

# 均衡配分

---

交通・都市・国土学研究室  
増田慧樹・鈴木大樹

# 目次

---

1. 四段階推定法
2. 配分モデル
3. 均衡配分の考え方
4. 分析と表現

## 目標

均衡配分の概要，分析の大まかな流れを理解する

# 1. 四段階推定法

## 1.1. 四段階推定法の概要

- 将来の交通量を予測するために開発された手法である。
- 1950年にアメリカで開発された。
- 日本では1967年に広島都市圏で最初に使用された。
- 総合的な交通計画を立案する際の予測手法として定着している。
- 将来の都市活動と将来交通網をインプットし、アウトプットとして各交通施設における将来の交通量の推定値が得られる。

## 1.2. 四段階推定法のプロセス（次頁の図参照）

計画データ（将来の都市活動・将来交通網）をインプットする.

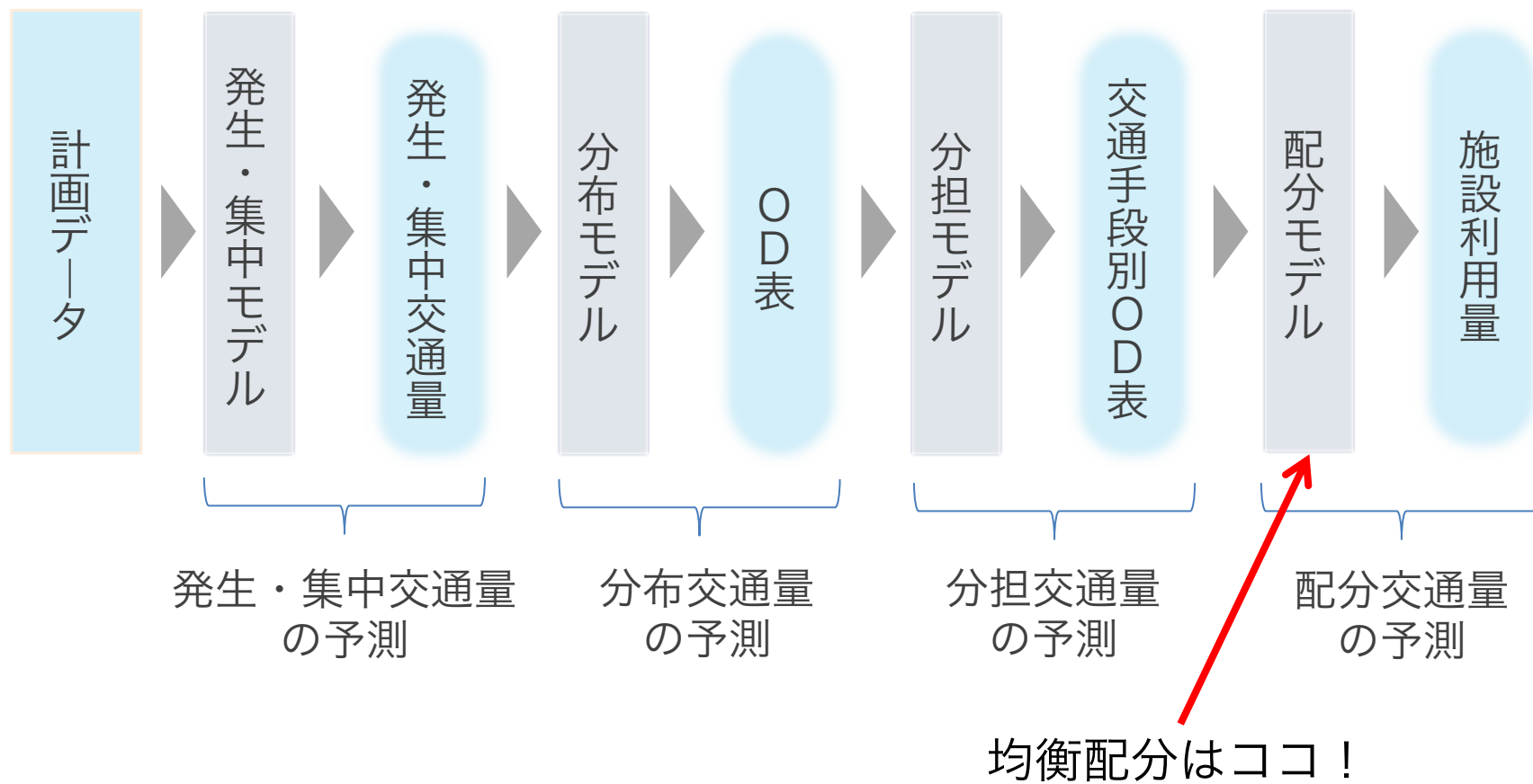


発生・集中交通量の予測→分布交通量の予測→分担交通量の予測→  
配分交通量の予測 の順に推定する. 各段階において, 発生・集中モ  
デル, 分布モデル, 分担モデル, 配分モデルが使用される.

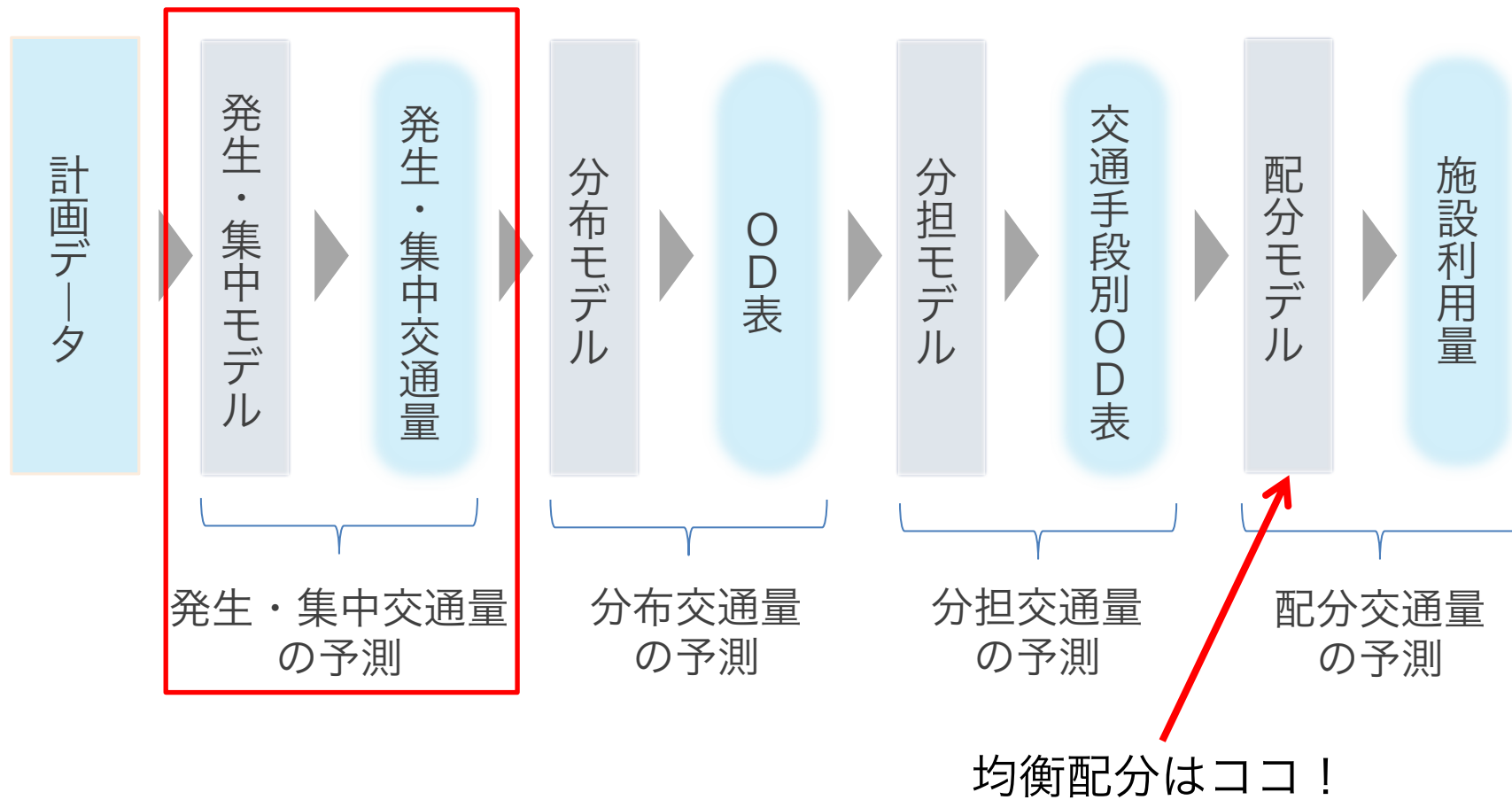


最終的なアウトプットとして, 各交通施設の利用量が推定される.

## 1.2. 四段階推定法のプロセス

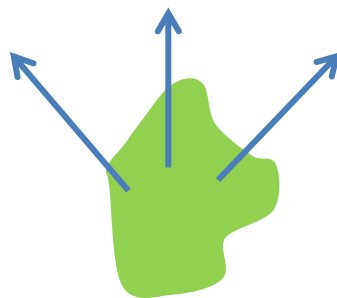


## 1.3. プロセスの詳細①ー発生・集中交通量の予測ー

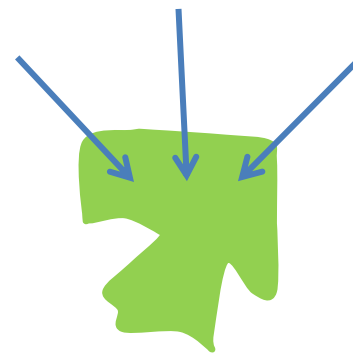


### 1.3. プロセスの詳細①ー発生・集中交通量の予測ー

- PT調査と同様，四段階推定法においても，対象圏域をいくつかのゾーンに分け，ゾーンからゾーンへの交通として捉える。
- ゾーン*i*から発生する交通量が発生交通量  $G_i$ ，ゾーン*i*に集中する交通量が集中交通量  $A_i$ である。このステップでは，**各ゾーンの将来における発生・集中交通量を予測する。**



発生交通量  $G_i$



集中交通量  $A_i$



### 1.3. プロセスの詳細①－発生・集中交通量の予測－

- はじめに対象都市圏全体の生成交通量 $T$ を予測し、それを基にゾーン別の発生・集中交通量 $G_i, A_i$ を予測する。
- 生成交通量 $T$ は、属性別の1人あたりあるいは世帯当たりのトリップ数を生成原単位とし、これに将来の対象地域内の属性別人口あるいは世帯数を乗ずることによって推定する。
- ゾーン別の発生・集中交通量 $G_i, A_i$ の推定には、発生・集中モデルが用いられる。具体的には、原単位法、グロス分類法、重回帰モデル法などがある（詳細が気になる人は調べてみてください）。

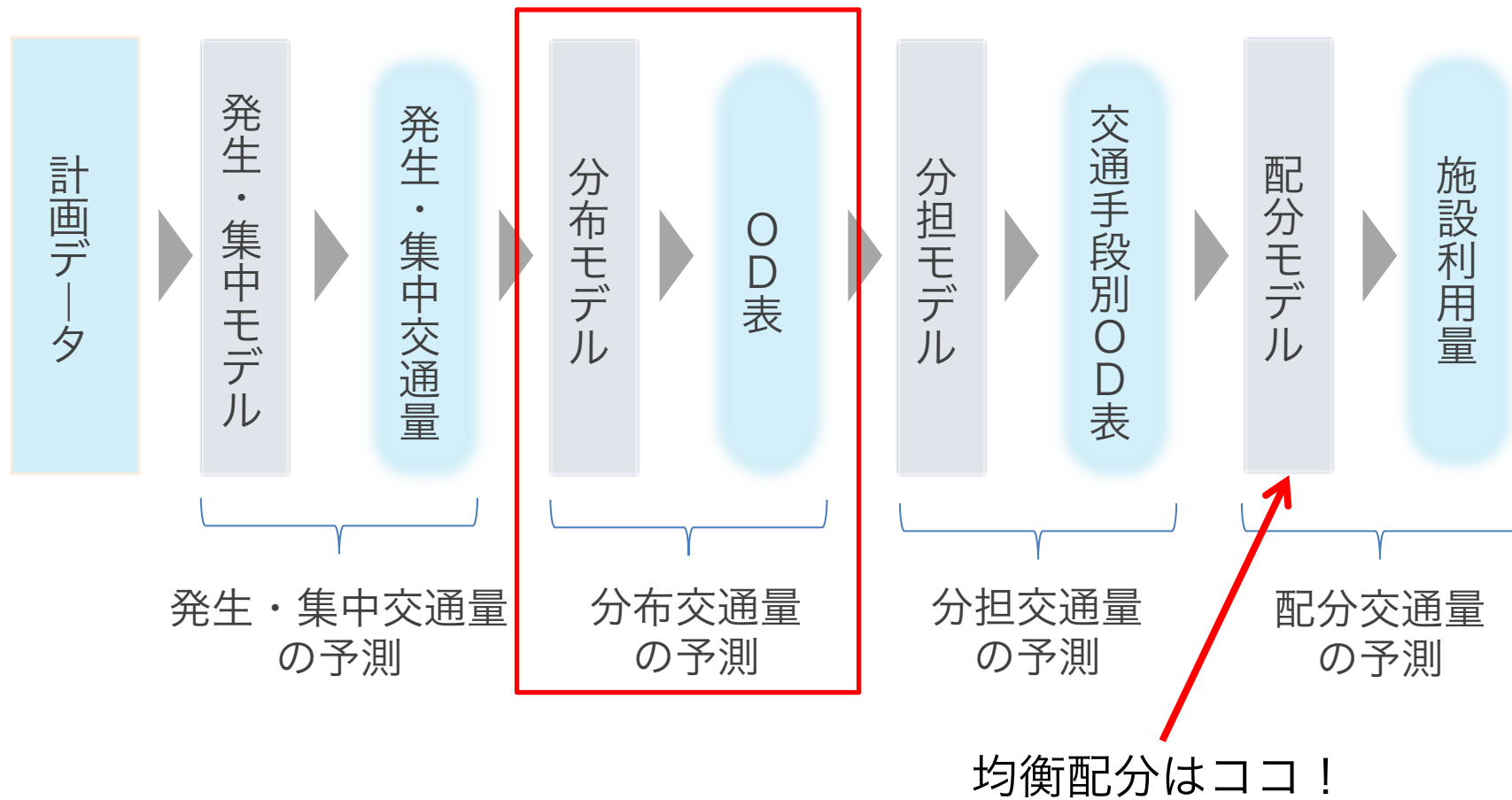
## 1.3. プロセスの詳細①－発生・集中交通量の予測－

- 発生・集中交通量の予測は，将来のOD表において，下図の赤線で囲まれた部分を推定するステップである。

| ゾーン | 1     | ..... | $j$      | ..... | $N$   | 発生量   |
|-----|-------|-------|----------|-------|-------|-------|
| 1   | ?     | ?     | ?        | ?     | ?     | $G_1$ |
| ⋮   | ?     | ?     | ?        | ?     | ?     | ⋮     |
| $i$ | ?     | ?     | $T_{ij}$ | ?     | ?     | $G_i$ |
| ⋮   | ?     | ?     | ?        | ?     | ?     | ⋮     |
| $N$ | ?     | ?     | ?        | ?     | ?     | $G_N$ |
| 集中量 | $A_1$ | ..... | $A_j$    | ..... | $A_N$ | $T$   |

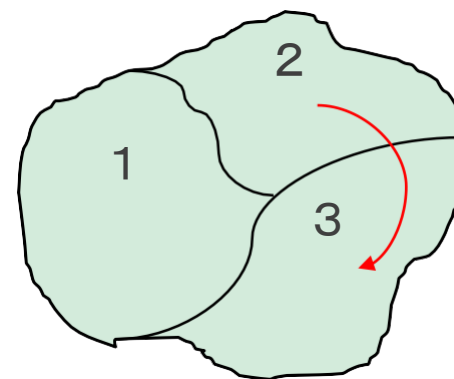
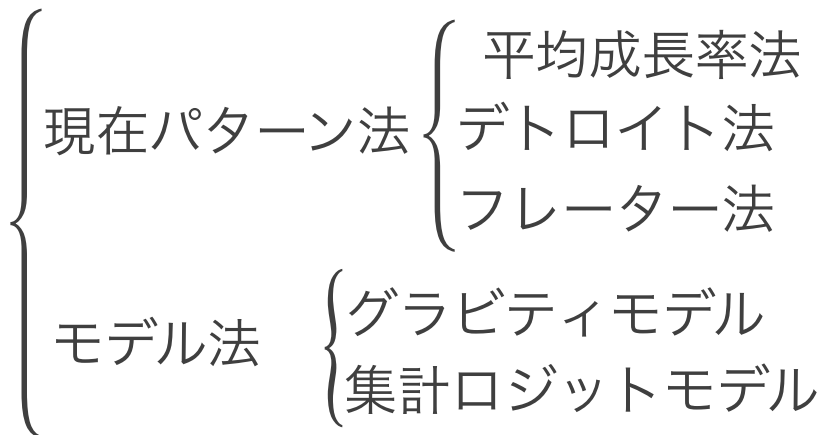
発生・集中モデル

## 1.4. プロセスの詳細②ー分布交通量の予測ー



## 1.4. プロセスの詳細②ー分布交通量の予測ー

- 分布交通量 $T_{ij}$ とは、**ゾーン*i*から発生し、ゾーン*j*に集中する交通量**のことである。
- 前のステップで推定した発生・集中交通量を基に予測する。
- 予測には分布モデルが用いられる。具体的には、以下のような種類がある（詳細が気になった人は各自調べてみてください）。

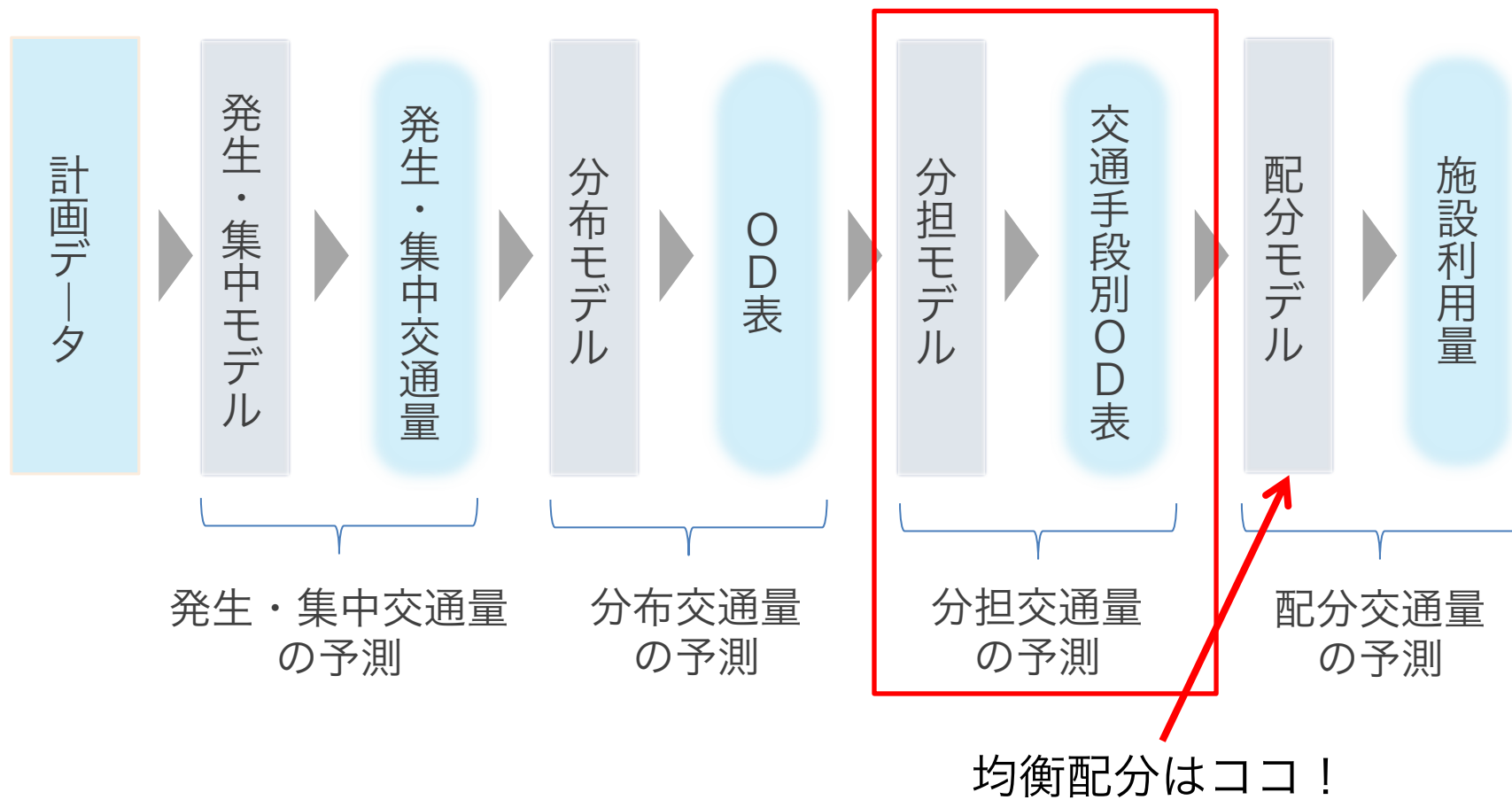


## 1.4. プロセスの詳細②ー分布交通量の予測ー

- 分布交通量の予測は、将来のOD表において、以下の赤線で囲まれた部分を推定するステップである。つまり、分布交通量の予測が終了する段階で、将来のOD表は完成する。

| ゾーン | 1     | ..... | $j$      | ..... | $N$   | 発生量   |
|-----|-------|-------|----------|-------|-------|-------|
| 1   | ?     | ?     | ?        | ?     | ?     | $G_1$ |
| ⋮   | ?     | ?     | ?        | ?     | ?     | ⋮     |
| $i$ | ?     | ?     | $T_{ij}$ | ?     | ?     | $G_i$ |
| ⋮   | ?     | ?     | ?        | ?     | ?     | ⋮     |
| $N$ | ?     | ?     | ?        | ?     | ?     | $G_N$ |
| 集中量 | $A_1$ | ..... | $A_j$    | ..... | $A_N$ | $T$   |

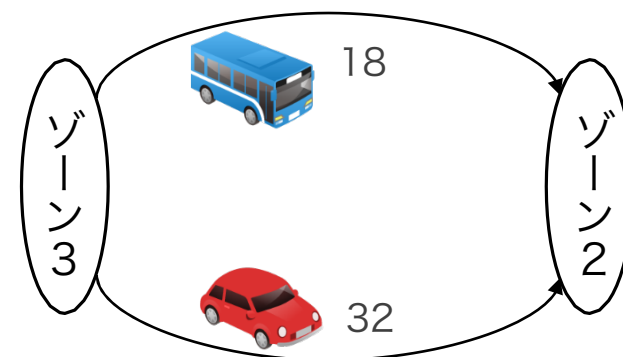
## 1.5. プロセスの詳細③ー分担交通量の予測ー



## 1.5. プロセスの詳細③ー分担交通量の予測ー

- 分担交通量 $T_{ijm}$ とは、ゾーン*i*からゾーン*j*までの、**交通手段*m*による交通量**のことである。つまり、各ODペアの交通がどの交通手段をどれだけ利用するかを予測するステップである。

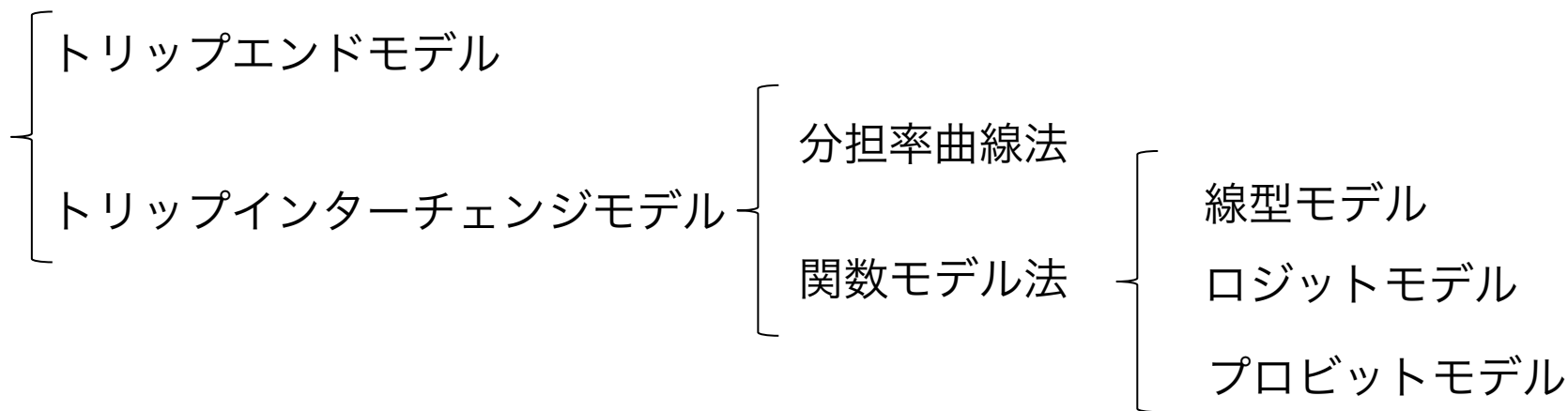
| 分布       |    |    |    |     | 分担        |    |
|----------|----|----|----|-----|-----------|----|
| ゾーン      | 1  | 2  | 3  | 発生  | $T_{ijm}$ |    |
| 1        | 11 | 12 | 20 | 43  | マストラ      | 18 |
| 2        | 32 | 30 | 3  | 65  | 車         | 32 |
| 3        | 8  | 50 | 55 | 113 |           |    |
| $A_j$ 集中 | 51 | 92 | 78 | 221 |           |    |



ゾーン3を出発してゾーン2に到着する50トリップのうち、車を利用しているのが32トリップ

## 1.5. プロセスの詳細③ー分担交通量の予測ー

- 前のステップで推定した分布交通量を基に予測が行われる。
- 予測には分担モデルが用いられる。具体的には、以下のようなものがある（詳細が気になった人は各自で調べてみてください）。





## 1.5. プロセスの詳細③ー分担交通量の予測ー

分担交通量の予測は、前の段階で作成した将来のOD表を、交通手段別のOD表に分割するステップである。

全交通機関のOD表

| ゾーン | 1     | ..... | $j$      | ..... | $N$   | 発生量   |
|-----|-------|-------|----------|-------|-------|-------|
| 1   | ?     | ?     | ?        | ?     | ?     | $G_1$ |
| ⋮   | ?     | ?     | ?        | ?     | ?     | ⋮     |
| $i$ | ?     | ?     | $T_{ij}$ | ?     | ?     | $G_i$ |
| ⋮   | ?     | ?     | ?        | ?     | ?     | ⋮     |
| $N$ | ?     | ?     | ?        | ?     | ?     | $G_N$ |
| 集中量 | $A_1$ | ..... | $A_j$    | ..... | $A_N$ | $T$   |



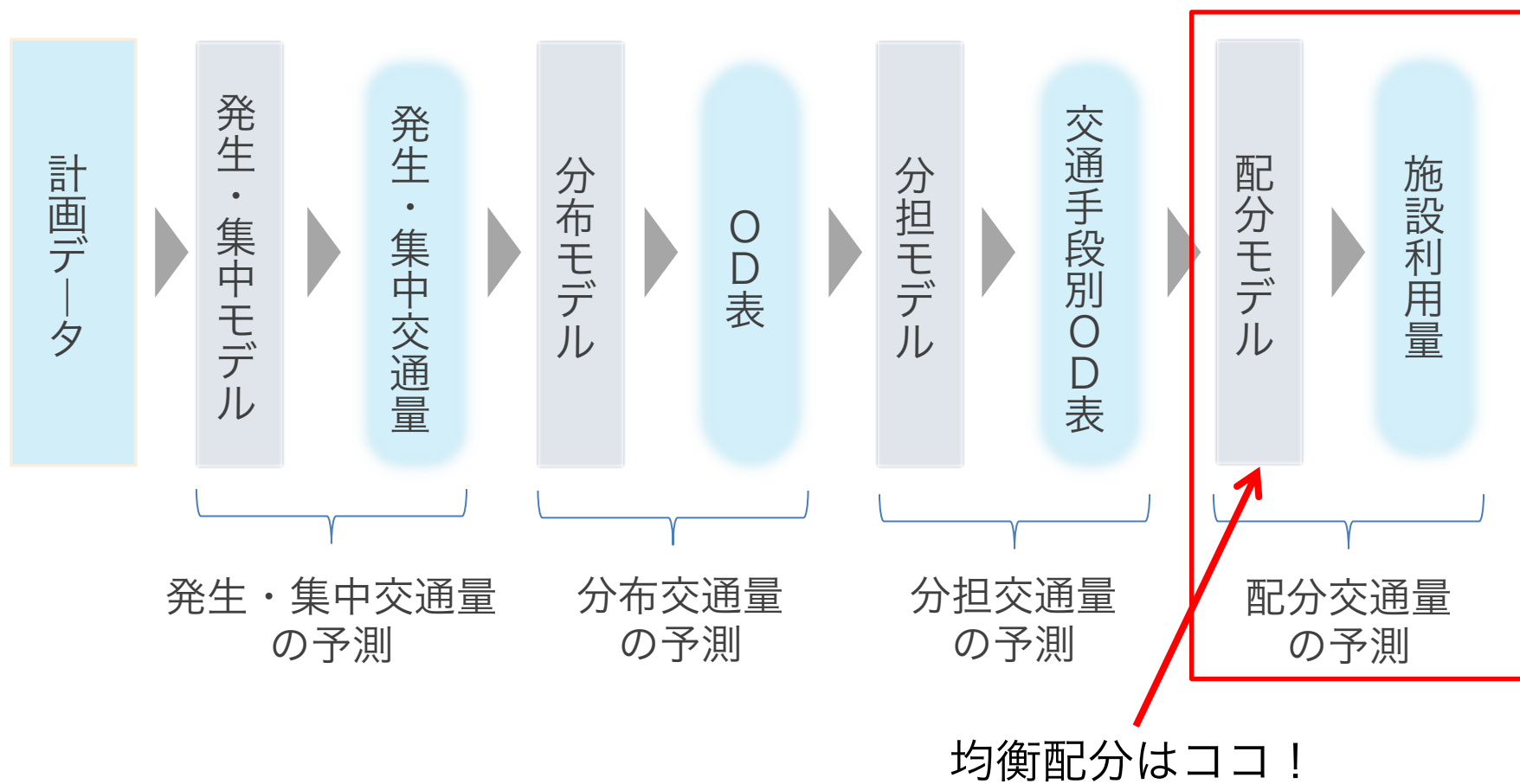
| ゾーン | 1     | ..... | $j$      | ..... | $N$   | 発生量   |
|-----|-------|-------|----------|-------|-------|-------|
| 1   | ?     | ?     | ?        | ?     | ?     | $G_1$ |
| ⋮   | ?     | ?     | ?        | ?     | ?     | ⋮     |
| $i$ | ?     | ?     | $T_{ij}$ | ?     | ?     | $G_i$ |
| ⋮   | ?     | ?     | ?        | ?     | ?     | ⋮     |
| $N$ | ?     | ?     | ?        | ?     | ?     | $G_N$ |
| 集中量 | $A_1$ | ..... | $A_j$    | ..... | $A_N$ | $T$   |

車のOD表

| ゾーン | 1     | ..... | $j$      | ..... | $N$   | 発生量   |
|-----|-------|-------|----------|-------|-------|-------|
| 1   | ?     | ?     | ?        | ?     | ?     | $G_1$ |
| ⋮   | ?     | ?     | ?        | ?     | ?     | ⋮     |
| $i$ | ?     | ?     | $T_{ij}$ | ?     | ?     | $G_i$ |
| ⋮   | ?     | ?     | ?        | ?     | ?     | ⋮     |
| $N$ | ?     | ?     | ?        | ?     | ?     | $G_N$ |
| 集中量 | $A_1$ | ..... | $A_j$    | ..... | $A_N$ | $T$   |

バスのOD表

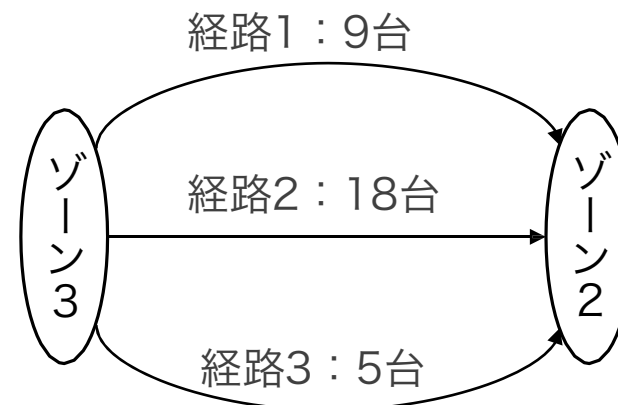
## 1.6. プロセスの詳細④ー配分交通量の予測ー



## 1.6. プロセスの詳細④－配分交通量の予測－

- 配分交通量 $T_{ijmr}$ とは、ゾーン*i*から発生しゾーン*j*に集中する交通のうち、交通機関*m*を用いて、**経路*r*を通過していく交通量**のことである。つまり、前の段階で推定した交通機関別のOD交通量を、与えられた交通網に割り当てる段階である。
- この段階の終了をもって、各交通施設の交通量が予測され、四段階推定法は終了する。

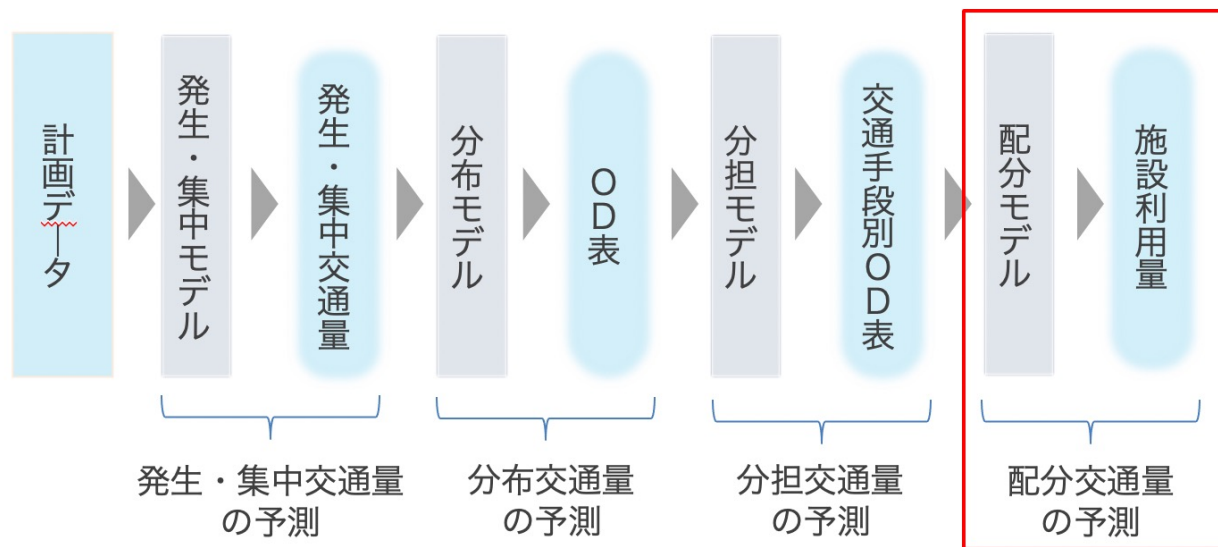
| 分布       |    |          |    |     | 分担        |    | 配分         |    |
|----------|----|----------|----|-----|-----------|----|------------|----|
|          |    | $T_{ij}$ |    |     | $G_i$     |    |            |    |
| ゾーン      | 1  | 2        | 3  | 発生  | $T_{ijm}$ |    | $T_{ijmr}$ |    |
| 1        | 11 | 12       | 20 | 43  | マストラ      | 18 | 経路 1       | 9  |
| 2        | 32 | 30       | 3  | 65  | 車         | 32 | 経路 2       | 18 |
| 3        | 8  | 50       | 55 | 113 |           |    | 経路 3       | 5  |
| $A_j$ 集中 | 51 | 92       | 78 | 221 |           |    |            |    |



## 1.6. プロセスの詳細④ー配分交通量の予測ー

- 配分交通量の予測には、配分モデルが用いられる。配分モデルのうちの1つが、均衡配分である。配分モデルについて、次のスライドから詳細に扱っていく。

## 2. 配分モデル



## 2.1. 配分モデルの種類

- 配分モデルには、①需要配分、②実際配分、③最適配分の3種類がある。
- 需要配分は、各ODペアについて、交通量が0で混雑がない場合の最短経路にOD交通量を100%割り当てる方法である。
- 最適配分はシステム最適配分（System Optimum, SO）とも呼ばれ、全利用者の総走行時間が最小になるようにOD交通量を各リンクへ配分する方法である。
- 実際配分については次のスライドを参照

## 2.2. 実際配分

- 実際配分とは、容量制限を考慮して実際の交通量により近い配分交通量を推計しようとするもの
- 実際配分の1つが**利用者均衡配分**。他には、容量制限付き分割配分法などがある
- 以下では、**利用者均衡配分**について詳細に扱っていきます。需要配分や最適配分、分割配分法などについて詳細が気になった人は各自で調べてみてください。

# 3. 均衡配分の考え方



## 3.1. リンクコストパフォーマンス関数

- 利用者均衡配分では、各経路を選択したときの旅行時間が重要
- リンクの交通量とリンクコスト（旅行時間）との関係を表す関数が、リンクコストパフォーマンス関数である。
- リンクコストパフォーマンス関数にはいくつか種類があるが、米国道路局が開発した以下のBPR関数がよく使われる。

$$t = t_0 \left\{ 1 + \alpha \left( \frac{x}{C} \right)^\beta \right\}$$

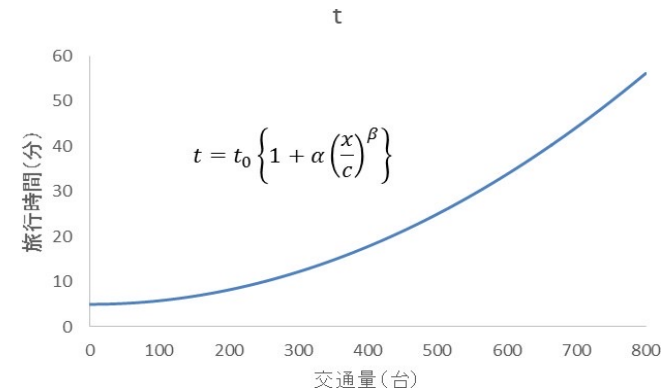
$t$ : 旅行時間

$t_0$ : 交通量0のときの旅行時間（自由旅行時間）

$C$ : 交通容量

$x$ : 交通量

$\alpha, \beta$ : パラメータ



## 3.2. 様々な利用者均衡配分モデル.

### 確定的利用者均衡配分(UE) (User Equilibrium assignment)

- 利用者は常に利用可能な経路についての**完全な情報**を得ているという前提を基に配分交通量を計算する.
- 最も基本的な配分手法である.
- 中でも, OD需要を固定したものを, **需要固定型利用者均衡配分**という.

### 確率的利用者均衡配分(SUE) (Stochastic User Equilibrium assignment)

- 利用者の経路選択の多様性や**不確実性**に着目した方法である.
- より現実的な配分モデルである.

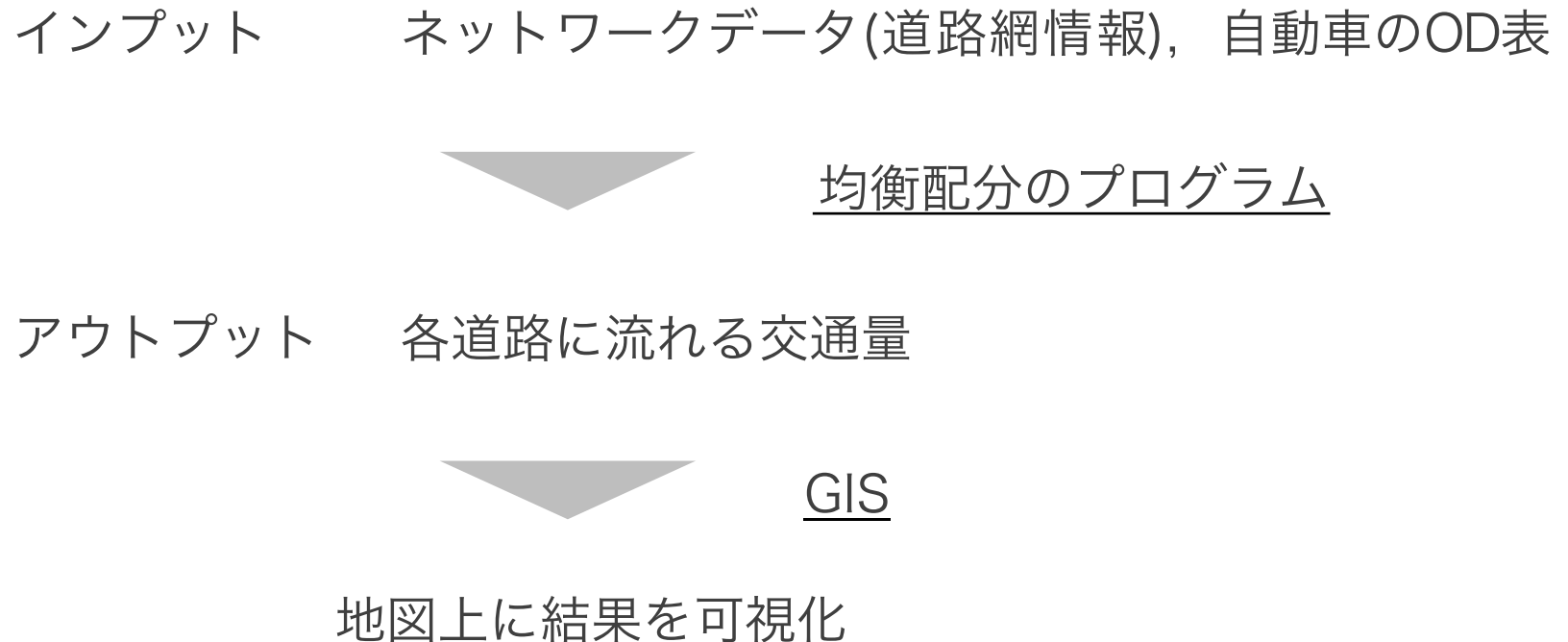
本演習では, 「**需要固定型利用者均衡配分**」を扱います. 需要固定型でないものや, 確率的利用者均衡配分について詳細が気になった人は各自で調べてみてください.

### 3.3. Wardropの第1原則

- それぞれの利用者は、常に利用可能な経路について完全な情報を得ており、自分にとって最も所要時間（3.1.のリンクコストパフォーマンス関数から計算）が少ない経路を選択するという前提を置く.
- その結果、「利用される経路の旅行時間はみな等しく、利用されない経路の旅行時間より小さいか、せいぜい等しい」という均衡状態に達する. これがWardropの第1原則である.
- 需要固定型利用者均衡配分では、最終的にWardropの第1原則を満たすように、配分交通量を逐次更新しながら繰り返し収束計算を行う（詳しいアルゴリズムについては、スライドのp.42記載の資料参照）.

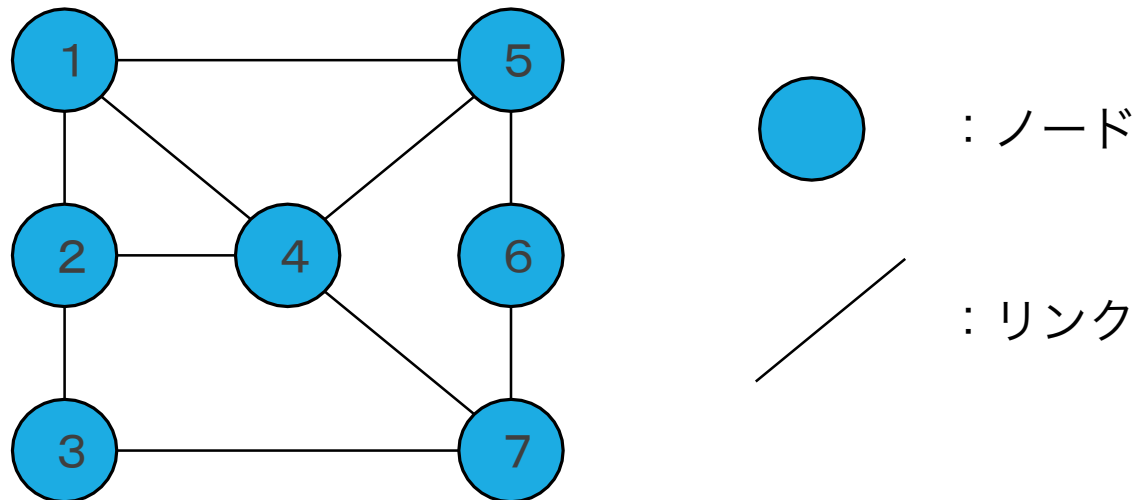
# 4. 分析と表現

## 4.1. 分析と表現の大まかな流れ



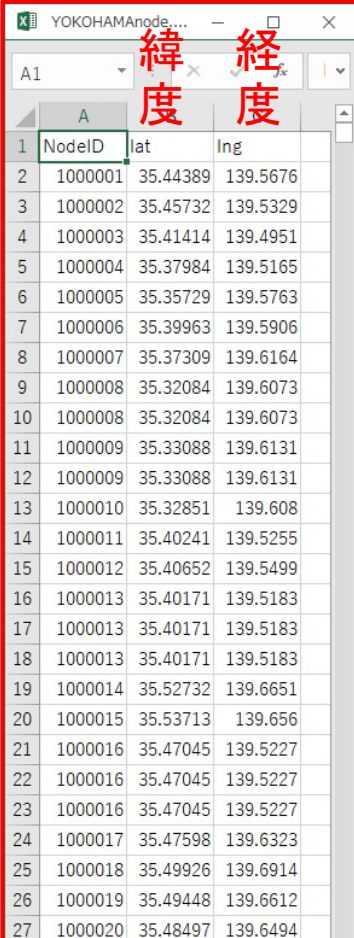
## 4.2. インプットデータ（ネットワークデータ）

- ネットワークデータ（道路網情報）は、ノードとリンクにより構成されている。



## 4.2. インプットデータ（ネットワークデータ）

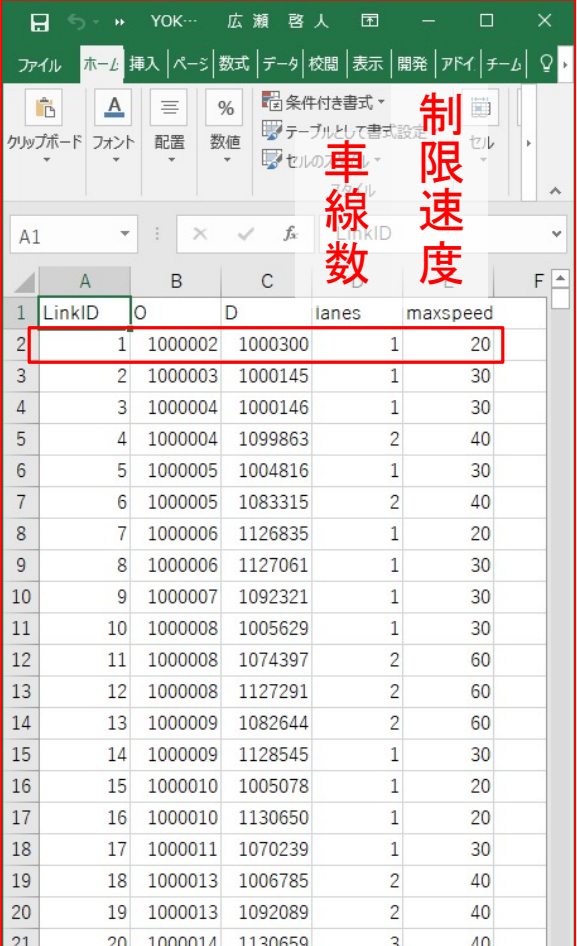
- ネットワークデータはリンクとノードにより構成されているので，ノードの情報とリンクの情報を持つ。
- ノードの情報は，そのノードが存在する緯度と経度の情報を持つ。



|    | 緯度      | 経度       |          |
|----|---------|----------|----------|
| 1  | NodeID  | lat      | lng      |
| 2  | 1000001 | 35.44389 | 139.5676 |
| 3  | 1000002 | 35.45732 | 139.5329 |
| 4  | 1000003 | 35.41414 | 139.4951 |
| 5  | 1000004 | 35.37984 | 139.5165 |
| 6  | 1000005 | 35.35729 | 139.5763 |
| 7  | 1000006 | 35.39963 | 139.5906 |
| 8  | 1000007 | 35.37309 | 139.6164 |
| 9  | 1000008 | 35.32084 | 139.6073 |
| 10 | 1000008 | 35.32084 | 139.6073 |
| 11 | 1000009 | 35.33088 | 139.6131 |
| 12 | 1000009 | 35.33088 | 139.6131 |
| 13 | 1000010 | 35.32851 | 139.608  |
| 14 | 1000011 | 35.40241 | 139.5255 |
| 15 | 1000012 | 35.40652 | 139.5499 |
| 16 | 1000013 | 35.40171 | 139.5183 |
| 17 | 1000013 | 35.40171 | 139.5183 |
| 18 | 1000013 | 35.40171 | 139.5183 |
| 19 | 1000014 | 35.52732 | 139.6651 |
| 20 | 1000015 | 35.53713 | 139.656  |
| 21 | 1000016 | 35.47045 | 139.5227 |
| 22 | 1000016 | 35.47045 | 139.5227 |
| 23 | 1000016 | 35.47045 | 139.5227 |
| 24 | 1000017 | 35.47598 | 139.6323 |
| 25 | 1000018 | 35.49926 | 139.6914 |
| 26 | 1000019 | 35.49448 | 139.6612 |
| 27 | 1000020 | 35.48497 | 139.6494 |

## 4.2. インプットデータ（ネットワークデータ）

- リンクの情報は
  - 始点ノード
  - 終点ノード
  - 車線数
  - 制限速度の4つである。



|    | A      | B       | C       |       | F        |
|----|--------|---------|---------|-------|----------|
| 1  | LinkID | O       | D       | lanes | maxspeed |
| 2  | 1      | 1000002 | 1000300 | 1     | 20       |
| 3  | 2      | 1000003 | 1000145 | 1     | 30       |
| 4  | 3      | 1000004 | 1000146 | 1     | 30       |
| 5  | 4      | 1000004 | 1099863 | 2     | 40       |
| 6  | 5      | 1000005 | 1004816 | 1     | 30       |
| 7  | 6      | 1000005 | 1083315 | 2     | 40       |
| 8  | 7      | 1000006 | 1126835 | 1     | 20       |
| 9  | 8      | 1000006 | 1127061 | 1     | 30       |
| 10 | 9      | 1000007 | 1092321 | 1     | 30       |
| 11 | 10     | 1000008 | 1005629 | 1     | 30       |
| 12 | 11     | 1000008 | 1074397 | 2     | 60       |
| 13 | 12     | 1000008 | 1127291 | 2     | 60       |
| 14 | 13     | 1000009 | 1082644 | 2     | 60       |
| 15 | 14     | 1000009 | 1128545 | 1     | 30       |
| 16 | 15     | 1000010 | 1005078 | 1     | 20       |
| 17 | 16     | 1000010 | 1130650 | 1     | 20       |
| 18 | 17     | 1000011 | 1070239 | 1     | 30       |
| 19 | 18     | 1000013 | 1006785 | 2     | 40       |
| 20 | 19     | 1000013 | 1092089 | 2     | 40       |
| 21 | 20     | 1000014 | 1130659 | 3     | 40       |



## 4.2. インプットデータ（自動車のOD表）

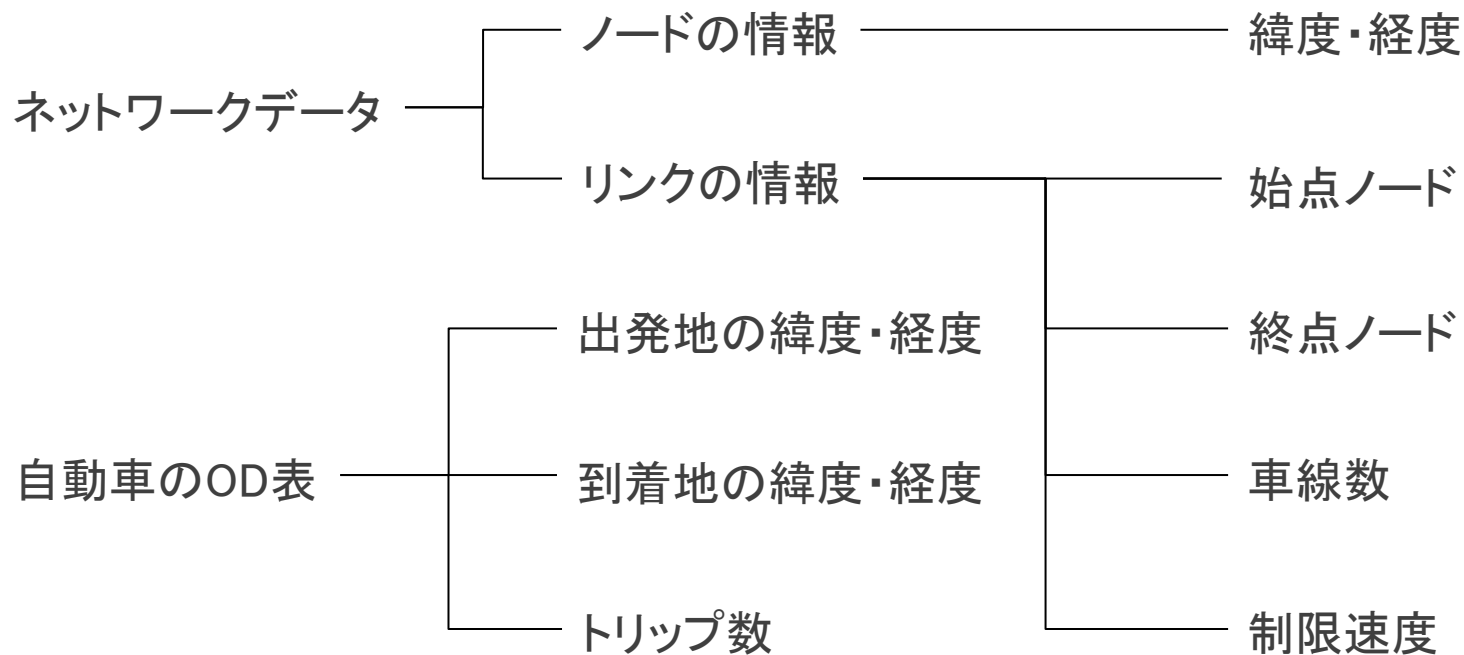
- 出発地（出発したゾーンのセントロイド）の緯度・経度
- 到着地（到着したゾーンのセントロイド）の緯度・経度
- トリップ数

The screenshot shows an Excel spreadsheet with the following data:

|    | A        | B        | C        | D        | F      |
|----|----------|----------|----------|----------|--------|
| 1  | olat     | olon     | dlat     | dlon     | number |
| 2  | 35.45668 | 139.6183 | 35.45668 | 139.6183 | 7626   |
| 3  | 35.45668 | 139.6183 | 35.43028 | 139.6559 | 6710   |
| 4  | 35.45668 | 139.6183 | 35.39219 | 139.614  | 1407   |
| 5  | 35.45668 | 139.6183 | 35.42531 | 139.5873 | 2058   |
| 6  | 35.45668 | 139.6183 | 35.33581 | 139.6051 | 1545   |
| 7  | 35.45668 | 139.6183 | 35.39426 | 139.5782 | 1443   |
| 8  | 35.45668 | 139.6183 | 35.4613  | 139.5783 | 3627   |
| 9  | 35.45668 | 139.6183 | 35.40913 | 139.5301 | 1457   |
| 10 | 35.45668 | 139.6183 | 35.36859 | 139.5354 | 279    |
| 11 | 35.45668 | 139.6183 | 35.42068 | 139.4993 | 694    |
| 12 | 35.45668 | 139.6183 | 35.47645 | 139.5275 | 1591   |
| 13 | 35.45668 | 139.6183 | 35.47    | 139.4881 | 608    |
| 14 | 35.45668 | 139.6183 | 35.52814 | 139.6228 | 1715   |
| 15 | 35.45668 | 139.6183 | 35.54189 | 139.5775 | 851    |
| 16 | 35.45668 | 139.6183 | 35.52098 | 139.5045 | 814    |
| 17 | 35.45668 | 139.6183 | 35.54999 | 139.518  | 564    |
| 18 | 35.45668 | 139.6183 | 35.51591 | 139.6732 | 1336   |
| 19 | 35.45668 | 139.6183 | 35.48146 | 139.6185 | 5114   |
| 20 | 35.43028 | 139.6559 | 35.45668 | 139.6183 | 6059   |
| 21 | 35.43028 | 139.6559 | 35.43028 | 139.6559 | 48175  |

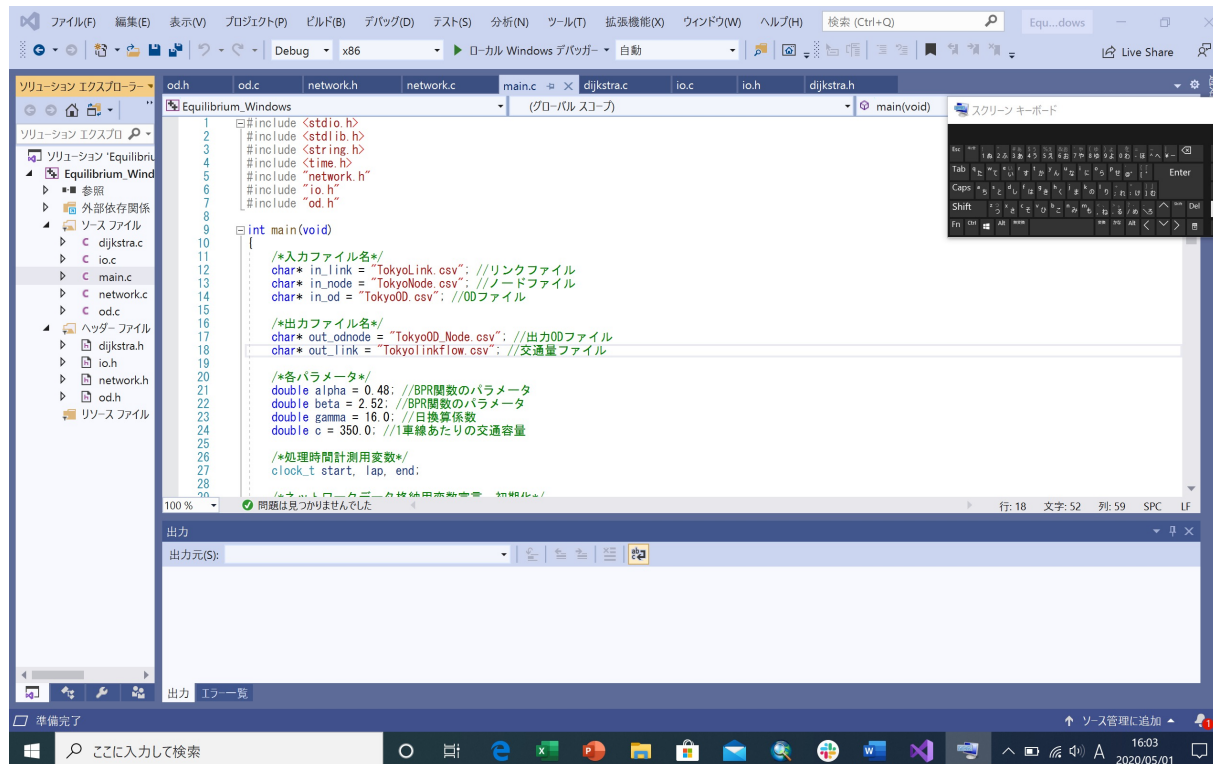
## 4.2. インプットデータ（まとめ）

- まとめると，以下の情報をインプットしている。



## 4.3. プログラム

- 均衡配分のコードはC言語で書かれている。
- XcodeやMicrosoft Visual Studioという統合開発環境を用いる。



## 4.3. プログラム

3つの入力ファイル  
(ノードの情報, リンクの情報, 自動車のOD表)

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <time.h>
5  #include "network.h"
6  #include "io.h"
7  #include "od.h"
8
9  int main(void)
10 {
11     /*入力ファイル名*/
12     char* in_link = "TokyoLink.csv"; //リンクファイル
13     char* in_node = "TokyoNode.csv"; //ノードファイル
14     char* in_od = "TokyoOD.csv"; //ODファイル
15
16     /*出力ファイル名*/
17     char* out_odnode = "TokyoOD_Node.csv"; //出力ODファイル
18     char* out_link = "TokyoLinkFlow.csv"; //交通量ファイル
19
20     /*各パラメータ*/
21     double alpha = 0.48; //BPR関数のパラメータ
22     double beta = 2.52; //BPR関数のパラメータ
23     double gamma = 16.0; //日換算係数
24     double c = 350.0; //1車線あたりの交通容量
25
26     /*処理時間計測用変数*/
27     clock_t start, lap, end;
28
29     /*...*/
30 }

```

リンクコストパフォーマンス  
関数のパラメータを設定

$$t = t_0 \left\{ 1 + \alpha \left( \frac{x}{c} \right)^\beta \right\}$$

出力

出力元(S):

準備完了

ここに入力して検索

16:03  
2020/05/01

## 4.3. プログラム

```

/*各パラメータ*/
double alpha = 0.48; //BPR関数のパラメータ
double beta = 2.52; //BPR関数のパラメータ
double gamma = 16.0; //日換算係数
double c = 350.0; //1車線あたりの交通容量

```

$$t = t_0 \left\{ 1 + \alpha \left( \frac{x}{C} \right)^\beta \right\}$$

$t_0$ : 交通量0のときの旅行時間

$C$ : 交通容量,  $x$ : 交通量,  $\alpha, \beta$ : パラメータ

$\alpha, \beta$

そのまま

$$C = l_a \cdot c \cdot \gamma_a$$

$l_a$ : リンクaの車線数

$c$ : 一車線当たりの交通容量

$\gamma_a$ : 日換算係数

$t_0$

リンク毎の制限速度の逆数から

## 4.3. プログラム

```
/*各パラメータ*/  
double alpha = 0.48; //BPR関数のパラメータ  
double beta = 2.52; //BPR関数のパラメータ  
double gamma = 16.0; //日換算係数
```

$$t = t_0 \left\{ 1 + \alpha \left( \frac{x}{C} \right)^\beta \right\}$$

リンクコストパフォーマンス関数のパラメータや、ネットワークデータ、トリップデータに手を加えて交通流の分析をしてみましょう！

 $t_0$ 

リンク毎の制限速度の逆数から

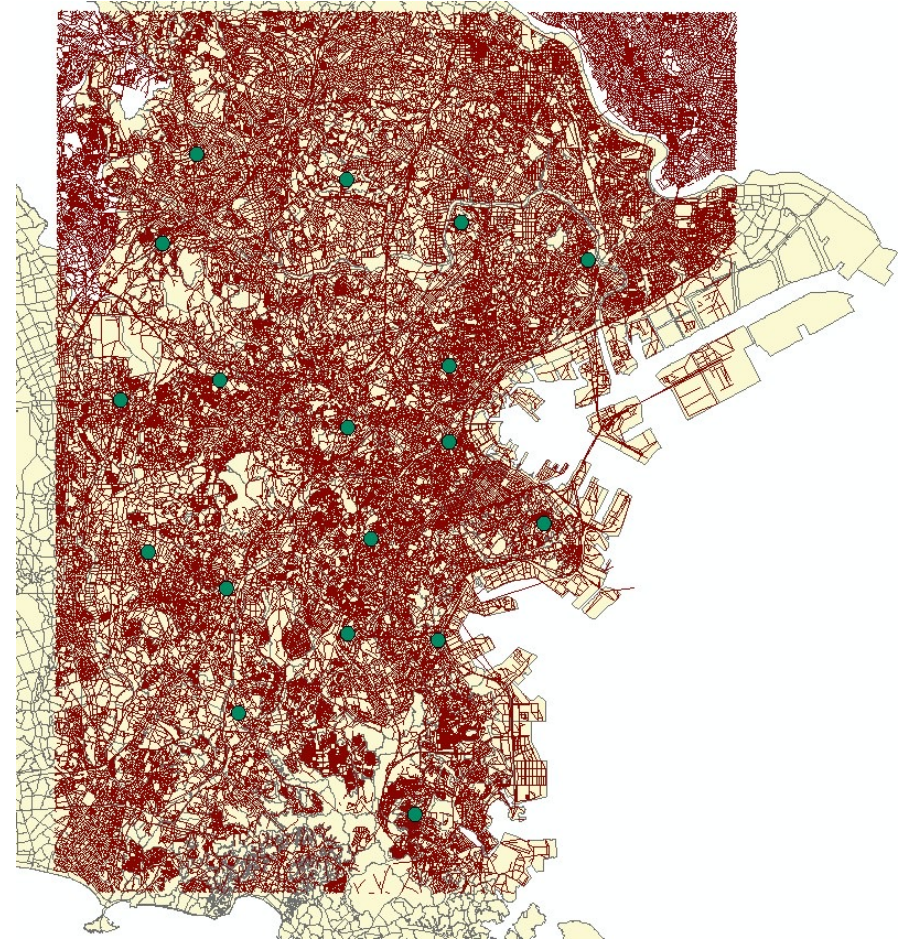
タ

## 4.4. 分析と表現の例

- 横浜市の自動車交通を配分する例
- 横浜市を18のゾーン（区）に分ける
- ネットワークデータ（道路網情報）と、自動車のOD表をインプットし、均衡配分のプログラムを回して、ゾーン間の自動車トリップを需要固定型利用者均衡配分により配分する。

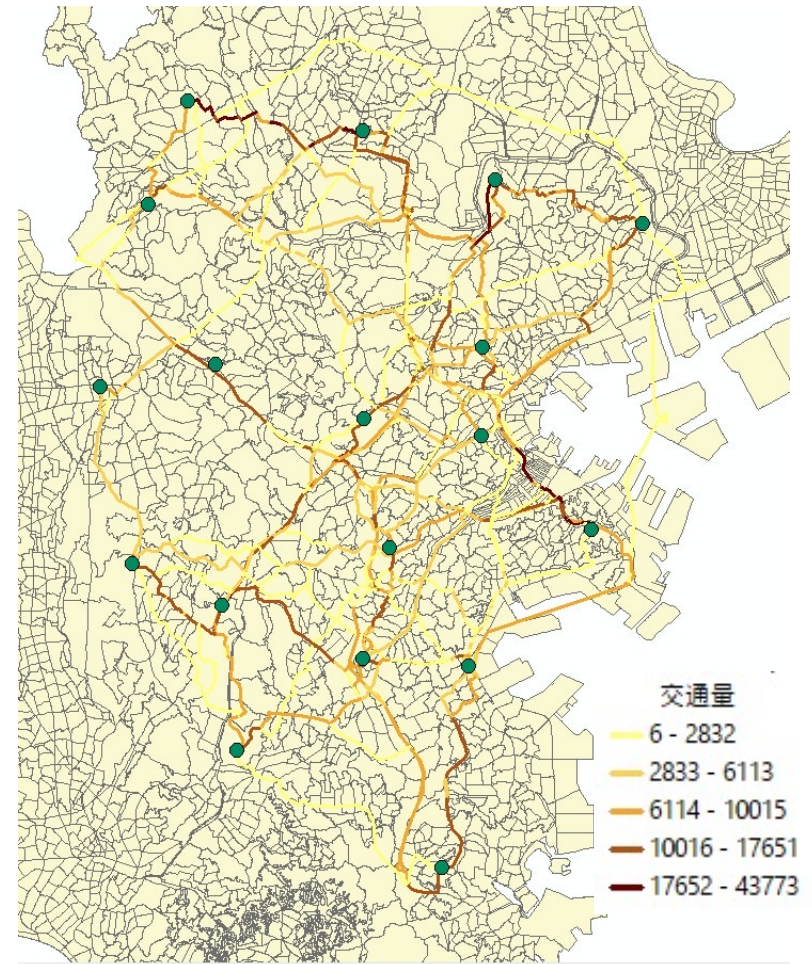
● 各区のセントロイド

横浜市の道路網とセントロイド



## 4.4. 分析と表現の例

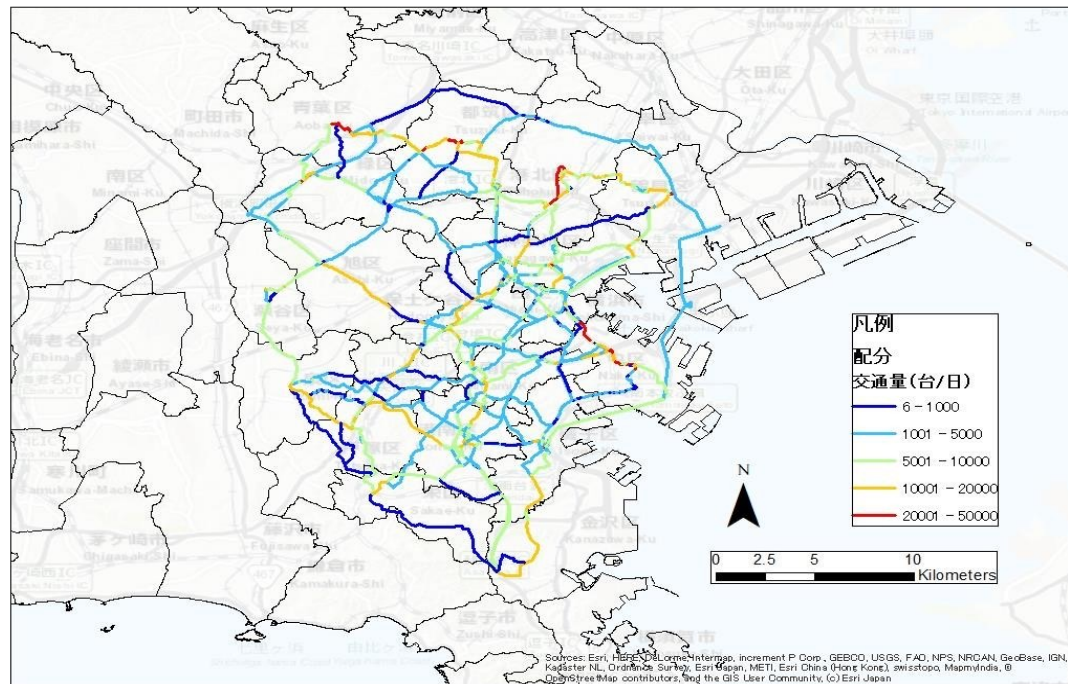
- 配分結果をQGISを用いて可視化する
- さらにゾーンを細かく区切って、各ゾーンにセントロイドをおき、ゾーン間トリップをインプットすればさらに細かい分析も可能





## 4.4. 分析と表現の例

- 先ほどの可視化したものを更に見やすくしたもの。
- 赤に近いほど交通量が多いため、ボトルネックになりやすい。



---

## 江東区の分析に向けて

- ある道路の車線を減らして歩行者空間を増やすとその道路の交通量はどうなるだろうか？
- 災害時に被災地→避難場所の交通が大きく増加すると、どの道路が使われるだろうか？
- そもそもリンクコストパフォーマンス関数は正しいのか？

均衡配分のインプットデータやパラメータを変更することで、政策・提案を考える上での疑問に対して定量的に答えを出し、提案につなげていきましょう！

---

## 補足その1

更に詳しい説明については、以下のファイルを参照してください。  
全てGoogle Driveの中の「課題資料→均衡配分」の中にあります。

- 均衡配分のアルゴリズム：

「需要固定型利用者均衡配分のアルゴリズム.pdf」

- プログラムの実行・可視化までの一連の流れに関する、実際的な手続き：

「均衡配分のマニュアル（Mac版）.pdf」

または「均衡配分のマニュアル（Windows版）.pdf」

- 均衡配分で分析可能な事項についての詳細とその方法：

「均衡配分で分析可能な事項とその方法.pdf」

## 補足その2

- マニュアルではMacの統合開発環境としてXcodeを使う方法を記載していますが、Xcodeはインストール時に大きな空き容量が必要なので、インストールできないかもしれません。
- その場合は統合開発環境としてCLionをインストールするとよいでしょう。  
(手順)
  1. [このサイト](#)を参考にCLionをインストールします。
  2. Google Driveの「課題資料 > 均衡配分 > code > Equilibrium\_Mac > CLion > equilibrium」を自分のPCにダウンロードします。
  3. CLionを起動し、メニューから「ファイル|開く」を選択し、2.でダウンロードしたequilibriumフォルダを選択し開きます。 [こちらのサイト](#)を参考にすると良いでしょう。
  4. 以降はXcodeの場合と同じく、main.cでインプットファイルとアウトプットファイルのパスを指定し、プログラムを実行すればOKです。

---

## 課題（選択）

東京で均衡配分を実行し、その結果をQGISに表現してください。

実際的な方法は、Google Driveの「課題資料→均衡配分→manual」フォルダ内の「均衡配分のマニュアル（Mac版）.pdf」または「均衡配分のマニュアル（Windows版）.pdf」に書いているので、これを参照してください。

※些細なことでもわからないところがあれば、slackの均衡配分チャンネル等で質問してください！