

未踏ターゲット事業 成果報告会

アニメーリングマシンを用いた 配送計画最適化技術の開発

2021年2月11日

お茶の水女子大学 大石美賀・松本奈紗

田村PM

1. 概要

2. 車両への荷物の割り当て

3. ルート最適化

4. アプリケーション

5. まとめと今後の展望

プロジェクトの背景

運送業の人手不足

- ネットショップの普及
- 少子高齢化
- 宅配便の多様化による集積率の低さ

配送計画問題の複雑化

- 荷物の大きさや冷凍便
- 時間指定や宅配ボックス
- 配送と集荷の同時進行

アニーリングマシンを用いて
宅配便の配送計画の最適化を行うアプリケーションを開発

配達の現状

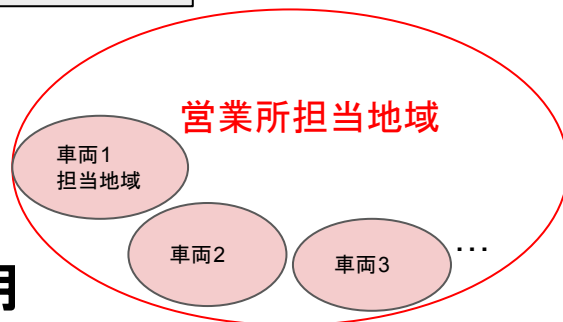
- **1日3回 (朝・昼・夕方) に分けて営業所に荷物が届き、約6割が朝に届く**

- 4輪車両は 8~14時・14~18時・18~21時の3回に分けて営業所から出発
- 使用できる車両の最大数は決められているため、忙しさを分散させたい



- **配達地域が固定**

- 各車両ごとに固定の配達地域が決められている
- 地域を可変にすれば、さらなる効率化が望める



- **配送会社や配達地域によっては車両のみを使用**

- 去年のプロジェクト: 都心部の一部地域で、途中で車両を駐車して台車を使って配達
- 配送会社や配達地域によっては車両のみを使用する場合も多い

プロジェクトの概要

目標: 汎用的な**営業所全体**の配送計画の最適化

1

車両への荷物の割り当ての最適化

2

地域に合ったアルゴリズムを用いてルートを最適化

昨年度のプロジェクト



配達先が過密している地域



車両のみを用いたルート算出



都心部や地方など

1. 概要

2. 車両への荷物の割り当て

3. ルート最適化

4. アプリケーション

5. まとめと今後の展望

車両への荷物の割り当ての流れ

アニーリングマシンを使用

① 午前中に届いた時間指定なしの荷物を
3つの配達時間帯に分ける



② 荷物を各車両に分ける



8~14時に配達する荷物

14~18時に配達する荷物

18~21時に配達する荷物

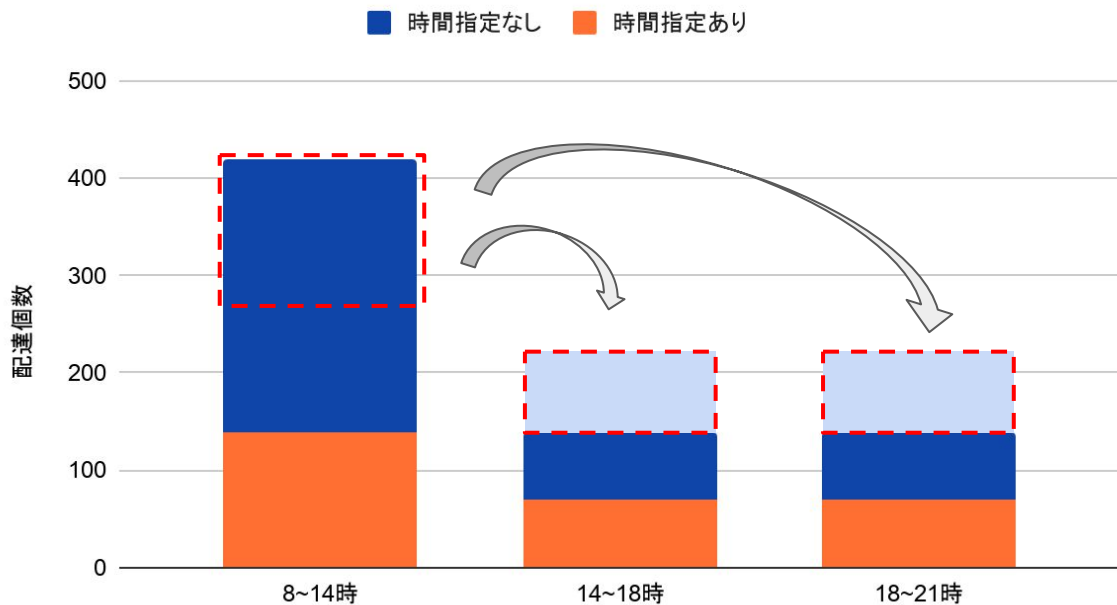
車両①の荷物

車両②の荷物

車両③の荷物

⋮

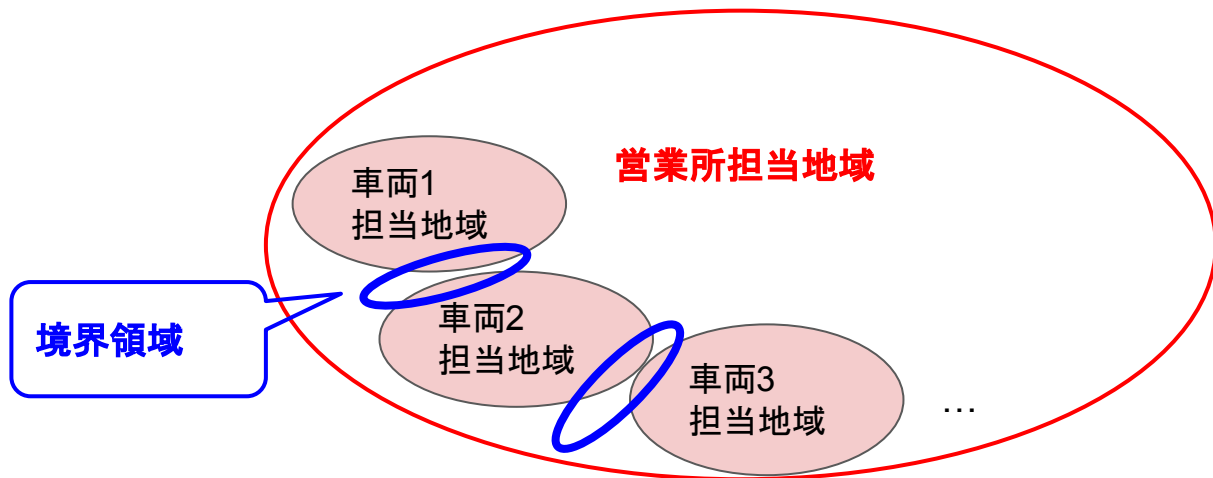
① 午前中に届いた時間指定なしの荷物を3つの配達時間帯に分ける



8~14時の指定なしのうち、
14~18時 or 18~21時に指定ありの配達先近くのものを移動

② 荷物を各車両に分ける

アニーリングマシンを使用



荷物をバランスよく車両に割り当てるために
位置関係と配達個数を考慮して、境界領域の荷物をクラスタリング

定式化 (コミュニティ検出を用いたクラスタリング手法を考案)

境界領域の配達先同士で近くのものと同じグループになるようにするため

$$H = - \sum_{g=1}^G \sum_{i < j}^N k_g \exp(-d_{ij}/\sigma) x_{i,g} x_{j,g} + \sum_{g=1}^G \sum_{i < j}^N \gamma x_{i,g} x_{j,g} + H_{p1}$$

近くの頂点を同じグループにする傾向

$$H_{p1} = \alpha \sum_{i=1}^N \left(\sum_{g=1}^G x_{i,g} - 1 \right)^2$$

制約項: 各配達先の所属車両は1つ

グループ数を増やす傾向

G : 車両数

d_{ij} : 配達先 i, j 間の移動コスト

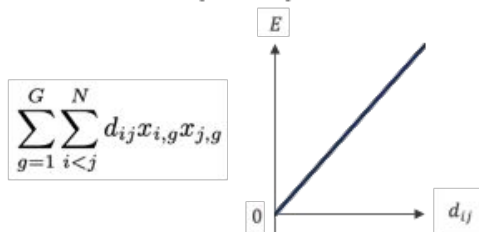
$x_{i,g}$: 配達先 i が車両 g に属する場合1, 属さない場合0

α, γ, σ : 正のパラメータ

k_g : コスト関数の深さを決めるパラメータ

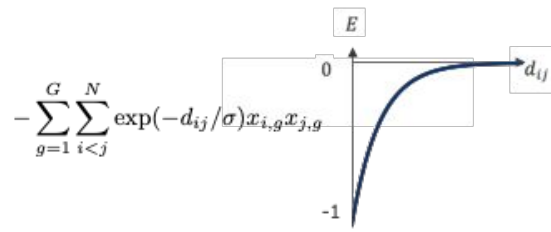
ただし、境界領域以外の配達先は $x_{i,g}$ が定数となる

一般的な手法 (手法1)



2点間が遠いほどエネルギーが上がる

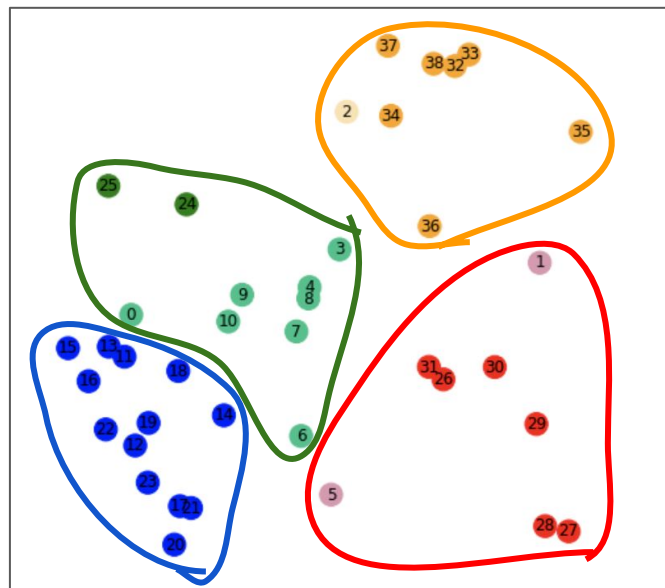
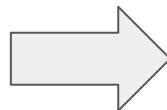
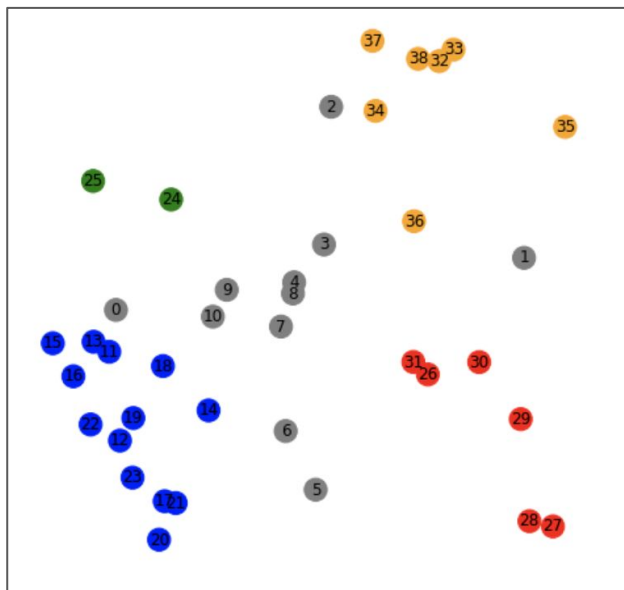
コミュニティ検出を用いた手法



2点間が近いほどエネルギーが下がる

小規模データでの実行結果

富士通デジタルアニーラ(DA)を使用

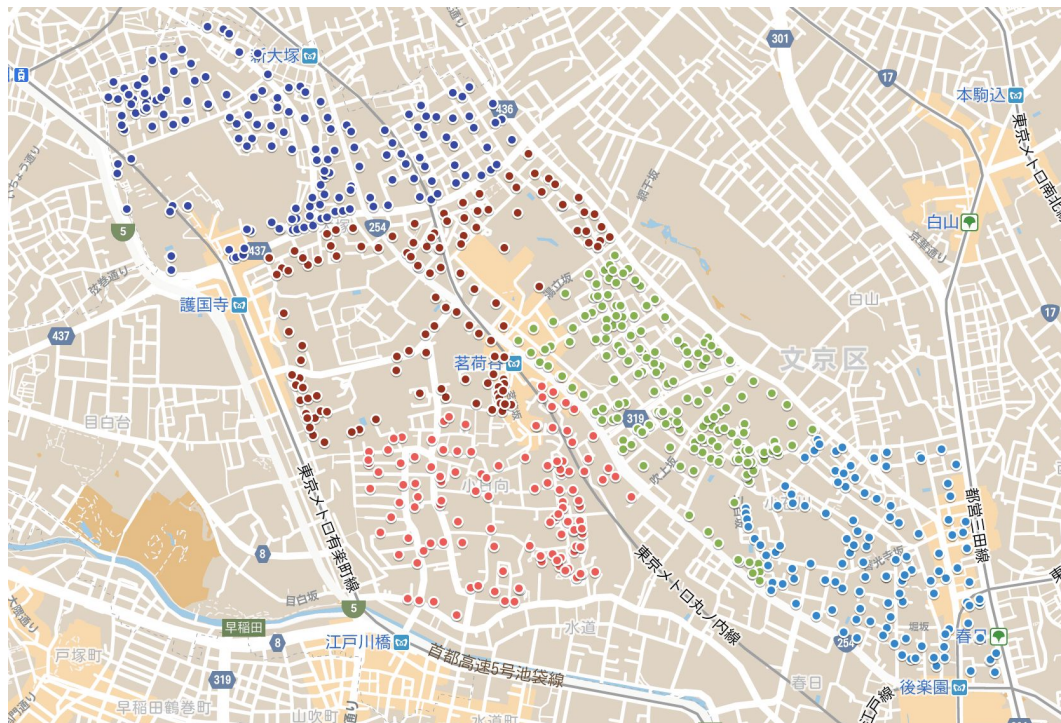


●: 境界領域の荷物

仮想データを用いた実行

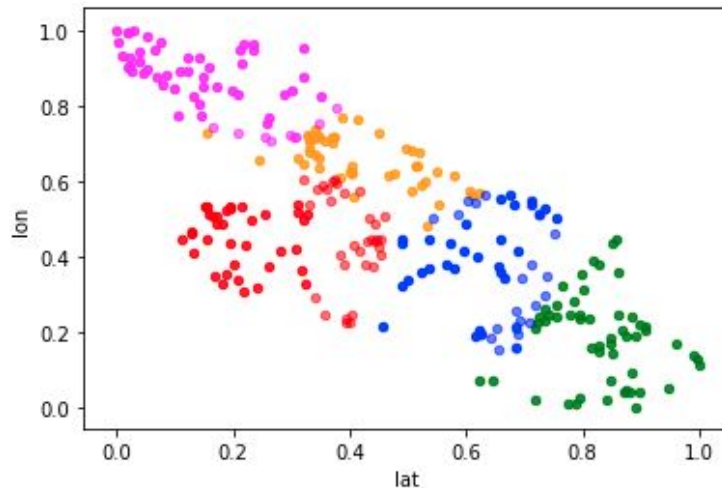
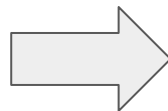
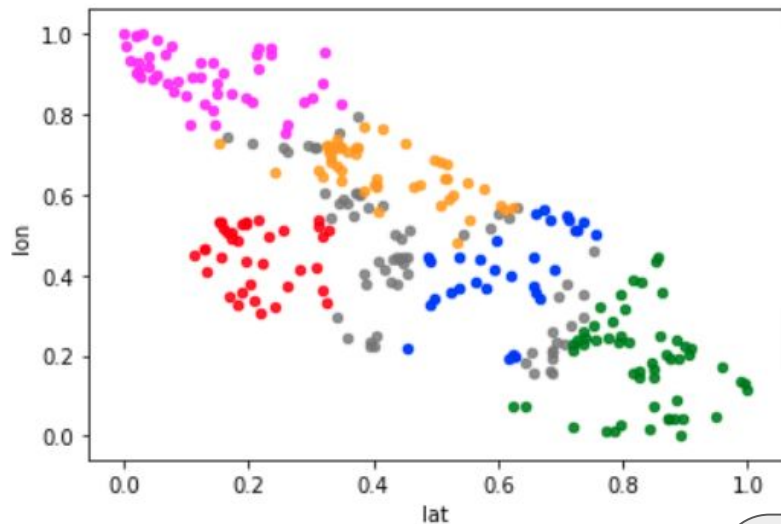
【使用した仮想データ】

- 1日の荷物個数:700個
- 8~14時の荷物個数:280個
- 配達地域:文京区大塚周辺
- 2点間の距離:直線距離
- 車両数:5台



仮想データを用いた実行 (DAを使用)

8~14時の配達データ

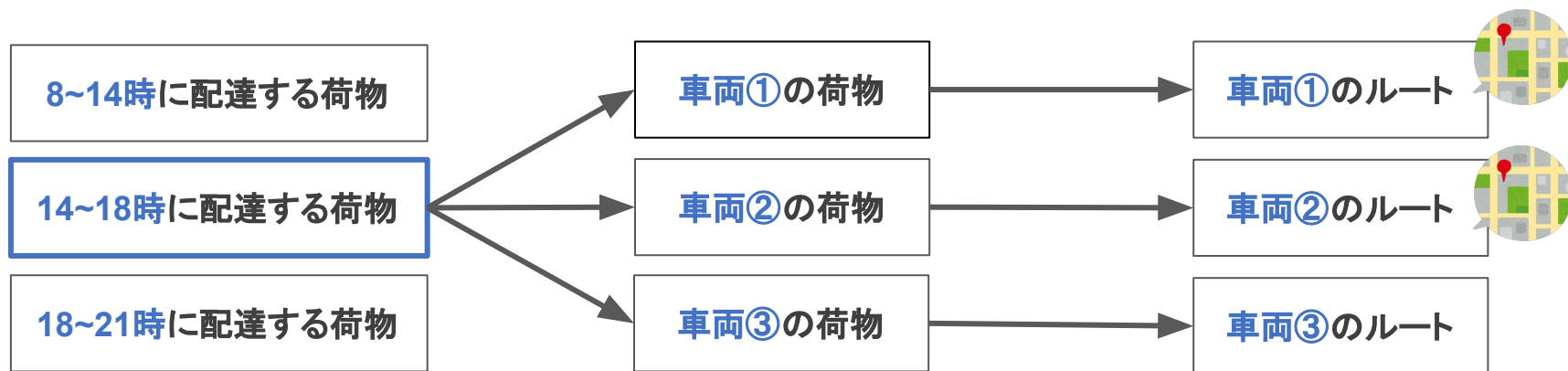


●:境界領域の荷物

車両1の荷物個数: 53 → 53
車両2の荷物個数: 65 → 54
車両3の荷物個数: 43 → 69
車両4の荷物個数: 69 → 48
車両5の荷物個数: 50 → 56
標準偏差 : 9.6 → 7.0

1. 概要
2. 車両への荷物の割り当て
3. ルート最適化
4. アプリケーション
5. まとめと今後の展望

3. ルート最適化



⋮

地域にあったアルゴリズムを用いてルート最適化

昨年度のプロジェクト

配達先が過密している地域
(大学など)

車両のみを使用したルート算出



都心部や地方など汎用的に使用

ルート最適化

1

配達先をクラスタリング

2

グループ間のTSPを解く.

3

グループ内の始点と終点の候補を決める.

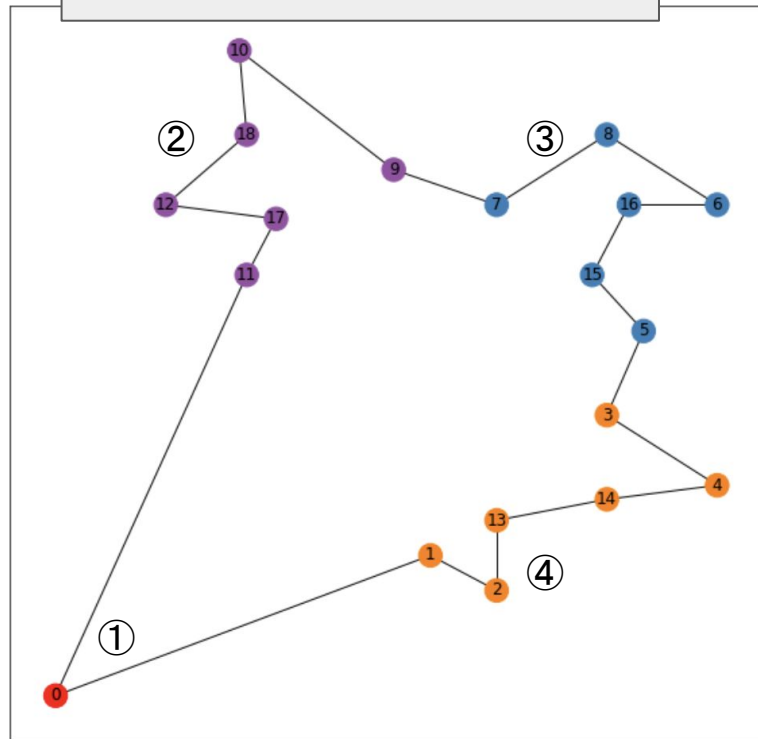
4

始点と終点を変えてグループ内のTSPを解く.

5

どの始点と終点を使うのかを決める.

車両のみを用いたルート算出
配達先数とビット数の問題から
5段階に分けて解く



配達先をクラスタリング

手法1 (一般的な手法)

各グループの頂点数がほぼ同じになりやすいため一様分布に適している

$$H = \sum_{g=1}^G \sum_{i < j}^N d_{ij} x_{i,g} x_{j,g} + \alpha \sum_{i=1}^N \left(\sum_{g=1}^G x_{i,g} - 1 \right)^2$$

コスト項: 移動コストの合計

制約項: 各配達先の所属グループは1つ

d_{ij} : 配達先 i, j 間の移動コスト

$x_{i,g}$: 配達先 i がグループ g に属する場合1, 属さない場合0

N : 配達先数 G : グループ数 α : 正のパラメータ

グループに含まれる点の間の移動コストを足し合わせているため、グループ内の足し合わせる数が2乗で大きくなる。

→ 各グループに含まれる点の数が同一に近くなるような作用が移動コストより強く生じる。

配達先をクラスタリング

手法2 (本プロジェクトで考案)

ばらつきのある分布に対応するため

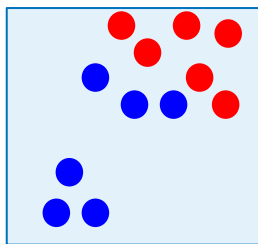
$$H = \sum_{g=1}^G \frac{\sum_{i<j}^N d_{ij} x_{i,g} x_{j,g}}{N_g(N_g - 1)} + \alpha \sum_{i=1}^N \left(\sum_{g=1}^G x_{i,g} - 1 \right)^2$$

コスト項: 平均移動コストの合計

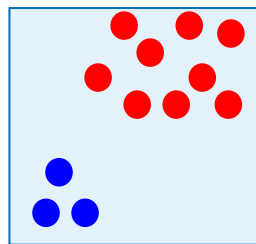
制約項: 各配達先の所属グループは1つ

 $N_g = \sum_{i=1}^N x_{i,g}$: グループ g に属する頂点数

⇒ 移動コストの合計を、各グループに属する頂点数の組合せ数 $N_g(N_g - 1)/2$ で割ることで、各グループの平均移動コストを計算



手法1



手法2

1

配達先をクラスタリング

2

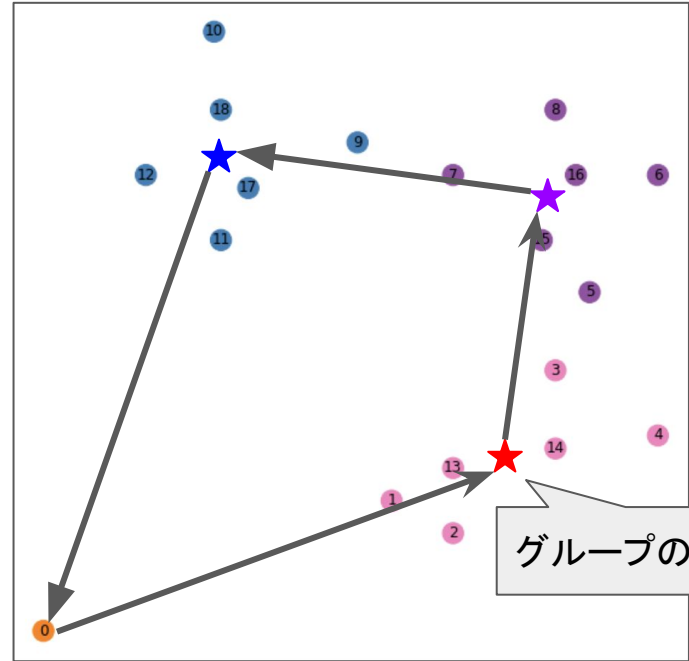
グループ間のTSPを解く.

配達先の分布により
手法1/手法2を使用

各時刻で一箇所にいる

$$H = \sum_{i,j=1}^G \sum_{t=1}^G d_{ij} x_{i,t} x_{j,t+1} + H_{p1} + H_{p2}$$

各グループに一度だけ訪問

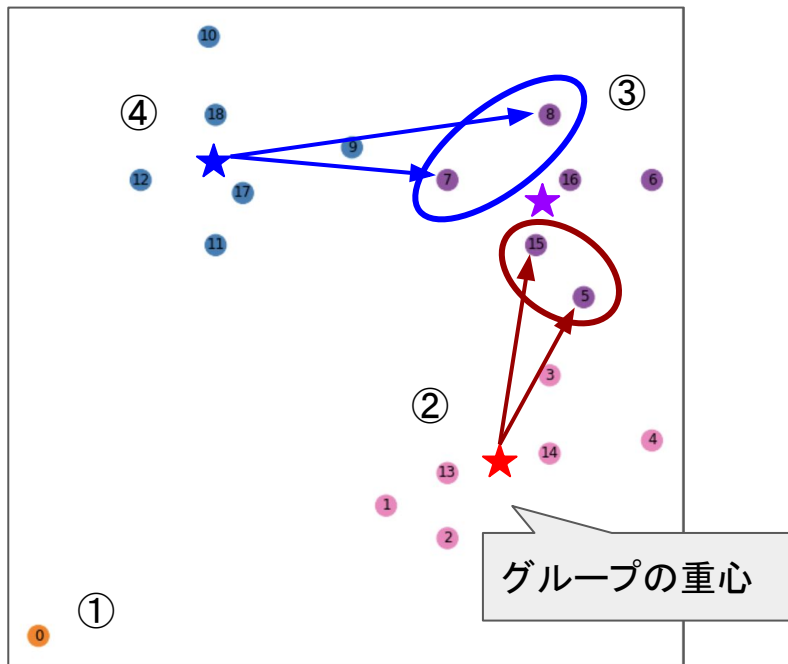
 d_{ij} :グループの重心 i, j 間の距離 $x_{i,t} = 1$:グループ i に t 番目に配達する0:グループ i に t 番目に配達しない G :グループ数 $i = 0$ は 営業所 $x_{0,0} = x_{0,G+1} = 1$ の固定値必要ビット数: G^2 

グループの重心

3

2の結果からグループ内の始点と終点の候補を決める.

アニーリングマシンを
使わない



【始点】

前のグループの重心から近い順に2箇所ずつ割り当て

【終点】

次のグループの重心から近い順に2箇所ずつ割り当て

4

始点と終点を変えてグループ内のTSPを解く.

$$H = \sum_{i,j=1}^N \sum_{t=1}^N d_{ij} x_{i,t} x_{j,t+1} + H_{p1} + H_{p2}$$

各時刻で一箇所にいる

各配達先に一度だけ訪問

d_{ij} : 配達先 i, j 間の距離

$x_{i,t} = 1$: 配達先 i に t 番目に配達する

0: 配達先 i に t 番目に配達しない

N : 始点と終点を除いたグループ内の配達先数

$i = 0$: 始点

$i = N + 1$: 終点

$x_{0,0} = x_{N+1,N+1} = 1$ の固定値

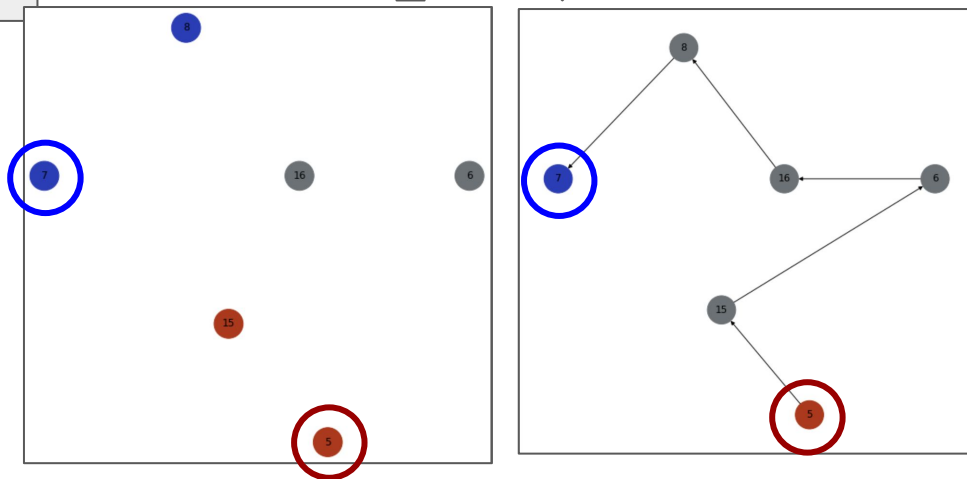
必要ビット数: N^2

始点と終点の組合せに対して

制約を満たしたものの移動距離を C_p

×4

ルート, 移動時間



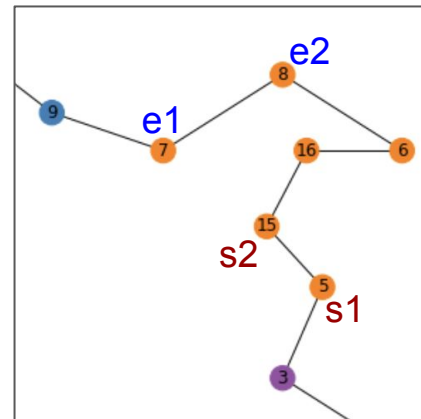
5

どの始点と終点を使うのかを決める。

$$H = \underbrace{\sum_{g=1}^G \sum_{p \in Q_g} \sum_{q \in Q_{g+1}} d_{pq} x_p x_q}_{\text{グループ間の移動距離}} + \underbrace{\sum_{g=1}^G \sum_{p \in Q_g} C_p x_p}_{\text{グループ内の移動距離}} + H_{p1}$$

グループ間の移動距離

グループ内の移動距離

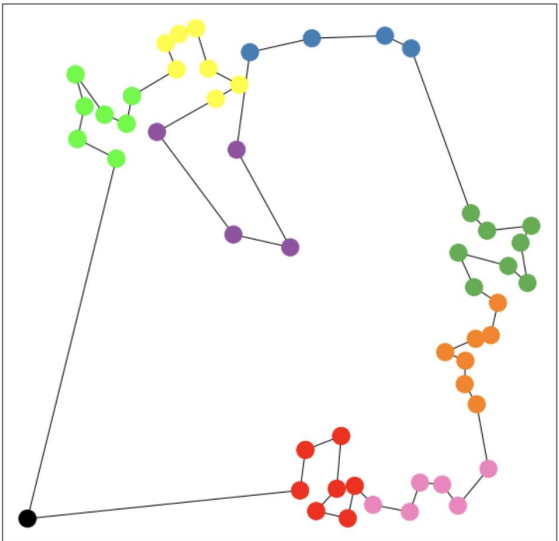
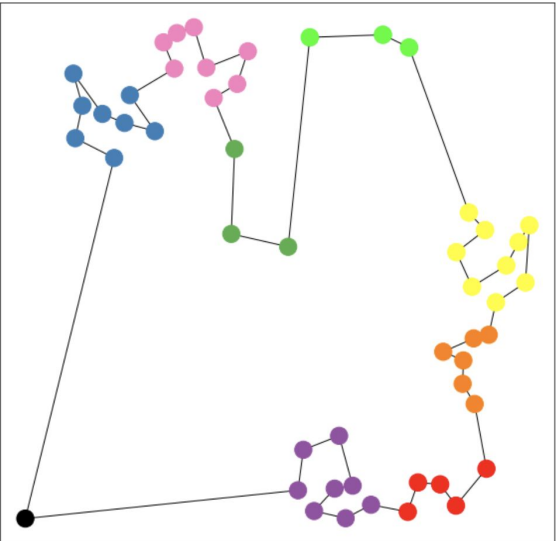


各グループ内で組合せを
1つだけ選ぶ

d_{pq} : 組合せ p, q 間の距離
 $x_p = 1$: 組合せ p を選ぶ
 0: 組合せ p を選ばない
 Q_g : グループ g 内の組合せの集合
 C_p : 組合せ p の時のグループ内の移動距離
 G : グループ数
 P : グループ内の組合せ数
 必要ビット数: GP

| | s1e1 (p=1) | s1e2 (p=2) | s2e1 (p=3) | s2e2 (p=4) |
|----|---------------|---------------|---------------|---------------|
| g1 | 0 | 1 | 0 | 0 |
| g2 | 0 | 0 | 0 | 1 |
| g3 | | | | |
| g4 | | | | |

ルート最適化の実行結果

| | 手法1 | 手法2(本PJで考案) |
|------|---|--|
| 移動距離 | 4.94 | 4.87 |
| ルート |  |  |

【データ】

- ・(0,1)上に50箇所
- ・グループ数8で
クラスタリング
- ・DAで各50回ずつ実行

【結果】

- ・移動距離: 手法1 > 手法2

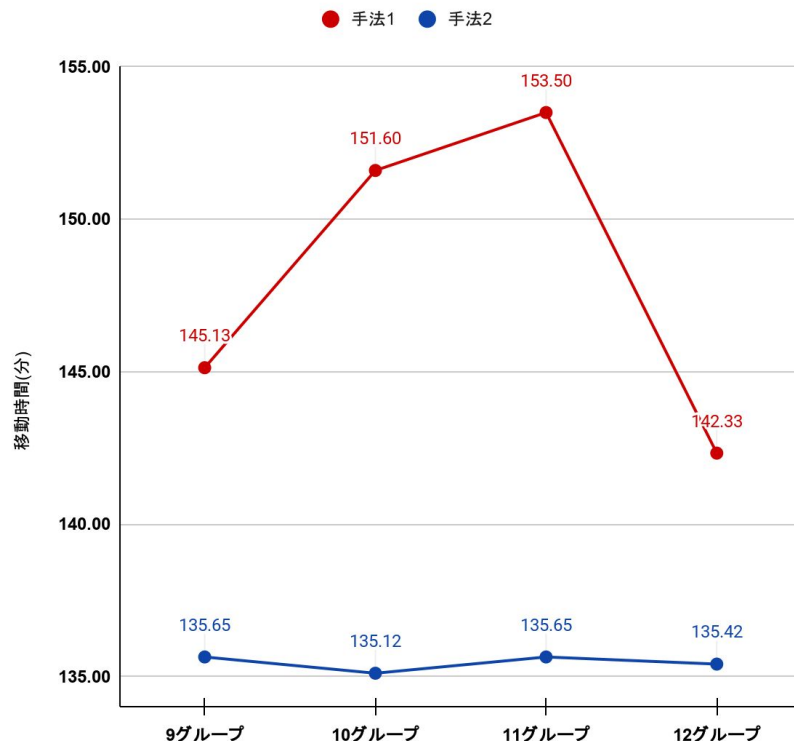
実データを用いたルート最適化の実行結果

【使用したデータ】

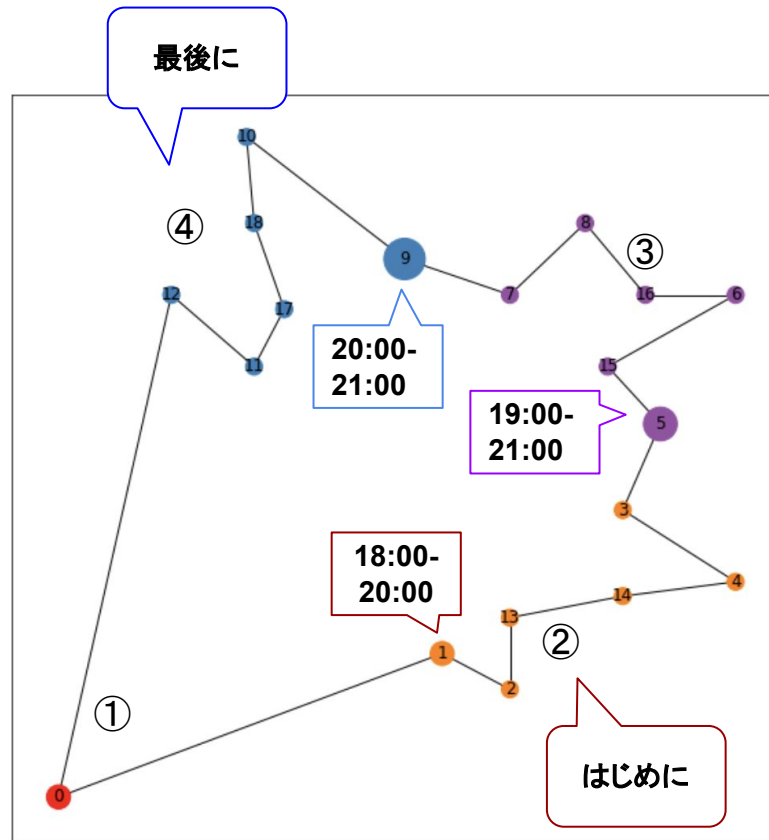
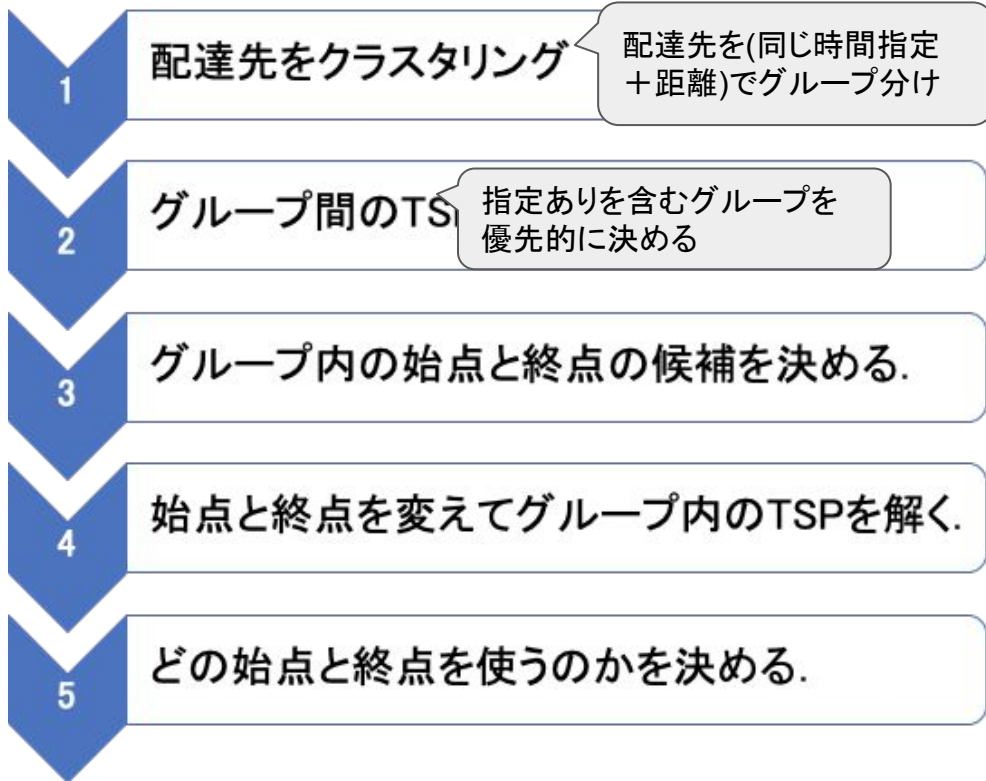
- ・配達先数: 61箇所
- ・営業所: 宝塚郵便局
- ・配達地域: 宝塚
- ・配達時間枠: 4時間程度
- ・2点間の移動時間: 交通情報を考慮した移動時間
- ・DAで実行

【結果】

- ・移動時間: 手法1 > 手法2(本PJで考案)
- ・手法2の方がグループ数によらない



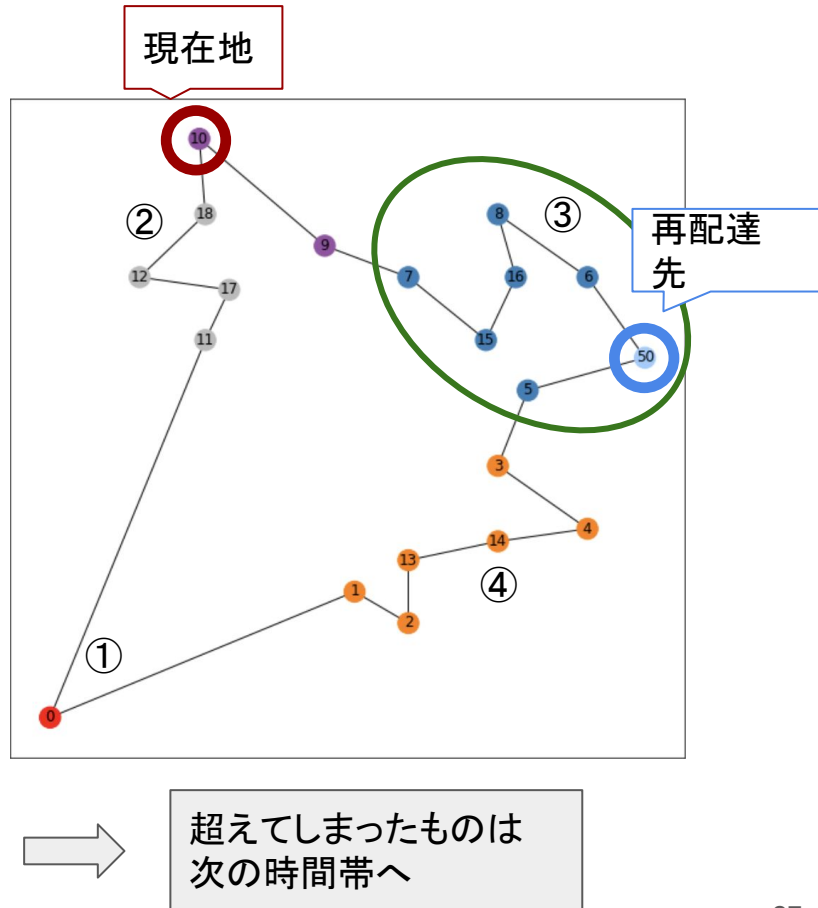
時間指定への対応



再配達への対応

- 1 再配達先を現在いるグループ以降に割り当て
- 2 割り当てられたグループの始点と終点を更新
- 3 割り当てられたグループ内のTSPのみ解く
- 4 割り当てる前と比較し、配達可能か判断

配達時間枠を超えないか



1. 概要
2. 車両への荷物の割り当て
3. ルート最適化
4. アプリケーション
5. まとめと今後の展望

アプリケーションのデモ

- 仮想データでの実行
- Google Maps APIを用いて移動時間を算出



① 午前中の荷物の割り当て



② 荷物を各車両に分ける



③ ルート最適化

1. 概要
2. 車両への荷物の割り当て
3. ルート最適化
4. アプリケーション
5. まとめと今後の展望

まとめ

- **開発内容**

- 荷物の割り当て→車両ごとのルート最適化で、1営業所全体の最適化
- 必要に応じて、新たなクラスタリング/ルート最適化アルゴリズムの開発

- **利用者や社会にとっての価値**

- 営業所全体を最適化することで忙しさを分散させることが可能
- ドライバーや営業所の負担軽減

- **中長期における達成目標**

- 規模を変えることで、幼稚園バスなどにも対応可能
- Uターンなどを避けるために、進行方向を考慮したルート最適化の開発

今後の展望

- 1. より多くの実データを用いることで、アルゴリズムの精度向上**
 - 実問題を解くことで、アルゴリズムを改良し、よりドライバーが納得できる配送計画の最適化を目指す。
- 2. 規模が小さい配送業者などへのアプローチ**
 - SAでも実行可能であるため、規模が小さい幼稚園バスや老人ホームの送迎車などへのアプローチを試みる。

発表実績

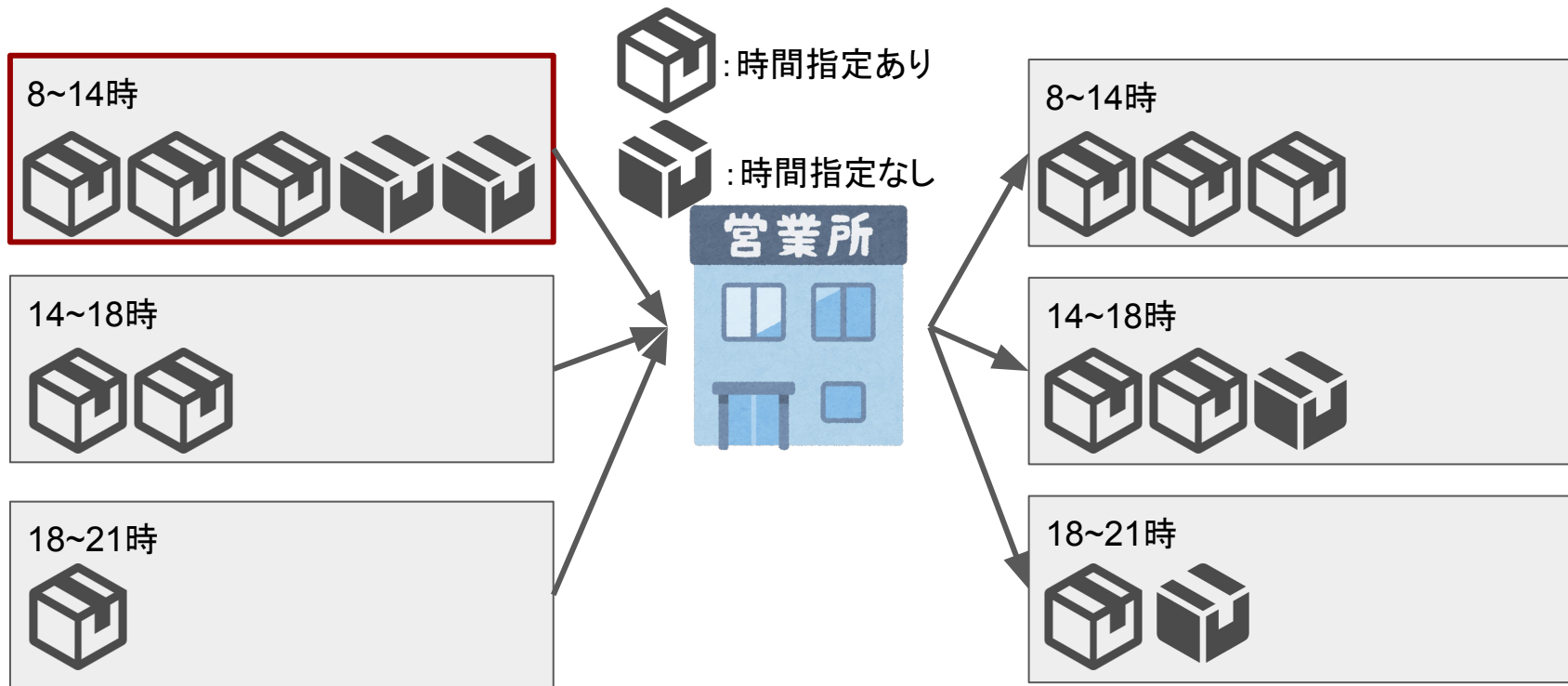
- 日本物理学会(2020.9)
- 情報論的学習理論ワークショップ(2020.11)
- OR学会(2021.3)

補足資料

① 午前中に届いた時間指定なしの荷物を3つの配達時間帯に分ける

| | 時間指定 | 最適化前の荷物の個数 | 最適化後の荷物の個数 | 当日の朝までに把握している個数 |
|--------|------|------------|------------|-----------------|
| 8~14時 | あり | 140 | 140 | 140 |
| | なし | 280 | 140 | |
| 14~18時 | あり | 70 | 70 | 20 |
| | なし | 70 | 140 | |
| 18~21時 | あり | 70 | 70 | 20 |
| | なし | 70 | 140 | |

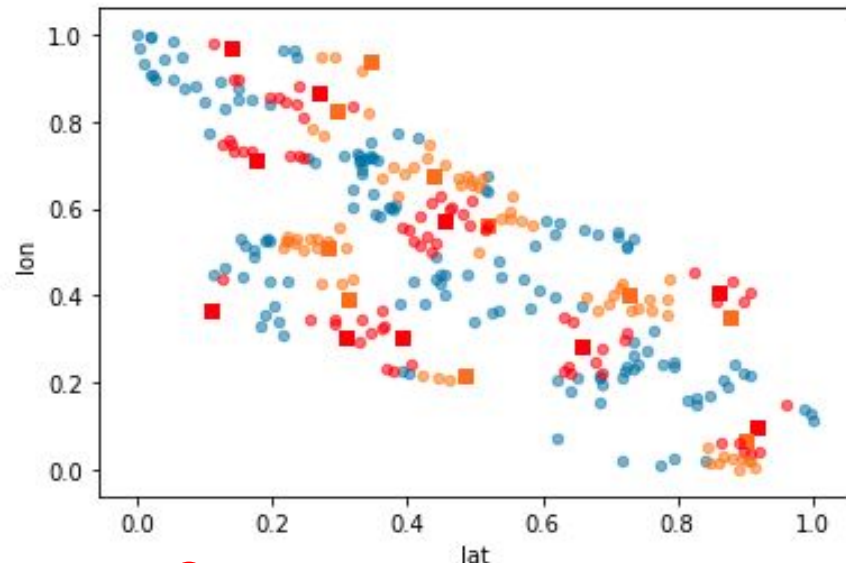
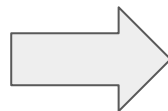
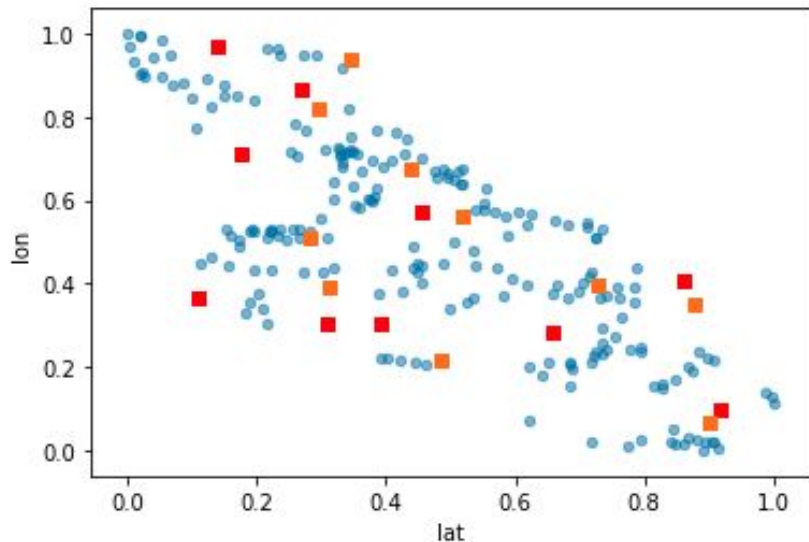
① 午前中に届いた時間指定なしの荷物を3つの配達時間帯に分ける



午前中に届いた時間指定なしの荷物の中で
午後に時間指定されている荷物近くに配達されるものを午後に回す

① 午前中に届いた時間指定なしの荷物を3つの配達時間帯に分ける 実行結果

8-14時の時間指定なし280個→140個に



- :8-14時に配達予定の時間指定なし
- :14-18時に配達予定の時間指定あり
- :18-21時に配達予定の時間指定あり

- :14-18時に配達予定の時間指定なし
- :18-21時に配達予定の時間指定なし

クラスタリング手法2(分数コスト)の式変形

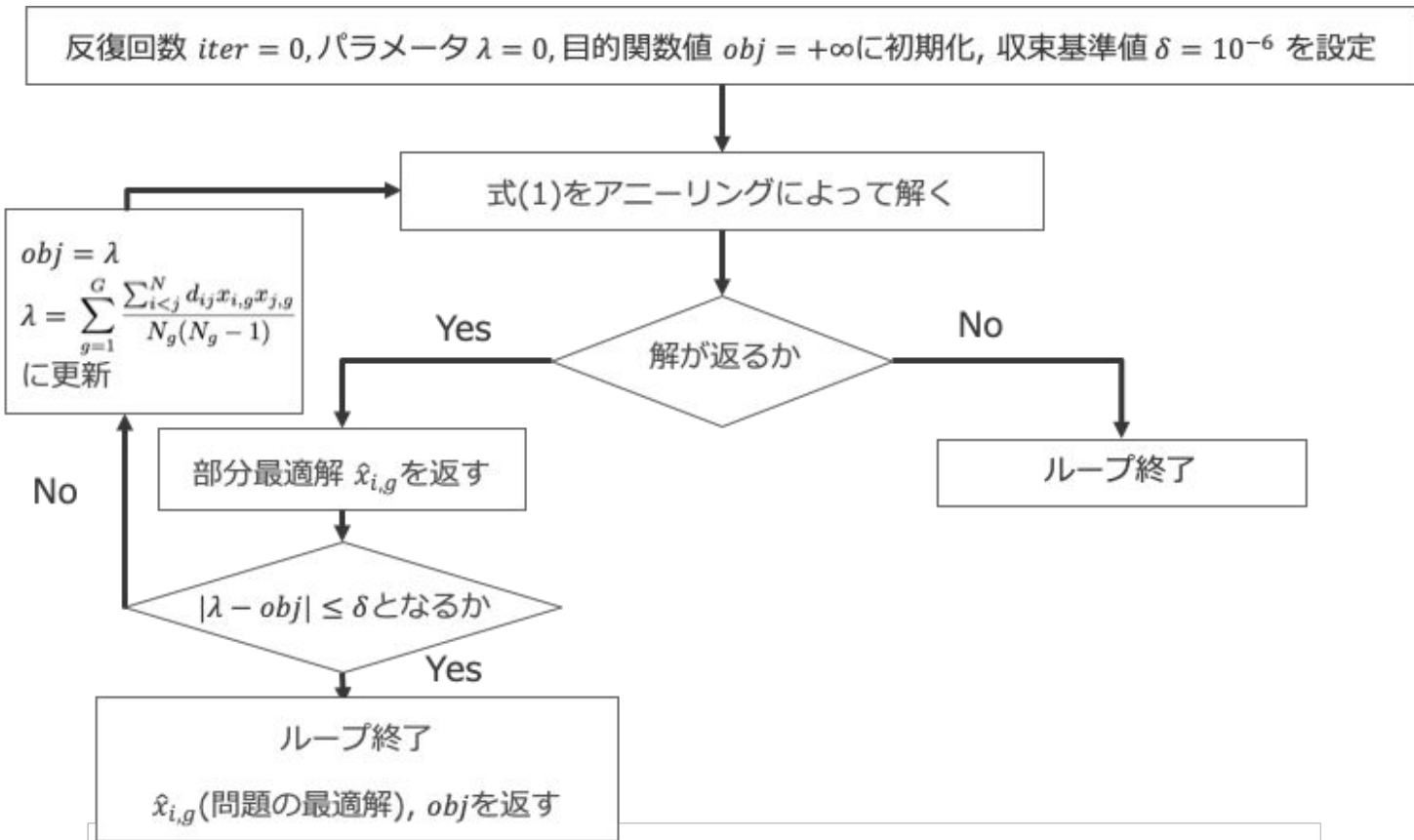
$$H = \sum_{g=1}^G \frac{\sum_{i<j}^N d_{ij} x_{i,g} x_{j,g}}{N_g(N_g - 1)} + \alpha \sum_{i=1}^N \left(\sum_{g=1}^G x_{i,g} - 1 \right)^2$$



$$H = \sum_{g=1}^G \left(\sum_{i<j}^N d_{ij} x_{i,g} x_{j,g} - \lambda N_g(N_g - 1) \right) + \alpha \sum_{i=1}^N \left(\sum_{g=1}^G x_{i,g} - 1 \right)^2 \quad \dots \text{式(1)}$$

参考文献： A. Ajagekar, T. Humble, F. You, "Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems", Computers and Chemical Engineering, **132**, 106630 (2020).

クラスタリング手法2(分数コスト)のアルゴリズム



配達時間帯

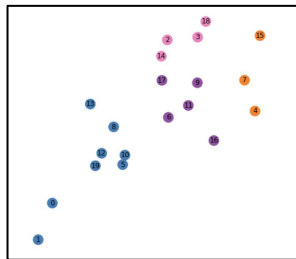
| ルート算出のタイミング (出発前) | 時間指定の時間帯 |
|----------------------|----------|
| 8時 | 午前中 |
| | 12時～14時 |
| 14時 | 14時～16時 |
| | 16時～18時 |
| 18時 | 18時～20時 |
| | 19時～21時 |
| | 20時～21時 |

【午前中のルート】
(午前中+12時～14時)の
荷物を入力

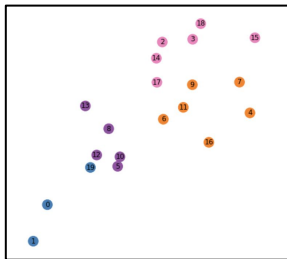
クラスタリング手法の比較

4グループに分けた場合の例

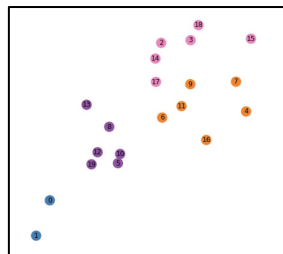
k-means



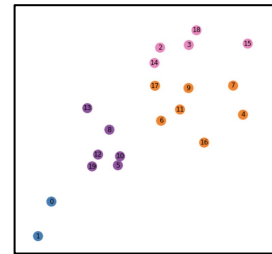
手法1



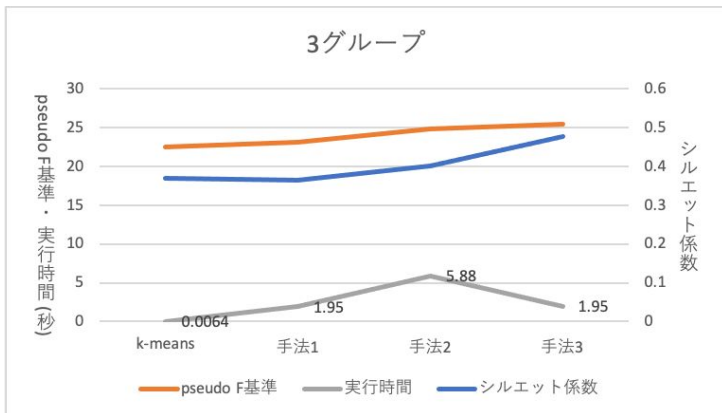
手法2



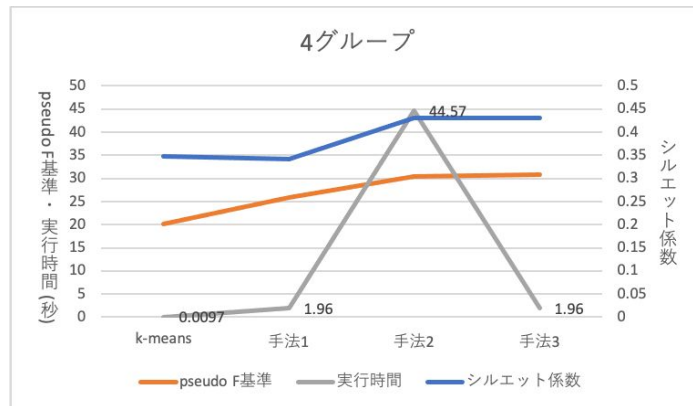
手法3(コミュニティ検出)



3グループ



4グループ



実行結果を踏まえたクラスタリング手法の比較

| | 手法1 | 手法2 | コミュニティ検出 |
|-----------------------------|---------------------|-------------------------------|---------------------------------|
| クラスタ内の 頂点数 | 均一になりやすい 一様分布に有効 | 異なりやすい 離れた頂点がある 場合に有効 | 異なりやすい 離れた頂点がある 場合に有効 |
| アルゴリズムの 複雑さ | 単純 | 複雑 | 複雑 |
| アニーリング回数 (探索回数:100,000回) | 1回 | 複数回 (実行時間が 手法1,手法3の整数倍) | 1回 |
| パラメータチューニング のしやすさ | ◎ パラメータ数1 | ○ パラメータ数1 | △ パラメータ数3~4 |
| 枝刈り可能か | × | × | ○ 直線距離で枝刈りし、移動 時間の計算を短縮可能 |