日本オペレーションズ・リサーチ学会論文誌
Journal of the Operations Research
Society of Japan
Vol. 19, No. 4, December, 1976



# 大規模巡回セールスマン問題の一解法

北海道大学　久保洋・沖野教郎

# A NEW PRACTICAL SOLUTION FOR LARGE-SCALE TRAVELING SALESMAN PROBLEMS

HIROSHI KUBO and NORIO OKINO
*Hokkaido University*

(Received May 6, 1975; Revised October 27, 1975)

**Abstract.** The solution for the traveling salesman problem is of importance because of its many application in the fields of engineering. Although many good solutions have been previously proposed for the traveling salesman problem, they are difficult to apply to practical large scale problems. In this paper we propose a new solution which can solve large scale problems covering more than 100 cities in high speed computation. In experiments using a computer (HITAC 8800 at TOKYO University Computing Center) the solution proposed in this paper succeeded in succeeded in solving 300-city problems in less than ten minutes. In this paper it will be shown that the success of this solution lies in effectively applying the branch and bound technique.

## 1. まえがき

巡回セールスマン問題はＯＲの分野では基本的な問題の一つであり，対称問題についてはHeldとKarp[4]による有効な解法が存在するが，非対称な場合には実用的に百都市以上の問題を解くことのできる解法はまだないようである。従来の巡回セールスマン問題の解法には主としてブランチアンドバウンド法が利用されているが，この方法は下界の設定や分岐の方法如何によって効率に著しい差が生じる。したがって，問題に適した下界の設定や分岐の方法を考えることによって解法をさらに発展させ得る可能性がある。本研究は，このような発想からブランチアンドバウンド法の有効な適用法についての考察を加え，この結果をもとに多数都市問題の厳密解を非常に高速に求め得る非対称な場合の新しい巡回セールスマン問題の解法を導いたので報告する。ＨＩＴＡＣ８８００（東大大型計算機センター）を用いての実験では，計算時間を最大１０分に制限して３桁乱数による非対称３００都市問題を，１０例中６例解くことに成功している。

## 2. ブランチアンドバウンド法の考察

初めにブランチアンドバウンド法（以下B－B法と呼ぶ）の適用法の効率化を高めることを目的として考察を加え，考慮すべき三点について明らかにする。

説明の都合上，B－B法について簡単に記述する。式（1）の目的関数$C_i$を持つ一つの最適化問題を，その変数$t_i$の領域$T_i$を用いて式（2）[1)]のように記述し，拘束条件は$T_i$に含めて考える。

$$(1)\quad C_i^* = C_i(t_i^*) = \min_{t_i \in T_i} C_i(t_i)$$

$$(2)\quad P_i = P(T_i)$$

＊印はこの問題の厳密解を示す。B－B法の処理手続きは，この問題$P_i$を図1に示すようにk個の問題$P_{i1}$，$P_{i2}$，…，$P_{ik}$に分割する式（3）の分岐のアルゴリズムB（Branching）と，各問題$P_i$の下界$v_i$を計算する式（4）の下界のアルゴリズムL（Bounding）の主に二つから構成される。このとき，分割された各問題$P_{ij}$の変数の領域$T_{ij}$は式（5）を満たす必要があり，この条件により$P_i$を解くこととこれらk個の問題を解くこととは等価となる。また下界$v_i$は$P_i$の厳密解$C_i$＊以下であることが必要である。

$$(3)\quad B:\ \{P_{i1}, P_{i2}, \cdots\cdots, P_{ik}\} = B(P_i)$$

$$(4)\quad L:\ v_i = L(P_i) \leq C_i^*$$

$$(5)\quad T_i = \bigcup_{j=1}^{k} T_{ij},\quad T_{ij} \subset T_i,\ j = 1, 2, \cdots, k$$

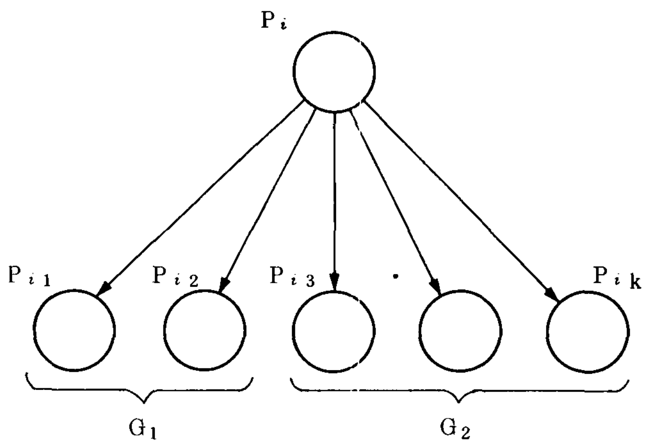



図1　分岐の状態

分岐の機能は，各$P_i$の解$C_i$＊が求まらない場合に分岐を繰返すことにより，最初に与えられた問題$P_o$を式（6）の多数のより簡単な問題の集合に分割することである。これら解く必要のある問題を活性な問題と呼び，これら全てを解くことと$P_o$を解くこととは式（7）より等価である。

$$(6)\quad P_o = \{P_1, P_2, \cdots, P_m\}$$

$$(7)\quad C_o^* = \min\{C_1^*, C_2^*, \cdots, C_m^*\}$$

ある活性な問題$P_i$の解$C_i{}^*$が求まると，これを$P_o$の可能解と呼び，これら可能解の最小のものを上界uとするとu以上の下界$v_i$を持つ問題$P_j$には$P_o$の厳密解$C_o{}^*$が存在しないことが保証される。このように下界の機能は，$C_o{}^*$の存在しない問題の廃棄すなわちこの問題の変数の領域を廃棄することである。

したがって，これら分岐と下界のアルゴリズムを，最初に与えられた問題の変数の大きな領域を廃棄できる可能性を高めるように決めることが，B－B法を適用するさいに最も重要な点であろう。

Ⅰ　分岐の方法について

廃棄の可能性を高めることが解法にとって根本的であるとの観点から，分岐のアルゴリズムは次の二点が充分に考慮されるべきである。

第一には，この廃棄の概念を用いてできる限り大きな変数の領域を廃棄するように分岐の方法を決めるべきである。いま図1に示すように$P_i$を分岐して作られる問題を二つのグループ$G_1$と$G_2$に分けると，これらは下界としてそれぞれのグループ内の問題の下界の最小値を下界として持つ二つの問題と見なすことができる。このとき両方のグループの下界とも大きい方が良いのは明らかであるが，とくに廃棄の可能性を高めるには片方のグループの下界と変数の領域のみを特に大きくすればよい。これを実現するには，同じ$P_i$を分岐するにもできる限り多くの分岐の方法を考え，その中で廃棄のグループの下界が最大となるものを選ぶべきである。

第二には，分岐される問題の各変数の領域の独立性，すなわち式(3)で$P_i$を分岐するとき各$P_{ij}$の変数の領域$T_{ij}$は式(8)を満たすべきである。少なくとも上で述べた二つのグループ間にこの条件が満たされないと，片方の下界を大きくすることはできないであろう。

(8)　$T_{ij_1} \cap T_{ij_2} = \phi \qquad j_1 \neq j_2,\ j_1,\ j_2 = 1,\ 2, \cdots,\ k$

この他に，一回の分岐での問題の個数式(3)のkは，廃棄するための問題を一個とすると残りは少なくすべきである。これはⅡでも述べるが，組合的な要素を除くためである。

Ⅱ　下界の計算法について

下界の計算アルゴリズムは，廃棄の概念からも少しでも大きな下界を与えるものが良い。しかしながら，下界を大きくするには計算時間が漠大となり，極限ではその問題自身を組合せ的な計算で解くことになる。また，下界を小さくすると活性な問題が漠大となり，極限では組合せ的な個数の問題を解くことになる。したがって一個の下界の計算での適当な値と時間が存在する。

一般に組合せ問題の次数が小さいとB－B法より列挙法の方が良いことが知られているが，これは次数が小さいと下界の機能は必要ないことを示している。一方次数が大きくなると，同じ下界のアルゴリズムでは相対的に下界の推定が悪くなり，活性な問題が漠大となる。したがって，次数が大きくなればなる程下界を少しでも大きくすることが必要となり，下界のアルゴリズムは組合せ的な計算を含まなければできる限り大きくすべきである。とくに廃棄の概念から，廃棄する問題の下界をできる限り大きくすることが必要である。

Ⅲ　分岐の戦略について

式(6)の活性な問題の集合の中で次にどの問題を分岐するかの分岐の戦略は，B－B法にとって

非常に重要な問題の一つである。これは常に最小の下界を持つ活性な問題を分岐する方法と，毎回最小の下界から分岐を開始すると，可能解が得られるまで分岐して得られた問題の一つを繰返して分岐してゆく方法などがある。後者は主に早い時間で近似解を得るために使われている。本論ではこれを分岐拘束法と呼ぶが，次の二つの理由からこれを積極的に利用すべきであることを示す。

第一には，分岐した問題を続けて分岐する場合多くのパラメータをそのままか，あるいは若干の修正で使用することができる。したがって分岐拘束法を用いると，毎回これらのパラメータを計算する時間が大幅に減少され，全体の分岐の回数（ノード数と呼ばれる）は増加するが，逆に全体の計算時間は減少する[8]。第二の理由は，廃棄の概念は逆に他の部分を選択すべきことを示しており，この廃棄を続けてゆくと良い可能解が得られることを示唆している。

## 3. 提案する解法とアルゴリズム

巡回セールスマン問題は広く周知の問題なので簡単に定式化を行い，前節のB－B法の考察をもとにした提案する解法の分岐の方法，下界の計算法，分岐の戦略について述べる。またそのアルゴリズムをも示す。

n個の都市があり，それら都市間のコストとして式（9）の$n \times n$マトリクス$D_o$が与えられているとする。さらにこの$D_o$に矛盾なくすべての都市を唯一度ずつ訪問する一巡経路の可能なすべての集合を$T_o$，その要素を式（10）の$t_o$で表わす。ただし，$d_{ij}$は都市$i$から$j$へのコスト，$t_o$の要素の都市対（$i$，$j$）は都市$i$から直接$j$への経路が一巡経路に含まれていることを示す。

（9）　$D_o = [\, d^o_{ij} \,]$ ，　$d^o_{ij} \geq 0$　　$i,\ j = 1,\ 2, \cdots,\ n$

（10）　$t_o = \{(i_1,\ i_2),\ (i_2,\ i_3), \cdots\cdots,\ (i_n,\ i_1)\}$ ，　$t_o \in T_o$

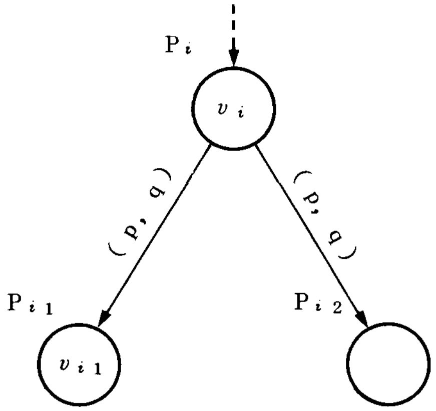
このとき，巡回セールスマン問題は$t_o$を変数，$T_o$を領域とし，式（11）の目的関数$C_o$を持った式（1）と（2）の最適化問題として定式化される。

（11）　$C_o(t_o) = \sum_{(i,\ j) \in t_o} d^o_{ij}$

### 3．1　分岐の方法

前節の観点から，分岐には複数個の分岐と，さらにそれぞれに独立性が必要である。本論では，ある活性な問題を$P_i$とすると，これを任意の都市対（p，q）を一巡経路に必ず含まない問題$P_{i1}$と，必ず含む問題$P_{i2}$に分割する分岐の方法を用いる。この状態を図2に示す。

巡回セールスマン問題は，そのコストマトリクス$D_i$が定まると一巡経路の集合$T_i$が定まり，問題$P_i$が定式化される。最初のn都市問題$P_o$から分岐を繰返して，$P_i$では（$n - \ell$）個の都市対がすでに選ばれていてこの集合を$S_i$，またすでに禁止されている都市対の集合を$R_i$とする。いま$P_o$が（p，q）で分岐されたとすると，これを禁止した問題$P_1$のマトリクス$D_1$は式（12）で，選択した問題$P_2$のマトリクス$D_2$は図3のようにp，qを一個の都市rと考えることができて式（13）でそれぞれ作ることができる。このとき$P_2$の方は，目的関数$C_2$が式（14）[3]となる$n-1$都




$$P_i : \begin{cases} S_i \\ R_i \end{cases}$$

$$P_{i1} : \begin{cases} S_{i1} = S_i \\ R_{i1} = R_i U(p, q) \end{cases}$$

$$P_{i2} : \begin{cases} S_{i2} = S_i U(p, q) \\ R_{i2} = R_i \end{cases}$$

図2　二分岐の方法

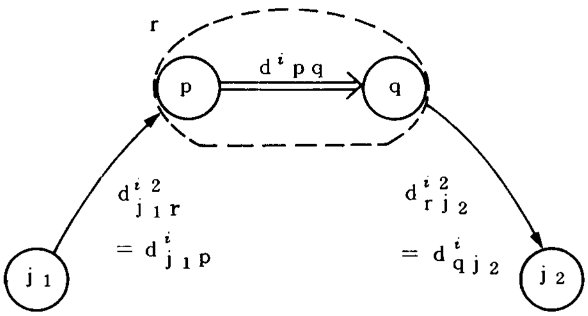



図3　$P_i$ と $P_{i2}$ の関係

市の問題となる。

$$(12)\quad D_1 = [\, d^1_{j_1 j_2} \,] \quad : \quad d^1_{pq} = \infty, \quad d^1_{j_1 j_2} = d^o_{j\ j_2}$$

ただし $j_1, j_2 \neq p, q$

$$(13)\quad D_2 = [\, d^2_{j_1 j_2} \,] \quad : \quad d^2_{j_1 r} = d^o_{j_1 p}, \quad d^2_{r j_2} = d^o_{q j_2},$$

$$d^2_{rr} = \infty, \quad d^2_{j_1 j_2} = d^o_{j_1 j_2} \quad \text{ただし } j_1, j_2 \neq p, q$$

$$(14)\quad C_2(t_2) = d^o_{p\ q} + \sum_{(j_1,\ j_2)} + d^2_{j_1 j_2}$$

したがって上述の$P_i$では，$D_o$に$R_i$の要素すべてについて式(12)を，$S_i$の要素すべてについて式(13)を繰返し適用することにより$\ell \times \ell$マトリクス$D_i$が得られ，$P_i$は式(15)の目的関数$C_i$を持つ$\ell$都市問題となる。このとき$S_i$と$R_i$の要素の適用の順序は無関係となり，$P_i$は$S_i$と

$R_i$より一意に定まる。

$$(15)\quad C_i(t_i) = \sum_{(j_1,\ j_2)\in S_i} d^{o}_{j_1\ j_2} + \sum_{(j_1,\ j_2)\in t_i} d^{i}_{j_1\ j_2}$$

$P_i$を新たな都市対(p，q)で分岐すると，禁止した問題$P_{i1}$と選択した$P_{i2}$では$P_1$, $P_2$のときと同様にマトリクス$D_{i1}$，$D_{i2}$はそれぞれ式(12)と式(13)を$D_i$に適用することにより作ることができる。$S_{i1}$，$R_{i1}$と$S_{i2}$，$R_{i2}$は図2に示す関係が成立し，$T_i$と$T_{i1}$，$T_{i2}$は式(5)と式(8)の条件を満たしている。

$P_i$でどの都市対(p，q)について分岐するかは，廃棄の概念から次節での$P_{i1}$の下界を最大にするように選ばれる。ただし，(p，q)は$D_i$で禁止されていないことが必要である。また$P_i$で$S_i$の要素が(n－2)個あると，$P_i$は2都市問題となって可能解が得られる。$D_i$に禁止を含まなければ，$T_i$, $T_{i1}$，$T_{i2}$の要素の個数はそれぞれ($\ell$－1)！個，($\ell$－2)×($\ell$－2)！個，($\ell$－2)！個となり，廃棄できる可能性のある$T_{i1}$の要素の個数が非常に大きくなっている。

### 3.2　下界の計算法

前節の$P_i$での下界のアルゴリズムは，式(15)の$C_i$から式(4)を満たすできる限り大きな値を得るものが必要である。式(15)の第一項は定数となるので，残りのマトリクス$D_i$の$\ell$都市の巡回セールスマン問題の下界として周知の割当て問題の解を用いる。したがって，この$D_i$の割当て問題を解いた後のマトリクスを$\overline{D}_i$，そのコストを$\overline{C}_i$，解を$\overline{t}_i$で表わすと，下界$v_i$は式(16)となる。

$$(16)\quad v_i = e_i + \overline{C}_i \leq C^*_i \quad ただし \quad e_i = \sum_{(j_1,\ j_2)\in S_i} d^{o}_{j_1\ j_2}$$

また割当て問題の双対問題として知られているように，$D_i$と$\overline{D}_i$には式(17)の関係がある。すなわち，割当て問題は$\overline{d}^{i}_{j_1,\ j_2}$が非負の条件のもとで式(18)の値を最大にする問題と等価である。ここで$a_{j_1}$は行方向，$b_{j_2}$は列方向に減算する定数である。この式(17)より式(15)は式(19)となる。すなわと，定数$e_i$を持った$D_i$を解くことと，$v_i$を持った$\overline{D}_i$の巡回セールスマン問題を解くこととは等価となる。

$$(17)\quad \overline{d}^{i}_{j_1 j_2} = d^{i}_{j_1 j_2} - a_{j_1} - b_{j_2} \geq 0 \quad j_1,\ j_2 = 1,\ 2,\ \cdots,\ \ell$$

$$(18)\quad \overline{C}_i = \max\left( \sum_{j_1=1}^{\ell} a_{j_1} + \sum_{j_2=1}^{\ell} b_{j_2} \right)$$

$$(19)\quad C_i(t_i) = v_i + \sum_{(j_1,\ j_2)\in t_i} \overline{d}^{i}_{j_1\ j_2}$$

次に，この$P_i$より分岐する都市対(p，q)を決めるために，前節で述べたように$D_i$で禁止されていない都市対の中で，廃棄の問題の下界を最大にするものを選ぶ必要がある。このため$P_i$で各(p，q)を禁止した場合の問題を$P_{ipq}$とすると，目的関数$C_{ipq}$は式(19)より式(20)となる。また下界$v_{ipq}$の$v_i$よりの増分をこの(p，q)の選択指標と呼び$\theta_{ipq}$で表わすと，これは式(21)を満たせばよい。

$$(20)\quad C_{ipq}(t_{ipq}) = v_i + \sum_{(j_1, j_2)\in t_{ipq}} d^{ipq}_{j_1 j_2}$$

$$\text{ただし}\quad d^{ipq}_{pq} = \infty, \quad \text{その他は}\quad d^{ipq}_{j_1 j_2} = \bar{d}^{i}_{j_1 j_2}$$

$$(21)\quad \theta_{ipq} = v_{ipq} - v_i \leq \sum_{(j_1, j_2)\in t_{ipq}} d^{ipq}_{j_1 j_2}$$

$D_{ipq}$の下界に割当て問題の解を用いると，（p, q）が$\bar{t}_i$の要素でなければ$\bar{t}_{ipq} = \bar{t}_i$となり，$\theta_{ipq} = 0$となる。したがって，（p, q）は式(22)を満たすものとなり，廃棄する問題$P_{i1}$の下界は式(23)となる。

$$(22)\quad \max_{(p, q)\in \bar{t}_i} \theta_{ipq} \qquad \text{ただし}\quad \theta_{ipq} = \bar{C}_{ipq}$$

$$(23)\quad v_{i1} = v_i + \theta_{ipq}$$

これら$\bar{t}_i$の$\ell$個の要素について割当て問題を解くことは，時間がかかるばかりでなく，$\bar{D}_i$をもこわしてしまうので，実際には式(24)で$\theta_{ipq}$を計算する。これは双対問題より，$D_{ipq}$の割当て問題を近似的に解くことに相当する。

$$(24)\quad \theta_{ipq} = \min_{j_1 \neq p} d^{ipq}_{j_1 q} + \min_{j_2 \neq q} d^{ipq}_{p j_2}$$

本論では分岐拘束法を用いるので，（p, q）の選択した$P_{i2}$はすぐ次に解かれるため，$P_{i2}$の下界は必要ない。この$P_{i2}$の定式は次節で述べる。また$P_{i1}$での下界は，この$P_{i1}$より分岐が開始されるときに割当て問題を解いた厳密な下界が計算される。

### 3．3　分岐の戦略

本方法で用いる分岐拘束法の長所については，2－Ⅱで述べた。ここでは前節の$P_i$を分岐した得られる$P_{i2}$の計算量が如何に減少するかを示す。

$P_i$で(p, q)を選択した$P_{i2}$では，$D_{i2}$を$D_{ipq}$と同様に$\bar{D}_i$と(p, q)から，式(13)で簡単に計算できる。また$C_{i2}$は式(25)となる。

$$(25)\quad C_{i2}(t_{i2}) = v_i + \sum_{(j_1\ j_2)\in t_{i2}} d^{i2}_{j_1 j_2} \quad \because,\ \bar{d}^{i}_{pq} = 0$$

このとき，この$D_{i2}$と$C_{i2}$は式(15)の場合を途中まで割当て問題を解いたものとなっている。さらに式(26)が満たされると式(27)が成立して，割当て問題は解く必要がない。

$$(26)\quad (q, p) \notin \bar{t}_i$$

$$(27)\quad \bar{t}_{i2} = \bar{t}_i\ ,\quad \bar{C}_{i2} = \bar{C}_i$$

$v_i$が上界u以上になるか，$\ell = 2$が成立するまでこの式(25)を用いて選択した方の$P_{i2}$は計算される。またその途中で，（q, p）$\in \bar{t}_i$となった場合にのみ割当て問題が解かれる。したがって式(15)から毎回$P_i$を作る場合より大幅に計算量が減少する。

### 3．4 アルゴリズム

提案する解法のフローチャートを図4に示し，以下に簡単にアルゴリズムを示す。

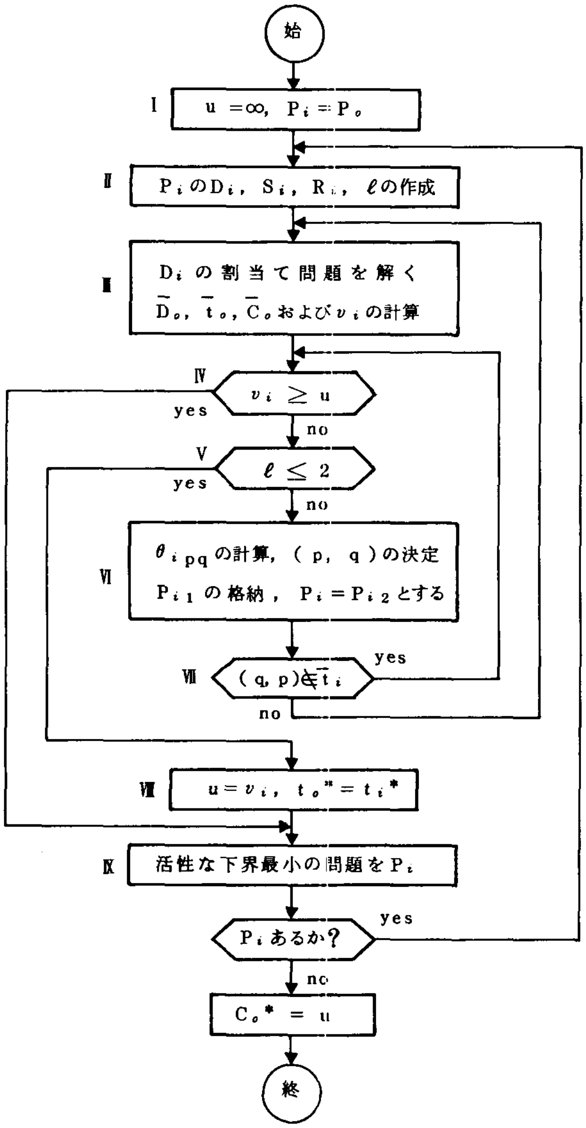



図4 フローチャート

I 上界$u=\infty$とし，分岐する問題$P_i$を$P_o$とする。

II 式(15)で$P_i$の$D_i$, $S_i$, $R_i$, $\ell$を作る。

III $D_i$の割当て問題を解き，下界$u_i$を計算する。

IV $u_i \geq u$ならばIXへ。

Ⅴ　$\ell \leq 2$ ならばⅧへ。

Ⅵ　式(24)で選択指標$\theta_{ipq}$を計算し，分岐する都市対$(p, q)$を決定する。$v_{i1} < u$ならば$P_{i1}$を活性な問題として記憶する。式(25)で$P_{i2}$を作り，$P_i = P_{i2}$，$\ell = \ell - 1$とする。

Ⅶ　式(26)を満たせばⅣへ，他はⅢへ。

Ⅷ　$u = v_i$とし，$u$より大きな下界の問題を不活性とする。$t_o^* = t_i^*$とする。

Ⅸ　活性な問題の中で下界最小のものを$P_i$としてⅡへ。問題がなければ$C_o^* = u$として終了。

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | ∞ | 24 | 18 | 22 | 31 | 19 | 33 | 25 | 30 | 26 |
| 2 | 15 | ∞ | 19 | 27 | 26 | 32 | 25 | 31 | 28 | 18 |
| 3 | 22 | 23 | ∞ | 23 | 16 | 29 | 27 | 18 | 16 | 27 |
| 4 | 24 | 31 | 18 | ∞ | 19 | 13 | 28 | 9 | 19 | 27 |
| 5 | 23 | 18 | 34 | 20 | ∞ | 31 | 24 | 15 | 25 | 8 |
| 6 | 24 | 12 | 17 | 15 | 10 | ∞ | 11 | 16 | 21 | 31 |
| 7 | 28 | 15 | 27 | 35 | 19 | 18 | ∞ | 21 | 21 | 19 |
| 8 | 13 | 24 | 18 | 13 | 13 | 22 | 25 | ∞ | 29 | 24 |
| 9 | 17 | 21 | 18 | 24 | 27 | 24 | 34 | 31 | ∞ | 18 |
| 10 | 18 | 19 | 29 | 16 | 23 | 17 | 18 | 31 | 23 | ∞ |

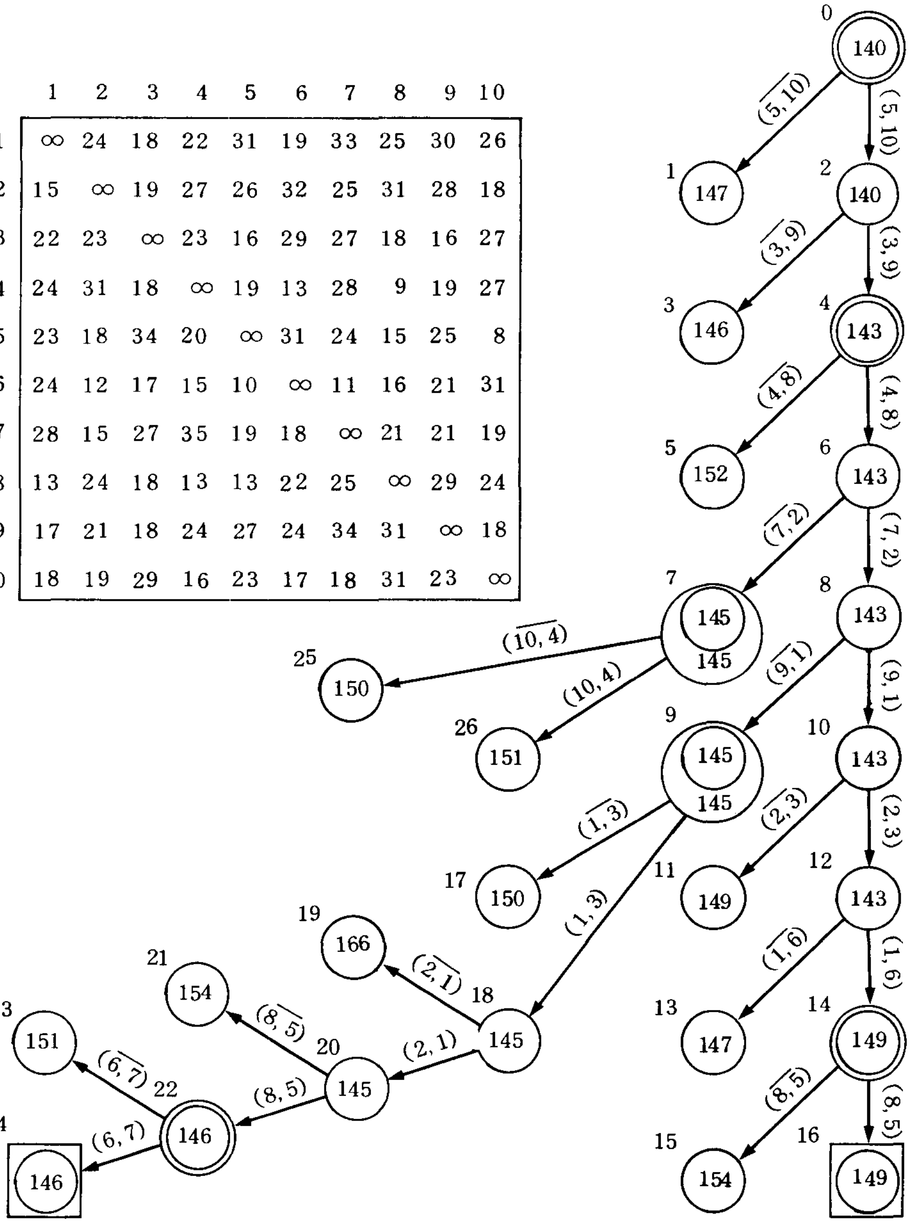



図5　10都市問題の分岐図

図5に10都市問題の1例[2]を本方法で解いたときの分岐図と，そのコストマトリクスを示す。最初ノード0より選択の分岐を繰返してノード16で上界149が得られ，次にノード9の最小下界から分岐が開始され上界146が得られる。割当て問題が解かれたときの下界は二重丸の内側に，可能解は四角で示す。3回目の分岐をノード7より開始してノード26で分岐は終了し，ノード24が厳密解となる。全体で解かれた割当て問題の回数は，10，8，7都市が各1回，6，3都市が各2回である。なお，割当て問題の解法はＳｉｌｖｅｒの方法[6]に若干手を加えて用いた。

## 4. 実験と考察

本方法のアルゴリズムをＦＯＲＴＲＡＮでゴーデングし，乱数を用いたコストマトリクスで実験を行った。以下に例題の作成法，結果と考察および本方法の特徴について述べる。なお，使用した計算機は，デバックには北海道大学大型計算機センターのＦＡＣＯＭ２３０／６０，実験には仮想記憶のある東京大学大型計算機センターのＨＩＴＡＣ８８００である。

### 4．1　例題の作成

実験は，都市数５０から３００までを５０刻みでそれぞれ１０例ずつ行った。各都市数でのコストマトリクスは，図6に示すように３００都市での場合を整数3桁の一様乱数で作り，左上隅の都市数のマトリクスを使用した。乱数はセンターのサブルーチンＧ５ＲＡＮＤＵＮを，各パラメータ標準値のままで使用した。図中には１０都市の場合の数値を示してあり，－１は禁止を表わす。

| i＼j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | · | · | · | 50 | · | · | 300 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | −1 | 835 | 210 | 183 | 32 | 136 | 740 | 403 | 644 | 769 | · | · | · | 593 | · | · | 910 |
| 2 | 509 | −1 | 594 | 455 | 100 | 0 | 778 | 555 | 858 | 864 | · | · | · | 690 | · | · | 718 |
| 3 | 970 | 302 | −1 | 273 | 410 | 411 | 794 | 928 | 19 | 864 | · | · | · | 889 | · | · | 362 |
| 4 | 290 | 71 | 807 | −1 | 851 | 238 | 404 | 906 | 143 | 501 | · | · | · | 380 | · | · | 198 |
| 5 | 775 | 812 | 855 | 450 | −1 | 655 | 608 | 431 | 249 | 842 | · | · | · | 705 | · | · | 939 |
| 6 | 20 | 278 | 41 | 834 | 194 | −1 | 918 | 998 | 538 | 132 | · | · | · | 150 | · | · | 591 |
| 7 | 127 | 955 | 688 | 33 | 407 | 852 | −1 | 270 | 463 | 956 | · | · | · | 471 | · | · | 105 |
| 8 | 509 | 108 | 219 | 998 | 359 | 493 | 795 | −1 | 281 | 490 | · | · | · | 432 | · | · | 69 |
| 9 | 340 | 726 | 304 | 204 | 688 | 168 | 906 | 684 | −1 | 185 | · | · | · | 180 | · | · | 664 |
| 10 | 538 | 458 | 687 | 726 | 543 | 961 | 703 | 599 | 855 | −1 | · | · | · | 837 | · | · | 109 |
| · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · |
| · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · |
| · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · |
| 50 | 191 | 553 | 46 | 5 | 871 | 617 | 139 | 549 | 836 | 975 | · | · | · | −1 | · | · | 797 |
| · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · | · |
| 300 | 883 | 770 | 493 | 202 | 915 | 559 | 693 | 360 | 611 | 113 | · | · | · | 63 | · | · | −1 |

図6　コストマトリクスの一例

表1 実 験 結 果

| 都　市　数　(N) | | 50 | 100 | 150 | 200 | 250 | 300 |
|---|---|---|---|---|---|---|---|
| 例　題　数 | | 10 | 10 | 10 | 10 | 10 | 10 |
| 解けなかった例題数 | | 0 | 0 | 0 | 3 | 2 | 4 |
| 計算時間(CPU)秒 | 最小 | 0.8 | 4.6 | 10.6 | 17.6 | 22.1 | 270.5* |
| | 平均 | 4.8 | 45.2 | 109.5 | 281.3* | 252.5* | 270.5* |
| | 最大 | 13.4 | 154.5 | 408.9 | 600.0* | 600.0* | 600.0* |
| 上界の個数(平均) | | 3.3 | 3.8 | 3.2 | 2.7* | 3.0* | 2.5 |
| 最初の上界が厳密解の場合 | | 2 | 1 | 2 | 3 | 3 | 0 |
| 分岐の回数 | 最小 | 48 | 98 | 148 | 198 | 248 | 315 |
| | 平均 | 167.9 | 588.7 | 765.6 | ... | ... | ... |
| | 最大 | 407 | 1675 | 2155 | ... | ... | ... |

### 4.2 実験結果と考察

表1に各都市数での結果を示す。計算時間はすべて10分で打切った。この時間内に10例すべてが解けたのは150都市問題までで，200，250，300都市ではそれぞれ3，2，4個解けないものがあった。これらについては，計算時間に＊印を付けて10分とし，その他は除いて計算してある。

Svestka らの論文[7]では，計算機UNIVAC1108を使用して60都市問題までの多数の問題を解き，その結果から100都市問題では30分程度の計算時間が必要であると予想している。本方法では，同程度の計算機でありながら平均45秒程度で，最大で2.5分程度で100都市問題を解くことが可能である。また10例中6例までの300都市問題を10分以内に解くことができたが，これは現在解かれている最も大きな都市数である。

また表1では，最初の上界がそのまま厳密解である場合が多く，1例題当りに得られる上界の個数も少ないことを示している。これは廃棄の概念による分岐の方法と分岐の戦略の正当性を示していると思われる。

表2には，各都市数で解けなかった1例ずつの上界の得られた時間とそのコスト，およびその時の活性な問題の1番小さな下界の値を示してある。300都市の例では，約54秒で最初の上界1902が得られ，未分岐の1番小さな下界は1707である。これが約2分で上界1710，下界1707となり，あと3の最適性の保証のために漠大な時間が費やされる。これは2－Ⅱで述べた廃棄する方の下界をできる限り大きくすべきことの必要性を示している。また都市数が大きくなると，式(25)の選択指標を使うよりは，式(23)の毎回割当て問題を解く方が良い可能性をも示している。

表２　解けなかった場合の例

| | ２００都市の一例 | | | ２５０都市の一例 | | | ３００都市の一例 | | |
|---|---|---|---|---|---|---|---|---|---|
| | 時間 秒 | 得られた上界 | その時最小の下界 | 時間 秒 | 得られた上界 | その時最小の下界 | 時間 秒 | 得られた上界 | その時最小の下界 |
| 1 | 16.3 | 1796 | 1718 | 21.8 | 1659 | 1538 | 54.1 | 1902 | 1707 |
| 2 | 24.8 | 1729 | 1718 | 35.8 | 1636 | 1538 | 72.3 | 1840 | 1707 |
| 3 | 182.3 | 1726 | 1721 | 47.0 | 1598 | 1538 | 93.6 | 1715 | 1707 |
| 4 | 319.5 | 1724 | 1722 | 52.0 | 1545 | 1538 | 111.1 | 1710 | 1707 |
| 5 | 600.0 | ・・・ | 1723 | 600.0 | ・・・ | 1541 | 600.0 | 0 ・・・ | 1708 |

表３　乱数の桁数による影響（都市数は１００）

| | | 計算時間（秒） | 分岐の回数 | 上界の個数 | 最初の上界の得られる時間（秒） |
|---|---|---|---|---|---|
| ３桁乱数（０〜1,０００） | 最小 | 3.91 | 98 | （1）* 1 | 3.88 |
| | 平均 | 35.88 | 587.8 | 3.9 | 4.40 |
| | 最大 | 119.47 | 1675 | 6 | 5.53 |
| ２桁乱数（０〜１００） | 最小 | 1.56 | 9.8 | （2）* 1 | 1.54 |
| | 平均 | 15.18 | 416.2 | 2.9 | 2.02 |
| | 最大 | 72.51 | 1265 | 6 | 3.16 |
| １桁乱数（０〜１０） | 最小 | 1.10 | 98 | （5）* 1 | 1.07 |
| | 平均 | 2.22 | 122.8 | 1.7 | 1.51 |
| | 最大 | 2.97 | 218 | 4 | 1.84 |

なお乱数の桁数の影響を見るために，同じ乱数で桁数のみを１桁，２桁，３桁として１００都市問題１０例ずつを解いた結果を表３に示す。使用した計算機は北海道大学大型計算機センターのＦＡＣＯＭ２３０／７５である。結果は，乱数であれば桁数が小さい程解き易いことを示しており，特に１桁の場合には１０例中５例まで（図中＊印で示す）最初の上解が厳密解となっている。これは最初の上界の得られる時間から割当て問題を解く時間が非常に短かくなっていること，さらに分岐の回数などから選択で分岐を繰返す途中での下界の増加が小さくなっていることなどが原因であろう。

### 4．3　従来の解法との比較

従来の巡回セールスマン問題の厳密解法は，主に次のB－B法を用いた二つのグループになる。[1] [2] 一つはLittle[3]らによるTour Building法であり，他はShapiro[5]その他[7]によるSubtour Elimination法である。以下にこれらとの比較検討を行ない，本方法がこれら二つの解法グループを綜合的に再構成した形となっていることを示す。

Tour Building法は，分岐の方法は本方法とほとんど同じで，分岐の際に廃棄の概念も含まれているが，下界にはマトリクスから決まった順序で行と列から定数を引いたものが使われている。したがって，下界が小さくなり充分に廃棄の機能が働かず活性な問題の数が非常に増加してしまう。本方法は，割当て問題の解を用いてこの下界の最適化を計ったことに相当する。さらに分岐拘束法により，この割当て問題を解く回数を大幅に減じている。

Subtour Elimination法は，下界には割当て問題を用いているが，分岐の方法はこの割当て問題の解のサブサイクルを禁止するように一意に決めている。したがって，2－Ⅱの廃棄の概念がなく，また独立性をも満たしていない。下界の影響を除くためにD。の要素がすべてゼロの場合を考えると，この解法では割当て問題の解が偶然にサブサイクルを含まなくなるまで非常に多くの，場合によっては（n－1）！個以上のn次の割当て問題を解かなければならない。本方法では（n－2）回の分岐で，またn次と若干のさらに小さな次数の割当て問題を解くのみで1個の可能解が求まる。

したがって，本方法は分岐にはTour Building法，下界の計算にはSubtour Elimination法のそれぞれの長所をとり，それらを綜合的に再構成した形となっている。さらに本方法の予想をこえる結果については，2で述べたB－B法の適用法の考察の正しさを示していると思われる。

## 5. むすび

巡回セールスマン問題は組合せ問題の一つであり，次数の増加は解法にとって非常に困難な問題である。B－B法の適用法の考察のもとに本論文で提案した解法は，現在までに解かれたことのない300都市問題を解いた。実験結果は，さらに大きな次数の問題をもかなりの近似度で実用的に解き得ることを示している。このB－B法の適用法で述べた三点については，巡回セールスマン問題に限らず他の問題の解法でも充分に考慮されるべきであろう。

おわりに本研究にさいし，種々ご協力をいただいた北大工学部精密工学科精密機器学第一講座の諸氏に深甚の謝意を表わします。

## 参考文献

〔1〕 Bellmore, M. and G.L.Nemhauser, ″The Traveling Salesman Problem : A Survey″, Opns.Res.16, (1968)

〔2〕 Lawler, E.L. and D.E.Wood″, Branch+and−Bound Methad : A Survey″, Opns.Res.14, (1966)

〔3〕 Little, J.D.C., K.G.Murty, D.W.Sweeney and C.Karel″, An Algorithm for the Traveling Salesman Problem″, Opns.Res.11, (1963)

〔4〕 Held, M. and R.M.Karp,″ The Traveling Salesman Problem and Minimim Spanning Trees : Part II″, Math.Progr.1(1971)

〔5〕 Shapiro, D, ″Algorithms for thesolution of the Optimal Cost Traveling Salesman Problem″, Sc.D.Thesis, Washington University, St.Louis, (1966)

〔6〕 Silver, R., ″Algorithm 27, Assignment″, Commwn.of ACM.11, (1960)

〔7〕 Svestka, J.A. and V.E.Huckfeldt, ″Computational Experience With an M-Salesman Traveling Algorithm, ″Managemeut Sci. 19, (1973)

〔8〕 久保洋, 沖野教郎, "巡回セールスマン問題の実用的解法に関する研究" 精密機械 4 1, 4, (1975)

## 脚 注

1） 問題$P_i$が変数の領域$T_i$のみで定まることより, $P_i$をこの$T_i$の関数と考える。

2） $D_o$の要素が無限大の都市村は, 禁止されていると考えて一巡経路に含めない。

3） toはn個, $t_2$は(n－1)個の都市村からなるが, $t_2$のrをもとのp, gに戻すとtoと$t_2$は1対1に対応するので, 以下でも同様にそのままで取扱う。