整数/組合せ計画法の現状（その4）

巡回セールスマン問題

整数計画法研究部会　山本芳嗣

1. 問　題

これまでの報告で重ねて指摘されたように，一般的な整数計画問題に対して汎用性のある強力な解法は今のところ存在しないだけでなく，その本質的なむずかしさが徐々に納得されつつある．そして，いきおい研究は特殊な構造をもった問題へと向かっている．その中で巡回セールスマン問題については1970年以降の研究に興味深いものがある．本報告では1970年以降の研究を中心に，それ以前の成果をも振り返りながら話を進めたいと思う．

巡回セールスマン問題とは，都市 i から都市 j へ行くのに必要なコストが c_{ij} で与えられているとき，すべての都市をちょうど一度訪れなければならないという条件のもとで，旅行に必要なコストを最小にする訪問の順序を求める問題である．つまり，n（都市の数）個の頂点とコスト c_{ij} を付加された $n(n-1)$ 本の枝とから成るグラフ（図 1.1）で，最小コストの巡回路（各頂点をちょうど一度通る閉路）を求める問題である．

したがって巡回路のどの性質に注目するかが解法上の1つの要点となる．

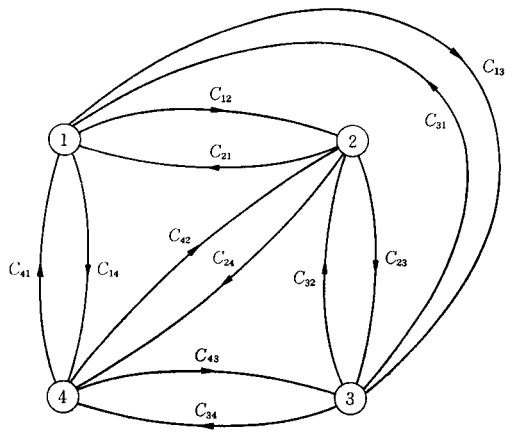


図 1.1　対象とするグラフ

$n(n-1)$ 次元のベクトル $x=(x_{12}, x_{13}, \cdots, x_{n,n-1})$ に対して，枝集合 $S(x)$ を，

$$S(x)=\{(i,j) \mid x_{ij} \neq 0\}$$

と定義すると，巡回セールスマン問題 P_S は，

$$\text{最小化} \quad \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} c_{ij} x_{ij} \tag{1.1}$$

$$\left.\begin{aligned} &\text{条　件} \quad \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij}=1 \quad (i=1,2,\cdots,n) \\ &\qquad\quad \sum_{\substack{i=1 \\ i \neq j}}^{n} x_{ij}=1 \quad (j=1,2,\cdots,n) \end{aligned}\right\} \tag{1.2}$$

$$\text{枝集合 } S(x) \text{ の作るグラフは連結} \tag{1.3}$$

$$x_{ij} \text{ は } 0 \text{ あるいは } 1 \ (i,j=1,2,\cdots,n) \tag{1.4}$$

と書ける．この問題から条件 (1.3) を取り除くと，よく知られた割当問題となり，条件 (1.2) を，

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} (x_{ij}+x_{ji}) \geqq 1 \quad (i=1,2,\cdots,n)$$

で置きかえると，（$c_{ij}>0$ のもとで）最小コストの木（3節参照）を求める問題となる．したがって，条件 (1.2) に注目するか，条件 (1.3) に注目するかで大きく2種の方法を考えることができる．

2. 割当問題を用いた算法

割当問題の実行可能解 x について $S(x)$ の作るグラフは一般に図 2.1 のように部分巡回路（長さ n 未満の閉路）がいくつか集まった非連結なグラフとなる．したがって割当問題に部分巡回路を含まない条件

$$\sum_{(i,j) \in C} x_{ij} \leqq |C|-1 \quad \begin{pmatrix} \text{長さ } n \text{ 未満のすべて} \\ \text{の閉路 } C \text{ について} \end{pmatrix} \tag{2.1}$$

あるいは連結性の条件

$$\sum_{i \in V} \sum_{j \in \bar{V}} x_{ij} \geqq 1 \quad \begin{pmatrix} \phi \neq V \subset \{1,2,\cdots,n\} \\ \text{なるすべての頂点集} \\ \text{合 } V \text{ について} \end{pmatrix} \tag{2.2}$$

を付け加えることによって巡回セールスマン問題を表わすことができる．ここで，$|C|$ は閉路 C の長さを表わす．これを一般的な解法で解けばよいことになるが，条

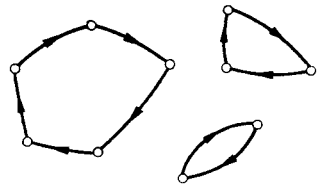


図 2.1 部分巡回路

件(2.1), (2.2)はいずれもその個数がぼう大なうえ，割当問題のように容易に解ける問題ではない．したがって，分枝限定法で部分巡回路を排除するように分枝を行ないながら，繰り返し割当問題を解く算法が考えられる．今，$(1,2),(2,3),\cdots,(k-1,k),(k,1)$ を割当問題を解いて得られた部分巡回路とすると，全都市を訪れる巡回路では $x_{12}=0, x_{23}=0, \cdots, x_{k-1,k}=0, x_{k,1}=0$ のいずれかが成り立つから，

$$P_1 : x_{12}=0$$
$$P_2 : x_{23}=0$$
$$\vdots$$
$$P_k : x_{k1}=0$$

なる k 個の部分問題に分枝する方法が考えられる．

この分枝によって作られた部分問題の実行可能領域は互いに排反ではないが，

$$P_1 : x_{12}=0$$
$$P_2 : x_{12}=1, x_{23}=0$$
$$\vdots$$
$$P_k : x_{12}=x_{23}=\cdots=x_{k-1,k}=1, x_{k,1}=0$$

とすれば排反にできる[3]．また，巡回路であるためには，$1,2,\cdots,k$ の内の少なくとも1つの頂点から $1,2,\cdots,k$ 以外の頂点へと路が延びていなければならないから

$$P_1 : x_{1j}=0 \quad (j=1,2,\cdots,k,\ j\neq 1)$$
$$P_2 : x_{2j}=0 \quad (j=1,2,\cdots,k,\ j\neq 2)$$
$$\vdots$$
$$P_k : x_{kj}=0 \quad (j=1,2,\cdots,k,\ j\neq k)$$

と分枝することもでき，この分枝法は前の2つの方法と比べてより多くの部分巡回路を排除できる[3]．さらに，

$$P_1 : x_{1j}=0 \quad (j=1,2,\cdots,k,\ j\neq 1)$$
$$P_2 : x_{1l}=0 \quad (l\neq 1,2,\cdots,k)$$
$$\quad x_{2j}=0 \quad (j=1,2,\cdots,k,\ j\neq 2)$$
$$\vdots$$
$$P_k : x_{1l}=x_{2l}=\cdots=x_{k-1,l}=0 \quad (l\neq 1,2,\cdots,k)$$
$$\quad x_{kj}=0 \quad (j=1,2,\cdots,k,\ j\neq k)$$

とすれば，やはりこれら k 個の部分問題の実行可能領域を排反にできる[10]．

すべての i, j について $c_{ij}=c_{ji}$ なる問題をとくに対称問題といい，この問題についても以上の方法をそのまま用いることができるが，対称性により $i<j$ について変数 x_{ij} を用意すれば十分で，条件(1.2), (2.2)はそれぞれ

$$\sum_{i<j} x_{ij}+\sum_{j<k} x_{jk}=2 \quad (j=1,2,\cdots,n) \tag{2.3}$$

$$\sum_{(i,j)\in(V,\bar{V})} x_{ij}\geqq 2 \tag{2.4}$$

に置きかえることができる．ここで，$(i,j)\in(V,\bar{V})$ は $i\in V, j\in\bar{V}$ あるいは $i\in\bar{V}, j\in V$ のいずれか一方が成り立つことを示すものとする．Miliotis [23, 24] は対称問題を，

0. (1.1), (1.4), (2.3)を解く．
1. 条件(2.4)の中で，その解によって満たされていない制約を捜す．
2. その制約を付け加えた問題を解き，1.へもどる．

という手順で解いているが，手順0と2で整数解を得るために Gomory カットを用いた方法では，0～100の乱数でコストを与えた90都市の問題2題を CDC 7600 を用いて，4秒足らずで解いている．しかも，報告されている17題のいずれについても必要としたカットの本数は30本未満であることは注目に値する．また条件(1.4)(2.3)を満たす解はb-マッチングとよばれ，Edmonds-Johnson [9]によって問題(1.1), (1.4), (2.3)を解く高々 $O(n^4)$ の手間の算法が提案されている．Bellmore-Malone [3] はこの算法を用いて UNIVAC 1108 で30都市の乱数コストの問題を5題，平均321.6秒で解いたと報告している．

3. 最小コストの木と巡路

話を簡単にするため，しばらく対称問題だけを扱うことにする．われわれが前節で扱ってきた問題では，セールスマンはすべての都市を巡った後，出発した都市にもどって来ることになっていたが，都市1から都市 n へ向かって，すべての都市をちょうど一度通過してゆく路(これを巡路という)を求める問題 P_S'

$$\text{最小化} \quad \sum_{i<j} c_{ij}x_{ij} \tag{3.1}$$

$$\left.\begin{array}{ll} \text{条件} & \displaystyle\sum_{i<j} x_{ij}+\sum_{j<k} x_{jk}=2 \\ & \qquad (j=2,3,\cdots,n-1) \\ & \displaystyle\sum_{j=2}^{n-1} x_{1j}=1 \\ & \displaystyle\sum_{j=2}^{n-1} x_{jn}=1 \end{array}\right\} \tag{3.2}$$

$$\sum_{(i,j)\in(V,\bar{V})} x_{ij}\geqq 1 \tag{3.3}$$

を考えよう．連結で閉路を含まず，すべての頂点をつなぎあわせている部分グラフ T をそのグラフの木(極大木)とよび，そのコスト $c(T)$ を木に属する枝のコストの総和とする．また，$C(T,(p,q))$ を木 T に枝 (p,q) を付け加えたときにできる閉路の枝集合，$D(T,(s,t))$ を木 T

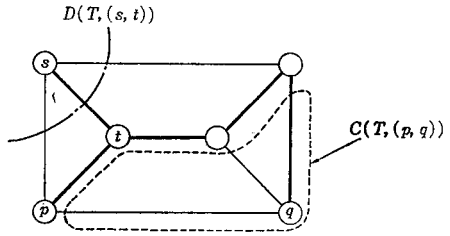


図 3.1 太線を木Tとすると，横に破線の引かれた閉路が$C(T,(p,q))$．一点鎖線の交わる枝集合が$D(T,(s,t))$である．

から枝(s,t)を取り除いたときにできる2つの部分木を結ぶ枝集合とすると（図3.1），Tが最小コストの木であることと，

(i) Tに属さない任意の枝(p,q)について，

$$c_{pq}=\max\{c_{ij}\mid (i,j)\in C(T,(p,q))\},$$

(ii) Tに属する任意の枝(s,t)について，

$$c_{st}=\min\{c_{ij}\mid (i,j)\in D(T,(s,t))\},$$

のそれぞれが同値であることが知られている．これらの性質から最小コストの木を求めるためのつぎの2つの算法を導くことができる．

Kruskal の算法[21]

0. $T=\phi$, $F=E$(グラフの枝集合)とする．
1. $c_{pq}=\min\{c_{ij}\mid (i,j)\in F\}$なる枝$(p,q)$を選び，$T\cup(p,q)$が閉路を含まなければ $T=T\cup(p,q)$, $F=F-(p,q)$として2.へ．閉路を含めば $F=F-(p,q)$として1.へ．
2. $|T|=n-1$ となったら終了．そうでなければ1.へ．

Prim-Dijkstra の算法[8, 30]

0. 任意に1つの頂点，たとえば頂点1，を選んで $U=\{1\}$, $T=\phi$ とする．
1. Uに属さない頂点jについて

$$\beta_j=c_{i(j),j}=\min\{c_{ij}\mid i\in U\}$$

を求め，頂点jにラベル$[i(j),\beta_j]$を付ける．
2. $\beta_k=\min\{\beta_j\mid j\notin U\}$

なる頂点kを選び，$U=U\cup\{k\}$, $T=T\cup(i(k),k)$とする．
3. $|U|=n$ となったら終了．そうでなければ4.へ．
4. Uに属さない頂点jで

$$\beta_j>c_{kj}$$

となる頂点jのラベルを$[k,c_{kj}]$に変えて2.へ．

これらの算法の計算効率については[7]に詳しい議論がされている．このように簡単に最小コストの木が求まることと，巡路が1つの木であることを用いると，つぎの分枝限定法による算法を考えることができる．ここで$\delta_i(T)$は頂点iに接続する木Tの枝集合，$d_i(T)$はその本数（頂点iのTに関する次数）を表わし，$\mathscr{P}$は問題の集合とする．

0. $\mathscr{P}=\{P_S'\}$, $u=+\infty$ とする．
1. $\mathscr{P}=\phi$なら終了(uを達成している巡路が最小コストの巡路)．そうでなければ2.へ．
2. $\mathscr{P}$の中から問題Pを選び$\mathscr{P}=\mathscr{P}-\{P\}$とする．
3. 問題Pのコストの下での最小コストの木を作り，

$$c(T)\geqq u$$

なら1.へ．

$$c(T)<u$$

$$d_1(T)=d_n(T)=1,\ d_i(T)=2\quad (i\neq 1,n)$$

なら $u=c(T)$として1.へ．そうなければ，$d_1(T)>1$あるいは$d_n(T)>1$となっているか，$d_i(T)>2$なる頂点iが存在するので，その中の1つの頂点をjとする．$\delta_j(T)$のそれぞれの枝のコストを$+\infty$と修正した$d_j(T)$個の問題を$\mathscr{P}$に付け加えて2.へ．

今，コスト$C=(c_{ij})$のもとでの最小コストの木Tに関する頂点iの次数が$d_i(T)>2$であるとする．このとき頂点iに接続する枝のコストだけを一律に$c'_{ij}=c_{ij}+\pi$と上げると，新しいコスト$C'=(c'_{ij})$のもとでの最小コストの木T'に関する頂点iの次数$d_i(T')$が$d_i(T)$よりも下がることが期待できる．しかも，このようなコストの修正を行なっても巡路間のコストの大小関係は保たれる．実際 $\pi=(\pi_1,\pi_2,\cdots,\pi_n)$によってコストを，

$$c'_{ij}=c_{ij}+\pi_i+\pi_j \tag{3.4}$$

と修正した場合を考えると，$c(H_1)\leqq c(H_2)$ なる2つの巡路 H_1, H_2 について

$$\begin{aligned}c'(H_1)&=\sum_{(i,j)\in H_1}(c_{ij}+\pi_i+\pi_j)\\&=\sum_{(i,j)\in H_1}c_{ij}+(\pi_1+2\sum_{i=2}^{n-1}\pi_i+\pi_n)\\&\leqq\sum_{(i,j)\in H_2}c_{ij}+(\pi_1+2\sum_{i=2}^{n-1}\pi_i+\pi_n)\\&=\sum_{(i,j)\in H_2}(c_{ij}+\pi_i+\pi_j)=c'(H_2)\end{aligned}$$

が成立する．Christofides[5, 6]は以上の性質を用いて，頂点罰金法と名付けた発見的算法を提案している．

0. 十分大きな正の数Mを用いて，

$$\pi_1=\pi_n=M,\ \pi_i=0\ (i\neq 1,n)$$

として(3.4)によって$C'=(c'_{ij})$を作る(この修正によりコストC'のもとでの最小コストの木に関する頂点1, nの次数は必ず1となる)．
1. C'での最小コストの木Tを求める．
2. $d_i(T)=2(i=2,3,\cdots,n-1)$であれば$T$は最小コストの巡路となるので終了．そうでなければ何らかの方法で罰金$\pi=(\pi_2,\pi_3,\cdots,\pi_{n-1})$を決めて

$$c'_{ij}:=c'_{ij}+\pi_i+\pi_j$$

として1.へ．

この算法によってかなり効率良く最適解が得られたと報告されている（罰金をつぎの(3.5)(3.6)によって決めた場合には，10～60都市の乱数コストの問題が33題解かれており，60都市問題3題の平均反復回数と平均時間はCDC 6600で，111回，13.6秒である）[6]が，残念ながら常に最適解を作り出すとは限らないことが示されている[16]．罰金πの決め方としてつぎのようなものが提案されている．

(i) $\gamma>0$ を用いて，

$\pi_i=\gamma(d_i(T)-2)$　$d_i(T)>2$なる頂点iについて

$\pi_i=-\gamma$　　$d_i(T)=1$　$(i\neq 1,n)$　〃

(ii) 反復ごとに $\gamma:=\alpha\cdot\gamma(0<\alpha<1)$として(i)を用いる．

(iii) $d_s(T)>2$ の場合 $\delta_s(T)$の枝(s,t)それぞれについて，

$$\left.\begin{aligned} &c_{i(t)j(t)}=\min\{c_{ij}\mid(i,j)\in D(T,(s,t)),\\ &\qquad i,j\neq s\}\end{aligned}\right\}\quad(3.5)$$

を求め，

$$\left.\begin{aligned}&\pi_s=\min\{c_{i(t)j(t)}-c_{st}\mid(s,t)\in\delta_s(T)\}\end{aligned}\right\}$$

とする．

$d_s(T)=1$ の場合は頂点 s に接続する T に属さない枝(s,t')それぞれについて，

$$\left.\begin{aligned} &c_{i(t')j(t')}=\min\{c_{ij}\mid(i,j)\in C(T,(s,t')),\\ &\qquad i,j\neq s\}\end{aligned}\right\}\quad(3.6)$$

を求め，

$$\left.\begin{aligned}&\pi_s=\max\{c_{i(t')j(t')}-c_{st'}\mid(s,t')\notin T\}\end{aligned}\right\}$$

とする．

脚注

4. 最小コストの“1-木”と巡回路

前節では巡路が1つの木であることを用いて問題 P_S' を解く算法について話してきたが，つぎに同様な考え方を問題 P_S にも応用してみよう．そのためには，巡回路に対して木に相当するものを見つけなければならない．Held-Karp [16, 17] はそのような性質をもつものとして“1-木”を定義している．グラフから1つの頂点，たとえば，頂点1，とそれに接続する枝集合，δ_1，を取り去ったグラフ G_1 を作り，G_1 の木と δ_1 の任意の2本の枝とをあわせたグラフを1-木と定義する(図4.1)．この定義から明らかなように巡回路は1-木である．そのうえ，最小コストの1-木は G_1 の最小コストの木と δ_1 の最もコストの小さい2本の枝をあわせて作ることができる．また，罰金 π を用いて(3.4)によってコストを修正しても

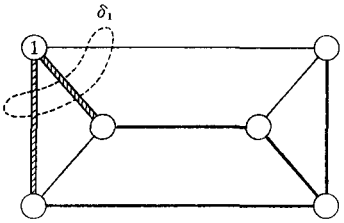


図 4.1 1-木

やはり巡回路のコスト相互の大小関係も変わらない．したがって前節の頂点罰金法を問題 P_S に対しても適用できるが，ここではこの算法を違った角度からながめてみることにする．前節でも触れたように頂点罰金法では必ずしも最適解が得られるとは限らないので，最適解を得るには分枝限定法を用いざるを得ないが，その際，部分問題の最適解の良い（なるべく大きな）下界を求めることは限定操作のうえで重要である．すべての1-木に$1,2,\cdots,k,\cdots,K$ と番号を付け，コスト C のもとでの k 番目の1-木 T_k のコストを c_k，問題 P_S の最適解 H^* のコストを c^* と書くと，コスト $c_{ij}'=c_{ij}+\pi_i+\pi_j$ のもとでのそれぞれのコストは $c_k+\sum_{i=1}^{n}\pi_i d_i(T_k)$，$c^*+2\sum_{i=1}^{n}\pi_i$ となる．したがって H^* が1つの1-木であることから，

$$c^*+2\sum_{i=1}^{n}\pi_i\geqq\min\{c_k+\sum_{i=1}^{n}\pi_i d_i(T_k)\mid k=1,2,\cdots,K\}$$

なる関係が得られる．つまり，$v_{ik}=d_i(T_k)-2$，$\boldsymbol{v}_k=(v_{1k},v_{2k},\cdots,v_{nk})^t$ とすると，

$$\begin{aligned}c^*&\geqq\min\{c_k+\sum_{i=1}^{n}\pi_i d_i(T_k)\mid k=1,2,\cdots,K\}-2\sum_{i=1}^{n}\pi_i\\&=\min\{c_k+\boldsymbol{\pi}^t\boldsymbol{v}_k\mid k=1,2,\cdots,K\}\end{aligned}$$

が任意の $\boldsymbol{\pi}$ について成立する．ここで，

$$w(\boldsymbol{\pi})=\min\{c_k+\boldsymbol{\pi}^t\boldsymbol{v}_k\mid k=1,2,\cdots,K\}\quad(4.1)$$

とすると最良の下界を求めるためには，

$$\text{最大化}\quad w(\boldsymbol{\pi})\qquad(4.2)$$

を解けば良いことになる．(4.1) からわかるようにこの問題は区分的線形凹関数の最大化問題である．図4.2の1本の直線(実際は n 次元超平面)が1本の1-木に対応している．とくに巡回路では $\boldsymbol{v}_k=\boldsymbol{0}$ であるから，$\boldsymbol{\pi}$ 軸に平行な直線が巡回路に対応していることがわかる．図4.2(a)のようにうまく $\boldsymbol{\pi}$ を決めるとコスト $c_{ij}+\pi_i+\pi_j$ での最小コストの1-木が巡回路になる（したがって問題 P_S の最適解）こともあるが，一般には図4.2(b)のようにどのように $\boldsymbol{\pi}$ を決めても巡回路が最小コストの1-木として現われない場合がある．この場合には分枝を行なう必要があるが，それは巡回路に対応する水平な直線を下からおおい隠している何本かの直線を取り去ることに対応している．

$d_S(T)>2$ の場合の π_S の決め方にならえば，

$c_{i(t)j(t)}=\max\{c_{ij}\mid(i,j)\in C(T,(s,t')),i,j\neq s\}$

とすべきである．

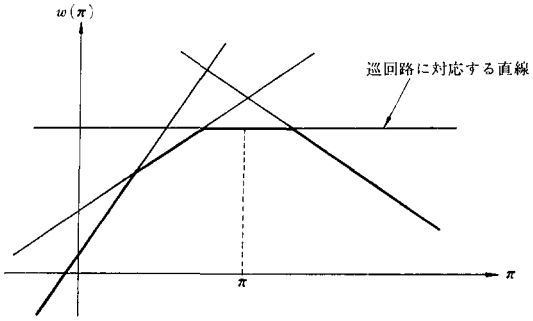


図 4.2(a)　$w(\pi)$

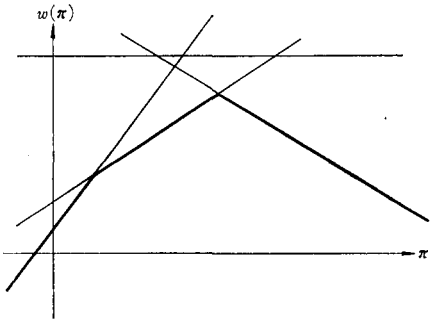


図 4.2(b)　$w(\pi)$

新しい変数 w を用いれば，問題(4.2)は線形計画問題

最大化　w

条　件　$w \leqq c_k + \pi^t v_k \quad (k=1,2,\cdots,K)$

に書き直すことができる．この双対問題

$$\text{最小化} \quad \sum_{k=1}^{K} c_k y_k$$

$$\text{条　件} \quad y_k \geqq 0 \quad (k=1,2,\cdots,K)$$

$$\sum_{k=1}^{K} y_k = 1, \quad \sum_{k=1}^{K} (-v_k) y_k = 0$$

を改訂単体法を用いて解けば，初めにすべての 1-木を知っている必要がないうえ，価格ベクトルを $(\theta, \pi_1, \pi_2, \cdots, \pi_n)$ とすると新しく基底に入れるべき列ベクトル $(1, -v_{1k}, \cdots, -v_{nk})^t$ を求めるには，

$$c_k - \{\theta - \sum_{i=1}^{n} \pi_i v_{ik}\} = \{c_k + \sum_{i=1}^{n} \pi_i d_i(T_k)\} - \{\theta + 2\sum_{i=1}^{n} \pi_i\}$$

を最小にする1-木を求めれば良いことになるが，結局これは $c_{ij}+\pi_i+\pi_j$ のもとでの最小コストの1-木を求めることになる．残念ながらこの算法の収束は遅いことが知られている[16]．

Held-Karp は問題(4.2)を

$$\pi^{\nu+1} = \pi^{\nu} + \varepsilon(\pi^{\nu}, d^{\nu}) \cdot d^{\nu} \qquad (4.3)$$

と反復法で解くことを提案している．彼らが[16]で示した方法は，目的関数 $w(\pi)$ の π^{ν} での上昇方向 d^{ν} を求め区分的線形関数 $w(\pi)$ の区切り目まで d^{ν} 方向に進むよう $\varepsilon(\pi^{\nu}, d^{\nu})$ を求めて，(4.3)で $\pi^{\nu+1}$ を決め，……というものである．T を1-木とし，T の枝 (s,t) と T に属さない枝 (i,j) について，$T^U(i,j)-(s,t)$ が再び 1-木となるとき2本の枝が交換可能であるとよぶことにすると，

$$\varepsilon(\pi^{\nu}, d^{\nu}) = \min\left\{ \varepsilon \;\middle|\; \begin{array}{l} c_{st} + \pi_s^{\nu} + \pi_t^{\nu} \neq c_{ij} + \pi_i^{\nu} + \pi_j^{\nu} \\ \text{あるいは} \\ d_s^{\nu} + d_t^{\nu} \neq d_i^{\nu} + d_j^{\nu} \\ \text{となる，} T\text{に関して交換可能な枝の対} (s,t) \in T \text{と} (i,j) \notin T \text{が存在して，} \end{array} \right.$$

$$\left| \begin{array}{l} c_{st} + \pi_s^{\nu} + \pi_t^{\nu} + \varepsilon(d_s^{\nu} + d_t^{\nu}) \\ \quad = c_{ij} + \pi_i^{\nu} + \pi_j^{\nu} + \varepsilon(d_i^{\nu} + d_j^{\nu}) \\ \text{となる } \varepsilon. \end{array} \right.$$

と選べば良いことが示される．

また彼らはつぎのような反復法を提案している[17]．今，$\bar{w} < \max w(\pi)$ を目標値として設定し，$w(\pi) \geqq \bar{w}$ なる π を見つける問題を考えると，この問題は凸多面体

$$P_{\bar{w}} = \{\pi \mid w(\pi) \geqq \bar{w}\}$$
$$= \{\pi \mid c_k + \pi^t v_k \geqq \bar{w} (k=1,2,\cdots,K)\}$$

の1点を見つける問題となる．Motzkin-Schoenberg [25]，Agmon[1]は一般的な凸多面体

$$A = \{x \mid a_i^t x \leqq b_i \quad (i=1,2,\cdots,m)\}$$

の点を求める反復法として，

0.　初期点 x^0 を与え，$\nu=0$ とする．

1.　x^{ν} が満たしていない制約の内で超平面 $H_i = \{x \mid a_i^t x = b_i\}$ までの距離が最大のものを選び $H_{i(\nu)}$ とする．

2.　$\alpha_{\nu} = (b_{i(\nu)} - a_{i(\nu)}^t x^{\nu}) / \|a_{i(\nu)}\|^2$ として，$0 < \lambda \leqq 2$ を用いて $x^{\nu+1} = x^{\nu} + \lambda \alpha_{\nu} a_{i(\nu)}$ とし $\nu := \nu+1$ として 1. へ．

を提案しており，こうして作り出される点列 $\{x^{\nu}\}$ が，

(i)　$x^{\nu} \neq x^{\nu+1} \quad (\nu=0,1,\cdots)$

(ii)　A の任意の点 y について $\|x^{\nu} - y\| \geqq \|x^{\nu+1} - y\|$ $(\nu=0,1,\cdots)$

なる性質をもつ（このとき点列 $\{x^{\nu}\}$ は A に関して，Féjer 単調であるという）ことを示して彼らの算法の収束を保証している．$P_{\bar{w}}$ の1点を求める問題についても，$k(\pi)$ をコスト $c_{ij}+\pi_i+\pi_j$ のもとでの最小コストの1-木の番号とし，

$$0 < \varepsilon < \lambda^{\nu} \leqq 2$$

$$t^{\nu} = \lambda^{\nu} (\bar{w} - w(\pi^{\nu})) / \|v_k(\pi^{\nu})\|^2$$

として，

$$\pi^{\nu+1} = \pi^{\nu} + t^{\nu} v_k(\pi^{\nu}) \qquad (4.4)$$

によって点列 $\{\pi^{\nu}\}$ を作ると，$\{\pi^{\nu}\}$ は $P_{\bar{w}}$ に関して Féjer 単調であることが示せて(図4.4)，$\{\pi^{\nu}\}$ は有限回で $P_{\bar{w}}$ に入るか，$P_{\bar{w}}$ の境界に収束するかのいずれかが成立する．Held-Karp [17] は(4.4)で $t^{\nu} = \bar{t}$(一定)とし

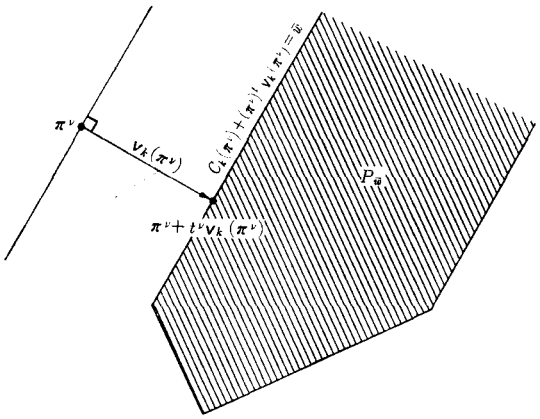


図 4.4　$P_{\bar{w}}$ と点列 $\{\pi_\nu\}$

て条件，サイズの異なった19題の対称問題を解いているが，分枝を行なう前に得られた下界は最適解の値に非常に近く，両者の差がないものが7題あったことが報告されている．その後この算法の改良が提案されており[15, 32]，たとえば[15]にはIBM 360/75を用いて70都市の乱数コストの問題を15題平均78秒で解いたと報告されている．これまで，話を対称問題に限ってきたが，非対称な問題についても条件(1.3)を，

$S(x)$は1-木である

に置きかえることができる．その際

$$\theta(\lambda,\mu)=\min\{c_k+\lambda^t y_k+\mu^t z_k \mid k=1,2,\cdots,K\} \tag{4.5}$$

を最大にする λ, μ を求めれば最良の下界が得られる[2]．ここで，頂点 i から出る(i に入る)1-木 T の枝の本数を $d_i^+(T)$ ($d_i^-(T)$)とすると

$$y_k=(d_1^+(T_k)-1,\cdots,d_n^+(T_k)-1)^t$$
$$z_k=(d_1^-(T_k)-1,\cdots,d_n^-(T_k)-1)^t$$

である．この算法によって UNIVAC 1108 を用いて，60都市の乱数コストの問題が約35秒で解かれている．

本節で述べた算法は Geoffrion[11] のラグランジュ緩和法(第3回報告の双対法参照)の一種であり，ラグランジュ変数 λ, μ を介して目的関数(1.1)に条件(1.2)を組み込んだものが(4.5)で，ラグランジュ変数 π を介入して(1.1)に条件(2.3)を組み込んだものが(4.1)である．そして，いずれについても目的関数に組み込まれずに残された条件「$S(x)$は1-木である」は第3節で述べたように元問題と比べて非常に扱いが容易であることがこれらの算法の効率に寄与している．

5. その他の話題

巡路(巡回路)は各頂点の次数が2あるいは1と定められた木(1-木)であるから，次数が制限された最小コストの木(1-木)を効率良く求めることができれば前節の算法を加速できる見込みがある．Glover-Klingman[12]はつぎの2つの事実を示している．

$$\mathscr{T}(m)=\{T \mid T\text{は木}, d_1(T)=m\}$$

とし，T を $\mathscr{T}(m)$ の中の最小コストの木とする．

(i)
$$c_{pq}-c_{st}=\min\left\{c_{ij}-c_{kl}\,\middle|\,\begin{array}{l}(i,j)\in\delta_1-T\\(k,l)\in C(T,(i,j))\\ -\delta_1\end{array}\right\}$$

とすると，$T'=T\cup(p,q)-(s,t)$ は $\mathscr{T}(m+1)$ の最小コストの木である．

(ii)
$$c_{pq}-c_{st}=\min\left\{c_{ij}-c_{kl}\,\middle|\,\begin{array}{l}(k,l)\in\delta_1(T)\\(i,j)\in D(T,(k,l))\\ -\delta_1\end{array}\right\}$$

とすると，$T'=T\cup(p,q)-(s,t)$ は $\mathscr{T}(m-1)$ の最小コストの木である．

(i) (ii) を用いれば，次数に制限のない最小コストの木から出発して，指定された次数をもつ最小コストの木を作り出すことができる．$\mathscr{T}(m)$ の最小コストの木を求める問題はグラフの枝集合上に定義された2つのマトロイド（グラフ的マトロイドと分割マトロイド）の最小コストの共通基底を求める問題に定式化でき，その問題のための算法もいくつか提案されている[13, 19, 22]．

Murty[26, 27]は巡回路と割当問題の基底解の関係に注目し，

$$\sum_{i=1}^{n} x_{ij}=1 \quad (j=1,2,\cdots,n)$$

$$\sum_{j=1}^{n} x_{ij}=1 \quad (i=1,2,\cdots,n)$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}=\beta$$

の作る制約行列で，変数 $x_{11}, x_{22}, \cdots, x_{nn}, x_{i_1j_1}, x_{i_2j_2}, \cdots, x_{i_nj_n}$ に対応する列が実行可能基底を作るとき，(i_1,j_1), (i_2,j_2), $\cdots$, (i_n,j_n) はそのコストが β 以下の巡回路であることを示している．彼は，あらかじめ指定された列（ここでは変数 $x_{11}, x_{22}, \cdots, x_{nn}$ に対応する列）を含む実行可能基底を求める問題を基本問題と名付け，最小被覆問題(第5回報告参照)を繰り返し解いてゆく，本質的には列挙法的な算法を提案しているが，計算結果は報告されていない．巡回路に対応する0, 1ベクトル $x=(x_{12}, x_{13}, \cdots, x_{n,n-1})$ の集合を X_H，その凸包を ${}_{co}X_H$ と書くことにし，一対の巡回路 H_1, H_2 に対応するベクトルが ${}_{co}X_H$ 上で隣接しているとき，H_1 と H_2 が隣接しているということにする．

$$\mathscr{H}_{12}=\left\{H\,\middle|\,\begin{array}{l}H\text{は巡回路．}H\neq H_1, H\neq H_2.\\H_1\cap H_2\subseteq H\subseteq H_1\cup H_2.\end{array}\right\}$$

とし，$H\in\mathscr{H}_{12}$ に対して，

$$\bar{H}=(H_1\cap H_2)\cup(H_1\cup H_2-H)$$

が巡回路であるとき，H と $\bar{H}$ の組を補巡回路対とよぶ．

Rao[31]は $|\mathscr{H}_{12}|\leqq 1$ なら H_1 と H_2 は隣接しており，H_1 と H_2 が隣接しているなら $\mathscr{H}_{12}$ は補巡回路対を含まない（だからといって $|\mathscr{H}_{12}|\leqq 1$ とは限らない）ことを示しているが，いずれの条件も必要十分ではない．実は一対の巡回路が隣接しているかどうかを判定する問題はNP-完全(第6回報告参照)であることが示されている[29]．ところが，H_1 から始まり H_2 で終わる互いに隣接した巡回路の列の中で最短のものの長さを $d(H_1, H_2)$ とし，$co X_H$ の直径を，

$$\rho(co X_H)=\max d(H_i, H_j)$$

と定義する $\rho(co X_H)\leqq 2$ とであることが示されている[28]．

6. おわりに

本稿では数多い発見的算法についてはほとんど取り上げることができなかったが，これまでに述べた算法を用いて最適解を求める場合にも，発見的算法によって初めに良い実行可能解を知っていることは限定操作のうえで非常に重要であることを付け加えておく．興味のある読者は総合報告[4]，文献表[14, 20]を参照していただきたい．最後に原稿に対し貴重な御助言をくださった整数計画法研究部会の方々に感謝いたします．

参考文献

[1] Agmon, S., "The Relaxation Method for Linear Inequalities", *Canadian J. Math.* **6** (1954) 382-396.

[2] Bazaraa, M. S., Goode, J. J., "The Traveling Salesman: A Duality Approach", *Math. Prog.* **13** (1977) 221-237.

[3] Bellmore, M., Malone, J. C., "Pathology of Traveling Salesman Subtour-Elimination Algorithms", *Operations Research* **19** (1971) 278-307.

[4] Bellmore, M. Nemhauser, G. L., "The Traveling Salesman Problem: A Survey", *Operations Research* **16** (1968) 538-558.

[5] Christofides, N., "The Shortest Hamiltonian Chain of a Graph", *SIAM J. Appl. Math.* **19** (1970) 689-696.

[6] —— Graph Theory: An Algorithmic Approach, (Academic Press, New York, 1975).

[7] Cheriton, D., Tarjan, R .E., "Finding Minimum Spanning Trees", *SIAM J. Comp.* **5** (1976) 724-742.

[8] Dijkstra, E. W., "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik* **1** (1959) 269-271.

[9] Edmonds, J., Johnson, E., "Matching: A Well-Solved Class of Integer Linear Programs" in Combinatorial Structures and Their Applications (Gordon and Breach, New York, 1970).

[10] Garfinkel, R. S., "On Partitioning the Feasible Set in Branch-and-Bound Algorithm for the Asymmetric Traveling-Salesman Problem", *Operations Research* **21** (1973) 340-343.

[11] Geoffrion, A. M., "Lagrangean Relaxation for Integer Programming", *Math. Prog. Study* **2** (1974) 82-114.

[12] Glover, F., Klingman, D., "Finding Minimum Spanning Trees with a Fixed Number of Links at a Node" in Combinatorial Programming: Methods and Applications (D. Reidel, Holland, 1975).

[13] Greene, C., Magnanti, T., "Some Abstract Pivot Algorithm", *SIAM J. Appl. Math.* **29** (1975) 530-539.

[14] Golden, D. L. Magnanti, T. L., "Deterministic Network Optimization: A Bibliography", *Networks* **7** (1977) 149-183.

[15] Hansen, K. H., Krarup, J., "Improvements of the Held-Karp Algorithm for the Symmetric Traveling-Salesman Problem", *Math. Prog.* **7** (1974) 87-96.

[16] Held, M., Karp, R. M., "The Traveling Salesman Problem and Minimum Spanning Trees", *Operations Research* **18** (1970) 1138-1162.

[17] —— "The Traveling Salesman Problem and Minimum Spanning Trees Part II", *Math. Prog.* **1** (1971) 6-25.

[18] Held, M., Wolfe, P., Crowder, H. P., "Validation of Subgradient Optimization", *Math. Prog.* **6** (1974) 62-88.

[19] Iri, M., Tomizawa, N., "An Algorithm for Finding an Optimal 'Independent' Assignment", *J. Operations Research Society of Japan* **19** (1976) 32-57.

[20] Kastning, C. (ed.), Integer Programming and Related Areas: A Classified Bibliography, Lecture Notes in Economics and Mathematical Systems 128 (Springer, 1976).

[21] Kruskal, J. B., "On the Shortest Subtree of a Graph and the Traveling Salesman Problem", *Proceedings of the American Mathematical Society* **2** (1956) 48-50.

[22] Lawler, E. L., "Matroid Intersection Algorithm", *Math. Prog.* **9** (1975) 31-56.

[23] Miliotis, P., "Integer Programming Approaches to the Traveling Salesman Problem", *Math. Prog.* **10** (1976) 367-378.

[24] —— "Using Cutting Planes to Solve the Symmetric Traveling Salesman Problem", *Math. Prog.* **15** (1978) 177-188.

[25] Motzkin, T. S., Schoenberg, I. J., "The Relaxation Method for Linear Inequalities", *Canadian J. Math.* **6** (1954) 393-404.

[26] Murty, K. G., "On the Tours of a Traveling Salesman", *SIAM J. Control* **7** (1969) 122-131.

[27] —— "A Fundamental Problem in Linear Inequalities with Applications to the Traveling Salesman Problem", *Math. Prog.* **2**(1972) 296-308.

[28] Padberg, M. W., Rao, M. R., "The Traveling Salesman Problem and a Class of Polyhedra of Diameter Two", *Math. Prog.* **7**(1974) 32-45.

[29] Papadimitriou, C. H., "The Adjacency Relation on the Traveling Salesman Polytope is NP-Complete", *Math. Prog.* **14**(1978) 312-324.

[30] Prim, R. C., "Shortest Connection Matrix Network and Some Generalization", *Bell System Tech. J.* **36** (1957) 1389-1401.

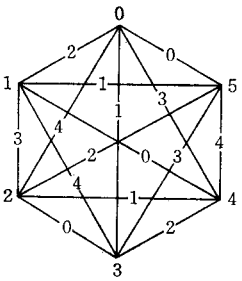
[31] Rao, M. R., "Adjacency of the Traveling Salesman Tours and 0-1 Vertices", *SIAM J. Appl. Math.* **30** (1976) 191-198.

[32] Smith, T. H., Thompson, G. L., "A Lifo Implicit Enumeration Search Algorithm for the Symmetric Traveling Salesman Problem Using Held and Karp's 1-Tree Relaxation", *Annals of Discrete Mathematics* **1** (1977) 479-493.

(やまもと・よしつぐ　慶応義塾大学)

数理パズルを楽しもう (19)

問題　正六角形の6辺と9本の対角線に，図のように，0から4までの5種類の色を塗ります．すると，どの2色を交互にたどっても，6個の頂点を1度ずつ通って元にもどります．正八角形の8辺と20本の対角線についても，0から6までの7種類の色を使って，どの2色も8個の頂点を1度ずつ通るように塗り分けてください．



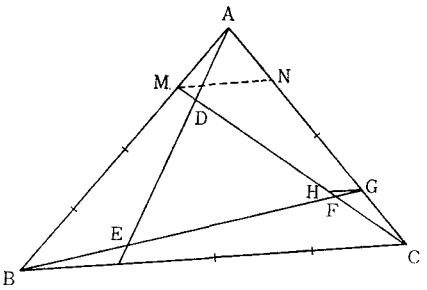
〔4月号(231ページ)の解答〕 三角形の1辺BCに平行に，点Mから線分MN，点Gから線分GHをそれぞれ図のように引く．すると，ABはAMの4倍であるから，

$$MN=BC/4$$

となる．また，CNはCGの3倍であるから，

$$GH=MN/3=BC/12$$

となる．ところで，△FBCと△FGHは相似形であり，点Fから対辺までの高さの比は12：1である．一方，



点Aから辺BCまでの高さは，点Gから辺BCまでの高さの4倍であるから，点Fから辺BCまでの高さの

$$4\times 13/12=13/3 \text{ (倍)}$$

となる．よって，△FBCの面積は△ABCの面積の3/13倍となる．まったく同理由で，△DCAと△EABの面積もそれぞれ△ABCの面積の3/13倍となるから，これらの3つの三角形の面積の和は9/13倍である．よって，残りの部分として与えられる△DEFの面積は，△ABCの面積の4/13倍となる．なお，この方法は一般のn等分についても適用できる〔1〕．

〔1〕 Graham, L. A., *Ingenious Mathematical Problems and Methods*, Dover Pub., New York, 1959.

(中村義作　信州大学工学部)