

文 献 抄 録

Hartigan, J.A., "Probabilistic Completion of a Knochout Tournament," *Annals of Mathematical Statistics*, **37**, 2 (1966), 495—503.

[決定/統計/理論的]

[問題]

トーナメントの試合において、実力の上の者が必ず勝つという前提のものでは1位は確定するが2位以下の順位は定まらない。この論文では 2^n 人の選手によるトーナメントの試合において、各選手の順位がどのように推定されるかを論じている。

[仮定]

1. 2^n 人の選手でトーナメントを行なう。その際の組合せ法は $(2^n)!$ 通りあるが、トーナメントの開始にあたってこの中の一つがランダムに選ばれるものとする。

2. a, b 二人の選手の順位については、 $a=a_1, b=a_k$ として a_i が a_{i-1} に勝っているという選手の系列

$$(1) a_1, a_2, \dots, a_n$$

が存在するとき $a < b$ であるとする。

[トーナメントの2進表示]

トーナメントにおいて a よりも上位あるいは同等のものを $a_1, a_2, \dots, a_k$ として a_i は n_i 回戦まで勝ち進んだとする。

$$(2) f(a) = \sum_{i=1}^k 2^{n_i-1}$$

なる写像を考えると、トーナメントの結果に整数

$$(3) 2^n, 2^n+1, \dots, 2^{n+1}-1$$

に1対1の対応をもって写される。 A_n を 2^n 人の選手の集合、 f_n を A_n から $2^n, 2^n+1, \dots, 2^{n+1}-1$ への写像として、 A_n と f_n の組合せも

$$(4) T_n = (A_n, f_n)$$

とし、 ω を優勝者、 C_ω を ω を優勝者とした場合の他の選手の可能な順位づけの集合とする。

$$(5) A_k = \{a | 2^k \leq f_n(a) - 2^n \leq 2^{k+1} - 1\}$$

$$(6) f_k(a) = f_n(a) - 2^n, a \in A_k, T_k = (A_k, f_k)$$

とすれば

$$(7) A_n = \{\omega\} \cup A_1 \cup A_2 \cup \dots \cup A_{n-1}$$

$$(8) T_n = C_\omega \cap T_0 \cap T_1 \cap \dots \cap T_{n-1}$$

となる。この分解により大きなトーナメント試合をいくつかの小さなトーナメントに分けて考えることができ、確率の計算が簡単になる。

[確率、期待順位の計算]

$$(9) P_n(a < b) = P(a < b | T_n)$$

$$(10) P_{n,a}(i | T_n) = P(a \text{ の最終順位が } i | T_n)$$

$$(11) E_{n,a}(i | T_n) = i \text{ の期待値}$$

とすれば、 $a=\omega$ の場合は

$$(12) P_{n,a}(i | T_n) = 1 \quad (i=1 \text{ のとき})$$

となる。 A_k の要素はトーナメントの結果に矛盾しない適当な順位がつけられており、 A_k に含まれない $A-A_k$ の要素も同様に順位づけられているとし、両者を合わせた場合の順位を考えると、 A_k と $A-A_k$ の間で成り立たなければならない条件は次の条件だけである。

$$(13) b < \omega \quad \text{for all } b \neq \omega$$

したがって A_k の p 位のものの上に $A-A_k$ の要素がくる個数は全く確率的に決まり、

$$(14) P_{n,a}(i | T_n) = \sum_{j+p=i-1} P_{k,a}(p | T_k) \cdot$$

$$u_{j,n_1,n_2}$$

となる。ただし

$$(15) n_1 = 2^n - 2^k - 1, \quad n_2 = 2^k$$

で、 u_{j,p,n_1,n_2} は $A-A_k$ の j 個の要素が a の前に配列される確率である。

次に $P_n(a < b)$ について考えると、もし a と b が共に A_k に含まれておれば

$$(16) P_n(a < b) = P_k(a < b)$$

で、 a, b が別の A_k たたとえば A_{k_1}, A_{k_2} にあれば

$$(17) P_n(a < b) = \sum_{k < i} P_{k_1,a}(i) P_{k_2,a}(j) \cdot$$

$$u_{k,j,n_1,n_2}$$

となる。最後に $E_{n,a}(i | T_n)$ についてはもし $a=\omega$ であれば $E_{n,a}(i | T_n) = 1$ となり、 $a \in A_k$ でこの中で p 位であれば全体では

$$(18) p + p(2^n - 2^k - 1) / (2^k + 1) + 1$$

位になる。ここで第2項は $A-A_k - \{\omega\}$ の要素でよりも上位にくるものの期待数、第3項は ω による。これから

$$(19) E_{n,a}(i | T_n) =$$

$$E(p + p(2^n - 2^k - 1) / (2^k + 1) | T_k)$$

$$(20) E_{n,a}(i | T_n) = E_{k,a}(i | T_k) 2^n / (2^k + 1) + 1$$

$$(21) E_{n,a}(i | T_n) =$$

$$1 + 2^n / (2^{n-k} + 1) + 2^n \cdot 2^{n-k-1} / \dots$$

などと表わすことができる。またその期待値の分散については

$$(22) E_{n,a}(i(i-1) | T_n) = E_{k,a}(i(i+1) | T_k)$$

$$\times 2^n(2^n+1)/(2^k+1)(2^k+2)$$

から求めることができる。原論文には $0 \leq n \leq 6$ について、各要素の期待順位とその分数を計算した表がある。(久米 均)

Gilbert, J.P. and F. Mosteller, "Recognizing the Maximum of a Sequence," J. American Statistical Association, 61, 313 (1966), 35-73.

[決定/確率の計算/理論的]

1. よく知られている次のような問題 (著者は **dowry problem** とよぶ) から出発する。

「壺の中に n 枚の札があり、各々の札にはすべて異なる数字が書いてある。player はこの壺から 1 枚ずつ random に非復元抽出をするものとし、札を引くとそれまでに引いた札の中での大きさの順位が知らされ、それによってその札をとるか否かをきめる。とることに決めるとゲームは終了、最後の札が n 枚の中で最大数の書かれた札であるときのみ成功とする。

このとき成功する確率を最大にするような抽出の方法はどんなものか? またそのとき成功する確率はどのくらいか?

解答は次のようである: 最初何枚か ($s-1$ 枚) をみておいて (どれも棄てる) s 枚目以後はそれまでの最大数より大きいものが出現したときだけそれをとるという方法がよい。 s の大きさは不等式

$$\sum_{i=s}^{n-1} \frac{1}{i} < 1 < \sum_{i=s-1}^{n-1} \frac{1}{i}$$

をみたすものが好ましく、このような s とそれに応じた成功の確率 $\frac{s-1}{n} \sum_{k=s-1}^{n-1} \frac{1}{k}$ が表にされている (表 2)。

$n \rightarrow \infty$ のとき s は大体 n/e で確率は e^{-1} に近づく。

2. 次に問題を一般化して、同じ状況の下で札を r 枚とることが許される場合 (上の場合は $r=1$) はどうなるかをみている。勿論 r 枚の中に最大数の書かれた札があれば成功とする。

例えば $r=2$ のときは、最初に何枚かパスして (s_1 枚) 以後にそれまでの中で最大数のものを選び、1 枚選ばれたらあとは $s_2 (> s_1)$ 枚までを見ていき、その後 1 の場合のようにそれまで知ったの数よりも大きなものが出てきたときその札をとることにする。表 3 には $r=s_1+1$, $s=s_2+1$ として両者の最適値 r^* , s^* およびそれに応じて成功する

確率が各種の n について載っている。

$n \rightarrow \infty$ のとき, r^* , s^* はそれぞれ $n/e^{1/2}$, n/e に近く、成功する確率は 0.5910 程度である。

同じく $r=2$ のとき、より単純な手段として、1 枚までパスしてその後大きな 2 枚をとるという方法が考えられる。 n が大きいとき最適な t は約 $n/e^{1/2}$ で、成功する確率は約 0.5869 である。これは上の最適手段の場合の確率には及ばないが、十分近い値であることは注目すべきであろう。

$r > 2$ のとき、議論は相当複雑になるけれども考え方は $r=2$ のときと同様である。

3. はじめの問題で少し状況をかえ player が札に書かれた数字の分布についての知識をもつとすれば手段は変わってくる。大小の順が問題だから尺度は適当にかえてよく、連続分布の場合なら、札の数字は母集団分布が一樣分布のときの大いさ n の任意標本と考えてよい。ゲームは札を引く度にそれに書いてある数値が知らされるものとし、あとは 1 と同様にして最大なものを選ぶことを目標とする。この場合は **decision number** ($d. n.$) と呼ばれる減少数列 $d_1, d_2, \dots, d_n (1 \leq d_i \leq 0)$ を予めきめておき、 i 回目の抽出で d_i より大きいものが現れたらそれをとることにするのが良い手段である。もし r 回まではどれも $d. n.$ より小さく $r+1$ 回目にはじめて d_{r+1} をこえたとすればその札をとることになるが、このとき成功する確率は

$$P(r+1) = \sum_{i=1}^r d_i^r / r(n-r) - \sum_{i=1}^r d_i^n / n(n-r) - d_{r+1}^n / n, \quad (n-1 \geq r \geq 1).$$

これをもとにして最適な $d. n.$ の列をえらぶこと、および成功する確率等が議論される。

4. この他、相手があって札の順序をかえうるような場合はゲームの問題として扱えること、また 3 のように分布が知られているときは **payoff** すなわち平均値を最大にする手段等々が扱われている。全体を通じ比喩的説明が各所にあるし、また多くの詳細な表は実際家にとっても大いに役立つと思われる。(飛田武幸)

Thompson, G.L., F.M. Tonge and S. Zions, "Techniques for Removing Nonbinding Constraints and Extraneous Variables from Linear Programming Problems," Management Science, 12, 7 (1966), 588-608.

〔線型計画／いらない制約と変数除去／理論的〕

LPの問題を定式化するとき、どの制約が最適値にきいているかが簡単にはわからないので、実際には最適値にきいていない制約条件も含めて考えている。LP問題をつぎのようにかく、

$$\begin{aligned} &\text{maximize} && C'X \\ &\text{subject to} && AX \leq B, X \geq 0. \end{aligned}$$

ある制約条件を $A_iX \leq b_i$ としたとき、残りの制約条件を $A'X \leq B'$, $X \geq 0$ とかくことにする。

定義・制約条件 $A_iX \leq b_i$ が *redundant* $\leftrightarrow AX \leq B$, $X \geq 0$ で表わされる *convex set* と $A'X \leq B'$, $X \geq 0$ で表わされる *convex set* が同一のものであること。

定義・最適点で等号を満足する *non-redundant* な制約条件を *binding constraint* という。そ(ノ)

(ノ)の他は *non-binding constraint* という。もし制約条件 $A_iX \leq b_i$ が最適点 X^0 に対して *non-binding* であれば、 $A_iX < b_i$ である。

定義・最適点で *non-positive* な変数を *extra-neous* であるという。

この論文では、まず制約条件の中から *redundant* なものを取り除く方法を述べている。つぎに解を求めてゆく途中で *non-binding* な制約条件や *extraneous* な変数をみつけ、取り除き *matrix* を縮小しながらシンプレックス計算をしてゆく方法を述べている。この方法のメリットは、

- (1) 計算時間の短縮
- (2) *degeneracy* と *cycling* の解決

である。実際に5つのLP問題についてこの方法を適用した結果が次表のごとく出ている。

問 題 番 号	1	2	3	4	5
問 題 の 大 き さ	27×40	55×104	53×77	9×20	24×29
繰 返 し 回 数 (普通のシンプレックス法)	39	87	85	9	46
繰 返 し 回 数 (手を加えてからのシンプレックス法)	35	80	74	9	32
最 終 の 問 題 の 大 き さ	15×24	44×70	32×64	7×18	11×16

(若山邦紘)

Rothkoph, Michael H. "Scheduling with Random Service Times," *Management Science*, **12**, 9, 1966, 707~713.

〔順序づけ／ランダム作業時間／理論的〕

1工程だけで加工が済む仕事に既に m 個ある。現時点 ($t=0$) から、各仕事が完了するまでの時間には待ち費用がかかる。総費用を最小にするには、 m 個をどんな順序で加工したらよいか。

この問題には、個々の仕事の(番号*i*)所要時間が既知の定数であり、待ち時間に比例して費用 $c_i t$ がかかる場合に関しては、機械が1台の場合にも、同種の機械が並行に数台あってどれにかけてもいい場合にも、最適順序を求めることはできる。

待ち費用に割引率を掛けた $c_i e^{-rt} dt$ とときや、単調非減少な函数のときについても、この著者は、最適順序を与える手順と、作業を分割しない(未完了作業を機械からはずさない)最適順序のあることを既に示している (*Management Sci.*, **12**, 5 (1966), 431~47)。

本論文は、作業時間がランダムで、待ち費用が1次あるいは指数函数の場合には、作業時間が既知定数の場合に帰着して、最適順序を求められること、やはり分割を含まない最適順序があることを示(＊)

(＊) したものである。

待ち費用が1次で率が c_i 、作業時間を T_i とし、仕事が完了するまで機械からはずさないことにすれば、総待ち費用 C は(番号順に加工すれば)、

$$C = \sum_{i=1}^m c_i \sum_{k=1}^i T_k = \sum_{i=1}^m \sum_{k=i}^m c_k T_i.$$

この値は、 c_i/T_i の減少順が最適順序である。

T_i が分布し、同時分布函数を $P(T_1 \cdots T_m)$ 、 T_i の期待値を T_i^* とすると、 C の期待値 C^* は、

$$\begin{aligned} C^* &= \int_0^\infty \cdots \int_0^\infty \sum_{i=1}^m \sum_{k=i}^m T_i dP(T_1 \cdots T_m) \\ &= \sum_{i=1}^m \sum_{k=i}^m c_k T_i^* \end{aligned}$$

となるので、定作業時間のときのように c_i/T_i^* の減少順が最適である。

待ち費用が $c_i e^{-rt}$ になり、機械1台だと、

$$\begin{aligned} C &= \sum_{i=1}^m c_i \left[1 - \exp(-r \sum_{k=1}^i T_k) \right] / r \\ &= \sum_{i=1}^m c_i \left[1 - \prod_{k=1}^i \exp(-r T_k) \right] / r \end{aligned}$$

となる。

T_i が独立に分布し、分布函数を $P_i(T_i)$ とすると、 T_i に関して期待値をとれば、期待値は、

$$C^* = \sum_{i=1}^m c_i \left[1 - \prod_{k=1}^i P_k^T(r) \right] / r$$

となる。この $P_k^T(r)$ は、 $P_i(T_i)$ の Laplace-Stieltjes 変換である。そこで $P_k^T(r)$ を、 C 中の $\exp(-rT_k)$ と考えれば、この場合の最適順序も、著者の前記の論文にある方法で、求められる。

以上は、機械が並行に複数台の場合にも論じられている。

待ち費用が時間に比例するときに、作業時間が独立な指数分布をしていると、分割のない最適順序が存在することを、機械が1台および複数台のときについて証明している。
(真鍋龍太郎)

Soriano, A. "Comparison of Two Scheduling Systems," *Operations Research*, **14**, 3 (1966), 388—97.

[病院/待ち行列/応用的]

病院、診療所などでは外来患者の診察を合理的に行ない、その診察時間を少なくすることが主要な問題の1つであり、米国でもジョン・ホプキンス大学ケース工科大学などの OR グループがこの問題を検討している。本論文もこの種の研究の一部をなすもので、ウィルマー診療所 (Wilmer Outpatient Clinic) において実際適用された2つの「指定時診療制度」の比較が中心になっている。

現在米国で最も普通に行なわれている「指定時診療制度」は以下の3つに大別出来る。

(1) Pure Block Appointment System

診察日が同一のすべての患者に対して、その日の診察開始時を診察時刻として指定するもの

(2) Individual Appointment System

各患者に別々の診察時を指定するもので、指定時刻の間隔は一定である。

(3) Mixed Block-Individual Appointment System

何人かの患者のグループに対しては、診察開始時刻を診察時刻として指定し、他の患者に対しては(2)と同様に各人別々に診察時を指定するもので、(1)と(2)の混合タイプである。

ところで本論文では Block Appointment System を採用すること ((1), (3) の方式) はヒューマニティの見地から適当でないとし、一方(2)の方式では医師の診察時間が長引いた場合問題があるとの見地から、新しい System として "Two-at-a-Time Appointment System" を導入している。これは2人づつの患者に同一診察時を指定するもので、指定時刻の間隔は一定である。標題の2つの System

の比較とは、前述の(2)と、この新しい System との比較のことである。(以下簡単のために前者を System I, 後者を System II とする。)

[仮定]

(i) 窓口1つ、先着順サービス

(ii) 指定時に患者が到着するものとして input は regular とする

(iii) 診察時間はガンマ分布とする

(実際ウィルマー診療所において、平均20.1分、標準偏差10.05分のガンマ分布をすることを適合度検定により確めている)

[解析方法、結果]

(i) System II では、所謂 batch arrival (大きさは2)の待行列問題である。この論文では、与えられた ρ を満たすための指定時刻の間隔の表が計算されている。(これは何等特別の理論を必要とせず容易に計算できる。)

(ii) System I と System II の比較

著者が本論文と同一雑誌に発表している "On the Problem of Batch Arrivals and Its Application to a Scheduling System" *Oper. Res.* **14**, 3 (1966) 398—408 の結果の一部を利用している。

待行列のタイプ: D/E/1

$F(x)$: System I の steady state な待時間の分布関数 $F_1(x)$; $F_2(x)$: System II における steady state な時間の分布関数で、 $F_1(x)$ は先に診察をうける患者のもの、 $F_2(x)$ は後のものの

上記文献によると $F(x)$ 等は以下の様与えられる。

$$F(x) = 1 + \sum_{j=1}^4 c_j e^{z_j x}$$

ただし、 c_j, z_j は $\left(\frac{\mu}{2}\right)^4 / \left[(\mu/2) + \frac{z_j}{2}\right]^4 = e^{-\frac{z_j}{2}}$

$$\sum_{j=0}^4 \left\{ c_j / \left[\left(\frac{\mu}{2}\right) + \frac{z_j}{2} \right]^k \right\} = 0, \quad (k=1, 2, 3, 4)$$

の根として与えられる。(ここで $c_0=1, z_0=0$ とする。)

$$F_1(x) = 1 + \sum_{j=1}^8 c_j e^{z_j x}$$

$$F_2(x) = \frac{4}{3} \int_0^x F_1(x-y) y^3 e^{-\mu y} dy$$

ここで c_j, z_j は $\mu^8 / (\mu + z_j)^8 = e^{-z_j}$

$$\sum_{j=0}^8 [c_j / (\mu + z_j)^8] = 0, \quad (k=1, 2, \dots, 8)$$

の根として与えられる。(ただし $c_0=1, z_0=0$)

以上の公式をもとに Borrough 220 による計算結果がグラフ化されている。(1例として横軸に待時間 x , 縦軸に $F(x), F_1(x), F_2(x)$ の値をとっている)これによると待時間の少い順序は $F_1(x), F(x), F_2(x)$ となっている。 ρ が増加するに従って, その差は小さくなる。 $F_1(x)$ の待時間が $F(x)$ のそれより少いのは後から診察を受けるものの犠牲の上に成り立っていることであり, グラフ上で判断する限り, System II が System I に比して合理的であると結論出来ない様に思われる。ただしウィルマー診療所では System II の採用により患の平均待時間を50%下げることができたとのべている。

(梅林光寿)

Pritsker, A.A.B. and W.W. Happ, "GERT: Graphical Evaluation and Review Technique, Part I. Fundamentals," J. of Industrial Engineering, 17, 5 (1966), 267—274, "Part II. Probabilistic and Industrial Engineering Applications," 17, 6 (1966), 293—301.

[ネットワーク/グラフ理論/応用的]

この論文では Part I で GERT の基本的な考え方および処理手続を説明しており, Part II でその応用可能性を論じている。GERT は論理 node と数個のパラメータを持った branch によって形成された stochastic network を解析する方法である。GERT は通常次のステップで処理がなされる。

1. 問題を stochastic network の形であらわす。
2. network の branch を表わすデータを集める。
3. network の関数および等価関数を決定する。
4. 等価関数を次の2つの network 特性量に変換する。
 - i) 特定の node が実現される確率。
 - ii) 等価 network に対する時間の M.G.F..
5. 4で得られた情報より system の評価をおこなう。

まずステップ 1, 2 では Exclusive-or, Inclusive-or, And 等の論理記号を使って stochastic network を作成し, network の branch のパラメータを決定する。

このパラメータは一般に branch の実現確率と所要時間とであるが, 個数や属性には制約がない。

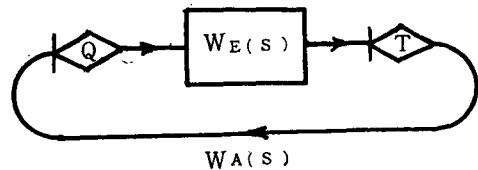
ステップ 3, 4 では network 解析をおこなうわけであるが, Exclusive-or node に関してのみ数学的に取扱われ, Inclusive-or, And node は Exclusive-or node への変換をせねばならない。解析の目的は multi-branch network を one branch network へ等価変換することである。第 1 に branch の時間 t は t の M.G.F. $M_t(s) = E\{e^{st}\}$ で定義される。第 2 に branch の実現確率を p とすると, w 関数 $w(s) = pM_t(s)$ が branch について定義される。2本の branch A, B の直列, 並列結合を考えると, 等価 w 関数 $w_E(s)$ はそれぞれ $w_a(s) \cdot w_b(s), w_a(s) + w_b(s)$ になる。Exclusive-or の場合には線型 network になるからトポロジー方程式を使える。閉じた network でのトポロジー方程式は

$$H(s) = 1 + \sum_m \sum_i (-1)^m L_i(m, s) = 0$$

となる。 $L_i(m)$ は order m の i 番目の loop である。 $L_i(m, s)$ は次式によって決定される。

$$L_i(1, s) = \prod_{j \in i} w_j(s), \quad L_i(m, s) = \prod_k L_k(1, s)$$

ここで k は m 個の disjoint loop の番号をとる。



いま図のような network で black box の $w_E(s)$ を決定するには, $w_A(s)$ という branch を加えるとトポロジー方程式は

$$H(s) = 1 - w_E(s) w_A(s) = 0$$

となり

$w_E(s) = 1/w_A(s)$ と決定される。 $w_E(s)$ はまた Mason's Rule を使っても決定される。 $w_E(s)$ が求まると, P_E と M_E は $p_E = w_E(0), M_E(s) = w_E(s)/w_E(0)$ として求められ, $M_E(s)$ より moment, cumulant 等が求められる。

ステップ 5 では以上の結果より, system の特性値に対する信頼限界, branch のパラメータ変化に対する感度等を求めて system の評価をおこなう。

以上の 5 ステップによって GERT の解析がおこなわれるが, いままでのところ数学的にうまくゆくのは Exclusive-or node の場合だけであり, その他の論理 node の場合にはまだうまい方法がみつ

かっていないようである。Part II では、確率および I.E. に関する 9 例題を実際に GERT を使って解析をおこない、実際の計算方法をしめし、stochastic な問題への GERT の応用可能性を論じている。

GERT 手法はまだ計算手続、評価手続がまとまっていず、今後の問題として残されているところが多い。
(森 健一)

Menon, V.V., "The Minimal Cost Flow Problem with Convex Cost," *Naval Research Logistics Quarterly*, 12, 2 (1965), 163-172.

[ネットワーク/最小費用のフロー/理論的]

Minimal cost flow 問題において、各 arc の輸送費が凸関数によって与えられているより一般的な場合の解法について述べている。輸送費が一次関数で与えられる問題については、primal-dual 法がよく知られているが、ここでは primal algorithm によって問題を解決している。

今 network に flow $F = \{x_{ij} | (v_i, v_j) \in a\}$ が与えられている。この network 中の特定の simple cycle C について、次のような blow の変更を与える、

$$(1) \quad x_{ij}^* = \begin{cases} x_{ij} + \varepsilon & (v_i, v_j) \text{ が } C \text{ の前向き arc} \\ x_{ij} - \varepsilon & (v_i, v_j) \text{ が } C \text{ の逆向き arc} \\ x_{ij} & (v_i, v_j) \text{ が } C \text{ の arc でない} \end{cases}$$

$F^* = \{x_{ij} | (v_i, v_j) \in a\}$ が各 arc の容量制約を満たすならば、このフローの変更を

$$(2) \quad F^* = T_{c, \varepsilon}(F)$$

と書き、 $T_{c, \varepsilon}$ を valid transformation と呼ぶ。

各 arc に費用関数 $f_{ij}(x_{ij})$ が与えられた時、simple cycle C を回る flow の cost の総和に關して、

$$(3) \quad \sum_C f_{ij}(x_{ij}) \leq \sum_C f_{ij}(x_{ij}^*)$$

が成立つならば、flow F に関して cycle C が admissible であると呼ぶ。cycle C の廻りの flow を a_i 、 C の前向き arc の集合を I^+ 、逆向き arc の集合を I^- としたとき、次のような x に関する凸関数を考える。

$$(4) \quad C(a_i, i \in I; x) = \sum_{i \in I^+} f_i(a_i + x) + \sum_{i \in I^-} f_i(a_i - x)$$

ただし x は $T_{c, x}$ valid がであるような値だけを考慮することにする。すると flow F に関して cycle が admissible であるための必要十分条件は、(4) 式の間関数が $x=0$ において minimum になることである。この条件は、

$$(5) \quad \begin{cases} \sum_{i \in I^+} f_i^+(a_i + x) \geq \sum_{i \in I^-} f_i^-(a_i), & x > 0 \\ \sum_{i \in I^+} f_i^+(a_i + x) \leq \sum_{i \in I^-} f_i^-(a_i), & x < 0 \end{cases}$$

とも表わすことができる。

以上の準備をした上で、minimal cost flow のための criterion として次の定理が証明される。

“flow F が optimal であるための必要十分条件は、 F のすべての simple cycle が admissible となることである。”

上の定理にもとずき、次のような algorithm が作られる。

- 1) arc の与えられた容量制約の下で maximal flow 問題を解き、flow F_0 を作る。
- 2) network の simple cycle を列挙する。
- 3) 各 simple cycle C について、admissible であるか否かを調べ、
- 4) もし admissible でないなら、 $T_{c, \varepsilon_0}(F)$ によって C が admissible になるような ε_0 を求め、 $T_{c, \varepsilon_0}(F)$ について 3) 以下を繰返す。
- 5) すべての C が admissible になれば、その時の flow は minimal cost flow である。

終りに制約、flow に整数条件が与えられた場合、2, 3 の簡単な関数での ε_0 の計算法についても触れいる。
(山本正明)

Pontecorvo, A.B. "A Method of Predicting Human Reliability" *Annals of Reliability and Maintainability*, Vol. 4 (Fourth Annual Reliability and Maintainability Conference, July 1965), 337-342.

[信頼性/回帰分析/応用的]

高度に自動化された機器系であってもその運転や保全のために何らかの程度まで人間の手作業が入ることは否定できない事実である。この場合、系全体の信頼度や保全度を評価するためにはこの Human Factor を無視することはできない。そして問題なのはこの Factor をほかの機械部品の動作特性と共通の尺度で表現し、系全体の信頼度や保全度にそれが寄与する度合いを量的に表現することである。実用上強く要望されながらも信頼性理論のなかで最もたちおくれているこの分野にメスを入れ、人手による作業の信頼度を推定しようというのがこの論文のねらいである。推定の手順を著者は 6 段階に分けて説明している。まず 1. 機械の“動作点検”のように 1 つの完結した動作をあらわす作業 (task) を明確にすること、この作業は一連の単純作業 (sub-

task) からなるものであること. 2. 作業要素(task element) すなわち単純作業を明確にすること, この単純作業は作業を完了するために必要不可欠なものであり, ほかの作業にも共通する作業要素であること. 3. 単純作業の遂行の信頼度をあらわす経験データを手に入れること, この場合, 作業環境や作業状態は次の述べる“評点”のときと同じ条件であること. 4. 単純作業の難しさ, すなわち, まちがいの起しやすさをあらわす評点を行なうこと, これは作業内容全体を作業員の熟練度に至るまでよく知り抜いている人達に判定してもらうもので, “評点”は単純作業の難しさの程度が十分判定できさえすればよいが, 10点の scale でもよいこと. 5. できるだけ多への単純作業に対して上の 3. のまちがいを起こす確率の経験データと 4. の評点のデータを集め, 両者の関係を regression line の形にあらわし, 適合度の検定を行なうこと. この論文に示されている実例では

$$\text{Log } E = -2.9174 + 0.006122R$$

という式が得られている. ここに E はまちがいを起こす確率 (error rate), R は 4. の評点の合計をあらわす. この関係式を使うと経験データのない単純作業の信頼度 $1-E$ を R の値から推定できること. 最後に 6. 作業の信頼度を 5. の手順で推定した各単純作業の信頼度の積として計算すること. この場合, 機械部品の redundancy と同様に人手による作業でも余分な審査や作業の重複によって信頼

Task Element	Rating Mean	Reliability S.D.	Estimate
Read technical instruction	8.3	2.2	.9918
Read time (Brush Recorder)	8.2	2.1	.9921
Read electrical or flow meter	7.0	2.8	.9945
Inspect for loose bolts and clamp	6.4	1.9	.9955
Position multiple position electrical switch	6.3	2.4	.9957
Mark position of component	6.2	2.1	.9958
Install lockwire	6.0	2.3	.9961
Inspect for bellows distortion	6.0	2.7	.9961
Install Marman clamp	6.0	1.8	.9961
Install gasket	6.0	2.1	.9962
Inspect for rust and corrosion	5.9	2.1	.9963
Install "O" ring	5.7	2.2	.9965
Record reading	5.7	2.3	.9966
Inspect for dents, cracks and scratches	5.6	2.4	.9967

度を向上させることができ, それらの数学的な処理法も機械部品の場合と全く同様である.

以上の方法によって推定された作業要素 (task element) の信頼度 (reliability) が評点 (rating) のデータと一緒に表に示されている. この表の一部分を下に示す.

このような推定量には作業環境, 作業条件, 作業員の訓練, 作業員の資質, 評点を行なう人の勘などいろいろな原因のカタヨリが入り込みやすいと思われるのでその取扱いは注意しなければならないがひとつの貴重な参考資料となるであろう.

(阿部俊一)

Van de Panne, C. "Programming with a Quadratic Constraint," *Management Science*, 12, 11 (1966), 798—815.

[2次計画/LP+2次の制約式/理論的]

LP に2次式の制約条件が1つ加わった, 次の問題の解法を論じたもの.

$$(1) \begin{cases} \text{目的函数: } c'x \rightarrow \text{最大} \\ \text{制約条件: } p'x + \frac{1}{2}x'Cx \leq \beta \\ Ax \leq b \\ x \geq 0 \end{cases}$$

この問題は次のような場合に発生する. 条件 $a'x \geq \beta$ をある確率以上にしたという chance-constraint:

$$(2) \text{Prob } \{a'x \geq \beta\} \geq \gamma$$

がLPに加わっているとす. 係数 a が正規分布し, 期待値を α , 共分散行列を V , とすると(2)は,

$$(3) \alpha'x - \phi \cdot (x'Vx)^{\frac{1}{2}} \geq \beta$$

となる. ϕ は要求水準 γ に当る標準偏差の倍数. (3) を2乗して制約式に加えると, 問題(1)になっている.

(1)を解く第1段階として, まず

$$(4) \begin{cases} \text{目的函数: } c'x \rightarrow \text{最大} \\ \text{制約条件: } Ax \leq b, x \geq 0 \end{cases}$$

を解く. 最適解 x^0 を得たとす. $p'x^0 + \frac{1}{2}x^0'Cx^0 \leq \beta$ のときは, x^0 が(1)の解にもなっている. $>\beta$ なら次の段階に進む.

第2段階では, 次の2次計画 (QP) 問題

$$(5) \begin{cases} \text{目的函数: } f(x) = p'x + \frac{1}{2}x'Cx \rightarrow \text{最小} \\ \text{制約条件: } Ax \leq b \\ c'x \geq \lambda, x \geq 0 \end{cases}$$

を, $\lambda_0 = c'x^0$ から出発して, λ を減少させてパラメ

トリックに解く。その結果は次の2つの場合になる。

(a) ある λ の値に対する最適解で目的函数はまだ β 以上なのに、第2の制約式が効かなくなったとき。このときには、許容な λ と x がなく(1)が解けない場合である。

(b) $\lambda = \lambda_F$ に対し、 $p'x + \frac{1}{2}x'Cx = \beta$ になったら、(5)の解は(1)の解にもなっている。これは(5)と(1)の Kuhn-Tucker 条件が一致することで証明されている。

この解法での特色は次の点にある。(5)の Kuhn-Tucker 条件

$$(6) \begin{cases} -p - Cx + cv_1 - A'v + u = 0 \\ Ax + y = b \\ c'x - y_1 = \lambda_F \\ u'x + v'y + v_1y_1 = 0 \\ x, y, y_1, u, v, v_1 \geq 0 \end{cases}$$

の上の3式と、(5)の目的函数を変形した

$$(7) F(x) \equiv 2f(x) = -p'x + b'v - \lambda v_1$$

をシンプレックス表に作る。

$$(8) \begin{array}{c} u \\ y \\ y_1 \\ F_c \\ F_1 \end{array} \left(\begin{array}{cc|cccccc} 1 & \lambda & u & v & v_1 & x & y & y_1 & F \\ \hline p & & I & -A' & c & -C & & & \\ b & & & & & A & I & & \\ -1 & & & & & -c' & & 1 & \\ \hline 0 & & & -b' & & p' & & & 1 \\ 0 & & & & 1 & & & & 1 \end{array} \right)$$

F_c と F_1 の行は (7) を λ に関し独立な項と従属する項に分けたもの。

問題(4)は独立に解くのではなく、初め $\lambda=0$ と考えてタブロー(8)の(4)に当る部分でピボット選択をし、消去は全タブローに対して行なう。ただし、 y_1 の行で most negative な要素に対する主変数 x_s を基底に入れるステップを行なった直後に、このステップで追い出された変数に対する双対変数を基底に入れ、基底に入れた x_0 に対する双対変数を追出す、双対なシンプレックス計算をしておく。これは、(6)の第4式を満たすタブローにしておくためである。

(4)の解 x^0 が決ったら、 $\lambda_0 = c'x^0$ に対する(5)の解を求めてから、以下 λ を減少してゆく。このQPの計算では、van de Panne-Whinston のシンプレックス法と双対法 (Oper. Res. Quart. 15 (1964), 355-88) を使っている。パラメトリック LP で双対法が要るのと同様、簡単な数値例も添えてある。

(真鍋龍太郎)