European Journal of Operational Research 272 (2019) 43–49

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Precedence theorems and dynamic programming for the single-machine weighted tardiness problem

Salim Rostami[a,b], Stefan Creemers[a,b], Roel Leus[b,*]

[a] Management Department, IÉSEG School of Management, (LEM-CNRS 9221), Lille, France
[b] ORSTAT, KU Leuven, Belgium

ARTICLE INFO

Article history:
Received 18 September 2017
Accepted 4 June 2018
Available online 15 June 2018

Keywords:
Scheduling
Single machine
Precedence constraints
Weighted tardiness
Dynamic programming

ABSTRACT

We tackle precedence-constrained sequencing on a single machine in order to minimize total weighted tardiness. Classic dynamic programming (DP) methods for this problem are limited in performance due to excessive memory requirements, particularly when the precedence network is not sufficiently dense. Over the last decades, a number of precedence theorems have been proposed, which distinguish dominant precedence constraints for a job pool that is initially without precedence relation. In this paper, we connect and extend the findings of the foregoing two strands of literature. We develop a framework for applying the precedence theorems to the precedence-constrained problem to tighten the search space, and we propose an exact DP algorithm that utilizes a new efficient memory management technique. Our procedure outperforms the state-of-the-art algorithm for instances with medium to high network density. We also empirically verify the computational gain of using different sets of precedence theorems.

1. Introduction

We consider a set $N = \{1, \ldots, n\}$ of jobs (activities) and a set E of precedence constraints: for any $i, j \in N$, if $(i, j) \in E$ then job i should be scheduled before job j. More specifically, E is a strict partial order on N, i.e., it is irreflexive (pairs $(j, j) \notin E$), asymmetric (if $(i, j) \in E$ then $(j, i) \notin E$), and transitive (if (i, j), $(j, k) \in E$ then $(i, k) \in E$). Associated with each job $i \in N$ is a processing time $p_i \in \mathbb{N}_0$, a due date $d_i \in \mathbb{N}$ and a tardiness weight $w_i \in \mathbb{N}_0$. All jobs are available at time 0 to be processed on a single continuously available machine. The problem is to find a sequence $\mathbf{s} = (s_1, s_2, \ldots, s_n)$ of the jobs that minimizes the total weighted tardiness

$$T(\mathbf{s}) = \sum_{i \in N} w_i \max\{0, C_i - d_i\},$$

where $C_i = \sum_{j=1}^{\ell} p_{s_j}$ is the earliest completion time of job i, and $s_\ell = i$. We define $B_i^E = \{j \in N | (j, i) \in E\}$ and $A_i^E = \{j \in N | (i, j) \in E\}$ as the job sets that should be processed before and after i according to E, respectively. Using the notation of Graham, Lawler, Lenstra, and Rinnooy Kan (1979), this problem is denoted by $1|\text{prec}|\sum w_j T_j$. The problem is strongly NP-hard (Lawler, 1977).

Two related problems have received quite some attention in the scheduling literature. The single-machine scheduling problem to minimize total weighted tardiness, $1||\sum w_j T_j$, has been surveyed by Abdul-Razaq, Potts, and Van Wassenhove (1990), who describe various dynamic programming (DP) and branch-and-bound (B&B) algorithms. Potts and Van Wassenhove (1985) propose a B&B algorithm that solves instances with up to 50 jobs to optimality within practical time and memory limits. Tanaka, Fujikuma, and Araki (2009) extend the Successive Sublimation DP (SSDP) of Ibaraki and Nakamura (1994) and solve relatively large instances with up to 300 jobs. The precedence-constrained single-machine scheduling problem to minimize total weighted completion time, $1|\text{prec}|\sum w_j C_j$, has been studied by, among others, Sidney (1975), Lawler (1978), Potts (1985), Hoogeveen and van de Velde (1995), van de Velde (1995), Margot, Queyranne, and Wang (2003), Correa and Schulz (2005), and Schulz and Uhan (2011). Instances with up to 100 jobs were solved to optimality already 30 years ago (Potts, 1985).

In contrast to the two aforementioned problems, the literature on $1|\text{prec}|\sum w_j T_j$, which is a generalization, is rather scarce. Schrage and Baker (1978) propose a DP method, the performance of which is very limited mainly due to memory insufficiency. Tanaka and Sato (2013) propose an extension of the algorithm of Tanaka et al. (2009) for the precedence-constrained problem that solves instances with up to 100 jobs (within practical time and memory limits) when the density of the precedence network is very low or very high.

* Corresponding author.
E-mail addresses: salim.rostami@kuleuven.be, s.rostami@ieseg.fr (S. Rostami), stefan.creemers@kuleuven.be, s.creemers@ieseg.fr (S. Creemers), roel.leus@kuleuven.be (R. Leus).

https://doi.org/10.1016/j.ejor.2018.06.004
0377-2217/

Davari, Demeulemeester, Leus, and Talla Nobibon (2016) also report computational results for this problem, although their algorithm is developed for a generalized variant with release dates and deadlines; their algorithm solves instances with up to 50 activities.

2. Precedence theorems

Below, we will distinguish the set E of *technological* precedence constraints from the set D of all *dominant* precedence constraints, where a precedence constraint (i, j) is dominant iff there is at least one optimal solution in which i precedes j. Seeing that all feasible solutions respect E, we have $D \supseteq E$. In other words, D is the union of all optimal complete orders. A set of precedence constraints is called *acyclic* only if it is transitive and asymmetric. If there exist multiple optimal solutions then D is not acyclic. A *selection* $S \subseteq D$ is said to be dominant if there is at least one optimal solution that respects all its constraints. Consequently, acyclicity is a necessary but not sufficient condition for the dominance of sets of precedence constraints. Below, we describe precedence theorems and dominance rules to identify a dominant selection that extends E.

2.1. Precedence theorems for $E = \emptyset$

The three precedence theorems that Emmons (1969) proposes, are arguably some of the most fruitful results for $1||\Sigma T_j$; most of the exact approaches rely on these theorems. Later on, Rinnooy Kan, Lageweg, and Lenstra (1975) and Rachamadugu (1987) have extended Emmons' results to the weighted tardiness case $1||\sum w_j T_j$. These theorems distinguish dominant precedence constraints for a job pool with $E = \emptyset$. Starting from $S = \emptyset$ and using Emmons' theorems, one can add job pairs to S in an iterative fashion. Next, by solving the problem instance with precedence constraints S to optimality, an optimal solution to the original instance with $E = \emptyset$ can be found. In line with Emmons (1969) and Rinnooy Kan et al. (1975), for any $X \subseteq N$, we define $P(X) = \sum_{i \in X} p_i$ and $\bar{X} = N \setminus X$. Similar to B_i^E and A_i^E, we define B_i^S and A_i^S based on S instead of E. Given a dominant S, and $i, j \in N$, Emmons' conditions are as follows:

E1. $p_i \le p_j$ and $w_i \ge w_j$ and $d_i \le \max\{d_j, P(B_j^S) + p_j\}$.
E2. $w_i \ge w_j$ and $d_j \ge \max\{d_i, P(\bar{A}_i^S) - p_j\}$.
E3. $d_j \ge P(\bar{A}_i^S)$.

Emmons (1969) proves that when $E = \emptyset$, any of these conditions is sufficient to conclude $(i, j) \in D$. More recently, Kanet (2007) has generalized Emmons' results with seven new conditions (K1 to K7). These are stated in Appendix. Emmons (1969) and Kanet (2007) show that combining the dominant constraints that are identified by these theorems iteratively does not remove all optimal solutions, i.e., any thus-obtained S is dominant iff it is acyclic.

Given a dominant S and job pair (i, j), we define $I(S, i, j)$ as the indicator function of Emmons' and Kanet's theorems that returns 1 if (i, j) satisfies at least one condition, and 0 otherwise. Therefore, when $E = \emptyset$, $I(S, i, j) = 1$ implies $(i, j) \in D$. We also define $C(S) = E \cup \{(i, j) | I(S, i, j) = 1\}$. When $E = \emptyset$ then $C(S) \subseteq D$, but $C(S)$ is not necessarily acyclic. Moreover, extending S iteratively can only improve the theorem conditions for other job pairs to be identified as dominant. Hence, for given dominant S_1 and S_2 the following result is intuitive.

Proposition 1. *If $S_1 \subset S_2$ then $C(S_1) \subseteq C(S_2)$.*

2.2. Extended precedence theorems for general E

With a general set E and for any (i, j), the acyclicity of $E \cup \{(i, j)\}$ becomes a necessary condition for the dominance of (i, j). Furthermore, the precedence theorems that were discussed in Section 2.1 may not be applicable as is. Consider the example depicted in Fig. 1, with $E = \{(1, 3), (2, 4)\}$ and $S = E$. We investigate an additional precedence constraint from job 3 to job 2. Since $I(E, 3, 2) = 1$ (based on K1, K4 and K5), we add the pair (3,2) to S. As depicted in Fig. 1(c), (3,2) implies the transitive edges (1,2), (1,4) and (3,4). Thus, we end up with the sequence $\mathbf{s}^1 = (1, 3, 2, 4)$ with $T(\mathbf{s}^1) = 159$, while for the optimal sequence $\mathbf{s}^* = (2, 4, 1, 3)$, $T(\mathbf{s}^*) = 119$. The two transitive edges (1,2) and (1,4) are not dominant, and consequently remove the optimal solutions. This counterexample shows that with general E, Kanet's and Emmons' conditions cannot be directly invoked, i.e., $I(S, i, j) = 1$ is not sufficient to conclude $(i, j) \in D$. Hence, if $E \neq \emptyset$ then $C(S)$ is not necessarily a subset of D.

Kanet (2007) uses "swap" and "insert-after" strategies to prove his dominance theorems. Conditions E2–3 and K4–7 are obtained via the insert-after strategy, while Conditions E1 and K1–3 are derived using the swap strategy. Condition K1 generalizes E1, K4 and K5 generalize E2, K7 is the same as E3, and K2, K3 and K6 are entirely new in the sense that they can lead to the conclusion that a pair $(i, j) \in D$ even when $w_i < w_j$.

An illustration of the swap and insert-after strategies for general E and a given dominant S is provided in Fig. 2, where $\beta_i = (M \cap B_i^E)$, $\alpha_j = (M \cap A_j^E)$ and $\gamma_{ij} = M \setminus (\alpha_j \cup \beta_i)$. The symbol M represents the set of intermediate jobs between i and j. "From" represents *any* sequence that respects S, and "To" is the resulting sequence after swapping j and i or inserting j after i. The latter sequence respects E but not necessarily S, i.e., a number of dominant precedence constraints in $S \backslash E$ might be violated. A sufficient condition for the dominance of (i, j) has the structure

$$\mathrm{LB}(\mathrm{TI}(i)) \ge \mathrm{UB}(\mathrm{TD}(j)) + \mathrm{UB}(\mathrm{TD}(\gamma_{ij})) + \mathrm{UB}(\mathrm{TD}(\alpha_j)), \tag{1}$$

where $\mathrm{LB}(\cdot)$ and $\mathrm{UB}(\cdot)$ are lower and upper bound functions, respectively, $\mathrm{TI}(i)$ is the tardiness improvement of job i, and $\mathrm{TD}(i)$ the tardiness degradation. Note that $\mathrm{TD}(\beta_i) = 0$.

If an activity pair (i, j) satisfies Condition (1) for *every* feasible M and $G(N, E \cup \{(i, j)\})$ is acyclic, then if j precedes i in a given schedule, we can exchange the two jobs without increasing the tardiness function. Thus, for an acyclic set of activity pairs $\{(i, j), (k, l), \ldots\}$ that each satisfy Condition (1), any optimal schedule that is not compatible with one or more of these pairs cannot be harmed by making as many interchanges as necessary to obtain an optimal schedule that respects all the pairs. Therefore, if the "To" sequence does not respect S, then by a finite number of swaps and insert-afters, it can be transformed into a sequence that respects S, such that the final sequence is at least as good as the intermediate sequences. Given an instance $G(N, E)$, let $V \subseteq D$ be the set of all activity pairs that satisfy Condition (1). We conclude:

Proposition 2. *Any $S \supseteq E$ for which $(S \backslash E) \subseteq V$ is dominant iff S is acyclic.*

Hence, we search for an inclusion-maximal acyclic $S \supseteq E$ such that $(S \backslash E) \subseteq V$.

In Condition (1), the completion times of jobs i and j depend on $P(\alpha_j)$ and $P(\beta_i)$, so $\mathrm{TI}(i)$ and $\mathrm{TD}(j)$ depend on β_i and α_j. Also, $\mathrm{TD}(\alpha_j)$ can be positive in both strategies. Finally, even if $p_i \le p_j$, the value $\mathrm{TD}(\gamma_{ij})$ can still be positive in the swap strategy. We therefore extend Emmons' and Kanet's theorems under the extra requirement that $\alpha_j = \beta_i = \emptyset$.

Proposition 3. *$A_j^E \subseteq A_i^S$ is a sufficient condition for $\alpha_j = \emptyset$.*

Proof. Remember that $\alpha_j = M \cap A_j^E$. The requirement that α_j is empty means all jobs in A_j^E are scheduled after job i. Intuitively, the condition $A_j^E \subseteq A_i^E$ is sufficient to ensure $\alpha_j = \emptyset$. Since the "From"