



Minimizing weighted mean absolute deviation of job completion times from their weighted mean

Erdal Erel^{a,*}, Jay B. Ghosh^b

^a Faculty of Business Administration, Bilkent University, Bilkent 06800, Ankara, Turkey

^b Apratech, LLC, Los Angeles, USA

ARTICLE INFO

Keywords:

Scheduling
Dynamic program
Approximation scheme

ABSTRACT

We address a single-machine scheduling problem where the objective is to minimize the weighted mean absolute deviation of job completion times from their weighted mean. This problem and its precursors aim to achieve the maximum admissible level of service equity. It has been shown earlier that the unweighted version of this problem is NP-hard in the ordinary sense. For that version, a pseudo-polynomial time dynamic program and a 2-approximate algorithm are available. However, not much (except for an important solution property) exists for the weighted version. In this paper, we establish the relationship between the optimal solution to the weighted problem and a related one in which the deviations are measured from the weighted median (rather than the mean) of the job completion times; this generalizes the 2-approximation result mentioned above. We proceed to give a pseudo-polynomial time dynamic program, establishing the ordinary NP-hardness of the problem in general. We then present a fully-polynomial time approximation scheme as well. Finally, we report the findings from a limited computational study on the heuristic solution of the general problem. Our results specialize easily to the unweighted case; they also lead to an approximation of the set of schedules that are efficient with respect to both the weighted mean absolute deviation and the weighted mean completion time.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

Consider a set N of n independent jobs that are available at time zero for non-preempted processing on a continuously available machine which can process at most one job at a time. Let job j , $j \in N$, have an integer processing time p_j ($p_j > 0$) and an integer weight w_j ($w_j \geq 0$). (For future reference, let $p_N = \sum_{j \in N} p_j$ and $w_N = \sum_{j \in N} w_j$.) Similarly, let S be an ordered set of the job indexes representing a job sequence and $[i]$ be the index of the job in position i in S . Assuming w.l.o.g. a zero start time and no inserted machine idle time, S translates to a schedule in which the completion time of the job in position i in S is given by $c_{[i]}(S) = \sum_{1 \leq k \leq i} p_{[k]}(S)$. We define $v(S, d) = (1/w_N) \sum_{1 \leq i \leq n} w_{[i]}(S) |c_{[i]}(S) - d|^r$, for d and $r \geq 0$, to be a measure of the variability of the job completion times in S about d . Here, d can either be a due-date that is common to all the jobs (independent of S) or a measure of the central tendency of the job completion times in S . Similarly, r can represent the degree to which the deviations of the job completion times from d are penalized. We define the weighted mean and the median of the job completion times in S as $c_{\text{mean}}(S) = (1/w_N) \sum_{1 \leq i \leq n} w_{[i]}(S) c_{[i]}(S)$ and $c_{\text{med}}(S) = c_{[m]}(S)$ where $m = \min\{i : 1 \leq i \leq n \text{ and } \sum_{1 \leq k \leq i} w_{[k]}(S) \geq \sum_{i < k \leq n} w_{[k]}(S)\}$, respectively. It is well known that, for any given S , $d = c_{\text{med}}(S)$ minimizes $v(S, d)$ when $r = 1$

* Corresponding author.

E-mail address: erel@bilkent.edu.tr (E. Erel).

and $d = c_{\text{mean}}(S)$ minimizes $\nu(S, d)$ when $r = 2$. The general objective is to find a schedule S_{opt} that minimizes $\nu(S, d)$ over all S , for a particular choice of d and r . In this paper, we focus only on those cases where $d = c_{\text{med}}(S)$ or $c_{\text{mean}}(S)$ and $r = 1$ or 2 .

A problem such as the above becomes relevant when one wants to achieve the maximum admissible level of service equity. Merten and Muller [14] have studied the problem in the context of organizing computer databases in such a way as to provide users as uniform an access time as possible. Specific versions of the problem also appear as sub-problems while minimizing the earliness and tardiness penalties about an unrestrictively large common due-date; see the survey by Baker and Scudder [2] for examples.

We now summarize the past research on the problem briefly. We state the most relevant developments and do not claim to be exhaustive. Chronologically, the unweighted problem when $d = c_{\text{mean}}(S)$ and $r = 2$ has been addressed first; it is commonly known as the completion time variance (CTV) problem. Eilon and Chowdhury [7] have shown that an optimal CTV schedule is V-shaped w.r.t. the processing time p_j , i.e., the jobs preceding the smallest job are scheduled in the longest processing time first (LPT) order whereas those following it are scheduled in the shortest processing time first (SPT) order. Kubiak [12] has proved that CTV is NP-hard. De et al. [6] have given an $O(n^2 p_N)$ pseudo-polynomial time dynamic program and an $O(n^3/\varepsilon)$ fully-polynomial time approximation scheme (FPTAS) for its solution; Kubiak et al. [13] have recently discovered an $O(n^2/\varepsilon)$ FPTAS. An FPTAS delivers a solution whose value is no more than $(1 + \varepsilon)$ times that of the optimal and executes in time that is bounded by a polynomial in the problem size (which in our case involves n , $(1/\varepsilon)$ and $\log(\max_j\{p_j, w_j\})$).

The weighted version of CTV, called WCTV, is clearly NP-hard at least in the ordinary sense. Cai [3] has addressed a special case of WCTV where the weights are agreeable (viz., where $w_i \geq w_j \Rightarrow p_i \leq p_j$ for any $i, j \in N$). He has established the V-shape property of an optimal schedule w.r.t. p_j and has provided an $O(np_N w_N)$ pseudo-polynomial time dynamic program for this case. Cheng and Kubiak [4] have subsequently given an $O(n^4 \log(\max_j\{p_j, w_j\})/\varepsilon)$ FPTAS for the same problem; more recently, Erel and Ghosh [8] have found an improved $O(n^2 \log(n)/\varepsilon)$ FPTAS. However, the complexity status of the general problem remains open, i.e., it is not known whether the problem is strongly NP-hard or not.

The unweighted problem when $d = c_{\text{med}}(S)$ and $r = 1$ has also been addressed early on; it is known as MAD (we will refer to it as MAD_mC here). Kanet [10] has given an $O(n \log(n))$ time solution in this case. An optimal schedule here is also V-shaped w.r.t. p_j .

The weighted version of MAD, called WET and referred to as WMAD_WmC by us here, has been studied mostly in earliness-tardiness scheduling about an unrestrictively large common due-date. Hall and Posner [9] have shown that the problem is NP-hard in the ordinary sense, having given an $O(np_N)$ pseudo-polynomial time dynamic program for its solution. De et al. [5] have independently proposed an $O(nw_N)$ pseudo-polynomial time dynamic program. Kovalyov and Kubiak [11] have given an $O(n^2 \log^3(\max_j\{p_j, w_j, 1/\varepsilon\})/\varepsilon^2)$ FPTAS for the same problem; recently, Erel and Ghosh [8] have found an improved $O(n^2 \log(\max_j\{p_j, w_j\})/\varepsilon)$ FPTAS. Note that an optimal schedule in this case is V-shaped w.r.t. the p_j/w_j ratio, i.e., the jobs preceding the job with the smallest ratio appear in the non-increasing ratio order (weighted LPT or WLPT) and those following it appear in the non-decreasing ratio order (weighted SPT or WSPT).

Recently, a new variation of the problem where $d = c_{\text{mean}}(S)$ and $r = 1$ has received attention. The rationale for studying this variation stems perhaps from the observations that the mean completion time is a more accepted measure of centrality than the median and further that there may be situations where it is more appropriate not to penalize the larger deviations from the central reference at a progressively higher rate than the smaller ones.

We will refer to the unweighted version of this latter problem as MAD_MC and the weighted version as WMAD_WMC. Aneja et al. [1] have shown that an optimal schedule for MAD_MC is V-shaped w.r.t. p_j . They have also proved that the problem is NP-hard in the ordinary sense, having given an $O(n^3 p_N^2)$ pseudo-polynomial time dynamic program for its solution. Mosheiov [15] has subsequently presented an $O(n \log(n))$ time algorithm, which though optimal for MAD_mC, is only 2-approximate for MAD_MC (i.e., the solution delivered has a value that is no more than twice that of the optimal). Through a limited computational study, Mosheiov [15] has also shown that the MAD_mC solution in fact provides a much better approximation than this.

In contrast, precious little has been done relative to WMAD_WMC. Thus far, it has been shown by Aneja et al. [1] that an optimal schedule is W-shaped w.r.t. p_j/w_j . (A schedule S is called W-shaped if there exists a job—the so-called pivot job—such that the jobs preceding/following it appear in the WLPT/WSPT order; a V-shaped schedule is W-shaped as well.) However, Aneja et al. [1] have shown, through a limited computational study, that a V-shaped schedule is often a very good surrogate.

In this paper, we address WMAD_WMC more completely. First, we provide a solution framework for the problem and present some basic results that are both useful and interesting. We show in particular that the optimal schedule for WMAD_WmC provides (though not in polynomial time) a 2-approximate solution to WMAD_WMC; this generalizes a result due to Mosheiov [15] for MAD_MC. Next, we go on to develop an $O(n^2 p_N^2 w_N^2)$ pseudo-polynomial time dynamic program for solving WMAD_WMC exactly; this establishes that the problem is NP-hard only in the ordinary sense. This dynamic program easily specializes to MAD_MC, yielding an $O(n^3 p_N^2)$ time solution as in Aneja et al. [1]. We then propose an $O(n^2 \max\{n^2/\varepsilon^2, \log(\max_j\{p_j, w_j\})\})$ FPTAS for WMAD_WMC, which upon appropriate modification, yields an $O(n^3/\varepsilon^2)$ FPTAS for MAD_MC. Finally, in line with Mosheiov [15], we report the findings of a computational study where the solution to WMAD_WmC is used as a heuristic solution to WMAD_WMC. We close by discussing how our dynamic programming solution can approximate the set of schedules that are efficient w.r.t. both the weighted mean absolute deviation and the weighted mean completion time.

2. Solution framework and basic results

First off, we assume that all the jobs are numbered such that $p_1/w_1 \leq \dots \leq p_n/w_n$ and further that N is ordered as $\{1, 2, \dots, n\}$. We then define $v_{\text{mean}}(S) = \nu(S, d)$, where $d = c_{\text{mean}}(S)$. Our objective in WMAD_WMC is to find a schedule S_{mean} which minimizes $v_{\text{mean}}(S)$ over all possible S . Since there is a W-shaped schedule (w.r.t. p_j/w_j) that is optimal in this case (see Aneja et al. [1]), we can restrict our attention to such schedules only.

A W-shaped schedule S is of the form: $S = \{L(S), \{q\}, R(S)\}$, where $q (q \in N)$ is the so-called pivot job, and $L(S)$ and $R(S)$ are, respectively, WLPT and WSPT ordered sets of the remaining jobs such that $L(S) \cup R(S) = N - \{q\}$ and $L(S) \cap R(S) = \emptyset$. In solving WMAD_WMC (i.e., searching for S_{mean}), one difficulty is that, while we can take S_{mean} to be W-shaped, we cannot fix the identity of its pivot job *a priori*. This leads us, in effect, to try each $q \in N$ as a pivot.

Another difficulty, while solving WMAD_WMC, is that we do not know $c_{\text{mean}}(S_{\text{mean}})$, until we have found S_{mean} itself. This leads us potentially to evaluate $\nu(S, d)$, for a given S , over all $d, d \in \{0, 1/w_N, 2/w_N, \dots, p_N\}$. (The values selected for d ensure that $c_{\text{mean}}(S) = d$.) If q is the pivot job in S as before, we can further impose the constraint that $d \leq c_q(S) \leq d + p_q$, effectively making the pivot job q straddle d . (We can do this since, for a given d , it is easily seen that there is an optimal schedule in which the jobs completing on or before d are WLPT ordered and those starting on or after d are WSPT ordered.)

We can then solve WMAD_WMC roughly as follows: try each $q, q \in N$, as the pivot job; consider all potentially optimal W-shaped schedules with q as the pivot job; over this set (of which, say, S is a member), evaluate $\nu(S, d)$ for all d such that $d \in \{0, 1/w_N, 2/w_N, \dots, p_N\}$ and $c_q(S) - p_q \leq d \leq c_q(S)$, and find an (S, d) pair with $c_{\text{mean}}(S) = d$ which minimizes $\nu(S, d)$ over all available pairs; finally, pick the best pair over all q as the optimal solution. The exact details as to how this is done are given in the next section. In the mean time, it may be worthwhile to look at a W-shaped schedule via a numerical example.

Example 2.1. Consider a 5-job problem with the following parameters and let $\{5, 2, \square, 1, 4\}$ be a W-shaped schedule with job 3 as the pivot job (call it q). Fig. 2.1 shows the schedule (call it S).

j :	1	2	3	4	5
p_j :	1	2	3	3	3
w_j :	3	4	1	1	1

Notice here that, for any d that is an integer multiple of $(1/w_N)$ (0.1 in this case) and between $c_q(S) - p_q$ and $c_q(S)$ (here 5 and 8, respectively), job q in S straddles d . As noted earlier, these d values are the only ones that are appropriate for this S , and we will thus evaluate $\nu(S, d)$ over these values alone. (We say that we consider only these (S, d) pairs.) □

We now present a few basic observations and our first results that are both useful and interesting. (They do not require any particular assumptions about the shape of a schedule S .) However, before we do so, we need to formally introduce WMAD_WmC. To that end, we define $v_{\text{med}}(S) = \nu(S, d)$, where $d = c_{\text{med}}(S)$. Our objective in WMAD_WmC is to find a schedule S_{med} which minimizes $v_{\text{med}}(S)$ over all possible S .

First, consider S with d' and d'' for d . Assume w.l.o.g. that $\nu(S, d') \geq \nu(S, d'')$.

Observation 2.1. $\nu(S, d') \leq \nu(S, d'') + |d' - d''|$.

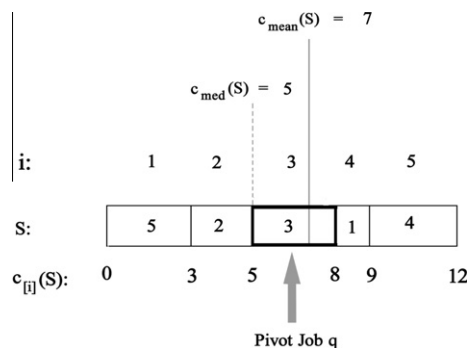


Fig. 2.1. Proper W-shaped schedule.

Proof

$$\begin{aligned} v(S, d') &= (1/w_N) \sum_{1 \leq i \leq n} w_{[i]}(S) |c_{[i]}(S) - d'| = (1/w_N) \sum_{1 \leq i \leq n} w_{[i]}(S) |c_{[i]}(S) - d'' + d'' - d'| \\ &\leq (1/w_N) \sum_{1 \leq i \leq n} w_{[i]}(S) |c_{[i]}(S) - d''| + (1/w_N) \sum_{1 \leq i \leq n} w_{[i]}(S) |d'' - d'| = v(S, d'') + |d' - d''|. \quad \square \end{aligned}$$

The following is easy to visualize; we give a formal proof anyhow.

Observation 2.2. $|c_{\text{mean}}(S) - d| \leq v(S, d)$.

Proof

$$\begin{aligned} |c_{\text{mean}}(S) - d| &\leq \left| (1/w_N) \sum_{1 \leq i \leq n} w_{[i]}(S) c_{[i]}(S) - d \right| = (1/w_N) \left| \sum_{1 \leq i \leq n} w_{[i]}(S) c_{[i]}(S) - w_N d \right| \\ &= (1/w_N) \left| \sum_{1 \leq i \leq n} w_{[i]}(S) c_{[i]}(S) - \sum_{1 \leq i \leq n} w_{[i]}(S) d \right| = (1/w_N) \left| \sum_{1 \leq i \leq n} w_{[i]}(S) [c_{[i]}(S) - d] \right| \\ &\leq (1/w_N) \sum_{1 \leq i \leq n} w_{[i]}(S) |c_{[i]}(S) - d| = v(S, d). \quad \square \end{aligned}$$

The following result, which draws upon the above observations, makes an important connection between $v_{\text{mean}}(S)$ and $v_{\text{med}}(S)$.

Result 2.1. $v_{\text{mean}}(S) \leq 2v_{\text{med}}(S)$.

Proof. From [Observation 2.1](#), using $d' = c_{\text{mean}}(S)$ and $d'' = c_{\text{med}}(S)$, we get

$$v_{\text{mean}}(S) = v(S, c_{\text{mean}}(S)) \leq v(S, c_{\text{med}}(S)) + |c_{\text{mean}}(S) - c_{\text{med}}(S)|.$$

From [Observation 2.2](#), we then get

$$v_{\text{mean}}(S) \leq v(S, c_{\text{med}}(S)) + v(S, c_{\text{med}}(S)) = 2v_{\text{med}}(S). \quad \square$$

Recall that S_{mean} is an optimal schedule for WMAD_WMC (which minimizes $v_{\text{mean}}(S)$ over all S) and that S_{med} is an optimal schedule for WMAD_WmC (which minimizes $v_{\text{med}}(S)$ over all S). Since WMAD_WmC is a relaxation of WMAD_WMC, we get the following.

Observation 2.3. $v_{\text{mean}}(S_{\text{mean}}) \geq v_{\text{med}}(S_{\text{med}})$.

Proof. Because $v_{\text{med}}(S) \leq v_{\text{mean}}(S)$ and $v_{\text{med}}(S_{\text{med}}) \leq v_{\text{med}}(S)$ for any S , we have

$$v_{\text{mean}}(S_{\text{mean}}) \geq v_{\text{med}}(S_{\text{mean}}) \quad \text{and} \quad v_{\text{med}}(S_{\text{mean}}) \geq v_{\text{med}}(S_{\text{med}}).$$

Putting the above inequalities together, we get

$$v_{\text{mean}}(S_{\text{mean}}) \geq v_{\text{med}}(S_{\text{med}}). \quad \square$$

This immediately leads us to the main result of this section.

Result 2.2. $v_{\text{mean}}(S_{\text{med}}) \leq 2v_{\text{mean}}(S_{\text{mean}})$.

Discussion. Applying [Result 2.1](#) and [Observation 2.3](#) in succession, we get

$$v_{\text{mean}}(S_{\text{med}}) \leq 2v_{\text{med}}(S_{\text{med}}) \leq 2v_{\text{mean}}(S_{\text{mean}}). \quad \square$$

[Result 2.2](#) shows that any schedule that is optimal for WMAD_WmC also provides a solution for WMAD_WMC that has a value which is never more than twice the optimal value. This is a direct generalization of a similar result for MAD_MC in Mosheiov [15]. Unfortunately, WMAD_WmC is NP-hard and cannot be solved in polynomial time. We will, however, be able to exploit this result while developing the FPTAS for WMAD_WMC.

3. Dynamic programming algorithm

Our dynamic program implicitly enumerates over all potentially optimal (S, d) pairs, where S is a W -shaped schedule built around a pivot job q and d has a value that is appropriate for S (i.e., it is such that q straddles d). We do this for all q . For a given q , S is of the form $\{L(S), \{q\}, R(S)\}$, where $L(S)$ and $R(S)$ are respectively WLPT and WSPT ordered. We build S inside-out, starting with job q and assigning the remaining jobs in $N - \{q\}$, one at a time and in increasing order of their indexes, either to the front of $L(S)$ or the back of $R(S)$.

For notational purposes, let $p_X = \sum_{j \in X} p_j$ and $w_X = \sum_{j \in X} w_j$, where X is a subset of N . Now, suppose that, at stage k during our construction process, A is a k -job partial schedule consisting of job q (as its pivot) and the first $k - 1$ jobs in $N - \{q\}$. Let d_A be an appropriate d for A . This (A, d_A) pair is characterized by $c_{\text{mean}}(A)$ and $v(A, d_A)$. Similarly, let A' be another k -job partial schedule, for which q is again the pivot job and $d_{A'}$ is an appropriate d . This $(A', d_{A'})$ pair is characterized by $c_{\text{mean}}(A')$ and $v(A', d_{A'})$. We state below the conditions under which the (A, d_A) pair is said to *dominate* the $(A', d_{A'})$ pair. Note here that dominance simply means that we need not carry forward the $(A', d_{A'})$ pair further in our enumeration; it does not mean that we discard the partial schedule A' at this stage (as another pair including A' may survive). (Refer to Fig. 3.1, built on Example 2.1, for an illustration of dominance. Notice that the $(c_{\text{mean}}(A), v(A, d_A))$ values for an (A, d_A) pair appear next to A in this figure.)

Observation 3. If $c_{\text{mean}}(A') = c_{\text{mean}}(A)$, $d_{A'} = d_A$ and $v(A', d_{A'}) > v(A, d_A)$, then the $(A', d_{A'})$ pair cannot lead to an optimal solution upon completion if the (A, d_A) pair does not.

Proof. Assume that $(S', d_{S'})$, obtained upon the completion of $(A', d_{A'})$, leads to an optimal solution. Let U be the set of the last $n - k - 1$ jobs in $N - \{q\}$, ordered as they appear in S' and given by $\{L(U), R(U)\}$, where $L(U)$ ($R(U)$) appears leftmost (rightmost) in S' . It is clear that $S' = \{L(U), A', R(U)\}$ and $d_{S'} = d_{A'} + p_{L(U)}$.

Now, complete (A, d_A) in the same way as above to get (S, d_S) . Clearly, $S = \{L(U), A, R(U)\}$ and $d_S = d_A + p_{L(U)}$. Since $d_A = d_{A'}$, we have $d_S = d_{S'}$ (see Fig. 3.1).

It is possible to see (refer to Fig. 3.1 for perspective) that, after some algebra, we get:

$$w_N c_{\text{mean}}(S') = w_{A'} c_{\text{mean}}(A') + w_U c_{\text{mean}}(U) + w_{R(U)} p_{A'} + w_A p_{L(U)}, \quad \text{and}$$

$$w_N c_{\text{mean}}(S) = w_A c_{\text{mean}}(A) + w_U c_{\text{mean}}(U) + w_{R(U)} p_A + w_A p_{L(U)}.$$

It follows from the above that $c_{\text{mean}}(S') = c_{\text{mean}}(S)$. We have already noted that $d_{S'} = d_S$.

Finally, turning to $v(S', d_{S'})$ and $v(S, d_S)$, we get after some algebra (refer again to Fig. 3.1):

$$w_N v(S', d_{S'}) = w_{A'} v(A', d_{A'}) + w_U v(U, d_{S'} - d_{A'}) + w_{R(U)} p_{A'} + (w_{L(U)} - w_{R(U)}) d_{A'}, \quad \text{and}$$

$$w_N v(S, d_S) = w_A v(A, d_A) + w_U v(U, d_S - d_A) + w_{R(U)} p_A + (w_{L(U)} - w_{R(U)}) d_A.$$

Since $v(A', d_{A'}) > v(A, d_A)$, it should be clear that $v(S', d_{S'}) > v(S, d_S)$.

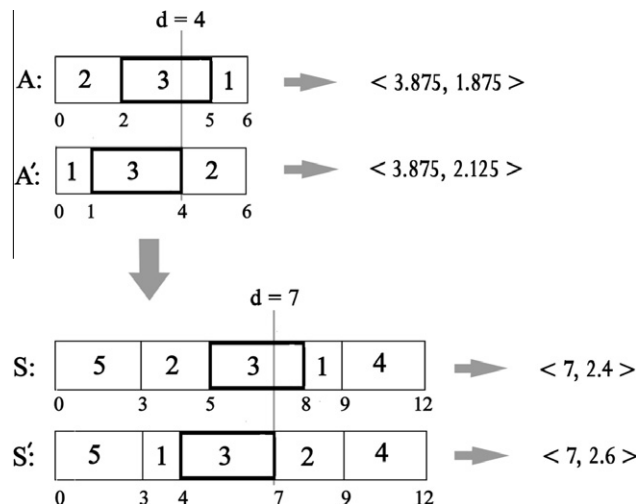


Fig. 3.1. Dominance among partial schedules.

Note at this point that, because of the assumed optimality of $(S', d_{S'})$, $c_{\text{mean}}(S') = d_{S'}$. This implies in turn that $c_{\text{mean}}(S) = d_S$ and further that $v_{\text{mean}}(S') = v(S', d_{S'})$, $v_{\text{mean}}(S) = v(S, d_S)$ and $v_{\text{mean}}(S') > v_{\text{mean}}(S)$. Thus, $(S', d_{S'})$ obtained from $(A', d_{A'})$ cannot yield an optimal solution to WMAD_WMC, contradicting the original premise and proving the observation's validity. \square

We can now describe the dynamic program (which is stated in an enumerative form and which we call DP_OPT) for solving WMAD_WMC exactly. Let Ω_k be the set of the (A, d_A) pairs at stage k of the dynamic program, corresponding to a pivot job q . Also, let Ω be the set of the optimal (S, d) pairs corresponding to each $q, q \in N$.

DP_OPT:

1. Set $\Omega = \{\}$.
2. For each $q \in N$:
 - 2.1. Set $\Omega_1 = \{(\{q\}, d) : d = 0, 1/w_N, 2/w_N, \dots, p_q\}$.
 - 2.2. For $k = 2$ through n :
 - 2.2.1. Let j be the index of the $(k - 1)$ th job in $N - \{q\}$.
 - 2.2.2. For each (B, d_B) pair in Ω_{k-1} :
Add two new pairs to Ω_k : $(\{j\} \cup B, d_B + p_j)$ and $(B \cup \{j\}, d_B)$.
 - 2.2.3. From among all the (A, d_A) pairs in the current Ω_k that have the same $c_{\text{mean}}(A)$ value and the same d_A value, keep one that has the smallest $v(A, d_A)$ value (cf. Observation 3). Let the resulting set be the final Ω_k .
 - 2.3. From Ω_n , find all the (S, d_S) pairs in which $c_{\text{mean}}(S) = d_S$. From among them, find one with the smallest $v(S, d_S)$ value and add it to Ω .
3. From Ω , find an (S, d) pair with the smallest $v(S, d)$ value. Deliver the corresponding schedule S_{mean} as an optimal schedule for WMAD_WMC.

Before we move on, it is instructive to see how an (A, d_A) pair in stage $k - 1$ yields two new pairs for stage k ; Fig. 3.2 below, where the $(c_{\text{mean}}(A), v(A, d_A))$ values for an (A, d_A) pair appear next to or below A , shows this. We now establish the correctness and the time complexity of DP_OPT.

Result 3. DP_OPT solves WMAD_WMC correctly in $O(n^2 p_N^2 w_N^2)$ time.

Proof. The procedure given above is correct as it implicitly enumerates over all legitimate (S, d) pairs. It never discards a pair, which upon completion may lead to an optimal solution (unless, of course, there is yet another pair that is equivalent).

At the end of stage k for a given q , the procedure retains exactly one pair for each distinct combination of $c_{\text{mean}}(A)$ and d_A values. It can be seen that the number of distinct values of $c_{\text{mean}}(A)$, as well as that of d_A , is bounded above by $p_N w_N$. Thus, the cardinality of Ω_k is $O(p_N^2 w_N^2)$. Over the n values of q and the $n - 1$ values of k , this translates to an overall time complexity of $O(n^2 p_N^2 w_N^2)$. \square

DP_OPT can solve MAD_MC if we simply set $w_j = 1$ for all $j \in N$; the time complexity in this case is $O(n^4 p_N)$ (since $w_N = n$). We can do better if we exploit the V-shape property; this obviates the search for a pivot job. In addition to setting $w_j = 1$ for all $j \in N$, we carry out the enumeration in Step 2 of DP_OPT only for $q = 1$ (not for all $q \in N$). Also, for any (S, d) pair that we

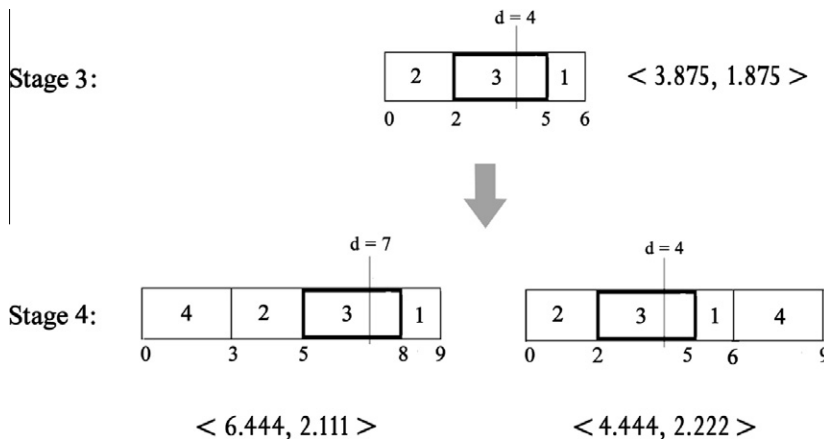


Fig. 3.2. Partial schedule generation in step 2.2.2 of DP_OPT.

consider, we ensure that the jobs scheduled before (after) the smallest job, i.e., job 1, are LPT (SPT) ordered, and further that job 1 either straddles d or appears immediately before or after the job that does; this implies that $d - p_n \leq c_1(S) \leq d + p_n + p_1$. (Recall that n is the longest job.) We address the latter by seeding Ω_1 in Step 2.1 as follows:

$$\Omega_1 = \{(\{1\}, d) : d = -p_n, -p_n + 1/n, \dots, 0, \dots, p_1 + p_n\}.$$

Finally, for any (S, d) pair, we need to make sure that all the jobs in S scheduled before (after) the smallest job start before (complete after) d . We thus never create a new pair in Step 2.2.2 unless this condition is met. By doing all of the above, we save enumerating over the n values of q . The time complexity now becomes $O(n^3 p_N)$, which is the same as what Aneja et al. [1] report for MAD_MC.

4. Fully polynomial time approximation scheme

Our FPTAS for WMAD_WMC, which we call DP_APX, is a mild modification of DP_OPT. Let \bar{v} be an upper bound on the value of the optimal solution to WMAD_WMC; in addition, let \underline{v} be a valid lower bound. We can get these bounds from a 2-approximate solution to WMAD_WmC, obtained in $O(n^2 \log(\max_j \{p_j, w_j\}))$ time via the FPTAS of Erel and Ghosh [8] with $\varepsilon = 1$. Calling the schedule delivered S_{heur} and letting S_{mean} and S_{med} be, as before, the optimal schedules for WMAD_WMC and WMAD_WmC, respectively, we can set $\underline{v} = 1/2 v_{\text{med}}(S_{\text{heur}})$ and $\bar{v} = v_{\text{mean}}(S_{\text{heur}})$. Note that: $v_{\text{mean}}(S_{\text{mean}}) \leq v_{\text{mean}}(S_{\text{heur}})$ (because of the optimality of S_{mean}), and $v_{\text{mean}}(S_{\text{heur}}) \leq 2 v_{\text{med}}(S_{\text{heur}})$ (by Result 2.1). The former inequality establishes the validity of \bar{v} , while the latter shows that $\bar{v} \leq 2 v_{\text{med}}(S_{\text{heur}})$. Note also that: $v_{\text{med}}(S_{\text{heur}}) \leq 2 v_{\text{med}}(S_{\text{med}})$ (by the 2-approximation guarantee), and $v_{\text{med}}(S_{\text{med}}) \leq v_{\text{mean}}(S_{\text{mean}})$ (due to Observation 2.3). Together, they establish that $v_{\text{mean}}(S_{\text{mean}}) \geq 1/2 v_{\text{med}}(S_{\text{heur}})$ and validate \underline{v} . Note at this point that $(\bar{v}/\underline{v}) \leq 4$ for our particular choices of \bar{v} and \underline{v} .

As in DP_OPT, in stage k of DP_APX for a given q , let A be a k -job partial schedule made up of job q (as the pivot) and the first $k - 1$ jobs in $N - \{q\}$, and let d_A be an appropriate d for A . This (A, d_A) pair is associated with $c_{\text{mean}}(A)$ and $v(A, d_A)$. Assume, as before, that U is the set of $n - k$ unscheduled jobs in N at this stage. We can now make the following observation regarding the viability of considering (A, d_A) further during our enumeration.

Observation 4.1. The (A, d_A) pair cannot lead to an optimal solution if the following condition holds: $w_A v(A, d_A) + w_U \min\{d_A, p_A - d_A\} > w_N \bar{v}$.

Proof. Suppose that (A, d_A) is completed with the jobs in U , as in the proof of Observation 3, to yield the optimal (S, d_S) pair with $S = \{L(U), A, R(U)\}$. Because of the supposed optimality of (S, d_S) , we have: $c_{\text{mean}}(S) = d_S$.

Using the expression for $w_N v(S, d_S)$ used in the proof of Observation 3, we can write:

$$\begin{aligned} w_N v_{\text{mean}}(S) &= w_N v(S, d_S) = w_A v(A, d_A) + w_U v(U, d_S - d_A) + w_{R(U)} p_A + (w_{L(U)} - w_{R(U)}) d_A \\ &= w_A v(A, d_A) + w_U v(U, d_S - d_A) + w_{R(U)} [p_A - d_A] + w_{L(U)} d_A \\ &= w_A v(A, d_A) + w_U v(U, d_S - d_A) + [w_{R(U)} + w_{L(U)}] \min\{d_A, p_A - d_A\} \\ &= w_A v(A, d_A) + w_U v(U, d_S - d_A) + w_U \min\{d_A, p_A - d_A\} > w_N \bar{v} \quad (\text{because of the stated condition}) \\ &= w_N v_{\text{mean}}(S_{\text{heur}}). \end{aligned}$$

S cannot thus be optimal. \square

We now obtain a couple of useful bounds when the condition in Observation 4.1 does not hold for an (A, d_A) pair.

Observation 4.2. For the (A, d_A) pair with $w_A v(A, d_A) + w_U \min\{d_A, p_A - d_A\} \leq w_N \bar{v}$:

- (i) $w_A |c_{\text{mean}}(A) - d_A| \leq w_N \bar{v}$; and
- (ii) $w_U \min\{d_A, p_A - d_A\} \leq w_N \bar{v}$.

Proof. The first bound follows from Observation 2.2 and the stated condition. The second follows trivially from the stated condition. \square

Let us suppose, as we have done prior to stating Observation 3, that A' (just like A) is a k -job partial schedule, for which q is the pivot and $d_{A'}$ is an appropriate d for A' . This $(A', d_{A'})$ pair corresponds to $c_{\text{mean}}(A')$ and $v(A', d_{A'})$. Note that A and A' contain the same set of k jobs (i.e., $w_{A'} = w_A$ and $p_{A'} = p_A$).

We can now compute the maximum loss in the final objective function value (call it the v -value) that results if we retain the (A, d_A) pair and discard the $(A', d_{A'})$ pair at stage k of the enumeration.

Observation 4.3. The maximum loss (in the final v -value) incurred when (A, d_A) is retained and $(A', d_{A'})$ is discarded at stage k is given by $(1/w_N)\Delta\alpha + 2(w_U/w_N^2)\Delta\beta$, if the following conditions hold: (i) $|w_A[c_{\text{mean}}(A) - d_A] - w_{A'}[c_{\text{mean}}(A') - d_{A'}]| \leq \Delta\alpha$; (ii) $w_N|d_A - d_{A'}| \leq \Delta\beta$; and (iii) $w_A v(A, d_A) \leq w_{A'} v(A', d_{A'})$. (Note that both $\Delta\alpha$ and $\Delta\beta > 0$.)

Proof. Suppose that a possible completion of the k -job $(A', d_{A'})$ pair yields an optimal n -job $(S', d_{S'})$ pair such that $c_{\text{mean}}(S') = d_{S'}$. Let $S' = \{L(U), A', R(U)\}$, where $U = \{L(U), R(U)\}$ is the set of $n - k$ unscheduled jobs in N that are ordered as in S' . From the competing k -job (A, d_A) pair, construct the n -job (S, d_S) pair such that $S = \{L(U), A, R(U)\}$. We now establish an upper bound on the difference $v_{\text{mean}}(S) - v_{\text{mean}}(S')$. First, from the expressions for $c_{\text{mean}}(S)$, $c_{\text{mean}}(S')$, $v(S, d_S)$ and $v(S', d_{S'})$ used in the proof of **Observation 3**, we get:

$$w_N c_{\text{mean}}(S) - w_N c_{\text{mean}}(S') = w_A c_{\text{mean}}(A) - w_{A'} c_{\text{mean}}(A'); \quad \text{and}$$

$$w_N v(S, d_S) - w_N v(S', d_{S'}) = [w_A v(A, d_A) - w_{A'} v(A', d_{A'})] + [w_{L(U)} - w_{R(U)}](d_A - d_{A'}) \leq (w_U/w_N)\Delta\beta.$$

The above inequality follows from conditions (ii) and (iii) in the statement of **Observation 4.3** and the fact that $w_{L(U)} - w_{R(U)} \leq w_U$.

We now have:

$$w_N v_{\text{mean}}(S) - w_N v_{\text{mean}}(S') \leq [w_N v(S, d_S) - w_N v(S', d_{S'})] + |w_N c_{\text{mean}}(S) - w_N d_S|$$

(from **Observation 2.1** and the assumption that $c_{\text{mean}}(S') = d_{S'}$)

$$\leq (w_U/w_N)\Delta\beta + |[w_N c_{\text{mean}}(S) - w_N c_{\text{mean}}(S')] - [w_N d_S - w_N d_{S'}]|$$

(from the expression for the v -difference above and the assumption that $c_{\text{mean}}(S') = d_{S'}$)

$$\leq (w_U/w_N)\Delta\beta + |[w_A c_{\text{mean}}(A) - w_{A'} c_{\text{mean}}(A')] - [w_N d_S - w_N d_{S'}]|$$

(from the expression for the c -difference above)

$$\leq (w_U/w_N)\Delta\beta + |[w_A c_{\text{mean}}(A) - d_A] - [w_{A'} c_{\text{mean}}(A') - d_{A'}]| - w_U |d_A - d_{A'}|$$

(since $d_S - d_{S'} = d_A - d_{A'}$ and $w_N = w_A + w_U$)

$$\leq (w_U/w_N)\Delta\beta + |w_A c_{\text{mean}}(A) - d_A| - w_{A'} |c_{\text{mean}}(A') - d_{A'}| + w_U |d_A - d_{A'}| \leq \Delta\alpha + 2(w_U/w_N)\Delta\beta$$

(from conditions (i) and (ii) in the statement of **Observation 4.3**).

Dividing both sides by w_N , we finally get: $v_{\text{mean}}(S) - v_{\text{mean}}(S') \leq (1/w_N)\Delta\alpha + 2(w_U/w_N^2)\Delta\beta$. (This is the maximum error DP_APX is liable to make at stage k of the enumeration. Suitable choices of $\Delta\alpha$ and $\Delta\beta$ will allow us to control the extent of this error and execute DP_APX in polynomial time at the same time.) \square

Let $\alpha_{\min}(\alpha_{\max})$ be the minimum (maximum) of $w_A |c_{\text{mean}}(A) - d_A|$ and $\beta_{\min}(\beta_{\max})$ be the minimum (maximum) of $w_N d_A$ over all appropriate members of Ω_k , i.e., the set of k -job (A, d_A) pairs, at stage k of DP_APX. We can now describe the FPTAS.

DP_APX:

1. Set $\Omega = \{\}$. Get \bar{v} and \underline{v} from S_{heur} .

2. For each $q \in N$:

2.1. Set $\pi_1 = p_q$ and $\omega_1 = w_q$.

Also, set $\Delta\alpha = (\varepsilon \underline{v} w_N) / (2n)$ and $\Delta\beta = (\varepsilon \underline{v} w_N^2) / [4n(w_N - \omega_1)]$.

Let $\Delta = \min\{(w_N/w_1)\Delta\alpha, \Delta\beta\}$.

2.1.1. Let $\beta_{\min} = \lceil \max\{0, (w_N/w_1)(\omega_1 \pi_1 - w_N \bar{v})\} \rceil$ and $\beta_{\max} = \lfloor \min\{(w_N^2 \bar{v}) / (w_N - \omega_1), 1/2(w_N \pi_1)\} \rfloor$.

Generate the following set of (C, d_C) pairs if $\beta_{\min} \leq \beta_{\max}$:

$\{(\{q\}, d): w_N d = \beta_{\min}, \beta_{\min} + \lfloor \Delta \rfloor, \beta_{\min} + 2 \lfloor \Delta \rfloor, \dots, \beta_{\max}\}$.

2.1.2. Let $\beta_{\min} = \lceil \max\{1/2(w_N \pi_1), (w_N/w_1)(\omega_1 \pi_1 - w_N \bar{v}), w_N \pi_1 - (w_N^2 \bar{v}) / (w_N - \omega_1)\} \rceil$, and $\beta_{\max} = w_N \pi_1$.

Generate the following set of (C, d_C) pairs if $\beta_{\min} \leq \beta_{\max}$: $\{(\{q\}, d): w_N d = \beta_{\min}, \beta_{\min} + \lfloor \Delta \rfloor, \beta_{\min} + 2 \lfloor \Delta \rfloor, \dots, \beta_{\max}\}$.

2.1.3. Combine the two sets generated in Steps 2.1.1 and 2.1.2 to get Ω_1 .

If Ω_1 is empty, return to Step 2 and try the next q ;

Otherwise, proceed to Step 2.2.

2.2. For $k = 2$ through n :

2.2.1. Let j be the index of the $(k - 1)$ th job in $N - \{q\}$.

Set $\pi_k = \pi_{k-1} + p_j$ and $\omega_k = \omega_{k-1} + w_j$.

2.2.2. For each (B, d_B) pair in Ω_{k-1} :

Add two new pairs to Ω_k :

$(\{j\} \cup B, d_B + p_j)$ and $(B \cup \{j\}, d_B)$.

2.2.3. Delete from Ω_k all (A, d_A) pairs for which

$$\omega_k v(A, d_A) + (w_N - \omega_k) \min\{d_A, \pi_k - d_A\} > w_N \bar{v}.$$

2.2.4. If $k = n$, skip to Step 2.3; continue otherwise.

$$\text{Let } \Delta\alpha = (\varepsilon \underline{v} w_N) / (2n) \text{ and } \Delta\beta = (\varepsilon \underline{v} w_N^2) / [4n(w_N - \omega_k)].$$

2.2.4.1. Consider first the (A, d_A) pairs in the current Ω_k with $d_A \leq 1/2 \pi_k$.

Compute α_{\min} , α_{\max} , β_{\min} and β_{\max} over this subset of Ω_k .

Divide the interval $[\alpha_{\min}, \alpha_{\max}]$ into subintervals of width $\Delta\alpha$ and the interval $[\beta_{\min}, \beta_{\max}]$ into subintervals of width $\Delta\beta$.

Of the (A, d_A) pairs in this subset with their $w_A [c_{\text{mean}}(A) - d_A]$ and $w_N d_A$ values in the same $\Delta\alpha \times \Delta\beta$ rectangle, retain in the final Ω_k one with the smallest $u(A, d_A)$.

2.2.4.2. Consider next the remaining (A, d_A) pairs in the current Ω_k .

Repeat the same process as in Step 2.2.4.1 for these pairs.

2.3. Add all the (S, d_S) pairs in Ω_n to Ω .

3. From Ω , extract an (S, d) pair such that $v_{\text{mean}}(S)$ is minimum, and deliver S as S_{APX} .

We now present the main result of this section, and, indeed, of the whole paper.

Result 4. DP_APX is an FPTAS in that it produces a $(1 + \varepsilon)$ -approximate solution to WMAD_WMC in $O(n^2 \max\{n^2/\varepsilon^2, \log(\max_j\{p_j, w_j\})\})$ time.

Proof. No (S, d_S) pair is discarded in stage n . At stage k ($2 \leq k \leq n - 1$), we do not discard any (A, d_A) pair whose completion will result in an error more than $(1/w_N)\Delta\alpha + 2[(w_N - \omega_k)/w_N^2]\Delta\beta$. (Refer to Observations 4.1 and 4.3.) It is easy to verify that this is true for stage 1 as well. Now, substituting the values of $\Delta\alpha$ and $\Delta\beta$ used in DP_APX, we see that the maximum error at any stage k is limited to $(\varepsilon/n)\underline{v}$. Over n stages, the error can thus accumulate to a maximum of $\varepsilon\underline{v}$. This guarantees the desired $(1 + \varepsilon)$ -approximation.

We now turn to the issue of a polynomial time guarantee. At stage k ($2 \leq k \leq n - 1$), the number of $w_A [c_{\text{mean}}(A) - d_A]$ subintervals in Steps 2.2.4.1 and 2.2.4.2 is bounded by:

$$\begin{aligned} [(\alpha_{\max} - \alpha_{\min})/\Delta\alpha] &\leq [2\max\{|\alpha_{\min}|, |\alpha_{\max}|\}]/\Delta\alpha + 1 \leq (2 w_N \bar{v})/\Delta\alpha + 1 \quad (\text{cf. Observation 4.2}) \leq 4(n/\varepsilon)(\bar{v}/\underline{v}) + 1 \\ &= 16(n/\varepsilon) + 1 \quad (\text{as } \bar{v}/\underline{v} = 4). \end{aligned}$$

As for the number of $w_N d_A$ subintervals, first note that: $(w_N - \omega_k)(\beta_{\max} - \beta_{\min}) \leq w_N^2 \bar{v}$ (this follows from Observation 4.2). The number of subintervals is bounded by:

$$[(\beta_{\max} - \beta_{\min})/\Delta\beta] \leq (w_N^2 \bar{v}) / [(w_N - \omega_k)\Delta\beta] \leq 4(n/\varepsilon)(\bar{v}/\underline{v}) + 1 = 16(n/\varepsilon) + 1.$$

There are clearly $O(n^2/\varepsilon^2)$ $\Delta\alpha \times \Delta\beta$ rectangles at stage k , $2 \leq k \leq n - 1$. For each of these rectangles, at most one (A, d_A) pair is retained; the total number of (A, d_A) pairs left at the end of stage k is thus $O(n^2/\varepsilon^2)$. (The same is true for stages 1 and n as well.) Since q and k take on n and $n - 1$ values, respectively, the total number of pairs considered by DP_APX is $O(n^4/\varepsilon^2)$. It takes $O(n^2 \log(\max_j\{p_j, w_j\}))$ time to compute \bar{v} and \underline{v} up front. The overall time complexity of DP_APX can thus be stated as $O(n^2 \max\{n^2/\varepsilon^2, \log(\max_j\{p_j, w_j\})\})$. \square

In order to get an FPTAS for MAD_MC, we can modify DP_OPT as described at the end of Section 3 and then apply DP_APX subject to these modifications. There are two main points to make: first, we do not have to enumerate over all n values of q ; second, we can get \bar{v} and \underline{v} by solving MAD_mC in $O(n \log(n))$ time via the algorithm due to Kanet [10]. Thus, $\bar{v}/\underline{v} = 4$ as before, and we get an $O(n^3/\varepsilon^2)$ FPTAS for MAD_MC. There is also the minor issue of populating Ω_1 initially; one needs to remember here that $-p_n \leq d \leq p_1 + p_n$ in this case. This issue, however, is easily addressed, and we choose to skip the details.

5. Computational study

In line with Mosheiov [15], we now report the findings of a computational study, where the optimal schedule for the WMAD_WmC problem is used as a heuristic solution for WMAD_WMC (which we already know is 2-approximate). We use the optimal solution to WMAD_WmC, as it provides both lower and upper bounds on the WMAD_WMC optimal solution values; we need both of these for performance evaluation. We use a dynamic program given by De et al. [5] to this end. (For the purpose of obtaining a near-optimal schedule alone, we could as well have used any good heuristic.)

We explore 4 problem sizes: $n = 25, 50, 75$, and 100. For each n , we randomly generate 25 problem instances. The p 's and the w 's are generated independently from two discrete uniform distributions over all integers from 1 to 100. The dynamic

Table 1
Performance evaluation results.

Size (n)	Performance ratio		CPU time (seconds)	
	Average	Maximum	Average	Maximum
25	1.002594	1.006998	0.691	0.937
50	1.000636	1.002101	2.647	3.234
75	1.000414	1.001224	5.969	7.438
100	1.000138	1.000711	9.971	12.312

program is coded in Liberty Basic and run on a Dell laptop (with 1.83 GHz Core 2 Duo CPU and 2 GB RAM) operating under Windows XP. No special efforts have been expended to achieve efficiency. Past experience [5] shows that such efforts could have had resulted in a speedup of at least an order of magnitude.

Table 1 summarizes the results of our study. For each n , it records the average and the maximum value of the “performance ratio” and the CPU time, respectively. The reported performance ratio, computed as $v_{\text{mean}}(S_{\text{med}})/v_{\text{med}}(S_{\text{med}})$, is a surrogate for the true one given by $v_{\text{mean}}(S_{\text{med}})/v_{\text{mean}}(S_{\text{mean}})$. Notice that $v_{\text{med}}(S_{\text{med}}) \leq v_{\text{med}}(S_{\text{mean}}) \leq v_{\text{mean}}(S_{\text{mean}})$ and further that the reported ratio is a conservative estimate of the true one. This implies that, if the reported ratio is small (close to 1) and the reported performance good, the true ratio is even smaller and the true performance even better than reported.

Table 1 shows that the optimal WMAD_WmC schedule S_{med} provides a very good solution to WMAD_WMC (well within 1% of its optimal solution value), much better than what its 2-approximate guarantee promises. This agrees with what Mosheiov [15] has found for MAD_MC and MAD_mC. Also, the results indicate that the performance ratio improves as the problem size increases. This seems to agree with the theoretical finding by Mosheiov [15] that S_{med} is asymptotically optimal for WMAD_WMC in the unweighted case.

Note that the time to obtain S_{med} is significantly smaller than what one would expect for S_{mean} . However, one can easily replace S_{med} with a heuristically obtained V-shaped schedule. This can be done at great time savings and no significant loss of accuracy by using a scheme such as the one given in De et al. [5].

Whether one is solving WMAD_WmC exactly or approximately, one is searching over V-shaped schedules only. As we have seen, such a search can quickly produce a very good schedule. We have also seen that this schedule continues to be good for WMAD_WMC. The upshot is that one can solve WMAD_WMC quite effectively by searching over V-shaped schedules. This has been a conclusion reached at by Aneja et al. [1] as well.

6. Conclusion

In this paper, we have addressed the WMAD_WMC problem, for which only minimal results have been available thus far. We have established a connection between the optimal solutions to this problem and the better known WMAD_WmC problem. We have further given a pseudo-polynomial time dynamic program (DP_OPT) and an FPTAS (DP_APX) for the exact and the approximate solution, respectively, of WMAD_WMC. We have indicated briefly how the above algorithms specialize to the MAD_MC problem. Finally, we have shown via a computational study that the optimal schedule for WMAD_WmC is near-optimal for WMAD_WMC.

To conclude, we may note that DP_OPT makes available to us, in the form of Ω_n at the end of Step 2.2, a set of (S, d) pairs that are efficient (nondominated) w.r.t. $v_{\text{mean}}(S)$ and $c_{\text{mean}}(S)$, where S is a W-shaped schedule with pivot job q . If, in Step 2.3, we now add all the members of Ω_n (rather than just the optimal one) to Ω , and if, in Step 3, we apply Step 2.2.3 to Ω , we get a completely representative set of W-shaped efficient schedules. While it is not clear at this time if all efficient schedules are necessarily W-shaped, the set delivered by DP_OPT (with the suggested modifications) nevertheless provides us with a rich and interesting subset of such schedules.

References

- [1] Y.P. Aneja, S.N. Kabadi, A. Nagar, Minimizing weighted mean absolute deviation of flow times in single machine systems, *Naval Research Logistics* 45 (1998) 297–311.
- [2] K.R. Baker, G.D. Scudder, Sequencing with earliness and tardiness penalties: a review, *Operations Research* 38 (1990) 22–36.
- [3] X. Cai, Minimization of agreeably weighted variance in single machine systems, *European Journal of Operational Research* 85 (1995) 576–592.
- [4] J. Cheng, W. Kubiak, A half-product based approximation scheme for agreeably weighted completion time variance, *European Journal of Operational Research* 162 (2005) 45–54.
- [5] P. De, J.B. Ghosh, C.E. Wells, CON due-date determination and sequencing, *Computers & Operations Research* 17 (1990) 333–342.
- [6] P. De, J.B. Ghosh, C.E. Wells, On the minimization of completion time variance with a bicriteria extension, *Operations Research* 40 (1992) 1148–1155.
- [7] S.E. Eilon, I.G. Chowdhury, Minimizing waiting time variance in the single machine problem, *Management Science* 23 (1977) 567–575.
- [8] E. Erel, J.B. Ghosh, FPTAS for half-products minimization with scheduling applications, *Discrete Applied Mathematics* 156 (2008) 3046–3056.
- [9] N.G. Hall, M.E. Posner, Earliness-tardiness scheduling problems, I: weighted deviation of completion times about a common due date, *Operations Research* 39 (1991) 836–846.
- [10] J.J. Kanet, Minimizing variation of flow time in single machine systems, *Management Science* 27 (1981) 1453–1459.
- [11] M.Y. Kovalyov, W. Kubiak, A fully polynomial approximation scheme for the weighted earliness-tardiness problem, *Operations Research* 47 (1999) 757–761.

- [12] W. Kubiak, Completion time variance minimization on a single machine is difficult, *Operations Research Letters* 14 (1993) 49–59.
- [13] W. Kubiak, J. Cheng, M.Y. Kovalyov, Fast fully polynomial approximation schemes for minimizing completion time variance, *European Journal of Operational Research* 137 (2002) 303–309.
- [14] A.G. Merten, M.E. Muller, Variance minimization in single-machine sequencing problems, *Management Science* 18 (1972) 518–528.
- [15] G. Mosheiov, Minimizing mean absolute deviation of job completion times from the mean completion time, *Naval Research Logistics* 47 (2000) 657–668.