

卒業論文(中間発表)

コンテンツ

- タイトル
- 問題背景
- 分析の下準備
- 分析

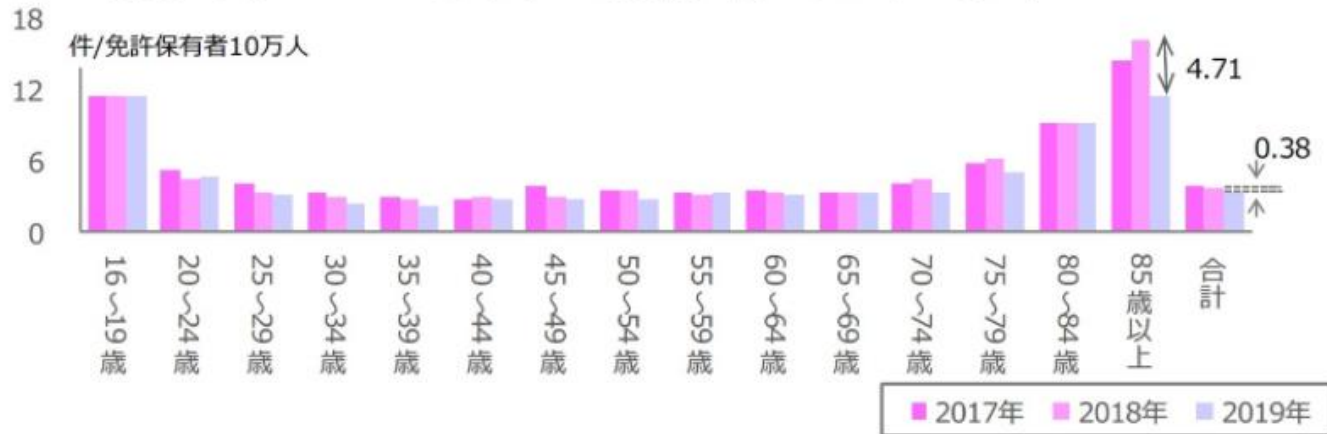
タイトル

地域人口に考慮した地方(日の出町北西部)のバス停最適配置

問題背景

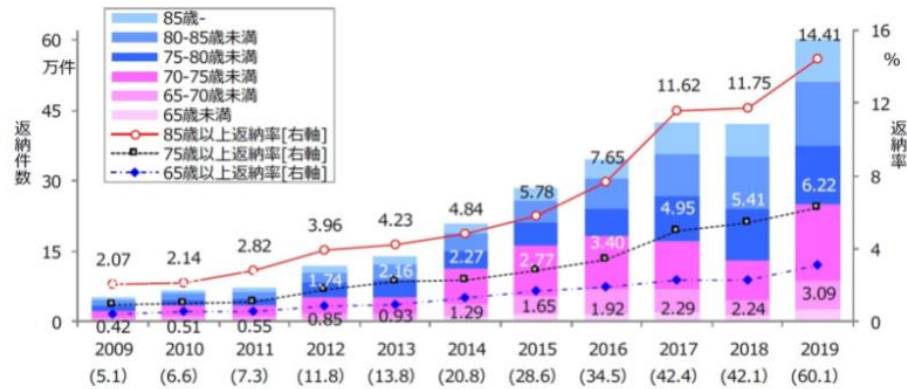
近年、高齢化に伴う事故により免許返納率が上昇

図表2 年齢層別免許保有者10万人当たり死亡事故件数（原付以上、第1当事者）



(資料) 警察庁交通局「令和元年における交通死亡事故の特徴等について」

図表4 免許返納数、返納率の推移



(注1) 各年の () 内は返納総数 (万件)

(注2) 2012年より返納時に身分証明として利用できる運転経歴証明書を希望者に発行、2017年に認知機能検査厳格化

(資料) 警察庁「運転免許統計」(各年)

問題背景

目次 [非表示]

[1 免許返納後の移動手段：バス・電車などの公共交通機関](#)

[2 免許返納後の移動手段：タクシー](#)

[3 免許返納後の移動手段に悩まない生活を](#)

[4 まとめ](#)

免許返納後は公共交通機関を利用

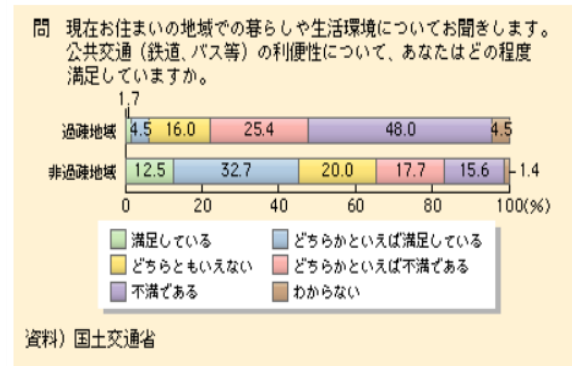
問題背景

しかし、バス停の配置満足度は低い

(過疎地域等における移動ニーズ)

意識調査では、過疎地域において、「公共交通（鉄道、バス等）の利便性」の満足度は6.2%と非常に低く、非過疎地域で45.2%となっているのに比べ大きな開きがある。また、「公共交通が整備されていること」に対する不満度は70.7%と満足度9.7%に比べ非常に高くなっている。

図表I-1-3-17 地域別公共交通（鉄道、バス等）の利便性に関する満足度



→ 最適な配置を求める

分析地区

過疎化が進んでいる「奥多摩町」を選択



【全部過疎市町村】

- 青ヶ島村 (あおがしまむら)
- 大島町 (おおしままち)
- 奥多摩町 (おくたままち)
- 新島村 (にいじまむら)
- 八丈町 (はちじょうまち)
- 檜原村 (ひのほらむら)
- 三宅村 (みやけむら)

過疎化マップ(東京都)

分析地区

◆公共交通関連施策

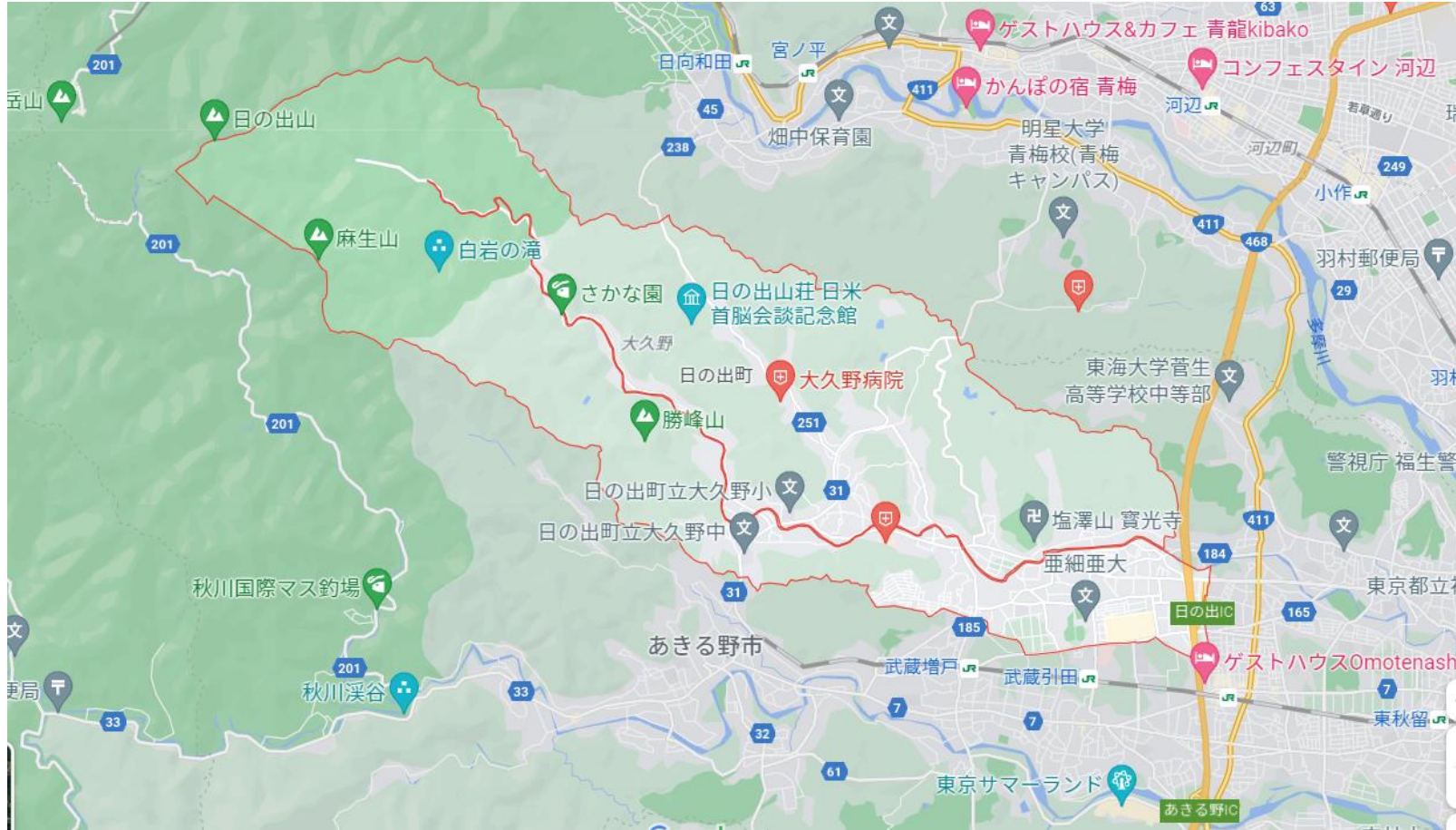
公共交通に関しては、特に路線バスに対する満足度評価は低く、優先施策の第2位となっています。また、基本目標2においては、「定住と交流を生み出す生活基盤づくり」を示しており、だれもが利用しやすい公共交通の充実を図ることを掲げています。具体的には下記の項目を示しています。

<公共交通の利便性向上>

- ・JR五日市線・青梅線の利便性向上を近隣市町と協力して関係機関に働きかけるとともに、町民の重要な足としての路線バスについても路線維持やJR駅との結節・利便性向上等を働きかけていきます。
- ・町内を循環するコミュニティバス（町内循環バス、外出支援バス）については、今後とも安全運行の徹底とサービスの向上に努めます。

奥多摩群のバスの満足度は低い(特に日の出町という場所)

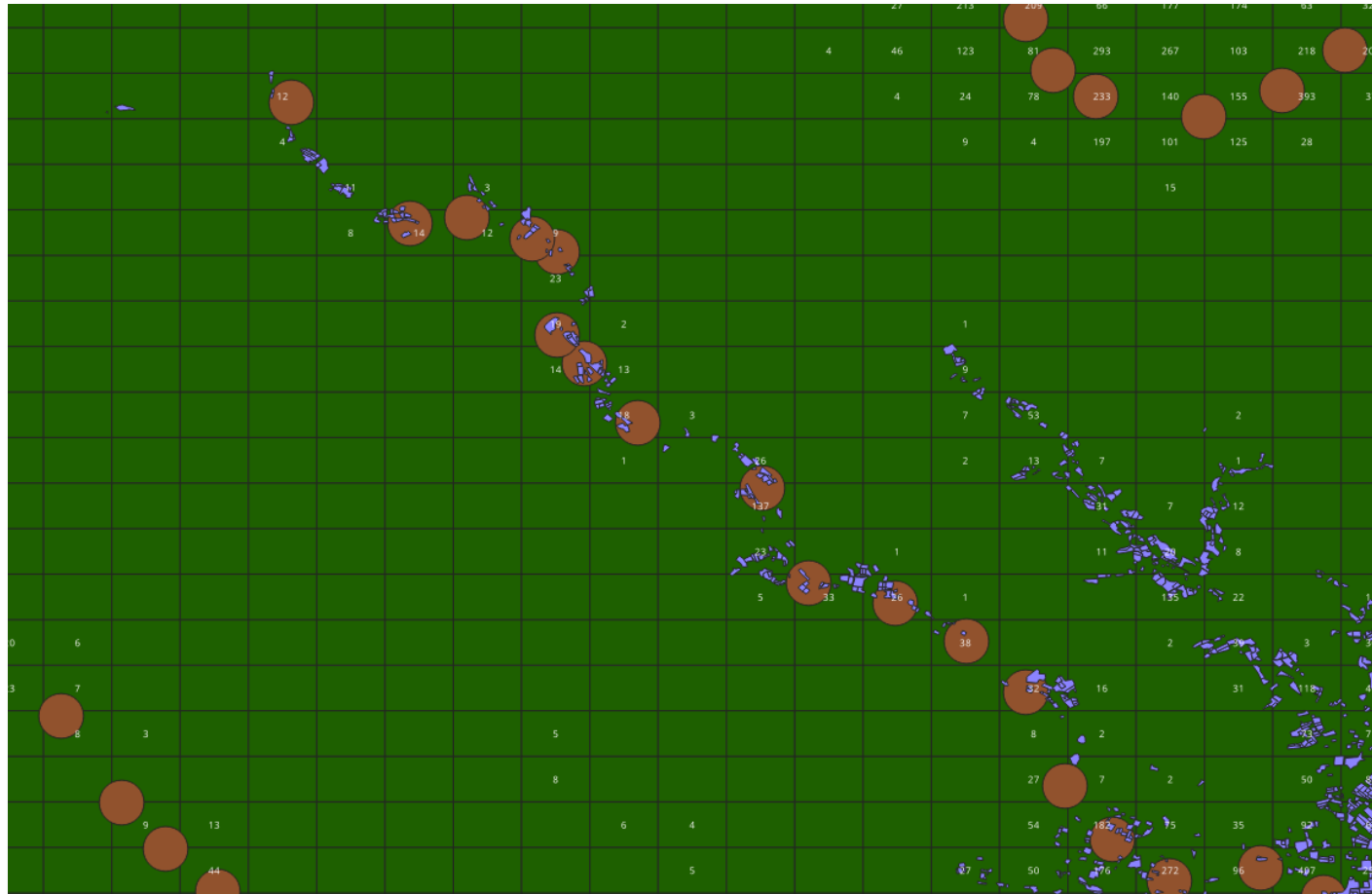
奥多摩(西多摩郡)に位置する日の出町



分析の下準備

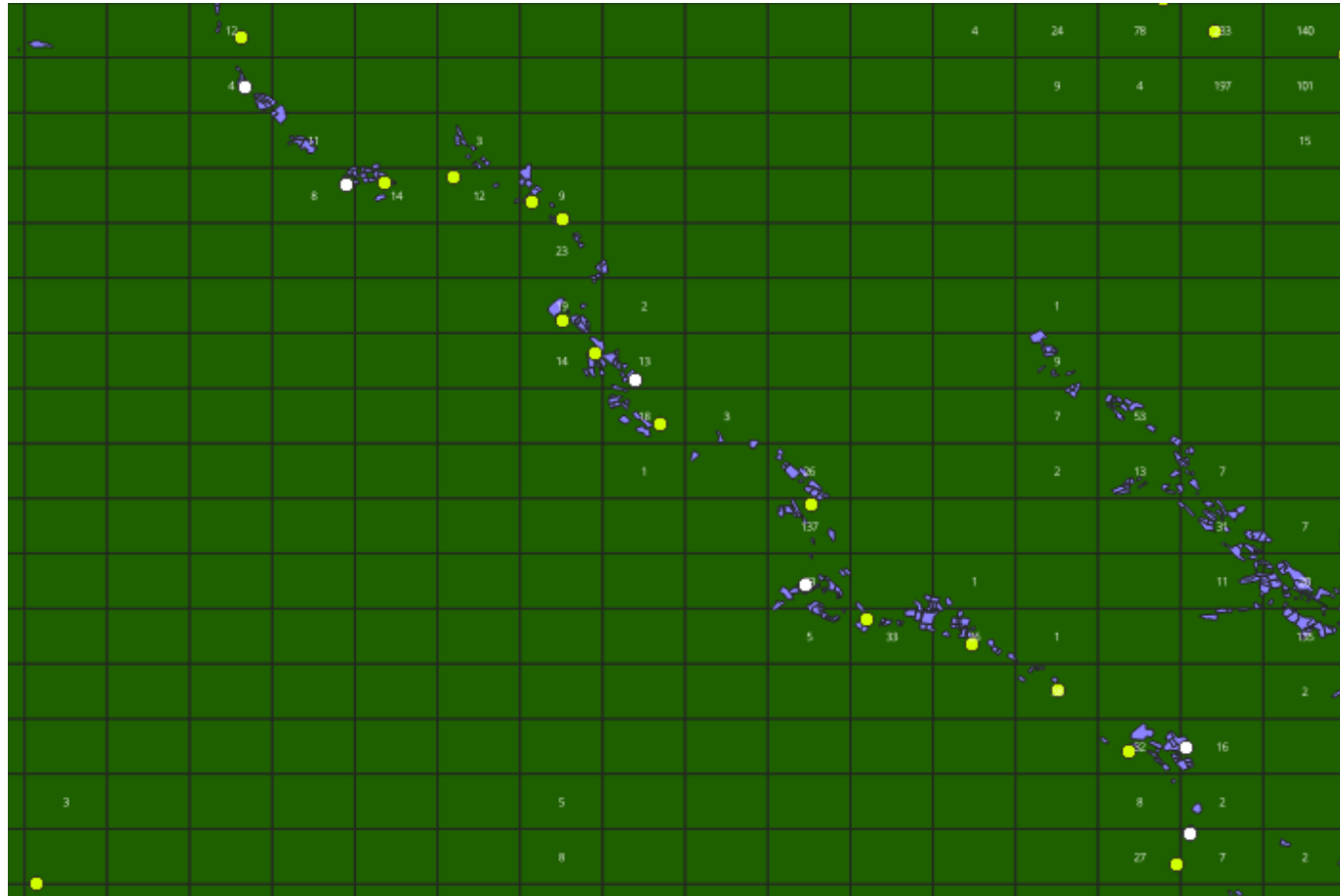
- ①人口メッシュからバス停位置を可視化、距離を測定
- ②定式化
- ③p-median問題を解く

①QGISを用いて現在のバス停位置を可視化



日の出町北西部のバス停

既存のバス停と比較



○が建設候補地
●が現在のバス停

②定式化

定義:

バス停の候補地集合 $F_i = \{f_0, f_1, f_2, \dots\}$

利用者の集合 $d_i = \{d_0, d_1, d_2, \dots\}$

利用者 d_i がバス停 f_j を使用 = (Binary) X_{ij}

バス停間の距離 $C_{ij} = \{150, 100, \dots\}$

②定式化

目的関数 (min):

$$\sum_i \sum_j W_i * C_{ij} * X_{ij}$$

制約条件:

$$\textcircled{1} \sum_i F_i == 9 \quad \# \text{配置は現在と同じで9に設定}$$

$$\textcircled{2} \sum_j X_{ij} == 1, \quad \# \text{利用できる施設は一つのみ}$$

$$\textcircled{3} X_{ij} \leq F_j, \quad \# \text{利用者がいないならば使えない(binary)}$$

③ p-median問題を解く

PuLPを使ったコードを実行例

```
!pip install pulp
import pulp

prob = pulp.LpProblem('p-median')

# コスト (距離) 表の列から日の出町の各距離データをタプル型で括る
# 整理 f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 f10 f11 f12 f13 f14
cs = [[0, 400, 1100, 1400, 1570, 1824, 2154, 2654, 3054, 3844, 4144, 4544, 4714, 4774, 5014], # d0
      [400, 0, 700, 1000, 1170, 1424, 1754, 2254, 2654, 3444, 3744, 4144, 4314, 4374, 4614], # d1
      [1100, 700, 0, 1700, 1870, 2124, 2454, 2954, 3354, 4144, 4444, 4844, 5014, 5074, 5314], # d2
      [1400, 1000, 1700, 0, 170, 424, 754, 1254, 1654, 2444, 2744, 3144, 3314, 3374, 3614], # d3
      [1570, 1170, 1870, 170, 0, 254, 584, 1084, 1484, 2274, 2574, 2974, 3144, 3204, 3444], # d4
      [1824, 1424, 2124, 424, 254, 0, 330, 830, 1230, 2020, 2320, 2720, 2890, 2950, 3190], # d5
      [6380, 4556, 3132, 1008, 584, 330, 0, 500, 900, 1690, 1990, 2390, 2560, 2620, 2860], # d6
      [16490, 10110, 5554, 2422, 1414, 830, 500, 0, 400, 1190, 1490, 1890, 2060, 2120, 2360], # d7
      [37720, 21230, 11120, 5566, 3144, 1730, 900, 400, 0, 790, 1090, 1490, 1660, 1720, 1960], # d8
      [82600, 44880, 23650, 12530, 6964, 3820, 2090, 1190, 790, 0, 300, 700, 870, 930, 1170], # d9
      [178814, 96214, 51334, 27684, 15154, 8190, 4370, 2280, 1090, 300, 0, 444, 614, 674, 914], # d10
      [385874, 207060, 110846, 59512, 31828, 16674, 8484, 4114, 1834, 744, 444, 0, 170, 230, 470], # d11
      [827584, 441710, 234650, 123804, 64292, 32464, 15790, 7306, 3192, 1358, 614, 170, 0, 60, 300], # d12
      [1753234, 925650, 483940, 249290, 125486, 61194, 28730, 12940, 5634, 2442, 1084, 470, 300, 0, 350], # d13
      [3650394, 1897160, 971510, 487570, 238280, 112794, 51600, 22870, 9930, 4296, 1854, 770, 300, 350, 0]] # d14

# 人数(今回は、f0~f14まで)
ws = [12, 4, 8, 14, 9, 23, 14, 13, 13, 18, 137, 23, 38, 16, 32]

# 変数
# fs[i] : 施設 fi を配置するか否か
# xs[i][j] : di が fi を利用するか否か
#
fs = [pulp.LpVariable('f{}'.format(i), cat='Binary') for i in range(14)]
xs = []
for i in range(14):
    xs1 = [pulp.LpVariable('x{}_{}'.format(i, j), cat='Binary') for j in range(14)]
    xs.append(xs1)

# 目的関数
prob += pulp.lpSum([ws[i] * pulp.lpDot(cs[i], xs[i]) for i in range(14)])

# 制約条件
prob += pulp.lpSum(fs) == 9
for i in range(14):
    prob += pulp.lpSum(xs[i]) == 1
for i in range(14):
    for j in range(14):
        prob += xs[i][j] <= fs[i]

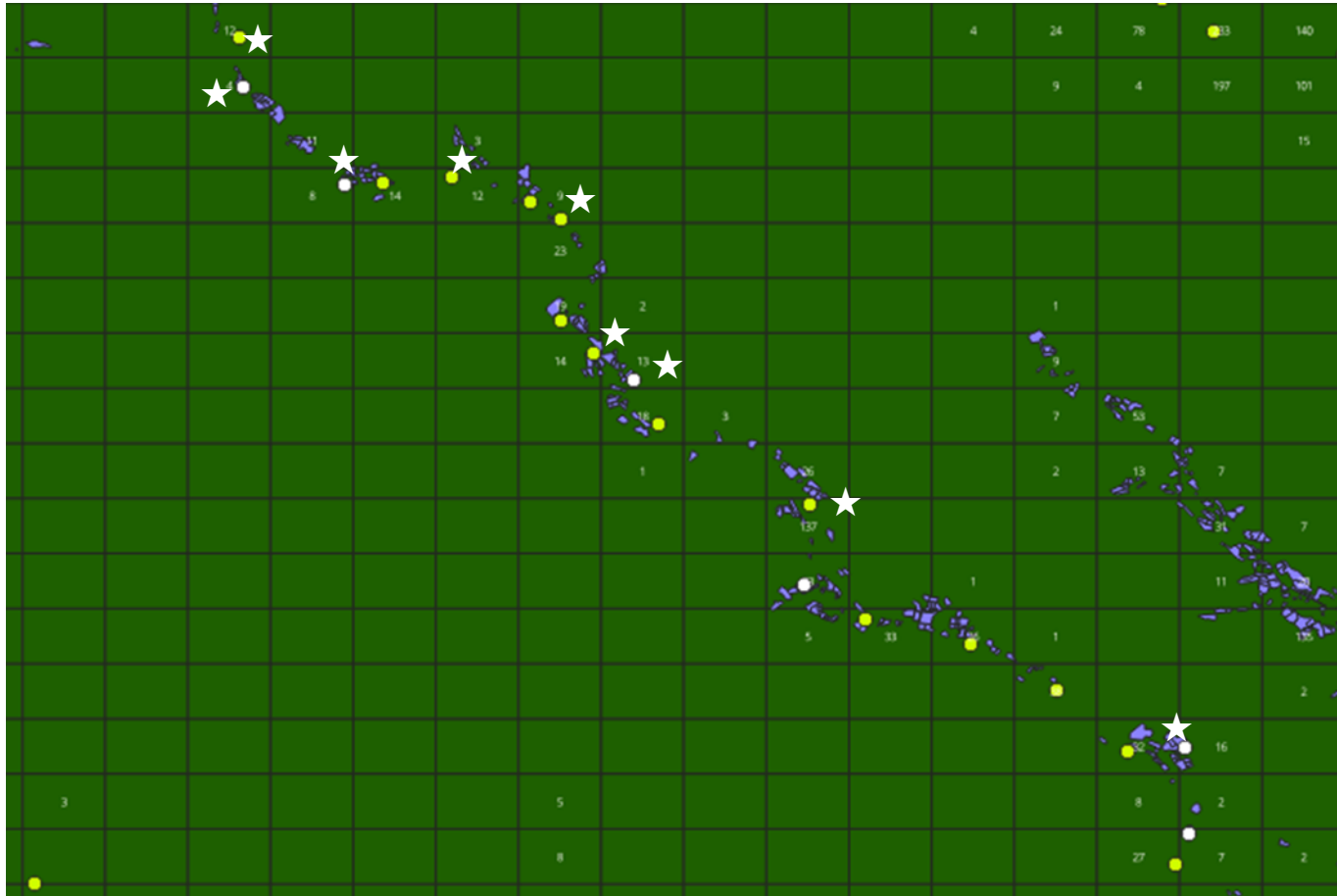
#
status = prob.solve()
print("Status", pulp.LpStatus[status])
print([f.value() for f in fs])
for xs1 in xs:
    print([x.value() for x in xs1])
print(prob.objective.value())
```

実行結果

```
↳ Requirement already satisfied: pulp in /usr/local/lib/python3.7/dist-packages (2.5.0)
Status Optimal
[1.0, 0.0, 1.0, 1.0, 0.0, 1.0, 0.0, 1.0, 0.0, 1.0, 1.0, 1.0, 0.0, 1.0]
[1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0]
[0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0]
15230.0
```

適正な施設配置はf(0,2,3,5,7,9,10,11,13)
という結果になった。

P-medianでの再配置結果(図)



★が適正配置