Accepted Manuscript

Scheduling Markovian PERT networks to maximize the net present value: New results

Ben Hermans, Roel Leus

PII: S0167-6377(17)30422-4
DOI: https://doi.org/10.1016/j.orl.2018.01.010
Reference: OPERES 6333

To appear in: *Operations Research Letters*

Received date : 26 July 2017
Revised date : 26 January 2018
Accepted date : 26 January 2018

Scheduling Markovian PERT networks to maximize the net present value: New results

Ben Hermans[a], Roel Leus[a,*]

[a]*Research Center for Operations Research & Business Statistics (ORSTAT), Faculty of Economics and Business, KU Leuven, Belgium*

Abstract

We study project scheduling so as to maximize the expected net present value when task durations are exponentially distributed. Based on the structural properties of an optimal solution we show that, even if preemption is allowed, it is not necessary to do so. Next to its managerial importance, this result also allows for a new algorithm which improves on the current state of the art with several orders of magnitude, both in CPU time and in memory usage.

Keywords: project scheduling, net present value, exponentially distributed activity durations, Markov decision process, monotone optimal policy

1. Introduction

Consider a project in which a set of tasks $N = \{1, \ldots, n\}$ needs to be performed in order to reach a given goal such as, for example, the development of a new product. The strict partial order $A \subset N \times N$ defines precedence constraints, with $(i, j) \in A$ indicating that task j can only start if i is finished. We suppose task n reflects the project's completion and $(i, n) \in A$ for all $i \in N \setminus \{n\}$. Each task $i \in N$ has a random duration $\tilde{d}_i$ with support $\mathbb{R}_{\geq 0}$ and, initially, we require activities to be processed without interruption. Finally, each task has a cash flow which is discounted to take into account the time value of money.

In this article, we study the problem of deciding when to start each task so as to maximize the project's *expected net present value* (eNPV), i.e., the expected sum of discounted cash flows. Throughout, we assume independent and exponentially distributed activity durations $(\tilde{d}_i)_{i\in N}$ with rate parameter $(\lambda_i)_{i\in N} \in \mathbb{R}^n_{>0}$. This problem was also studied by [1, 2, 9]; we briefly review these articles below. For an excellent and more detailed literature review, also for the case where task durations are not exponentially distributed, see [11].

All models in [1, 2, 9] are based on the seminal work of Kulkarni and Adlakha [6], who use a continuous-time Markov chain to evaluate the moments and distribution of the project's earliest completion time. Buss and Rosenblatt [1] adapt this Markov chain to evaluate the eNPV when each task is initiated as soon as possible. Next, they determine the optimal delay for up to two activities beyond their earliest possible starting time. While the delays in [1] are fixed before the project's start, Sobel et al. [9] consider the more general case of making scheduling decisions adaptively during the project's execution, solving the problem using a stochastic dynamic program (SDP). To mitigate the SDP's excessive memory usage, Creemers et al. [2] partition the state space such that not all states have to be stored in memory at the same time. This significantly improves performance and their algorithm is considered to be the current state of the art [11].

We address the same problem as [2], which we define in Section 2 and for which we give a new SDP-formulation in Section 3. The major difference is that we act *as if* activities can be interrupted, but show that, even if preemption is allowed, it is not necessary to do so (Section 4). Consequently, the preemptive case solves the non-preemptive case as well. Next to its managerial importance, this result also allows for a new algorithm that improves on

*Corresponding author. Faculty of Economics and Business, KU Leuven, Naamsestraat 69, 3000 Leuven, Belgium.
Email address: `roel.leus@kuleuven.be` (Roel Leus)