Computers & Operations Research 64 (2015) 245–261

Contents lists available at ScienceDirect

Computers & Operations Research

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

History-dependent scheduling: Models and algorithms for scheduling with general precedence and sequence dependence

Niraj Ramesh Dayama [a,b,d,*], Mohan Krishnamoorthy [a,d], Andreas Ernst [c], Narayan Rangaraj [b], Vishnu Narayanan [b]

[a] IITB Monash Research Academy, IIT Bombay, Powai, Mumbai 400076, India
[b] Industrial Engineering and Operations Research, IIT Bombay, Powai, Mumbai 400076, India
[c] CSIRO Mathematical and Information Sciences, Private Bag 10, Clayton South MDC, Vic. 3169, Australia
[d] Department of Mechanical and Aerospace Engineering, Monash University, Melbourne, Victoria 3168, Australia

A R T I C L E I N F O

Available online 23 June 2015

Keywords:
Sequence-dependent scheduling
Crane scheduling
Fixed interval scheduling
History-dependent scheduling
Combinatorial optimization
Integer programming

A B S T R A C T

In this paper, we extend job scheduling models to include aspects of *history-dependent scheduling*, where setup times for a job are affected by the aggregate activities of *all* predecessors of that job. Traditional approaches to machine scheduling typically address objectives and constraints that govern the relative sequence of jobs being executed using available resources. This paper optimises the operations of multiple unrelated resources to address sequential *and* history-dependent job scheduling constraints along with time window restrictions. We denote this consolidated problem as the *general precedence scheduling problem* (GPSP). We present several applications of the GPSP and show that many problems in the literature can be represented as special cases of history-dependent scheduling. We design new ways to model this class of problems and then proceed to formulate it as an integer program. We develop specialized algorithms to solve such problems. An extensive computational analysis over a diverse family of problem data instances demonstrates the efficacy of the novel approaches and algorithms introduced in this paper.

1. Introduction

A recurring theme in the modeling of several optimization problems (encountered while scheduling multiple operations) is that the time duration required for any task depends on the sequence in which the tasks are executed. Most approaches that solve such sequence-dependent task scheduling problems discuss only the “immediate” precedence of tasks. Immediate precedence refers to schedules, where two distinct tasks (or activities or operations) are executed immediately after each other (by the same resource). Such an immediate precedence between a pair of tasks may involve some additional setup time, change-over effort, monetary cost or penalty. The immediate precedence feature may present itself in the objective function or in some of the constraints of the scheduling problem.

In addition, several scheduling problems also involve *time windows*. This means that certain operations need to be performed within specific time bounds (or time windows). In addition, even some resources may only be available within certain time bounds. Allocation of these resources or operations outside their prescribed time windows may either be infeasible or may involve considerable additional costs. The class of problems referred to as *fixed interval scheduling problems* considers the scheduling of resources for machine scheduling operations within fixed time windows, but without any sequence-dependence (for example, see [19]).

Many scheduling problems also involve a combination of (immediate-precedence) sequence-dependence and time bounds. For example, the traveling salesman problem (TSP) with time windows (TSPW) has been discussed in many seminal papers including [27,24,14], and others. Such problems and the broad class of vehicle routing problems (VRP) encounter sequence-dependent costs (for immediate-precedence of nodes traversed) in the objective and time bounds in constraints.

Another interesting feature that appears in several problems is one of *aggregate general precedence* (as opposed to immediate precedence). This feature relies on aggregate history and not just immediate history of a job (or task). In such problems, the costs/restrictions of executing one task (say j) depend on whether or not

* Corresponding author at: IITB Monash Research Academy, IIT Bombay, Powai, Mumbai 400076, India.
E-mail address: niraj.ramesh@iitb.ac.in (N.R. Dayama).

http://dx.doi.org/10.1016/j.cor.2015.06.007
0305-0548/

some other task (say k) has been already initiated at some point earlier on in the sequence. We are not interested in whether or not the other task k has been performed *immediately* before task j. We need to know whether or not k has already been initiated *any time* before j. Aggregate general precedence considerations may be incorporated in the objective function or may be a part of some of the constraints in the model. Such problems have been studied under the context of *history dependent scheduling* (see [20,1]). To avoid ambiguity, we clarify that our discussion is restricted to problems with deterministic parameters (without any stochastic or probabilistic elements).

In this paper, we propose to combine features of immediate-precedence, time bounds and *aggregate* history-dependence in scheduling tasks. Further, we assume that there are multiple identical and parallel resources available simultaneously for the execution of jobs. Given that each of these features individually leads to non-trivial (NP-hard) problems, the unified problem is clearly NP-hard. We term the unified problem as the *general precedence scheduling problem* (or GPSP).

In this paper we present definitions, models, formulations and algorithms for the GPSP. The document is organized as follows: Section 2 sketches the reasons for our interest in studying the GPSP. Section 3 defines the problem formally while Section 4 reviews the relevant literature to identify previous contributions of interest to us. Section 5 develops suitable models for GPSP and Section 6 designs algorithms to solve GPSP. Section 7 discusses the problem data instances for GPSP and finally Section 8 presents the results of extensive computational analysis to demonstrate the efficacy of techniques proposed in this paper.

2. Motivation for studying the gpsp

Our primary motivation behind the GPSP is to optimize the scheduling of cranes used in cargo container terminals to handle the internal movement of containers (see for example [9]). Cargo container terminals often hold thousands of containers at a time and the daily throughput may exceed 100–500 containers (see [21,22] for details). Also, space limitations mean that containers are piled up in *stacks, heaps or columns*.

Operational constraints often require that containers be internally moved within the container yard from one location to another. Cranes are specialized devices that can fetch a specific container from a stack, move (transport) it from its initial pickup location to its intended delivery location and finally place the container at the intended delivery location. Consider the optimization problem of efficiently moving a large number of containers using a limited fleet of cranes with a fixed time horizon. Superficially, the problems seem to be a pickup-and-delivery problem where immediate-precedence sequence-dependence is evident. But consider the situation shown in Fig. 1. The top half of Fig. 1 depicts the initial situation and the desired movement of containers. The lower part shows the final (intended) configuration. The containers are proposed to be handled in the ordered sequence $4 \rightarrow 1 \rightarrow 2 \rightarrow 3$. We neglect the immediate precedence costs induced by the physical movement of the cranes on the ground between locations for now. However, when any crane attempts to fetch container 1 from location B, it finds containers 4, 3, and 2 stacked over container 1. So, fetching container 1 needs stack rearrangement for three containers above it. Similarly, fetching containers 2 and 3 later will require stack rearrangements of more containers above the desired container every time. Instead, if the containers were being handled in the ordered sequence 3, 2, 1, and 4 then we would find that fetching any container does not call for any stack rearrangement for any other containers. For the example given here, the ideal ordered sequence 3, 2, 1, and 4 was obvious and

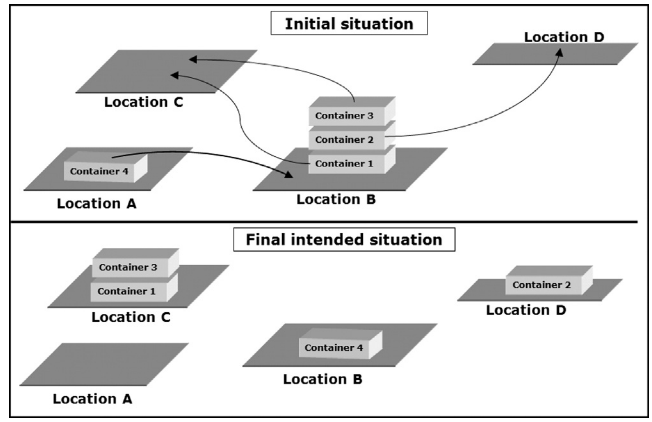

Fig. 1. Container handling at cargo terminals: cost of container handling depends strongly on overall history or non-immediate precedence.

trivial. But for several hundred containers with large stacking columns, the solution is often non-trivial.

If the container stack rearrangement costs are substantially larger than the costs of movement of cranes on the ground (between locations), then the overall optimization problem must stress upon minimizing the vertical stack rearrangement costs also. Now, these stack rearrangement costs do not depend on the immediate precedence sequence of containers being handled. Rather, it depends on the entire history of all containers handled by all cranes. This history determines the stack rearrangement cost of any subsequent container in the sequence(s). Thus, the problem of crane scheduling discussed here has features of immediate precedence sequence and also of general (non-immediate) precedence sequence. Finally, multiple cranes work simultaneously and their activities affect each others' schedules.

In addition, restrictions such as time window bounds also apply for specific containers. Consider that a container needs to be loaded onto a vehicle (say ship, train or truck) and dispatched. The movement of the concerned vehicle is often dictated by external factors (tidal patterns for ships, time-tables for trains, etc.). The assigned cranes must necessarily deliver this container at the appropriate location near the concerned vehicle, before the departure time of the vehicle. The converse is also true – cranes can only start handling a container after it has been brought in by some vehicle. So, some containers can only be picked up "after" the arrival time of specific vehicles. Effectively, the movement of any container is constrained by strict time windows.

Thus, the general crane scheduling problem with time windows, sequence dependent set-up (movement) times general precedence constraints (for stack rearrangement) justifies our motivation for GPSP. The GPSP in this context has a combination of multiple different precedence costs and timing related features within a single problem.

Apart from our primary motivation behind this study (that of scheduling of crane operations at ports), we notice that such problems (involving multiple aspects of sequencing in conjunction with time bounds) are encountered in many other contexts.

Scheduling of maintenance operations for trains requires that the maintenance crews incur immediate precedence dependent costs of moving between different operations. In addition, the history of operations already handled by other crews affects the amount of time needed by a specific crew while accomplishing its assigned tasks. If every team of personnel (crew) is considered as a separate resource and every task as an independent job to be scheduled, then the overall train maintenance scheduling problem maps to the GPSP.