Operations Research Letters 38 (2010) 127–128

Contents lists available at ScienceDirect

Operations Research Letters

journal homepage: www.elsevier.com/locate/orl

A faster algorithm for a due date assignment problem with tardy jobs

Christos Koulamas *

Department of Decision Sciences and Information Systems, Florida International University, Miami, FL 33199, United States

ARTICLE INFO

Article history:
Received 3 November 2008
Accepted 7 October 2009
Available online 5 November 2009

Keywords:
Single-machine sequencing
Weighted number of tardy jobs
Due date assignment

ABSTRACT

The single-machine due date assignment problem with the weighted number of tardy jobs objective, (the *TWNTD* problem), and its generalization with resource allocation decisions and controllable job processing times have been solved in $O(n^4)$ time by formulating and solving a series of assignment problems. In this note, a faster $O(n^2)$ dynamic programming algorithm is proposed for the *TWNTD* problem and for its controllable processing times generalization in the case of a convex resource consumption function.

1. Introduction

The single-machine due date assignment problem with the weighted number of tardy jobs objective (the *TWNTD* problem) can be defined as follows: There are n jobs available at time zero; job j has a positive processing time p_j and a positive weight w_j. The objective is to determine the due date d_j of each job j and a job sequence so that the function

$$TC = a\sum_{j=1}^{n} d_j + \sum_{j=1}^{n} w_j U_j \tag{1}$$

is minimized where a is the positive unit due date assignment cost and U_j is the tardiness indicator of job j defined as follows: $U_j = 1$ if $C_j > d_j$ and $U_j = 0$ if $C_j \leq d_j$ where C_j denotes the completion time of job j.

Shabtay and Steiner [1] proposed an $O(n^4)$ algorithm for the *TWNTD* problem by solving a series of assignment problems. In this note, we show that the *TWNTD* problem can be solved by dynamic programming (DP) in $O(n^2)$ time by utilizing additional information about the ordering of the jobs in an optimal sequence. An extension of the $O(n^2)$ DP algorithm to the more general resource allocation/scheduling problem with controllable job processing times and a convex resource allocation function is also presented.

2. A DP $O(n^2)$ algorithm for the *TWNTD* problem

Shabtay and Steiner [1] proved some useful properties of an optimal *TWNTD* solution summarized next: Let $[j]$ denote the job occupying the j position in an optimal *TWNTD* sequence S^*; $p_{[j]}$, $d_{[j]}$, $C_{[j]}$, $w_{[j]}$ and $U_{[j]}$ are defined analogously. Let E, T denote the sets of the non-tardy (early) jobs and tardy jobs respectively in S^*. According to Shabtay and Steiner [1], there is no-inserted idle time in S^* and the jobs in E are sequenced in the shortest-processing-time (SPT) order followed by the jobs in T sequenced in any order. Consequently, an optimal sequence can be constructed by sequencing the jobs in E in SPT order followed by the jobs in T also sequenced in SPT order. Furthermore, $d_{[j]} = C_{[j]}$ if $j \in E$ and $d_{[j]} = 0$ if $j \in T$. Assume that $|E| = l^*$ in S^*; then, the above properties facilitate the rewriting of (1) as follows:

$$\begin{aligned} TC(l^*) &= a\sum_{j=1}^{l^*} C_{[j]} + \sum_{j=l^*+1}^{n} w_{[j]} \\ &= a\sum_{j=1}^{l^*} (l^* - j + 1)p_{[j]} + \sum_{j=l^*+1}^{n} w_{[j]}. \end{aligned} \tag{2}$$

The contribution of the $[k]$ early job ($k \leq l^*$) to the objective function (2) is $a(l^* - k + 1)p_k$ and the contribution of any tardy job j to the objective function (2) is w_j. This observation combined with the SPT ordering of the jobs in both the E and T sets reduces the sequencing problem to a partitioning problem and therefore facilitates the solution of the *TWNTD* problem in $O(n^2)$ time by the following backward dynamic programming (DP) algorithm. The proof of correctness is embedded in the construction and presentation of the algorithm.

Assume that the jobs have been re-indexed in the SPT order. Consider all schedules for the *TWNTD* problem defined on the jobs $j, \ldots, n$ subject to the condition that the first job starts at time 0, the jobs are executed with no-inserted idle time, and k jobs among them are early where $k \leq n - j + 1$. Define any such schedule to be in state (j, k). The remaining $j - 1$ jobs can be scheduled optimally

* Tel.: +1 305 348 3309.
E-mail address: koulamas@fiu.edu.

0167-6377/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.orl.2009.10.013

by considering only an optimal schedule with minimum objective value (2) among all schedules in state (j, k).

Let $F_j(k)$ denote the optimal solution value of any schedule in state (j, k) and let $\sigma(j, k)$ be any schedule in state (j, k) with solution value $F_j(k)$. This schedule $\sigma(j, k)$ must have been constructed by taking one of the following two decisions in a previous state:

1. schedule job j early: in this case $\sigma(j, k)$ must have been obtained from schedule $\sigma(j + 1, k - 1)$ and $F_j(k) = F_{j+1}(k - 1) + akp_j$.
2. schedule job j tardy: in this case $\sigma(j, k)$ must have been obtained from schedule $\sigma(j + 1, k)$ and $F_j(k) = F_{j+1}(k) + w_j$.

The dynamic programming recursion is as follows. The initialization is

$$F_j(k) = \begin{cases} 0, & \text{if } j = 0 \text{ and } k = 0, \\ \infty, & \text{otherwise,} \end{cases}$$

and the recursion for $j = n, \ldots, 1, k = 0, \ldots, j$ is

$$F_j(k) = \min\{F_{j+1}(k - 1) + akp_j, F_{j+1}(k) + w_j\}. \tag{3}$$

The optimal solution is then found as $F^* = \min_{0 \le k \le n} F_1(k)$. The optimal E and T sets and the resulting optimal due dates are found by backtracking. The algorithm runs in $O(n^2)$ time.

3. Extensions

Shabtay and Steiner [2] extended their $O(n^4)$ algorithm for the *TWNTD* problem to the more general due date assignment problem with resource allocation decisions, controllable job processing times, and the weighted number of tardy jobs objective. In that case, the objective function (1) generalizes to

$$TC = a\sum_{j=1}^{n} d_j + \sum_{j=1}^{n} w_j U_j + \gamma C_{\max} + \sum_{j=1}^{n} u_j v_j \tag{4}$$

where u_j is the amount of resource allocated to job j, v_j is the cost of one unit of resource allocated to job j, $C_{\max}$ is the maximum job completion time (makespan) of the sequence and γ is the cost of one unit of operation time.

In the case of a convex resource consumption function, the actual processing time of job j is given as $p_j(u_j) = (\omega_j/u_j)^{\lambda}$ where ω_j is the workload of the processing operation of job j and $\lambda > 0$ is a positive constant. In that case, Shabtay and Steiner [2] showed that the objective function (4) can be written as

$$TC(l^*) = \tilde{\lambda} \times \left\{ \sum_{j=1}^{l^*} \theta_{[j]} \times [a(l^* - j + 1) + \gamma]^{1/(\lambda+1)} + \sum_{j=l^*+1}^{n} \theta_{[j]} \times \gamma^{1/(\lambda+1)} \right\} + \sum_{j=l^*+1}^{n} w_{[j]} \tag{5}$$

where $\theta_{[j]} = (\omega_{[j]} \times v_{[j]})^{\lambda/(\lambda+1)}$ and $\tilde{\lambda} = \lambda^{-\lambda/(\lambda+1)} + \lambda^{1/(\lambda+1)}$. Shabtay and Steiner [2] then proceeded to solve the problem, to be called the *CRANTD* problem from now on, in $O(n^4)$ time by solving a series of assignment problems similar to the ones solved in the case of the *TWNTD* problem.

The similarity of the objective functions (2) and (5) can be exploited to solve the *CRANTD* problem in $O(n^2)$ time by implementing the DP algorithm presented in the previous section. According to (5), the contribution of job $[j]$ ($j \le l^*$) to $TC(l^*)$ is $\tilde{\lambda}\theta_{[j]}[a(l^* - j + 1) + \gamma]^{1/(\lambda+1)}$ if $j \in E$ and $\tilde{\lambda}\theta_{[j]}\gamma^{1/(\lambda+1)} + w_{[j]}$ if $j \in T$; the above contributions have similar structure (with $\theta_{[j]}$ in the place of $p_{[j]}$) with the corresponding contributions in the *TWNTD* problem. Furthermore, similar to the SPT ordering in the *TWNTD* problem, the optimal sequencing of the jobs in the E and T sets is according to the non-decreasing θ_j order. The above observations indicate that the DP algorithm of the previous section can be utilized to solve the *CRANTD* problem in $O(n^2)$ time by re-indexing the jobs in non-decreasing θ_j order and rewriting the recursive equation (3) as

$$F_j(k) = \min\{F_{j+1}(k - 1) + \tilde{\lambda}\theta_j(ak + \gamma)^{1/(1+\lambda)}, F_{j+1}(k) + \tilde{\lambda}\theta_j\gamma^{1(\lambda+1)} + w_j\}.$$

Acknowledgement

We would like to thank the Associate Editor for suggesting the Dynamic Programming Algorithm and for his/her other suggestions that helped us to improve an earlier version of this note.

References

[1] D. Shabtay, G. Steiner, Two due date assignment problems in scheduling a single machine, Operations Research Letters 34 (2006) 683–691.
[2] D. Shabtay, G. Steiner, Optimal due date assignment and resource allocation to minimize the weighted number of tardy jobs on a single machine, Manufacturing & Service Operations Management 9 (2007) 332–350.