A Sampling Based Approach for Proactive Project Scheduling with Time-Dependent Duration Uncertainty

Wen Song, Donghun Kang, Jie Zhang[a], Hui Xi[b]
Rolls-Royce@NTU Corp Lab, Nanyang Technological University, Singapore
[a]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[b]Rolls-Royce Singapore Pte Ltd, Singapore
{songwen, donghun.kang, [a]zhangj}@ntu.edu.sg, [b]hui.xi@rolls-royce.com

Abstract

Most of the existing proactive scheduling approaches assume the durations of activities can be described by independent random variables that have no relation with time. We deal with the more challenging problem where the duration uncertainty is related to the scheduled time period. We propose a sampling based approach by extending the Consensus method from stochastic optimization. Experimental results show the effectiveness of our approach in solution quality and stability.

Introduction

Project scheduling is a common problem for business management. In most of the real-world environments, durations of activities are highly uncertain. Though it is possible to conduct online scheduling during execution, a *proactive schedule* generated before execution by exploiting stochastic knowledge of the duration uncertainty is of great value in providing visibility for coordination of the execution process (Van de Vonder, Demeulemeester, and Herroelen 2007). Many approaches have been proposed for proactive project scheduling, e.g. (Varakantham, Fu, and Lau 2016). These approaches assume that activity durations can be modeled as independent random variables that are not affected by their scheduled time periods. However, this may not hold under many circumstances. For example, suppose a product test project involving a set of outdoor testing activities that are sensitive to weather conditions (e.g. temperature, humidity, wind speed). Due to seasonality, activity durations could be affected differently based on their scheduled time periods.

In this work, we study the more challenging problem of proactive project scheduling with time-dependent duration uncertainty. Our approach is based on the Consensus method (Hentenryck and Bent 2009) from stochastic optimization. It first samples a set of scenarios from the probability distributions, then solves these scenarios as deterministic problems, and finally makes scheduling decisions according to a Consensus vote process. Different from its previous application in online scheduling, here we adopt Consensus to construct an offline schedule. After sampling scenarios, our approach incrementally constructs the schedule through an iterative solving and voting process.

Problem Description

A static Resource-Constrained Project Scheduling Problem (RCPSP) consists of a project with a set of non-preemptive activities $A = \{a_1, ..., a_n\}$, and a set of renewable resources $R = \{r_1, ..., r_m\}$. Each $a_i \in A$ has a duration of d_i time slots (days), and requires b_{ik} amount of $r_k \in R$ with a limited capacity of c_k units per time slot. A pair of activities could have a precedence relation $a_i \prec a_j$, indicating a_i must complete before the start of a_j. A schedule $S = [s_1, ..., s_n]$ is a vector where s_i is the start time of a_i. A feasible S must satisfy all the resource and precedence constraints. A feasible schedule S^* is optimal if it minimizes the makespan $MS(S) = max_j\{c_i\}$, where $c_i = s_i + d_i$ is the completion time of a_i. When uncertainty is considered, duration of a_i is not fixed but subject to the weather conditions of its scheduled interval $[s_i, c_i]$. Nevertheless, a_i must obtain at least d_i workable days in $[s_i, c_i]$. Here we associate each a_i with a type $z_i \in \{1, ..., Z\}$, and different types of activities could be affected differently by the same weather condition. Let p_t^z be the Probability of Workability (POW) of day t for activity type z. We assume the scheduling horizon $\{1, ..., T\}$ is split into a set of periods $\{E_1, ..., E_l\}$, where $E_g = [e_g, f_g]$ is a consecutive time period (e.g. season, month, week) starts at e_g and ends at f_g. We further assume for an activity type z, all days in a period have the same POW.

In general, a proactive schedule S can be evaluated from two aspects: quality and stability (Van de Vonder, Demeulemeester, and Herroelen 2007). Let $\widehat{S}$ be the final schedule after executing S in an actual scenario, and $\widehat{S}^*$ be the optimal schedule assuming to have full knowledge of that scenario. $\widehat{S}$ is obtained with a reactive strategy to handle schedule disruptions (here we adopt the commonly used early start policy with fixed resource allocation (Van de Vonder, Demeulemeester, and Herroelen 2007)), while $\widehat{S}^*$ can be obtained by solving an offline problem (here we solve it using Integer Linear Programming, ILP). The quality of S is evaluated using the makespan deviation of $\widehat{S}$ from $\widehat{S}^*$, i.e. $dev(S) = (MS(\widehat{S}) - MS(\widehat{S}^*))/MS(\widehat{S}^*)$. The stability of S is evaluated using a stability cost function $sc(S) = \sum_i |s_i - \widehat{s}_i|$.

Methodology

The first step of our approach is to sample a set of scenarios $\boldsymbol{Q} = \{Q_1, ..., Q_N\}$. A scenario Q is a matrix $[q_{zt}]_{Z\times T}$,

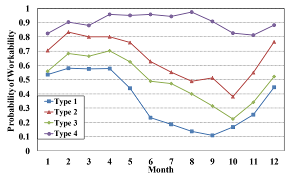

Figure 1: Monthly POW data

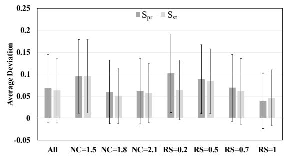

Figure 2: Average deviation

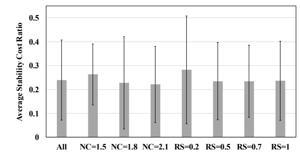

Figure 3: Average stability cost ratio

where $q_{zt} \in \{0, 1\}$ indicates whether day t is workable for activity type z. Then, a series of Consensus processes are conducted for each period E_g to schedule a set of activities $A_g \subseteq A$ to start within E_g, until all activities in A are scheduled. Each Consensus process consists of two consecutive stages, solving and voting.

Solving. In this stage, a schedule S_λ is generated for each $Q_\lambda \in \boldsymbol{Q}$ by solving a deterministic problem where the usability of each day is known. These problems have the same constraints as the static problem, but the duration of a_i is no longer a fixed parameter d_i; instead a duration constraint $\sum_{t=s_i}^{c_i} q_{z_i t} \geq d_i$ is added to guarantee that a_i receives enough processing days during its scheduled interval. Consequently, these problems extend the static problem (NP-hard) by adding a set of decision variables c_i and duration constraints. Here we design a polynomial-time algorithm based on the parallel generation scheme from static project scheduling. This algorithm first selects an unscheduled activity a_i using the latest finish time rule, then sets s_i^λ as the earliest resource and precedence feasible time, and sets c_i^λ as the earliest completion time that satisfies the duration constraints of a_i. Complexity of this algorithm is $\mathcal{O}(n^2 mT)$.

Voting. In this stage, the unscheduled activities are first ranked in the descending order of their votes. The vote of a_i is the number of S_λ where it is scheduled to start within the current period E_g. Activities with zero vote are not considered in the ranking. Then, the algorithm tries to schedule the eligible activities according to the vote ranking, until no activity is eligible or can be scheduled to start within E_g. An activity is eligible if all its predecessors have been scheduled. When an activity a_i is chosen to be scheduled, s_i is set to be the earliest resource and precedence feasible time. To determine c_i, we first estimate the duration of a_i using an average value $\overline{d}_i = \sum_{Q_\lambda \in \boldsymbol{Q}(a_i)} (c_i^\lambda - s_i^\lambda)/|\boldsymbol{Q}(a_i)|$, where $\boldsymbol{Q}(a_i)$ is the set of scenarios that vote for a_i, then set the completion time to be $c_i = s_i + \overline{d}_i$.

The above algorithm runs in polynomial time, with a complexity of $\mathcal{O}(lN \times n^2 mT)$.

Preliminary Results

We use the J30 problem set from PSPLIB (http://www.om-db.wi.tum.de/psplib/) for evaluation, which contains 480 project instances. The POW data used here is collected from a real-world aero engine test project. As shown in Figure 1, this data set contains POWs of each month for four activity types. We conduct 1000 tests to evaluate our approach. Each test consists of a randomly selected J30 instance where each activity is randomly assigned a type $z \in \{1, ..., 4\}$, and a scenario Q is sampled as the actual scenario. For each test, we run our approach with ($N = 50$) [1] to get the proactive schedule S_{pr}. We compare the makespan deviation and stability cost of S_{pr} and S_{st}, the static optimal schedule obtained by solving an ILP of the static problem. Figures 2 and 3 show the average deviation of S_{pr} and S_{st} and the average stability cost ratio $sc(S_{pr})/sc(S_{st})$ of all the 1000 tests, along with the corresponding values of the groups categorized based on two parameters of J30, Network Complexity (NC, higher value indicates more precedence relations) and Resource Strength (RS, higher value indicates tighter resource constraints). We can observe that, the qualities of both S_{pr} and S_{st} are comparable and close to the offline optimal solution (within 10% on average). However, S_{pr} is generated in polynomial time, while S_{st} requires exponential time computation. Both approaches tend to perform better on tighter constrained instances with higher NC and RS, since the deviation tends to decrease. On the other hand, the stability cost of S_{pr} is much smaller than S_{st} (improvement is 78% on average), and the improvement tends to increase on tighter constrained instances with higher NC and RS.

Future Work

The current approach relies on the assumption of horizon splitting. In the future, we plan to develop an approach based on the Sample Average Approximation method (Kleywegt, Shapiro, and Homem-de Mello 2002), which supports general probability models and has good theoretical properties in approximately solving complex stochastic optimization problems through sampling.

Acknowledgments. This work was supported by the National Research Foundation (NRF) Singapore.

References

Hentenryck, P. V., and Bent, R. 2009. *Online stochastic combinatorial optimization*. The MIT Press.

Kleywegt, A. J.; Shapiro, A.; and Homem-de Mello, T. 2002. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization* 12(2):479–502.

Van de Vonder, S.; Demeulemeester, E.; and Herroelen, W. 2007. A classification of predictive-reactive project scheduling procedures. *Journal of Scheduling* 10(3):195–207.

Varakantham, P.; Fu, N.; and Lau, H. C. 2016. A proactive sampling approach to project scheduling under uncertainty. In *AAAI*.

[1]The parameter testing experiments are omitted here for brevity.