

文 献 抄 録

Timperi, G. : Una Rappresentazione Matriciale delle Reti di Telecomunicazioni, Note Recensioni Notizie, Vol. 13 (1964), pp. 55-70.

通信網にたいして1つの行列表示を与え、この行列にある計算方法を定義すれば、通信網の特性がいろいろ導びかれることを示している。それらの特性には、node間を結ぶ可能な path の数、最小数の node を経由する path, 2 node間を結ぶ shortest spanning subtree などが含まれる。

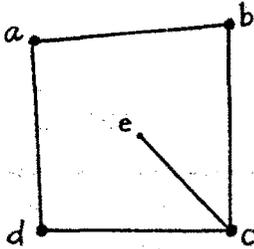


図 1

図1の通信網について、行列表示と要素間の計算方法を示そう。x行y列の要素 R_{xy} が

$$R_{xy} = \begin{cases} 1, & x=y \text{ のとき} \\ y, & x \neq y \text{ で } xy \text{ 間に link のあるとき} \\ 0, & x \neq y \text{ で } xy \text{ 間に link のないとき} \end{cases}$$

で与えられる正方行列を R とすると、図1にたいして

$$R = \begin{matrix} & \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{pmatrix} 1 & b & 0 & d & 0 \\ a & 1 & c & 0 & 0 \\ 0 & b & 1 & d & e \\ a & 0 & c & 1 & 0 \\ 0 & 0 & c & 0 & 1 \end{pmatrix} \end{matrix}$$

をうる。この R は通信網の情報を完全に含むもので、これが所要の行列表示である。いま要素間の積および和を

$$\text{積: } \begin{cases} xy = x & x=0, a, b, c, \dots & y=1, a, b, c, \dots \\ 1y = 0 & y=0, a, b, c, \dots \\ 11 = 1 \\ x0 = 0x & x=1, 0, a, b, c, \dots \end{cases}$$

$$\text{和: } \begin{cases} x+x = x \\ x+y = y+x \\ x+1 = 1 \\ x+0 = x \end{cases} \quad y, x = a, b, c, \dots$$

で定義する。すると、これから通信網の特性を表わす各種の行列が導びかれる。2例をあげよう。いま、その要素が

$$R^2_{xy} = \sum_k R_{xk} R_{yk} \quad k = a, b, c, \dots$$

で与えられる正方行列を R^2 とし、一般に R^n を R^{n-1} と R の積で定義する。すると、 $R^{P+1} = R^P$ なる P が存在し、図1では

$$R^4 = R^5 = \begin{matrix} & \begin{matrix} a & b & c & d & e \end{matrix} \\ \begin{matrix} a \\ b \\ c \\ d \\ e \end{matrix} & \begin{pmatrix} 1 & b+d & b+d & b+d & b+d \\ a+c & 1 & a+c & a+c & a+c \\ b+d & b+d & 1 & b+d & e \\ a+c & a+c & a+c & 1 & a+c \\ c & c & c & c & 1 \end{pmatrix} \end{matrix}$$

となる。このときの x 行 y 列の要素 R^P_{xy} は、node x から node y へ到るために経由する x の次の node を示している。

つぎに、その要素が

$$\begin{cases} R'_{xy} = R_{xy}, & R'_{xy} \neq 0 \text{ のとき} \\ R'_{xy} = \sum_k R_{xk} R_{ky}, & R_{xy} = 0 \text{ のとき} \end{cases}$$

で与えられる正方行列を R' とし、

$$\begin{cases} R''_{xy} = R'_{xy}, & R'_{xy} \neq 0 \text{ のとき} \\ R''_{xy} = \sum_k R'_{xk} R'_{ky}, & R'_{xy} = 0 \text{ のとき} \end{cases}$$

のようにして $R^{(n)}$ をつぎつぎと定義する。すると、すべての要素が0でない行列がついにえられる。図2の通信網では

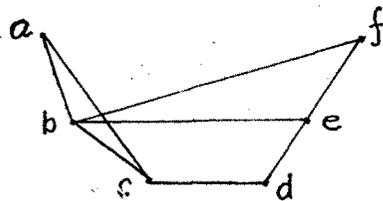


図 2

	a	b	c	d	e	f
a	1	b	c	c	b	b
b	a	1	c	c + e	e	f
c	a	b	1	d	b + d	b
d	c	c + e	c	1	e	e
e	b	b	b + d	d	1	f
f	b	b	b	e	e	1

となる。x行y列の要素は、最小数の node で node x から node y に到るための x の次の node を示している。(中村義作)

Kauffmann, A. et R. Cruon : ETUDE DE LA SENSIBILITE EN PROGRAMMATION DYNAMIQUE : POLITIQUES k-OPTIMALES EN AVENIR CERTAIN. *Revue Française de Recherche Opérationnelle Tom 8, No. 3 (1964) pp. 293-302.*

ある点から、いくつかの分岐を通して目的地に到達する、最も時間のかからない道、次に時間のかからない道、一般に、k番目に時間のかからない道を求める方法としては Bellman-Kalaba によってダイナミック・プログラミングによるものが示されている。本論文も、やはりダイナミック・プログラミングによる方法を示しているが、Bellman-Kalaba の漸化式が、k番目の最適を求める操作をふくむのに対し、ここでは、(第1番目の)最適を求める操作のみをふくむような漸化式が与えられている。

有限グラフ (X, Γ) 上点 x_0 ∈ X_0 ⊂ X を出発して径路 {x_1, x_2, ..., x_N} をたどって目的地 {x_N} に到るこれに次のような値が対応するものとする。

$$\begin{aligned} \nu_{0,N}(x_0, x_1, \dots, x_N) &= \nu_1(x_0, x_1) \\ &+ \nu_2(x_1, x_2) + \dots + \nu_n(x_{n-1}, x_n) + \dots \\ &+ \nu_N(x_{N-1}, x_N) \end{aligned}$$

ただし

$$x_n \in \Gamma(x_{n-1}) \quad n = 1, 2, \dots, N.$$

このとき、この値が第k番目の最適値(最大および最小)をとるような径路を求める問題を考える。そのため、第k番目の最適値を未知関数として次のように定義する。

$$\begin{aligned} f_{0,N}^{(k)}(x_0) &= OPT^{(k)} \nu_{0,N}(x_0, x_1, \dots, x_N) \\ x_n &\in \Gamma(x_{n-1}), \quad n = 1, \dots, N \end{aligned}$$

ただしここに OPT^{(k)} は k番目の最適値をとる演算を示す。これに対して、Bellman-Kalaba は

$$\begin{aligned} f_{N-1,N}^{(i)}(x_{N-1}) &= OPT^{(i)} \nu_N(x_{N-1}, x_N) \\ x_N &\in \Gamma(x_{N-1}) \end{aligned}$$

$$\begin{aligned} f_{n,N}^{(i)}(x_n) &= OPT^{(i)} [\nu_{n+1}(x_n, x_{n+1}) \\ &+ f_{n+1,N}^{(j)}(x_{n+1})] \\ x_{n+1} &\in \Gamma_{n+1}(x_n), \\ j &\in \{1, 2, \dots, i\}, \quad i = 1, 2, \dots, k, \\ n &= N-2, N-3, \\ &\dots, 0 \end{aligned}$$

という漸化式を与えているが、これは暗々裡に次のような、k-番目の最適性の原理を使っていることになる。

ω_{0,N} = (x_0, x_1, ..., x_N) が k-最適な1つの政策であるとき、そのうちの N-n+1 個の成分からなる部分政策 ω_{n,N} = (x_n, x_{n+1}, ..., x_N) は n=N の x_n に関して i-最適である。ただし i ≤ k。

これに対して、新しい、次のような漸化式を使うこともできる。

$$\begin{aligned} f_{N,N}^{(i)} &= 0, \quad \forall x_N; \\ f_{N,N}^{(i)}(x_N) &= \pm \infty, \quad \forall x_N, \quad i = 2, 3, \dots, k; \\ &+ \sim OPT : MIN \\ &- \sim OPT : MAX \end{aligned}$$

$$\begin{aligned} j^{(i)}(x_n, x_{n+1}) &= 1, \quad \forall x_n, x_{n+1}; \\ n &= 0, 1, \dots, N-1. \end{aligned}$$

$$\begin{aligned} f_{n,N}^{(i)}(x_n) &= OPT [\nu_{n+1}(x_n, x_{n+1}) \\ &+ f_{n+1,N}^{(j)}(x_{n+1})] \\ x_{n+1} &\in \Gamma_{n+1}(x_n) \end{aligned}$$

ここに

$$\alpha = j_n^{(i)}(x_n, x_{n+1}).$$

よ

$$\begin{aligned} \nu_{n+1}(x_n, x_{n+1}) + f_{n+1,N}^{(j)}(x_{n+1}) &\geq f_{n,N}^{(i)}(x_n) \\ &> \sim OPT : MIN \\ &< \sim OPT : MAX \end{aligned}$$

のとき、

$$\begin{aligned} j_n^{(i+1)}(x_n, x_{n+1}) &= j_n^{(i)}(x_n, x_{n+1}), \\ \nu_{n+1}(x_n, x_{n+1}) + f_{n+1,N}^{(j)}(x_{n+1}) &= f_{n,N}^{(i)}(x_n) \end{aligned}$$

のとき、

$$j_n^{(i+1)}(x_n, x_{n+1}) = j_n^{(i)}(x_n, x_{n+1}) + 1$$

すなわち、漸化式に k-最適を求める操作が含まれていないので、この方法は、電子計算機で計算するのに適していると述べ、更に簡単な数値例が与えられている。(柳井 浩)

Gomory, R. E. & T. C. Hu : Synthesis of a

Communication Network. Jour. Soc. Indust. Appl. Math., Vol. 12, No. 2 (1964), pp. 348-369.

n 箇の node からなる network で各 node 間における通信の需要が時間とともに変るとき最小費用の network を L.P. を用いて構成する方法を述べている。いま, network が m 箇の arc からなっているとすると, 任意の network は 1 つの m -vector Y で表示できる。時点 t における requirement $R_{p,q}(t)$, $p, q \in \text{node の set}$, を満たす feasible network N^t のすべての集合は 1 つの convex (unbounded) polyhedron を作るからその端点を N_i^t とすると問題は次のような L.P. に定式化される。

$$Y \geq \sum_i \lambda_i^t N_i^t, \quad t = t_1, \dots, t_s \quad (1)$$

$$1 \leq \sum \lambda_i^t,$$

のもとで $C \cdot Y$ を最小にする Y と $\lambda_i^t \geq 0$ を求めよ。ただし, t_1, \dots, t_s は requirement が与えられる時点, C は各 arc の単位容量当りの cost を表わす 1 つの m -vector である。

(1) は $m(s+1)$ の row と非常に多くの列 (N_i^t に対応) からなる L.P. となるからこれを次のように reduce する。

L.P. の双対定理から, ある Y に対して λ_i^t の解が存在する必要十分条件は $H \cdot (N_i^k, 1) \geq 0$ を満足するすべての $m+1$ vector $\Pi = (\Pi_1, -1)$ について $\Pi \cdot (Y, 1) \geq 0$ となることである。ここに Π_1 は非負の m -vector である。このような vector Π

の集合も 1 つの convex (unbounded) polyhedron を作るから, その端点を N_1^k, \dots, N_q^k とすると問題は以下のように書改められる。

$$N_i^k \cdot (Y, 1) \geq 0, \quad i = 1, \dots, q(t);$$

$$t = t_1, \dots, t_s \quad (2)$$

の条件の下で $C \cdot Y \rightarrow \min$

問題(2)に含まれる変数は $Y = (y_1, \dots, y_m)$ の m 箇のみになるが不等式の数は N_i^k の 1 つ 1 つに応じてきわめて数多くなる。(2) を解く実際の手順として dual simplex 法と primal simplex 法について述べられているがここでは前者についてのみ簡単に紹介する。

Y を 1 つの dual feasible (simplex 基準がすべて非負) sol. として, ある時点 $t = t_k$ について次の L.P. を (λ_i^k について) 解く。

$$\text{Maximize } \theta = \sum \lambda_{ik},$$

$$\text{subject to } Y \geq \sum \lambda_i^k N_i^k \quad (3)$$

もし, $\theta \geq 1$ となれば(1)から Y は (primal) feasible となるから最適解がある。もし, $\theta_{max} < 1$ であれば (1) を満たす λ_i^k が存在しないのであるからさきに述べたように

$$\Pi \cdot (Y, 1) < 0,$$

$$\text{および } \Pi \cdot (N_i^k, 1) \geq 0 \quad \text{all } i$$

であるような Π が存在し, dual simplex 表で新しい不等式条件として $\Pi \cdot (Y, 1) < 0$ を加えてゆけばよい。 N_i^k は最初からすべて与えておく必要はなく, (3) を解くときつぎつぎに generate され, そのため L.P. の表をあまり大きくしないですむ。

(倉橋和夫)

AGIFORS 紹介 (3)

Kolner, T.K., UAL : A LOGISTICS CONCEPT OF ROTABLE SPARE PARTS PRODUCTION & INVENTORY, AGIFORS, 1961.

エアラインで言うロータブルパーツとは、故障し機体からとりおろされても廃棄されずに、修理、オーバーホール後、再び使用可能になる部品のこと、高価なものが多い。空港でそのストアにある予備部品と交換された故障部品はメインベースに送られ、ある待ち時間後工場での修理またはオーバーホール、テストを経て、メインストアに入り、各空港からのオーダーにより現地へ空輸されてそのストアに入り、閉回路を構成する。待ち行列の問題として取り扱うのが普通であるが、在庫切れ率は修理能力と補給能力との関数であり、補給時間はオーダーに要する時間とメインストアでの待ち時間と空輸時間との和であり、メインストアでの待ち時間は修理時間、部品需要率、予備部品数の関数であるというように、パラメーターが密接に関係し合っている。そこで、コスト最小を目的とし、

1 考えられる最長の平均サービス時間 T_s (故障部品の輸送、修理、テスト時間) をとり、予備部品数 M_s を1つずつ増し、メインストアでの補給オーダーの待ち時間 E を計算する。

2 各 E に対し、空港の在庫及び在庫切れコストが最小になるように予備部品の配置を計算する。

3 M_s の各々に対し総コスト C_i を計算し、 C_i を最小にする M_s を選ぶ。

4 T_s を小さくして、その時のプロダクションコスト C_p を計算しながら 1, 2, 3 を繰り返す。そして C_i を最小にする C_p と T_s を選ぶ。

という方法で予備部品数と各空港への部品配分を決定した。

Sandiford, J.D., TCA : A THEORY OF OPTIMUM ALLOTMENT INCLUDING COST OF STOCK-OUT, AGIFORS, 1961.

予備部品数を決定する際に最も良く使われるのはアーランの損失式である。在庫切れの場合には緊急に空輸するか、故障機を客を乗せずにフェリーするかの処置をとるので、即時式なのである。ところで呼損率はポリシーとして決めるのが通常であるが、ここではコストを考慮して呼損率を決定する方法をとった。すなわち、在庫切れによる損式の期待値と

在庫費用 (原価を含む) の和が最小になるように呼損率を定め、それから部品保有数を求めた。実用には、最適呼損率を求めるグラフと呼損率から部品保有数を求めるグラフを準備した。大部分の部品の場合、最適数はパラメーターの誤差には比較的鈍感であり、特に需要率の小さいところではこの傾向が顕しいので、コスト計算はかなりラフなもので十分である場合が多い。

Napoli, Joseph C., UAL : DETERMINING STATION STOCKOUT RATES FOR ROTABLE SPARE PARTS IN AN AIRLINE SYSTEM, AGIFORS, 1961.

ロータブルパーツの在庫切れ確率を計算するのに、待ち行列を使わず、基本的確率の考えにより式の誘導を試みた。部品の需要は平均値 λ のポアソン分布がデータから確かめられている。補給時間は平均値 T_R の一般分布であり、ある空港には一定数 M の予備部品をおき、使用すれば直ちに基地に補給をオーダーし、オーダーは全て満足されねばならない。予備部品はその空港だけで使用する。部品の需要がポアソン分布であれば、補給部品の到着は、補給時間の分布にかかわらず、ポアソン分布になることが証明できる。補給のオーダーをしたがまだ空港に到着しない部品の数を m とすれば、 $E(m) = \lambda T_R$ が証明できる。ところで m もポアソン分布であることがわかるので、

$$P(m) = \frac{e^{-E(m)} [E(m)]^m}{m!} = \frac{e^{-\lambda T_R} (\lambda T_R)^m}{m!}$$

$m < M$ の確率は

$$P(m < M) = \sum_{m=0}^{M-1} \frac{e^{-\lambda T_R} (\lambda T_R)^m}{m!}$$

となるので在庫切れの確率 P_s は

$$P_s = 1 - P(m < M) = 1 - \sum_{m=0}^{M-1} \frac{e^{-\lambda T_R} (\lambda T_R)^m}{m!}$$

で与えられる。

Stahel, H., SW : MONTE CARLO SIMULATION ON ENGINE FAILURES, AGIFORS, 1961.

予備エンジンの保有数と各地空港への配分を決定するに際し、各地空港でのエンジン取御し率を求めなければならない。そのために、飛行時間に対する

エンジン取御しの分布をもとにして、モンテカルロシミュレーションを行った。一機についている4個のエンジンの寿命を分布からランダムに選び、ルートをスケジュールに従って飛行させ、飛行時間が、あるエンジンの寿命に達したなら、直前の空港に引返すか、目的地まで飛行を続けるか、代替空港へ飛行するかポリシーに従って空港を選び、そこで新エンジンに交換する。新エンジンの寿命を求め、再び飛行を続けるという手順をくり返す。オーバーホール間隔 (T.B.O. Time Between Overhaul, この場合は1600時間) までもったエンジンは主基地で取御す。このようにして、各地でのエンジン取御し率の予測を行った。

Jeannot, J.P., TCA : AIRCRAFT ENGINE LIFE CYCLE SIMULATION, AGIFORS, 1961.

航空機エンジンは、それを一つの部品とみると、最も高価なロークプルパーツとなる。したがって、できるだけ精密に予備台数を決めたいので、シミュ

レーションを行った。メインルーティンは一日ずつ日を進めながら、エンジンを飛ばすことである。エンジン取御し、または次の日にも使用可能であるかを毎日検討する。取御し的时候は、制限時間によるものであれば、オーバーホール、または部分オーバーホールか点検を行う。故障による取御し的时候は、基地に空輸し、工場に入る日を決め、修理かオーバーホールかを決める。さらにサービサブルになる日を求める。エンジン取御し的时候には、スเปアプールから要求に合う交換エンジンを選定する。予備水準が少なくなったときには工場の作業を急がせる。故障によるエンジン取御しの判定には、過去のデータをもとにしたワイブル分布を使用した。結果は、修理オーバーホール生産計画、工場負荷、予備台数が予測でき、工場マンパワー計画にも使用できる。特に新機種導入頭初で平衡状態に達していない場合や、フリートのリタイヤ計画にも有効である。(中山陸朗)