KU LEUVEN

FACULTY OF ECONOMICS AND BUSINESS

Two sequencing problems: equivalence, optimal solution, and state-of-the-art results

Creemers S.

KBI_1720

Two sequencing problems: Equivalence, optimal solution, and state-of-the-art results

Stefan Creemers
IESEG School of Management, Rue de la digue 3, 59000 Lille, France
KU Leuven, Research Center for Operations Management, Naamsestraat 69, 3000 Leuven, Belgium
s.creemers@ieseg.fr

We show that the serial SNPV and the LCFDP are equivalent. The serial SNPV is a special case of the SNPV that tries maximize the expected NPV of a project by sequencing activities that have stochastic durations and cash flows that are incurred at the start of an activity. The LCFDP, on the other hand, minimizes the expected cost of the sequential diagnosis of a set of tests that have known execution costs and failure probabilities.

Keywords: Least cost fault detection problem; sequential testing problem; stochastic net present value maximization; project scheduling

1 Introduction

The stochastic net present value maximization problem (SNPV) tries to maximize the expected NPV (eNPV) of a project with n activities that have stochastic durations. Each activity $i : i \in \mathbf{N} = \{1, \ldots, n\}$ has a duration distribution function f_i, and a cash flow $c_i \in \mathbb{R}$ is incurred at the start of activity i. Upon completion of the project, a payoff c_p is obtained. A solution to the SNPV is a policy that schedules activities such that the eNPV of the project (i.e., the expected sum of the discounted cash flows that are incurred during the lifetime of the project) is maximized. The SNPV has been considered by, among others, Sobel et al. (2009), Creemers et al. (2010), and Wiesemann et el. (2010). The literature on the SNPV has been reviewed by Wiesemann and Kuhn (2015), who not only stress the importance of stochastic project scheduling (over deterministic scheduling), but who also argue that NPV is a more important objective than project makespan.

The LCFDP is a variant of the sequential testing problem (STP) where n precedence-related tests have to be scheduled such that the expected cost of the diagnosis of a system is minimized. Each test $i : i \in \mathbf{N} = \{1, \ldots, n\}$ has a known cost c_i and a failure probability p_i. In this article, we consider the setting where a single test results in the failure of the system (i.e., we study so-called n-out-of-n or serial systems). For such a setting, it can be shown that there exists a full order sequence of tests in $\mathbf{N}$ that is globally optimal. The LCFDP is related to the R&D project scheduling problem studied in De Reyck and Leus (2008), who show that their problem is NP-hard. It follows that the LCFDP is also NP-hard if tests are precedence-related (Wei et al., 2013). The LCFDP arises in many practical contexts, such as the inspection of containers arriving at a port (Madigan et al., 2011) and the identification of toxic chemicals (Gowtham et al., 2012). A literature review on the STP in general, and on the LCFDP in particular, may be found with Ünlüyurt (2004), Wei et al. (2013), and Coolen et al. (2014).

The serial SNPV is a special case of the SNPV where activities have to be scheduled in series. The serial SNPV arises in contexts where there is a bottleneck resource, or in settings where projects have a serial structure. A solution to the serial SNPV is a sequence of activities that maximizes the eNPV of the project (over all possible sequences). In what follows, we show that: (1) the LCFDP is equivalent to the serial SNPV, (2) a well-known result from the literature on the LCFDP may be used to obtain the optimal solution to the serial SNPV if activities are not precedence related, and (3) methods for solving the SNPV can also be used to solve the LCFDP. In addition, we perform a computational experiment that shows that the state-of-the-art procedure for solving the SNPV (a more general problem) outperforms the state-of-the-art procedure for solving the LCFDP.

2 Equivalence of the serial SNPV and the LCFDP

Let $\mathbf{s} = \{s_1, \ldots, s_n\}$ denote a sequence of n activities, where s_i is the activity at position i in the sequence. As shown by Creemers (2016), the eNPV of a sequence $\mathbf{s}$ is given by

$$c_{s_1} + \sum_{i=2}^{n} \phi_{1,(i-1)}(r) c_{s_i},$$

where r is the discount rate, and $\phi_{1,i}$ is the discount factor for a sequence of activities $\{s_1, \ldots, s_i\}$. $\phi_{1,i}(r)$ is obtained as follows

$$\phi_{1,i}(r) = \prod_{j=1}^{i} \phi_{s_j}(r),$$

where $\phi_i(r) \equiv \phi_{i,i}(r)$ is the discount factor that applies for a single activity i. $\phi_i(r)$ is given by

$$\phi_i(r) = M_i(-r),$$

where $M_i(-r)$ is the moment-generating function of the duration distribution function f_i about $-r$.

The objective of the serial SNPV is to find a sequence that maximizes

$$c_{s_1} + \left(\sum_{i=2}^{n} c_{s_i} \prod_{j=1}^{i-1} \phi_{s_j}(r) \right) + \left(c_p \prod_{i=1}^{n} \phi_i(r) \right),$$

where the latter term is a constant that does not depend on the sequence of activities (i.e., the latter term may be ignored when making sequencing decisions), and hence the objective reduces to

$$\max_{\mathbf{s}} c_{s_1} + \left(\sum_{i=2}^{n} c_{s_i} \prod_{j=1}^{i-1} \phi_{s_j}(r) \right). \tag{1}$$

The objective of the LCFDP, on the other hand, is to find a sequence of tests that minimizes the cost of the sequential diagnosis of a system, and is given by

$$\max_{\mathbf{s}} c_{s_1} + \left(\sum_{i=2}^{n} c_{s_i} \prod_{j=1}^{i-1} q_{s_j} \right), \tag{2}$$

where $q_i = 1 - p_i$ is the success probability of test i. Eq. (1) and Eq. (2) are equivalent if $\phi_i(r) \equiv q_i$ for all $i \in \mathbf{N}$. We conclude that the LCFDP is equivalent to the serial SNPV, which in turn is a special case of the SNPV.

3 Optimal sequence

In the absence of precedence relationships, Boothroyd (1960) has shown that the optimal solution to the LCFDP is a sequence that arranges tests in (increasing) order of their ratio of cost over failure probability. Therefore, for the LCFDP, the optimal sequence can be determined in polynomial time, and if

$$\frac{c_{s_1}}{p_{s_1}} \leq \frac{c_{s_2}}{p_{s_2}} \leq \ldots \leq \frac{c_{s_n}}{p_{s_n}},$$

then $\mathbf{s} = \{s_1, s_2, \ldots, s_n\}$ is optimal.

The above result can also be used to determine the optimal sequence that maximizes the eNPV of a project where activities are not precedence related. More precisely, in the absence of precedence relationships, sequence $\mathbf{s} = \{s_1, s_2, \ldots, s_n\}$ is optimal if

$$\frac{c_{s_1}}{1 - \phi_{s_1}(r)} \leq \frac{c_{s_2}}{1 - \phi_{s_2}(r)} \leq \ldots \leq \frac{c_{s_n}}{1 - \phi_{s_n}(r)}.$$

To illustrate this finding, we use an example project with 5 activities that have exponentially-distributed durations with rate parameter $\lambda_i : i \in \mathbf{N} = \{1, 2, 3, 4, 5\}$. The data of the example project are summarized in Table 1. Note that ϕ_i is the moment-generating function of f_i about $-r$, and is given by

$$\phi_i(r) = \frac{\lambda_i}{\lambda_i + r} \tag{3}$$

for an exponentially-distributed duration with rate parameter λ_i. The optimal sequence executes activities 4, 2, 3, 5, and 1 in series, and yields an eNPV of 15.22.

4 Solving the LCFDP

In this section, we solve the LCFDP using the state-of-the-art procedure of Creemers (2017) that was designed to solve the SNPV. Creemers (2017) assumes that activity durations are exponentially distributed, and uses a continuous-time Markov chain (CTMC) to model the state space. A backward stochastic dynamic program (SDP) is used to obtain the globally optimal policy that maximizes the eNPV of a project (note that a solution to the SNPV is a scheduling policy rather

i	c_i	λ_i	ϕ_i	$c_i(1-\phi_i)^{-1}$	s_i
1	-10	$1/2$	0.833	-60	5
2	10	$1/4$	0.714	35	2
3	-15	$1/6$	0.625	-40	3
4	20	$1/8$	0.556	45	1
5	-36	$1/20$	0.250	-48	4

p	100
r	0.1

Table 1: Data of the example project

than a sequence). In contrast to most of the literature on the scheduling of Markovian PERT networks (i.e., PERT networks where activities have exponentially-distributed durations), Creemers (2017) does not use uniformly directed cuts (UDCs) to structure the state space, nor does he represent the state of the system using sets of idle, ongoing, and finished activities (see e.g., Creemers et al., 2010). Instead, Creemers (2017) uses arrays to store states that are defined only by the set of finished activities. The cardinality of a state (i.e., the number of finished activities) determines the array in which the state is stored (there is one array for each number of finished activities). Because states with cardinality $(i+1)$ are only accessible from states with cardinality i, at most two arrays need to be stored in memory (i.e., after calculating all value functions of states with cardinality i, states with cardinality $(i+1)$ are no longer needed, and they can be removed from memory). Together with a stricter definition of the state space (by only using the set of finished activities), this more efficient structuring of the state space results in a significant reduction of memory and computational requirements (when compared to other methods that solve the SNPV).

In order to solve an instance of the LCFDP by means of a procedure for solving the SNPV, tests first need to be "transformed" into activities. As explained in Section 2, the serial SNPV and the LCFDP are equivalent if $\phi_i(r) \equiv q_i$ for all $i : i \in \mathbf{N}$. In the procedure of Creemers (2017), activities are assumed to have exponentially-distributed durations, and therefore, the discount factor of an activity i is given by Eq. (3). As such, a test i with cost c_i and failure probability p_i can be transformed into an activity i with cost c_i and rate parameter

$$\lambda_i = \frac{p_i r}{1-p_i}.$$

After transforming all tests into activities, the procedure of Creemers (2017) can be used to solve an instance of the LCFDP. Note that, in order to make sure that activities are executed in a sequence, we impose impose a resource constraint (i.e., each activity requires one unit of a renewable resource that has unit availability).

We compare the performance of the above approach with the state-of-the-art procedure of Wei et al. (2013). Wei et al. (2013) propose both a branch-and-bound (B&B) as well as an SDP procedure to solve the k-out-of-n STP (i.e., at least k out of n components should be functional, otherwise the system is down). The SDP procedure significantly outperforms the B&B, and in what follows, we will compare its performance with that of the procedure of Creemers (2017). Note that, if we let $k = n$, the k-out-of-n STP corresponds to the LCFDP as defined by Boothroyd (1960). Similar to the procedures of Creemers et al. (2010) and Coolen et al. (2014), The SDP procedure of Wei et al. (2013) uses UDCs to structure the state space. Once the states of a UDC are no longer required, the UDC is discarded, and its memory is freed.

We use the instances of Wei et al. (2013) to compare the performance of both SDP procedures. Wei et al. (2013) use RanGen (Demeulemeester et al., 2003) to generate three data sets that each contain 10 instances for each value of $n : n \in \{10, 20, \ldots, 120\}$ and for each value of OS : OS $\in \{0.4, 0.6, 0.8\}$ (where OS is the order strength; a measure of the density of the project network). Each data set has different failure probabilities. Because failure probabilities do not impact the computational performance of the SDP procedures, we select the data set that has the lowest failure probabilities (i.e., failure probabilities are drawn from a uniform distribution with a minimum of 0.8 and a maximum of 1). Both procedures are tested on an Intel 3.3 GHz desktop computer with 16GB of RAM.

Table 2 reports on the number of instances solved by each approach, and shows that the procedure of Creemers (2017) outperforms the procedure of Wei et al. (2013). This can be explained by the more efficient memory-management techniques adopted by Creemers (2017). When comparing average computation times (in seconds) on instances that could be solved by Wei et al., however, Table 3 shows that the procedure of Creemers (2017) is somewhat slower (26.5% on average). Since the procedure of Creemers (2017) was designed to solve the SNPV (a more general problem), this does not come as a surprise. In addition, it is clear that memory requirements rather than CPU times are the bottleneck for the problem at hand. Even for larger problems, Table 4 shows that the procedure of Creemers (2017) is able to solve instances within a reasonable time frame.

5 Conclusions

In this article, we study two problems. On the one hand, we have the serial SNPV (a special case of the SNPV) that tries to sequence activities with stochastic durations as to maximize the eNPV of a project (i.e., the expected sum of the discounted cash flows that are incurred during the lifetime of a project). The LCFDP, on the other hand, tries to determine the sequence of tests that minimizes the expected cost of the diagnosis of a system. In contrast to the (serial) SNPV, the LCFDP does not take into account (test) durations, nor does it discount cash flows that are incurred during diagnosis. Although the serial SNPV and the LCFDP appear to be unrelated, we show that they are in fact equivalent. In addition, in the absence of precedence relationships, we show that the optimal solution to the serial SNPV may be obtained by adopting the result of Boothroyd (1960), who finds that the optimal solution to the LCFDP can be found by arranging tests according to their ratio of cost over failure probability. Last but not least, we show that the LCFDP can be seen as a special case of the SNPV, and that procedures for solving the SNPV can also be used to solve

	Creemers (2017)			Wei et al. (2013)		
n	OS = 0.8	OS = 0.6	OS = 0.4	OS = 0.8	OS = 0.6	OS = 0.4
10	10	10	10	10	10	10
20	10	10	10	10	10	10
30	10	10	10	10	10	10
40	10	10	10	10	10	10
50	10	10	10	10	10	10
60	10	10	10	10	10	10
70	10	10	10	10	10	9
80	10	10	10	10	10	0
90	10	10	0	10	10	0
100	10	10	0	10	10	0
110	10	10	0	10	7	0
120	10	9	0	10	0	0

Table 2: Number of instances solved (out of 10) by the procedures of Creemers (2017) and Wei et al. (2013)

	Creemers (2017)			Wei et al. (2013)		
n	OS = 0.8	OS = 0.6	OS = 0.4	OS = 0.8	OS = 0.6	OS = 0.4
10	0	0	0	0	0	0
20	0	0	0	0	0	0
30	0	0	0.01	0	0	0.01
40	0	0.01	0.17	0	0.01	0.13
50	0	0.04	1.99	0	0.02	1.44
60	0.01	0.19	25.4	0	0.11	19.3
70	0.01	0.94	178	0.01	0.58	156
80	0.03	4.00	–	0.01	2.40	–
90	0.05	15.0	–	0.02	9.48	–
100	0.11	77.1	–	0.05	45	–
110	0.24	223	–	0.10	151	–
120	0.56	–	–	0.24	–	–

Table 3: Comparison of average computation time (in seconds) for the instances that could be solved by Wei et al. (2013)

n	OS = 0.8	OS = 0.6	OS = 0.4
10	0	0	0
20	0	0	0
30	0	0	0.01
40	0	0.01	0.17
50	0	0.04	1.99
60	0.01	0.19	25.4
70	0.01	0.94	205
80	0.03	4.00	2,013
90	0.05	15.0	–
100	0.11	77.1	–
110	0.24	323	–
120	0.56	1,009	–

Table 4: Average computation time (in seconds) for the procedure of Creemers (2017)

the LCFDP. In addition, a computational experiment shows that the state-of-the-art procedure for solving the SNPV (a more general problem) outperforms the state-of-the-art procedure for solving the LCFDP. Future research should therefore focus on developing better performing procedures for solving the LCFDP.

References

Boothroyd, H. (1960). Least-cost testing sequence. *Operational Research Society*, 11(3), 137–138.

Coolen, K., Wei, W., Talla Nobibon, F., & Leus, R. (2014). Scheduling modular projects on a bottleneck resource. *Journal of Scheduling*, 17(1), 67-85.

Creemers, S., Leus, R., & Lambrecht, M. (2010). Scheduling Markovian PERT networks to maximize the net present value. *Operations Research Letters*, 38(1), 51-56.

Creemers, S. (2016). *Moments and distribution of the NPV of a project*, Working paper KBI_1623. KU Leuven, Faculty of Economics and Business, Department of Decision Sciences and Information Management.

Creemers, S. (2017). *Maximizing the expected net present value of a project with phase-type distributed activity durations: an efficient globally optimal solution procedure*, Working paper KBI_1718. KU Leuven, Faculty of Economics and Business, Department of Decision Sciences and Information Management.

De Reyck, B., & Leus, R. (2008). R&D-project scheduling when activities may fail, *IIE Transactions*, 40(4), 367–384.

Demeulemeester, E., Vanhoucke, M., & Herroelen, W. (2003). A random network generator for activity-on-the-node networks. *Journal of Scheduling*, 6(1), 17-38.

Gowtham, B., Bhavnani, S.K., & Scott, C. (2011). group-based active query selection for rapid diagnosis in time-critical situations. *IEEE Transactions on Information Theory*, 58(1), 459–478.

Madigan, D., Mittal, D., & Roberts, F. (2011). Efficient sequential decision-making algorithms for container inspection operations. *Naval Research Logistics*, 58(7), 637–654.

Sobel, M.J., Szmerekovsky, J.G., & Tilson, V. (2009). Scheduling projects with stochastic activity duration to maximize expected net present value. *European Journal of Operational Research*, 198(1), 697-705.

Wei, W., Coolen, K., & Leus, R. (2013). Sequential testing policies for complex systems under precedence constraints. *Expert Systems Applications*, 40(2), 611-620.

Wiesemann, W., Kuhn, D., & Rustem, B. (2010). Maximizing the net present value of a project under uncertainty. *European Journal of Operational Research*, 202(2), 356-367.

Wiesemann, W., & Kuhn, D. (2015). The stochastic time-constrained net present value problem. In C. Schwindt, & J. Zimmermann (Eds.), *Handbook on Project Management and Scheduling Vol. 2* (pp. 753–780). New York: Springer.

FACULTY OF ECONOMICS AND BUSINESS
Naamsestraat 69 bus 3500
3000 LEUVEN, BELGIË
tel. + 32 16 32 66 12
fax + 32 16 32 67 91
info@econ.kuleuven.be
www.econ.kuleuven.be