2002年日本オペレーションズ・リサーチ学会
秋季研究発表会

確率的PERT問題に対する加重サンプリング法による近似解法

01012560　駒澤大学　飯田 哲夫　IIDA Tetsuo
01206492　東北大学　鈴木 賢一　SUZUKI Ken-ichi

1　はじめに

研究開発や情報システム開発のプロジェクトでは，多くのアクティビティを含み，かつそれらの作業時間は不確実である場合がしばしばである．また，プロジェクト全体の遂行期間を短くするために，各アクティビティへの投下資金を効率的に配分することが望まれる．本研究では，各アクティビティの作業時間が不確実であるときのプロジェクト全体の完了期間の期待値を最小化する問題を扱う．

確率的 PERT 問題は，これまでに数多くの研究がなされてきている．多くはプロジェクト完了期間の期待値および確率分布を求める問題を扱っている (Iida (2000) を参照)．また，各アクティビティに資金を投下してプロジェクト完了期間の期待値を最小化する問題（クラッシング問題）に対して，Bowman (1994) は勾配推定においてIPA(Infinitesimal Perturbation Analysis) と SF(Score Function) の方法を開発した．Rubinstein and Shapiro (1993) もまたSF を用いて同様な問題を扱っている．

ここでは，確率的 PERT ネットワークにおけるクラッシング問題を，2 段階確率計画問題として定式化する．確率計画問題に対しては、確定的等価な問題への変換、切除平面法、確率的分割解法など、いくつかの解法が提案されている (例えば [5])。いずれの解法にも共通するのは、状態数の増加に対して、アルゴリズムの効率が悪化する点である。実用的な規模の問題を考える際は、状態数の増加によって生じる問題に適切に対処する必要がある。本研究では、モンテカルロ・サンプリング を採用することで、状態数の制御を試みる．Infanger (1992) は，確率計画問題に対して，推定の精度向上のために 加重サンプリング (Importance sampling) の方法を提案している．ここでは、Infanger の方法を基に，PERT ネットワークの特徴を考慮した加重サンプリングを提案する．

2　定式化

以下のような確率的 PERT ネットワークにおけるクラッシング問題を考える．プロジェクトのネットワーク表現において，アクティビティをアークとして用いる．

A：アークの集合
$S = \{1, 2, \ldots, N\}$：ノードの集合，ただし N はノード数
a_{ij}：ノード i からノード j へのアーク
T_{ij}：アーク a_{ij} の作業時間
T_j：ノード j への到達時間

- ノードのインデックスはプロジェクトのスタートノードを 1，完了ノードを N とし，アーク a_{ij} は $i < j$ となるようにノードの番号が付けられているものとする．
- 最早開始を仮定する．
- アクティビティのいくつかは，追加費用をかけることで作業時間を短縮できるとする．ただし，短縮には限度があるとする．

x_{ij}：アーク a_{ij} の短縮時間，ただし 上限 u_{ij} がある．
c_{ij}：アーク a_{ij} を単位時間短縮するのにかかる費用
ω：シナリオまたはサンプル
Ω：シナリオの集合
p^ω：シナリオ ω が実現する確率

各アークの作業時間や各ノードへの到達時間はシナリオ ω に依存するので，$T_{ij}^\omega, T_i^\omega$ と表す．そのとき，2 段階確率計画問題の拡張形の定式化は以下のとおりである．

$$
\begin{array}{ll}
\min & \sum_{\omega\in\Omega} p^\omega T_N^\omega \\
\mathrm{st} & T_j^\omega - T_i^\omega + x_{ij} \geq T_{ij}^\omega \\
 & \qquad \text{for} \quad \forall a_{ij} \in A \ \text{ and } \ \forall \omega \in \Omega \\
 & \sum_{a_{ij}\in A} c_{ij} x_{ij} \leq B \\
 & x_{ij} \leq u_{ij} \\
 & T_1 = 0,\ T_i^\omega \geq 0 \ \text{ for } \ \forall i \in S \ \text{ and } \ \forall \omega \in \Omega
\end{array}
$$

3　モンテカルロ・サンプリングを用いた L 字型解法

目的関数は区分線形の凸関数であるが，それを逐次的に近似していくアプローチである．まず，以下の親問題を

導入する．

$$\begin{array}{ll} \min & \theta \\ \text{st} & \sum_{a_{ij}\in A} c_{ij}x_{ij} \le B \\ & x_{ij} \le u_{ij} \\ & \alpha^l\theta - G^l a \ge g^l \qquad \text{for} \quad l=1,\ldots,L \quad (*) \end{array}$$

上記の問題の制約式 $(*)$ を繰り返し毎にカットとして加えていく．そのカットを生成するための子問題群として以下の問題を解く．各サンプル $\omega \in \Omega$ に対して，

$$\begin{array}{lll} \min & z^\omega = T_N^\omega & \\ \text{st} & T_j^\omega - T_i^\omega \ge T_{ij}^\omega - x_{ij} & \text{for} \quad \forall a_{ij} \\ & T_1 = 0,\ T_i^\omega \ge 0 & \text{for} \quad \forall i \end{array}$$

上記子問題の最適解に対する双対変数を $\pi_{ij}^\omega, a_{ij} \in A$ と表す．そのとき，親問題に追加するカットは，

$$g = \sum_{\omega\in\Omega} p^\omega \sum_{a_{ij}\in A} \pi_{ij}^\omega T_{ij}^\omega, \tag{1}$$

$$G = \left(\sum_{\omega\in\Omega} -p^\omega \pi_{ij}^\omega\right), \tag{2}$$

$\alpha^l = 0$ feasibility cut

$\alpha^l = 1$ optimality cut

式(1),(2)を厳密に評価するには、すべての $\omega \in \Omega$ について実現確率と実現値の情報が必要である。しかし、一般に問題の規模が増加するにつれシナリオの数は急速に増大し、各シナリオごとに子問題を精製し解を求めることは計算の負荷が大きい。この場合、すべてのシナリオについて計算を行わず、サンプリングにより限られたシナリオから解を推定する手法が有効である。このとき、カットの各係数が推定値に置き換わるため，推定されたカットを利用して得られる元問題の最適値の下限は，正確な下限ではなく推定値になる．その分散を g の分散を用いて計算する．

4 加重サンプリング

単純モンテカルロ・サンプリングは必ずしも効率が良くないので、カットの各係数の推定を効率的に行うために加重サンプリング法を用いる．

$\boldsymbol{x}$：1段階目の決定変数のベクトル，

$A = \{a_1, a_2, \ldots, a_K\}$ とする，ただし，K はアーク数を表す，

T_{a_i}：アーク a_i の作業時間の確率変数

アーク i に対して，以下の関数を考える．

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

とおく．ただし，$l_{a_i}(\tau, \boldsymbol{x})$ は，基準サンプル τ および1段階目の決定が $\boldsymbol{x}$ の下で，アーク a_i を含むパスの中で最長のもののアーク a_i の長さを除いた長さ，そして $L_{-a_i}(\tau, \boldsymbol{x})$ は，基準サンプル τ および1段階目の決定が $\boldsymbol{x}$ の下で，アーク a_i を含むパスを除いた最長パスの長さとする．基準サンプル τ は任意に選ばれるが，例えば，最小プロジェクト期間となる点が挙げられる．また，

$$\bar{M}_{a_i}(\boldsymbol{x}) = E[M_{a_i}(T_{a_i}, \boldsymbol{x})]$$

とする．そのとき，アーク a_i の作業時間のサンプリングを $p_{a_i} M_{a_i}/\bar{M}_{a_i}$ に従って行う．

同様に，開始ノードから終了ノードへのどのパスも共有しない複数のアークについて考える．表記を簡単にするため $a_1, \ldots, a_k$ がパスを共有しないアークとする．そのとき，

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

とおく．ただし，$L_{-a_1,\ldots,-a_k}(\tau, \boldsymbol{x})$ はアーク $a_1, \ldots, a_k$ のどれかを含むパスを除いた最長パスの長さとする．そして，

$$\bar{M}_{a_1,\ldots,a_k}(\boldsymbol{x}) = E[M_{a_1,\ldots,a_k}(T_{a_1}, \ldots, T_{a_k}, \boldsymbol{x})]$$

とする．そのとき，アーク $a_1, \ldots, a_k$ の作業時間のサンプリングを $p_{a_1,\ldots,a_k} M_{a_1,\ldots,a_k}/\bar{M}_{a_1,\ldots,a_k}$ に従って行う．

参考文献

[1] Bowman, R.A., "Stochastic Gradient-based Time-cost Tradeoffs in PERT Networks Using Simulation", *Annals of Operations Research*, 53(1994) 533-551.

[2] Iida, T., "Computing Bounds on Project Duration Distributions for Stochastic PERT Networks", *Naval Research Logistics*, 47(2000), 559-580.

[3] Infanger, G., "Monte Carlo (Importance) Sampling within a Benders Decomposition Algorithm for Stochastic Linear Programs", *Annals of Operations Research*, 39(1992) 69-95.

[4] Rubinstein, R. and Shapiro, A., *Discrete Event Systems*, John Wily & Sons, 1993.

[5] J. Birge and François Louveaux, F., *Introduction to Stochastic Programming*, Springer, 1997.