合流巡回セールスマン問題に関する研究

小林　克也　（沼田　一道　准教授）

1　はじめに

近年, 地域犯罪の増加に伴い, 警察官によるパトロールだけではなく, 学校や地域を挙げてのパトロールが実施されるようになり、その重要性が再確認されている．その際, 場所によっては複数人で見回るべき地点（犯罪多発地域、混雑地域）とそうでない地点があり, それらを踏まえ, 能率よく巡回パトロールする経路が求められている．今, 2 人が起点を出発し, それぞれがパトロール地点を経由し, ある地点では合流し 2 人でパトロールを行い, 起点に戻るケース (case1) を考える (図 1)．このとき, 合流しパトロールを行うため, 2 人が合流地点に到着する時間差は出来る限り小さいことが望ましい．case1 のような「巡回する」際のコストを最小化する問題として, 巡回セールスマン問題（Travelling Salesman Problem, 以下 TSP）や複数巡回セールスマン問題（Multiple Traveling Salesman Problem, 以下 MTSP）が存在する．TSP とは「全ての都市を一度だけ訪問し起点に戻るときのコストを最小化する問題」で, 組合せ最適化問題の中で最も重要な問題のひとつであり, 盛んに研究されてきた．MTSP とは「複数人で全ての都市を一度だけ訪問し起点に戻るときのコストを最小化する問題」で, TSP 同様研究が進んでいる．しかし case1 を TSP に帰着させると全ての点を 2 人で訪問する必要があり能率が悪く, 各地点の需要人数（1 人 or 2 人）が異なるため, 全ての点をどちらか 1 人が訪問する 2TSP に帰着させると制約を守れない (図 2)．つまり, case1 のような現実問題が存在し解決策が望まれているにもかかわらず, 解決方法はおろか問題すら明確に定義されていない．そこで本研究では, 各都市の需要人数が 1 人もしくは 2 人と異なり, 2 人が訪問する都市では合流する場合の巡回経路を求める問題を一般化し, その問題及び解法を提案する．

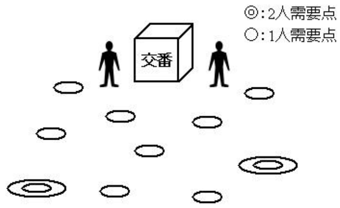


図 1　地点によって需要人数が異なるパトロール問題

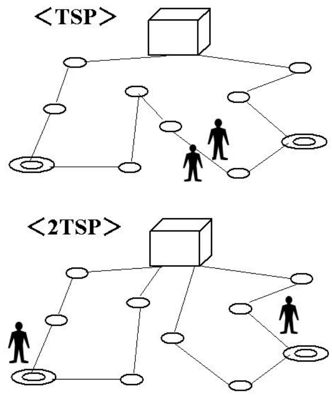


図 2　既存問題への帰着例

2　提案問題

2.1　状況設定

平面上に n 個の点—m 個の 2 人需要点 (以下合流点) と $n-m$ 個の 1 人需要点 (以下単点)—および任意の 2 点間を結ぶ潜在的な無向枝からなるグラフが与えられ, 2 人のセールスマンが存在しているものとする．頂点集合を $V=\{1,...,n\}$, 合流点集合を $J=\{1,...,m\}\ (\subset V)$, 単点集合を $S=V\backslash J$, セールスマン集合を $R=\{1,2\}$ で表す．

2.2　定義

本研究で提案する問題を合流巡回セールスマン問題 (Randezvous Traveling Salesman Problem, 以下 RTSP) と呼び, RTSP を「2 人のセールスマンが起点を出発し, 単点をどちらか 1 人が訪問し, 合流点では 2 人が合流した後訪問し, 2 人が起点に戻るまでの巡回時間を最小化する問題 (図 3) 」と定義する．

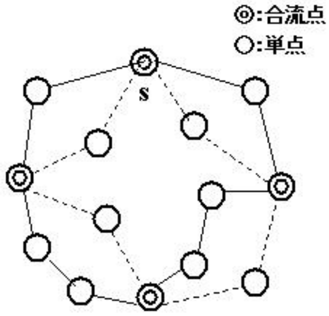


図 3　RTSP(n=14,m=4)

2.3 条件

以下に, RTSP の条件を記す.

- 2 人のセールスマンの移動速度は等しく一定であるとする.
- 合流点に先に着いた者は後から到着するものを待つ (図 4).
  詳しくは, 2.4 節で述べる.
- 巡回時間は点間の移動時間のみを考慮する.

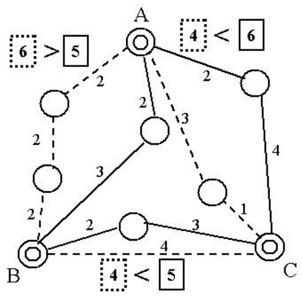


図 4　例題 (巡回時間 17)

2.4 巡回時間計算

RTSP の巡回時間がどのように計算されるか, n=9, m=3 の例題 (図 4) を用いて説明する. 図 4 において, 2 人が合流点 A を出発し, B, C の順に訪問した後, A に戻るケースを考える. 枝の数字がその点間の移動時間を表し, 実線で表す経路をパス 1, 点線で表す経路をパス 2 とする. 合流点 AB 間におけるパス 1, パス 2 の移動時間はそれぞれ 5, 6 である. このとき, パス 1 を辿った者は $6 - 5 = 1$ だけ待ち, 次の合流点 C を目指す. よって, 合流点 AB 間で費やした時間は両者とも 6 である. つまり, 2 人のうち後から到着する者の移動時間がその合流点間の 2 人の移動時間となる. 同様に計算し, BC 間では 5, CA 間では 6 となる. したがって, 例題の巡回時間は $6 + 5 + 6 = 17$ となる.

3 定式化

まず, 変数を定義する. z を総巡回時間とし, t_{ij} を点 (i, j) 間の移動時間とする. $x^{(p)}_{ijk\ell}$ を「2 人のセールスマンが合流点 k の次に合流点 ℓ を訪問し, セールスマン p が合流点 (k, ℓ) 間で点 i の次に点 j を訪問するとき 1, それ以外の場合 0」をとる決定変数とする.

以上の記号を用いて, RTSP を以下のように定式化する.

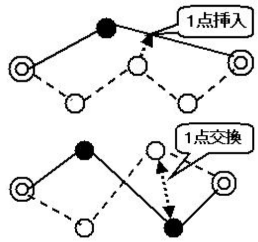
$$
\text{(RTSP)}\left|\begin{array}{lll}
\text{minimize} & z = \displaystyle\sum_{k\in J}\sum_{\ell\in J\setminus\{k\}} \max_{p=1,2} \sum_{i\in V}\sum_{j\in V\setminus\{i\}} t_{ij} x^{(p)}_{ijk\ell} & (1)\\
\text{subject to} & \displaystyle\sum_{p=1}^{2}\sum_{k\in J}\sum_{\ell\in J\setminus\{k\}}\sum_{j\in V\setminus\{i\}} x^{(p)}_{ijk\ell} = 1 \quad \forall i \in S & (2)\\
& \displaystyle\sum_{p=1}^{2}\sum_{k\in J}\sum_{\ell\in J\setminus\{k\}}\sum_{i\in V\setminus\{j\}} x^{(p)}_{ijk\ell} = 1 \quad \forall j \in S & (3)\\
& \displaystyle\sum_{p=1}^{2}\sum_{k\in J}\sum_{\ell\in J\setminus\{k\}}\sum_{j\in V\setminus\{i\}} x^{(p)}_{ijk\ell} = 2 \quad \forall i \in J & (4)\\
& \displaystyle\sum_{p=1}^{2}\sum_{k\in J}\sum_{\ell\in J\setminus\{k\}}\sum_{i\in V\setminus\{j\}} x^{(p)}_{ijk\ell} = 2 \quad \forall j \in J & (5)\\
& \displaystyle\sum_{i\in T}\sum_{j\in V\setminus T} x^{(p)}_{ijk\ell} \geq 1 \quad (T \neq \emptyset, T \neq V) \quad \forall k, \ell \in J \quad \forall p \in R & (6)\\
& x^{(p)}_{ijk\ell} \in \{0, 1\} \quad \forall i, j \in V \quad \forall k, \ell \in J \quad \forall p \in R & (7)
\end{array}\right.
$$

(1) 式は，総巡回時間を最小化することを表す．(2) 式は，単点をどちらか 1 人が訪問することを表す．(3) 式は，単点からどちらか 1 人が出て行くことを表す．(4) 式は，合流点を 2 人が訪問することを表す．(5) 式は，合流点から 2 人が出て行くことを表す．(6) 式は，部分巡回路の出現を禁止することを表す．(7) 式は，決定変数 $x^{(p)}_{ijk\ell}$ は 0 もしくは 1 をとることを表す.

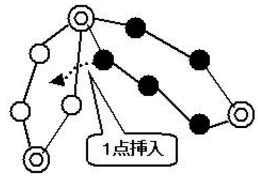
4　解法

4.1　方針

提案する解法 (以下, 提案法) は, 各単点をどの合流点間で訪問する (どの合流点間に割り当てる) かに帰着して考える．提案法は大別すると 2 段階で構成され, 第 1 段階では初期解を構築的に求める．まず合流点の訪問順序を求め, 次に各単点を最も近い合流点間をつなぐ線分 (合流点間) に割り当てる．第 2 段階では 2 つの局所探索を行う．まず各合流点間に割り当てられた単点集合を 2 つの group に分割する．両合流点を端点とし, group に属する単点を全て通る最短経路 (2-opt 法によって近似的に求める) をパスとする．また, 2 つのパスを合わせてルートとする．次に 2 つの group 間で単点を移動することで移動時間の長い方のパス長を短縮する．長い方のパス長で決まる合流点間の移動時間を改善するために行うこの操作をパス間局所探索 (以下 PLS) と呼ぶ (図 5(a))．全ての合流点間で PLS を行った後, ルート間で単点を移動し巡回時間が改善される場合, その単点の割当を変更するルート間局所探索 (以下 RLS) を行う (図 5(b))．RLS 後の巡回時間が暫定巡回時間よりも小さい場合, 暫定巡回時間を更新し, RLS 後の各合流点間に対する単点の割当を初期解として, PLS → RLS を繰り返す.



(a)　パス間局所探索



(b)　ルート間局所探索

図 5　局所探索

4.2　求解手順

以下に求解手順を示す.

[phase1] 初期解生成

step1 合流点の訪問順序 (最短巡回路) を TSP に帰着させ, 列挙法によって求める．求めた巡回路において, 任意に起点 $J_1(=J_{m+1})$ を定め, 隣接する合流点を J_2,...,J_m とし, J_k と J_{k+1} をつなぐ線分を P_k とする (図 6).

step2 各単点を最も近い合流点間をつなぐ線分 $(P_1,...,P_m)$ に割り当てる．但し, $i(\in S)$ と P_{k-1}, P_k との距離が等しく最も短い場合, $\angle iJ_kJ_{k-1} < \angle iJ_kJ_{k+1}$ なら P_{k-1} に割り当て, そうでなければ P_k に割り当てる (図 6)．P_k に割り当てられた単点集合を G_k とする.

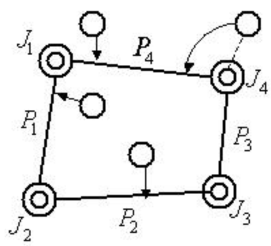


図 6　phase1

第 2 段階において, z を巡回時間とし, $z_{k\ell}^{(i)}$ を G_k から G_ℓ へ単点 i を挿入し, (J_k,J_{k+1}) 間, $(J_\ell,J_{\ell+1})$ 間でそれぞれ PLS を行ったときの巡回時間とする．また, (J_k,J_{k+1}) 間のパスと単点 i の最短距離を $d_k^{(i)}$ とする (図 7).

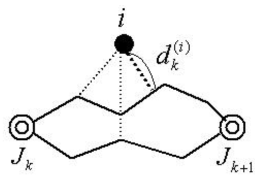


図 7　単点とパスの距離

[phase2] 局所探索 (PLS & RLS)

step0 $z^* \leftarrow \infty$ とする.

step1 $k \leftarrow 1$ とする.

step1-1 G_k をランダムに 2 つの単点集合 $g_k^{(1)}, g_k^{(2)}$ に分割する.

step1-2 $g_k^{(1)}, g_k^{(2)}$ に対して PLS を行い, (J_k,J_{k+1}) 間の移動時間を改善する.

step1-3 step1-1〜step1-2 を一定回数繰り返し, 最も小さい (J_k,J_{k+1}) 間の移動時間を z_k とする.

step1-4 $k<m$ なら, $k \leftarrow k+1$ とし, step1-1 へ．$k=m$ なら, $z \leftarrow \sum_{k=1}^{m} z_k$ とし, step2 へ.

step2 $z_{k\ell}^{(i)^*} \leftarrow \min\limits_{k=1,..,m} \min\limits_{\ell=1,...,m(\neq k)} \min\limits_{i\in G_k} z_{k\ell}^{(i)}$ とする.

step3 $z_{k\ell}^{(i)^*} < z$ なら, $z \leftarrow z_{k\ell}^{(i)^*}$, $(k',\ell',i') \leftarrow \arg z_{k\ell}^{(i)^*}, G_{k'} \leftarrow G_{k'} - \{i'\}, G_{\ell'} \leftarrow G_{\ell'} + \{i'\}$ とし, step2

へ. そうでなければ, step4 へ.

step4 $z < z^*$ なら, $z^* \leftarrow z$, $G_k^* \leftarrow G_{k(k=1,\ldots,m)}$, $w \leftarrow 1$ とし step5 へ. それ以外の場合, $w < m$ なら $w \leftarrow w+1$ とし, step5 へ, $w = m$ なら z^* を巡回時間, $G^*_{k(k=1,\ldots,m)}$ を解とし終了.

step5 $(i', \ell') \leftarrow \arg \min_{i \in G_w^*} \min_{\ell=1,\ldots,m(\neq w)} d_\ell^{(i)}$ とする. $G_w \leftarrow G_w^* - \{i'\}, G_{\ell'} \leftarrow G_{\ell'}^* + \{i'\}$ とし, step1 へ.

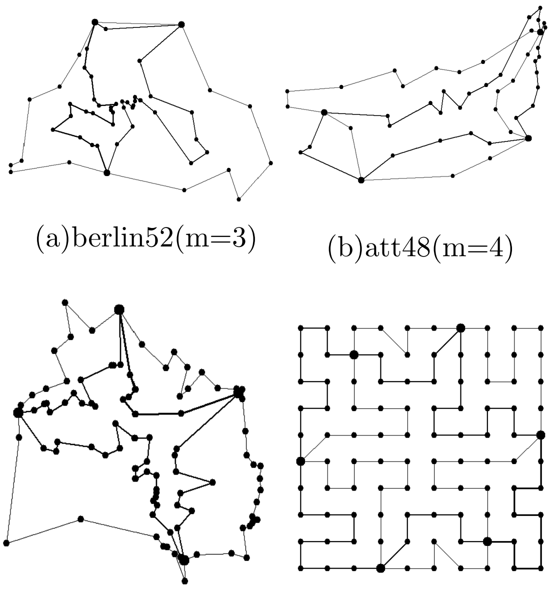
5 数値実験

提案解法を, Borland 社の delphi6 によって実装した. 点データとして, TSPLIB[2] の att48, berlin52, gr96, そして 10 行 10 列の格子状に点が存在するもの (以下 grid100) を用いた. データ名の後の数字が頂点数を表す. 本研究では, 3〜10 個の点をランダムに合流点に設定し実験を行った. 表 1 に実験結果, 図 8 に巡回の様子を示す.

表 1 より, どのデータも合流点数が増えると巡回時間が大きくなる. これには, 2 つの理由があると考えられる. 1 つは 2 人の訪問点数の和が $n+m$ で表され, 合流点数が増えれば訪問する点数も増加するため, 当然といえる. また, 提案解法では各合流点間で移動時間の長いパスを最小化する処理を行っている. 図 8(b) のように 2 本のパスが交差することでそれを実現する例もあり, そうした動きが m 個の合流点間で行われていることで, 巡回時間が増加すると考えられる. 次に, 計算時間も合流点数に比例して増加している. これは提案解法の RLS において巡回時間が改善されたとき, m 個の近傍が存在することによると考えられる. つまり合流点数が増えれば近傍が増え, それだけ解が改善される確率も高くなるが, 改善された場合, 更に m 個の近傍が存在するという性質を持っているため, 計算時間が合流点に比例して大きくなると考えられる.

表 1 実験結果

データ	合流点数	巡回時間	計算時間 (秒)
berlin52	3	1897	18
berlin52	5	2211	144
gr96	4	1683	93
gr96	8	2024	225



(a)berlin52(m=3) (b)att48(m=4)

(c)gr96(m=4) (d)grid100(m=6)

図 8 実験結果

6 まとめ

本研究では, RTSP という新たな問題を提起し, その解法を示した. そこで得られた知見は, 既存問題に帰着できない現実問題を解決する際の足がかりになると思われる. パトロール経路問題と同様に扱える現実問題として, 訪問看護経路問題や同じ地点を経由する循環型バス路線経路問題などが考えられる. また, 提案解法は局所探索を基本としたオーソドックスな手法であり, 今後, 様々な観点からの改善を試みる必要がある. さらに, 2 本のパス (2 人のセールスマン) を区別する, セールスマンの人数を増やすなど, 問題の発展・応用によって多くの可能性を秘めているが, これらは今後の課題である.

参考文献

[1] 山本芳嗣, 久保幹雄: 巡回セールスマン問題への招待, 朝倉書店 (1997)

[2] TSPLIB,http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/ (最終閲覧日 2009/11/30)

[3] 柳浦睦憲, 茨木俊秀: 組合せ最適化ーメタ戦略を中心としてー, 朝倉書店 (2001)