合流巡回セールスマン問題に関する研究

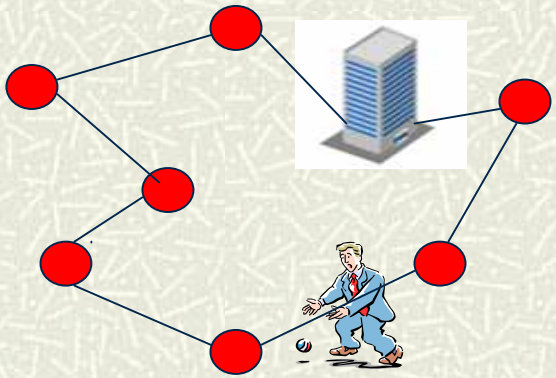
4406028
小林　克也　(沼田研究室)

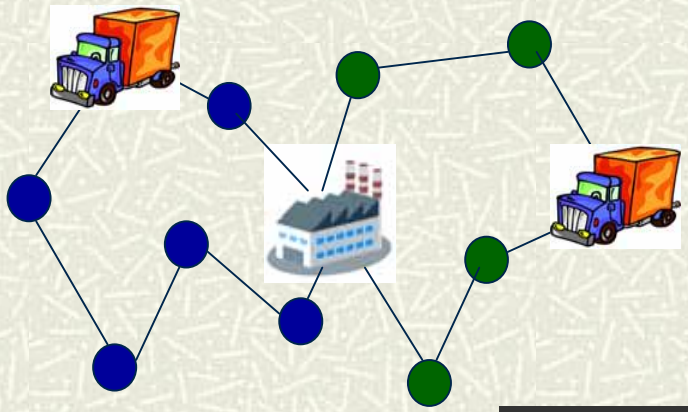
目次

1．はじめに(1)

- 代表的な組合せ最適化問題である巡回セールスマン問題（以下：TSP）は配送計画や基盤配線など多くの現実問題を一般化しており、その研究は盛んに行われてきた。

- TSPの発展として、M人のセールスマンで巡回する複数巡回セールスマン問題（以下：MTSP）が提案され、以降、研究が進んでいる。

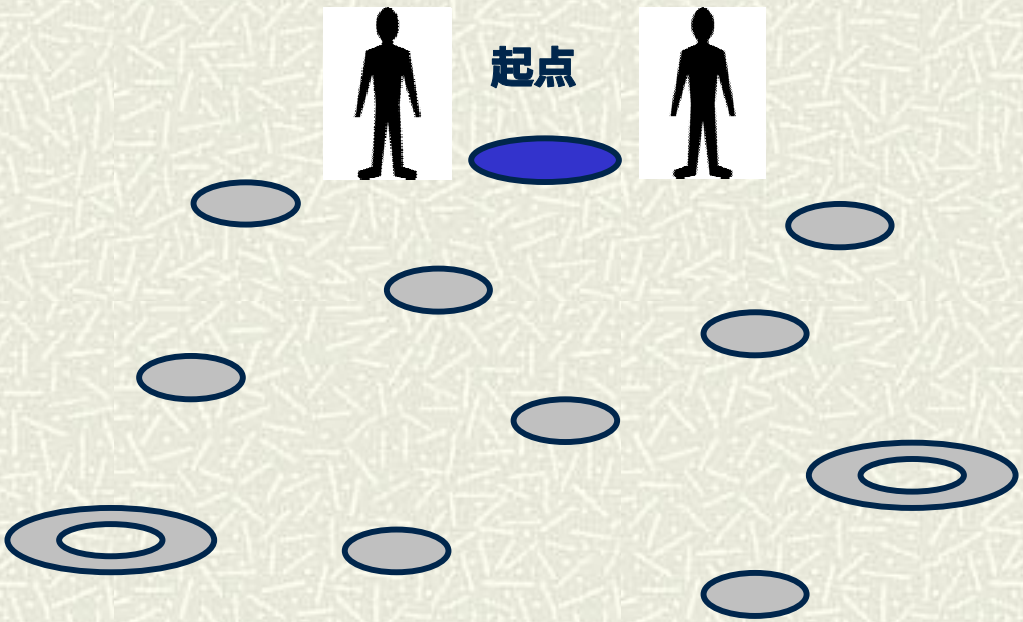




1．はじめに (2)

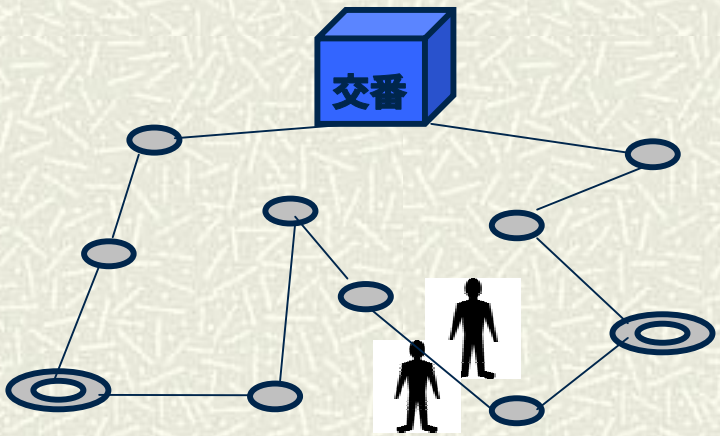
- しかし，「巡回する」という性質を持ちながら，TSP,MTSPとは異なる現実の問題が存在する。

<ex> ◎を2人，○を1人が訪問する場合

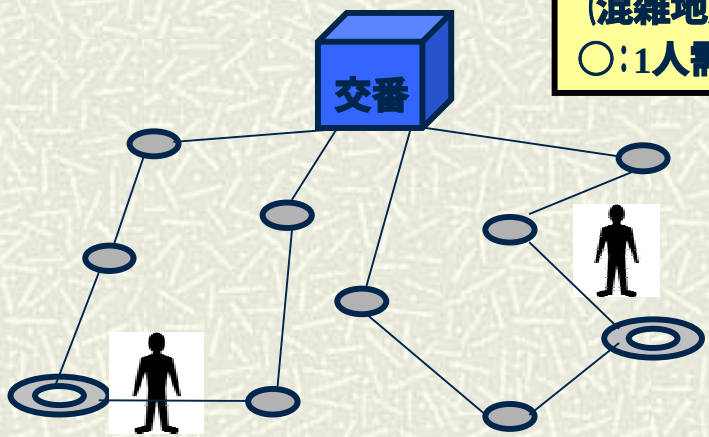


1．はじめに(3)
ー実例（パトロール経路問題）ー

＜TSP＞



＜2TSP＞



◎：2人需要点
（混雑地点，犯罪多発地点）
○：1人需要点

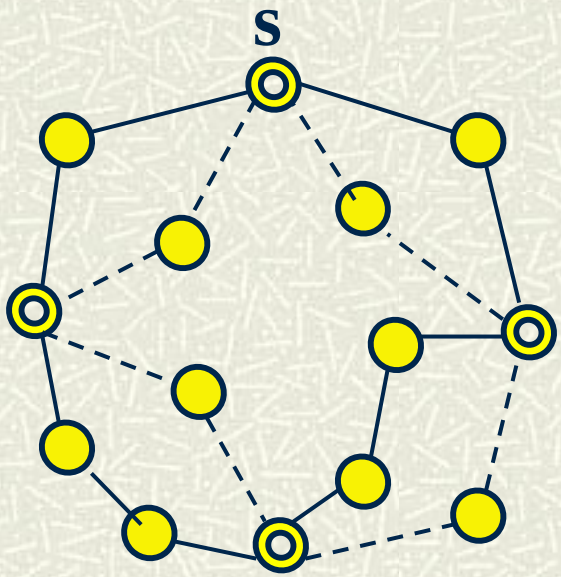
TSPに帰着する場合，○を2人で訪問することになり，能率が悪い．
2TSPに帰着する場合，◎を2人で訪問できず，制約を守れない．

2人需要点をカバーしながら，
全ての点を能率よく訪問する 経路が求められる．

2．提案問題

2．1　定義

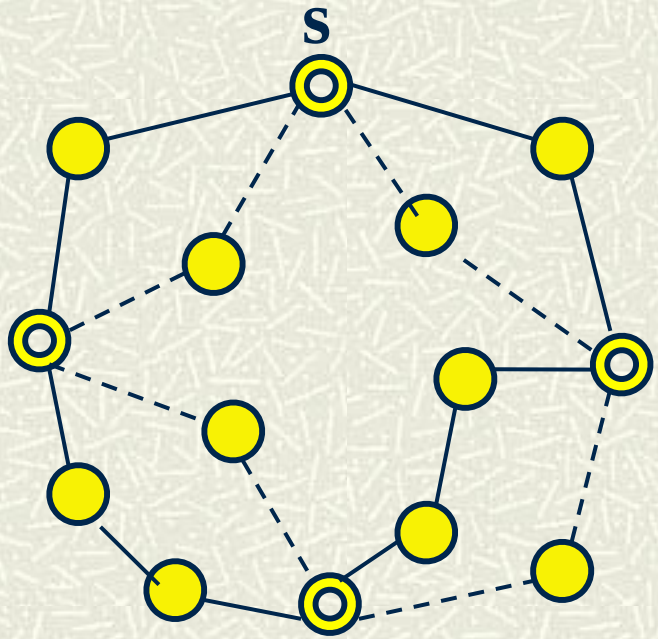
- **合流巡回セールスマン問題**（Rendezvous Traveling Salesman Problem：**以下**RTSP）**を、「2人のセールスマンが起点を出発し、単点をどちらか1人が訪問し，合流点では2人が合流した後訪問し，2人が起点に戻るまでの巡回時間を最小化する問題」と定義する。**



◎：合流点（2人需要点）
○：単点（1人需要点）

2．2　条件

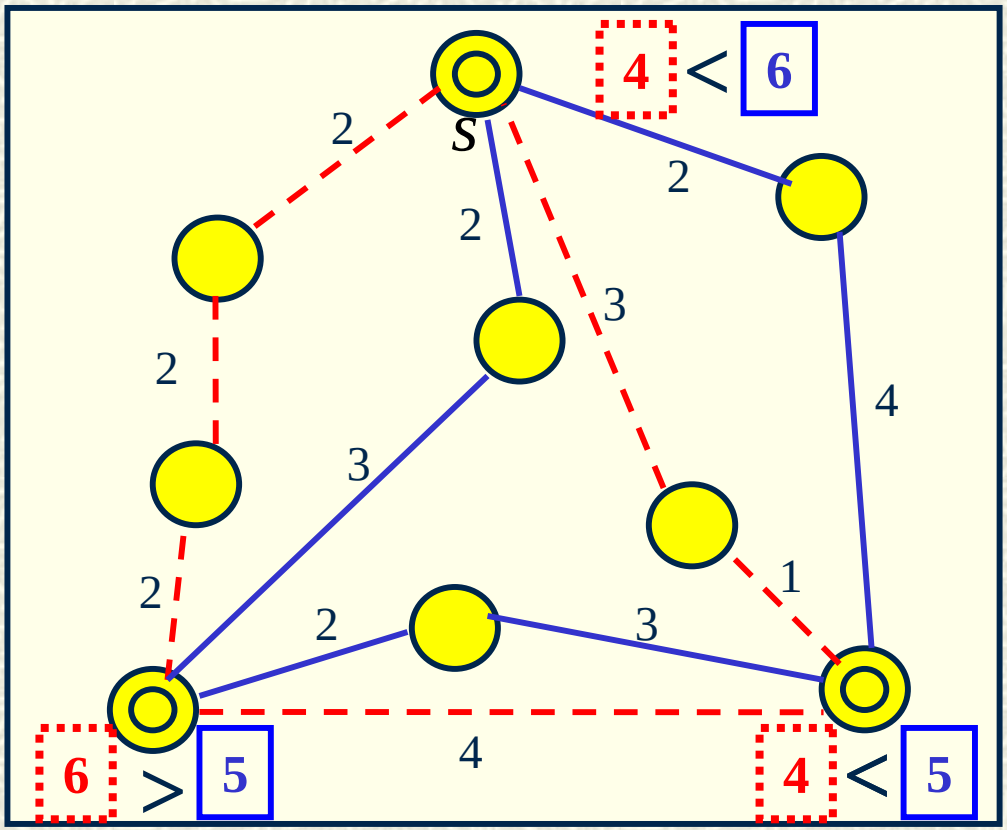
- 2人の移動速度は一定かつ等しい
- 合流点に先に着いた者は、後から到着する者を待つ
- 巡回時間は点間の移動時間のみを考慮する



◎：合流点（2人需要点）
○：単点（1人需要点）

2．3　巡回時間計算

巡回時間 T = 6 + 5 + 6 = 17



経路a：
経路b：

3　定式化

3．1 記号化(1)

$G=(V,E)$:グラフ

$V=\{1,2,\text{☹},n\}$:点集合

E:枝集合

$J=\{1,2,\text{☹},m\}(\subset V)$:合流点集合

$S=V\setminus J$:単点集合

$R=\{1,2\}$:セールスマン集合

3．1　記号化(2)

z : 巡回時間

t_{ij}: 点(i, j)間の移動時間

$$x_{ijkl}^{(p)} = \begin{cases} 1: \text{2人のセールスマンが合流点}k\text{の次に合流点}l\text{を訪問し,} \\ \quad \text{セールスマン}p\text{が合流点}(k,l)\text{間で点}i\text{の次に点}j\text{を訪問する} \\ 0: \text{それ以外} \end{cases}$$

3．2 定式化

$$\min \quad z = \sum_{k\in J}\sum_{l\in J-k} \max_{p=1,2} \sum_{i\in V}\sum_{j\in V-i} t_{ij} x_{ijkl}^{(p)} \qquad (1)$$

$$s.t. \quad \sum_{p=1}^{2}\sum_{k\in J}\sum_{l\in J-k}\sum_{j\in V} x_{ijkl}^{(p)} = 1 \qquad \forall i \in N \qquad (2)$$

$$\sum_{p=1}^{2}\sum_{k\in J}\sum_{l\in J-k}\sum_{i\in V} x_{ijkl}^{(p)} = 1 \qquad \forall j \in N \qquad (3)$$

$$\sum_{p=1}^{2}\sum_{k\in J}\sum_{l\in J-k}\sum_{j\in V} x_{ijkl}^{(p)} = 2 \qquad \forall i \in J \qquad (4)$$

$$\sum_{p=1}^{2}\sum_{k\in J}\sum_{l\in J-k}\sum_{i\in V} x_{ijkl}^{(p)} = 2 \qquad \forall j \in J \qquad (5)$$

$$\sum_{i\in S}\sum_{j\in V\setminus S} x_{ijkl}^{(p)} \geq 1 \qquad \forall S \subset V \quad (S \neq \phi, S \neq V) \quad \forall k,l \in J \qquad (6)$$

$$x_{ijkl}^{(p)} \in \{0,1\} \qquad \forall i, j, k, l \in V \qquad (7)$$

3．2．1　目的関数

$$\min \quad z = \sum_{k \in J} \sum_{l \in J-k} \max_{p=1,2} \sum_{i \in V} \sum_{j \in V-i} t_{ij} x_{ijkl}^{(p)} \qquad (1)$$

3．2．2 制約式(1)

$$\sum_{p=1}^{2}\sum_{k\in J}\sum_{l\in J-k}\sum_{j\in V} x_{ijkl}^{(p)} = 1 \qquad \forall i \in N \qquad (2)$$

$$\sum_{p=1}^{2}\sum_{k\in J}\sum_{l\in J-k}\sum_{i\in V} x_{ijkl}^{(p)} = 1 \qquad \forall j \in N \qquad (3)$$

$$\sum_{p=1}^{2}\sum_{k\in J}\sum_{l\in J-k}\sum_{j\in V} x_{ijkl}^{(p)} = \mathbf{2} \qquad \forall i \in J \qquad (4)$$

$$\sum_{p=1}^{2}\sum_{k\in J}\sum_{l\in J-k}\sum_{i\in V} x_{ijkl}^{(p)} = \mathbf{2} \qquad \forall j \in J \qquad (5)$$

3．2．2 制約式(2)

$$\sum_{i\in S}\sum_{j\in V\setminus S} x_{ijkl}^{(p)} \geq 1 \quad \forall S \subset V \quad (S \neq \phi, S \neq V) \quad \forall k,l \in J \qquad (6)$$

$$x_{ijkl}^{(p)} \in \{0,1\} \qquad \forall i, j, k, l \in V \qquad (7)$$

4. 解法

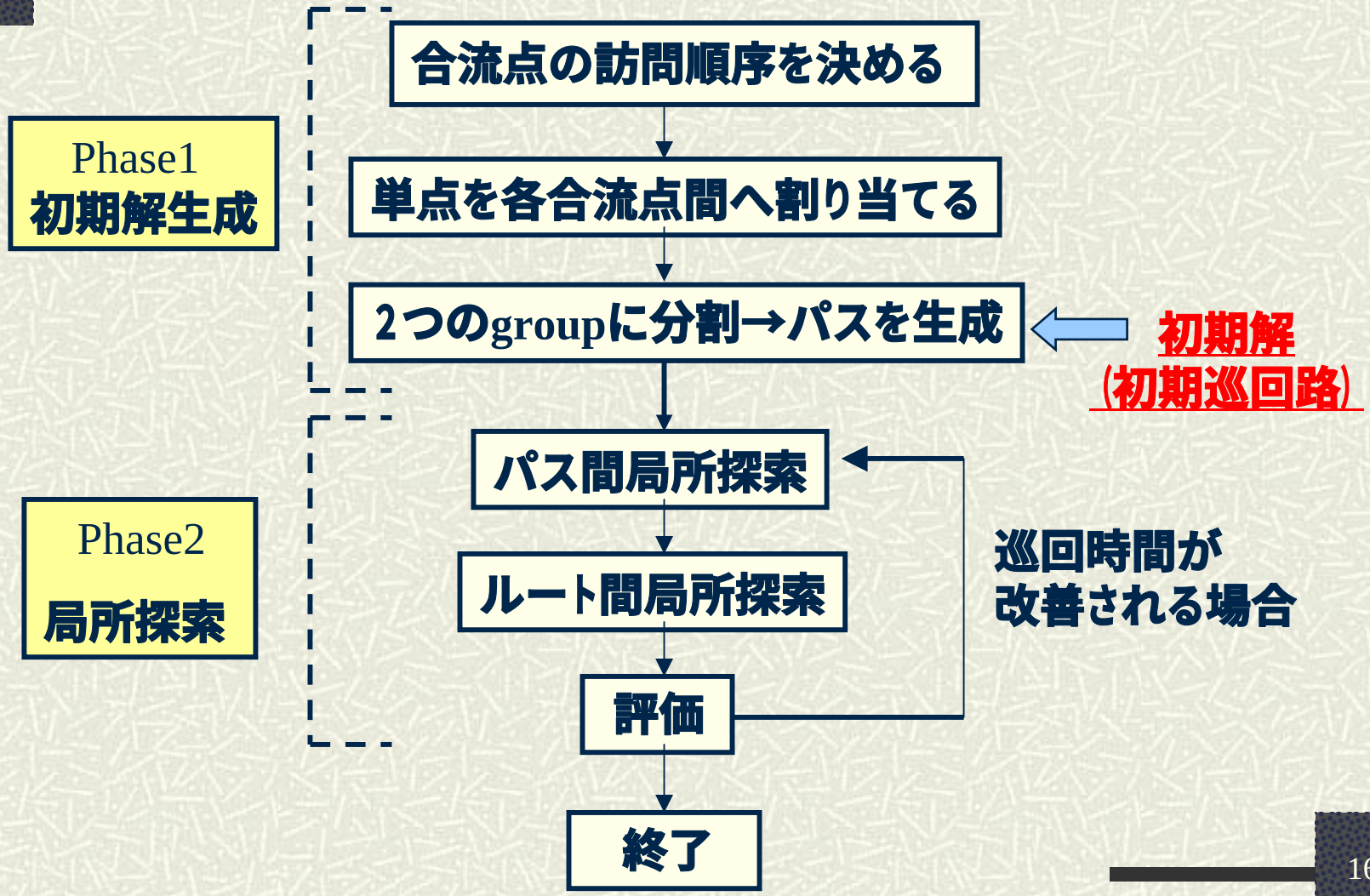
4．1　解法の概要(1)

- **提案解法は大別して2段階で構成される**

Phase1：**初期解を求める**

Phase2：**2つの局所探索法によって解を改善する**

4．1 解法の概要(2)

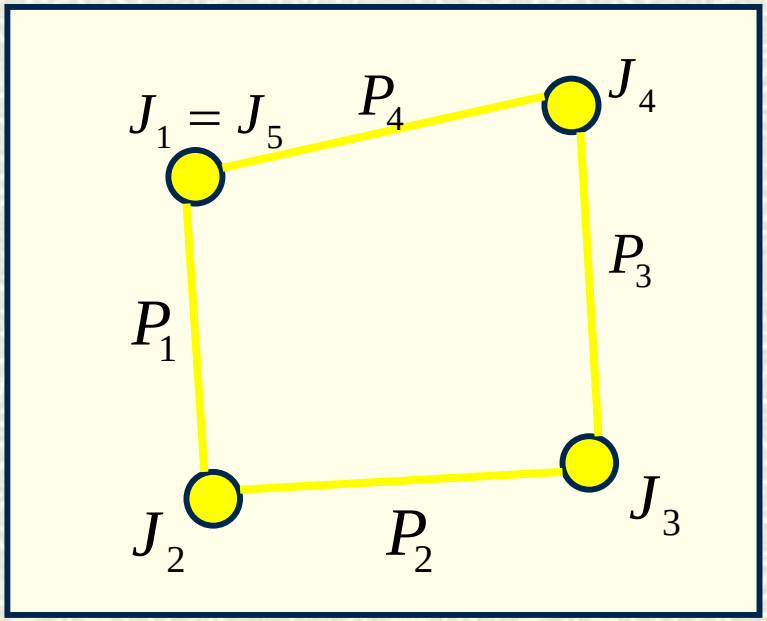

Phase1
初期解生成
合流点の訪問順序を決める
単点を各合流点間へ割り当てる
2つのgroupに分割→パスを生成
初期解
(初期巡回路)
Phase2
局所探索
パス間局所探索
ルート間局所探索
評価
巡回時間が
改善される場合
終了

4．2 Phase1

4．2．1 合流点の訪問順序の決定

Step1 合流点の訪問順序の決定

：全数列挙法により、m個の合流点で最短巡回路を求める

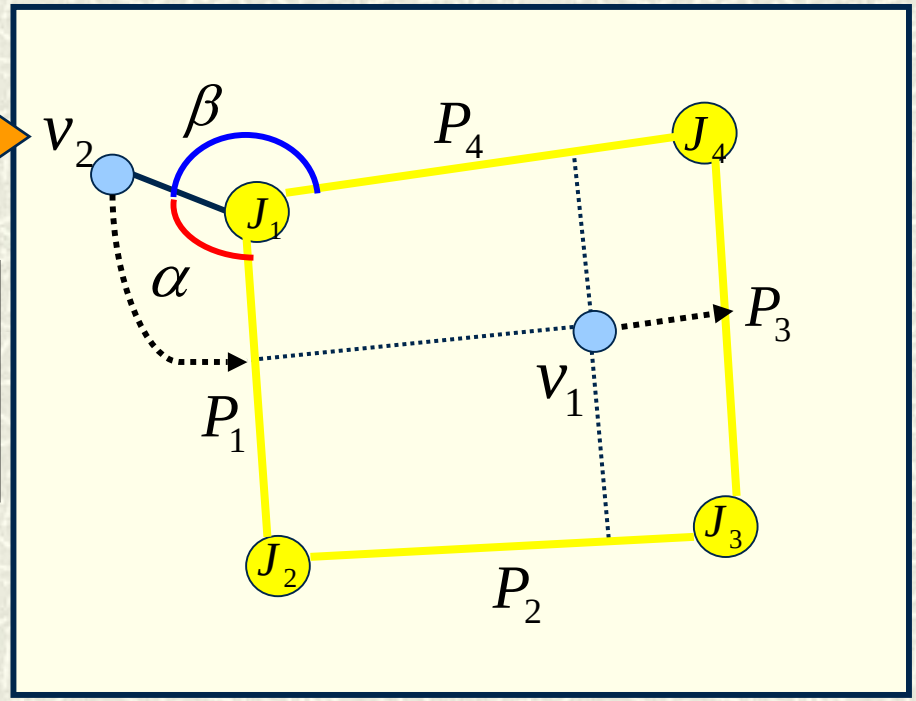


- **任意に起点** $J_1 (= J_{m+1})$ **を定め、**
  そのほかの合流点を巡回路に従い、
  $J_2, J_3, \cdots, J_m$ **とする**
- J_k **と** J_{k+1} **をつなぐ線分を** P_k **とする**

4．2．2　単点の初期割当

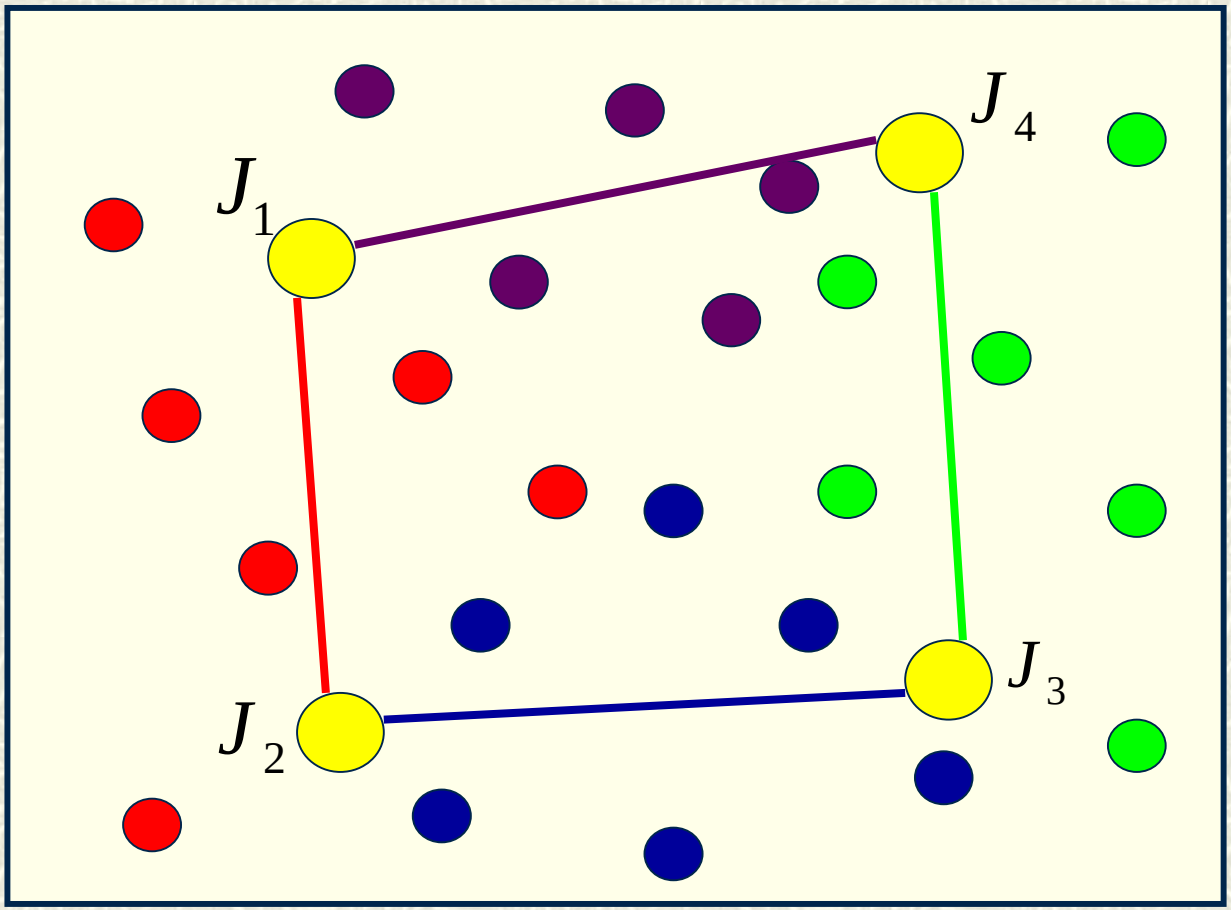
Step2　単点を各合流点間へ割り当てる

：単点と合流点間を結ぶ線分との距離を計算し，単点iがP_kと最も近い場合，iを合流点(J_k, J_{k+1})間に割り当てる



$\alpha \leq \beta$なら、
(J_1, J_2)間に
割り当てる

＜割当後＞



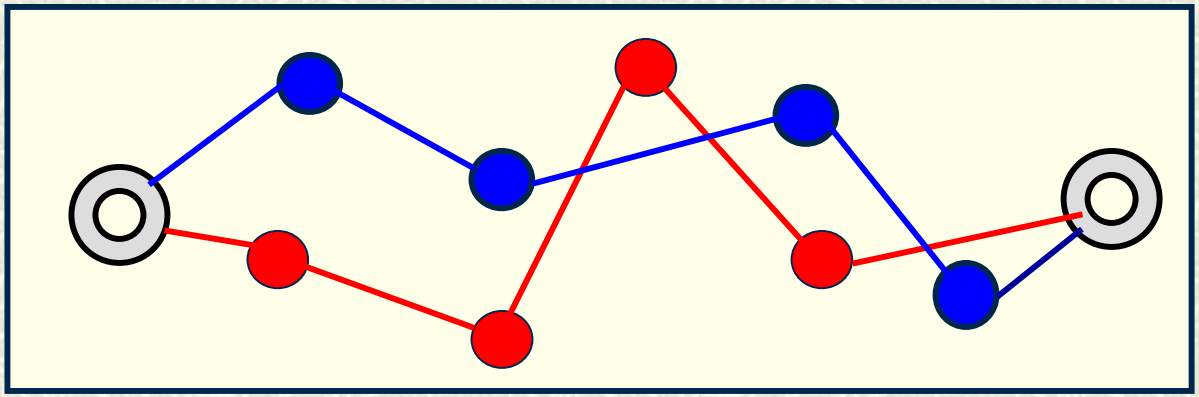
4．2．3　初期解生成 (1)

Step3　初期解生成

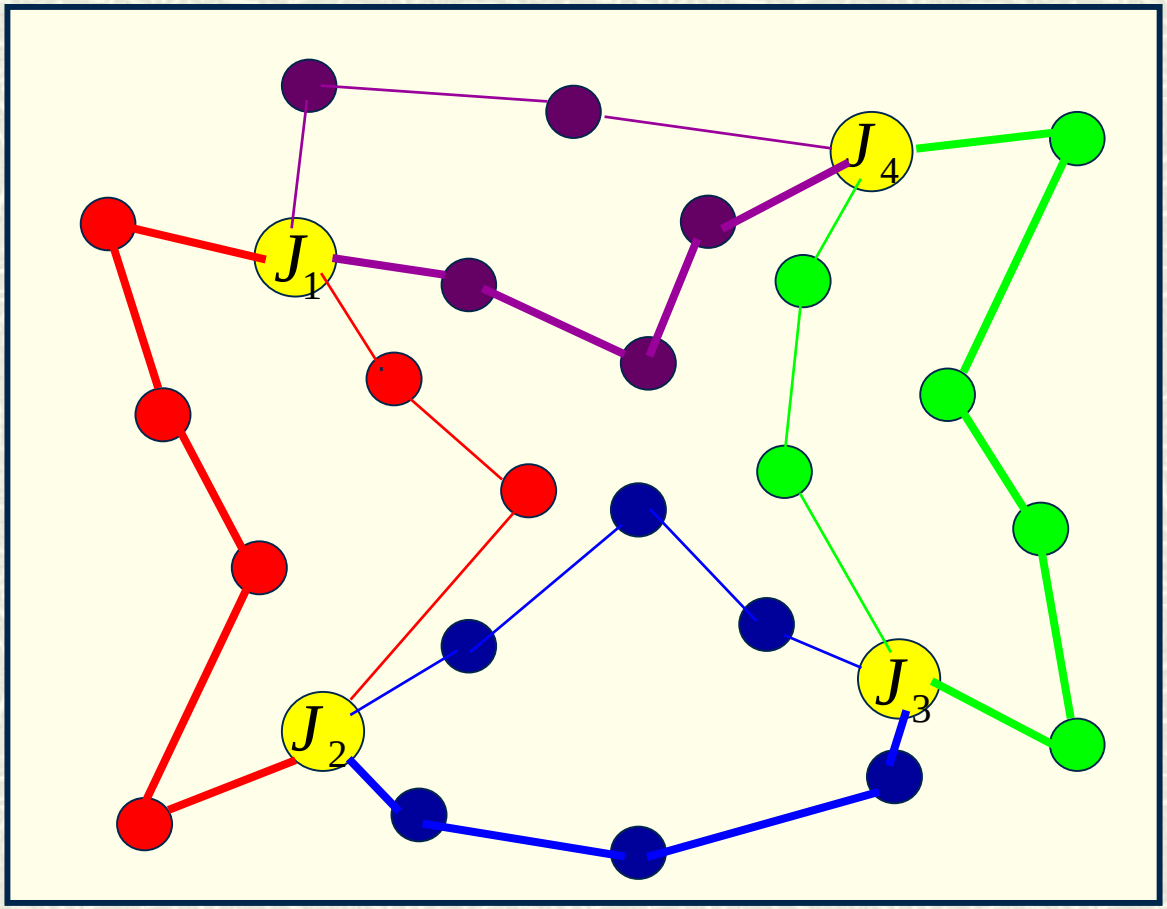
：Step2で各合流点間に割り当てられた単点集合G を2つの単点集合g_1, g_2 へ分割し、パスを生成する

パス：**2つの合流点を端点とし、$g_1(g_2)$ に属する単点をすべて通過する最短経路 (2-opt法によって近似的に求める)**

2つのパスをあわせてルートと呼ぶ



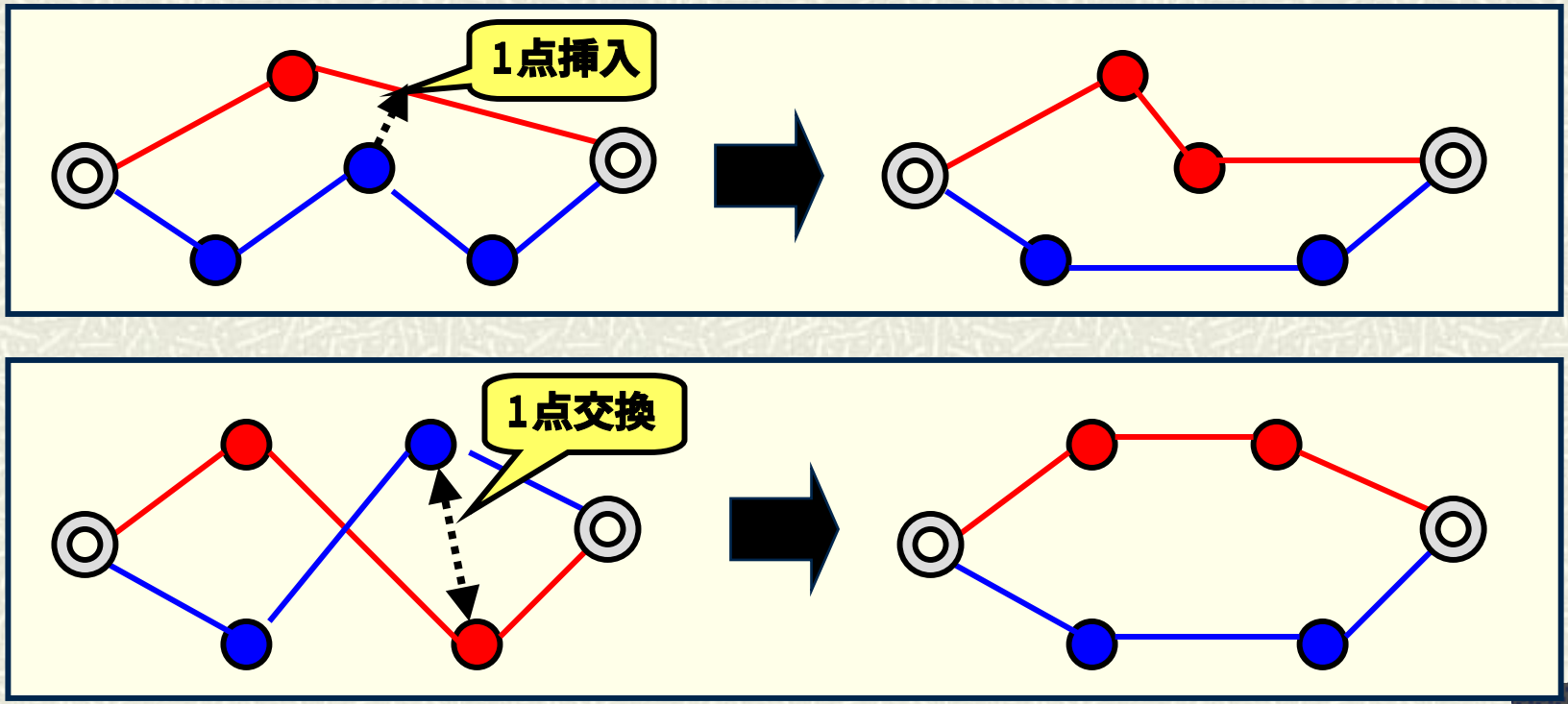
4．2．3　初期解生成 (2)



4．3　Phase2

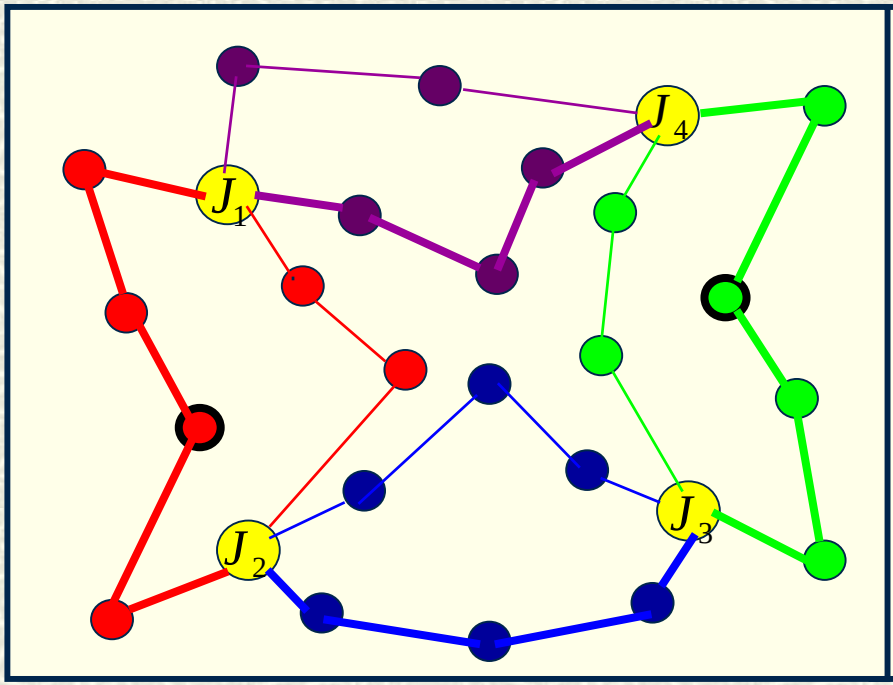
4．3．1　パス間局所探索 (1)

パス間局所探索：2つのパス間で単点を移動することで、
その合流点間の移動時間を改善する局所探索

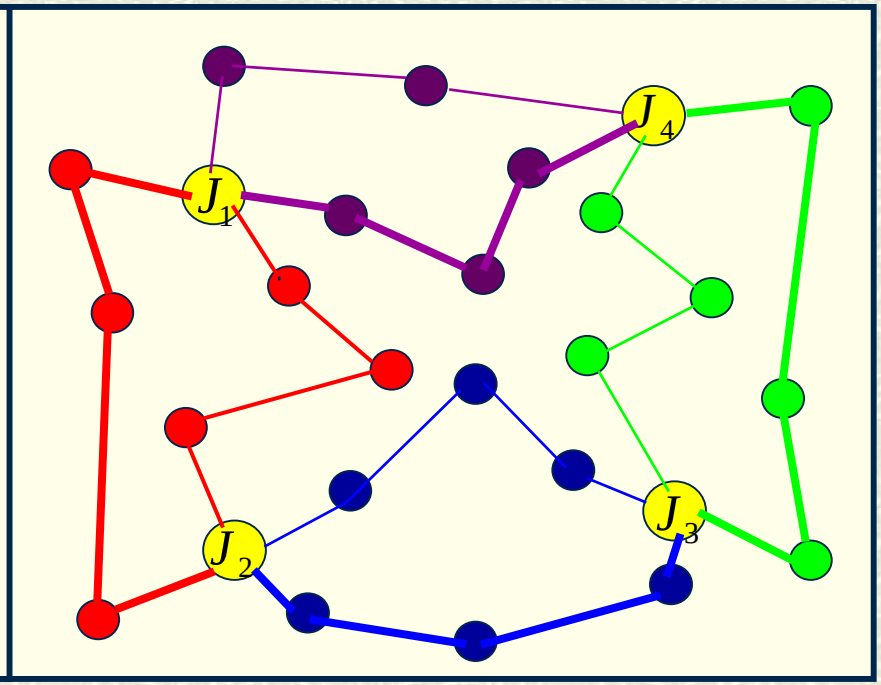


4．3．1　パス間局所探索 (2)

＜初期解＞



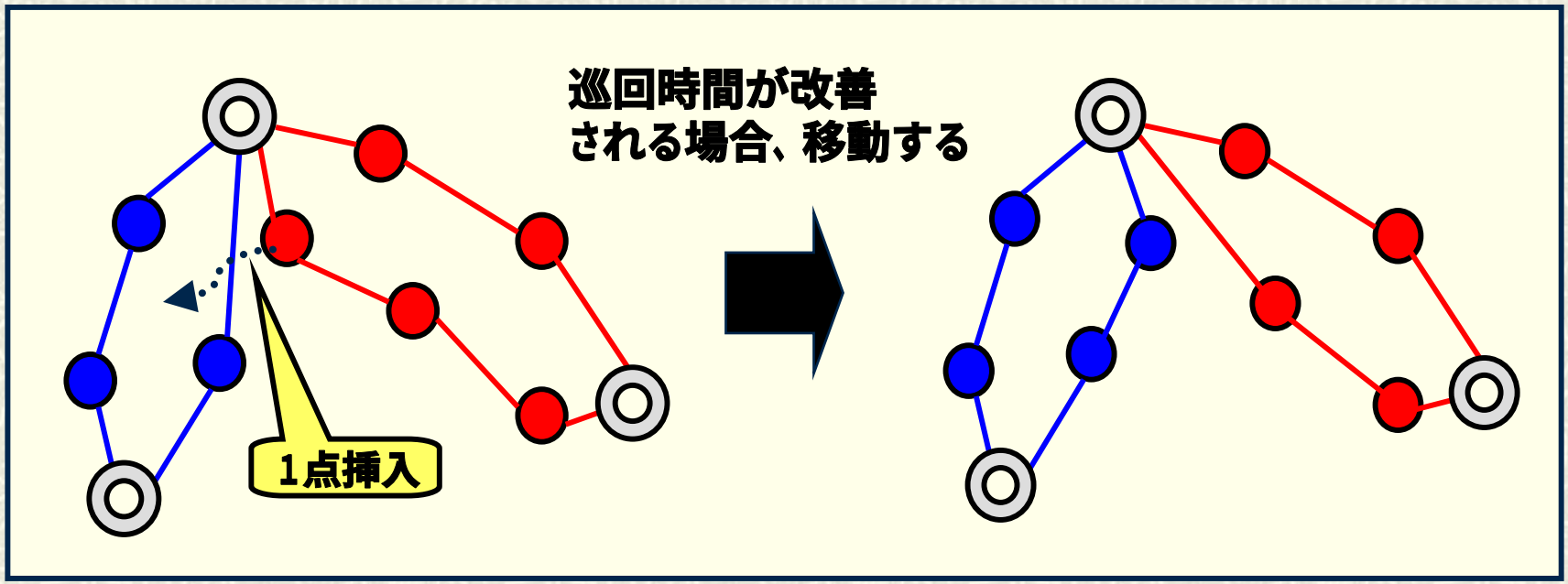
＜パス間局所探索後＞



4．3　Phase2

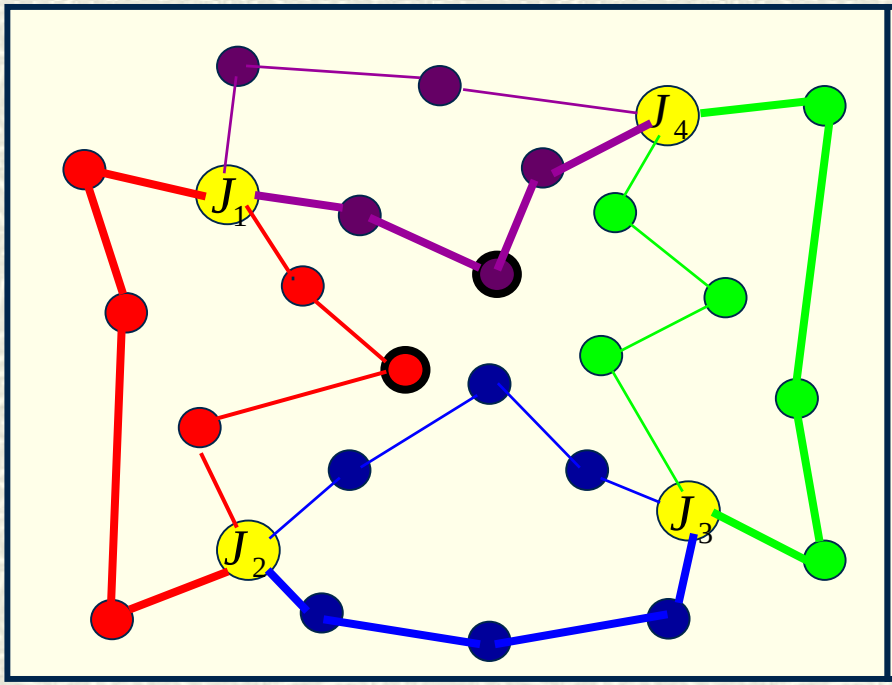
4．3．2　ルート間局所探索 (1)

ルート間局所探索：ルート間で単点の移動を行い、
巡回時間を改善する局所探索

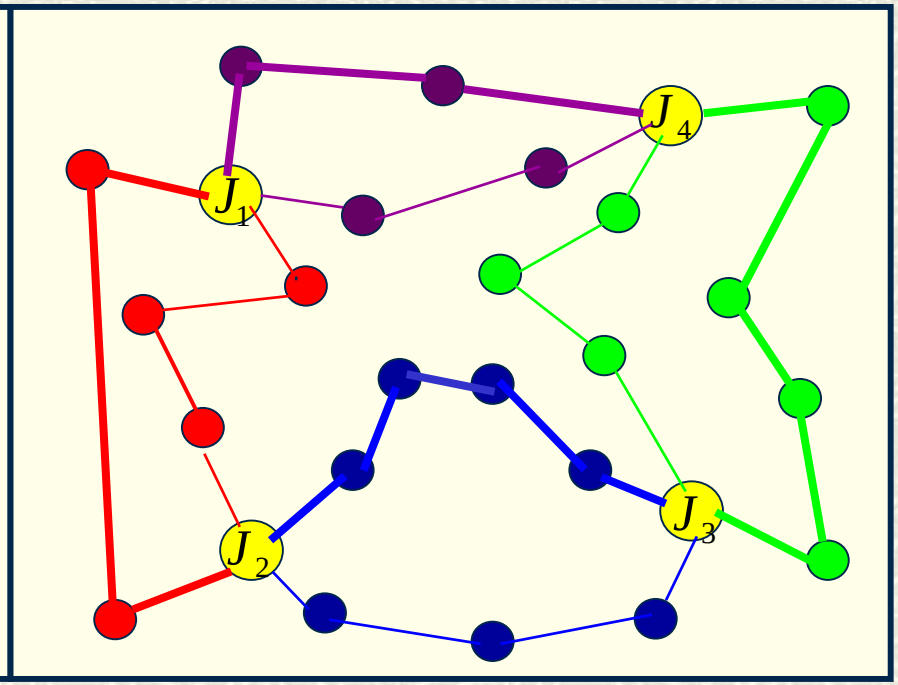


4．3．2　ルート間局所探索 (2)

＜パス間局所探索後＞



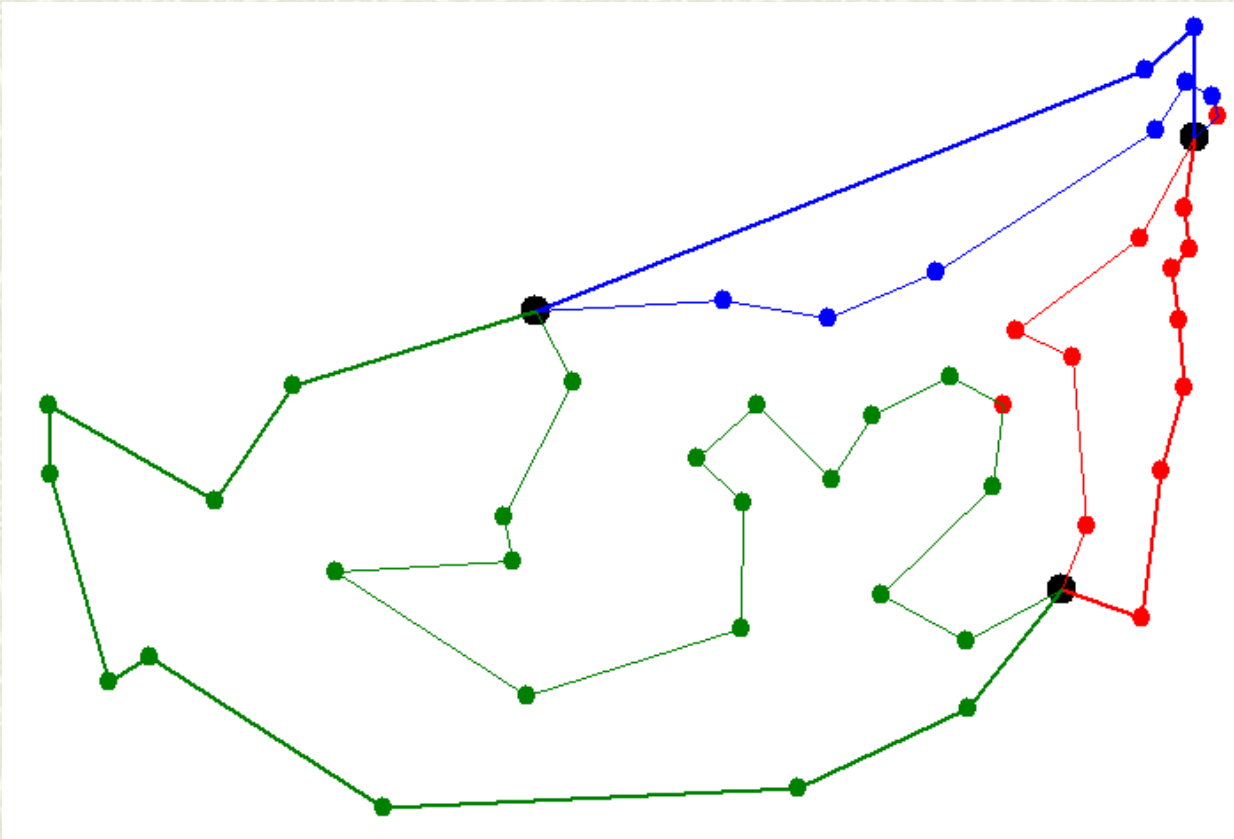
＜ルート間局所探索後＞



5．実験結果

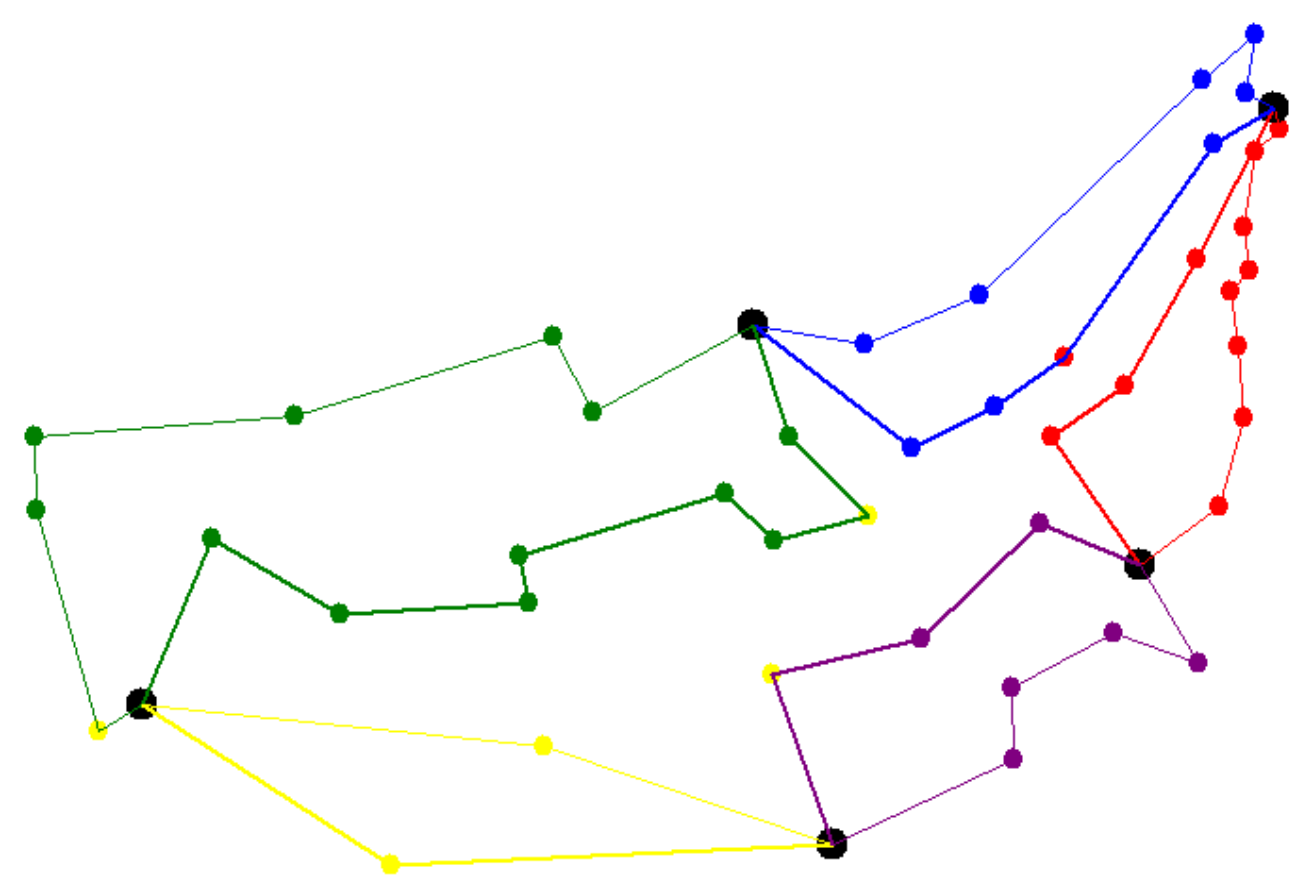
- 提案解法をDelphi6によって実装した
- 点データとして、TSPLIB[2]からatt48,gr96を用いた
- 3～10個の点をランダムに合流点に設定し、実験を行った

5．1　巡回の様子(1)



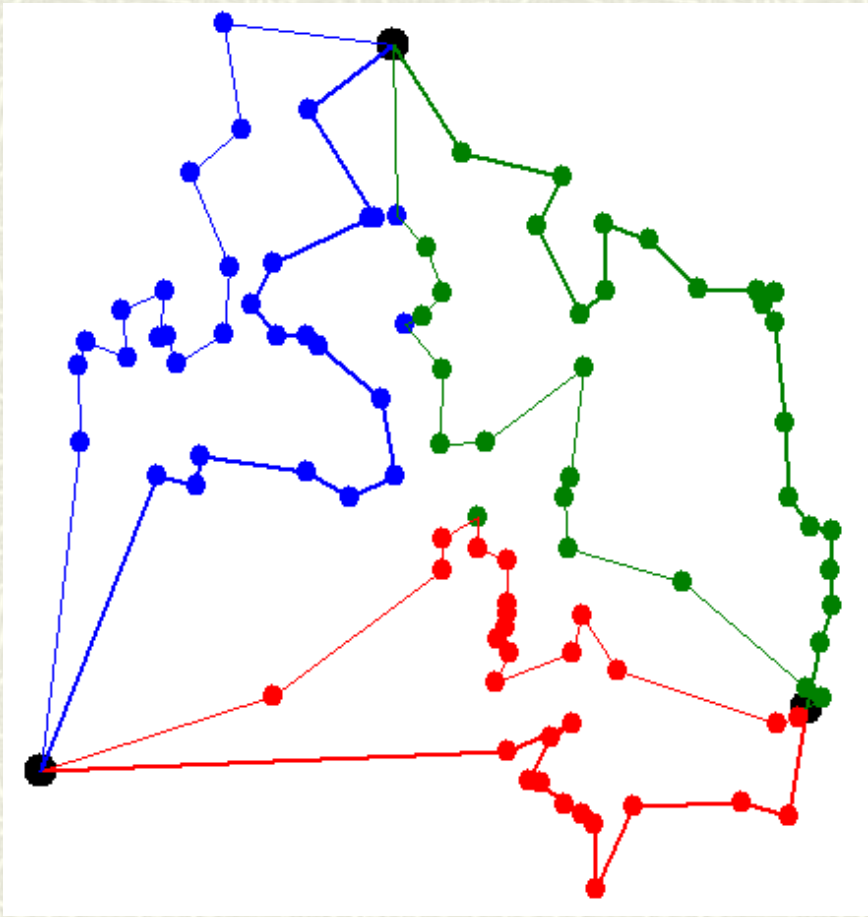
att48(m=3)

5．1　巡回の様子(2)



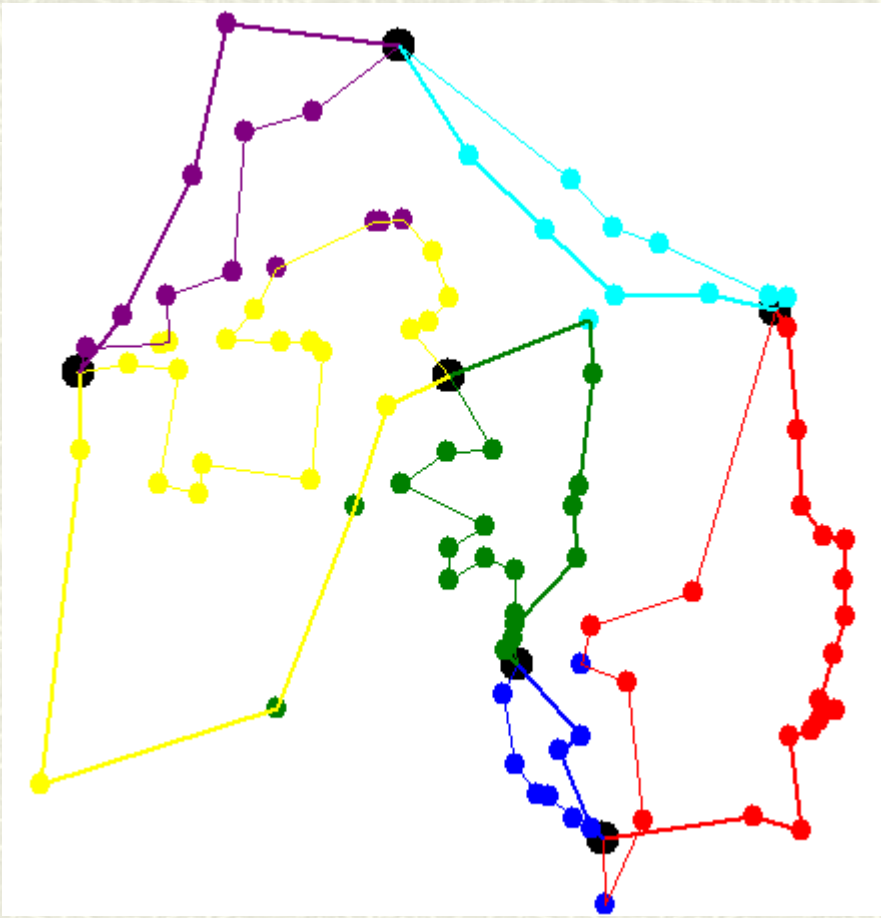
att48(m=5)

5．1　巡回の様子(3)



gr96(m=3)

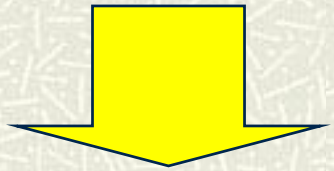
5．1　巡回の様子(4)



gr96(m=6)

5．2　考察 (1)

- 2本のパスの移動時間はほぼ等しい
- 他の合流点間のパスと交わることなく巡回している



・解法の方針は間違っていない

・アルゴリズムは正しく働いている

5．3　実験結果

DATA	m	巡回時間	計算時間 (秒)
att48	3	1886	11
	5	1998	2
gr96	3	1752	224
	6	1808	90

5．4　考察 (2)

同じ点集合データを扱うとき，

- 多くの場合，巡回時間は合流点数にほぼ比例して増加する
  →2人の訪問点数の和は n (点数) +m (合流点数) なので、
  mが増加すると、巡回時間が増加する

- 2本のパスが交差する結果が生じた
  →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

- 計算時間は合流点が多いときのほうが短い
  →各合流点間に割り当てられる単点の数が少なく、パス間
  局所探索における計算時間が短いため

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
- 既存問題に帰着できない現実の問題を扱うことが出来るようになる

7. 今後の課題

- 提案した解法は、局所探索を中心とした基本的な方法であり、今後様々な観点からのアプローチを試みる必要がある

- 問題の応用を考える

  (ex1) 2人のセールスマンを区別する (役割が異なる)

  (ex2) セールスマンの人数を増やす

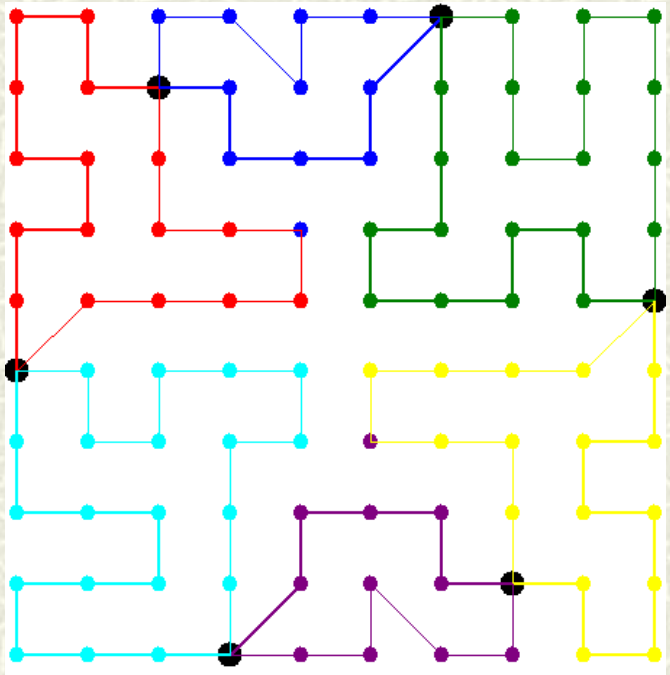
参考文献

[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)

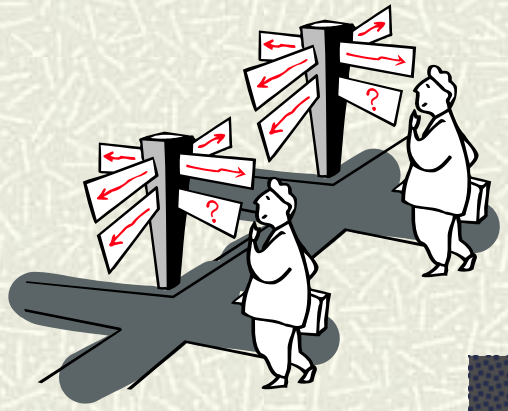
ご清聴ありがとうございました



＜付録1＞RTSPに帰着可能な現実問題

■RTSPはどのような現実問題に当てはまるか？

①パトロール経路決定問題
②訪問介護経路問題
（ex. 医師と看護師が合流. 合流点：問診と看護が必要な患者宅）
③同じ地点を経由する循環型バス経路問題



＜付録2＞全数列挙法による厳密解求解の可否

■組合せ最適化問題において、厳密解を得るには全ての組合せを列挙し、その中の最小（最大）の評価値を構成するものを解とすればよい

→組合せ最適化問題の多くは、決定変数の増加により，組合せ数が爆発的に増加し，現実的な時間内での計算が不可能

RTSPの組合せ数を一般化し，

全数列挙が可能かどうか考える

＜付録3＞組合せ数

■組合せ数は、以下の3つの組み合わせの積で表される

①合流点の訪問順序：$\frac{(m-1)!}{2}$

②1人訪問点の各合流点間経路への割当：$(2m)^{(n-m)}$

③1人訪問点の訪問順序：　P $(\geqq 1)$ とする

組合せ数　＝ $\frac{(m-1)!}{2}$ ・$(2m)^{(n-m)}$・P　＞ $\frac{(n-1)!}{2}$

→ＴＳＰの組合せ数より多い（計算量が多い）

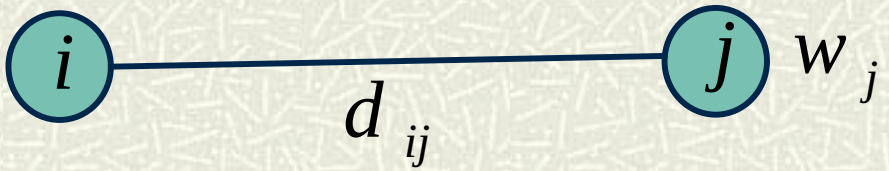
→列挙法による厳密解求解は難しい

＜付録4＞　各点の滞在時間を枝のコストに組み込む方法

d_{ij}：点(i, j)間の移動時間

w_j：点jにおける滞在時間

c_{ij}：点iの次に点jを訪問するときに費やす時間



$$\text{点}j\text{が}\begin{cases}\text{合流点のとき} & : c_{ij} = d_{ij} \\ \text{合流点でないとき} & : c_{ij} = d_{ij} + w_j\end{cases}$$

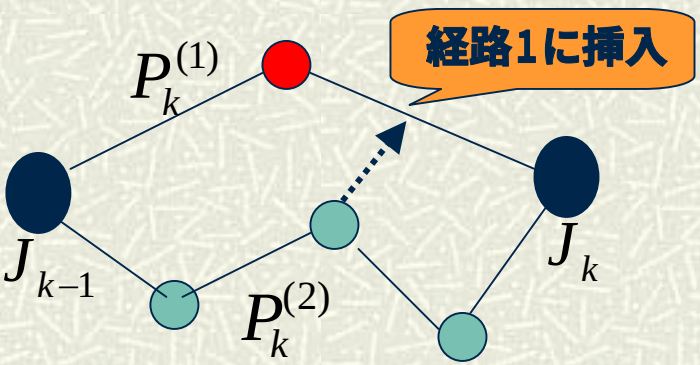
以上のようにc_{ij}を定義することで、各点に滞在する場合でも枝のコストのみで巡回時間を表すことができる

＜付録5＞　近傍探索(1)

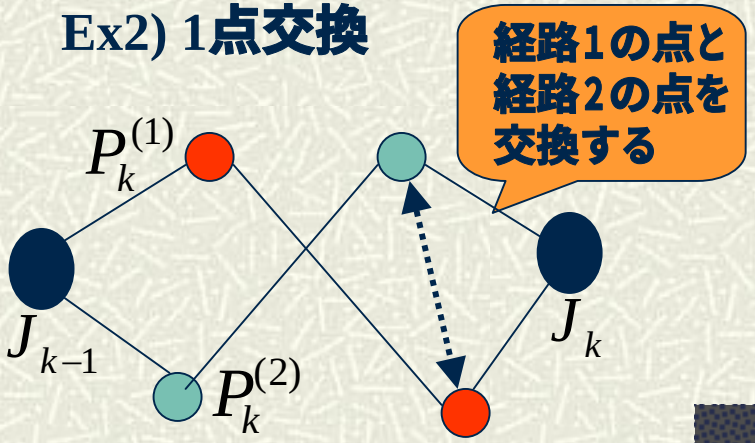
- **近傍探索：現解の近傍を探索する(現解に，局所的な変形を加える)ことで，解を改善する方法**

- **提案解法における近傍を，**
  「現解に対して，任意の合流点間の2path間での点の移動(1点挿入or1点交換)によって得られる解」とする．

Ex1) 1点挿入

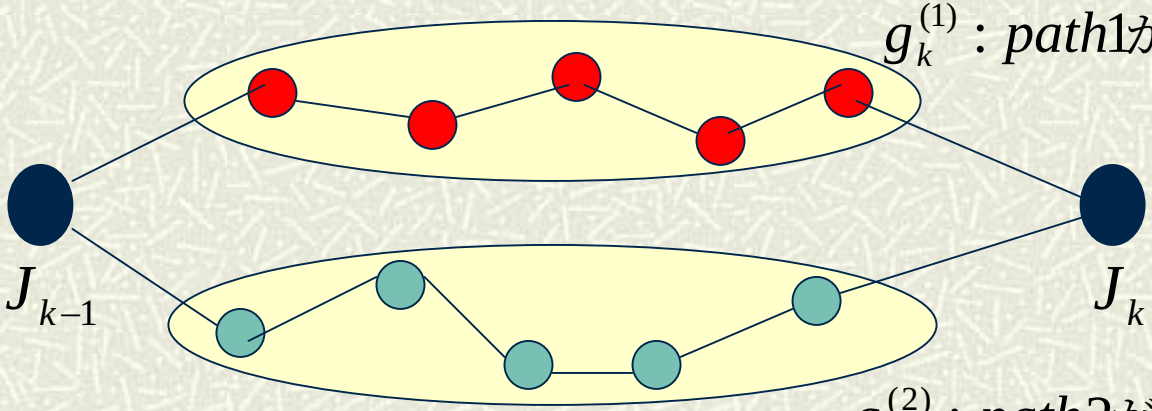


Ex2) 1点交換



＜付録5＞ 5．4．2 近傍探索(2)

$V_k = (g_k^{(1)}, g_k^{(2)})$を満たすように分割する



$t(g_k^{(p)})$：$g_k^{(p)}$の最短経路の移動時間

$$f(g_k^{(1)}, g_k^{(2)}) = \max_{p=1,2} t(g_k^{(p)})$$

＜付録5＞ 5．4．2 近傍探索(3)