

2017/6/15

交通ネットワーク分析ゼミ第2回②

- 第8章 利用者均衡モデルの解法

朝倉研B4 小泉大哉

# 目次

---

- 最短経路探索モデル

- Dijkstra法(ダイクストラ法、ラベル確定法)
  - ラベル修正法
  - 効率的な最短経路探索のためのデータ構造

← 割愛

- 利用者均衡モデルの解法

- Frank-Wolfe法**

- リンク容量制約を持つ場合のFrank-Wolfe法

- 利用者均衡配分のより効率的な解法

- 打ち切り二次計画法

- Simplicial Decomposition法

- リンク間に相互干渉のある利用者均衡配分の解法

- 緩和法(非線型対角化法)

- 射影法(線形近似法)

- Gap関数を利用した解法

# all-or-nothing配分

---

概要：最小交通費用を持つ経路へすべての需要を配分するもの。  
以下のステップを通じて配分する。

- ステップ1：交通が発生するすべての起点に対して最短経路探索を行い、各ノード $i$ に対する先行ポインタ $F_i$ を求める。
- ステップ2：各ODペア $rs$ 間において、最短経路上のリンクを $F_i$ を利用しながら列挙すると同時に、OD交通量 $Q_{rs}$ を列挙されたリンクに負荷していく
- ステップ3：全ODペア間のOD交通量を最短経路上のすべてのリンクに負荷する→終了

# 最適経路探索モデル

## -効率的な最短経路探索のためのデータ構造

- forward star形式のデータ構造

各リンクのデータを1行ごとに並べた上で、データを読み出すための配列表を作成する。

表-8.5 forward star形式のデータ構造

リンクナンバー	始点ノード	終点ノード	リンクコスト
1	1	2	2
2	1	4	4
3	1	5	9
4	2	3	8
5	2	4	6
6	3	7	1
7	4	5	4
8	4	7	2
9	5	4	1
10	5	6	1
11	6	5	2
12	6	7	3
13	7	3	3
14	7	6	1

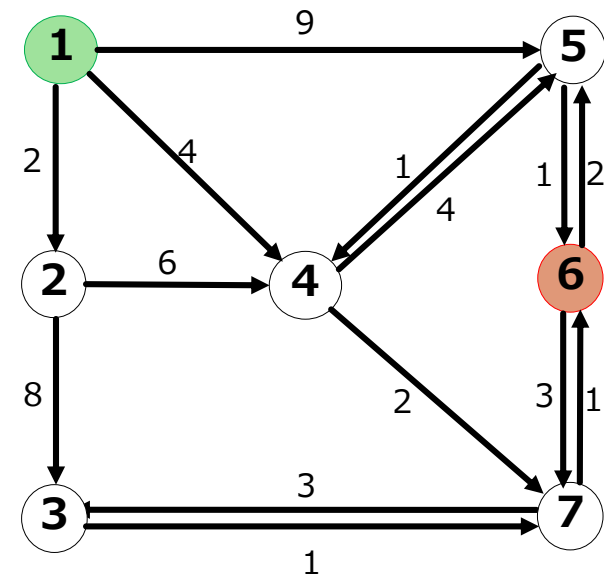


表-8.4 forward star形式データを呼び出すための配列 B

始点ノード	各始点ノードの始めのリンクナンバー
i	B(i)
1	1
2	4
3	6
4	7
5	9
6	11
7	13

B(i) : ノードに繋がるリンクの終点ノード  
→リンクナンバー

# 利用者均衡モデルの解法

---

＜利用者均衡状態とは＞

- 交通利用者は、経路に関する情報を完全に得ている
- 交通利用者は、最短経路を選択する

上記2つの前提のもとで、各交通利用者が行動した結果生じる状態のこと。

- 利用者均衡状態となるような交通量配分方法を、利用者均衡配分という。

# 利用者均衡モデルの解法

---

- Wardropの第一原則(利用者均衡)

「起終点間に存在する経路のうち、利用される経路の所要時間は皆等しく、利用されない経路の所要時間よりも小さいか、せいぜい等しい。」

- Wardropの第二原則(システム最適)

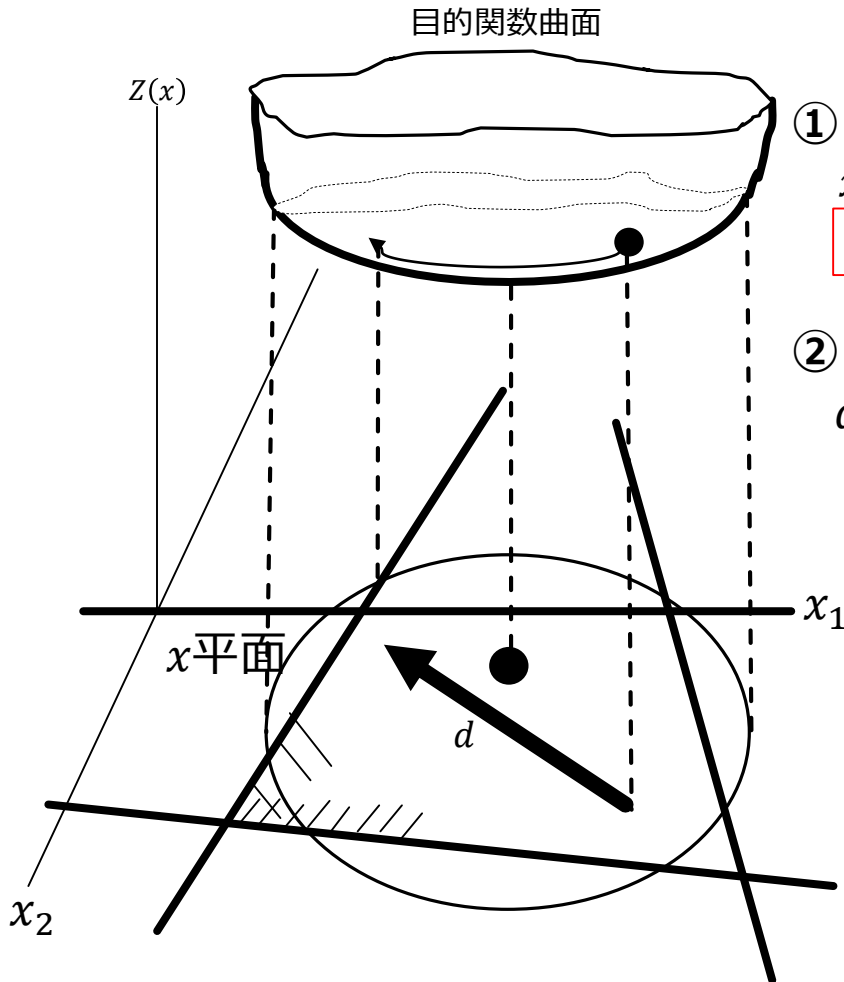
「交通ネットワーク上の総旅行時間(交通量×時間)が最小となる。」

# 利用者均衡モデルの解法

---

- 非線形最適化問題の解法
  - 変数が少なければクーン・タッカー条件でOK
  - しかし、変数の数が膨大になると面倒
- 考えるべきことは2つ
  - ①：最適化へ向かう降下方向ベクトルの識別
  - ②：①で求めた方向に、どれだけ進むか？

# 利用者均衡モデルの解法



① : 最適化へ向かう方向ベクトルの識別

$x^{(n)}$  :  $n$ 回計算後のリンク交通量ベクトル

$d = y - x^{(n)}$  ←  $y$ をどう置くか？

② : ①で求めた方向に、どれだけ進むか？

$\alpha$  : 降下距離を表す1次パラメータ

$(0 \leq \alpha \leq 1)$

$x^{(n+1)} = x^{(n)} + \alpha(y - x^{(n)})$

目的関数 $Z(x^{(n+1)})$ を最小にするもの

- : 最小点 $x^{(n+m)}$
- ➔ : 降下方向ベクトル $d = y - x^{(n)}$
- : 制約条件



## 最適経路探索モデル-Frank-Wolfe法

---

- 計算プログラムの作成が非常に簡単で、利用者均衡モデルのアルゴリズムとして広く用いられている
- リンク交通量ベクトルをベースとして最適解を探索するため、(列挙された)記憶容量が少なくてすむ
- 収束が進むにつれて収束スピードが遅くなることが知られている

# 最適経路探索モデル -効率的な最短経路探索のためのデータ構造

- 最小ラベルの探索におけるヒープ構造

根以外の各ノードgに対して、  
(gの親ノードのラベル)  $\leq$  (gのラベル)

※②をノードの根とする

配列順位 i	1	2	3	4	5	6	7	8	9	10	11	12	13
ラベル X(i)	2	5	10	6	9	14	11	20	15	18	13	22	16

図-8.3 ヒープ構造の配列表現

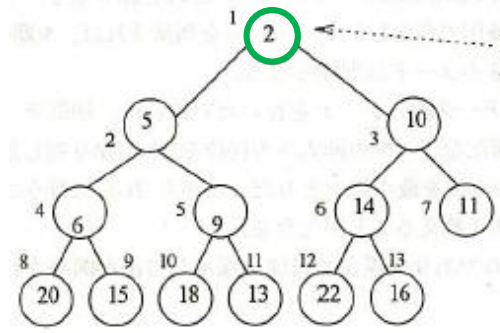


図-8.4 2分木におけるヒープ構造

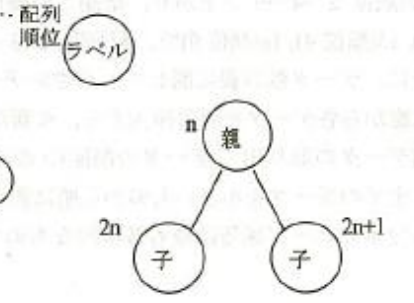


図-8.5 親子ノードと配列順位の関係

# 最適経路探索モデル -効率的な最短経路探索のためのデータ構造

- 最小ラベルの探索におけるヒープ構造

根以外の各ノードgに対して、  
(gの親ノードのラベル)  $\leq$  (gのラベル)

ex) 配列順位14(配列順位7の下)に、  
ラベル **3** を置くことを考える

配列順位	i	1	2	3	4	5	6	7	8	9	10	11	12	13
ラベル	X(i)	2	5	10	6	9	14	11	20	15	18	13	22	16

図-8.3 ヒープ構造の配列表現

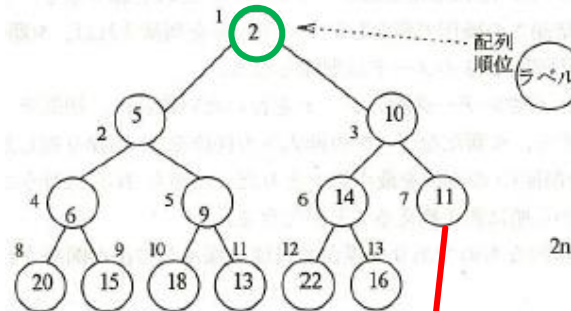


図-8.4 2分木におけるヒープ構造

図-8.5 親子ノードと配列順位の関係

**3**

# 最適経路探索モデル -効率的な最短経路探索のためのデータ構造

- 最小ラベルの探索におけるヒープ構造

根以外の各ノードgに対して、  
(gの親ノードのラベル)  $\leq$  (gのラベル)

ex) 配列順位14(配列順位7の下)に、  
ラベル **3** を置くことを考える



親ノード11と3を交替させる



親ノード10と3を交替させる

配列順位	i	1	2	3	4	5	6	7	8	9	10	11	12	13
ラベル	X(i)	2	5	10	6	9	14	11	20	15	18	13	22	16

図-8.3 ヒープ構造の配列表現

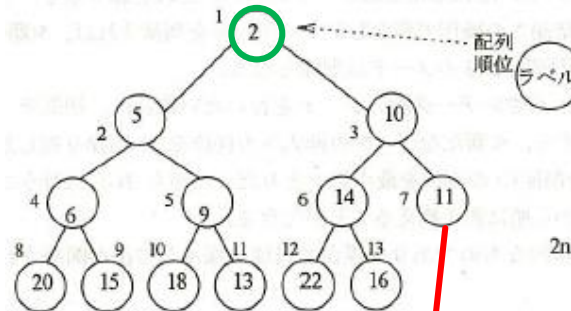


図-8.4 2分木におけるヒープ構造

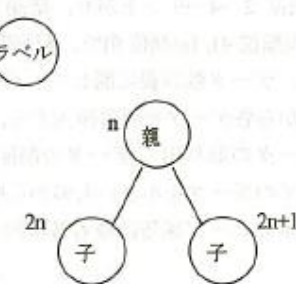


図-8.5 親子ノードと配列順位の関係

**3**

# 最適経路探索モデル-Frank-Wolfe法

- ①最適化へ向かう方向ベクトルの識別

→計算された $x^n$ に対して適切な $y$ (補助ベクトル)を探す  
 →目的関数を $y$ について線形近似する

$$\begin{aligned}
 Z_p(\mathbf{y}) &\approx Z_p(\mathbf{x}^n + \mathbf{d}) = Z_p(\mathbf{x}^n) + \nabla Z_p(\mathbf{x}^n)^T (\mathbf{y} - \mathbf{x}^n) \\
 &= Z_p(\mathbf{x}^n) + \sum_{a \in A} (\mathbf{y} - \mathbf{x}^n) \partial Z_p(\mathbf{x}^n) / \partial x_a^n \\
 &= Z_p(\mathbf{x}^n) + \sum_{a \in A} (\mathbf{y} - \mathbf{x}^n) t_a(x_a^n) \\
 &= \underbrace{Z_p(\mathbf{x}^n) - \sum_{a \in A} x_a^n t_a(x_a^n)}_{\text{定数項(計測可能)}} + \underbrace{\sum_{a \in A} y_a t_a(x_a^n)}_{\text{変数項}(y_a \text{未知})}
 \end{aligned}$$

$t_a$ ・・・リンクaのリンクコスト関数  
 $x_a$ ・・・リンクaの交通量

# 最適経路探索モデル-Frank-Wolfe法

- ①最適化へ向かう方向ベクトルの識別

前式より、

$$\min. Z_p(\mathbf{y}) = \sum_{a \in A} y_a t_a(x_a^n)$$

Subject to

$$\sum_k f_k^{rs} - Q_{rs} = 0 \quad \forall rs \in \Omega \quad y_a = \sum_k \sum_{rs} \delta_{a,k}^{rs} f_k^{rs} \quad \forall a \in A$$

という補助問題を得る。

ここで上式は、 $\{t_a(x_a^n)\}$ の下で総走行時間を最小にする $y$ を求めることである。 $\{t_a(x_a^n)\}$ は定数であるから、 $\{t_a(x_a^n)\}$ のリンク所要時間で求められる最短経路にすべてのOD交通量を流すという**all-or-nothing配分**をすれば、 $Z_p(\mathbf{y})$ を最小にするような $y$ が得られる。

最小交通費用をもつ経路へ  
すべての需要を配分するもの

# 最適経路探索モデル-Frank-Wolfe法

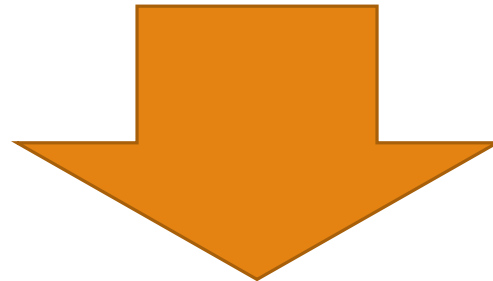
---

- ①最適化へ向かう方向ベクトルの識別

$Z_p(\mathbf{y}) \approx Z_p(\mathbf{x}^n + \mathbf{d}) = Z_p(\mathbf{x}^n) + \nabla Z_p(\mathbf{x}^n)^T (\mathbf{y} - \mathbf{x}^n)$ が最小

→ $\nabla Z_p(\mathbf{x}^n)^T (\mathbf{y} - \mathbf{x}^n)$ が最も小さい負の値を与える

→ $\mathbf{d} = \mathbf{y} - \mathbf{x}^n$  は目的関数 $Z_p$ の点 $\mathbf{x}^n$ におけるもっとも急な下り勾配に沿った方向ベクトルである



$\mathbf{d} = \mathbf{y} - \mathbf{x}^n$ を最小化へ向かう降下方向ベクトルとして利用

# 最適経路探索モデル-Frank-Wolfe法

---

- ② : ①で求めた方向に、どれだけ進むか？

降下方向ベクトルが与えられた後は、

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \alpha(\boldsymbol{y} - \boldsymbol{x}^{(n)}) \Leftrightarrow \boldsymbol{x}^{(n+1)} = \alpha\boldsymbol{y} + (1 - \alpha)\boldsymbol{x}^{(n)}$$

なる $\boldsymbol{x}^{(n+1)}$ を $Z_p$ に代入し、 $\alpha$ による一次元最適化を行う( $\alpha$ による一次方程式を解く)ことによって、次の施行点 $\boldsymbol{x}^{(n+1)}$ を求める。その点に対しても新たな $\boldsymbol{y}$ ,  $d$ が求まり…という作業を、収束判定を満たすまで繰り返し行う。

( $n=1$ では、 $t_a(0)$ の状態でのall-or-nothing配分を $\boldsymbol{x}^{(n+1)}$ とする)



# 最適経路探索モデル

## -リンク容量制約を持つ場合のFrank-Wolfe法

---

$$x_a \leq C_a \quad (C_a: \text{交通容量}) \quad \forall a \in A \text{ を定める}$$

Frank-Wolfe法はリンクコストが無限大に向かうため、計算量が膨大である。ここで、ステップサイズ $\alpha$ を以下のように制約すればよい。

$$\alpha \leq \alpha_{max} = \min_{x_a < y_a} \{(C_a - x_a^{(n)}) / (y_a - x_a^{(n)})\}$$

# 最適経路選択モデル

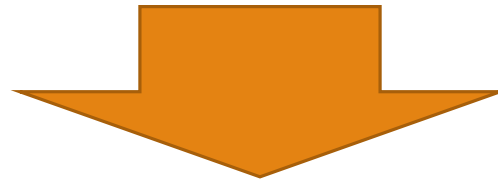
## -利用者均衡配分のより効率的な解法

---

Flank-Wolfe法の欠点を改善した様々な効率的解法について述べる

- 打ち切り二次計画法
- Simplicial Decomposition法
- Flank-Wolfe法には、解の近傍に近づいているものの、収束までに極めて多数の繰り返し計算回数を要するという欠点がある

→探索方向ベクトル  $y - x^{(n)}$  が、解の近傍では目的関数の勾配方向とほぼ垂直な方向となってしまう



$z_p$ を二次近似して得られる補助問題を立て、解く際はコストが増えないよう厳密には解かないようにする

# 最適経路選択モデル

## -利用者均衡配分のより効率的な解法

Flank-Wolfe法の欠点を改善した様々な効率的解法について述べる

- 打ち切り二次計画法
- Simplicial Decomposition法
- 以下の補助問題を解く

$$\min_y Z(\mathbf{y}) = \nabla Z(\mathbf{x}^{(n)})^T (\mathbf{y} - \mathbf{x}^{(n)}) + \frac{1}{2} (\mathbf{y} - \mathbf{x}^{(n)})^T \nabla^2 Z(\mathbf{x}^{(n)}) (\mathbf{y} - \mathbf{x}^{(n)})$$

Subject to

$$y_a = \sum_k \sum_{rs} \delta_{a,k}^{rs} f_k^{rs} \quad \forall a \in A \quad \sum_{k \in K_n} g_k^{rs} - q_n = 0 \quad \forall rs \in W$$
$$g_k^n \geq 0 \quad \forall k \in K_n, \quad \forall rs \in W$$

# 最適経路選択モデル

## -利用者均衡配分のより効率的な解法

Flank-Wolfe法の欠点を改善した様々な効率的解法について述べる

- 打ち切り二次計画法
- Simplicial Decomposition法

$Z_p$ を二次近似して得られる補助問題を立て、解く際はコストが増えないよう厳密には解かないようにする

- 代わりに以下の補助問題を解く

$$\begin{aligned} \min_{\mathbf{y}} Z_{QP}(\mathbf{y}) &= \sum_{a \in A} t_a(x_a^{(n)}) \cdot (\mathbf{y} - \mathbf{x}^{(n)}) + \frac{1}{2} \sum_{a \in A} m_a(x_a^{(n)}) \cdot (\mathbf{y} - \mathbf{x}^{(n)})^2 \\ \text{Subject to} \\ y_a &= \sum_k \sum_{rs} \delta_{a,k}^{rs} f_k^{rs} \quad \forall a \in A, \quad \sum_{k \in K_n} g_k^{rs} - q_n = 0 \quad \forall rs \in W \\ g_k^n &\geq 0 \quad \forall k \in K_n, \quad \forall rs \in W \end{aligned}$$

初めは  
 $m = 0, \mathbf{y}^{(0)} = \mathbf{x}^{(n)}$   
とする

# 最適経路選択モデル

## -利用者均衡配分のより効率的な解法

---

Flank-Wolfe法の欠点を改善した様々な効率的解法について述べる

- 打ち切り二次計画法
- Simplicial Decomposition法
- リンクコスト関数はリンク毎に分離可能

$$\nabla Z(x^{(n)}) = \left[ \dots, t_a(x_a^{(n)}), \dots \right]^T$$

- Hessian行列は、

初めは  
 $m = 0, \mathbf{y}^{(0)} = \mathbf{x}^{(n)}$   
とする

# 最適経路選択モデル

## -利用者均衡配分のより効率的な解法

---

Flank-Wolfe法の欠点を改善した様々な効率的解法について述べる

- 打ち切り二次計画法
- Simplicial Decomposition法
  - 重み係数 $\lambda$ を未知変数として、許容領域内で解く方法
  - all-or-nothing配分パターンが $e_i$ が所与であれば、需要固定型利用者均衡配分は、 $\lambda$ の最適化問題として表現できる

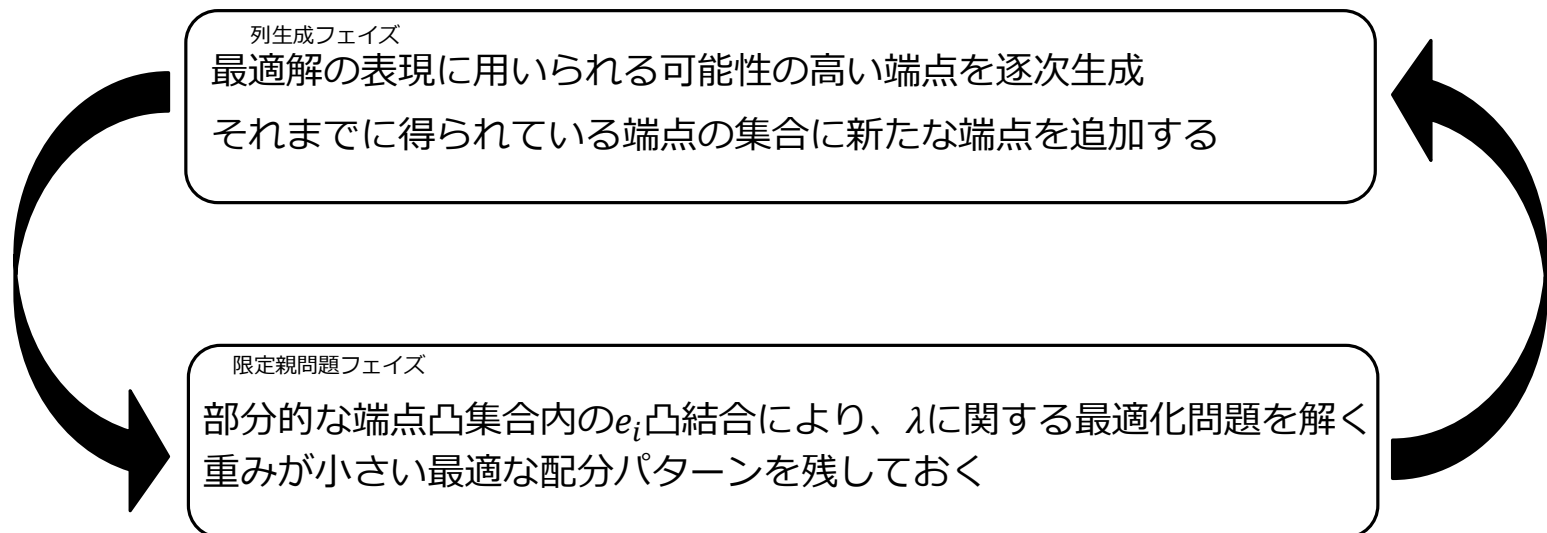
$$\min_{\lambda} \mathbf{Z}(x) \quad \text{subject to } x = \sum_{i=1}^M \lambda_i \mathbf{e}_i, \quad \sum_{i=1}^M \lambda_i = 1, \quad \lambda_i \geq 0 (i = 1, 2, \dots, M)$$

# 最適経路選択モデル

## -利用者均衡配分のより効率的な解法

Flank-Wolfe法の欠点を改善した様々な効率的解法について述べる

- 打ち切り二次計画法
- Simplicial Decomposition法
- 重み係数 $\lambda$ を未知変数として、許容領域内で解く方法
- all-or-nothing配分パターンが $e_i$ が所与であれば、需要固定型利用者均衡配分は、 $\lambda$ の最適化問題として表現できる



# 最適経路選択モデル

## -利用者均衡配分のより効率的な解法

Flank-Wolfe法の欠点を改善した様々な効率的解法について述べる

- 打ち切り二次計画法
  - Simplicial Decomposition法
- 具体的な手法(RSD法を例に)

初期設定

〈端点生成削除フェイズ〉

$$\min_y \sum_{a \in A} t_a(x_a^{(m)}) y_a \quad s.t. \ y \in \Omega_P(\text{フロー保存条件を満たしたリンク交通量パターン})$$

最短経路配分問題を解く

$$t(x^{(m)}) \cdot (y - x^{(m)}) \geq 0$$

$$t(x^{(m)}) \cdot (y - x^{(m)}) < 0$$

均衡解 :  $x^{(m)}$

Yes  
 $E^{(m)} < M$ に含まれる端点の個数が  $M$  未満

No

$$E^{(m+1)} := E^{(m)} \cup \{y\}, \\ F^{(m+1)} := F^{(m)}$$

$$F^{(m+1)} := \{x^{(m)}\}$$

$$\hat{E}^{(m+1)} := E^{(m)} \cup F^{(m+1)}$$

〈限定親問題フェイズ〉へ



# 最適経路選択モデル

## -リンク間に相互干渉のある利用者均衡配分の解法

- いずれも5.5節で紹介した変分方程式の解法を応用する

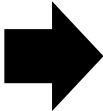
- 打ち切り二次計画法
- Simplicial Decomposition法

$$\begin{aligned} & \min_y \lambda Z(x^{(m+1)}) \\ & \text{subject to } x^{(m+1)} = \sum_{i=1}^{\hat{M}} \lambda_i e_i, \sum_{i=1}^{\hat{M}} \lambda_i = 1, \lambda_i \geq 0 \end{aligned}$$

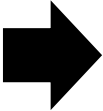
※ $\hat{M}$ は $\hat{E}^{(m+1)}$ の要素数

〈限定親問題フェイズ〉

需要固定型利用者均衡配分におけるリンク交通量配分パターン  $x^{(m+1)}$  を  $e_i \in \hat{E}^{(m+1)}$  の凸集合により表現した以下の咲いてか問題を  $\lambda$  について解く



$\lambda_i = 0$  となっているすべての  $e_i$  を  $E^{(m+1)}$  及び  $F^{(m+1)}$  から削除



$m = m + 1$  として  
〈端点生成削除フェイズ〉へ

# 最適経路選択モデル

## -リンク間に相互干渉のある利用者均衡配分の解法

---

①緩和法(非線型対角化法)

②射影法(線形近似法)

③Gap関数を利用した解法

- いずれも変分不等式問題を応用している  
凸集合 $\Omega$ と写像 $F:\Omega\rightarrow\Omega$ を所与とするとき  
以下のように定義される問題のことをいう

$$\text{Find } x \in \Omega \quad \text{such that } F(x) \cdot (y - x) \geq 0$$

# 最適経路選択モデル

## -リンク間に相互干渉のある利用者均衡配分の解法

---

### ①緩和法(非線型対角化法)

### ②射影法(線形近似法)

### ③Gap関数を利用した解法

- 各繰返し計算において、リンクコスト写像  $t(x)$  の相互干渉のある部分を固定して得られる「緩和問題」を解くことにより繰り返すアルゴリズム

- リンクコスト関数  $t(x)$  の相互干渉のある部分を固定して得られる緩和問題を解くことを繰り返す
- m回目繰返しにおけるリンク交通量パターン  $x^{(m)}$  が既知の場合、以下のような近似リンクコスト写像として考える：

$$\hat{t}_a(x, x^{(m)}) = t_a(x_1^{(m)}, \dots, x_a, \dots, x_L^{(m)}) \quad \forall a \in A$$

- 以下のようなリンク間に相互干渉のないコスト関数のもとでの利用者均衡配分と見なすことができる

$$\min. Z(x) = \sum_a \int_0^{x_a} t_a(x_1^{(m)}, \dots, x_a, \dots, x_L^{(m)}) = t_a(x^*) \quad \text{subject to } x \in \Omega_P$$

# 最適経路選択モデル

## -リンク間に相互干渉のある利用者均衡配分の解法

---

### ①緩和法(非線型対角化法)

### ②射影法(線形近似法)

### ③Gap関数を利用した解法

- 具体的な解法

Step0(初期許容解を求める) 繰返し回数  $m := 1$ ; 初期許容解(リンク交通量パターン) $x^{(1)}$ を想定

Step1(緩和問題を解く) Frank-Wolfe法のアルゴリズムで解き、その解を $x^{(m+1)}$ とする

Step2(収束判定)  $\|x^{(m+1)} - x^{(m)}\| \leq \varepsilon$ なら終了、そうでなければ $m = m + 1$ とし、Step1へ

- 各繰返し計算において、リンクコスト写像  $t(x)$ の相互干渉のある部分を固定して得られる「緩和問題」を解くことにより繰返すアルゴリズム

- 繰返し計算を4回未満で打ち切るなど、計算効率を良くするためのアルゴリズムが提案されている

# 最適経路選択モデル

## -リンク間に相互干渉のある利用者均衡配分の解法

---

### ①緩和法(非線型対角化法)

- もとのリンクコスト写像を線形近似して得られる補助的な変分不等式問題を解くことを繰り返すアルゴリズム

### ②射影法(線形近似法)

### ③Gap関数を利用した解法

- m回目繰返しにおけるリンク交通量パターン $\mathbf{x}^{(m)}$ が既知の場合、以下のような近似リンクコスト写像として考える：

$$\hat{t}(\mathbf{x}) = \mathbf{t}(\mathbf{x}^{(m)}) + \frac{1}{\rho} Q(\mathbf{x} - \mathbf{x}^{(m)})$$

Q: 適当な正定値行列,  
 $\rho$ : 十分小さな正値のパラメータ

- 近似的なリンクコスト写像による以下の補助的な変分方程式を繰り返す

$$\text{Find } \mathbf{x}^* \in \Omega_p \quad \text{such that } \hat{\mathbf{t}}(\mathbf{x}) \cdot (\mathbf{x} - \mathbf{x}^*) \geq \mathbf{0} \quad \text{subject to } \mathbf{x} \in \Omega_p$$

- 5章の射影の概念より、

$$\mathbf{x}^* = \text{Proj}_{\Omega, Q} \left( \mathbf{x}^{(m)} - \rho Q^{-1} \mathbf{t}(\mathbf{x}^{(m)}) \right)$$

# 最適経路選択モデル

## -リンク間に相互干渉のある利用者均衡配分の解法

---

### ①緩和法(非線型対角化法)

- もとのリンクコスト写像を線形近似して得られる補助的な変分不等式問題を解くことを繰り返すアルゴリズム

### ②射影法(線形近似法)

### ③Gap関数を利用した解法

- 具体的な解法

Step0(初期許容解を求める) 繰り返し回数  $m := 1$ ; 定数  $\rho, Q$ , 初期許容解(リンク交通量パターン)  $x^{(1)}$  を想定

Step1(緩和問題を解く) Frank-Wolfe法のアルゴリズムで解き、その解を  $x^{(m+1)}$  とする

Step2(収束判定)  $\|x^{(m+1)} - x^{(m)}\| \leq \varepsilon$  なら終了、そうでなければ  $m = m + 1$  とし、Step1へ

- アルゴリズムが大域的に収束するための十分条件は  $\rho$  が十分に小さく設定されて、なおかつ均衡解への収束が保証される

# 最適経路選択モデル

## -リンク間に相互干渉のある利用者均衡配分の解法

---

①緩和法(非線型対角化法)

②射影法(線形近似法)

- ①②の解法よりも緩い条件下で大域的収束が保証された効率的な解法 merit関数と射影演算を用いたものを説明する

③Gap関数を利用した解法

- 任意の凸集合 $\Omega$ 上で定義された一般的な偏微分方程式は

$$\text{Find } x^* \in \Omega \quad \text{such that } F(x^*) \cdot (x - x^*) \geq 0 \quad \forall x \in \Omega$$

- 上式のアルゴリズムの基本的な枠組みは、学習したような許容降下法と同様であり、以下の改訂式によって $\{x^{(n)}\}$ を生成することを繰り返す

$$x^{(n+1)} := x^{(n)} + \alpha^{(n)} d^{(n)} \quad n = 0, 1, 2, \dots$$

- Gap関数は、以下のように定義される

$$G_Q(x) := F(x^*) \cdot (x - H(x)) - \frac{1}{2} (H(x) - x) \cdot Q(H(x) - x)$$

- ただし、

$$H(x) := Proj_{\Omega, Q} \left( x - Q^{-1}(F(x)) \right), \quad d^{(n)} = H(x^{(n)}) - x^{(n)}$$

# 最適経路選択モデル

## -リンク間に相互干渉のある利用者均衡配分の解法

---

①緩和法(非線型対角化法)

②射影法(線形近似法)

③Gap関数を利用した解法

- ①②の解法よりも緩い条件下で大域的収束が保証された効率的な解法 merit関数と射影演算を用いたものを説明する

merit関数に対する一次元探索により $\alpha^{(n)} \in [0,1]$ を決定する

$$\min_{\alpha^{(n)}} G_Q(\mathbf{X}^{(n)} + \alpha^{(n)} \mathbf{d}^{(n)}) \quad \text{s.t.} \quad 0 \leq \alpha^{(n)} \leq 1$$

起点別リンク交通量 $\{x_{ij}^r\}$ を明示的未知変数とした等価変分不等式問題を解くことを考える  
<フロー保存条件>

$$\sum_i x_{ik}^r - \sum_j x_{kj}^r - q_{rk} = 0 \quad \forall k \in N, k \neq r, \forall r \in R$$



# 最適経路選択モデル

## -リンク間に相互干渉のある利用者均衡配分の解法

①緩和法(非線型対角化法)

②射影法(線形近似法)

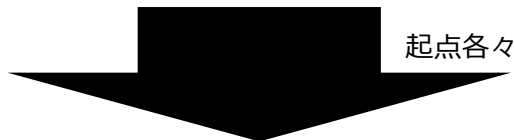
③Gap関数を利用した解法

- ①②の解法よりも緩い条件下で大域的収束が保証された効率的な解法 merit関数と射影演算を用いたものを説明する

<相補性条件>

$$t_{ij}(\mathbf{x}) + \tau_i^r = \tau_j^r$$

$$\begin{cases} x_{ij}^r \cdot \{t_{ij}(\mathbf{x}) + \tau_i^r - \tau_j^r\} = 0 \\ t_{ij}(\mathbf{x}) + \tau_i^r - \tau_j^r \geq 0 \\ x_{ij}^r \geq 0 \end{cases} \quad \forall ij \in A, \forall r \in R$$



起点各々の未知変数ベクトルをまとめる

$$\hat{\mathbf{x}} \equiv (\dots, \mathbf{x}^r, \dots)^T \in \prod_{r \in R} R^L, \quad \boldsymbol{\tau}^r \equiv (\dots, \boldsymbol{\tau}^r, \dots)^T \in \prod_{r \in R} R^N, \quad \hat{\mathbf{q}} \equiv (\dots, \mathbf{q}^r, \dots)^T$$

$\hat{A}$  : 起点数個のAを要素に持つブロック対角行列  
 $E$  : 起点数個のL\*L単位行列を横に並べた行列

# 最適経路選択モデル

## -リンク間に相互干渉のある利用者均衡配分の解法

①緩和法(非線型対角化法)

②射影法(線形近似法)

③Gap関数を利用した解法

写像  $F: \Omega \rightarrow \Omega$  を

$$F(X) \equiv \begin{bmatrix} E^T t(E\hat{x}) \\ \hat{q} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \hat{A}^T \\ -\hat{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{t} \end{bmatrix}$$

$$X \equiv [\hat{x} \quad \hat{t}]^T \in \Omega, \quad \Omega \equiv \prod_{r \in R} R_+^L \times R_+^N$$

$$\text{Find } X^* \in \Omega \quad \text{such that } F(X^*) \cdot X = \mathbf{0}, F(X^*) \geq \mathbf{0}, X^* \geq \mathbf{0}$$

以下の変分不等式問題と等価である

$$\text{Find } X^* \in \Omega \quad \text{such that } F(X^*) \cdot (X - X^*) \geq \mathbf{0}$$

許容領域  $\Omega$  が非負実数空間であるから、 $H(X) \equiv [X - Q^{-1}F(X)]$  に帰着

# まとめ

---

- 利用者均衡モデルの解法の確立

変数の少ない単純なモデルではクーン・タッカー条件から直接導き出せるが、大規模な問題となると大変である。

- Flank-Wolfe法は、リンク容量に制約がある場合、計算回数が膨大になったり、リンク相互干渉などの問題が生じてしまう。

- Flank-Wolfe法を改善し効率化を図ることは、より大きな枠組、あるいはより現実的な場合を考慮する上で重要なことである。