Evaluation of Scheduling Policies for the SRCPSP in a Dynamic Multi-Project Environment

Hendrik Weber and Rainer Kolisch

Technical University of Munich, Germany
hendrik.weber, rainer.kolisch@tum.de

Keywords: Stochastic scheduling, dynamic multi-project scheduling, genetic algorithm, generalized preprocessor policies.

1 Introduction

We study the dynamic stochastic resource-constrained multi-project scheduling problem where projects arrive stochastically over time. Each project has a deterministic network and deterministic resource demand of activities, however, activity durations are stochastic. Resource availabilities provided for processing all projects are deterministic. The objective is to minimize average weighted flow times. We propose a new solution approach which, for each interarrival time of projects, calculates a scheduling policy and executes the latter until a new project arrives. The generation of scheduling policies is based on the stochastic resource-constrained (single) project scheduling problem (SRCPSP). In a computational study we assess several policy approaches which have been proposed for SRCPSP. The remainder of the paper is organized as follows: In Section 2 we review the relevant literature. In Section 3 we detail our solution approach and in Section 4 we provide information about our computational study.

2 Literature Review

The relevant literature for our study stems from two main streams of research, namely the stochastic resource-constrained project scheduling problem and the dynamic stochastic multi-project scheduling problem.

A number of contributions have been presented for the stochastic resource-constrained project scheduling problem (SRCPSP). Stork (2001) develops exact solution procedures to solve the SRCPSP by employing preselective, linear preselective, activity-based and earliest start policies in a branch-and-bound framework. Another exact model is presented by Creemers (2015): Under the assumption of PH-distributed activity durations, Creemers develops a model that employs a backward stochastic dynamic-programming recursion solution procedure.

Golenko-Ginzburg and Gonik (1997) present a heuristic solution approach that solves a 0-1 integer programming model at each decision point in order to determine which activity to process next. Tsai and Gemmill (1998) employ Tabu Search as well as Simulated Annealing to schedule activities in an SRCPSP-setting. Ballestín (2007) employs activity-based priority policies in combination with sampling procedures and a genetic algorithm to generate precedence-feasible activity lists. A similar approach based on activity-based priority policies is adapted by Ballestín and Leus (2009) who present a greedy randomized adaptive search procedure (GRASP). A novel solution procedure is presented by Ashtiani *et al.* (2011) who propose a new class of preprocessor policies encompassing resource-based and earliest start policies by heuristically inserting new precedence constraints of the type finish-to-start into the project network. This concept of preprocessor policies is further developed by Rostami *et al.* (2018). By introducing additional start-to-start constraints, the authors propose a new class of so-called generalized preprocessor policies.

Only limited work is available for the dynamic stochastic multi-project scheduling problem. Adler *et al.* (1995) were the first to extend the single stochastic project scheduling problems to a dynamic multi-project setting. Employing the real-life example of a product development organization, they develop an empirically-based framework for analysing development time in such a context. Choi *et al.* (2007) model this dynamic stochastic multi-project scheduling problem as a Markov decision process and employ a Q-learning based approach to heuristically determine policies for starting activities. Melchiors and Kolisch (2009) proceed likewise, however solve the Markov decision process by value iteration. Extensive computational studies are provided by Melchiors (2015), in which the author evaluates different priority rules in the dynamic multi-project setting. Fliedner (2015) evaluates sampling procedures as well as the genetic algorithm proposed by Hartmann (1998) in an experimental setup.

Our work extends the current body of literature by examining the applicability of the most recently proposed solution procedures to the SRCPSP in the context of stochastic and dynamic multi-project scheduling.

3 Proposed Solution Procedure

In comparison to proactive and reactive scheduling, which both rely on the determination of a baseline schedule to address uncertainty, we focus our research on stochastic scheduling where no initial schedule is determined. Following this approach, scheduling decisions are made "online" and solutions take on the form of policies that only utilize a-priori knowledge of activity duration distributions as well as the information that is available at the corresponding decision point in time when an activity is completed. Regarding this non-anticipativity constraint, these scheduling policies gradually build a schedule during the project's execution as actual realizations of uncertain activity durations unfold. This is achieved via the combination of a heuristically predetermined ordered activity list and a schedule generation scheme that is applied in the dynamic multi-project stochastic setting.

The proposed solution procedure employs an activity-on-node representation of the project network: Whenever a new project enters the system, the network of the arriving project as well as the networks of the projects still in the system are combined to a super-network which consists of all project activities, which still have to be processed or are being processed. For activities currently being in process, distributions of activity durations will be updated based on the so far observed duration. Until the arrival of the next project, the super-network can be viewed as multi-project SRCPSP instances to which we can apply SRCPSP solution methods. However, as even the SRCPSP is known to be NP-hard, we focus our efforts solely on heuristic procedures and resort to a discrete-event simulation approach. With the objective of minimizing average weighted flow times of projects, we evaluate the policies against the lower bound derived by the critical path length of the deterministic equivalent of the project.

3.1 Experimental Study

We evaluate four of the SRCPSP-solution procedures outlined above, namely the regret-based biased random sampling method and the genetic algorithm of Fliedner (2015) as well as preprocessor policies proposed by Ashtiani *et al.* (2011) and generalized preprocessor policies suggested by Rostami *et al.* (2018). From a theoretic standpoint, preprocessor policies should, as a superset of the earliest start and resource-based policy classes, strictly dominate the class of resource-based policies. Analogously, the class of generalized preprocessor policies should strictly dominate all other policy classes as it entails all resource-based, earliest start and activity-based policies. In order to account for this, we restrict the allotted computation time for each policy generation by limiting the number of generated schedules. Also, in contrast to the proposed class of (generalized) preprocessor policies, we do not evaluate every possible additional predecessor constraint. To limit the computational effort, we only consider additional predecessor constraints that concern activities whose expected start or finish times lie within a small time window starting from the current decision point: As new projects arrive and the solution procedure is repeated, more informed decisions regarding the inclusion of additional predecessor constraints can then be made at a later point in the simulation.

The setup for the proposed experimental study is based on Fliedner (2015). The selection of two uniform distribution (U1, U2), two beta distributions (B1, B2) and an exponential distribution (EXP) for activity durations is in line with Ballestín and Leus (2009). We further assume known true means of the activity duration distributions.

Using the ProGen/max generator by Schwindt (1995) we generate 120 different RCPSP problem instances where each project consists of |N|= 15 non-dummy activities and |K|= 4 different renewable resources. The average number of required resources is controlled by the resource factor and set to either 0.25, 0.5, 0.75 or 1. The arrival of new projects follows a Poisson process with arrival rate λ. In order to evaluate several levels of resource scarcity, we adjust λ which then results in different average utilization levels $u \in 0.5, 0.7, 0.9$. Newly arriving projects are all of the same type.

After a steady number of projects in the system is reached, the flow times of 200 successive projects are averaged and compared to the critical path of the deterministic equivalent. We conduct a full-factorial experimental study with the parameters summarised in Table 1. Simulation results will be presented.

Table 1. Level of problem parameters

Paramter	**Value**
|K|	4
RF	0.25, 0.5, 0.75, 1
u	0.5, 0.7, 0.9
Distribution	U1, U2, B1, B2, EXP

References

Adler P., Mandelbaum A., Nguyen V. and Schwerer E., 1995, "From project to process management: An empirically-based framework for analyzing product development time", *Management Science*, Vol. 41, pp. 458-484.

Ashtiani B., Leus R. and Aryanezhad M., 2011, "New competitive results for the stochastic resource-constrained project scheduling problem: exploring the benefits of pre-processing", *Journal of Scheduling*, Vol. 14, pp. 157-171.

Ballestín F., 2007, "When it is worthwhile to work with the stochastic RCPSP?", *Journal of Scheduling*, Vol. 10, pp. 153-166.

Ballestín F., Leus R., 2009, "Resource–constrained project scheduling for timely project completion with stochastic activity durations", *Production and Operations Management*, Vol. 18, pp. 459-474.

Choi J., Realff M. and Lee J., 2007, “A Q–Learning–based method applied to stochastic resource constrained project scheduling with new project arrivals", *International Journal of Robust and Nonlinear Control: IFAC–Affiliated Journal*, Vol. 17, pp. 1214-1231.

Creemers S., 2015, “Minimizing the expected makespan of a project with stochastic activity durations under resource constraints", *Journal of Scheduling*, Vol. 18, pp. 263-273.

Fliedner T., 2015, “Considering Uncertainty in Project Management and Scheduling", PhD-Thesis. Munich.

Golenko-Ginzburg D., Gonik A., 1997, “Stochastic network project scheduling with non-consumable limited resources", *International Journal of Production Economics*, Vol. 48, pp. 29-37.

Hartmann S., 1998, “A competitive genetic algorithm for resource–constrained project scheduling", *Naval Research Logistics (NRL)*, Vol. 45, pp. 733-750.

Melchiors P., Kolisch R., 2009, “Scheduling of Multiple R&D Projects in a Dynamic and Stochastic Environment", *Operations Research Proceedings 2008* Springer. Heidelberg.

Melchiors P., 2015, “Dynamic and stochastic multi-project planning", PhD-Thesis. Springer. Heidelberg.

Möhring R., Stork F., 2000, “Linear preselective policies for stochastic project scheduling", *Mathematical Methods of Operations Research*, Vol. 52, pp. 501-515.

Rostami S., Creemers S. and Leus R., 2018, “New strategies for stochastic resource-constrained project scheduling", *Journal of Scheduling*, Vol. 21, pp. 394-365.

Schwindt C., 1995, “ProGen/max: A new problem generator for different resource-constrained project scheduling problems with minimal and maximal time lags", *Technical report. Institut für Wirtschaftstheorie und Operations Research, Universität Karlsruhe.*, Vol. 449, pp. 1-57.

Stork F., 2001, “Stochastic resource-constrained project scheduling", PhD-Thesis. Berlin.

Tsai Y., Gemmill D., 1998, “Using tabu search to schedule activities of stochastic resource-constrained projects", *European Journal of Operational Research*, Vol. 111, pp. 129-141.