1996年度日本オペレーションズ・リサーチ学会
春季研究発表会

連続的に資源を投入する場合の資源配分問題

01007584 大阪工業大学 ＊ 一森 哲男 ICHIMORI Tetsuo
01204194 流通科学大学情報学部 三道 弘明 SANDOH Hiroaki

1. はじめに

資源配分問題 [1,2,3] の応用範囲は広く，負荷配分，生産計画に関する問題ばかりでなく，ソフトウェア開発工程におけるテスト資源の配分 [4] にも応用が見られる．しかし，ソフトウェアの開発工程でのテスト資源配分問題のように，資源が時間や労力である場合，配分された資源を最初に一気に投入してしまうのではなく，連続して徐々に投入することが多い．

本研究では，このように，目標物を発見するまで，配分された資源を連続して徐々に投入する場合の資源配分問題を扱う．なお，ここでは，次のような問題を取り上げる．

n 個のモジュールから構成された１つのシステムを考える．このシステムの中に重大なフォールトが１つ存在していることがわかっている．このような状況の下で，これら複数のモジュールに対して同時にフォールト探索を行うという問題を考える．できる限り少ない投入資源でフォールトを検出しようとするとき，どのように資源を配分するのが資源節約という意味で効率的か．但し，以下では，モジュール i に投入する非負の資源量を $x_i(\geq 0)$ と書き，$\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$ と表す．また，次のように仮定する．

(1) モジュール $i(i=1,2,\cdot,n)$ に資源を x 投入したときのフォールト発見確率は $1-exp(-a_ix)$ であり，$a_i(a_i>0)$ はモジュール i に固有の定数である．

(2) フォールトがモジュール i に存在する確率は p_i である．よって，$\sum_{i=1}^n p_i=1$ である．

(3) フォールトの探索は，モジュール間の情報交換なしに，独立して行われる．

2. 定式化

前述のような仮定の下で，フォールト発見確率がある指定された値α以上であるためには

$$\sum_{i=1}^{n} p_i\left(1-e^{-a_ix_i}\right)\geq\alpha \tag{1}$$

が成立しなければならない．このとき

(1) １つだけ存在するフォールトを検出し，当該モジュールでは以降の資源投入を中止し，探索を終了する．しかし，他のモジュールでは，配分された資源を投入し尽くすまで探索を行う．

(2) いずれのモジュールにおいても，配分された資源を投入し尽くしてもフォールトを検出できず，すべての探索を終了する．

のいずれか一方が成立するまでの資源の期待投入量は

$$R(\boldsymbol{x}) = \sum_{i=1}^{n}\left\{p_i\left[\int_0^{x_i} y\left(1-e^{-a_iy}\right)'dy+x_ie^{-a_ix_i}\right] +(1-p_i)\,x_i\right\} \tag{2}$$

で与えられる．よって，式 (1) の制約の下で，式 (2) の期待投入量を最小にすればよい．

3. 解析

上に定式化した問題は，次のように整理することができる．

$$\begin{array}{ll} \min & \sum_{i=1}^n f_i(x_i) \\ \text{s.t.} & \sum_{i=1}^n g_i(x_i)\leq 1-\alpha \\ & x_i\geq 0,\ i=1,2,\cdots,n \end{array} \tag{3}$$

ここに

$$f_i(x_i) = \frac{p_i}{a_i}\left(1-e^{-a_i x_i}\right)+(1-p_i)\,x_i \quad (4)$$

$$g_i(x_i) = p_i e^{-a_i x_i} \quad (5)$$

である．

式(3)に示した問題に対するKuhn–Tucker条件を列挙すると，次のようになる．

$$p_i e^{-a_i x_i}+1-p_i+\lambda\left(-p_i a_i e^{-a_i x_i}\right) -\mu_i=0 \quad (i=1,2,\cdots,n) \quad (6)$$

$$\lambda\left[\sum_{i=1}^{n} p_i e^{-a_i x_i}-(1-\alpha)\right]=0 \quad (7)$$

$$\mu_i x_i=0 \quad (i=1,2,\cdots,n) \quad (8)$$

$$\mu_i \geq 0 \quad (i=1,2,\cdots,n) \quad (9)$$

$$\sum_{i=1}^{n} p_i e^{-a_i x_i} \leq 1-\alpha \quad (10)$$

$$\lambda \geq 0 \quad (11)$$

$$x_i \geq 0 \quad (i=1,2,\cdots,n) \quad (12)$$

ここでは詳細は省略するが，上に列挙したKuhn–Tucker条件を満たす$x_i, \mu_i (i=1,2,\cdots,n), \lambda$は，唯1つしか存在しないことが示される．このようなλをλ^*と書き，一般性を失うことなく

$$0<\frac{1}{a_1 p_1}<\frac{1}{a_2 p_2}<\cdots<\frac{1}{a_n p_n} \quad (13)$$

と仮定すると，式(3)の問題に対する最適解は，以下のようにして求めることができる．

[1] λ^*を含む区間

$$\left(\frac{1}{a_K p_K}, \frac{1}{a_{K+1} p_{K+1}}\right], \ K=1,2,\cdots,n-1$$

あるいは

$$\left(\frac{1}{a_n p_n}, +\infty\right)$$

を見つける．このためには

[2] 方程式

$$\sum_{i=1}^{K} \frac{1-p_i}{\lambda a_i-1}+\sum_{i=K+1}^{n} p_i=1-\alpha \qquad K=1,2,\cdots,n-1 \quad (14)$$

あるいは

$$\sum_{i=1}^{n} \frac{1-p_i}{\lambda a_i-1}=1-\alpha \quad (15)$$

を解き，その解λ^*を求める．

[3] λ^*が方程式(14)の解であれば，これに対応するKをK^*と表すと

$$\begin{cases} x_i^*=\frac{1}{a_i}\log\frac{p_i(\lambda^* a_i-1)}{1-p_i}, & i=1,2,\cdots,K^* \\ x_i^*=0, & i=K^*+1,\cdots,n \end{cases}$$

が最適解である．この場合，資源投入を行うのはモジュール$1,2,\cdots,K^*$に対してのみであり，$K^*+1,\cdots,n$に対しては資源投入を行わないこととなる．

[4] λ^*が方程式(15)の解であれば

$$x_i^* = \frac{1}{a_i}\log\frac{p_i(\lambda^* a_i-1)}{1-p_i} \qquad i=1,2,\cdots,n$$

が最適解である．この場合，すべてのモジュールに対して資源投入を行うこととなる．

4. 数値例

紙数の関係上，数値例は当日報告させて頂く．

参考文献

[1] B.O. Koopman, "The theory of search :part III, the optimum distribution of searching effort," *Operations Research*, Vol. 5, 1957, pp. 613-626.

[2] A. Charnes and W.W. Cooper, "The theory of search: Optimal distribution of effort," *Management Science*, Vol. 5, 1958, pp. 44-49.

[3] T. Ibaraki and N. Katoh, *Resource Allocation Problems–Algorithmic Approaches*, The MIT Press,1968.

[4] 西脇正敏，山田茂，一森哲男，尾崎俊治，"ソフトウェア開発における資源配分問題，" OR学会春季研究発表会アブストラクト集，1993, pp. 234-235.