

交通量配分

ここではまず(1)~(4)の4節に分けて、確定的利用者均衡配分の基礎理論について説明します。次に(5)では計算手順の大まかな流れを説明し、(6)で実際に利用するプログラムの説明を行い、(7)では実際の計算例を示します。

1 確定的利用者均衡配分の基礎概念

効率的な道路政策を進めるために道路投資の効果をいかに評価するかということが重要になってきている。そのためには道路整備前と後で交通量や、その旅行時間を推計することが必要となってくる。また従前は交通量の再現や予測にその焦点が当てられていたが、近年はマルチモーダル、*TDM*、*ITS* 施策といったソフト施策の評価が重要になってきている。そこで確定的利用者均衡配分による交通量推計が行われるようになった。確定的利用者均衡配分とは以下に示す *Wardrop* の第一原則を仮定したモデルである。

— *Wardrop* の第一原則(等時間原則) —

それぞれのドライバーは自分にとって最も旅行時間の短い経路を選択する。その結果として、起終点間に存在する経路のうち、利用される経路の旅行時間は皆等しく、利用されない経路の旅行時間よりも小さいか、せいぜい等しいという状態となる。

2 確定的利用者均衡配分モデル

ここでは確定的利用者均衡配分のモデルについて扱う。まずモデルを構築するにあたっての前提条件を二つ仮定する。

1. 起終点の旅行時間は、その通過経路を構成するリンクの旅行時間の和で表わされる。
2. 各リンクのリンク旅行時間はそのリンクの容量と交通量のみによって決まり、他のリンク(対向車線や交差する道路等)の交通状況には影響を受けない。

確定的確定的利用者均衡配分問題は、すべての *OD* 間のフロー数が与えられたときに *Wardrop* の第一原則を満たすリンクフローを求める問題である。まず定式化の際に用いる変数の定義を表 3.1 に示す。

表 3.1 変数の定義

a	リンク
r	出発地
s	目的地
k	バス(経路)
x_a	リンク a におけるフロー数
t_a	リンク a における旅行時間
f_k^{rs}	OD ペア rs を結ぶバス k におけるフロー数
c_k^{rs}	OD ペア rs を結ぶバス k における旅行時間
q_{rs}	OD ペア rs 間の総フロー数
$\delta_{a,k}^{rs}$	リンク a が OD ペア rs を結ぶバス k 上にあるとき 1 , それ以外のはきは 0

先の二つの前提条件を仮定したとき *Wardrop* の第一原則が成り立っている状態を数式で表現すると式(2.1)のようになる。

$$(c_k^{rs} - u_{rs}) \geq 0$$

$$c_k^{rs} - u_{rs} = 0 \quad (2.1)$$

ここで u_{rs} は OD ペア rs 間の最小旅行時間である。次に利用者均衡条件を満たす、つまり式(2.1)を満たすような等価な数学的問題に置き換える。これはすべての OD 間のフロー数 q_{rs} が与えられたときに、利用者均衡条件を満たす全てのリンクフロー x_a を求める問題である。このとき、式(2.2)のように定式化できる。

$$\min z(x) = \sum_a \int_0^{x_a} t_a(\omega) d\omega \quad (2.2)$$

ここで x はすべてのリンクを表わすベクトルで、 $x=(\dots, x_a, \dots)$ である。制約条件は、式(2.3)、式(2.4)のようになる。

$$\sum_k f_k^{rs} = q_{rs} \quad (2.3)$$

$$f_k^{rs} \geq 0 \quad (2.4)$$

また当然満たされるべき関係として式(2.5)、式(2.6)がある。

$$x_a = \sum_r \sum_s \sum_k f_k^{rs} \delta_{a,k}^{rs} \quad (2.5)$$

$$c_k^{rs} = \sum_a t_a \delta_{a,k}^{rs} \quad (2.6)$$

3 リンクパフォーマンス関数

前節(2)で、 t_a (リンク a における旅行時間)を使用して配分問題の定式化を行った。ここでは、リンクにおける旅行時間を求めるために使用されるリンクパフォーマンス関数について説明する。

リンクパフォーマンス関数とは、ネットワークを構成する個々のリンクのサービス水準(一般的に旅行時間)をリンク交通量とリンク属性の関数として表したものである。確定的利用者均衡配分計算に用いられるリンクパフォーマンス関数として、代表的なものは *BPR* 関数と *Davison* 関数が知られている。ここでは *BPR* 関数について説明する。

BPR 関数は、1964年に米国で作成され、実用的なリンクパフォーマンス関数として研究や実務の場面において古くから利用されている。概形は図(3.1)に示す。

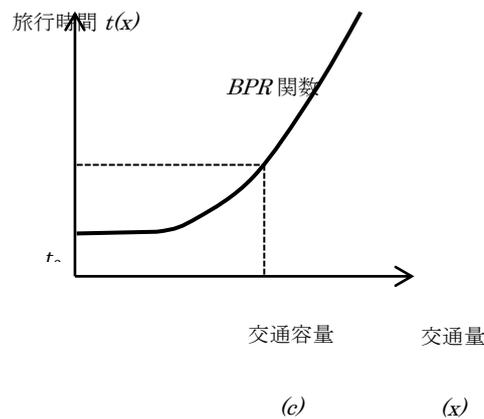


図 3.1 リンクパフォーマンス関数概形

BPR 関数は式(3.2)に示したような関数で、旅行時間をリンク交通量と以下に示すようなリンク属性で表している。

$$\text{BPR 関数: } t_a(x_a) = t_{a0} \left\{ 1 + \alpha \left(\frac{x_a}{C_a} \right)^\beta \right\} \quad (3.2)$$

t_a : リンク a の旅行時間

t_{a0} : リンク a の自由旅行時間

x_a : リンク a の時間交通量(台/時)

C_a : リンク a の時間交通容量(台/時)

α, β : パラメータ

なお、配分計算のアルゴリズムとして *Frank-Wolfe* 法(次節説明)などを用いる場合には、交通量のすべての領域で定義されたリンクパフォーマンス関数(*BPR* 関数など)の適用が前提とされている。

4 確定的利用者均衡配分問題解法アルゴリズム

(2)では、配分問題の定式化を行った。ここではその問題を解くアルゴリズムについて説明を行う。確定的利用者均衡配分の解法の一つ線形計画法である *Frank-Wolfe* 法を説明する。*Frank-Wolfe* 法は計算手順が比較的簡単でプログラムを作成しやすいという長所がある一方で、均衡解に近づくに従い収束が緩慢になる。具体的なアルゴリズムを以下に示す。また u_{rs}^n は n 回目の試行を行ったときの *OD* 間最小旅行時間を表わす。

ステップ 0: 初期実行可能解の設定

すべてのリンクの交通量 $x_a^0 = 0$ としリンク旅行時間 t_a^0 を計算する。この旅行時間に基づき *all-or-nothing* 配分(交通量 0 のときのリンク旅行時間を用いた最適経路配分)により初期実行可能解(リンク交通量の初期値) x_a^1 を設定する。このとき繰り返し計算の試行回数 $n=1$ とする。

ステップ 1: リンク旅行時間の更新

リンク交通量 x_a^n に対するリンク旅行時間 t_a^n を計算する。このとき *OD* ペアごとの最小旅行時間 u_{rs}^n を計算する。

ステップ 2: 降下方向ベクトルの探索

リンク旅行時間 t_a^n の状態ですべての *OD* 間の交通量を最短経路に配分し、そのときの交通量を y_a^n とする。 $d_a^n = y_a^n - x_a^n$ によって降下方向ベクトル d_a^n を計算する。

ステップ 3: 降下ステップサイズの探索

式(4.1)のような x_a^{n+1} に対して式(2.2)で表される目的関数の値が最も小さくなるような α^n を探索する。すなわち式(4.2)を解くことで α^n を探索すればよい。得られた α^n によって x_a^{n+1} を計算する。新たなリンク交通量から旅行時間 t_a^{n+1} を計算し、*OD* ペアごとの最小旅行時間 u_{rs}^{n+1} を計算する。

$$x_a^{n+1} = x_a^n + \alpha^n d_a^n = x_a^n + \alpha^n (y_a^n - x_a^n) \quad (4.1)$$

$$\min_{0 \leq \alpha \leq 1} \sum_a \int_0^{x_a^n + \alpha^n (y_a^n - x_a^n)} t_a(\omega) d\omega \quad (4.2)$$

ステップ 4: 収束判定

式(4.3)の収束条件(この収束判定基準は唯一のものではない)が満たされていれば、 $n=n+1$ としステップ 1 へ戻る。収束条件が満たされていれば、計算を終了しリンク交通量 x_a^{n+1} を解として出力する。ただし、式(4.3)の中の κ はあらかじめ決定された定数とする。

$$\sum_{rs} \frac{|u_{rs}^{n+1} - u_{rs}^n|}{u_{rs}^{n+1}} \leq \kappa \quad (4.3)$$

5 計算ダイアグラム

交通量配分計算は以下のような手順で計算を行います。

インプット

①リンク情報:リンクの始点と終点(上り下りは区別)・交通容量・リンク長・自由旅行時間・

BPR 関数のパラメータ α, β ・制限速度



ゾーンごとにセントロイドを設定して、セントロイドを起点とした交通量配分を考える。



リンクパフォーマンス関数の設定し、リンクごとの旅行時間と時間交通量の関係を求める

(今回のプログラムでは *BPR* 関数を使用)



配分計算

Wardrop の第一原則(等時間原則)を満たすように数理最適化問題を解く



アウトプット

交通量・リンク通過速度・リンク混雑度

実際にプログラムについて、(a)プログラム利用方法、(b)入力ファイルの形式について説明します。

(a)プログラム利用方法

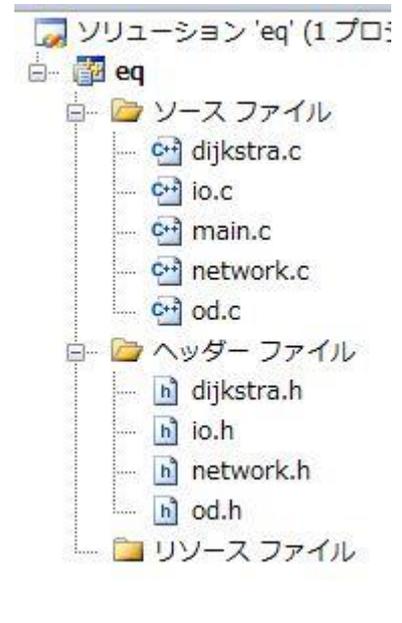
今回は、C 言語を用いて配分計算を行います。

配分プログラムに用いたファイルは右図のようにソースファイル 5 つ、ヘッダーファイル 4 つとなっています。

基本的にはソースファイル main.c 中のファイルの読み込みの部分を変更することでサンプルデータ以外でも計算を行うことが可能です。

下に main.c ファイルの一部を示します。

```
int main(void){
  /******
  /* 入力・出力データ */
  char *in_link= "YOKOHAMAlink.csv"; //リンクファイル
  char *in_node= "YOKOHAMAnode.csv"; //ノードファイル
  char *in_od= "YOKOHAMAOD.csv"; //ODデータファイル1
  char *in_odnode= "YOKOHAMAOd_node.csv"; //ODデータファイル2
  char *out_link= "YOKOHAMAlinkflow.csv"; //出力データ
  /******
  /* 設定パラメータ */
  double alpha = 0.48; //BPR関数のパラメータ
  double beta = 2.82; //BPR関数のパラメータ
  double gamma = 16.0; //日換算係数
  double c = 1500.0; //1車線あたり交通容量(台/時)
```



ダブルクォーテーションで囲まれた部分がファイル名でそれぞれリンクファイル、ノードファイル、OD データファイル 1、OD データファイル 2、出力データに対応しています。またこの場合は配分プログラムのプロジェクトの直下にデータを置いているためディレクトリーの指定などはありませんが、例えばデスクトップ上にリンクファイルがある場合であれば”YOKOHAMAlink.csv”を”C:/Users/(ここはユーザー名)/Desktop/YOKOHAMAlink.csv”に置き換えれば同様にプログラムを回すことが可能です。出力データについても同様にその場所を指定することができます。

(b)入力ファイルの形式

入力するファイルの形式について説明します。入力ファイルを以下のような形式に変更することでサンプルデータ以外についても同様にプログラムを用いて計算することができます。

①リンクファイルについて(YOKOHAMAlink.csv)

LinkID:リンクの ID 番号

O:リンクの起点ノード番号

D: リンクの終点ノード番号

lanes: リンクの車線数

maxspeed: リンクの制限速度

②ノードファイルについて(YOKOHAMAnode.csv)

NodeID: ノードの ID 番号

lat: ノードの緯度

lon: ノードの経度

③OD ファイルについて 1(YOKOHAMAOD.csv)

olat: 出発ノードの緯度

olon: 出発ノードの経度

dlat: 到着ノードの緯度

dlon: 到着ノードの経度

number: OD 間交通量

④OD ファイルについて 2(YOKOHAMAOd_node.csv)

onode: 起点ノード ID

dnode: 終点のード ID

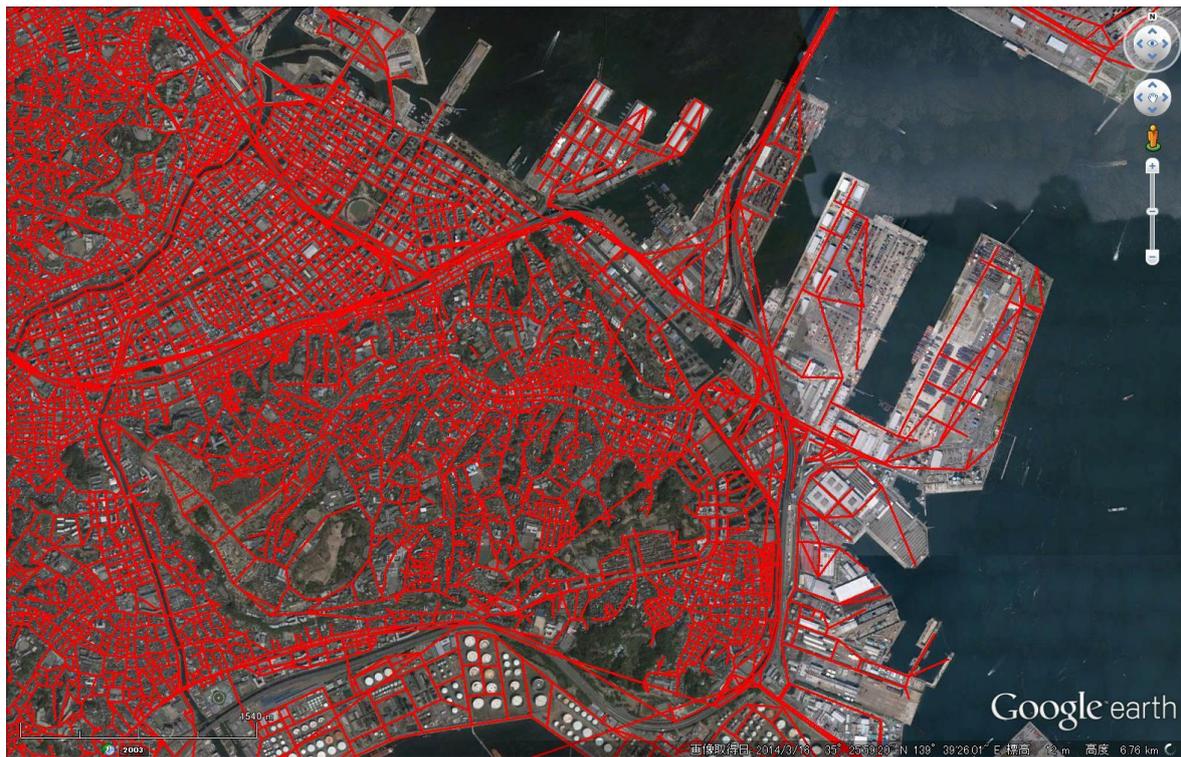
number: OD 間交通量

7 計算例

横浜のネットワークデータを用いて、配分計算をしました。入力情報と出力結果について説明します。

(a)入力情報について

ネットワークに関しては、OSM から取得したデータを使用しています。データの一部を以下に示します。

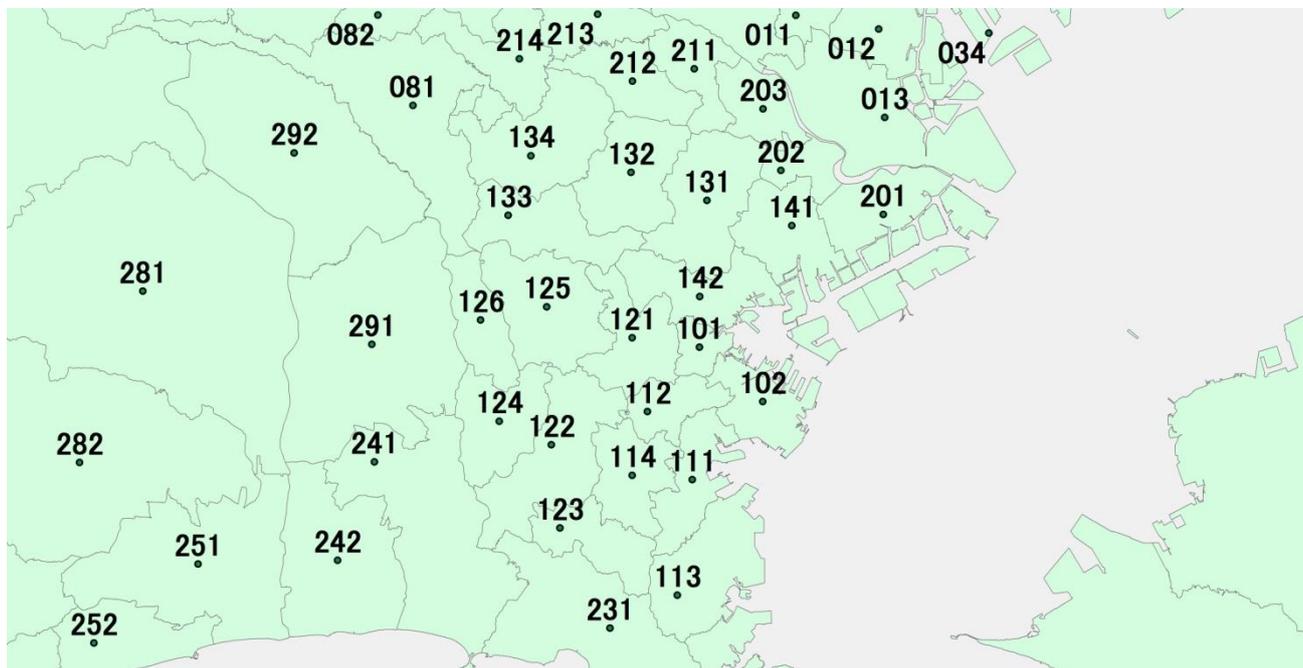


ノード数: 149545

リンク数: 379000

OD ペア数: 324

ゾーンは中ゾーンで、計算には横浜市に含まれる以下の図の 101, 102, 111, 112, 113, 114, 121, 122, 123, 124, 125, 126, 131, 132, 133, 134, 141, 142 の 18 ゾーンで行いました。



セントロイドに関しては、ゾーンの中心部を求め、そこに最も近い交差点に配置しています。

各ゾーン中のセントロイドの情報は centroid.csv ファイルを参照してください。

次に、リンク情報と自動車の OD 表の情報に関しては、以下のように設定しました。

①リンク情報

道路容量(台/日): 可能交通容量 1500 台/時/車線・日換算係数 16.0として導出

リンク長(km): OSM(Open Street Map)から取得

自由旅行時間 (分): リンク長を制限速度で割ることで導出

α, β [BPR 関数のパラメータ]: $\alpha = 0.48, \beta = 2.82$

制限速度(km/時): 全道路 20km/時と設定

②自動車の OD 表

(b) 出力結果

出力結果は以下のようになります.

ID: リンクの ID 番号

O: リンクの起点ノードの ID 番号

D: リンクの終点ノードの ID 番号

交通量: 配分後のリンク交通量

また, C 言語のコンソール画面は以下のように表示されます.

```
ノード数 : 149545
リンク数 : 379000
ODペア数 : 324
1回目   z = 5543673.778172   delta = 14255925.309256   xi = 0.618034
2回目   z = 5533811.717408   delta = 4883178.782518   xi = 0.763932
3回目   z = 5533721.737012   delta = 374631.045723   xi = 0.978714
4回目   z = 5533679.700930   delta = 265155.889022   xi = 0.983912
5回目   z = 5533642.661532   delta = 232580.786461   xi = 0.985938
6回目   z = 5533642.661532   delta = 0.000000   xi = 1.000000
計算時間 : 323.845000[s]

Program finished complete . Please press any key .
```