability and consistency. CTV minimization problems have been studied extensively with many dominant properties in the literature, while little study has been done to CB-CTV minimization. We prove that a CB-CTV minimization problem can be transformed into a series of CTV minimization problems. Hence, we bridge $1||CB\text{-}CTV$ and $1||CTV$ problems, which allows us to apply well developed properties and methods of CTV problems to CB-CTV problems which are NP-hard.

Computational tests are conducted for both small and large problem instances. In the small-problem scenario, the optimal sequence for the overall CTV and CB-CTV minimization is obtained by exhaustive enumeration. In the large-problem scenario, we apply two recently developed algorithms (VS and BS, which have been shown in Ye et al.

(2006)) to calculate the overall CTV and CB-CTV. Note that for the CB-CTV minimization, since there are at least 2^L optimal sequences that have the minimum CB-CTV, the one with the smallest CTV is chosen. Both scenarios show that there is a trade-off between CB-CTV and the overall CTV. However, the reduction of individual class's CTV is more significant than the sacrifice of the overall CTV. From the perspective of customers, it is more desirable to achieve CB-CTV minimization for class-based service stability and consistency.

This paper deals with CB-CTV minimization problems on a single machine. Future research can be conducted under the parallel-machine environment, denoted by $P_m||CB\text{-}CTV$. Also, the paper only considers the deterministic CB-CTV minimization problems where processing times of the jobs are given in advance. However, in the real world processing times are often unknown and random, which leads to the research of the stochastic version of CB-CTV minimization problems. Furthermore, it is of interest to investigate some variations of the CB-CTV minimization problem in which bi-criteria is considered to minimize a combination of regular and non-regular performance measures.

ACKNOWLEDGEMENTS

The authors are thankful to the two anonymous reviewers for their suggestions and insightful comments to improve the earlier versions of the paper.

REFERENCES

1. Al-Turki, U., Fediki, C., and Andijani, A. (2001). Tabu search for a class of single-machine scheduling problems. *Computers and Operations Research*, 28: 1223-1230.
2. Baker, K.R. and Scudder, G.D. (1990). Sequencing with earliness and tardiness penalties: A review. *Operations Research*, 38(1): 22-36
3. De, P., Ghosh, J.B., and Wells, C.E. (1992). On the minimization of completion time variance with a bicriteria extension. *Operations Research*, 40(6): 1148-1155.
4. Eilon, S. and Chowdhury, I.G. (1977). Minimizing waiting time variance in the single machine problem. *Management Science*, 23(6): 567-575.
5. Gajpal, Y. and Rajendran, C. (2006). An ant-colony optimization algorithm for minimizing the completion- time variance of jobs in flowshops. *International Journal of Production Economics*, 101: 259-272.
6. Graham, R.L., Lawler, E.L., Lenstra, J.K., and Rinnooy Kan, A.H.G. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 5: 287-326.
7. Gupta, M.C., Gupta, Y.P., and Kumar, A. (1993). Minimizing flow time variance in a single machine system using genetic algorithm. *European Journal of Operational Research*, 70: 289-303.
8. Hall, N.G. and Kubiak, W. (1991). Proof of a conjecture of schrage about the completion time variance problem. *Operations Research Letters*, 10: 467-472.
9. Kanet, J.J. (1981). Minimizing variation of flow time in single machine systems. *Management Science*, 27: 1453-1459.
10. Kubiak, W. (1993). Completion time variance on a single machine is difficult. *Operations Research Letters*, 12: 49–59.
11. Manna, D.K. and Prasad, V.R. (1997). Pseudopolynomial algorithms for ctv minimization in single machine scheduling. *Computers and Operations Research*, 24(12): 1119-1128.
12. Manna, D.K. and 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.
13. Merten, A. and Muller, M., May (1972). Variance minimization in single machine sequencing problems. *Management Science*, 18(9): 518-528.
14. Mittenthal, J., Raghavachari, M., and Rana, A. (1993). A hybrid simulated annealing approach for single machine scheduling problems with non-regular penalty functions. *Computers and Operations Research*, 20(2): 103-111.
15. Pinedo, M. (2002). *Scheduling Theory, Algorithms, and Systems*, 2nd ed., Prentice-Hall, Inc.
16. Schrage, L. (1975). Minimizing the time-in-system variance for a finite jobset. *Management Science*, 21(5): 540-543.
17. Vani, V. and Raghavachari, M. (1987). Deterministic and random single machine sequencing with variance minimization. *Operations Reserach*, 35(1): 111-120.
18. Viswanathkumar, G. and Srinivasan, G. (2003). A branch and bound algorithm to minimize completion time variance on a single processor. *Computers and Operations Research*, 30(8): 1135-1150.
19. Ye, N., Li, X., Farley, T., and Xu, X. (2007). Job scheduling methods for reducing waiting time variance. *Computers and Operations Research* (in press).