Sampling Based Approximation Method for the Stochastic PERT Problem

01012560 Komazawa University IIDA Tetsuo*
01206492 Tohoku University SUZUKI Kenichi

1 Introduction

Traditionally PERT/CPM has been used to manage large projects. The idea of the critical path still plays an important part in the scheduling. However, most of existing researches and applications of PERT tackled the deterministic problem. On the other hand, uncertainty is the most significant aspects of the practical problem, since it can cause delay of the schedule, increase of the cost, and deterioration of the quality of outputs.

Here we formulate a stochastic PERT problem as a two stage stochastic linear programming model with recourse (SLP), since it can handle optimization problems under uncertainty. It is well known that SLP can be solved by the decomposition based algorithm [1]. The main difficulty on the computation arises from the number of states: it may increase exponentially, consuming a large amount of memories, even if problem contains a few random variables. Hence, combination of decomposition algorithm and Monte Carlo sampling technique are proposed by several authors (e.g. [3]).

In our previous work (JORSJ general meeting, Fall 2002), we showed the efficiency of sampling based algorithm for the stochastic PERT problem. Also, we suggested the formula of new density on the importance sampling method, by which we expect to reduce variance of the estimator. Unfortunately the computational experiment were still preliminary one. In our subsequent research, we conducted more thorough computational experiments. We compare two variance reduction method: importance sampling and control variate.

2 formulation

We focus on the stochastic PERT problem with crashing, where the project manager decide the allocation of his/her limited resource (budget), given the the network of projects whose process time vary randomly. It is formulated as follows.

A: set of arcs (each arc represents subproject)
$S = \{1, 2, \ldots, N\}$:
: the set of nodes, where N is number of nodes
a_{ij}: arc from node i to node j
T_{ij}: the activity time of arc a_{ij}
T_j: the arrival time of node j
B: amount of budget

- The node 1 and the node N correspond the start node and the finish node, respectively. We index a_{ij} satisfying $i < j$.
- We assume the earliest start.
- We can reduce the activity time to certain extent by consuming resources (budget).

x_{ij} : reduced time on arc a_{ij}. It is less than upper bound u_{ij}.
c_{ij} : cost per unit time to reduce
ω : scenario or sample
Ω : set of scenario
p^ω : probability of realization of the scenario ω

[master problem]:

$$\left|\begin{array}{ll} \min & Q(x) \\ \text{s.t.} & \sum_{a_{ij}\in A} c_{ij}x_{ij} \le B \\ & 0 \le x_{ij} \le u_{ij}, \quad \forall a_{ij} \in A \end{array}\right. \tag{1}$$

where $Q(x) = E[Q(T_N;\omega)]$.

[child problem]:

$$Q(T_N;\omega) = \left|\begin{array}{ll} \min & T_N^\omega \\ \text{s.t.} & T_j^\omega - T_i^\omega \ge T_{ij}^\omega - x_{ij} \quad \forall a_{ij} \\ & T_1 = 0,\ T_i^\omega \ge 0 \quad \forall i \in S \end{array}\right. \tag{2}$$

3 L–shaped method using Monte Carlo sampling

To solve the problem (1), we need to evaluate the objective function $Q(x)$. Since it is defined as expected value of the objective in (2), we can evaluate it by Monte Carlo sampling. However, crude sampling is known to be inefficient to ensure the accuracy. Numerous variance reduction methods are classified to several categories. Even though theoretical characteristic of each methods are already clarified, we need to customize and modify them to apply to individual problems.

3.1 importance sampling

$\boldsymbol{x}$: 1st stage decision variable

Let $A = \{a_1, a_2, \ldots, a_K\}$, where K is number of arcs.

T_{a_i} : process time for arc a_i (r.v.)

For an arc i, let us define the following function:

$$M_{a_i}(T_{a_i}^{\omega}, \boldsymbol{x}) = \max\left\{T_{a_i}^{\omega} + l_{a_i}(\tau, \boldsymbol{x}), L_{-a_i}(\tau, \boldsymbol{x})\right\},$$

where $l_{a_i}(\tau, \boldsymbol{x}) = l_{a_i}^{*}(\tau, \boldsymbol{x}) - T_{a_i}$ and $l_{a_i}^{*}(\tau, \boldsymbol{x})$ indicates the length of the longest path containing a_i under the base sample τ and the first stage variable $\boldsymbol{x}$. Also $L_{-a_i}(\tau, \boldsymbol{x})$ indicates the length of the longest path among all the path which do not contain arc a_i. We select the base sample as the minimum project time.

Let $\bar{M}_{a_i}(\boldsymbol{x}) = E[M_{a_i}(T_{a_i}, \boldsymbol{x})]$, then we generate the samples from the distribution of $p_{a_i} M_{a_i}/\bar{M}_{a_i}$.

The above idea can be exploited to the advanced procedure, in which we consider multiple arcs $a_1, a_2, \ldots, a_k$ rather than single arc a_i.

$$\begin{aligned} &M_{a_1,\ldots,a_k}(T_{a_1}^{\omega}, \ldots, T_{a_k}^{\omega}, \boldsymbol{x}) \\ &= \max\left\{T_{a_1}^{\omega} + l_{a_1}(\tau, \boldsymbol{x}), \ldots, T_{a_k}^{\omega} + l_{a_k}(\tau, \boldsymbol{x}),\right. \\ &\qquad\qquad \left. L_{-a_1,\ldots,-a_k}(\tau, \boldsymbol{x})\right\}' \end{aligned}$$

3.2 control variates

The effectiveness of the control variates depends on the choice of variates. They should be easy to calculate expected values and be highly correlated with targeted variable, which is complete time of project in our case.

norm–based control: Higle suggested in [2] two method to construct the control variate for general stochastic linear programs. Here we apply the norm–based control. Under out notations, control Z is

$$Z = \sum_{a_{ij} \in A} (T_{ij} - E[T_{ij}]).$$

upper bound control: We propose another control based on the upper bound of completion time. For a certain realization $\hat{\omega}$, we can calculate a critical path. Of course it may not be the critical path in other realization of ω, but it provides upper bound of completion time. We can expect that the upper bound positively correlated with true completion time. Moreover, we can immediately calculate expectation value of upper bound because it is the sum of processing time on fixed path.

$$Z = \sum_{a_{ij} \in P_E} (T_{ij} - E[T_{ij}]),$$

where P_E indicates the critical path derived from the project network with expected processing time.

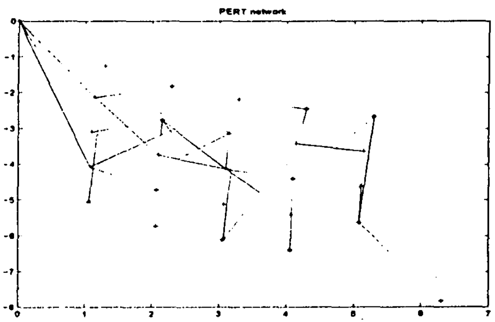

Figure 1: the project network used for the computational experiments: 26 nodes and 38 arcs

References

[1] John R. Birge and François Louveaux. *Introduction to Stochastic Programming.* Springer, 1997.

[2] Julia L. Higle. Variance reduction and objective function evaluation in stochastic linear programs. *INFORMS Journal of Computing*, 10(2):236–247, 1998.

[3] Gerd Infanger. Monte carlo (importance) sampling within a benders decomposition algorithm for stochastic linear programs. *Annals of Operations Research*, 39:69–95, 1992.