

# 交通ネットワーク配分

---

スタートアップゼミ #2

2024年4月11日

増田 慧樹

# 目次

## 1 均衡配分

- All-or-nothing配分
- 利用者均衡配分の定式化
- 利用者均衡配分の解法
- Braess's Paradox
- ゲームのお時間
- ポテンシャル・ゲーム
- システム最適配分と混雑課金

## 2 確率的利用者均衡モデル

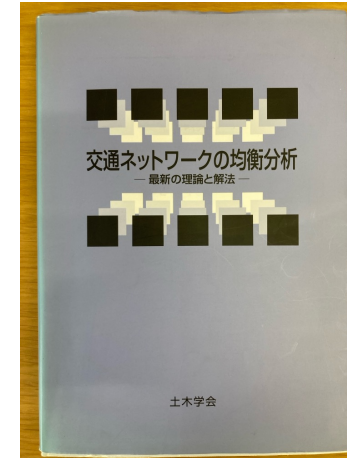
- SUEの定式化
- SUEの解法

## 3 その他の配分モデル

- マイクロシミュレーション
- ゾーンベースシミュレーション

	非混雑型 flow-independent	混雑型 flow-dependent
確定的経路 選択	<b>all-or-nothing配分</b>	<b>利用者均衡配分(UE)</b>
確率的経路 選択	<b>確率的配分</b>	<b>確率的利用者均衡配分 (SUE)</b>

- 土木学会. (1998). 交通ネットワークの均衡分析-最新の理論と解法.  
通称”青本”. 図書館や研究室の本棚にあります.  
羽藤研wikiや[福田研](#)の過去のゼミ資料も参考になります.



- Sheffi, Y. (1985). Urban transportation networks (Vol. 6). Prentice-Hall, Englewood Cliffs, NJ.  
著者が無償公開: <https://sheffi.mit.edu/book/urban-transportation-networks>  
2013年度Sheffiゼミ: <http://bin.t.u-tokyo.ac.jp/sheffi2013/>

# 目次

## 1 均衡配分

- All-or-nothing配分
- 利用者均衡配分の定式化
- 利用者均衡配分の解法
- Braess's Paradox
- ゲームのお時間
- ポテンシャル・ゲーム
- システム最適配分と混雑課金

## 2 確率的利用者均衡モデル

- SUEの定式化
- SUEの解法

## 3 その他の配分モデル

- マイクロシミュレーション
- ゾーンベースシミュレーション

	非混雑型 flow-independent	混雑型 flow-dependent
確定的経路 選択	<b>all-or-nothing配分</b>	<b>利用者均衡配分(UE)</b>
確率的経路 選択	<b>確率的配分</b>	<b>確率的利用者均衡配分 (SUE)</b>

# 交通量配分とは

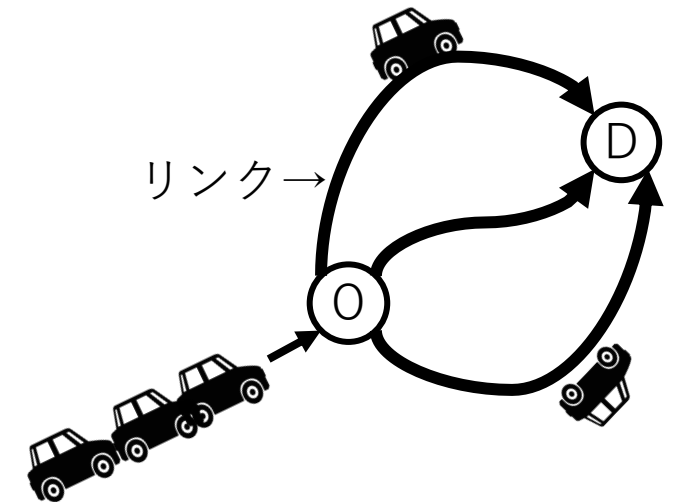
交通量配分問題 (traffic assignment problem) とは,  
交通ネットワークを対象に, 需要OD交通量と配分原則を与件として,  
ネットワークの各リンクを流れる交通量を予測する問題

交通ネットワーク: 交通網のグラフによるネットワーク表現

需要OD交通量: 各ODペア間の交通量

OD: Origin (出発地)-Destination (到着地)のペア

配分原則: 利用者の行動原理の仮定



# 四段階推定法の中での位置付け

四段階推定法 … 交通計画の立案に必要な将来交通需要の古典的予測手法。

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

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

ココ！

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

鉄道の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

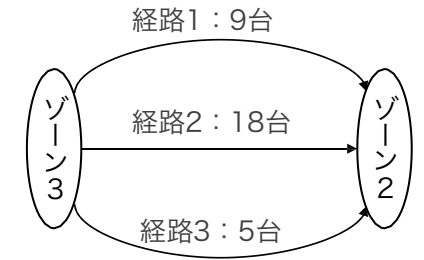
どの交通手段で

どこからどこへ

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

どの経路を通して

ゾーン	分布			発生	分担		配分	
	1	2	3		$T_{ijm}$	$T_{ijmr}$		
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				



# 配分原則

では、どのような配分原則（行動原理の仮定）に基づいて、OD交通量を各経路に配分するか

## ■ Wardropの第1原則

仮定：

- ①全ての利用者は常に**旅行時間を最小とするよう**に行動する
- ②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

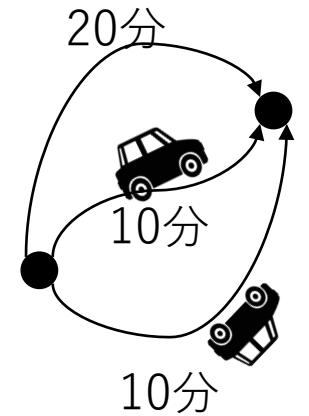
上の仮定のもと、利用者が自己の経路選択を最適化した結果到達する均衡状態が、

### Wardropの第1原則

**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。**

= **利用者均衡配分（UE: User Equilibrium assignment）**

均衡状態においては、もはやどの利用者也経路を変更することによって自己の旅行時間をそれ以上短縮することはできない（Nash均衡）



均衡状態の例

# いろいろな配分原則

## ■ Wardropの第2原則

道路網上の総旅行時間が最小となる。

=システム最適化配分 (SO: System Optimum assignment)

交通を中央集権的に管理することによって達成できる

## ■ 確率的利用者均衡(SUE: Stochastic User Equilibrium assignment)

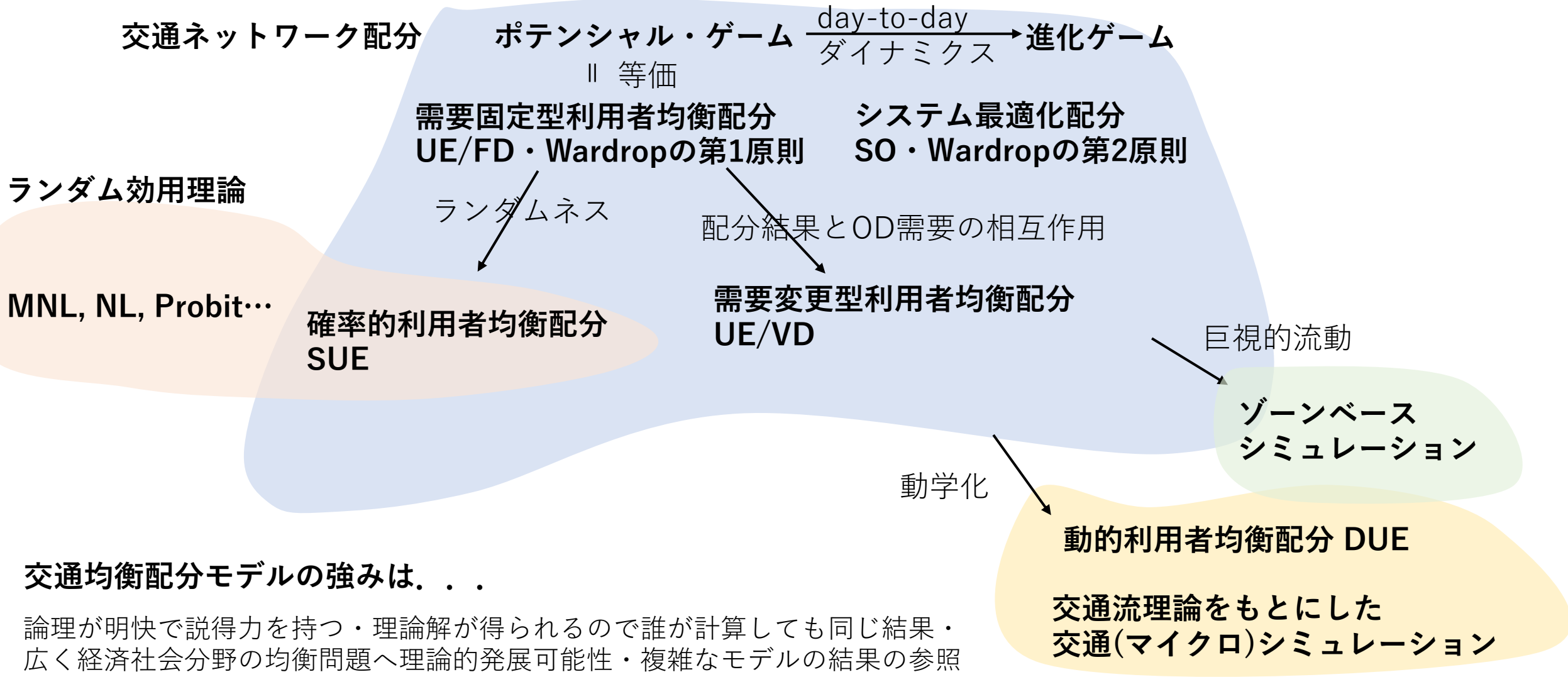
どの利用者も経路を変更することによって自己の旅行時間をそれ以上短縮することができないと 信じている 均衡状態。 Daganzo & Sheffi (1977), Fisk(1980)

## ■ 動的利用者均衡(DUE: Dynamic User Equilibrium assignment)

任意の時刻に出発した車が、目的地に到達するまでの経路の旅行時間に対して、常にWardrop均衡が成立する。 Ran et al. (1996)



# マッピング



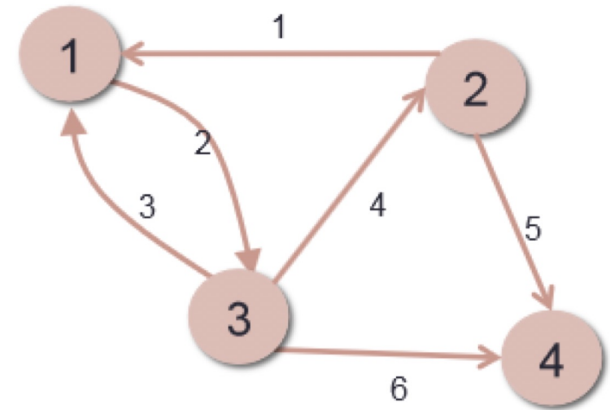
## 交通均衡配分モデルの強みは...

論理が明快で説得力を持つ・理論解が得られるので誰が計算しても同じ結果・  
広く経済社会分野の均衡問題へ理論的発展可能性・複雑なモデルの結果の参照  
点になる・余剰分析などミクロ経済の理論と結びついている

# 準備

# 交通ネットワークの記述

- ネットワークは、**ノード（点）**と**リンク（線）**の集合としてグラフ表現できる。  
ノード集合:  $N = \{1,2,3,4\}$   
リンク集合:  $A = \{1,2,3,4,5,6\}$
- 交通需要の発生ノード集合を $R$ 、集中ノード集合を $S$ とかき、 $r \in R, s \in S$ を**セントロイド (centroid)**という。
- **パス**: ODペア $rs$ 間を結ぶ経路  
一般に各ODペアに対して複数のパスが存在する



# リンクパフォーマンス関数

各リンクの旅行時間・走行費用・安全性・快適性など（抵抗 $t_a$ という）は、そのリンクの交通量 $x_a$ に依存して変化する。その関係を表したものを**リンクパフォーマンス関数**という。

$$t_a = t_a(x_a)$$

## 【用語】

### flow dependent

リンクコスト $t_a$ がリンク交通量（flow） $x_a$ に依存する場合  
例) 一般的な道路上の自動車交通



### flow independent

リンクコスト $t_a$ がリンク交通量（flow） $x_a$ に依存せず一定の場合  
 $t_a = t_a(x_a) = (\text{定数})$   
例) 鉄道路線網や混雑が酷くないパス

# リンクパフォーマンス関数

## ■ BPR関数

- 米国道路局（US Bureau of Public Road）で開発されたリンクパフォーマンス関数
- 交通均衡配分問題によく用いられる

$$t_a(x_a) = t_{a0} \left\{ 1 + \alpha \left( \frac{x_a}{C_a} \right)^\beta \right\}$$

$t_a$ : リンク $a$ の旅行時間

$x_a$ : リンク $a$ の交通量(台/時)

$t_{a0}$ : リンク $a$ の自由旅行時間 = 交通量0の時の旅行時間

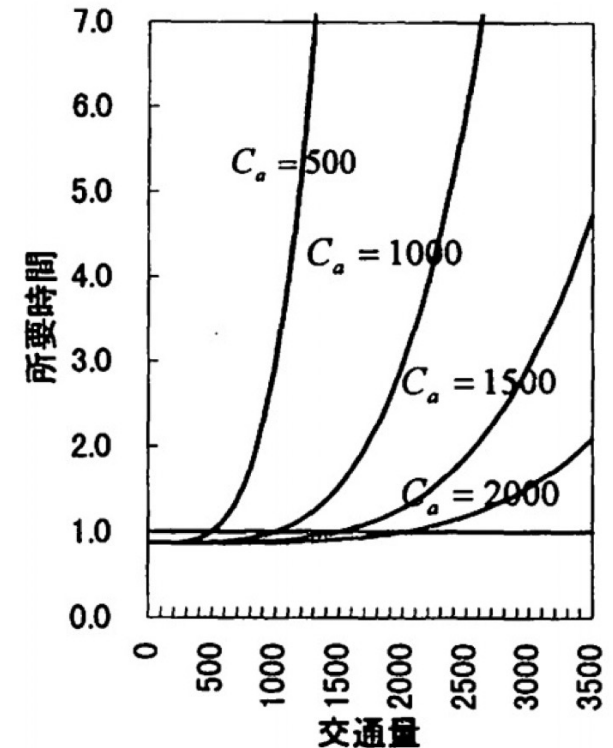
$C_a$ : リンク $a$ の交通容量 (台/時)

$\alpha, \beta$ : パラメータ

特徴:

- 交通量 $x_a$ に関して狭義の単調増加関数
- 交通量 $x_a$ の全ての領域で定義される

→後述する利用者均衡配分モデルが一意的な解を持ち、Frank-Wolfe法を適用するための条件を満たす（詳しくは青本2章を参照）



BPR型リンクパフォーマンス関数  
交通ネットワークの均衡分析（土木学会,1998）

# 簡単な配分をやってみよう

	非混雑型 flow-independent	混雑型 flow-dependent
確定的経路 選択	<b>all-or-nothing配分</b>	<b>利用者均衡配分(UE)</b>
確率的経路 選択	<b>確率的配分</b>	<b>確率的利用者均衡配分 (SUE)</b>

# all-or-nothing配分 – 単純な方法

## ■ all-or-nothing配分

- flow-independentな場合の配分アルゴリズム
- **最短経路探索** → **最小費用経路に全需要を配分**を全ての起点に対して実行（簡単！）

Step 1  $n = 0$ とする。全てのリンク交通量について $x_{ij}^{(0)} = 0$ とする。

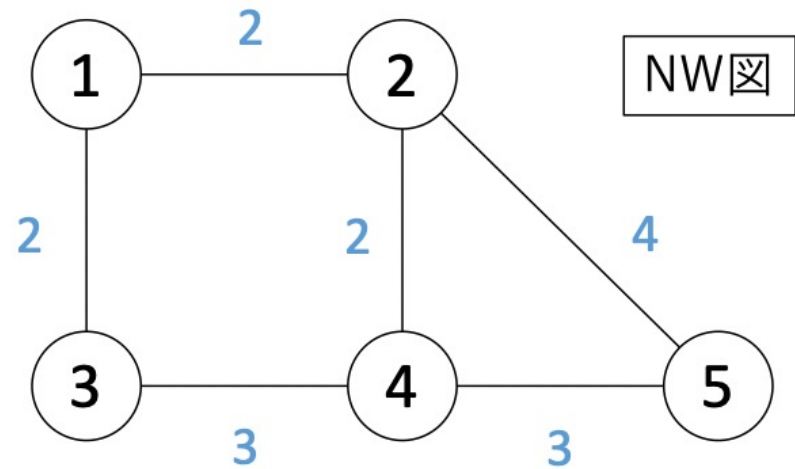
Step 2  $n$ 番目の起点を $o$ とする。起点 $o$ から他の全てのノードへ最短経路探索を行い、起点 $o$ から他の全てのノードへの最短経路費用 $\{C_{\min}[o \rightarrow i]\}$ と各ノードに対する先行ポインタ $F_i$ を求める。 $y_{ij} = 0$ とする。

Step 3 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。

Step 4  $n = N$ （ノード数）なら終了。そうでなければ $n = n + 1$ としてStep 2へ。

# 👉 all-or-nothing配分 – 単純な方法

OD交通量を全ての経路に流して、最後にリンクごとに集計する方法



OD表

値はリンクコストに対応

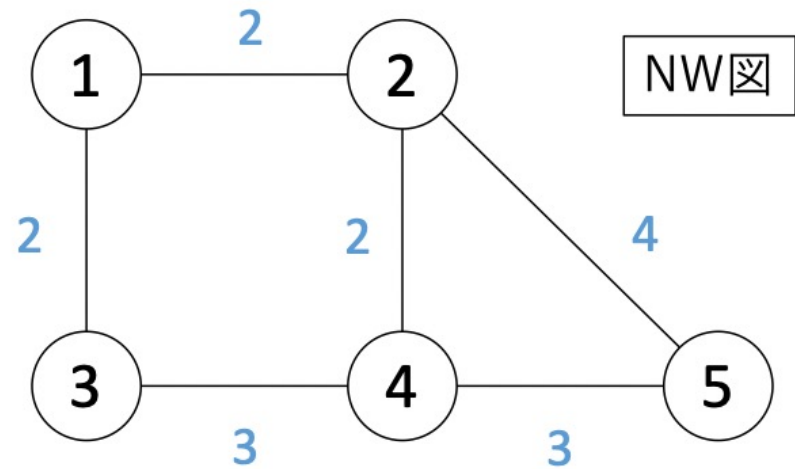
O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

リンク	流量
1-2	0
1-3	0
2-4	0
3-4	0
2-5	0
4-5	0
2-1	0
3-1	0
4-2	0
4-3	0
5-2	0
5-4	0



# 👉 all-or-nothing配分 – 単純な方法

OD交通量を全ての経路に流して、最後にリンクごとに集計する方法

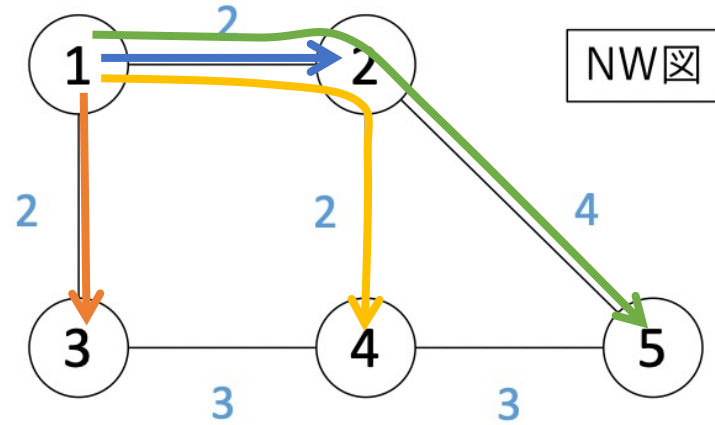


OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

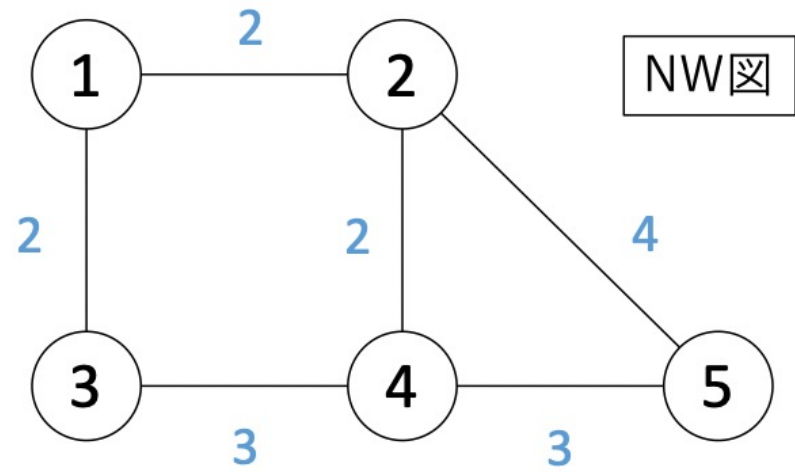
起点1に対する計算例



リンク	流量
1-2	4+2+3=9
1-3	5
2-4	2
3-4	0
2-5	3
4-5	0
2-1	0
3-1	0
4-2	0
4-3	0
5-2	0
5-4	0

# 👉 all-or-nothing配分 – 単純な方法

OD交通量を全ての経路に流して、最後にリンクごとに集計する方法

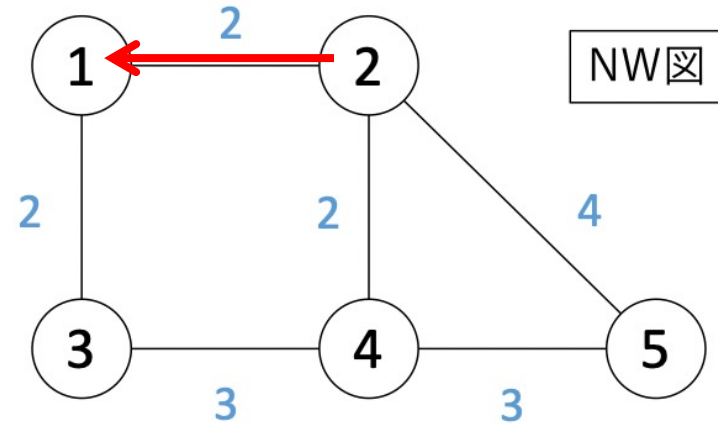


OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

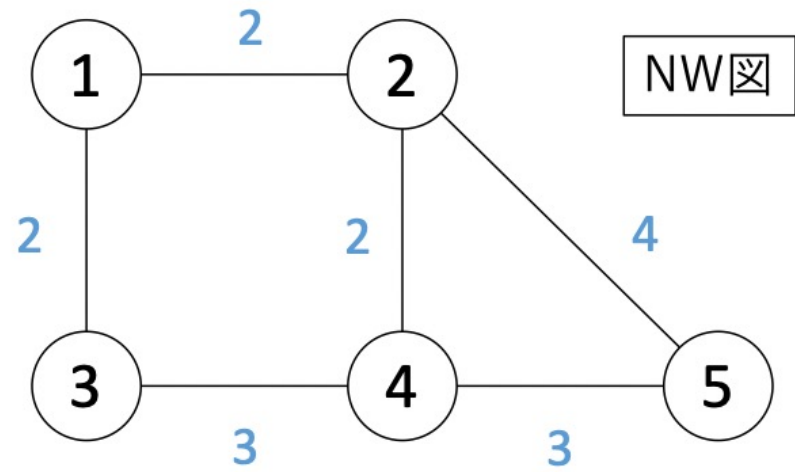
起点2に対する計算例



リンク	流量
1-2	9
1-3	5
2-4	2
3-4	0
2-5	3
4-5	0
2-1	0 + 3
3-1	0
4-2	0
4-3	0
5-2	0
5-4	0

# 👉 all-or-nothing配分 – 単純な方法

OD交通量を全ての経路に流して、最後にリンクごとに集計する方法

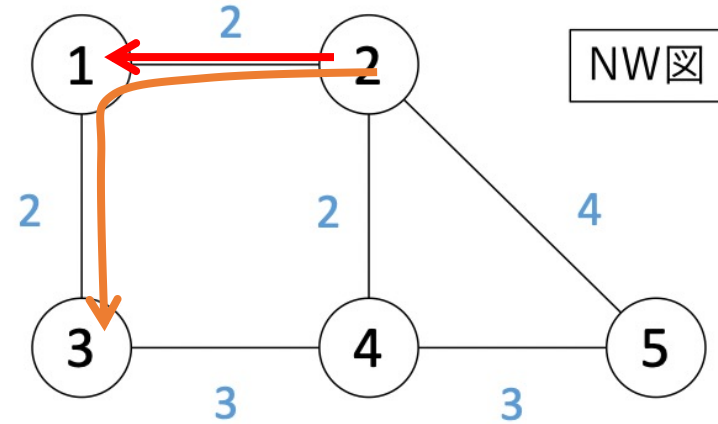


OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

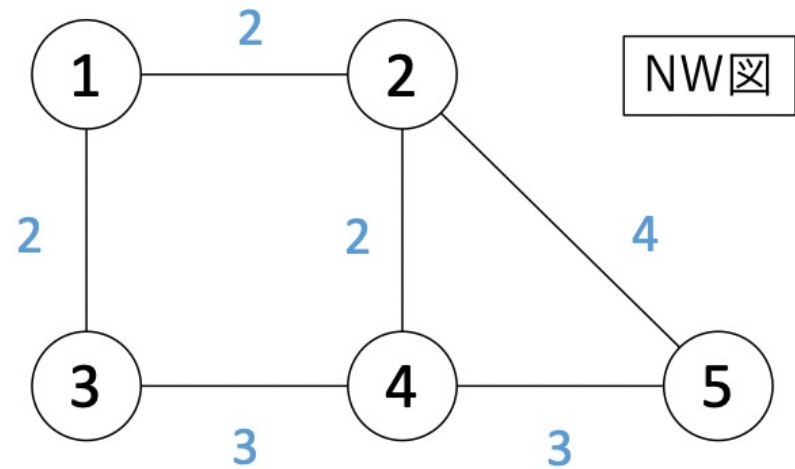
起点2に対する計算例



リンク	流量
1-2	9
1-3	5 + 4
2-4	2
3-4	0
2-5	3
4-5	0
2-1	0 + 3 + 4
3-1	0
4-2	0
4-3	0
5-2	0
5-4	0

# 👉 all-or-nothing配分 – 単純な方法

OD交通量を全ての経路に流して、最後にリンクごとに集計する方法

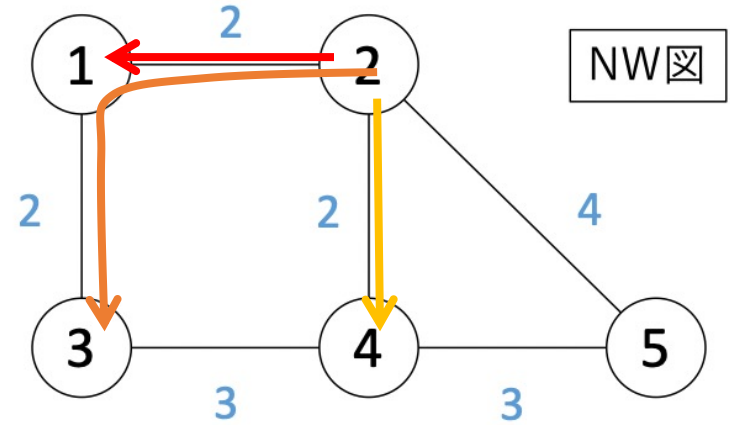


OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

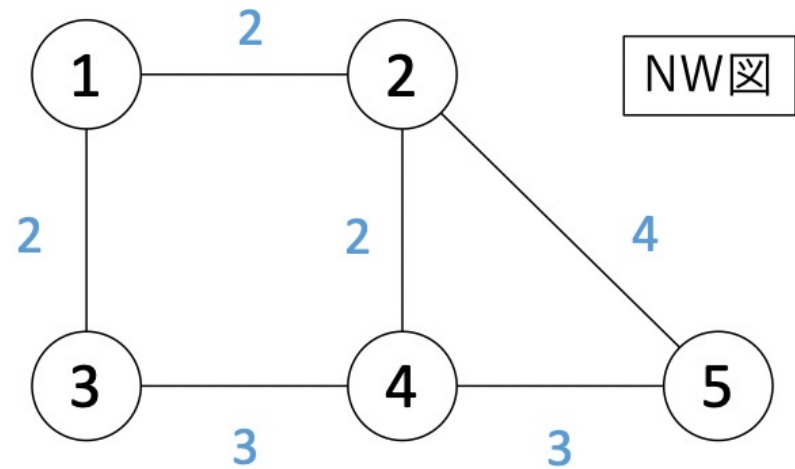
起点2に対する計算例



リンク	流量
1-2	9
1-3	5+4
2-4	2+1
3-4	0
2-5	3
4-5	0
2-1	0+3+4
3-1	0
4-2	0
4-3	0
5-2	0
5-4	0

# 👉 all-or-nothing配分 – 単純な方法

OD交通量を全ての経路に流して、最後にリンクごとに集計する方法



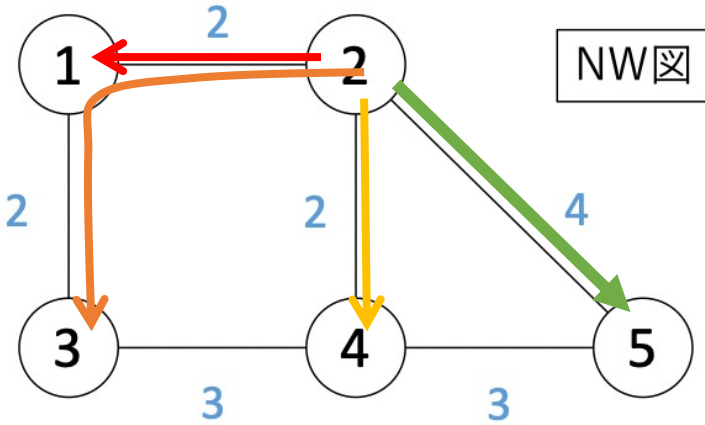
NW☒

OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

起点2に対する計算例

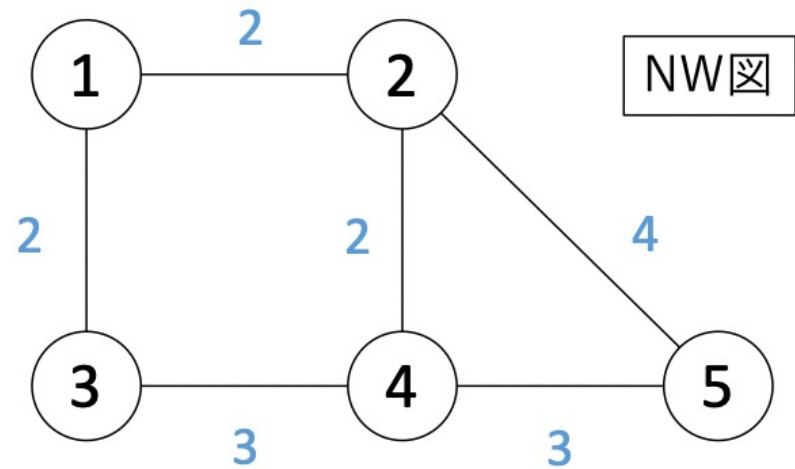


NW☒

リンク	流量
1-2	9
1-3	5+4
2-4	2+1
3-4	0
2-5	3+6
4-5	0
2-1	0+3+4
3-1	0
4-2	0
4-3	0
5-2	0
5-4	0

# 👉 all-or-nothing配分 – 単純な方法

OD交通量を全ての経路に流して、最後にリンクごとに集計する方法



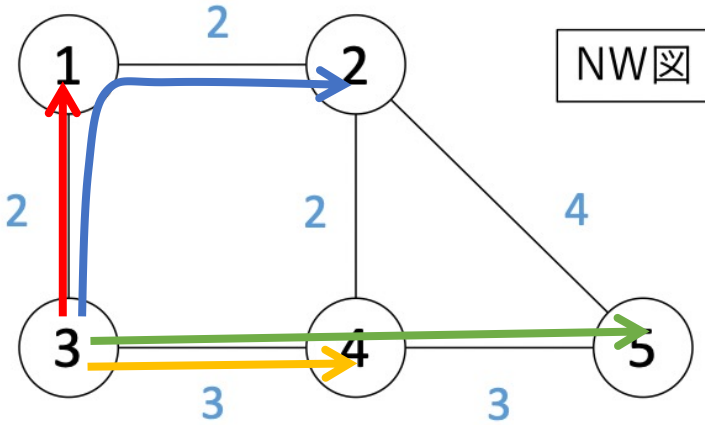
NW☒

OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

起点3に対する計算例

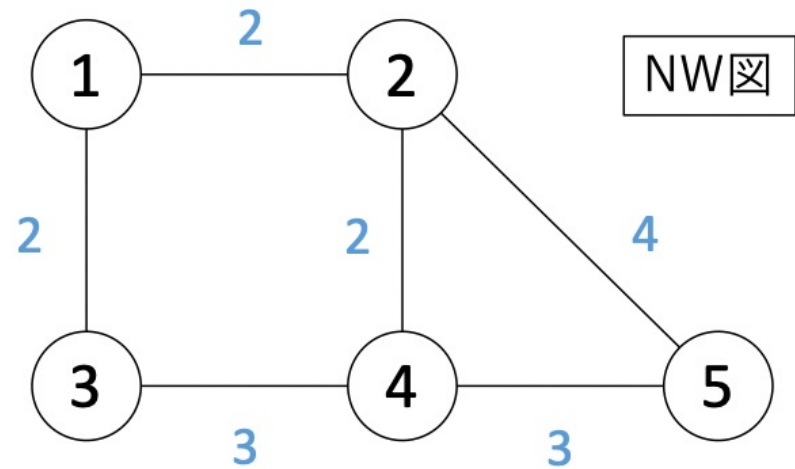


NW☒

リンク	流量
1-2	9+1=10
1-3	9
2-4	3
3-4	0+4+2=6
2-5	9
4-5	0+2=2
2-1	7
3-1	0+3+1=4
4-2	0
4-3	0
5-2	0
5-4	0

# 👉 all-or-nothing配分 – 単純な方法

OD交通量を全ての経路に流して、最後にリンクごとに集計する方法

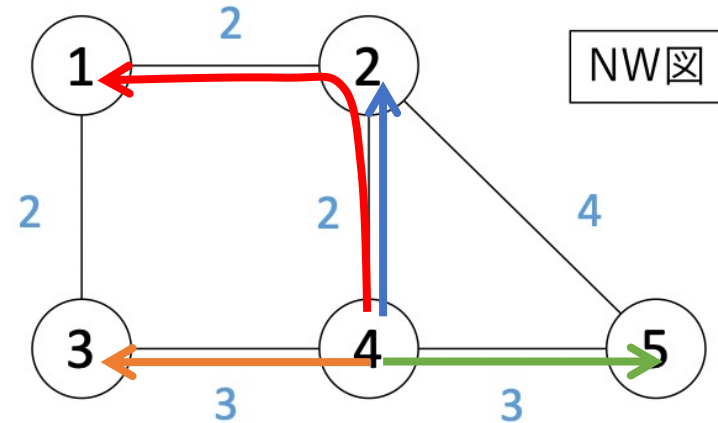


OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

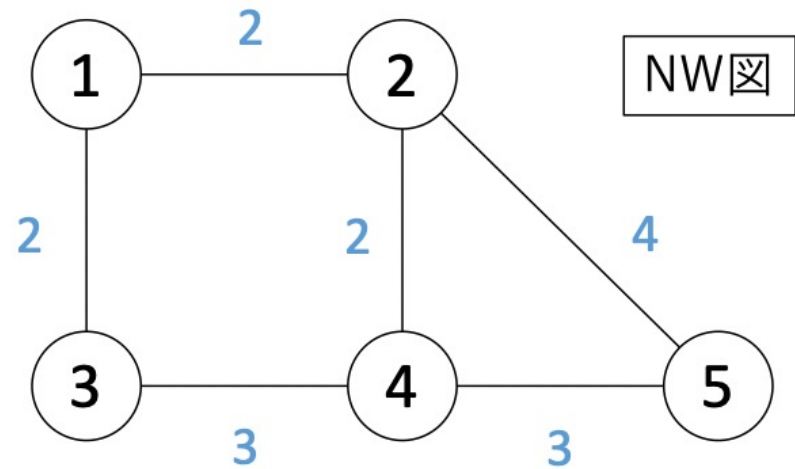
起点4に対する計算例



リンク	流量
1-2	10
1-3	9
2-4	3
3-4	6
2-5	9
4-5	2+2=4
2-1	7+2=9
3-1	4
4-2	0+2+4=6
4-3	0+2=2
5-2	0
5-4	0

# 👉 all-or-nothing配分 – 単純な方法

OD交通量を全ての経路に流して、最後にリンクごとに集計する方法

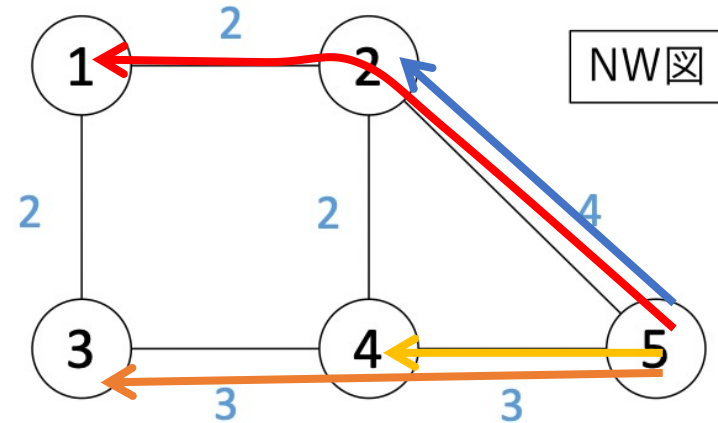


OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

起点5に対する計算例



リンク	流量
1-2	10
1-3	9
2-4	3
3-4	6
2-5	9
4-5	4
2-1	9+1=10
3-1	4
4-2	6
4-3	2+3=5
5-2	0+1+4=5
5-4	0+5+3=8



# all-or-nothing配分 – 単純な方法

## ■ all-or-nothing配分

- flow-independentな場合の配分アルゴリズム
- **最短経路探索 → 最小費用経路に全需要を配分**を全ての起点に対して実行（簡単！）

Step 1  $n = 0$ とする。全てのリンク交通量について $x_{ij}^{(0)} = 0$ とする。

Step 2  $n$ 番目の起点を $o$ とする。起点 $o$ から他の全てのノードへ最短経路探索を行い、起点 $o$ から他の全てのノードへの最短経路費用 $\{C_{\min}[o \rightarrow i]\}$ と各ノードに対する先行ポインタ $F_i$ を求める。 $y_{ij} = 0$ とする。

Step 3 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。

Step 4  $n = N$ （ノード数）なら終了。そうでなければ $n = n + 1$ としてStep 2へ。

→最短経路上のリンクの交通量を毎回検索するのは、（特に大規模なネットワークでは）効率が悪そう。  
→もう少し効率の良い方法を補足に載せておきます。

最短経路を全員が通ると、混雑によりもはや最短経路ではなくなるので...

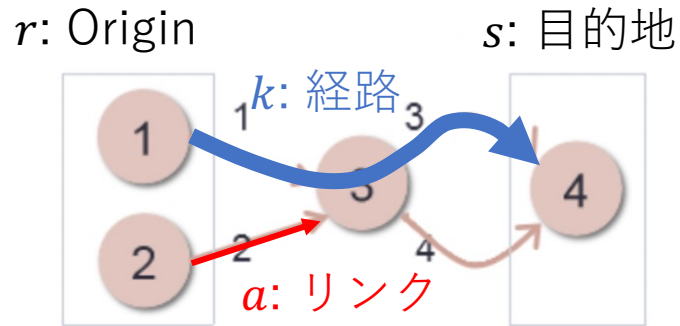
## 利用者均衡配分

	非混雑型 flow-independent	混雑型 flow-dependent
確定的経路 選択	<b>all-or-nothing配分</b>	<b>利用者均衡配分(UE)</b>
確率的経路 選択	<b>確率的配分</b>	<b>確率的利用者均衡配分 (SUE)</b>

# 利用者均衡(UE/FD)の定式化

Wardropの第1原則に従う, 需要固定型利用者均衡配分モデル (UE/FD: User Equilibrium with Fixed Demand) を定式化する

## ■ 変数の定義



$A = \{1,2,3,4\}$ : リンク集合  
 $\Omega = \{(1,4), (2,4)\}$ : OD集合

$q_{rs}$ : ODペア $rs$ 間の分布交通量

$\delta_{a,k}^{rs}$ : ODペア $rs$ 間の経路 $k$ がリンク $a$ を含むか否か  
 Trueなら1, Falseなら0

## ■ リンク単位の量とパス単位の量の関係

$c_k^{rs}$ : ODペア $rs$ を結ぶ経路 $k$ の旅行時間  
 $t_a$ : リンク $a$ の旅行時間

$$c_k^{rs} = \sum_{a \in A} t_a \delta_{a,k}^{rs}$$

$f_k^{rs}$ : ODペア $rs$ を結ぶ経路 $k$ の交通量  
 $x_a$ : リンク $a$ の交通量

$$x_a = \sum_{rs} \sum_k f_k^{rs} \delta_{a,k}^{rs}$$

例) 上のネットワークで  
 パス1: リンク1→リンク3  
 パス2: リンク1→リンク4  
 パス3: リンク2→リンク3  
 パス4: リンク2→リンク4  
 とすると,

$$c_1^{14} = t_1 \delta_{1,1}^{14} + t_2 \delta_{2,1}^{14} + t_3 \delta_{3,1}^{14} + t_4 \delta_{4,1}^{14} = t_1 \cdot 1 + t_2 \cdot 0 + t_3 \cdot 1 + t_4 \cdot 0$$

$$x_3 = f_1^{14} \delta_{3,1}^{14} + f_2^{14} \delta_{3,2}^{14} + f_3^{24} \delta_{3,3}^{24} + f_4^{24} \delta_{3,4}^{24} = f_1^{14} + f_3^{24}$$

# 利用者均衡(UE/FD)の定式化

Wardropの第1原則に従う, 需要固定型利用者均衡配分モデル (UE/FD: User Equilibrium with Fixed Demand) を定式化する

## ■ Wardropの第1原則の定式化

$$f_k^{rs} > 0 \text{ のとき } c_k^{rs} = c_{min}^{rs} \quad \forall k \in K_{rs}, \forall rs \in \Omega$$

$$f_k^{rs} = 0 \text{ のとき } c_k^{rs} \geq c_{min}^{rs} \quad \forall k \in K_{rs}, \forall rs \in \Omega$$

s.t.

$$\sum_{k \in K_{rs}} f_k^{rs} - q_{rs} = 0 \quad \forall rs \in \Omega \quad \text{流量保存則}$$

$$f_k^{rs} \geq 0 \quad \forall k \in K_{rs}, \forall rs \in \Omega \quad \text{流量は非負}$$

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

$f_k^{rs}$ : ODペア $rs$ 間のパス $k$ の流量  
 $c_k^{rs}$ : ODペア $rs$ 間のパス $k$ の旅行時間  
 $c_{min}^{rs}$ : ODペア $rs$ 間の最短経路所要時間  
 $q_{rs}$ : ODペア $rs$ 間の分布交通量

現実規模のネットワークでこの解を得るのは当初非常に困難だった

➡ Beckmann et al.(1956)が数理最適化問題に変換!

最適化問題であれば解法が確立されているため効率よく解ける!

# UE/FDの定式化 – 等価最適化問題への変換

## ■ Wardropの利用者均衡の等価最適化問題

### UE/FD-Primal

$$\min Z(x) = \sum_{a \in A} \int_0^{x_a} t_a(w) dw$$

s.t.

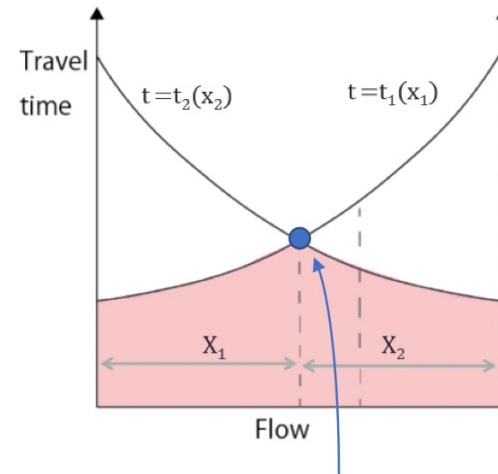
$$\sum_{k \in K_{rs}} f_k^{rs} - q_{rs} = 0 \quad \forall rs \in \Omega$$

$$x_a = \sum_{rs \in \Omega} \sum_{k \in K_{rs}} f_k^{rs} \delta_{a,k}^{rs} \quad \forall a \in A$$

$$f_k^{rs} \geq 0 \quad \forall k \in K_{rs}, \forall rs \in \Omega$$

$$x_a \geq 0 \quad \forall a \in A$$

目的関数は直感的には下のように理解できる



均衡点：積分の値が最小となる点

$t_a(x_a)$ : リンク  $a$  の旅行時間  
 $x_a$ : リンク  $a$  の交通量  
 $f_k^{rs}$ : ODペア  $rs$  間のパス  $k$  の流量  
 $q_{rs}$ : ODペア  $rs$  間の分布交通量  
 $\delta_{a,k}^{rs}$ : ODペア  $rs$  間のパス  $k$  がリンク  $a$  を含むか否か (True=1, False=0)

- 十分性の証明 (詳しい証明は教科書や昨年度資料参照)

UE/FD-PrimalのKKT条件が元の問題と一致することにより証明できる

- 解の一意性の証明 (詳しい証明は教科書や昨年度資料参照)

変数の実行可能領域が凸 (::制約条件式が全て線形)

目的関数が狭義の凸関数  $\Leftrightarrow$  Hessianが正定値 (::リンクパフォーマンス関数が単調増加)

# UE/FDの解法

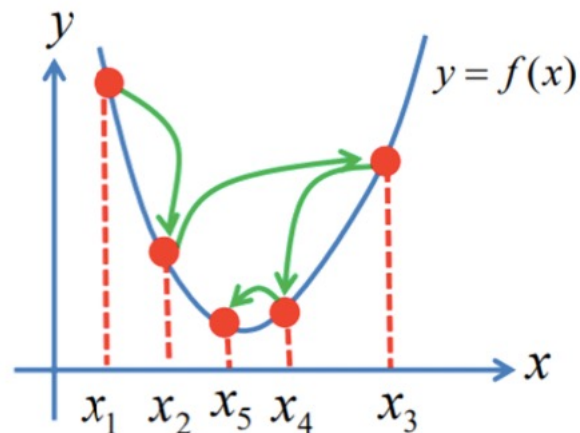
## ■ 非線形最適化問題の一般的な解法

### ステップA：降下方向の探索

どの方向に向かえば目的関数が減少するか

### ステップB：ステップサイズの探索

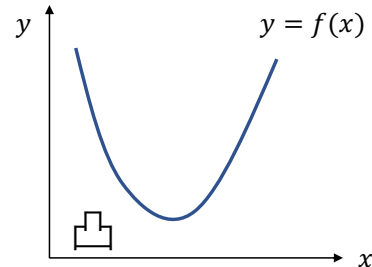
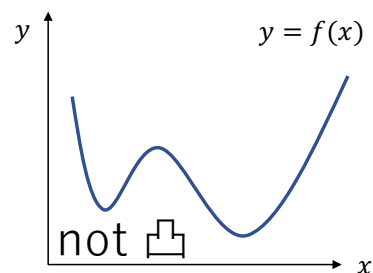
降下方向にどこまで進めるか



ステップAとステップBを繰り返すことで局所最適解を見つける

(UE/FD-Primalのような) 凸計画問題では、局所最適解が大域的最適解に一致する→解の一意性

目的関数が凸関数で、実行可能領域が凸集合であるような最適化問題



$f$ が凸関数

⇔任意の $a, b$ に対して,

$$\lambda f(a) + (1 - \lambda)f(b) \geq f(\lambda a + (1 - \lambda)b) \quad \forall \lambda \in [0, 1]$$

→関数上の任意の2点をとって線分を引いた時、その線が必ず元の関数より上に来る

# UE/FDの解法 – Frank-Wolfe法

## ■ Frank-Wolfe法の概要

解きたい問題

UE/FD-Primal

$$\min Z(x) = \sum_{a \in A} \int_0^{x_a} t_a(w) dw$$

s.t.

$$\sum_{k \in K_{rs}} f_k^{rs} - q_{rs} = 0 \quad \forall rs \in \Omega$$

$$x_a = \sum_{rs \in \Omega} \sum_{k \in K_{rs}} f_k^{rs} \delta_{a,k}^{rs} \quad \forall a \in A$$

$$f_k^{rs} \geq 0 \quad \forall k \in K_{rs}, \forall rs \in \Omega$$

$$x_a \geq 0 \quad \forall a \in A$$

凸計画問題

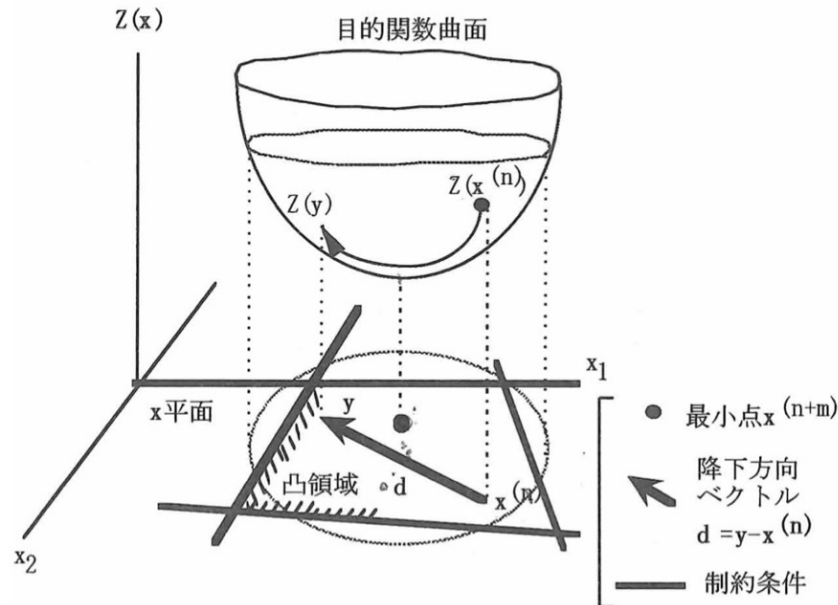


図-8.6 制約条件と方向ベクトル  $(y-x)$  と目的関数上の動き

n回目の解  $x^{(n)}$  から n+1回目の解  $x^{(n+1)}$  へ動く時

### ステップA：降下方向の探索

降下方向ベクトル  $d = y - x^{(n)}$  として、方向ベクトル  $y$  を求める。  
→別の最適化問題を解く

### ステップB：ステップサイズの探索

$x^{(n+1)} = x^{(n)} + \alpha d$  として、  
 $Z(x^{(n+1)})$  を最小化する  $x^{(n+1)}$  を求める。  
 $x^{(n)}$  と  $d$  は既に求まっているので、  
 $Z(x^{(n+1)})$  を最小にする  $\alpha$  を求めることになる。

これらの繰り返し

# UE/FDの解法 – Frank-Wolfe法

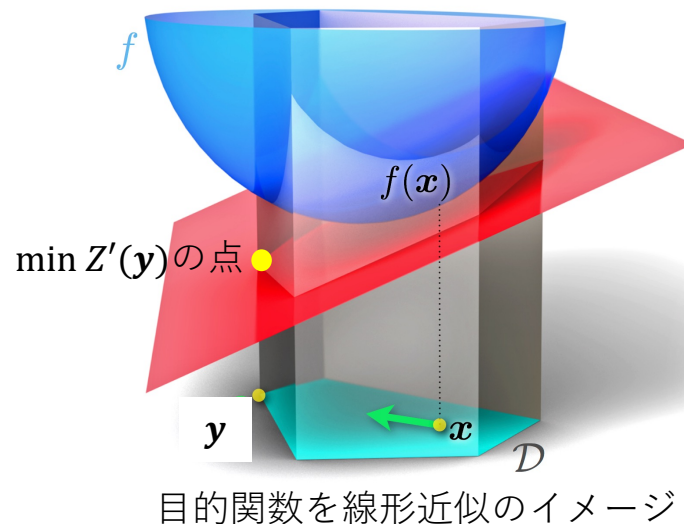
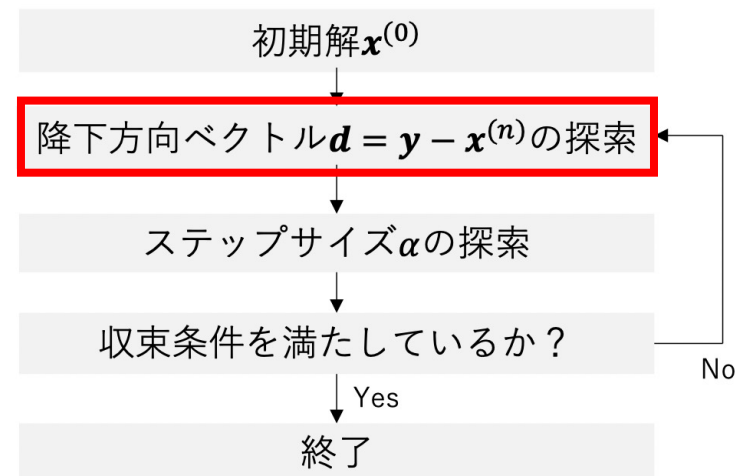
## ■ 降下方向ベクトル $\mathbf{d} = \mathbf{y} - \mathbf{x}^{(n)}$ の探索

目的関数  $Z(\mathbf{x})$  を  $\mathbf{x}^{(n)}$  で線形近似し,  $\mathbf{y} = \mathbf{x}^{(n)} + \mathbf{d}$  での目的関数の近似値  $Z'(\mathbf{y})$  を最小化する降下方向  $\mathbf{d}$  を決定

$$\begin{aligned}
 Z(\mathbf{y}) &\simeq Z'(\mathbf{y}) = Z(\mathbf{x}^{(n)}) + \nabla Z(\mathbf{x}^{(n)})^T (\mathbf{y} - \mathbf{x}^{(n)}) \\
 &= Z_p(\mathbf{x}^{(n)}) + \sum_{a \in A} (y_a - x_a^{(n)}) \frac{\partial Z(\mathbf{x}^{(n)})}{\partial x_a^{(n)}} \quad \text{Z(x)の定義より} \\
 &= Z_p(\mathbf{x}^{(n)}) + \sum_{a \in A} (y_a - x_a^{(n)}) t_a(\mathbf{x}^{(n)}) \\
 &= \underbrace{Z_p(\mathbf{x}^{(n)})}_{\text{定数}} - \sum_{a \in A} \underbrace{x_a^{(n)} t_a(\mathbf{x}^{(n)})}_{\text{定数}} + \sum_{a \in A} y_a^{(n)} t_a(\mathbf{x}^{(n)})
 \end{aligned}$$

以下の補助問題を解いて  $\mathbf{y}$  を出せば良い。

$$\min Z'(\mathbf{y}) = \sum_{a \in A} y_a^{(n)} t_a(\mathbf{x}^{(n)}) \quad \text{s.t.} \quad \sum_{k \in K_{rs}} f_k^{rs} - q_{rs} = 0, \quad y_a = \sum_{r \in \Omega} \sum_{k \in K_{rs}} f_k^{rs} \delta_{a,k}^{rs}$$





# UE/FDの解法 – Frank-Wolfe法

## ■ 降下方向ベクトル $d = y - x^{(n)}$ の探索

### 補助問題

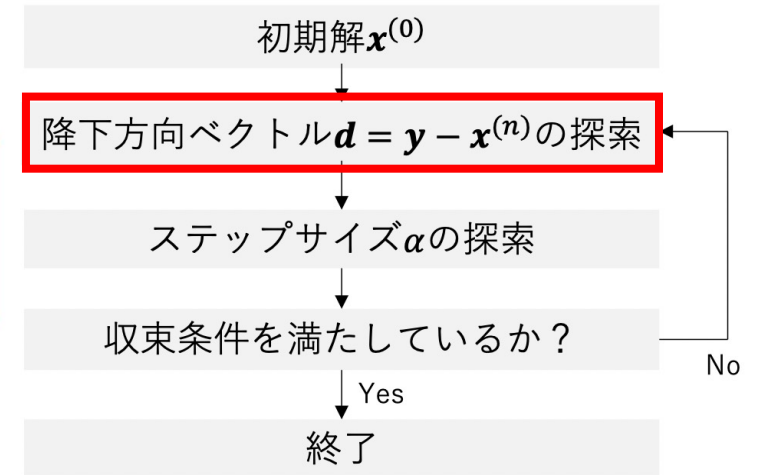
$$\min Z'(y) = \sum_{a \in A} y_a^{(n)} t_a(x_a^{(n)}) \quad \text{s.t.} \quad \sum_{k \in K_{rs}} f_k^{rs} - q_{rs} = 0, \quad y_a = \sum_{r \in \Omega} \sum_{k \in K_{rs}} f_k^{rs} \delta_{a,k}^{rs}$$

$t_a(x_a^{(n)}) = \text{const.}$ のもとで総旅行時間を最小とする  $y$  を求める

n回目の解のもとのリンク旅行時間

$t_a(x_a^{(n)})$ 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  
**all-or-nothing配分**により  $Z'(y)$  を最小にする  $y$  が得られる

降下方向ベクトル  $d = y - x^{(n)}$  が求まる



簡単なアルゴリズムであるall-or-nothing配分を繰り返し用いることにより、  
降下方向を確定できる！

# UE/FDの解法 – Frank-Wolfe法

## ■ ステップサイズ $\alpha$ の探索

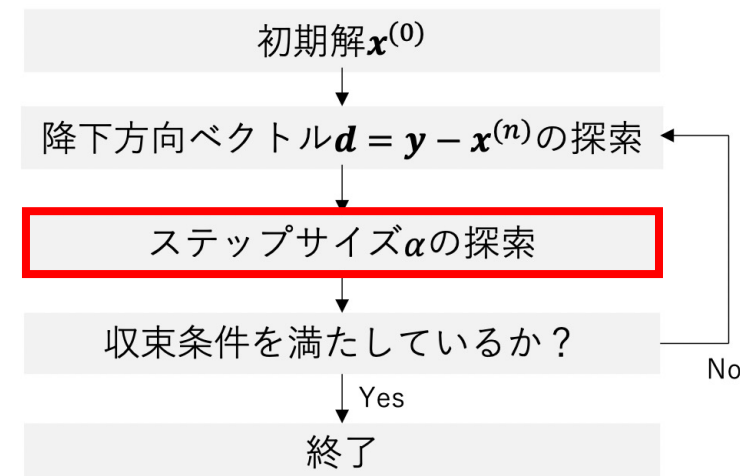
降下方向ベクトル $\mathbf{d} = \mathbf{y} - \mathbf{x}^{(n)}$ が求まったので、  
 $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \alpha \mathbf{d} = \mathbf{x}^{(n)} + \alpha(\mathbf{y} - \mathbf{x}^{(n)})$ となる。  
定数 定数

次は、 $Z(\mathbf{x}^{(n+1)})$ を最小にする $\alpha$ を求める。

$$\min_{0 \leq \alpha \leq 1} Z(\alpha) = \sum_a \int_0^{x_a^{(n+1)}} t_a(w) dw \quad \text{一変数関数の最小化}$$

- 積分を予め計算しておく
- 黄金分割法等の一次元探索法を用いる (末尾に補足)

といった工夫により、効率よく $\alpha$ を求めることができる



# UE/FDの解法 – Frank-Wolfe法

## ■ (まとめ) Frank-Wolfe法によるUEの解法

### Step 1 初期実行可能解の設定

交通量0におけるall-or-nothing配分により, 初期実行可能解となるリンク交通量 $\{x_a^{(n)}\}$ を与える( $n = 1$ ).

### Step 2 リンクコストの更新

$\{x_a^{(n)}\}$ に対する所要時間 $\{t_a(x_a^{(n)})\}$ を計算.

### Step 3 降下方向ベクトルの探索

all-or-nothing配分により全交通需要をリンクに負荷.  $\{y_a^{(n)}\}$ を求める.

### Step 4 ステップサイズの探索

黄金分割法等により $Z(x^{(n+1)})$ を最小にするステップサイズ $\alpha^{(n)}$ とリンク交通量 $\{x_a^{(n+1)}\}$ を求める.

### Step 5 収束判定

予め設定した十分小さい $\epsilon_1, \epsilon_2$ に対して, (総旅行時間変化)  $\leq \epsilon_1$ , (交通量変化)  $\leq \epsilon_2$ ,  $n > K$  ( $K$ は任意に与える)のいずれかを満たせば計算を終了する.

そうでなければ,  $n = n + 1$ としてStep2へ.

# UE/FDの解法 – Frank-Wolfe法

## ■ (まとめ) Frank-Wolfe法によるUEの解法

### 特徴

◎必要な記憶容量が少ない

◎手順が簡単

×収束が進むと解の近傍で”ジグザグ運動” → **収束スピードが緩慢**

### 改良手法

#### **打ち切り二次計画法**

目的関数のTaylor展開の二次の項まで用いる

#### **Simplicial Decomposition法**

均衡解を端点解(all-or-nothing配分)の凸結合(重みつき重ね合わせ)と捉え, それらの → 重村理論談話会にて  
ウェイトを求める

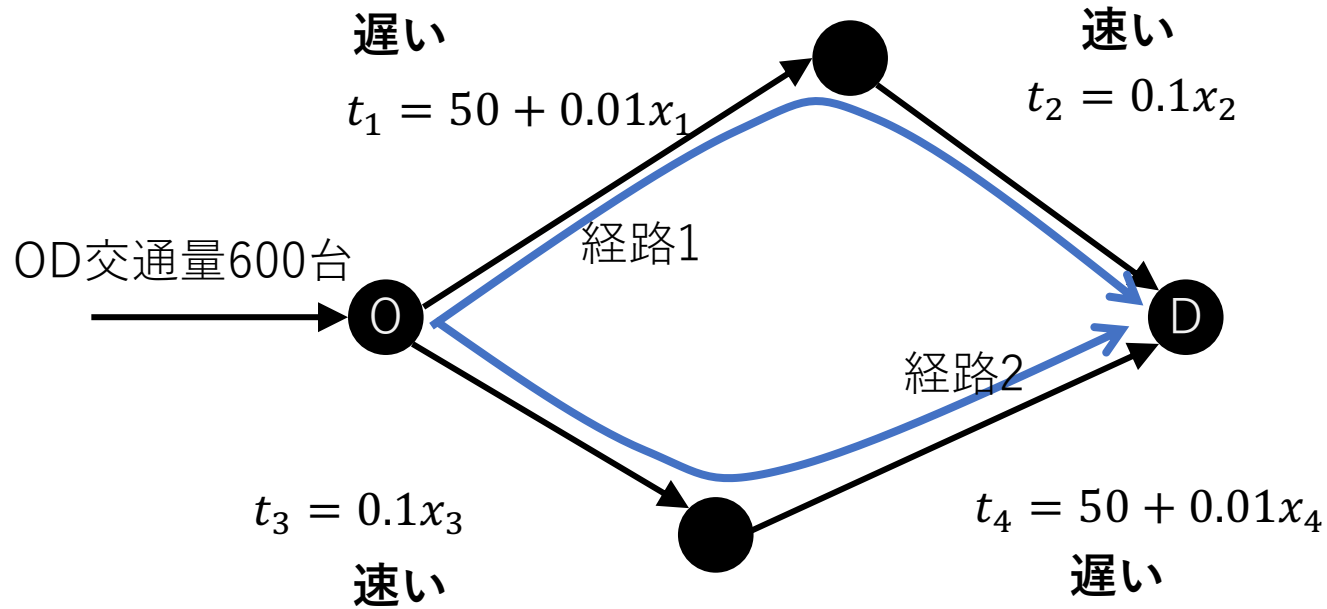
### 補足

リンクに容量制約がある問題では, ステップサイズ探索時に以下の制約条件を追加,

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

# 👉 Braess's paradox

以下のような道路ネットワークで、Wardropの等時間原則に従って配分する。



ネットワークは対称なので、

経路1,2の交通量:  $f_1 = f_2 = 300$

経路1,2の旅行時間:  $c_1 = c_2 = (50 + 3) + 30 = 83$

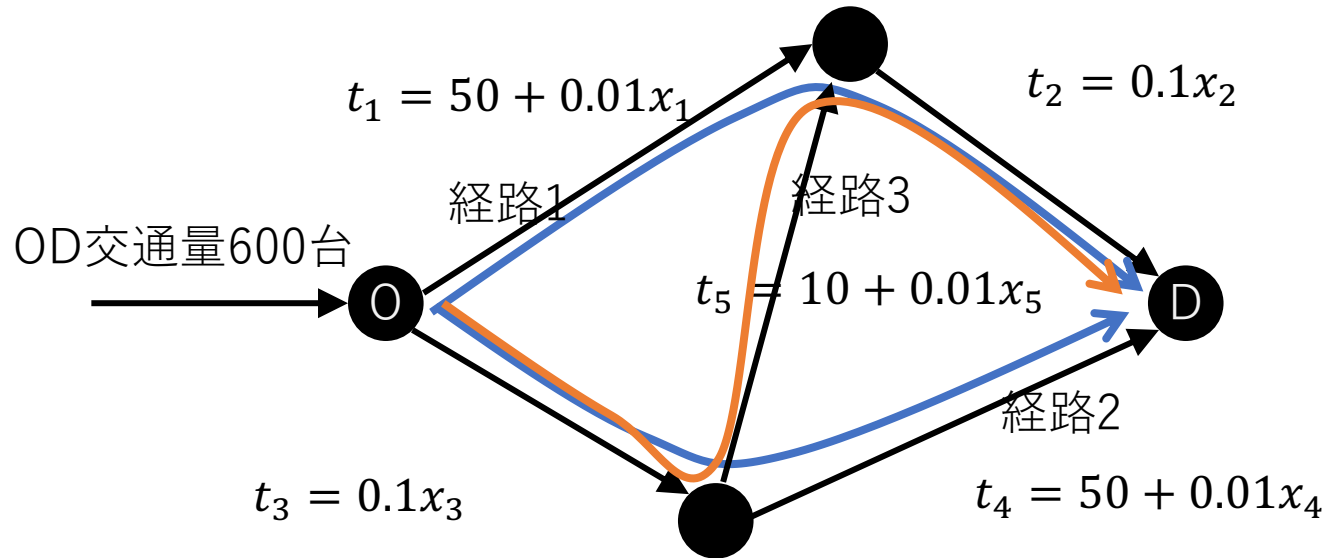
総所要時間(SOの目的関数):

$$300 * 83 + 300 * 83 = 49800$$

このネットワークに、計画者がサービスレベル向上を目指してバイパスを建設したことを考える。

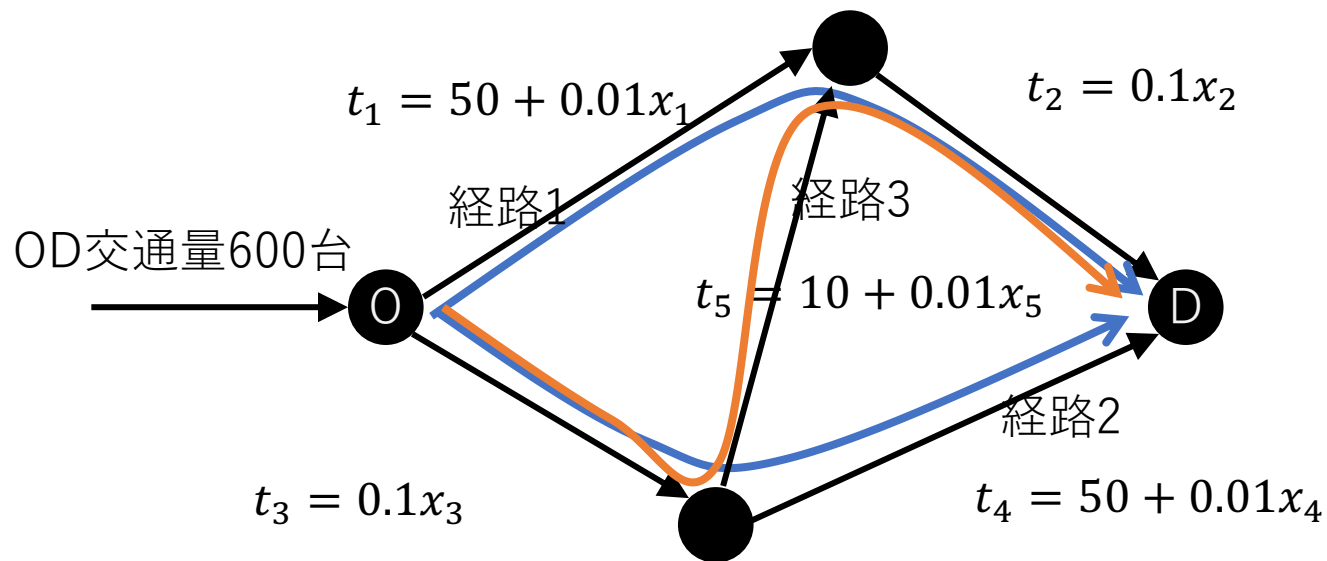
# 👉 Braess's paradox

(演習) バイパス建設後の交通量を, Wardropの等時間原則に従って求めてください (立式までOK) .



# 👉 Braess's paradox

(演習) バイパス建設後の交通量を, Wardropの等時間原則に従って求めてください.



$$f_1 = f_2 = f_3 = 200$$

$$c_1 = c_2 = c_3 = 92$$

総旅行時間(SOの目的関数):

$$200 * 92 + 200 * 92 + 200 * 92 = 55200$$

経路1,2,3の交通量を  $f_1, f_2, f_3$  として,

$$\begin{cases} (50 + 0.01f_1) + 0.1(f_1 + f_3) = 0.1(f_2 + f_3) + (50 + 0.01f_2) = 0.1(f_2 + f_3) + (10 + 0.01f_3) + 0.1(f_1 + f_3) \\ f_1 + f_2 + f_3 = 600 \end{cases}$$

総所要時間は49800から55200へと増加しており, 新規の道路建設がかえってサービスレベルを悪化させた.  
→Braessのパラドックス

# 粒としての交通 vs 流れとしての交通



# 粒の視点で目的関数を考え直してみる

- 目的関数の式は何を意味するか

## UE/FD-Primal

$$\min Z(\mathbf{x}) = \sum_{a \in A} \int_0^{x_a} t_a(w) dw$$

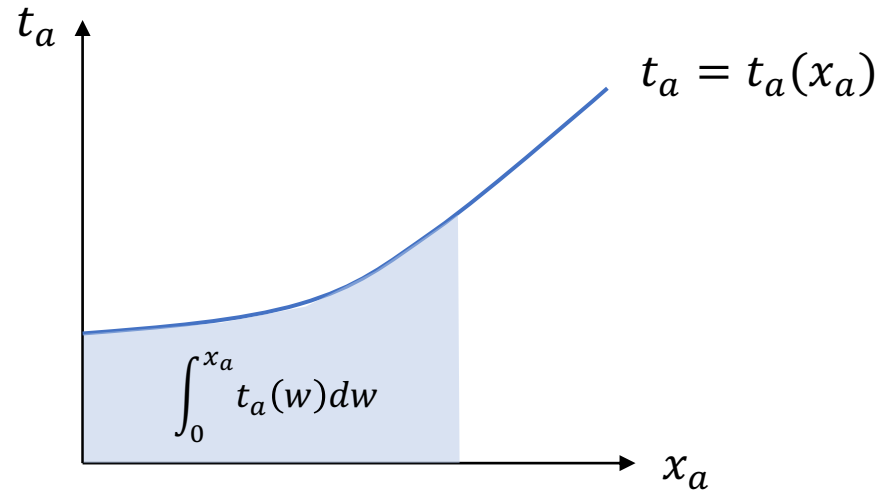
s.t.

$$\sum_{k \in K_{rs}} f_k^{rs} - q_{rs} = 0 \quad \forall rs \in \Omega$$

$$x_a = \sum_{rs \in \Omega} \sum_{k \in K_{rs}} f_k^{rs} \delta_{a,k}^{rs} \quad \forall a \in A$$

$$f_k^{rs} \geq 0 \quad \forall k \in K_{rs}, \forall rs \in \Omega$$

$$x_a \geq 0 \quad \forall a \in A$$



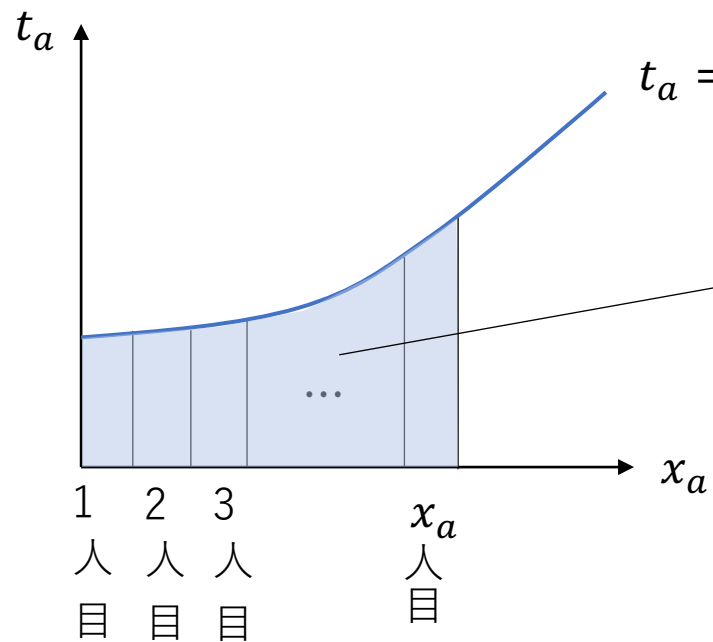
$x_a$ : リンク  $a$  の交通量  
 $t_a$ : リンク  $a$  の所要時間

リンクパフォーマンス関数

ひとりひとりの意思決定者に注目 → ゲーム理論のアプローチで解釈

# ポテンシャル・ゲームとしての記述

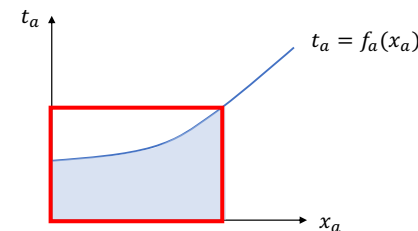
- 各時点で1人ずつ順番に選ばれて、「より早く目的地に到達できるルートがあれば経路を変更する」を繰り返すことを考える。（混雑ゲームという）



$Q_a = \sum_{x=1}^{x_a} t_a(x)$  という量を定義し、

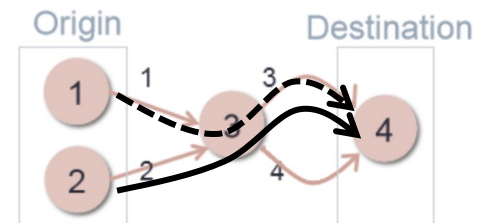
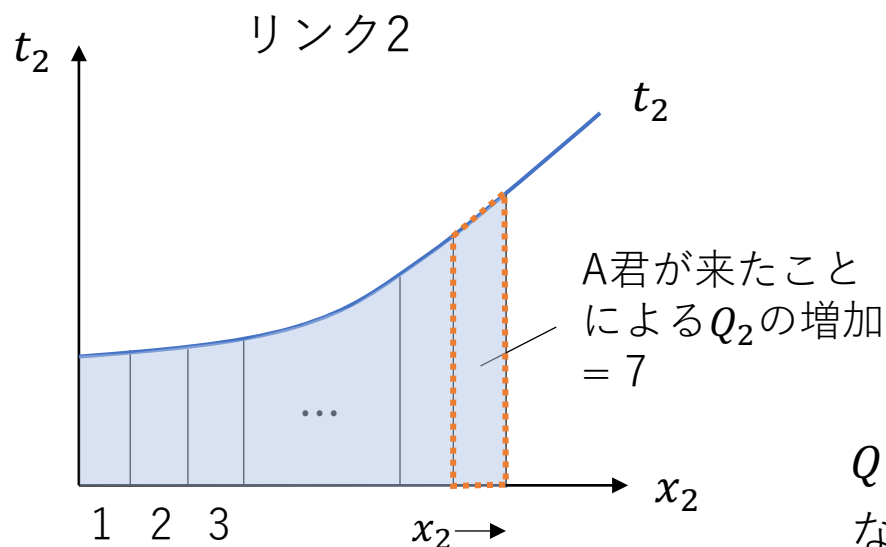
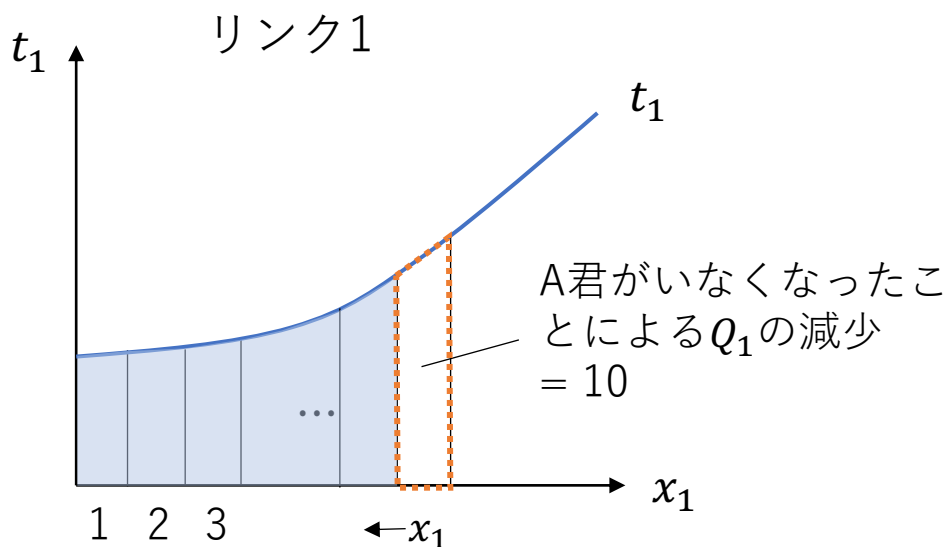
$Q = Q_1 + Q_2 + \dots + Q_k = \sum_a \sum_{x=1}^{x_a} t_a(x)$  という量を考える

※  $Q_a$  は総旅行時間ではない。  
総旅行時間は右図赤枠の面積



# ポテンシャル・ゲームとしての記述

例えば，A君がリンク1を通る経路から，リンク2を通る経路に変更することで，変更した部分の旅行時間が10分から7分になったとする．このとき $Q$ の変化量は，



$Q = Q_1 + Q_2 + \dots + Q_k$   
なので， $Q$ は3減った

プレイヤー1人だけが行動を変えたときの  
その人の利得（損失）の変化

= グローバルな量  $Q$  の変化

# ポテンシャル・ゲームとしての記述

プレイヤー1人だけが行動を変えたときの  
その人の利得（損失）の変化

= グローバルな量  $Q$  の変化

このとき  $Q$  を **ポテンシャル関数** といい、そのような量のあるゲームを **ポテンシャル・ゲーム** という。  
ポテンシャル関数  $Q$  が最小のとき、あるプレイヤーだけが行動を変えるとき  $Q$  の変化量は正で、  
その人の損失の変化も正。

→ プレイヤーは行動を変えるインセンティブを持たない = **Nash均衡**

自分一人が行動を変えることによって利得を増加させることができない状態

$$\min Q = \sum_a Q_a = \sum_a \sum_{x=1}^{x_a} t_a(x)$$

交通量を連続に

$$\min Q = \sum_a \int_0^{x_a} t_a(w) dw$$

(UEの目的関数)

ポテンシャルゲームの嬉しさ:

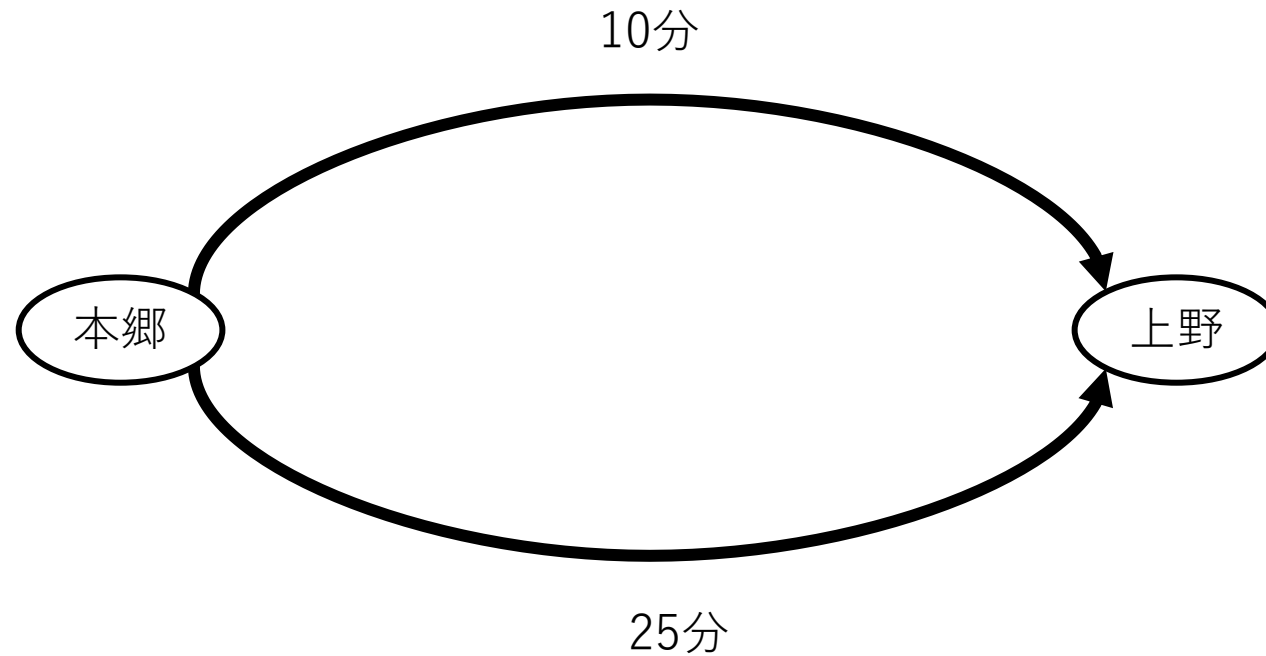
プレイヤー間の均衡をそのまま扱うのではなく、 **$Q$ の最小化問題**に置き換えることができる！  
→ 解きやすい

# ゲームの時間

# 経路選択ゲームをしてみよう

どちらの道を選びますか??

選ぶ人が増えるとどちらの道も混雑します.



# 動学的なゲームとしての記述

- 混雑ゲームでは、均衡状態での各経路の配分交通量のみを求める
  - 均衡状態に至るまでの**過程**を扱わない
  - Nash均衡において、プレイヤーは自分の行動結果と他のプレイヤーの意図を先読みして、自分にとって最も利得の高そうな戦略を取ること（**合理性**）を仮定  
→本当??
- **進化ゲーム**として記述
  - 先読みは特に仮定せず（**限定合理性**），実際にある戦略を取った後に、それがうまくいったかどうかによって事後的に戦略のシェアが増減する，という仮定をおく  
→ **Day-to-day（日ごと）の配分の調整過程**
  - 均衡に至るまでの経路やその安定性について解析

# 進化ゲームとしての記述

- Sandholm (2001)では、ポテンシャル・ゲームが均衡に達するまでの進化動学 $\mathbf{V}$ を導入  
ポテンシャル・ゲーム $\mathbf{g}$ について、以下の条件を満たす進化動学 $\dot{\mathbf{f}} = \mathbf{V}(\mathbf{f})$ を考える。
  - $\mathbf{V}$ はリプシッツ連続 …… 急激な変化は起こらない 各経路の交通量
  - $V_k(\mathbf{f}) \geq 0$  if  $f_k = 0$  …… 選ばれていない経路 $k$ が選ばれる可能性が存在
  - $\sum_{k \in K_{rs}} V_k(\mathbf{f}) = 0 \quad \forall \mathbf{f}, rs \in \Omega$  …… (増減が打ち消しあうので) 交通量 $f$ は一定
  - $\mathbf{V}(\mathbf{f}) \cdot \mathbf{g}(\mathbf{f}) > 0$  if  $\mathbf{V}(\mathbf{f}) \neq 0$  …… 経路交通量の変化と利得との間に正相関 (利得が大きい経路の交通量が増える)
  - $\mathbf{V}(\mathbf{f}) = 0$ のとき、 $\mathbf{f}$ はナッシュ均衡である …… ナッシュ均衡において配分結果は変化しない

例えば、以下のようなダイナミクスがある ( $k, l$ は経路,  $f$ は交通量,  $c$ はコスト)

$$V_k(\mathbf{f}) = - \sum_{l \neq k} f_l \cdot \max(c_k(\mathbf{f}) - c_l(\mathbf{f}), 0) + \sum_{l \neq k} f_l \cdot \max(c_l(\mathbf{f}) - c_k(\mathbf{f}), 0)$$

コストの大きい方に  
交通量が流出

コストの小さい方から  
流入



# 進化ゲームとしての記述

このとき、以下が成立する.

1. ダイナミクス $V$ の任意の解軌道は、ポテンシャル・ゲーム $g$ のナッシュ均衡の連結部分集合に収束する.
2. ポテンシャル関数 $\phi$ が**狭義凹関数**のとき、 $\phi$ を最大化する点はポテンシャルゲーム $g$ の**唯一のナッシュ均衡**であり、ダイナミクス $V$ のglobal attractorである

→ 適当な動学 $V$ を仮定してやれば、合理性の仮定を緩めても、day-to-dayの調整過程を終えた後には、**静学的な利用者均衡配分と同じ状態に収束する**

→ 三上・手代木理論談話会

# 確率的配分

	非混雜型 flow-independent	混雜型 flow-dependent
確定的經路 選擇	<b>all-or-nothing配分</b>	<b>利用者均衡配分(UE)</b>
確率的經路 選擇	<b>確率的配分</b>	<b>確率的利用者均衡配分 (SUE)</b>

# 利用者均衡の仮定を緩める

## ■ 経路選択が確率的である場合の配分原理

(再掲) Wardropの第1原則の仮定

- ① 全ての利用者は常に**旅行時間を最小とするように行動**する → 気紛れや個人の嗜好が影響するかも
- ② 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 → 利用者が完全な情報を得ているとは限らないし、データそのものに観測誤差が含まれているかも

	非混雑型 flow-independent	混雑型 flow-dependent
確定的経路 選択	all-or-nothing配分	利用者均衡配分(UE)
確率的経路 選択	確率的配分	確率的利用者均衡配分 (SUE)



ランダム効用理論による確率的な経路選択に基づく配分モデルが必要である。

= ロジット, プロビット

次にflow-independentな場合の確率的経路選択の配分方法である、**確率的配分**を計算してみる。

# ✍️ 確率的配分の例 – Dialのアルゴリズム

$O_i$ : ノード*i*から流出するリンクの終点集合,  $I_i$ : ノード*i*に流入するリンクの始点集合

## Step 1 最短経路探索

起点*r*から他の全てのノードへの最小交通費用*c(i)*を計算.  $y_{ij} = 0$ とする.

## Step 2 全リンクについてリンク尤度*L[i → j]*を計算

$$L[i \rightarrow j] = \begin{cases} \exp[\theta\{c(j) - c(i) - t_{ij}\}] & c(i) < c(j) \\ 0 & otherwise \end{cases}$$

※ 各リンクの選ばれやすさを表す.  
最短経路上で1, 後戻りする経路で0, それ以外だと0~1の間.  
最短経路との差が大きい経路ほど0に近くなる.

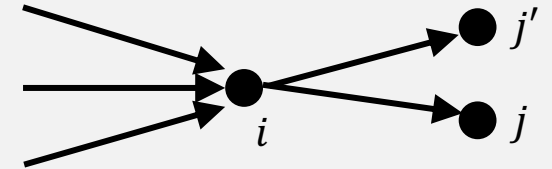
## Step 3 前進処理

起点*r*から*c(i)*の値の昇順 (*r*から近い順) にノードを考え, 各ノード*i*から流出するリンクの**リンクウェイト**  $W[i \rightarrow j]$ を計算

$$W[i \rightarrow j] = \begin{cases} L[i \rightarrow j] & for\ i = r \\ L[i \rightarrow j] \sum_{m \in I_i} W[m \rightarrow i] & otherwise \end{cases}$$

※ 各ノードでの分岐確率を表す.

*ij*の選ばれやすさに従って  
ウェイトを割り振る



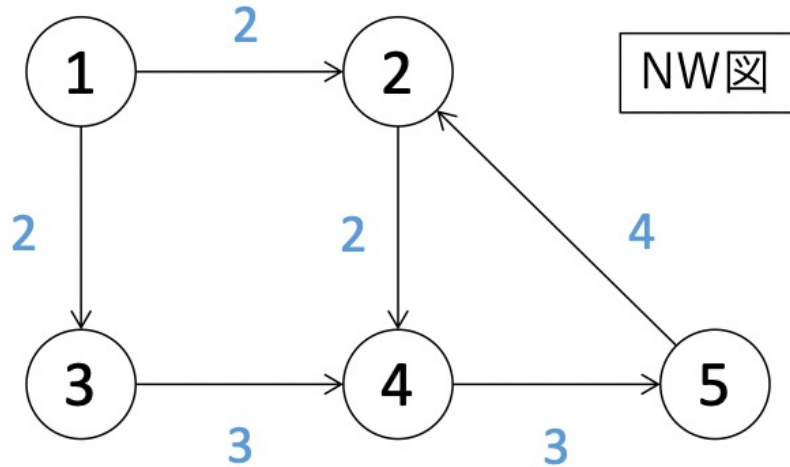
## Step 4 後退処理

*c(i)*の値の降順 (*r*から遠い順) にノードを考え, 各ノード*j*に流入するリンクの**交通量**  $x_{ij}$ を計算

$$y_{ij} = \left( q_{rj} + \sum_{m \in O_j} y_{jm} \right) \frac{W[i \rightarrow j]}{\sum_{m \in I_i} W[m \rightarrow i]}, \quad x_{ij}^{(n)} = x_{ij}^{(n-1)} + y_{ij}$$

# ✍️ 確率的配分の例 – Dialのアルゴリズム

## • Dialのアルゴリズム ( $\theta = 1$ )



OD表

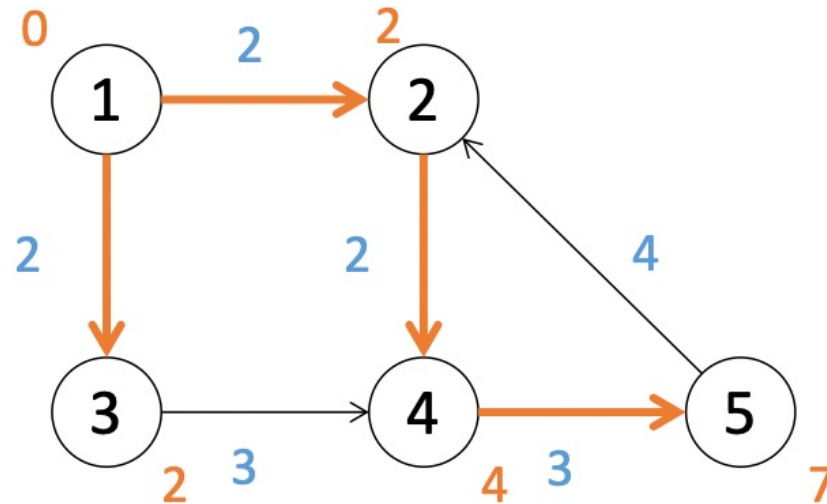
値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

Step 1

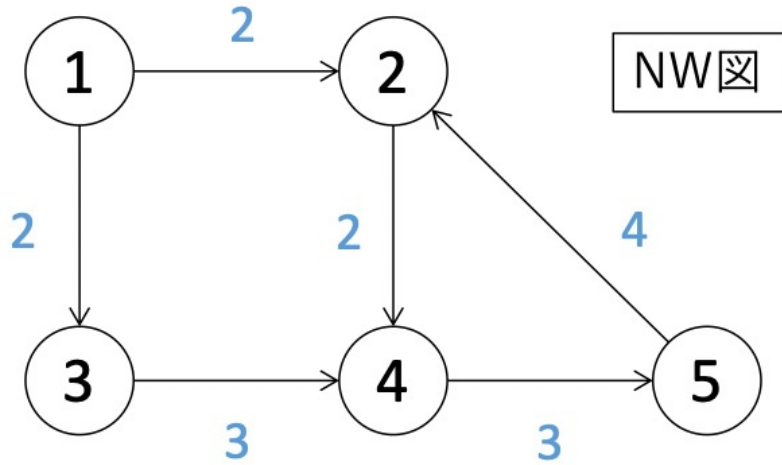
### 最短経路探索

起点  $r = 1$  から他のすべてのノードへの最小交通費用  $c(i)$  を計算



# ✍️ 確率的配分の例 – Dialのアルゴリズム

## • Dialのアルゴリズム ( $\theta = 1$ )



NW図

OD表

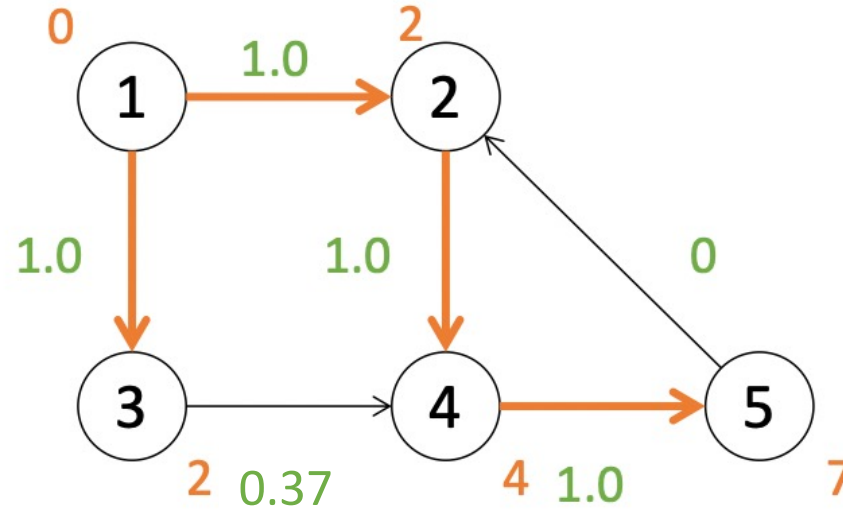
値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

全リンクについて  
リンク尤度  $L[i \rightarrow j]$  を計算

Step 2 
$$L[i \rightarrow j] = \begin{cases} \exp[\theta\{c(j) - c(i) - t_{ij}\}] & c(i) < c(j) \\ 0 & otherwise \end{cases}$$

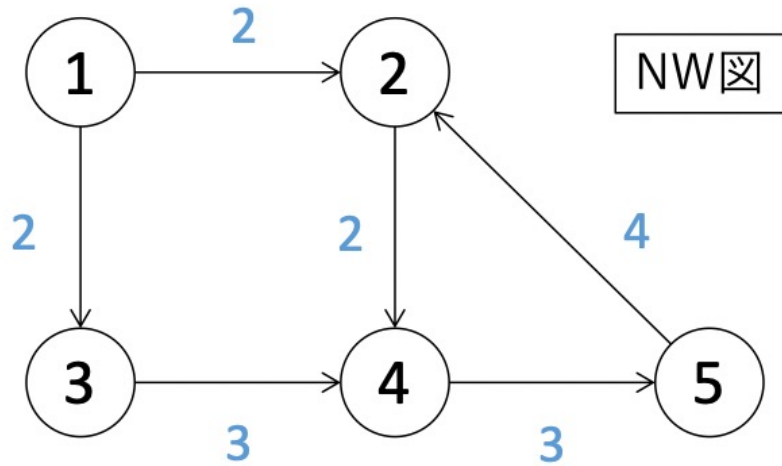
最短経路上は尤度1, 後戻りする経路は0



$$\begin{aligned} L[3 \rightarrow 4] &= \exp[1 * \{c(4) - c(3) - t_{34}\}] \\ &= \exp[4 - 2 - 3] \\ &= \exp[-1] = 0.37 \end{aligned}$$

# ✍️ 確率的配分の例 – Dialのアルゴリズム

## • Dialのアルゴリズム ( $\theta = 1$ )



NW図

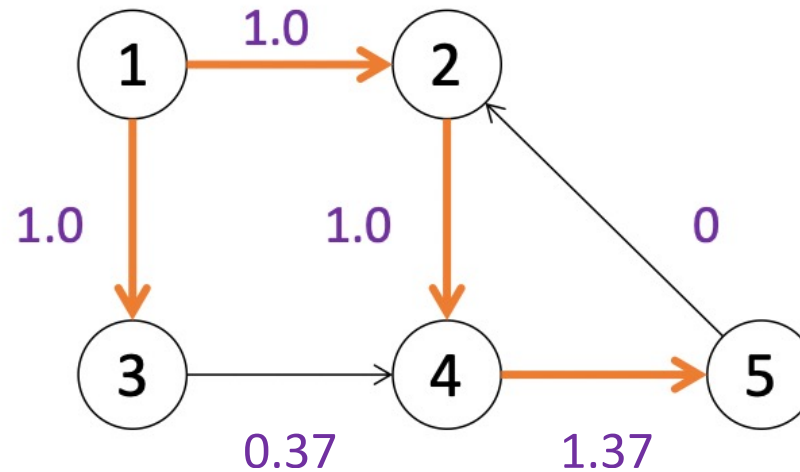
OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

**前進処理**  
 起点 1 から  $c(i)$  の値の昇順にノードを考え、各ノード  $i$  から流出するリンクのリンクウェイト  $W[i \rightarrow j]$  を計算

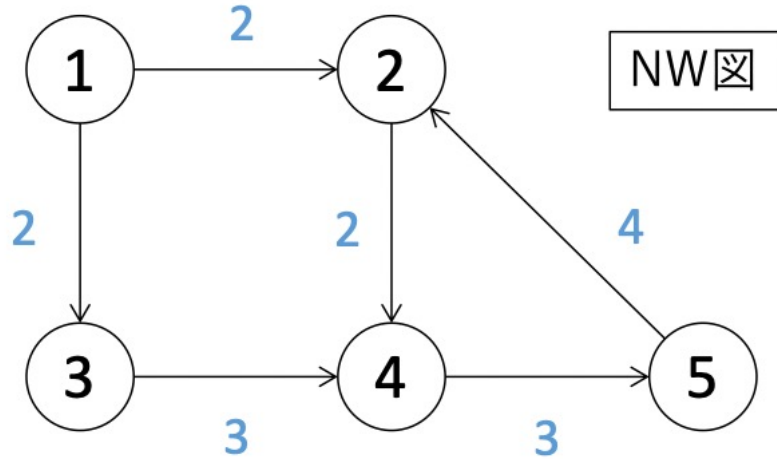
Step 3

$$W[i \rightarrow j] = \begin{cases} L[i \rightarrow j] & \text{for } i = r \\ L[i \rightarrow j] \sum_{m \in I_i} W[m \rightarrow i] & \text{otherwise} \end{cases}$$


$$W[4 \rightarrow 5] = L[4 \rightarrow 5](W[3 \rightarrow 4] + W[2 \rightarrow 4]) = 1.0(1.0 + 0.37) = 1.37$$

# ✍️ 確率的配分の例 – Dialのアルゴリズム

## • Dialのアルゴリズム( $\theta = 1$ )



NW図

OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

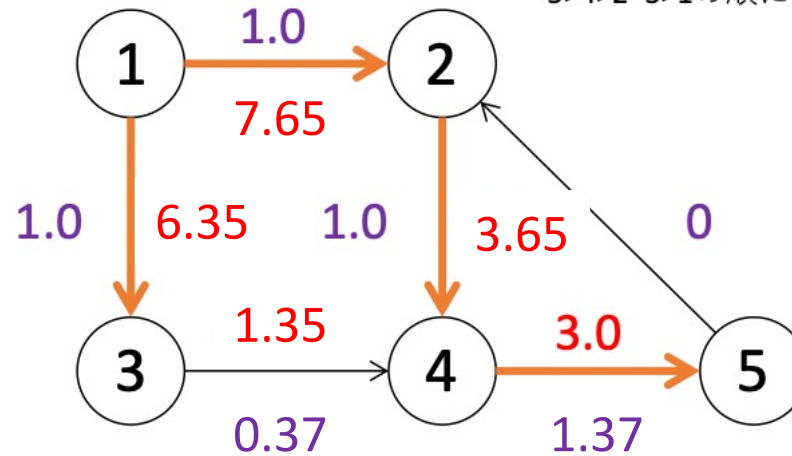
### 後進処理

Step 4

$c(i)$  の値の降順にノードを考え、各ノード  $j$  に流入するリンクの交通量  $x_{ij}$  を計算

$$x_{ij} = \left( q_{rj} + \sum_{m \in O_j} x_{jm} \right) \frac{W[i \rightarrow j]}{\sum_{m \in I_i} W[m \rightarrow j]}$$

5>4>2=3>1の順に考える



計算例

$$x_{45} = \left( q_{15} + \sum_{m \in O_j} x_{5m} \right) \frac{W[4 \rightarrow 5]}{\sum_{m \in I_i} W[m \rightarrow 5]}$$

$$= (q_{15} + 0) \frac{W[4 \rightarrow 5]}{W[4 \rightarrow 5]} = 3.0$$

$$x_{24} = (q_{14} + x_{45}) \frac{W[2 \rightarrow 4]}{W[2 \rightarrow 4] + W[3 \rightarrow 4]} = 3.65$$



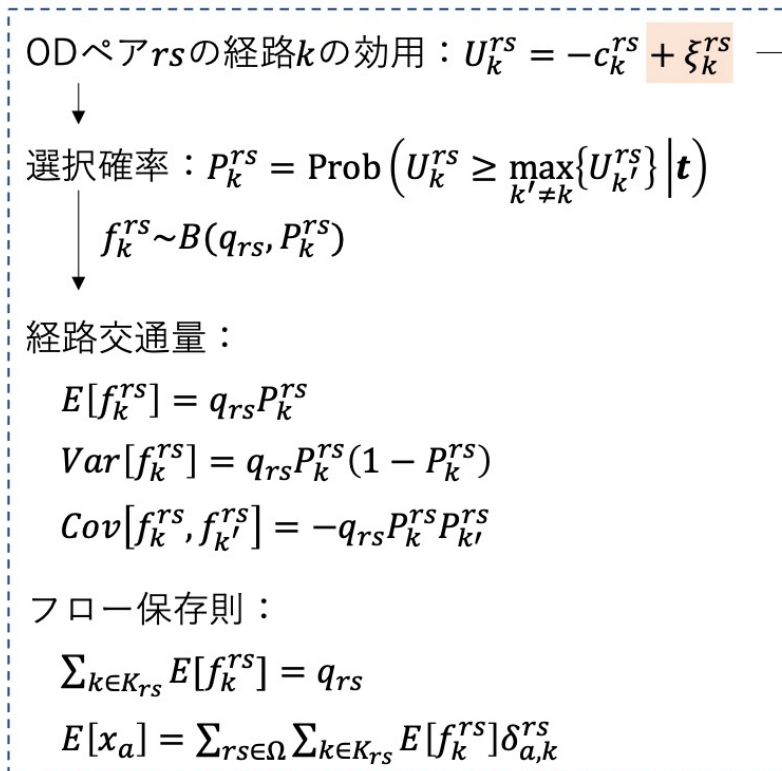
# 確率的利用者均衡配分

	非混雜型 flow-independent	混雜型 flow-dependent
確定的經路 選擇	<b>all-or-nothing配分</b>	<b>利用者均衡配分(UE)</b>
確率的經路 選擇	<b>確率的配分</b>	<b>確率的利用者均衡配分 (SUE)</b>

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

## ■ 確率的利用者均衡(SUE: Stochastic User Equilibrium assignment)

どの利用者也経路を変更することによって自己の旅行時間をそれ以上短縮することができないと 信じている 均衡状態。



$\xi_k^{rs}$  : 誤差項(r.v.)

ex) *i.i.d.*ガンベル分布 → **ロジットモデル**

$$E[f_k^{rs}] = q_{rs} \frac{\exp(-\theta c_k^{rs})}{\sum_{k' \in K_{rs}} \exp(-\theta c_{k'}^{rs})} \quad (*)$$

$\theta \rightarrow \infty$  : 確定的,  $\theta = 0$  : ランダム

$K_{rs}$ の設定：

- Simple path(同一リンクを二度以上通過しない)集合
- Simple pathからサンプリング
- 経路を限定しない

▶ UE同様、最適化問題に置換して解く

\*以降、期待値記号を省略する

	非混雑型 flow-independent	混雑型 flow-dependent
確定的経路 選択	<b>all-or-nothing配分</b>	<b>利用者均衡配分(UE)</b>
確率的経路 選択	<b>確率的配分</b>	<b>確率的利用者均衡配分(SUE)</b>

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

## ■ ロジット型SUEの等価最適化問題

SUE/FD-path

$$\min Z(\mathbf{f}) = \sum_{a \in A} \int_0^{x_a} t_a(w) dw - \frac{1}{\theta} \sum_{rs \in \Omega} q_{rs} H_{rs}(\mathbf{f}^{rs})$$

エントロピー項

→  $\theta \rightarrow \infty$ で消えてUEと一致

この項の由来は演習課題で調べてみてください！

s.t.

$$\sum_{k \in K_{rs}} f_k^{rs} - q_{rs} = 0 \quad \forall rs \in \Omega$$

$$x_a = \sum_{rs \in \Omega} \sum_{k \in K_{rs}} f_k^{rs} \delta_{a,k}^{rs} \quad \forall a \in A$$

$$f_k^{rs} \geq 0 \quad \forall k \in K_{rs}, \forall rs \in \Omega$$

$$H_{rs}(\mathbf{f}^{rs}) \equiv - \sum_{k \in K_{rs}} p_k^{rs} \ln p_k^{rs} = - \sum_{rs \in \Omega} \sum_{k \in K_{rs}} \frac{f_k^{rs}}{q_{rs}} \ln \frac{f_k^{rs}}{q_{rs}}$$

$t_a(x_a)$ : リンク  $a$  の旅行時間  
 $x_a$ : リンク  $a$  の流量  
 $f_k^{rs}$ : ODペア  $rs$  間のパス  $k$  の流量  
 $q_{rs}$ : ODペア  $rs$  間の分布交通量  
 $\delta_{a,k}^{rs}$ : ODペア  $rs$  間のパス  $k$  がリンク  $a$  を含むか否か。(TRUE=1, FALSE=0)  
 $c_k^{rs}$ : ODペア  $rs$  のパス  $k$  の旅行時間

十分性の証明:

SUE/FD-pathのKKT条件が前項のロジットの選択式(\*)と一致する.

解の一意性:

実行可能領域が凸 (:: 制約条件式がすべて線形)

目的関数が狭義凸 (:: リンクコスト関数が単調増加 + エントロピー関数が狭義凸)

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

## ■ 均衡配分の解法

SUEの等価最適化問題の解法

### 逐次平均法 ステップサイズの探索をしない

- ◎各種経路選択モデルに対応
- ×収束が非常に遅い

### 部分線形化法

- ◎収束が非常に早い
- ・起点別リンク交通量を未知変数
- ・ロジット型SUEのみに対応

### Simplicial Decomposition法

- ◎収束が非常に早い
- ・限定された経路交通量を未知変数
- ・ロジット型SUEのみに対応

## ■ 確率的配分の解法

Flow-independentな場合の配分手法

### ロジット型確率的配分

#### Dialのアルゴリズム

- ◎経路列挙不要
- ×経路を限定

#### Markov連鎖配分

- ◎経路列挙不要
- ◎経路限定不要
- ×過大なcyclic flow
- ×IIA特性を増幅

### プロビット型確率的配分

#### モンテカルロシミュレーション

計算過程でFlow-independentな場合の配分計算(確率的配分)が必要



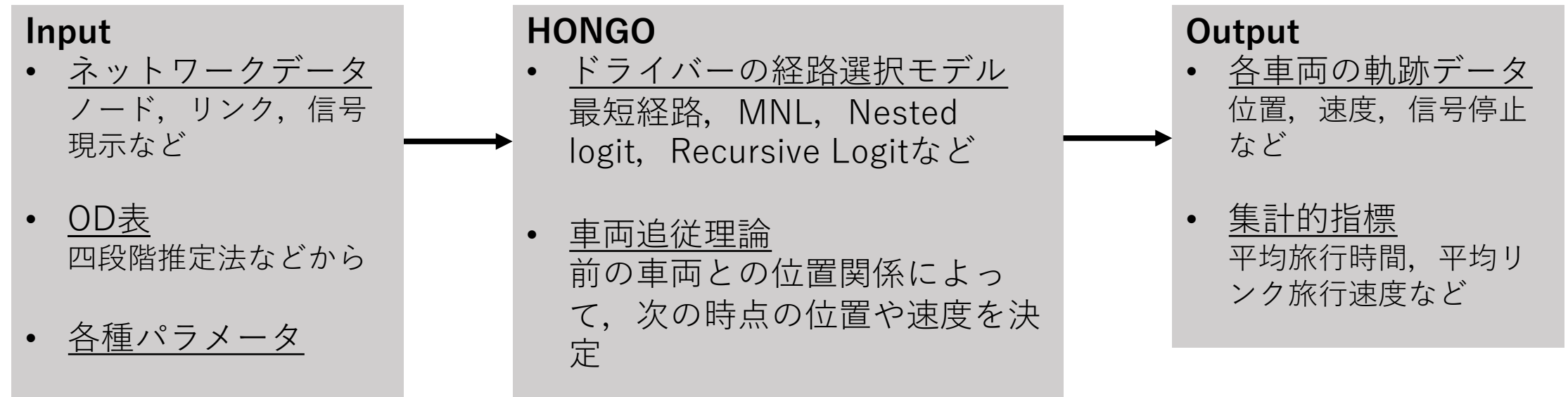
# そのほかの配分

# マイクロシミュレーション

- 車両（エージェント）をプログラム上で発生させ、各エージェントの行動ルールをモデル化し、それらの行動の集積として交通現象を再現する。

MATSim, SUMO, HONGO etc...

例) HONGO



# マイクロシミュレーション

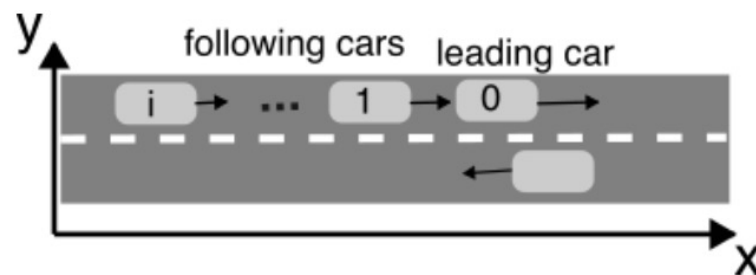
- 車両追従理論の例 ( BinN共有ドライブ>seminar> BinNStartupSeminarEnglish2022>#04 小川回資料より)

GM [Gazis, Herman & Rothery, 1961]

13ページに移動

$$\dot{v}_i(t) = \alpha \frac{v_i^\beta(t)}{\Delta x_i^\gamma(t - \Delta t)} \Delta v_i(t - \Delta t)$$

- $v_i(t)$  : velocity of vehicle  $i$  at time  $t$
- $\Delta x_i(t)$  : headway to the forward vehicle
- $\Delta v_i(t)$  : relative speed to the forward vehicle
- $\Delta t$  : delay time
- $\alpha, \beta, \gamma$  : parameters



Flexible and fit well to many measurement results



# MFDを用いたゾーンベース配分

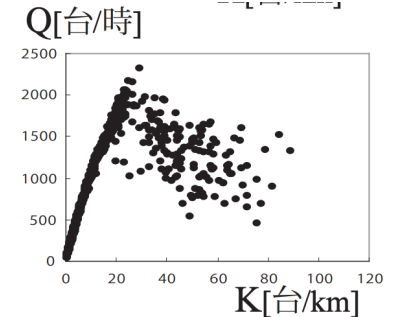
- これまで説明してきた交通量配分は道路リンクを最小単位とする
- 一方、交通状態を巨視的に捉え、ゾーンの単位で交通制御を行う研究が勃興

## → Macroscopic Fundamental Diagram (MFD)

= あるネットワーク内 (≠リンク) の車両密度と平均流量の関係 (集計的・巨視的)

※ 平均流量や平均密度は検知器で観測可能

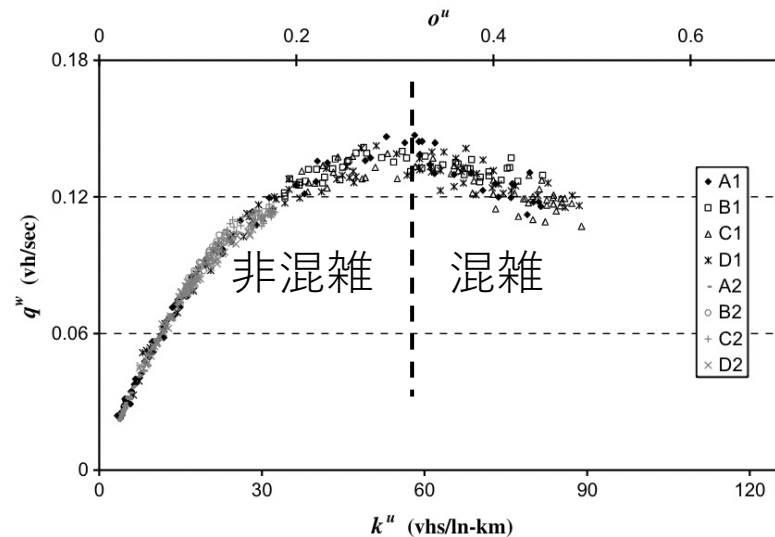
cf.) Fundamental Diagram



**Production** [台・km/h]

$$= \frac{\text{平均流量}}{\text{NW長}}$$

※単位時間あたりの移動総量



**Accumulation** [台]

$$= \frac{\text{平均密度}}{\text{NW長}} \times \text{NW内の車両台数}$$

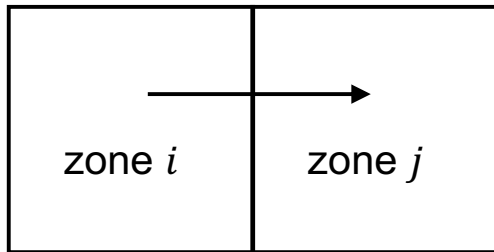
車両台数が少ない時は、車両の増加に従って総流量も増えるが、車両台数がある値を超えると混雑が発生し始め、総流量は低下する。

# MFDとは

- Production（単位時間あたり総移動量）の代わりに、トリップ完了率やoutflow（流出台数）も使われる
  - トリップ完了率 [台/h] =  $(\text{Production}) / (\text{平均トリップ長})$   
※単位時間あたりにトリップが終了する台数（NWからの流出とNW内でのトリップ終了を含む）
  - **Outflow** [台/h] = NWの境界部での流出台数 → Productionに比例  
※ゾーンからの流出によってのみトリップが完了するとした場合、トリップ完了率に相当
- homogeneous（混雑が空間的に偏在していない）なNWである必要  
→ 混雑が特定の地域に集中する状況では、MFDのばらつきが大きくなったり存在しない場合も
- MFDの形は需要パターンによらない（制御にOD表が必要ない）  
→ 観測とNWへの流入の制御を繰り返すだけで渋滞を制御できる（perimeter control）
- 信号、交通情報提供、流入制御等の政策を最適化・リアルタイム制御するために、計算負荷の小さい集計的なモデルとして利用されている。

# MFDを用いたゾーンベース配分

- 詳細なリンクレベルのシミュレーションを実NWに適用するのは計算が大変
  - リンクでなく，ゾーンの系列を選択するモデルで簡潔にできないか = **ゾーンベースの経路選択**
  - **ゾーン間移動**について「**交通容量**（単位時間あたり何台の車が移動できるか）」の概念が必要
  - **MFD**を利用



フロー保存則

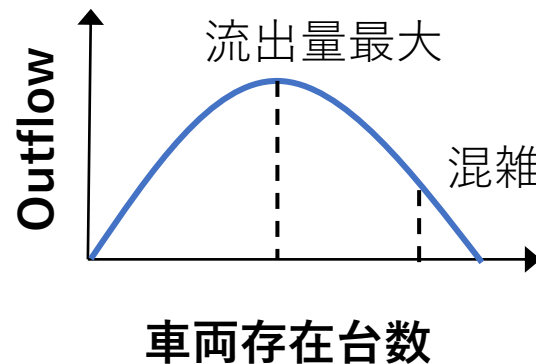
$$n_i(t+1) = n_i(t) - n_{i,out}(t) + n_{i,in}(t)$$

zone  $i$  の時間  $t+1$  での車両台数

ゾーン  $i$  から  
流出する車両数

ゾーン  $i$  へ  
流入する車両数

**MFDにより規定される**



※ MFDの形はマイクロシミュレーションや実測データを用いて推定する

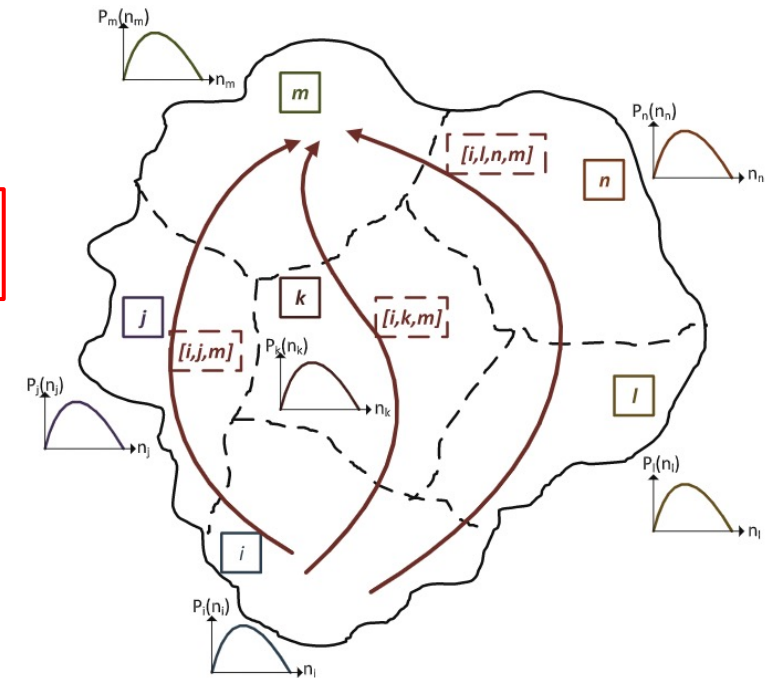


Fig. 1. A multi-region urban network.

# MFDを用いたゾーンベース配分

- Regional pathを扱った既往研究

	内容	ゾーン経路選択の仮定
Haddad et al. (2013)	モデル予測制御でゾーン境界と高速道路入り口で流入制御	旅行時間最小の経路
Yildirimoglu et al. (2014) Batista et al. (2019)	MFDを動的交通量配分に導入	動的確率的利用者均衡 (DSUE)
Knoop et al. (2015)	シミュレーションアプローチ	最短経路
Kim et al. (2018)	シミュレーションアプローチ	MNL
MFD-RL	シミュレーションアプローチ	Recursive logit

- 最適化やリアルタイムの制御に使うための**高速化**の位置付け  
<課題>
- HomogeneousなNWへの動的な分割方法
- 同じゾーン間の移動でも実際のトリップ長の構成はさまざま考えられる  
→ regional pathごとの各ゾーン内のトリップ長の分布



# 参考文献

土木学会. (1998). 交通ネットワークの均衡分析-最新の理論と解法.

Sheffi, Y. (1985). Urban transportation networks (Vol. 6). *Prentice-Hall, Englewood Cliffs, NJ*.

Wardrop .J. G. (1952). Some theoretical aspects of road traffic research. *Proceedings of the institute of Civil Engineers Part II*, pp. 325-378.

Daganzo, C. F., & Sheffi, Y. (1977). On stochastic models of traffic assignment. *Transportation science*, 11(3), 253-274.

Fisk, C. (1980). Some developments in equilibrium traffic assignment. *Transportation Research Part B: Methodological*, 14(3), 243-255.

赤松隆. (1996). 交通流の予測・誘導・制御と動的なネットワーク配分理論. *土木計画学研究・論文集*, 13, 23-48.

Beckmann. M., C. B. McGuire & C. B. Winsten (1956). *Studies in the Economics of Transportation*. *Yale University Press, New Haven, CT*.  
<https://cowles.yale.edu/sites/default/files/files/pub/misc/specpub-beckmann-mcguire-winsten.pdf>

Dial. R. B. (1971). A Probabilistic Multipath Traffic Assignment Algorithm Which Obviates Path Enumeration. *Transportation Research* 5, pp.83-111.

Bell, M. G., & Cassir, C. (2002). Risk-averse user equilibrium traffic assignment: an application of game theory. *Transportation Research Part B: Methodological*, 36(8), 671-681.

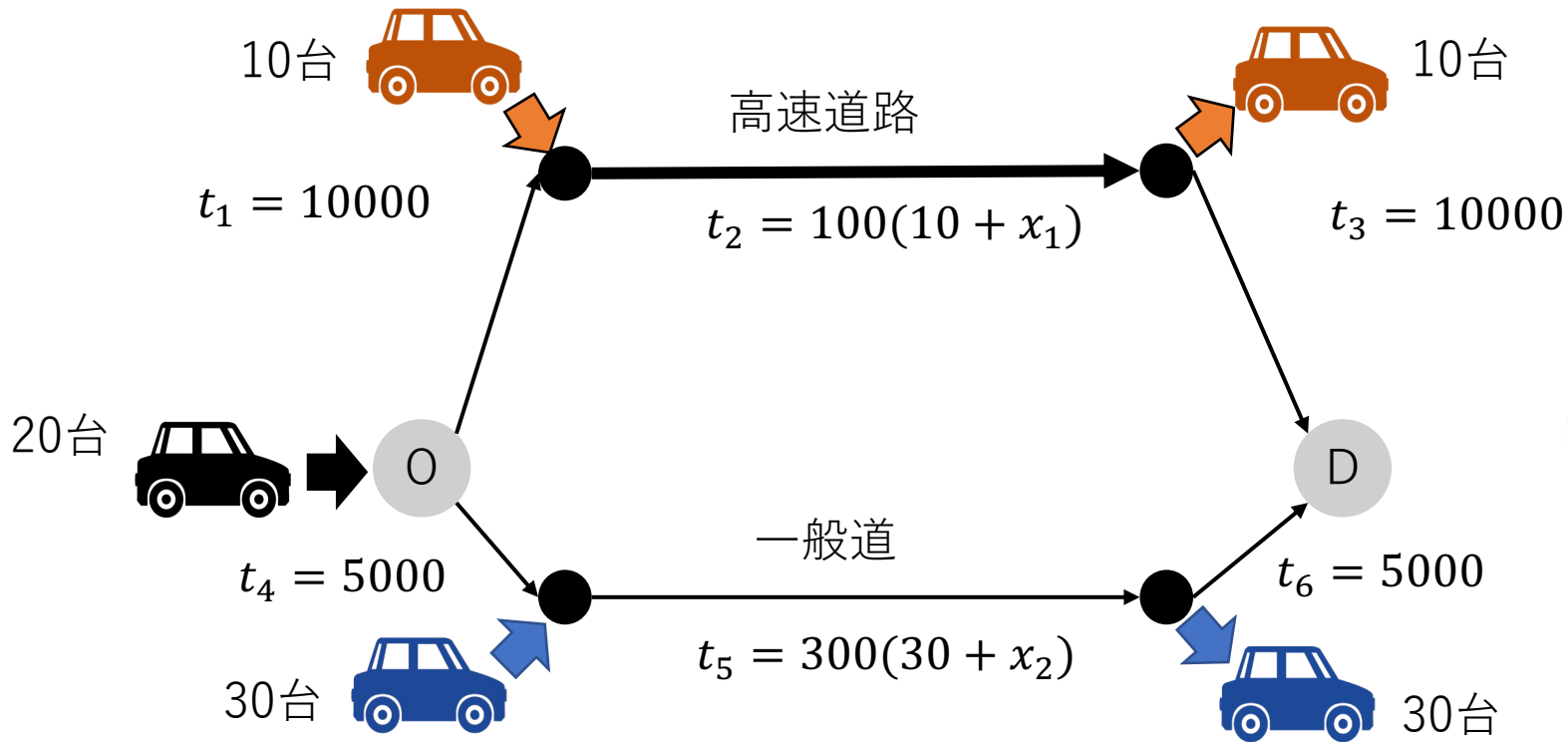
# 演習課題

必須課題2つ + 任意課題を1つ以上

# 必須課題 1

(1) 以下のネットワークにおいて、利用者均衡を求めるコードを書いてください

- Jupyter notebookファイルにまとめたので、Google colabでぽちぽち回しながら確認してください
- 言語はPythonですが、他言語に直してもらってもOK



$t$ : リンクコスト  
 $x$ : 経路交通量

ヒント：総所要時間は、  
 $t_1 \cdot x_1 + t_2 \cdot (10 + x_1) + t_3 \cdot x_1 +$   
 $t_4 \cdot x_2 + t_5 \cdot (30 + x_2) + t_6 \cdot x_2$



## 必須課題 2

(2) (1)のコードが書けたら、以下のことについて考察してください

- 今、あなたは地域の道路計画を考えています.
- リンク4とリンク6の所要時間を信号などの交通制御によって、 $1500 \leq t_4, t_6 \leq 7500$ の間で操作できるとします. (直感的には所要時間を最小値1500にするのが良さそう. . . )
- 制御変数を $L = t_4 = t_6$ とし、 $L$ を変化させながら均衡配分を行い、 $L$ と総所要時間 ( $\sum_a x_a t_a$ ) の関係を図示して考察してください. 総所要時間を最小にする $L$ はいくつでしょうか.

# 任意課題 1

実世界の都市と交通の問題を配分計算で考えてみましょう。  
問題に適した簡単なネットワークを作り、均衡配分の計算をして、結果を考察してください。

視点：

- リンクパフォーマンス関数（自由旅行速度や道路容量の改善）の操作
- どのリンクを拡張・追加すると総旅行時間が最小になるか（ネットワーク設計問題）
- 災害時に切断されると一番影響が大きいリンクはどこか

# 任意課題 2

都市と交通にまつわる現象・人々の選択で、均衡に達しているものと、均衡状態にないものをそれぞれできるだけ多く挙げてください。

また、そのときの選択行動と、人々の間に働いている相互作用を書いてください。

(ヒント)

経路選択ゲームの序盤：均衡状態に達していない

経路選択ゲームの終盤：均衡状態に達した

(例) 均衡

日々の車通勤の経路選択. 選択行動：どの道路を通るか. 相互作用：道路混雑

# 任意課題 3

SUEのエントロピー項について、その由来・解釈を調べて、まとめてください。

また、スライドp60のSUE/FD-pathがロジット配分と等価であることを証明してください。

(ヒント)

KKT条件がロジットの選択式と一致することを示せばいいです。

SUE/FD-path

エントロピー項

$$\min Z(f) = \sum_{a \in A} \int_0^{x_a} t_a(w) dw - \frac{1}{\theta} \sum_{rs \in \Omega} q_{rs} H_{rs}(f^{rs})$$

s.t.

$$\sum_{k \in K_{rs}} f_k^{rs} - q_{rs} = 0 \quad \forall rs \in \Omega$$
$$x_a = \sum_{rs \in \Omega} \sum_{k \in K_{rs}} f_k^{rs} \delta_{a,k}^{rs} \quad \forall a \in A$$
$$f_k^{rs} \geq 0 \quad \forall k \in K_{rs}, \forall rs \in \Omega$$
$$H_{rs}(f^{rs}) \equiv - \sum_{k \in K_{rs}} P_k^{rs} \ln P_k^{rs} = - \sum_{rs \in \Omega} \sum_{k \in K_{rs}} \frac{f_k^{rs}}{q_{rs}} \ln \frac{f_k^{rs}}{q_{rs}}$$

ロジットの選択式

$$E[f_k^{rs}] = q_{rs} \frac{\exp(-\theta c_k^{rs})}{\sum_{k' \in K_{rs}} \exp(-\theta c_{k'}^{rs})} \quad (*)$$

$\theta \rightarrow \infty$  : 確定的,  $\theta = 0$  : ランダム

# 任意課題 4

必須課題1のネットワーク（か、もう少し複雑なネットワーク）において、人々が何らかの進化動学に従って行動を変化させたとするとき、各経路を選択する人数や旅行時間はどのように変化していきますか。

好きな進化動学を実装して、確認してください。

# 利用者均衡配分とNash均衡の等価性

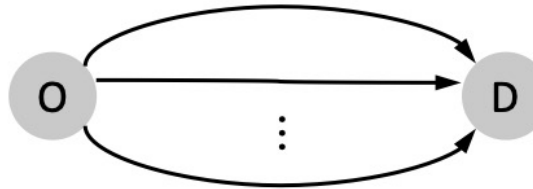
# 補足① 利用者均衡とNash均衡の等価性の証明 2021年度夏学期ゼミ#4より

Prop.1

n人による決定論的利用者均衡は、nが大のとき、  
n人非協力ゲームの混合戦略ナッシュ均衡と等価である。

条件

- 単一ODペア
- 複数経路
- n人の同質な利用者



$h_j$  : 経路  $j$  の交通量  
 $\mathbf{h}$  : 経路交通量ベクトル  
 $g_j$  : 経路  $j$  のコスト

前提：利用者均衡の必要十分条件（マクロ）

$$h_j = 0 \Leftrightarrow g_j(\mathbf{h}) > \min_k g_k(\mathbf{h}) \text{ for all paths } j,$$

: 最小コストでない経路は選択されない（交通量0）

$$h_j > 0 \Rightarrow g_j(\mathbf{h}) = \min_k g_k(\mathbf{h}) = g_{OD}(\mathbf{h}).$$

: 利用されている経路のコストは最小

これがn人非協力ゲームの混合戦略ナッシュ均衡と等価であることを示す

n人非協力ゲームとして考える。(ミクロ)

利用者aはどの経路を選択するか(戦略)を確率的に選ぶ(混合戦略)

利用者aの混合戦略  $s_a = \sum_j \pi_{aj} p_{aj}$

利用者aが経路jを選択する純粋戦略

利用者aが経路jを選択する確率

純粋戦略：一つの選択肢を確定的に選ぶ戦略

混合戦略：確率的に選択肢を選ぶ戦略

利用者aの被るコストの期待値

利用者a以外の利用者の混合戦略ベクトル  $c_a(s) = \sum_j p_{aj} c_a(\underline{s}_{-a}, \pi_{aj})$

他の利用者の戦略を所与としたとき、利用者aが経路jを選択して被るコスト



# 補足① 利用者均衡とNash均衡の等価性の証明 2021年度夏学期ゼミ#4より

- 利用者は同質と仮定→認知コストは全員同じ
- 全員同じ選択肢集合
- 全員コストを最小化しようとする行動する



利用者は入れ替え可能

$$p_{1j} = p_{2j} = \dots = p_{nj} = p_j$$

$$c_1(\mathbf{s}_{-1}, \pi_{1j}) = c_2(\mathbf{s}_{-2}, \pi_{2j}) = \dots = c_n(\mathbf{s}_{-n}, \pi_{nj}).$$

**nが大ならば, 大数の法則により  $h \cong pn$**

$$\underline{c_a(\mathbf{s}_{-a}, \pi_{aj})} \cong \underline{g_j(\mathbf{h})}$$

利用者aが経路jを  
選ぶときのコスト

交通量に基づく  
経路jのコスト

ミクロ

マクロ

利用者 a が経路 j を選んだ時のコストは他の人の混合戦略（どの経路にどれくらいの確率で行くか）によるが、各利用者の選択確率が等しいので、混合戦略は全員同じ。nが大ならば各利用者の混合戦略の結果として、確率に基づく交通量が出てくる（大数の法則）。

よって、nが大ならば、他の人の戦略にもとづいて利用者 a が経路 j を選ぶコストは、交通量（他の人の戦略が実現した結果）に基づくコストとほぼ同じ。

## 補足① 利用者均衡とNash均衡の等価性の証明 2021年度夏学期ゼミ#4より

利用者均衡の必要十分条件  $\begin{cases} g_j(\mathbf{h}) > g_{OD}(\mathbf{h}) \Rightarrow p_j = 0 \\ g_j(\mathbf{h}) = g_{OD}(\mathbf{h}) \Leftarrow p_j > 0 \end{cases}$  を書き直すと

$$\begin{cases} p_{aj} = 0 \Leftarrow c_a(\mathbf{s}_{-a}, \pi_{aj}) > \min_k c_a(\mathbf{s}_{-a}, \pi_{ak}) \quad \forall a \text{ and } \forall j \\ \quad : \text{利用者}a\text{が最小コストでない経路}j\text{を選択する確率はゼロ} \\ p_{aj} > 0 \Rightarrow c_a(\mathbf{s}_{-a}, \pi_{aj}) = \min_k c_a(\mathbf{s}_{-a}, \pi_{ak}) \quad \forall a \text{ and } \forall j. \\ \quad : \text{利用者}a\text{の経路}j\text{の選択確率が正ならばコストは最小} \end{cases}$$

これは混合戦略ナッシュ均衡の必要十分条件である。

### ※ナッシュ均衡

自分以外のプレイヤーの戦略を所与とした場合、どのプレイヤーも自分の戦略を変更することによってより高い利得（より低いコスト）を得ることができない戦略の組み合わせのこと

# 一次元探索の工夫 – 黄金分割法

## 補足② 黄金分割法の詳細 2021年度夏学期ゼミ#3より

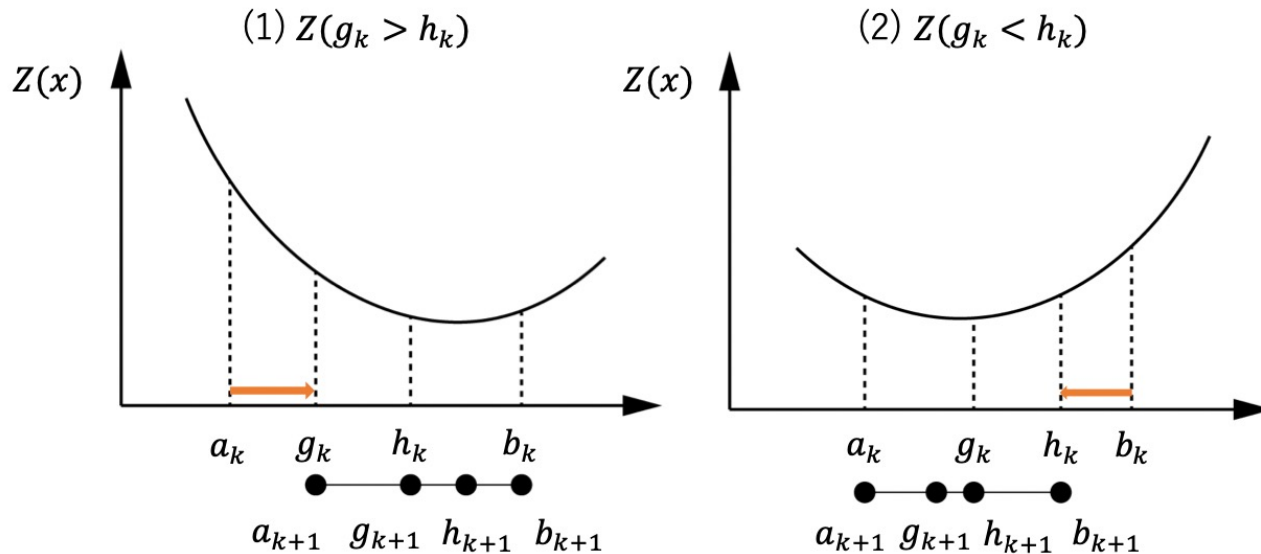
### Frank-Wolfe法のステップサイズを効率よく探索するための方法

モチベーション：

一変数の凸関数  $Z(x)$  の区間  $[a, b]$  における最小値を求めたい。

概要：

最小点を含む  $k$  回目の閉区間  $[a_k, b_k]$  から、縮小率  $s$  だけ小さく、かつ最小値が存在する次の閉区間  $[a_{k+1}, b_{k+1}]$  を順次列挙して最小点を探索する。



内点  $g_k, h_k$  を  
取りながら探索する

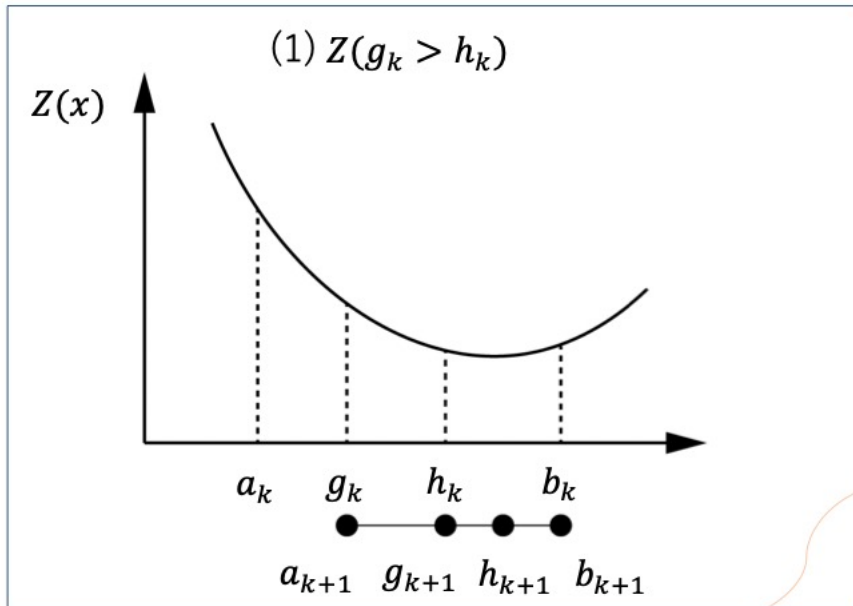
最小値は  
(1) のとき  $[g_k, b_k]$   
(2) のとき  $[h_k, a_k]$   
にあることが分かる。

## 補足② 黄金分割法の詳細 2021年度夏学期ゼミ#3より

### 一次元探索の手法：黄金分割法

この資料もわかりやすい

<http://www-optima.amp.i.kyoto-u.ac.jp/~nobuo/Ryukoku/2002/course6.pdf>



内点  $g_k, h_k$  を,  
 $b_k - g_k = h_k - a_k = s(b_k - a_k)$   
 を満たすようにとると,

$$\textcircled{1} g_k = b_k - s(b_k - a_k)$$

$$\textcircled{2} h_k = a_k + s(b_k - a_k)$$

(1)  $Z(g_k) > Z(h_k)$  の場合,  $k+1$  回目の点を左図のようにとれる.

$$\textcircled{3} a_{k+1} = g_k, g_{k+1} = h_k, b_{k+1} = b_k$$

※  $k$  回目の3点を  $k+1$  回目で利用可能

$g_{k+1} = b_{k+1} - s(b_{k+1} - a_{k+1})$  に③を代入すると,

$$\textcircled{4} h_k = b_k - s(b_k - g_k)$$

④に①, ②を代入すると,

$$a_k + s(b_k - a_k) = b_k - s * s(b_k - a_k)$$

$$s^2(b_k - a_k) + s(b_k - a_k) - (b_k - a_k) = 0$$

$$\textcircled{5} s^2 + s - 1 = 0 (\because a_k \neq b_k)$$

⑤を解くと,  $s = \frac{\sqrt{5}-1}{2} = 0.618 \dots$

c.f. 黄金数  $\frac{\sqrt{5}+1}{2} = \frac{1}{s}$

# All-or-nothing配分 – 効率の良い方法

# ✍️ all-or-nothing配分 – 効率の良い方法

## ■ all-or-nothing配分

- flow-independentな場合の配分アルゴリズム
- 最短経路がツリー構造になっていることを利用

Step 1  $n = 0$ とする。全てのリンク交通量について $x_{ij}^{(0)} = 0$ とする。

Step 2  $n$ 番目の起点を $o$ とする。起点 $o$ から他の全てのノードへ最短経路探索を行い、起点 $o$ から他の全てのノードへの最短経路費用 $\{C_{\min}[o \rightarrow i]\}$ と各ノードに対する先行ポイント $F_i$ を求める。  $y_{ij} = 0$ とする。

Step 3  $\{C_{\min}[o \rightarrow i]\}$ の降順 ( $o$ から遠い順) にノード $j$ を考える。ノード $j$ に流入し、かつ最短経路ツリーに含まれるリンク $i (= F_j) \rightarrow j$ の交通量 $x_{ij}$ を以下で改訂する。

$$y_{ij} = q_{oj} + \sum_{m \in O_j} y_{jm}$$
$$x_{ij}^{(n)} = x_{ij}^{(n-1)} + y_{ij}$$

$O_j$ はノード $j$ から流出するリンクの終点集合、 $q_{oj}$ は起点 $o$ からノード $j$ までの分布交通量。

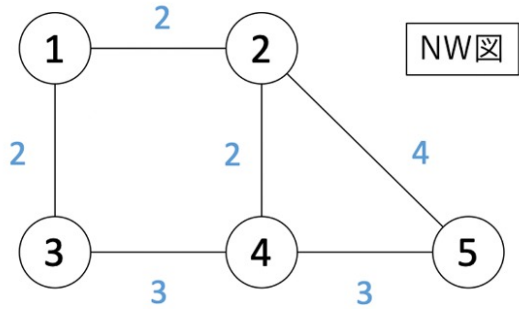
Step 4  $n = N$  (ノード数) なら終了。そうでなければ $n = n + 1$ としてStep 2へ。

# 👉 all-or-nothing配分 – 効率の良い方法

p15の例でいうと、

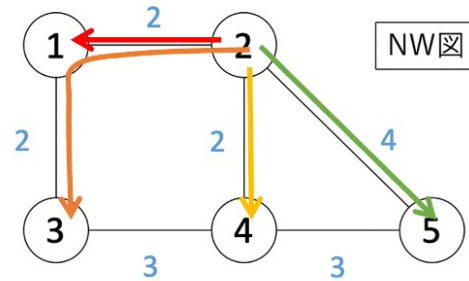
## 👉 all-or-nothing配分 – 単純な方法

OD交通量を全ての経路に流して、最後にリンクごとに集計する方法



NW図

起点2に対する計算例



NW図

OD表 値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

リンク	流量
1-2	9
1-3	5+4=9
2-4	2+1=3
3-4	0
2-5	3+6=9
4-5	0
2-1	0+3+4=7
3-1	0
4-2	0
4-3	0
5-2	0
5-4	0

接続する道路に流出する交通量

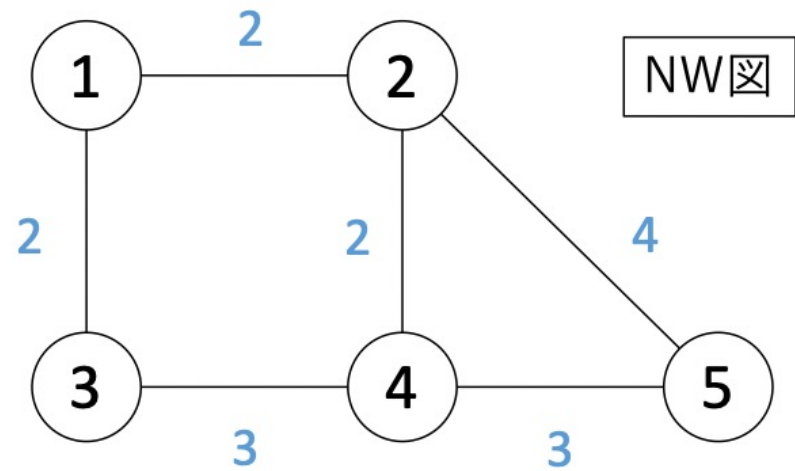
$$y_{ij} = q_{oj} + \sum_{m \in O_j} y_{jm}$$

$$x_{ij}^{(n)} = x_{ij}^{(n-1)} + y_{ij}$$

(n-1)回目までの交通量



# 👉 all-or-nothing配分 – 効率の良い方法



NW図

OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

## 起点1に対する計算例

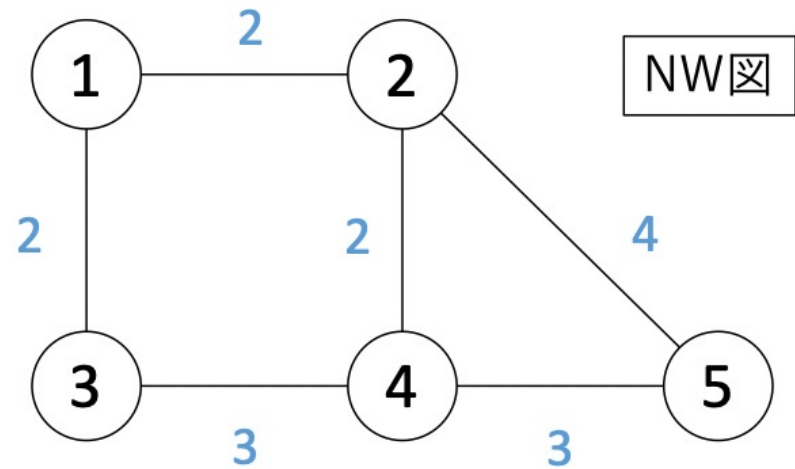
(Step 1) 全  $x_{ij}$  について  $x_{ij} = 0$  とする

(Step 2) Dijkstra法での最短経路探索( $o = 1$ )

(Step 3)  $C_{min}[o \rightarrow j]$  の降順で  $x_{ij}$  を更新

リンク	流量
1-2	0
1-3	0
2-4	0
3-4	0
2-5	0
4-5	0
2-1	0
3-1	0
4-2	0
4-3	0
5-2	0
5-4	0

# 👉 all-or-nothing配分 – 効率の良い方法



NW図

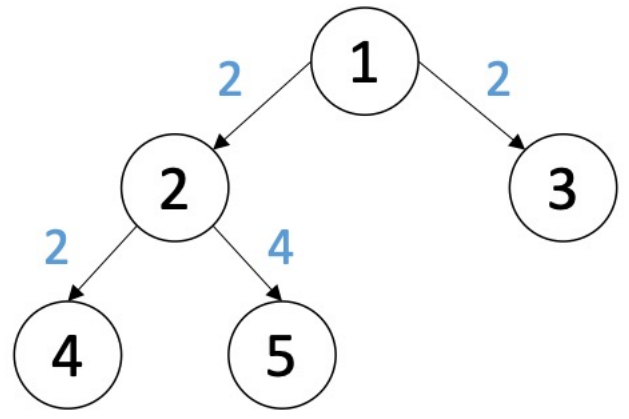
OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

## 起点1に対する計算例

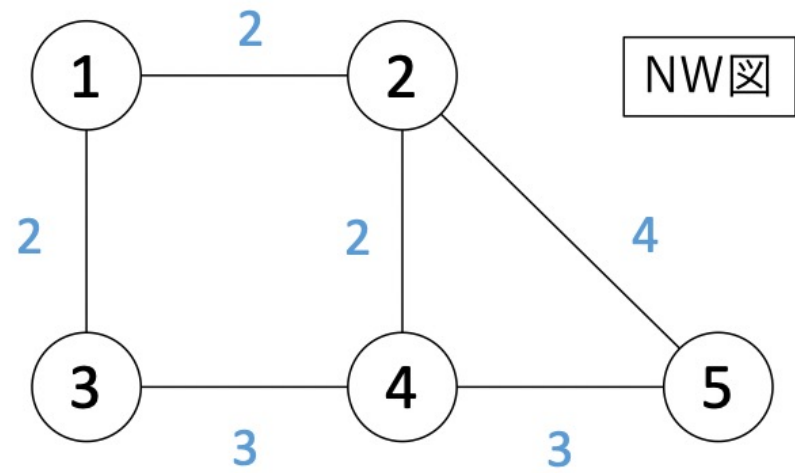
- (Step 1) 全  $x_{ij}$  について  $x_{ij} = 0$  とする
- (Step 2) Dijkstra法での最短経路探索( $o = 1$ )



(Step 3)  $C_{min}[o \rightarrow j]$  の降順で  $x_{ij}$  を更新

リンク	流量
1-2	0
1-3	0
2-4	0
3-4	0
2-5	0
4-5	0
2-1	0
3-1	0
4-2	0
4-3	0
5-2	0
5-4	0

# 👉 all-or-nothing配分 – 効率の良い方法



NW図

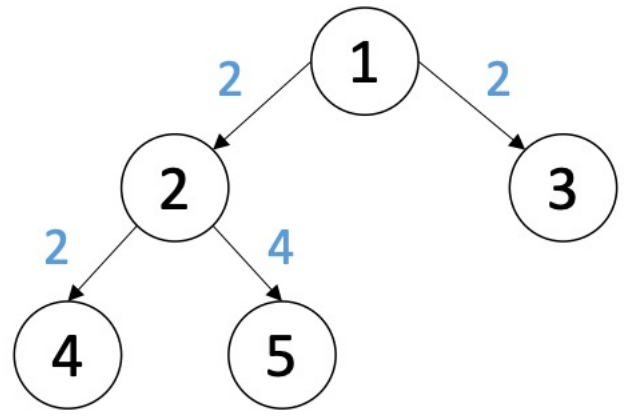
OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

## 起点1に対する計算例

- (Step 1) 全  $x_{ij}$  について  $x_{ij} = 0$  とする
- (Step 2) Dijkstra法での最短経路探索( $o = 1$ )



(Step 3)  $C_{min}[o \rightarrow j]$  の降順で  $x_{ij}$  を更新

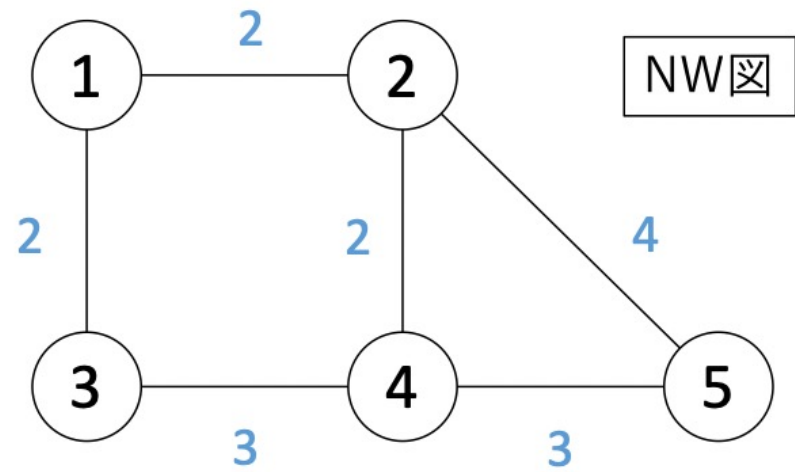
$$y_{ij} = q_{oj} + \sum_{m \in O_j} y_{jm}$$

$$x_{ij} = x_{ij} + y_{ij}$$

$O_j$ : ノード  $j$  から流出するリンクの終点集合  
 $q_{oj}$ : 起点  $o$  からノード  $j$  までの分布交通量.

リンク	流量
1-2	$0 + (4 + 3 + 2) = 9$
1-3	$0 + (5 + 0) = 5$
2-4	$0 + (2 + 0) = 2$
3-4	0
2-5	$0 + (3 + 0) = 3$
4-5	0
2-1	0
3-1	0
4-2	0
4-3	0
5-2	0
5-4	0

# 👉 all-or-nothing配分 – 効率の良い方法



NW図

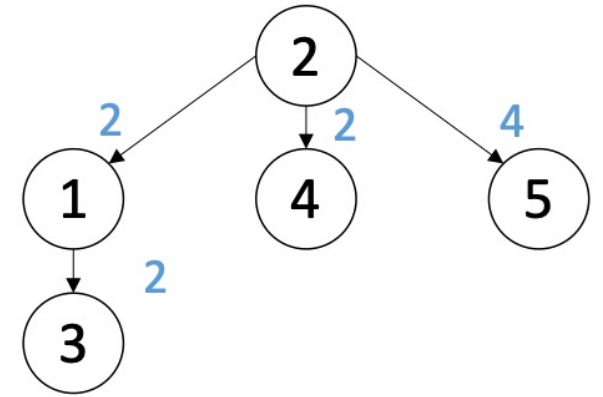
OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

## 起点2に対する計算例

(Step 2) Dijkstra法での最短経路探索( $o = 2$ )



(Step 3)  $C_{min}[o \rightarrow j]$  の降順で  $x_{ij}$  を更新

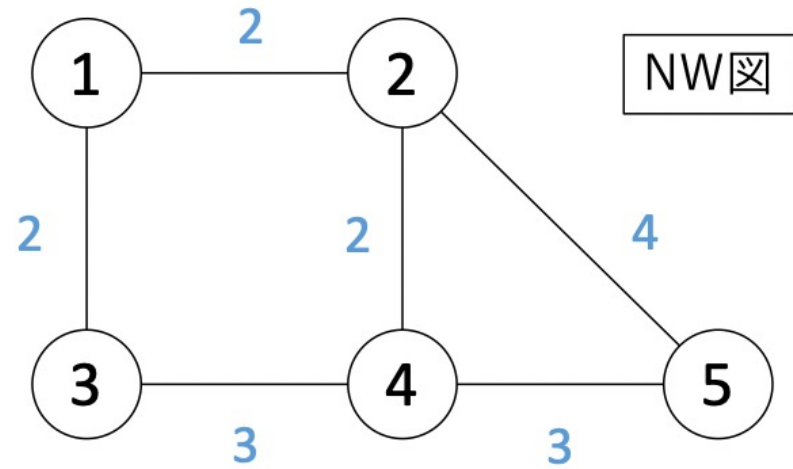
$$y_{ij} = q_{oj} + \sum_{m \in O_j} y_{jm}$$

$$x_{ij} = x_{ij} + y_{ij}$$

$O_j$ : ノード  $j$  から流出するリンクの終点集合  
 $q_{oj}$ : 起点  $o$  からノード  $j$  までの分布交通量.

リンク	流量
1-2	9
1-3	$5 + (4 + 0) = 9$
2-4	$2 + (1 + 0) = 3$
3-4	0
2-5	$3 + (6 + 0) = 9$
4-5	0
2-1	$0 + (3 + 4) = 7$
3-1	0
4-2	0
4-3	0
5-2	0
5-4	0

# 👉 all-or-nothing配分 – 効率の良い方法



NW図

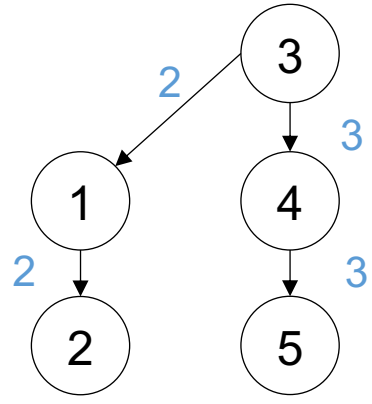
OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

## 起点3に対する計算例

(Step 2) Dijkstra法での最短経路探索( $o = 2$ )



(Step 3)  $C_{min}[o \rightarrow j]$  の降順で  $x_{ij}$  を更新

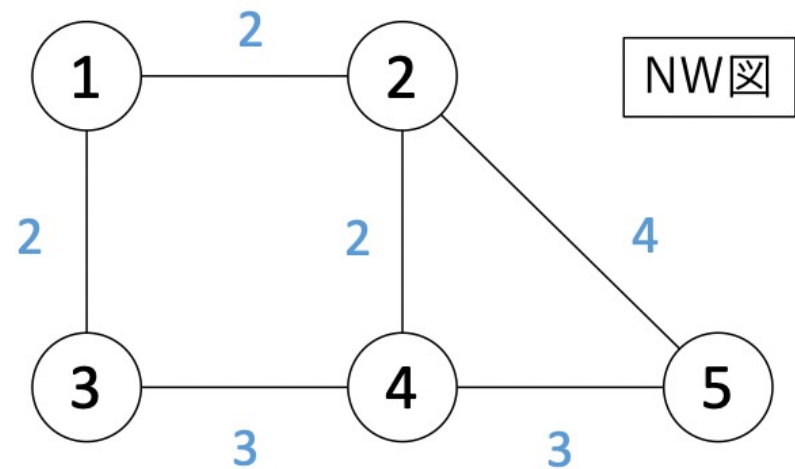
$$y_{ij} = q_{oj} + \sum_{m \in O_j} y_{jm}$$

$$x_{ij} = x_{ij} + y_{ij}$$

$O_j$ : ノード  $j$  から流出するリンクの終点集合  
 $q_{oj}$ : 起点  $o$  からノード  $j$  までの分布交通量.

リンク	流量
1-2	$9 + (1 + 0) = 10$
1-3	9
2-4	3
3-4	$0 + (4 + 2) = 6$
2-5	9
4-5	$0 + (2 + 0) = 2$
2-1	7
3-1	$0 + (3 + 1) = 4$
4-2	0
4-3	0
5-2	0
5-4	0

# 👉 all-or-nothing配分 – 効率の良い方法



NW図

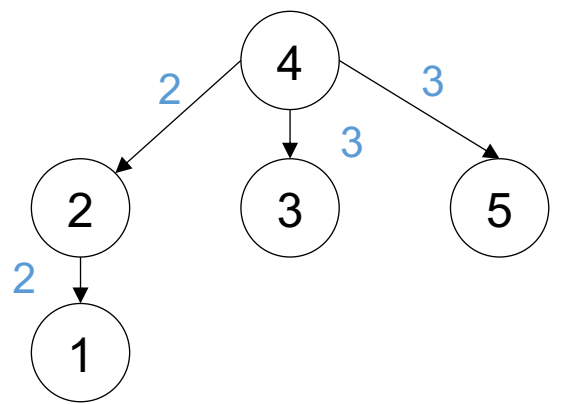
OD表

値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

## 起点4に対する計算例

(Step 2) Dijkstra法での最短経路探索( $o = 2$ )



(Step 3)  $C_{min}[o \rightarrow j]$  の降順で  $x_{ij}$  を更新

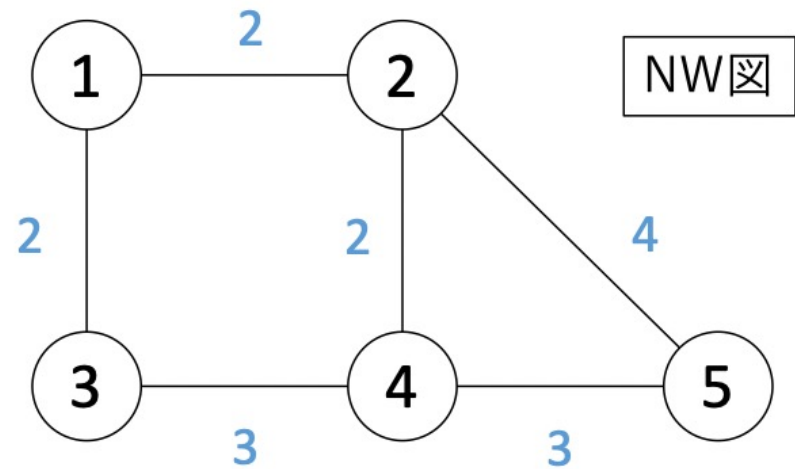
$$y_{ij} = q_{oj} + \sum_{m \in O_j} y_{jm}$$

$$x_{ij} = x_{ij} + y_{ij}$$

$O_j$ : ノード  $j$  から流出するリンクの終点集合  
 $q_{oj}$ : 起点  $o$  からノード  $j$  までの分布交通量.

リンク	流量
1-2	10
1-3	9
2-4	3
3-4	6
2-5	9
4-5	$2 + (2 + 0) = 4$
2-1	$7 + (2 + 0) = 9$
3-1	4
4-2	$0 + (4 + 2) = 6$
4-3	$0 + (2 + 0) = 2$
5-2	0
5-4	0

# 👉 all-or-nothing配分 – 効率の良い方法



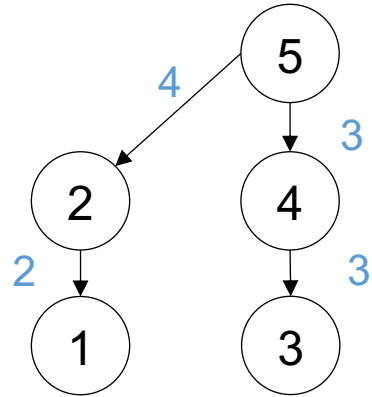
NW図

OD表 値はリンクコストに対応

O/D	n1	n2	n3	n4	n5
n1	0	4	5	2	3
n2	3	0	4	1	6
n3	3	1	0	4	2
n4	2	4	2	0	2
n5	1	4	3	5	0

## 起点5に対する計算例

(Step 2) Dijkstra法での最短経路探索( $o = 2$ )



(Step 3)  $C_{min}[o \rightarrow j]$  の降順で  $x_{ij}$  を更新

$$y_{ij} = q_{oj} + \sum_{m \in O_j} y_{jm}$$

$$x_{ij} = x_{ij} + y_{ij}$$

$O_j$ : ノード  $j$  から流出するリンクの終点集合  
 $q_{oj}$ : 起点  $o$  からノード  $j$  までの分布交通量.

リンク	流量
1-2	10
1-3	9
2-4	3
3-4	6
2-5	9
4-5	4
2-1	$9 + (1 + 0) = 10$
3-1	4
4-2	6
4-3	$2 + (3 + 0) = 5$
5-2	$0 + (4 + 1) = 5$
5-4	$0 + (5 + 3) = 8$

# システム最適配分と混雑課金の余剰分析



# システム最適配分(SO)と混雑課金

## ■ SO (Wardropの第2原則) の定式化

= 道路網上の総旅行時間が最小となる。

### SO-Primal

$$\min Z(\mathbf{x}) = \sum_{a \in A} x_a t_a(x_a)$$

s.t.

$$\sum_{k \in K_{rs}} f_k^{rs} - q_{rs} = 0 \quad \forall rs \in \Omega$$

$$x_a = \sum_{rs \in \Omega} \sum_{k \in K_{rs}} f_k^{rs} \delta_{a,k}^{rs} \quad \forall a \in A$$

$$f_k^{rs} \geq 0 \quad \forall k \in K_{rs}, \forall rs \in \Omega$$

$$x_a \geq 0 \quad \forall a \in A$$

UE/FD-Primalから目的関数が変わるだけ。  
UEと全く同じアルゴリズムで解ける。

$t_a(x_a)$ : リンク  $a$  の旅行時間  
 $x_a$ : リンク  $a$  の交通量  
 $f_k^{rs}$ : ODペア  $rs$  間のパス  $k$  の流量  
 $q_{rs}$ : ODペア  $rs$  間の分布交通量  
 $\delta_{a,k}^{rs}$ : ODペア  $rs$  間のパス  $k$  がリンク  $a$  を含むか否か (True=1, False=0)

参考

### UE/FD-Primal

$$\min Z(\mathbf{x}) = \sum_{a \in A} \int_0^{x_a} t_a(w) dw$$

s.t.

$$\sum_{k \in K_{rs}} f_k^{rs} - q_{rs} = 0 \quad \forall rs \in \Omega$$

$$x_a = \sum_{rs \in \Omega} \sum_{k \in K_{rs}} f_k^{rs} \delta_{a,k}^{rs} \quad \forall a \in A$$

$$f_k^{rs} \geq 0 \quad \forall k \in K_{rs}, \forall rs \in \Omega$$

$$x_a \geq 0 \quad \forall a \in A$$

# システム最適配分(SO)と混雑課金

## ■ UEとSOの関係・混雑課金

### 私的限界費用

1人の利用者が道路に加わるときにその個人が支払うコスト,  $t(x)$

### 社会的限界費用

1人の利用者が道路に加わるときの総所要時間の増加分

総所要時間の微分

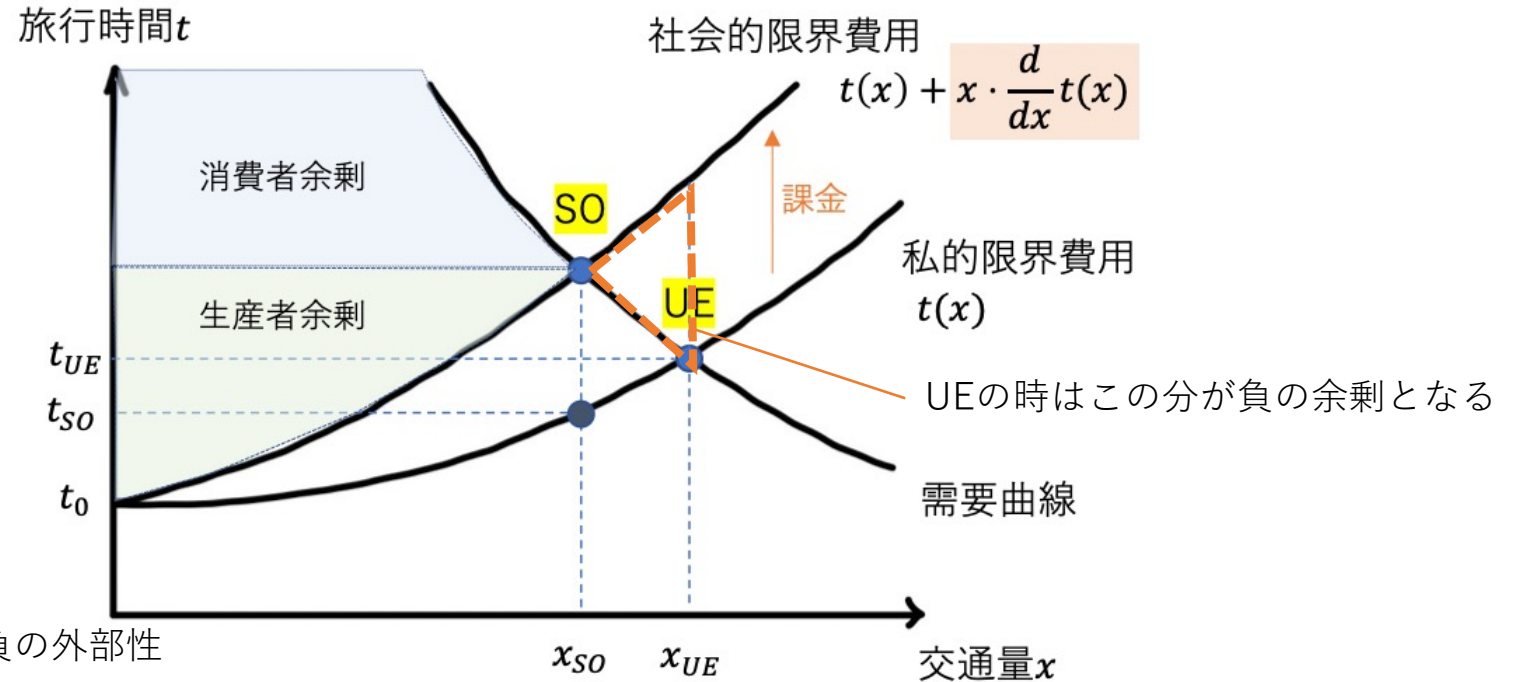
$$\frac{d}{dx}(xt(x)) = t(x) + x \cdot \frac{d}{dx}t(x) \quad (*)$$

周りの利用者に与えた負の外部性

社会的余剰 = 消費者余剰 + 生産者余剰：  
需要曲線と社会的限界費用とで挟まれた部分の面積

SOで社会的余剰最大, UEでは供給過多

※ 外部不経済の余剰分析について詳しくは[このサイト](#)など参照



(\*)の第二項の分を混雑課金することで  
UEをSOに一致させられる。

「限界費用課金」

# システム最適配分(SO)と混雑課金

## ■ UEとSOの関係・混雑課金

### 私的限界費用

1人の利用者が道路に加わるときにその個人が支払うコスト,  $t(x)$

### 社会的限界費用

1人の利用者が道路に加わるときの総所要時間の増加分

総所要時間の微分

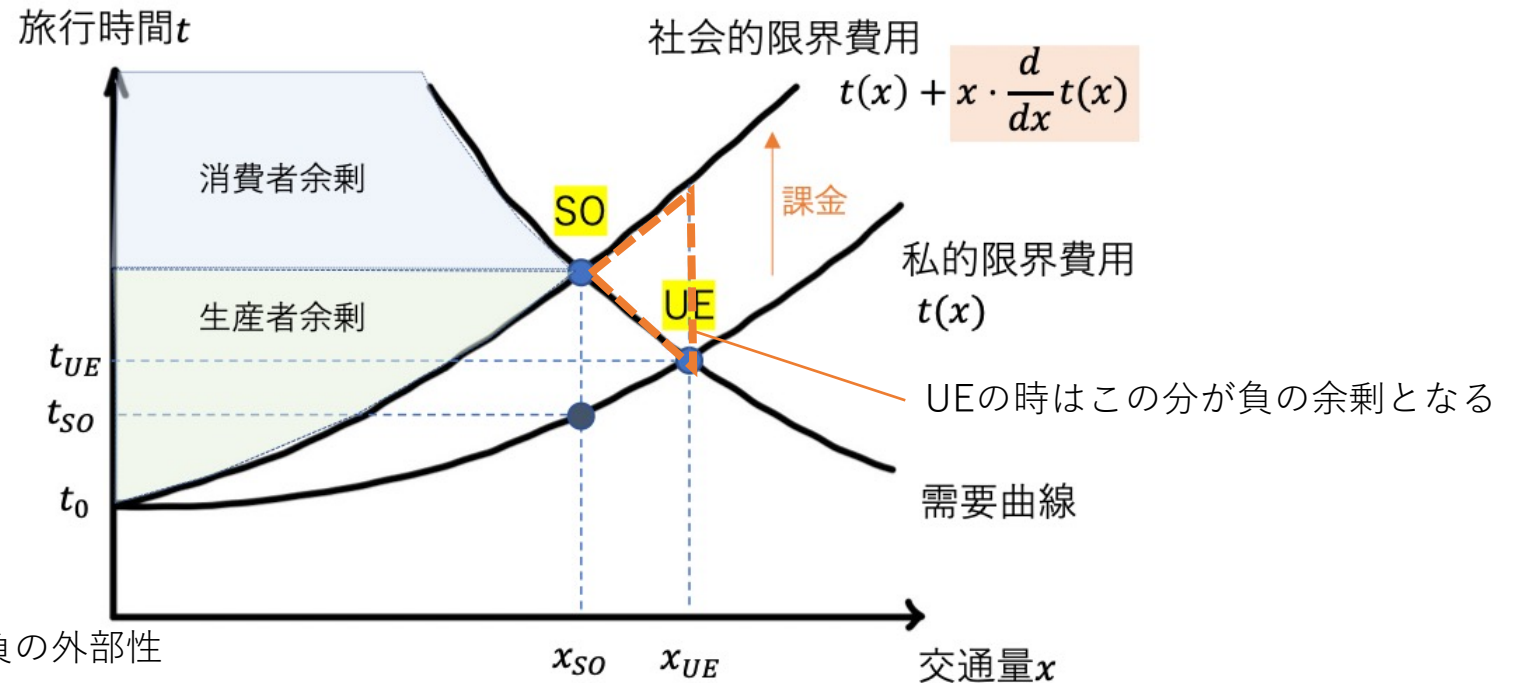
$$\frac{d}{dx}(xt(x)) = t(x) + x \cdot \frac{d}{dx}t(x) \quad (*)$$

周りの利用者に与えた負の外部性

社会的余剰 = 消費者余剰 + 生産者余剰：  
需要曲線と社会的限界費用とで挟まれた部分の面積

SOで社会的余剰最大, UEでは供給過多

※ 外部不経済の余剰分析について詳しくは[このサイト](#)など参照

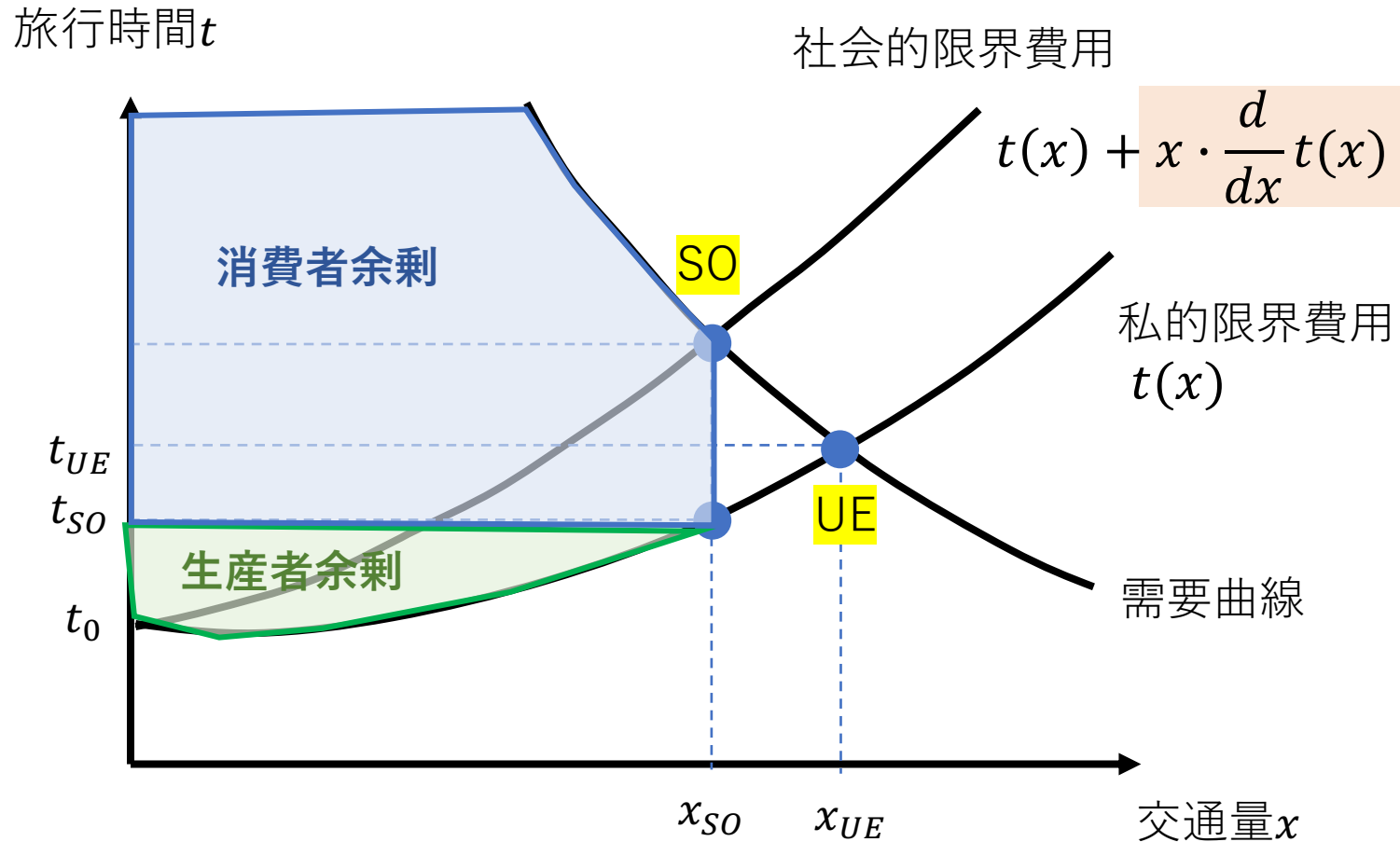


(\*)の第二項の分を混雑課金することで  
UEをSOに一致させられる。

「限界費用課金」

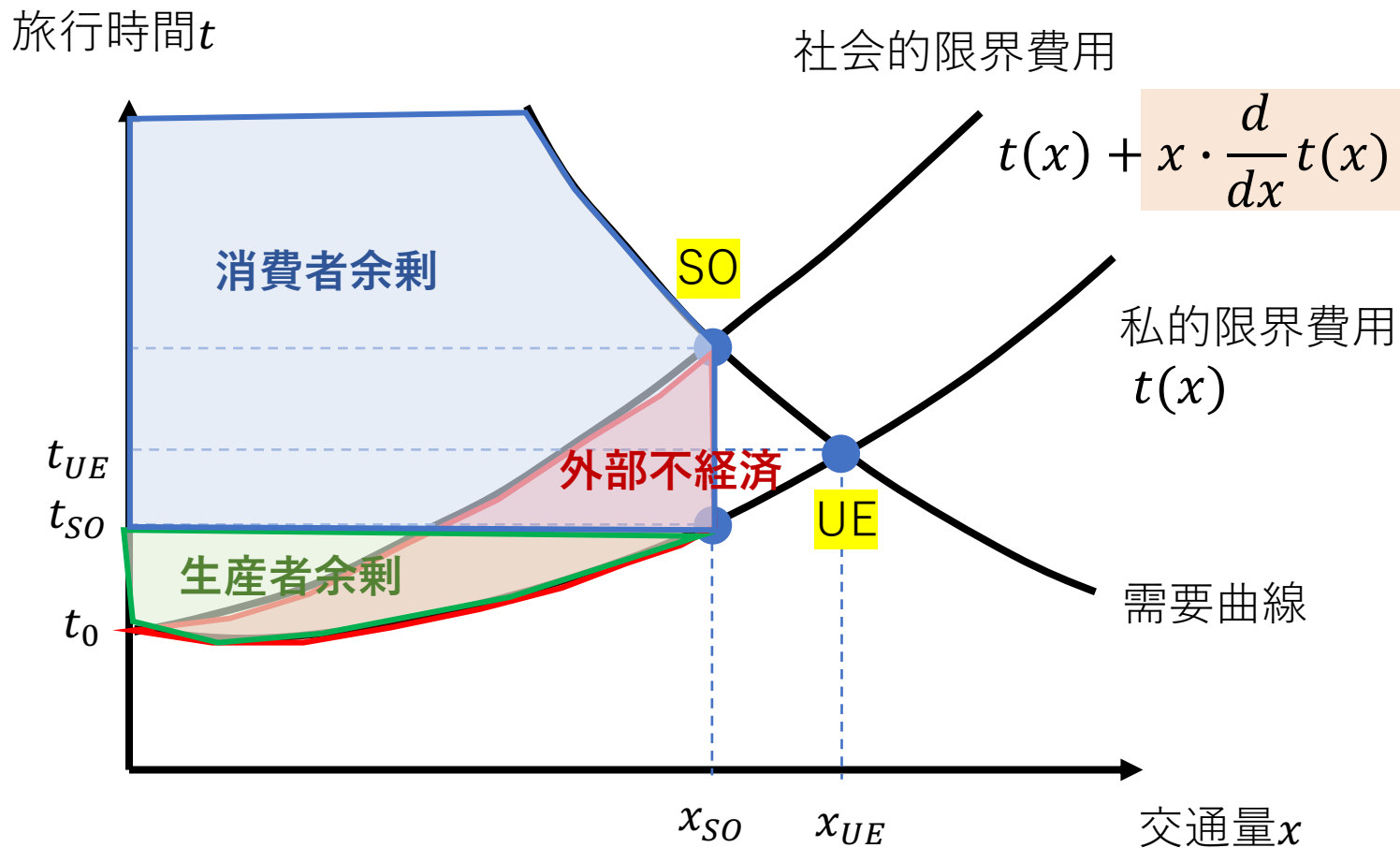
# システム最適配分(SO)と混雑課金

## SOのときの余剰



# システム最適配分(SO)と混雑課金

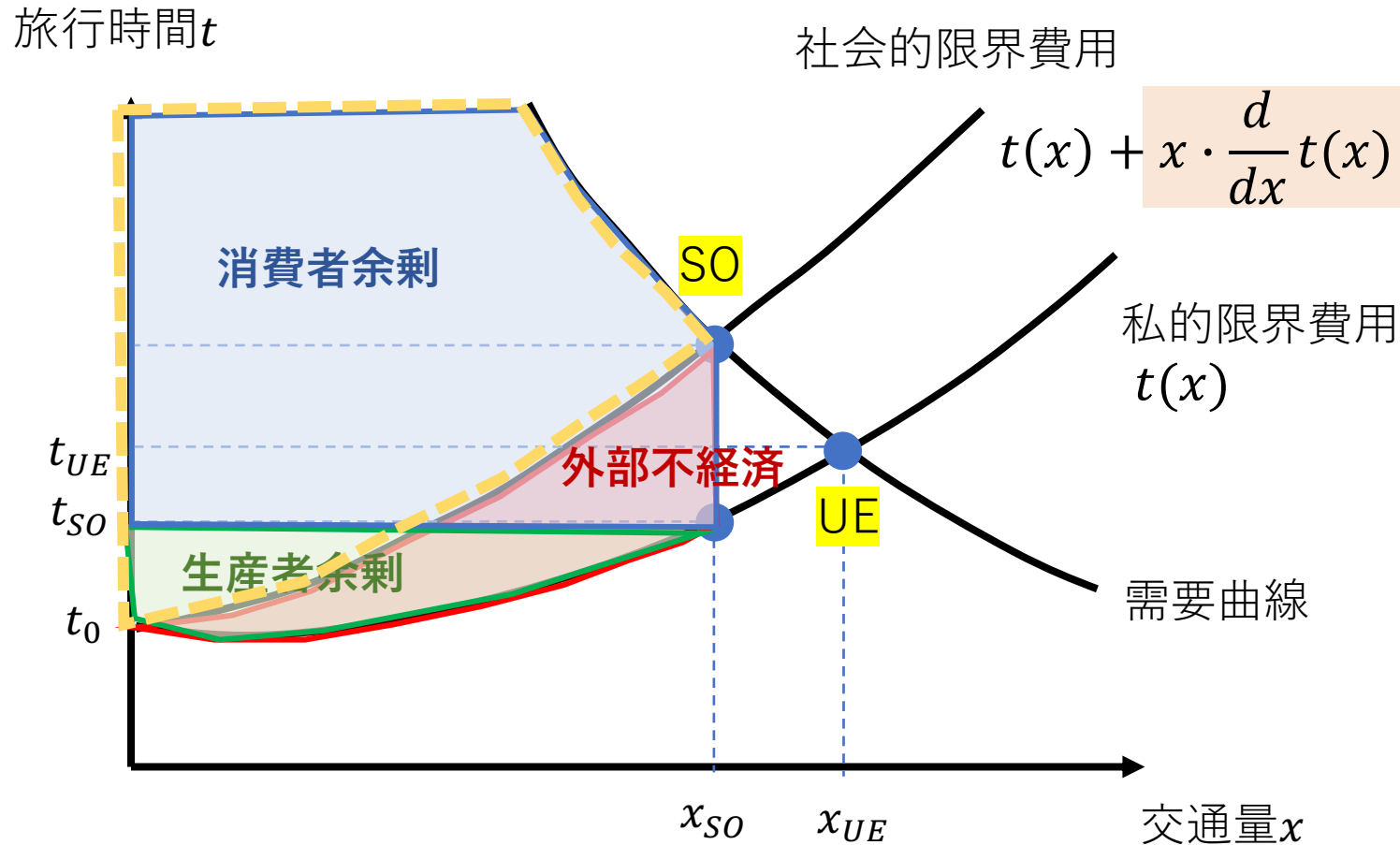
## SOのときの余剰



利用者が交通に加わることによって周囲に与えたコスト (=外部不経済)は負の余剰として働く

# システム最適配分(SO)と混雑課金

## SOのときの余剰



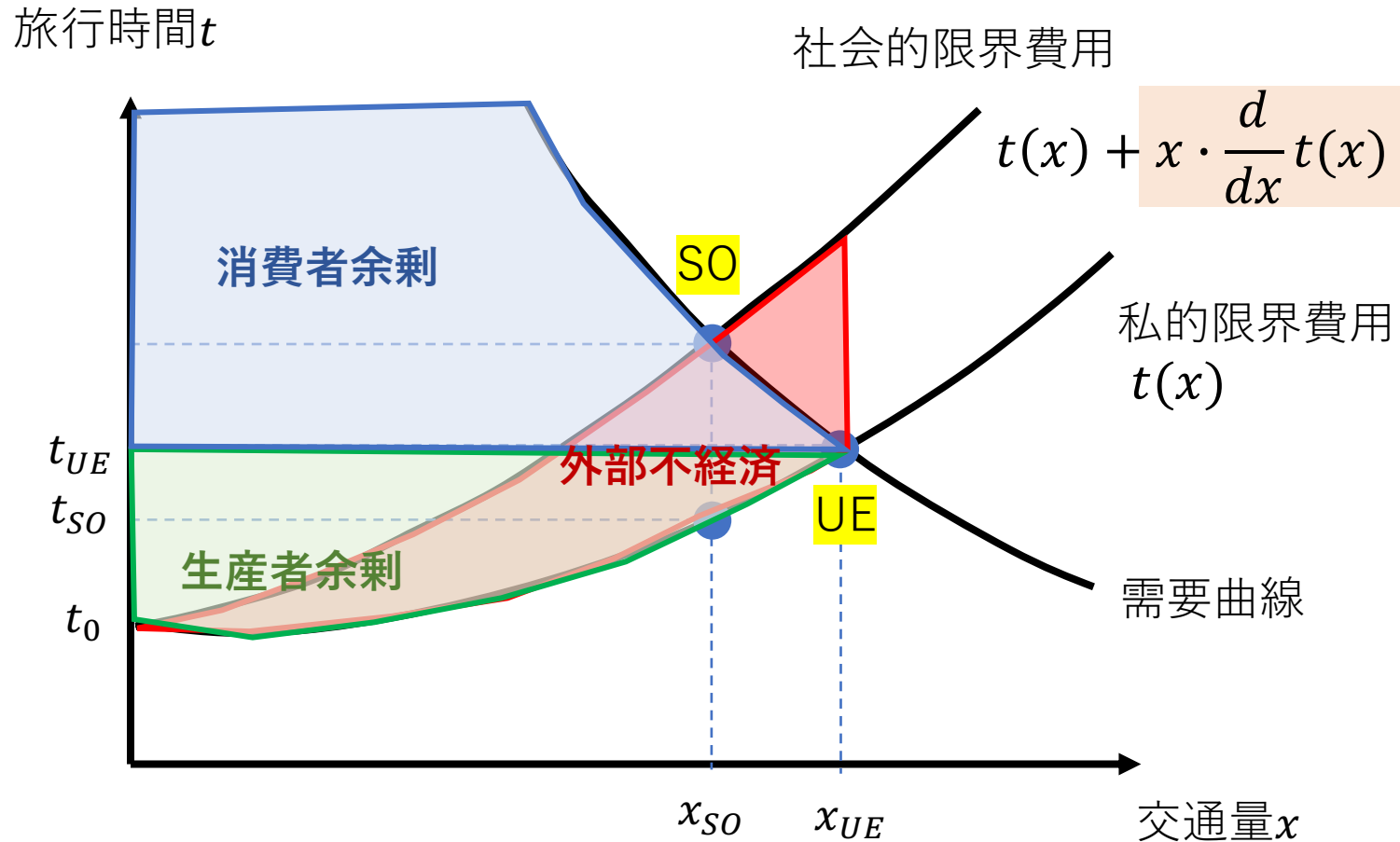
利用者が交通に加わることによって周囲に与えたコスト (=外部不経済)は負の余剰として働く



がSOの時の総余剰

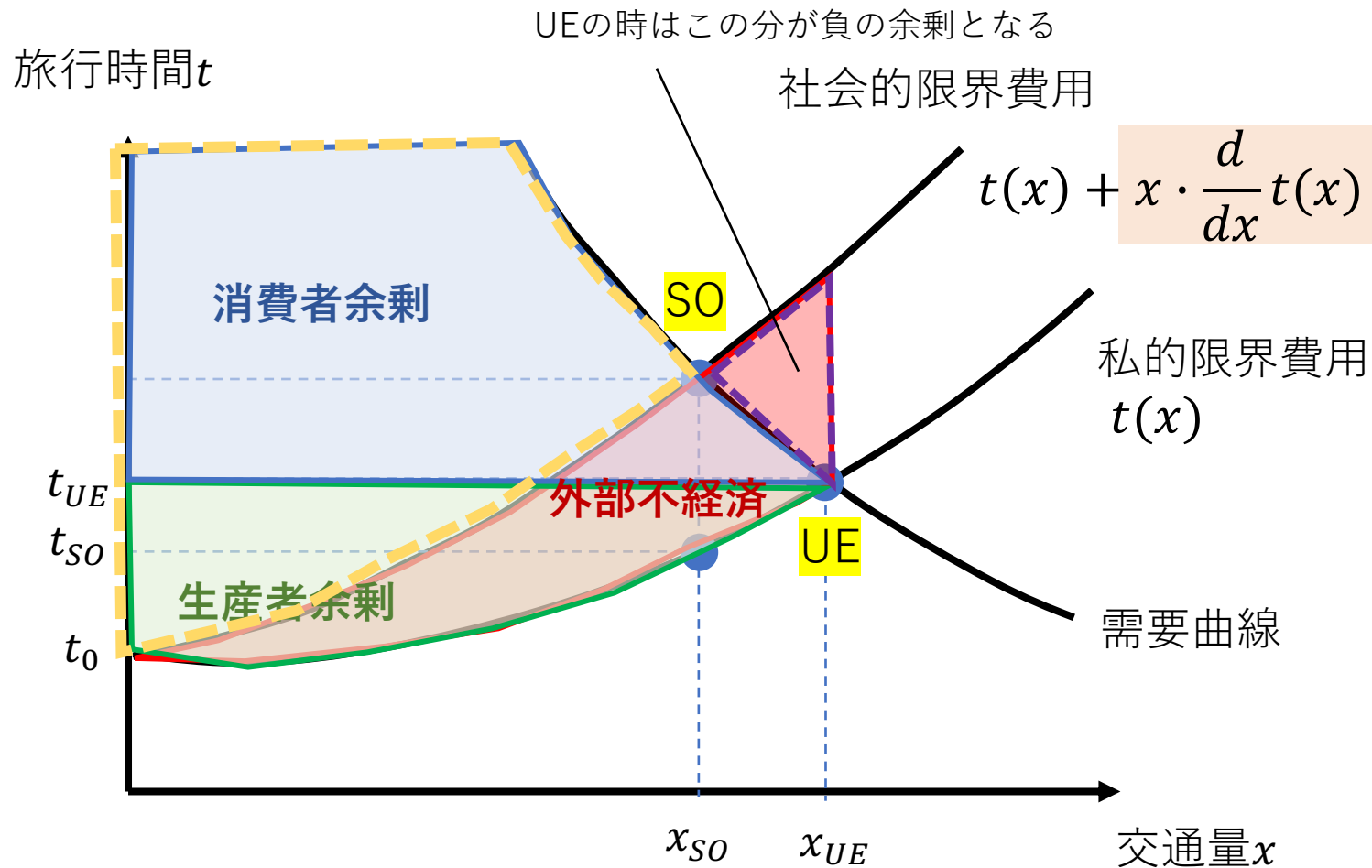
# システム最適配分(SO)と混雑課金

## UEのときの余剰



# システム最適配分(SO)と混雑課金

## UEのときの余剰



総余剰は



UEはSOより紫の部分だけ余剰が小さい!