第3回スタートアップゼミ　2017/4/20

最適化問題
最短経路探索とダイクストラ法
均衡配分法のアルゴリズム

広瀬啓人

今日の内容

1. 最適化問題（均衡配分モデルの導出における数学的手法）
2. 最短経路探索とダイクストラ法
3. 均衡配分法のアルゴリズム

今日の内容

1. 最適化問題（均衡配分モデルの導出における数学的手法）
2. 最短経路探索とダイクストラ法
3. 均衡配分法のアルゴリズム

1.1 最適化問題

N個の制約条件 $g_n(x_1, \dots, x_M) \geq d_n (n = 1, \dots, N)$ を満たしながら、

目的関数 $Z(x_1, \dots, x_M)$ を最小にするような

M個の最適変数の組$(x_1^*, \dots, x_M^*)$を求める問題

$$\min \underline{Z(x_1, \dots, x_M)}$$

最小化したい関数

$$subject\ to$$

$$\boxed{\begin{array}{c} g_1(x_1, \dots, x_M) \geq d_1 \\ \vdots \\ g_N(x_1, \dots, x_M) \geq d_N \end{array}}$$

制約条件

1.2 解法

以下を調べる

① 必要性:1次の最適性条件･･･局所的最適解

② 十分性:2次の最適性条件

②－1:目的関数が狭義凸関数

「Zのヘッセ行列が正定値」⇒「関数Zは狭義凸関数」

※ヘッセ行列(Hessian) :$\boldsymbol{H} = \nabla^2 Z(\boldsymbol{x})$

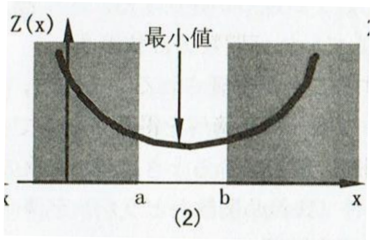
ヘッセ行列が正定値:任意の$\boldsymbol{x}$にたいして、${}^t\boldsymbol{xHx} > \boldsymbol{0}$

②－2:実行可能領域が凸集合

実行可能領域内の2点を結ぶ線分が常に実行可能領域内

1.3 一変数の場合（M=1）

① 1次の最適性条件

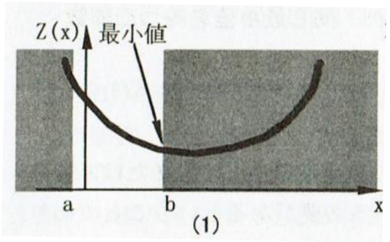


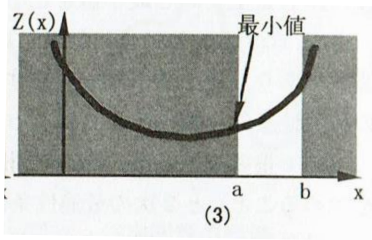
最小点が実行可能領域内: $\dfrac{dZ(x^*)}{dx} = 0$

最小点が実行可能領域外・・・制約条件上で最小値になっている

最小点で<u>効いている</u>制約条件g_tについて

$$\frac{dZ(x^*)}{dx} = h_t \frac{dg_t(x^*)}{dx} \quad (h_t > 0)$$





Kuhn-Tucker条件

一般化

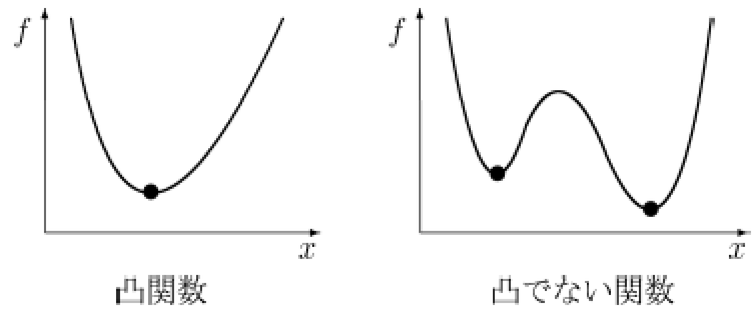
$$\frac{dZ(x^*)}{dx_m} = \sum_n h_n \frac{dg_n(x^*)}{dx_m}$$

$$h_n \geq 0, h_n[d_n - g_n(x^*)] = 0, g_n(x^*) \geq d_n$$

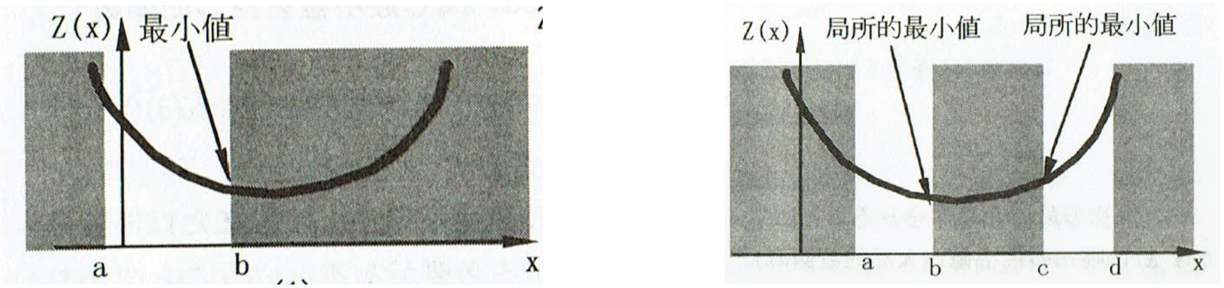
1.3 一変数の場合（M=1）

② 2次の最適性条件

②－1：$\dfrac{d^2Z}{dx^2} > 0$



②―2：



$a \leq x \leq b$

$a \leq x \leq b, c \leq x \leq d$

$g_1(x) = x \geq a$
$g_2(x) = -x \geq -b$

凸領域

凸領域ではない

1.4 多変数の場合（制約条件なし）

①1次の最適性条件・・・$\nabla Z(\boldsymbol{x}^*) = (\frac{\partial Z(\boldsymbol{x}^*)}{\partial x_1}, \ldots, \frac{\partial Z(\boldsymbol{x}^*)}{\partial x_M}) = 0$

②2次の最適性条件

②ー１: ${}^t\boldsymbol{xHt} > 0$

$$\boldsymbol{H} = \nabla^2 Z(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial^2 Z(\boldsymbol{x})}{\partial x_1^2} & \cdots & \frac{\partial^2 Z(\boldsymbol{x})}{\partial x_1 \partial x_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 Z(\boldsymbol{x})}{\partial x_M \partial x_1} & \cdots & \frac{\partial^2 Z(\boldsymbol{x})}{\partial x_M^2} \end{pmatrix}$$

②ー２: 自明

(2変数の場合）

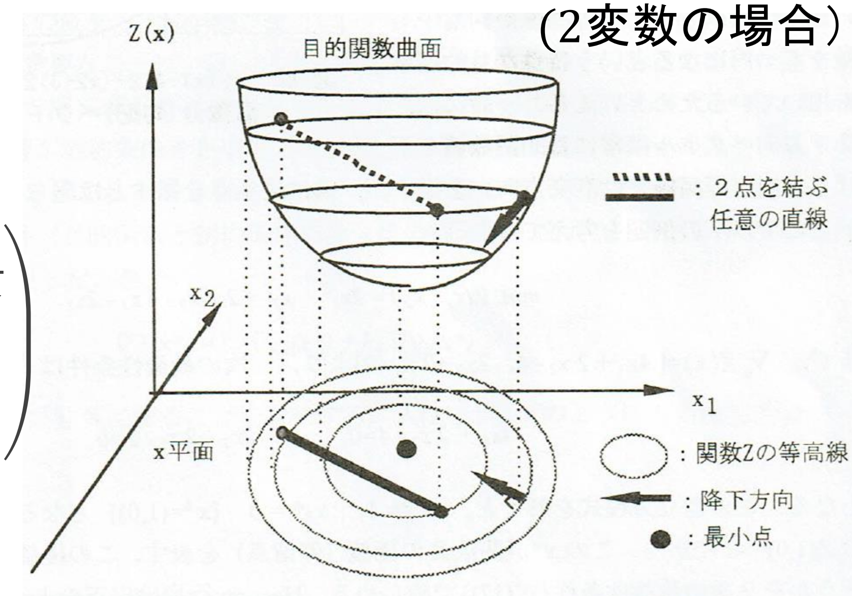


図-6a　狭義凸関数とx平面上における関数Zの等高線

1.5 多変数の場合（制約条件あり）

$$\begin{aligned} &\min Z(x_1, \dots, x_M) \\ &\quad \text{subject to} \\ &\qquad g_1(x_1, \dots, x_M) \geq d_1 \\ &\qquad \vdots \\ &\qquad g_n(x_1, \dots, x_M) \geq d_n \end{aligned}$$

① 1次の最適性条件

・最小点が実行可能領域内・・・$\nabla Z(\boldsymbol{x}^*) = 0$

・最小点が実行可能領域外

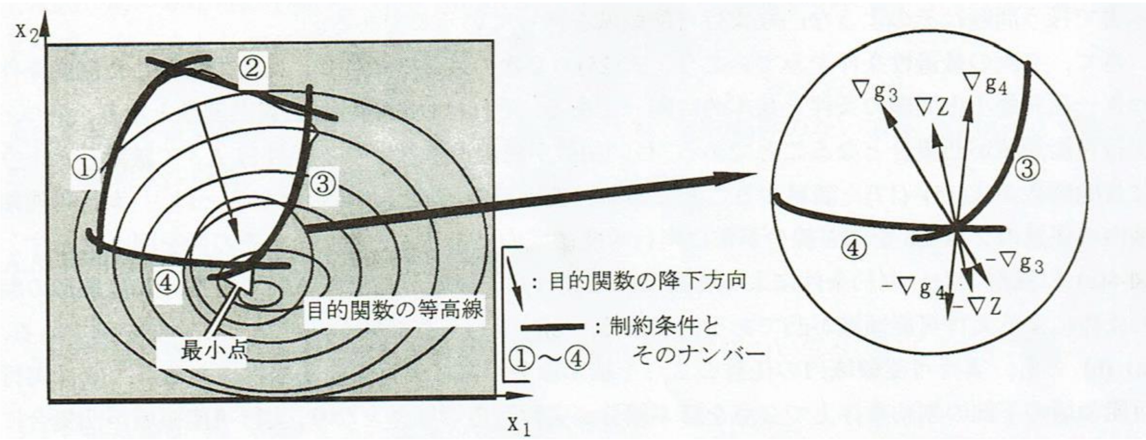
最小点が効いている制約条件$\{t\}$上で最小となるとき

$$\frac{\partial Z(\boldsymbol{x}^*)}{\partial x_m} = \sum_t h_t \frac{\partial g_t(\boldsymbol{x}^*)}{\partial x_m} \quad (h_t > 0)$$

たとえば・・・

$$\frac{\partial Z(\boldsymbol{x}^*)}{\partial x_n} = h_3 \frac{\partial g_3(\boldsymbol{x}^*)}{\partial x_n} + h_4 \frac{\partial g_4(\boldsymbol{x}^*)}{\partial x_n}$$

$h_3 \geq 0, h_4 \geq 0$



1.5 多変数の場合（制約条件あり）

$$\min Z(x_1,\dots,x_M)$$
$$\quad subject\ to$$
$$\quad\quad g_1(x_1,\dots,x_M) \geq d_1$$
$$\vdots$$
$$\quad\quad g_n(x_1,\dots,x_M) \geq d_n$$

一般化した1次の最適化条件

Kuhn-Tucker条件

$$\frac{\partial Z(\boldsymbol{x}^*)}{\partial x_m} = \sum_n h_n \frac{\partial g_n(\boldsymbol{x}^*)}{\partial x_m}$$
$$h_n \geq 0, h_n[d_n - g_n(\boldsymbol{x}^*)] = 0, g_n(\boldsymbol{x}^*) \geq d_n$$

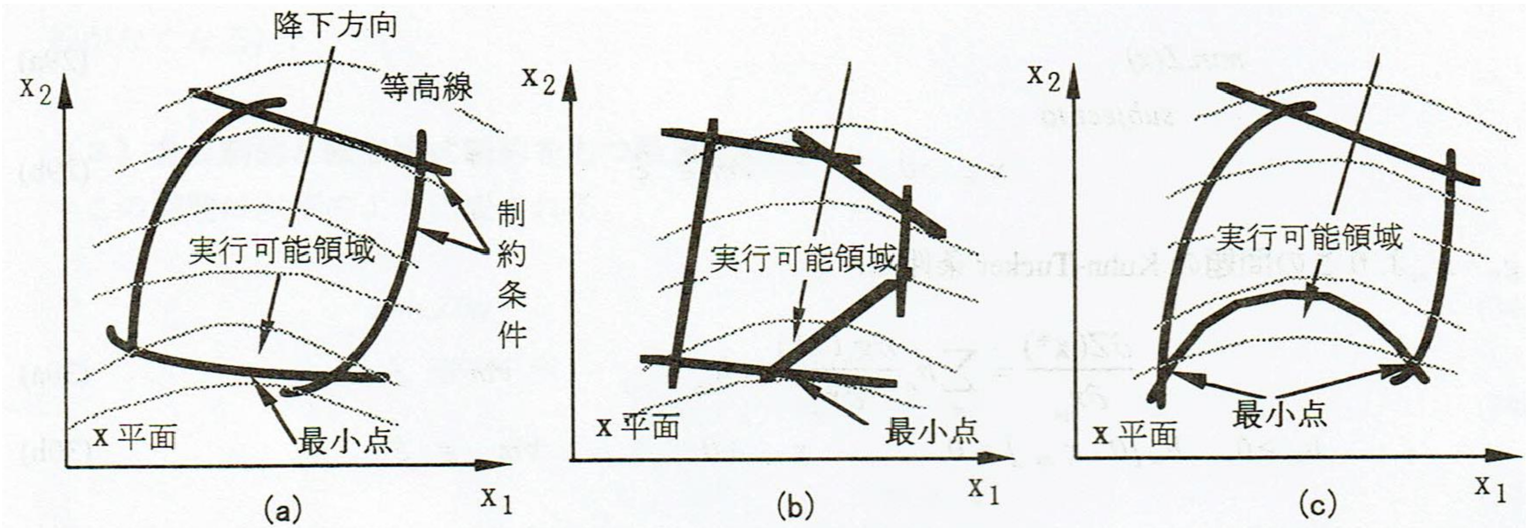
h_nをLagrangian乗数（Kuhn-Tucker乗数）という

1.5 多変数の場合（制約条件あり）

② 2次の最適性条件

②－１: ${}^{t}\boldsymbol{xHt} > 0$

②－２:



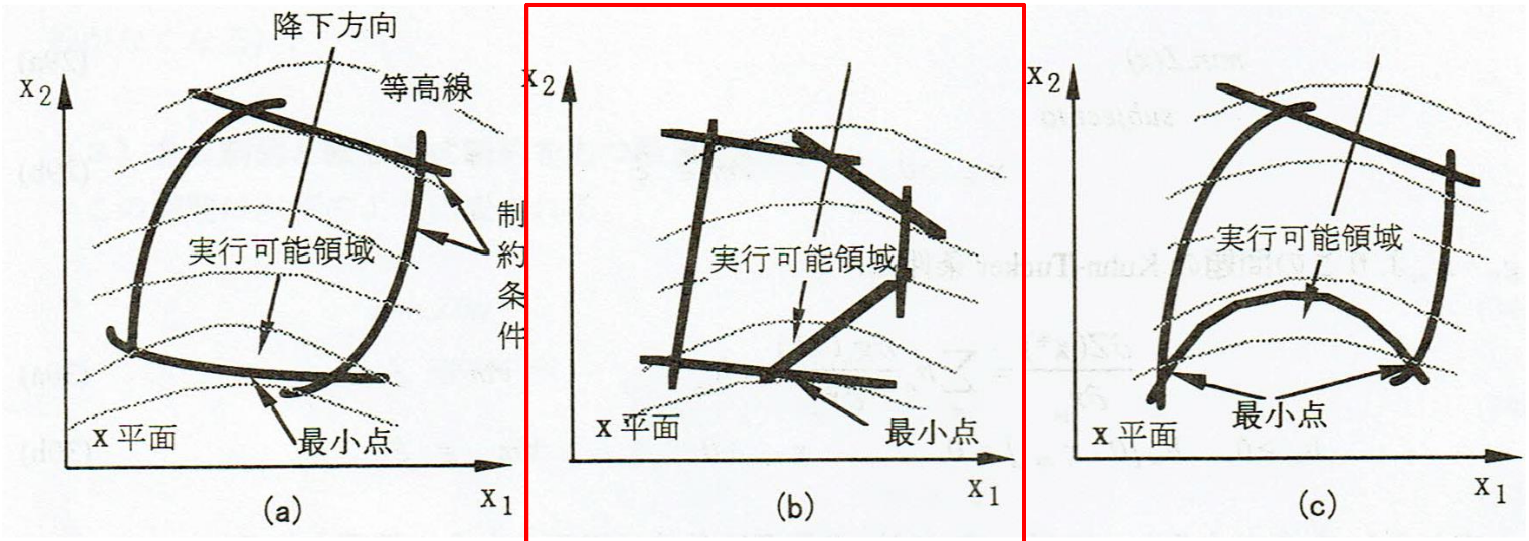
凸領域　　凸領域　　凸領域ではない

1.5 多変数の場合（制約条件あり）

② 2次の最適性条件

②－１: ${}^{t}\boldsymbol{x}\boldsymbol{H}\boldsymbol{t} > 0$

②－２:



凸領域　　凸領域　　凸領域ではない

1.5 多変数の場合（制約条件あり）

練習問題

$$\min Z(x_1, x_2) = 2x_1^2 + x_2^2 + 2x_1x_2 - 4x_1 - 2x_2$$
$$subject\ to\ \ 2x_1 + x_2 \geq 5$$

1.5 多変数の場合（制約条件あり）

$$\min Z(x_1,x_2) = 2x_1^2 + x_2^2 + 2x_1x_2 - 4x_1 - 2x_2$$
$$subject\ to\ \ 2x_1 + x_2 \geq 5$$

解答

$\nabla Z(\boldsymbol{x}) = (4x_1 + 2x_2 - 4, 2x_2 + 2x_1 - 2)$ゆえ、Kuhn-Tucker条件より、

$$\begin{cases} 4x_1 + 2x_2 - 4 = 2h_1 \\ 2x_2 + 2x_1 - 2 = h_1 \\ h_1 \geq 0 \\ h_1(2x_1 + x_2 - 5) = 0 \\ 2x_1 + x_2 \geq 5 \end{cases}$$

$h_1 = 0$とすると、第1式と第2式より$\boldsymbol{x} = (1,0)$を得るが、これは第5式を満たさない。
（※これは、制約条件がないときの最適解となっている）

従って$h_1 > 0$とし、連立方程式を解くと、

$$\boldsymbol{x} = \left(\frac{5}{2}, 0\right), h_1 = 3, Z\left(\frac{5}{2}, 0\right) = \frac{5}{2}$$

$Hessian$行列は、

$$\nabla^2 Z(\boldsymbol{x}) = \begin{pmatrix} 4 & 2 \\ 2 & 2 \end{pmatrix}$$

であり、任意の$\boldsymbol{x}(\neq \boldsymbol{0})$について、

$${}^t\boldsymbol{x}\boldsymbol{H}\boldsymbol{x} = (2x_1 + x_2)^2 + x_2^2 > 0$$

よって、$\boldsymbol{x} = \left(\frac{5}{2}, 0\right)$は大局的最適解である。

1.5 多変数の場合(制約条件あり)

$$\min Z(x_1, x_2) = 2x_1^2 + x_2^2 + 2x_1x_2 - 4x_1 - 2x_2$$
$$subject\ to\ \ 2x_1 + x_2 \geq 5$$

参考:Rを使って解くとこんな感じになります

R Console

```
> fQP<-function(x){
+ 2*x[1]^2+x[2]^2+2*x[1]*x[2]-4*x[1]-2*x[2]
+ }
> constrOptim(c(3,5),fQP,NULL,ui=c(2,1),ci=5)
$par
[1] 2.4999278928 0.0001442147

$value
[1] 2.5

$counts
function gradient
     246       NA

$convergence
[1] 0

$message
NULL

$outer.iterations
[1] 3
```

http://www.okadajp.org/RWiki/?%E7%B7%9A%E5%BD%A2%E4%B8%8D%E7%AD%89%E5%BC%8F%E5%88%B6%E7%B4%84%E4%BB%98%E3%81%8D%E3%81%AE%E6%9C%80%E9%81%A9%E5%8C%96%E9%96%A2%E6%95%B0%20constrOptim

1.6 特定条件下のKuhn-Tucker条件

非負制約と、線形等式制約をもつ場合

$$\min Z(\boldsymbol{x})$$
$$subject\ to$$
$$\sum_m f_{mn} x_m = d_n$$
$$x_m \geq 0$$

Kuhn-Tucker条件

$$x_m^* \left(\frac{\partial Z(\boldsymbol{x}^*)}{\partial x_m} - \sum_n h_n f_{mn} \right) = 0 \ \ and \ \frac{\partial Z(\boldsymbol{x}^*)}{\partial x_m} - \sum_n h_n f_{mn} \geq 0, x_m^* \geq 0$$
$$\sum_m f_{mn} x_m = d_n$$

1.7 Lagrangian関数

$$L(\boldsymbol{x},\boldsymbol{h}) = Z(\boldsymbol{x}) + \sum_n h_n(d_n - g_n(\boldsymbol{x})) \quad (h_n \geq 0)$$

・最適化問題を解く上で便利な関数
（制約条件を関数内に取り込める）

もともとの最適化問題

$$\begin{aligned} &\min Z(x_1,\dots,x_M) \\ &\quad subject\ to \\ &\qquad g_1(x_1,\dots,x_M) \geq d_1 \\ &\qquad \vdots \\ &\qquad g_n(x_1,\dots,x_M) \geq d_n \end{aligned}$$

同値

Lagrangian関数の鞍点を求める ⟺ 元の制約条件付き最適化問題を解く

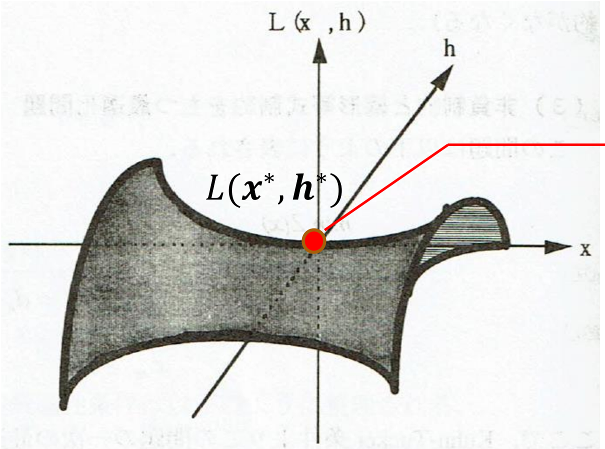


図-9　Lagrangian関数の鞍点

$L(\boldsymbol{x}^*,\boldsymbol{h}^*)$は$\boldsymbol{x}$について最小化し、$\boldsymbol{h}$について最大化した鞍点

1.7 Lagrangian関数

同値

Lagrangian関数の鞍点を求める ⟺ 元の制約条件付き最適化問題を解く

Lagrangian関数の一次の最適性条件（制約なし）

$$\frac{\partial L(\boldsymbol{x}^*, \boldsymbol{h}^*)}{\partial x_m} = 0$$

$$h_n \frac{\partial L(\boldsymbol{x}^*, \boldsymbol{h}^*)}{\partial h_n} = 0 \; and \; \frac{\partial L(\boldsymbol{x}^*, \boldsymbol{h}^*)}{\partial h_n} \geq 0, h_n \geq 0$$

⟺

Kuhn-Tucker条件

$$\frac{\partial Z(\boldsymbol{x}^*)}{\partial x_m} - \sum_n h_n \frac{\partial g_n(\boldsymbol{x}^*)}{\partial x_m} = 0$$

$$h_n \geq 0, h_n[d_n - g_n(\boldsymbol{x}^*)] = 0, d_n - g_n(\boldsymbol{x}^*) \leq 0$$

※スライド9

1.7 Lagrangian関数

$$L(\boldsymbol{x},\boldsymbol{h}) = Z(\boldsymbol{x}) + \sum_n h_n\left(d_n - \sum_m f_{mn}x_m\right), x_m \geq 0$$

Lagrangian関数の一次の最適性条件（非負制約、線形等式制約付き）

$$x_m \frac{\partial L(\boldsymbol{x}^*,\boldsymbol{h}^*)}{\partial x_m} = 0 \; and \; \frac{\partial L(\boldsymbol{x}^*,\boldsymbol{h}^*)}{\partial x_m} \geq 0, x_m \geq 0$$

$$\frac{\partial L(\boldsymbol{x}^*,\boldsymbol{h}^*)}{\partial h_n} = 0$$

もともとの最適化問題

$$\min Z(\boldsymbol{x})$$
$$subject\ to$$
$$\sum_m f_{mn}x_m = d_n$$
$$x_m \geq 0$$

$\Longleftrightarrow$

$$x_m^*\left(\frac{\partial Z(\boldsymbol{x}^*)}{\partial x_m} - \sum_n h_n f_{mn}\right) = 0 \; and \; \frac{\partial Z(\boldsymbol{x}^*)}{\partial x_m} - \sum_n h_n f_{mn} \geq 0, x_m^* \geq 0$$

$$\sum_m f_{mn}x_m = d_n$$

※スライド15

Kuhn-Tucker条件

1.8 確定的利用者均衡配分

- 等価な最適化問題への変換

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

制約条件：

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

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

$$f_k^{rs} \geq 0, x_a \geq 0$$

非負制約、線形等式制約つきの
最適化問題

今日の内容

1. 最適化問題（均衡配分モデルの導出における数学的手法）
2. 最短経路探索とダイクストラ法
3. 均衡配分法のアルゴリズム

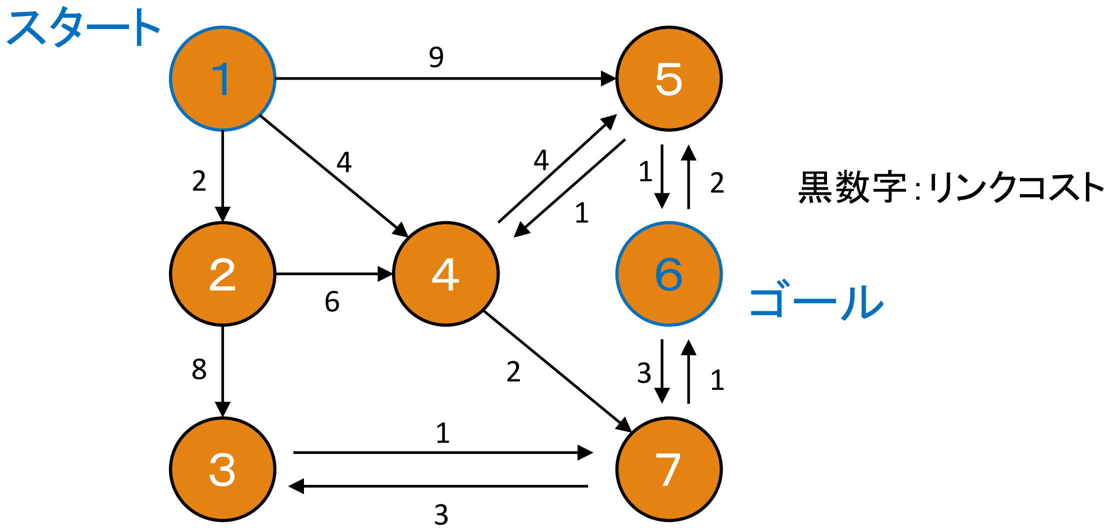
2.1 最短経路探索

交通ネットワークの最適化問題を解く上での問題

考えられる膨大な経路から、いかに効率的に利用経路を列挙するか

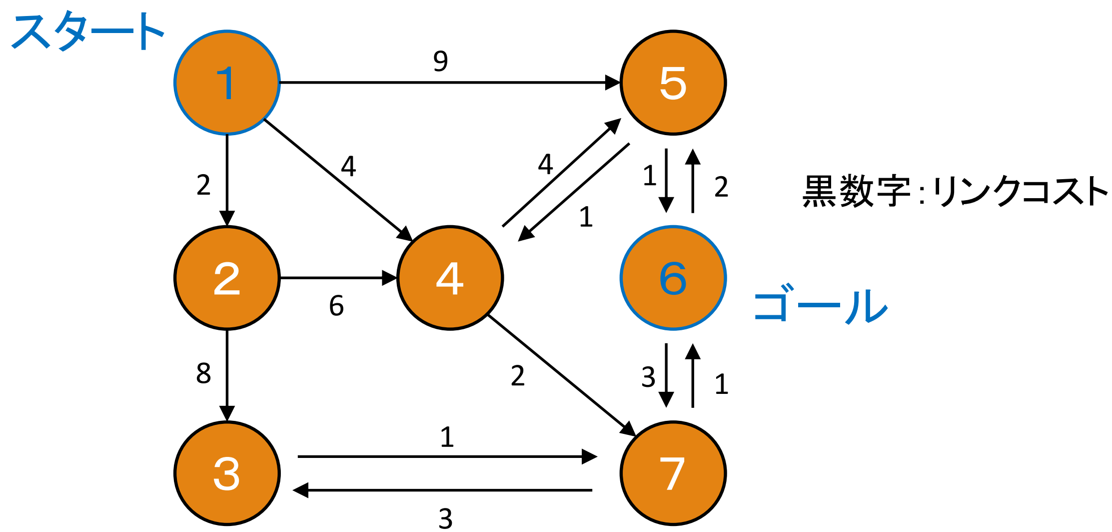
2.1 最短経路探索

下図で、ノード1からノード6までの最短ルートは？



2.1 最短経路探索

下図で、ノード1からノード6までの最短ルートは？



答えは、 1 → 4 → 7 → 6

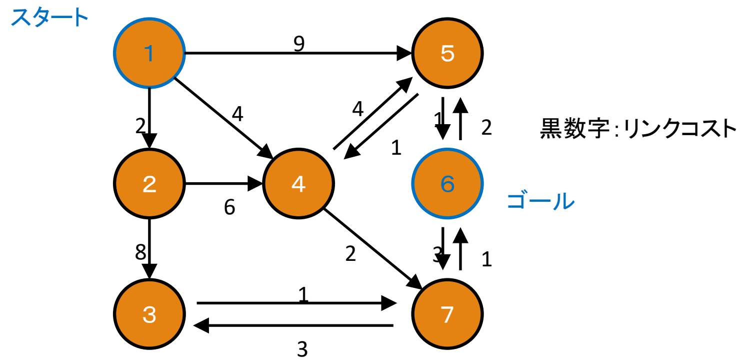
どのようにして探すか？全ての経路を書き出す？

1 → 2 → 4 → 7 → 6 (11) / 1 → 2 → 3 → 7 → 6 (14) / 1 → 4 → 7 → 6 (7)

1 → 5 → 4 → 7 → 6 (16) / 1 → 5 → 6 (10)

→全てのノードを求める場合や、ネットワークが巨大な場合、計算量が膨大

2.1 最短経路探索



たとえば・・・

答え 1 → 4 → 7 → 6

ノード7までの最短経路

1 ~~→ 2 → 4 → 7~~ → 6 (11) / 1 → ~~2 → 3 → 7 → 6 (14)~~ / 1 → 4 → 7 → 6 (7)

1 ~~→ 5 → 4 → 7~~ → 6 (16) / 1 → 5 → 6 (10)

より効率的に解を見つけられる

2.1 最短経路探索

交通ネットワークの最適化問題を解く上での問題

考えられる膨大な経路から、いかに効率的に利用経路を列挙するか

最短経路探索法は利用者均衡配分のアルゴリズムの一部として

よく用いられる

効率がよいとされる方法として以下の2つ

- Dijkstra法（ラベル確定法）
- ラベル修正法

2.2 Dijkstra法

ノードとリンクからなるネットワークを考え、ある始点から

他のノードまでの最短経路（＝最小費用で行ける経路）を求める

Edsger Wybe Dijkstra
(1930-2002)

＜変数の設定＞

・すべてのノードは、既に最短経路が求まっているノードの集合Kと

まだ最短経路が求まっていないノードの集合$\overline{K}$のどちらかに分類できる

・ c_i : 始点からノードiまでの最小交通費用

$i \in K$のとき　c_i: 最小交通費用（確定）

$i \in \overline{K}$のとき　c_i:部分的最小費用・・・暫定の最小費用のようなもの

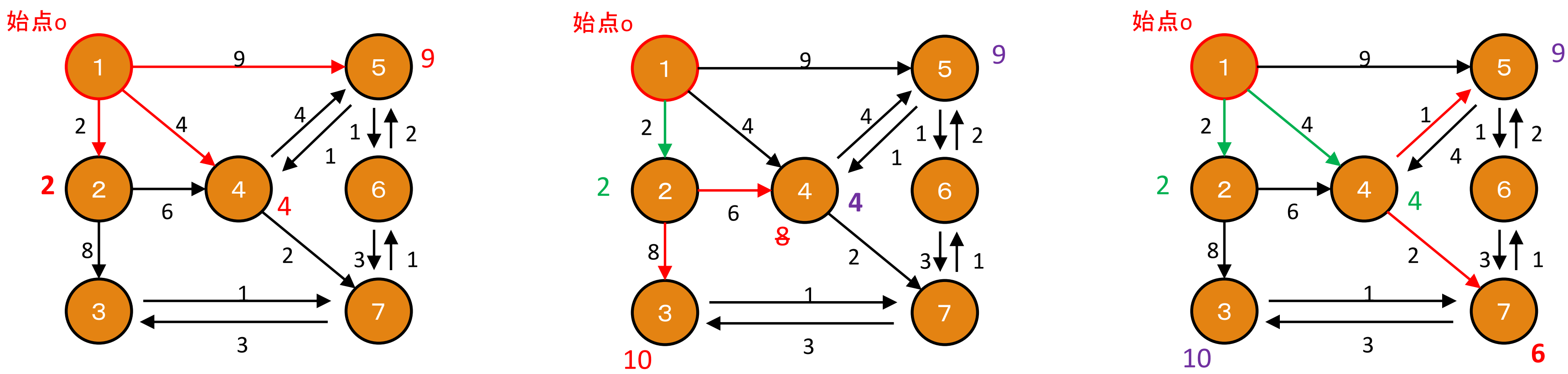
・ t_{ij} : ノードiからノードjまでの交通費用

・各ノードについて、ポインタF_iという量を考える（最短経路を列挙する際に使う）

2.2 Dijkstra法

＜イメージ＞

「最短だとわかるところから決めていく」



2.2 Dijkstra法

アルゴリズム

Step1

全てのノード$\{j\}$について、部分的最小費用$c_j = \infty, F_j = 0, j \in \overline{K}$とする。
起点をoとし、$c_o = 0, i = o$とする。ノードoを集合Kに移す

Step2

ノードiから出る全リンクの終点ノード$\{m\}$について、
$c_m > c_i + t_{im}$ならば、$c_m = c_i + t_{im}$に更新し、$F_m = i$とする

Step3

集合$\overline{K}$に属すすべてのノードの中での部分的最小費用が最小となっているノードを求め、これをノードjとする。ノードjを集合Kにうつす。

$$c_j = \min_p (c_p) \ (p \in \overline{K})$$

Step4

$c_p = \infty$以外のすべてのノードが集合Kに移されているかチェック

NO

$i = j$として
Step2へ

YES

終了

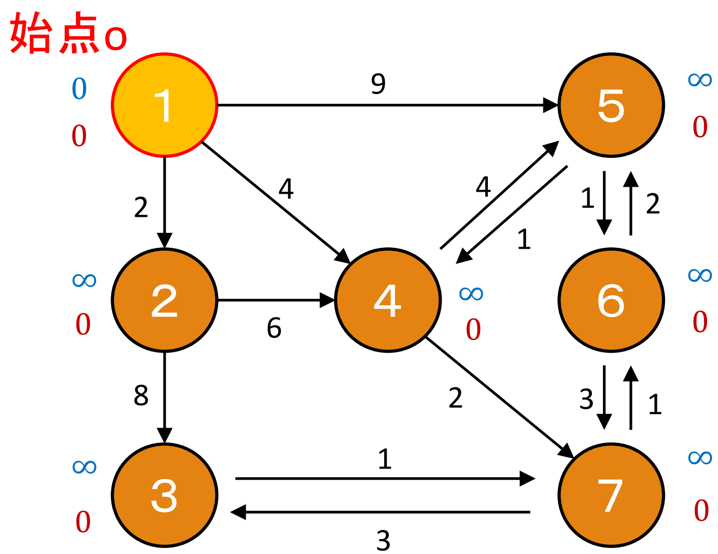
2.3 Dijkstra法の例題

図のような、7つのノードとそれらがリンクで結ばれているネットワークを考える。

始点をノード1として、ノード1から各ノードまでの最短距離と最小費用を求めよう。

<Step1>

$i = 1$



黒数字：リンクコスト
赤数字：F_m（ポインタ）
青数字：c_m（最小費用）

c_m							F_m						
1	2	3	4	5	6	7	1	2	3	4	5	6	7
0	∞	∞	∞	∞	∞	∞	0	0	0	0	0	0	0

K	1
$\overline{K}$	2,3,4,5,6,7

2.3 Dijkstra法の例題

<Step2>

ノード1から行けるのはノード2, 4, 5→ $m = 2{,}4{,}5$

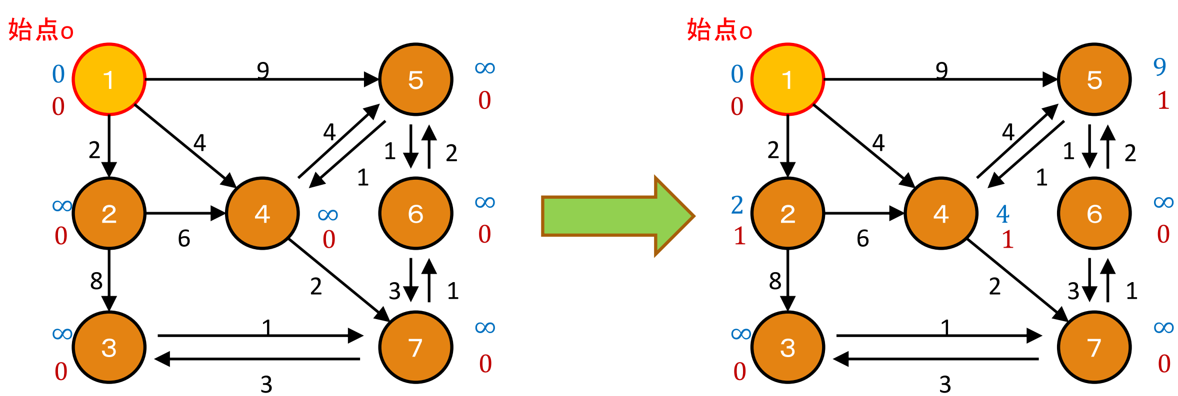
$$c_2' = c_1 + t_{12} = 0 + 2 = 2 < \infty (= c_2)$$

$$c_4' = c_1 + t_{14} = 0 + 4 = 4 < \infty (= c_4)$$

$$c_5' = c_1 + t_{15} = 0 + 9 = 9 < \infty (= c_5)$$

$m = 2, 4, 5$すべてで$c_m' < c_m$となっているので、すべて更新。

また、$F_m = i = 1$とする。



黒数字:リンクコスト
赤数字:F_m(ポインタ)
青数字:c_m(最小費用)

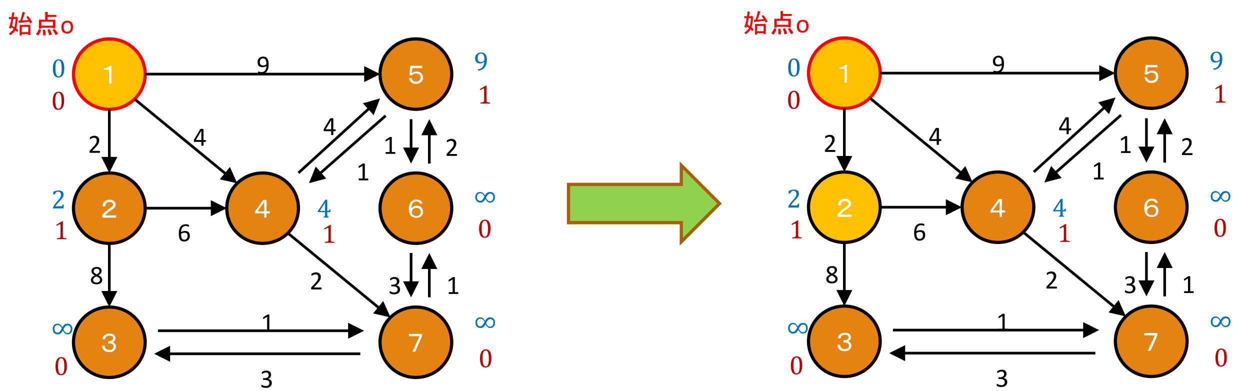
2.3 Dijkstra法の例題

<Step3>

$\overline{K} = \{2, 3, 4, 5, 6, 7\}$に属すノードの中で費用が最小となるのは、ノード2

$\rightarrow j = 2$ として、ノード2を集合Kへと移す

ノード2を含む最短経路：$1 \rightarrow 2$　最小交通費用：$c_2 = c_1 + t_{12} = 0 + 2 = 2$



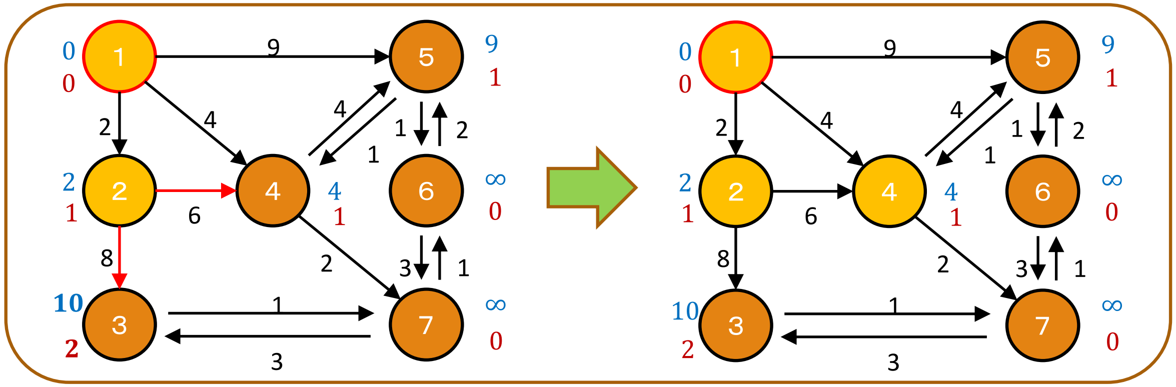
<Step4> $c_p \neq \infty$かつ$p \in \overline{K}$なるpが残っている

→次は、$i = 2$でstep2～step4を行う

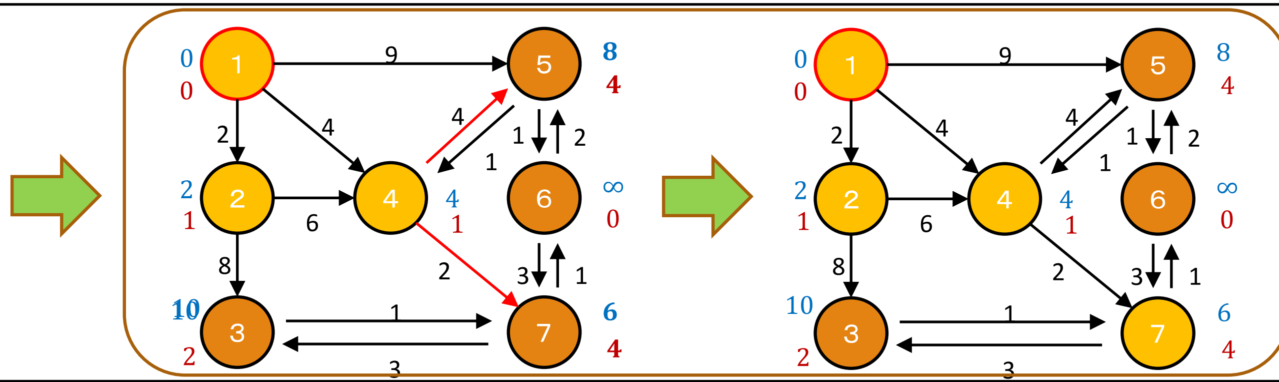
2.3 Dijkstra法の例題

太字は更新された箇所

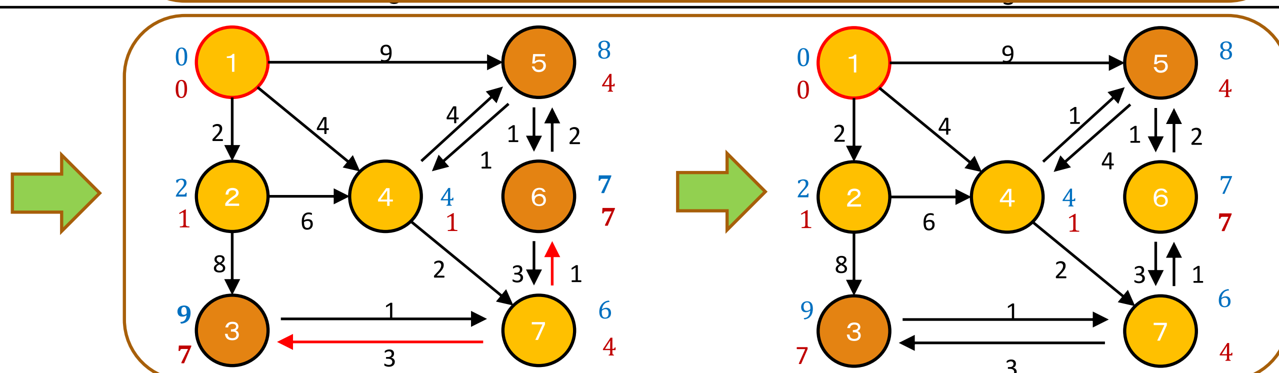
$i = 2$



$i = 4$

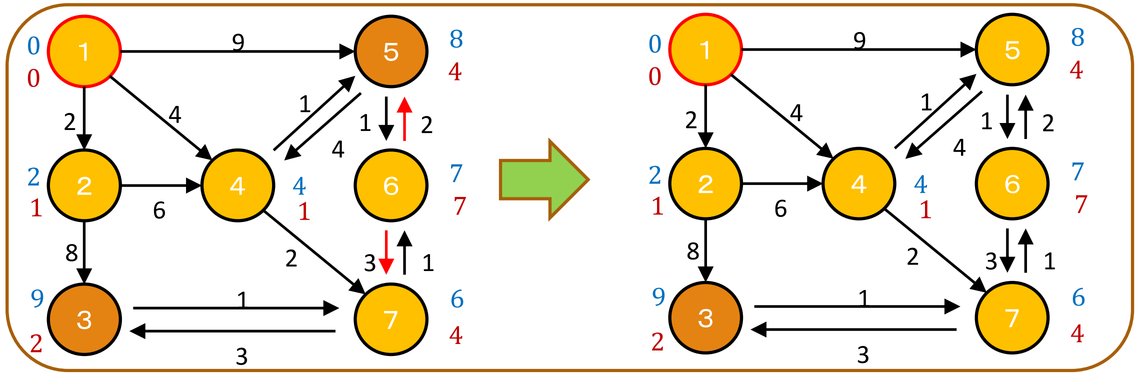


$i = 7$

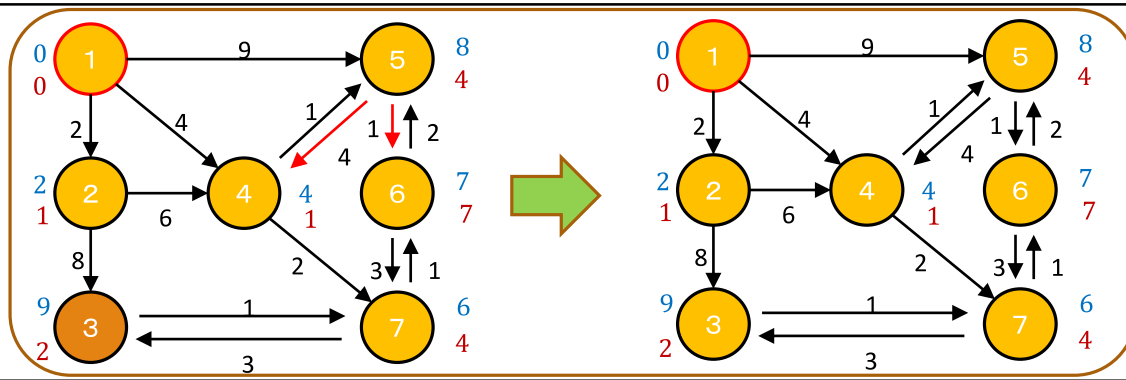


2.3 Dijkstra法の例題

$i = 6$

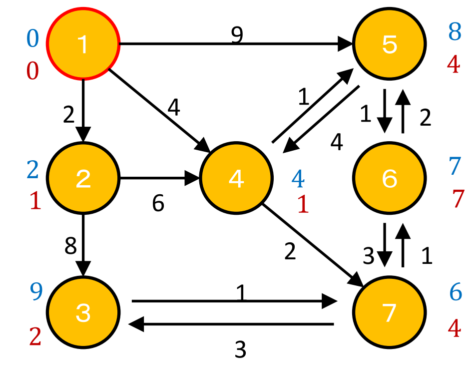


$i = 5$



終了

2.3 Dijkstra法の例題



＜最短経路の列挙＞

ポインタF_mを目印に遡っていく

・ノード6までの最短経路

$$6 \rightarrow (F_6 =)7 \rightarrow (F_7 =)4 \rightarrow (F_4 =)1$$

2.4 ヒープ構造

・最小値を効率的に求めるために用いるデータ構造

・ダイクストラ法の中で使われるデータ処理方法

ダイクストラ法のStep3

集合$\overline{K}$に属すすべてのノードの中での部分的最小費用が最小となっているノードを求め、これをノードjとする。ノードjを集合Kにうつす。

$$c_j = \min_p (c_p)\ (p \in \overline{K})$$

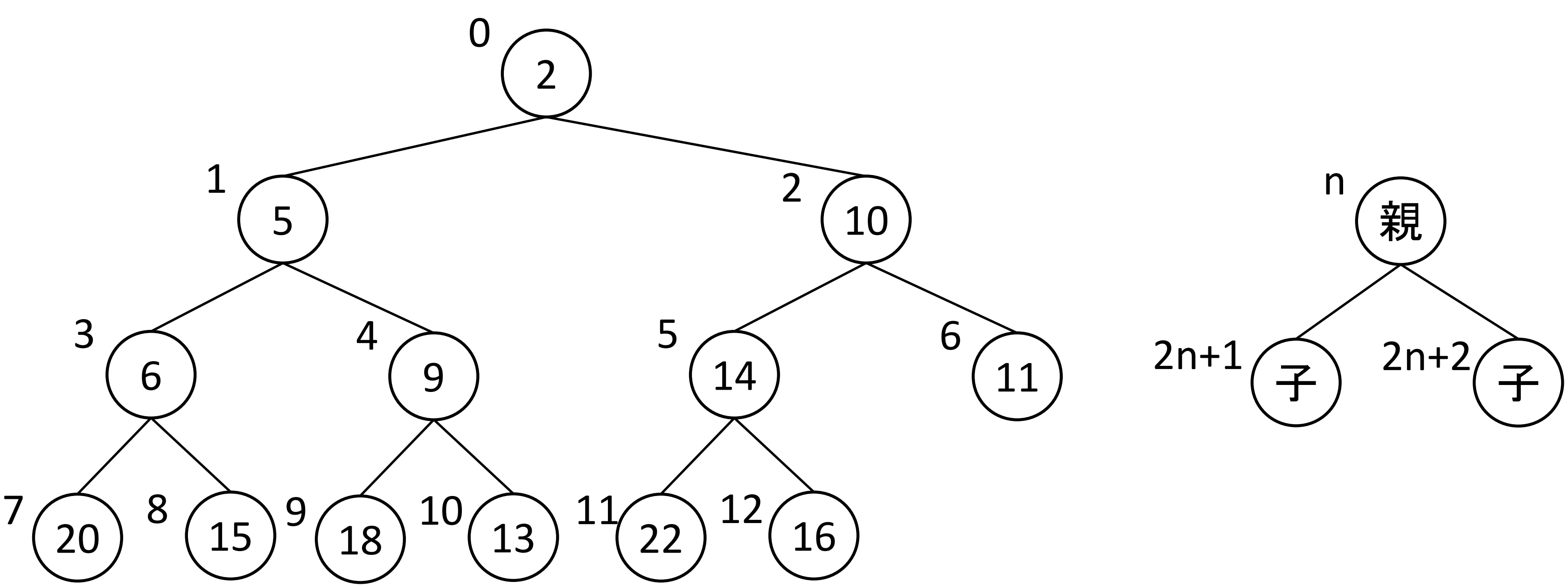
2.4 ヒープ構造

- 「ヒープ条件」を満たすように数字を並べる

インデックスn	0	1	2	3	4	5	6	7	8	9	10	11	12
配列B(n)	2	5	10	6	9	14	11	20	15	18	13	22	16

ヒープ条件

$$B(n) \geq B(2n+1), B(2n+2)$$

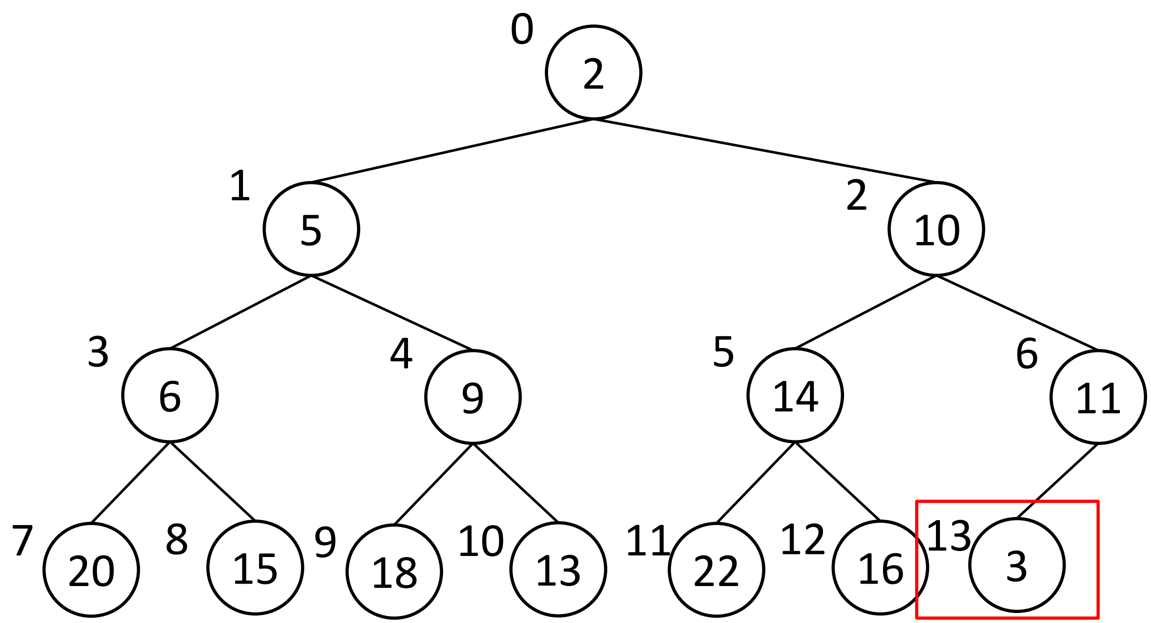


2.4 ヒープ構造

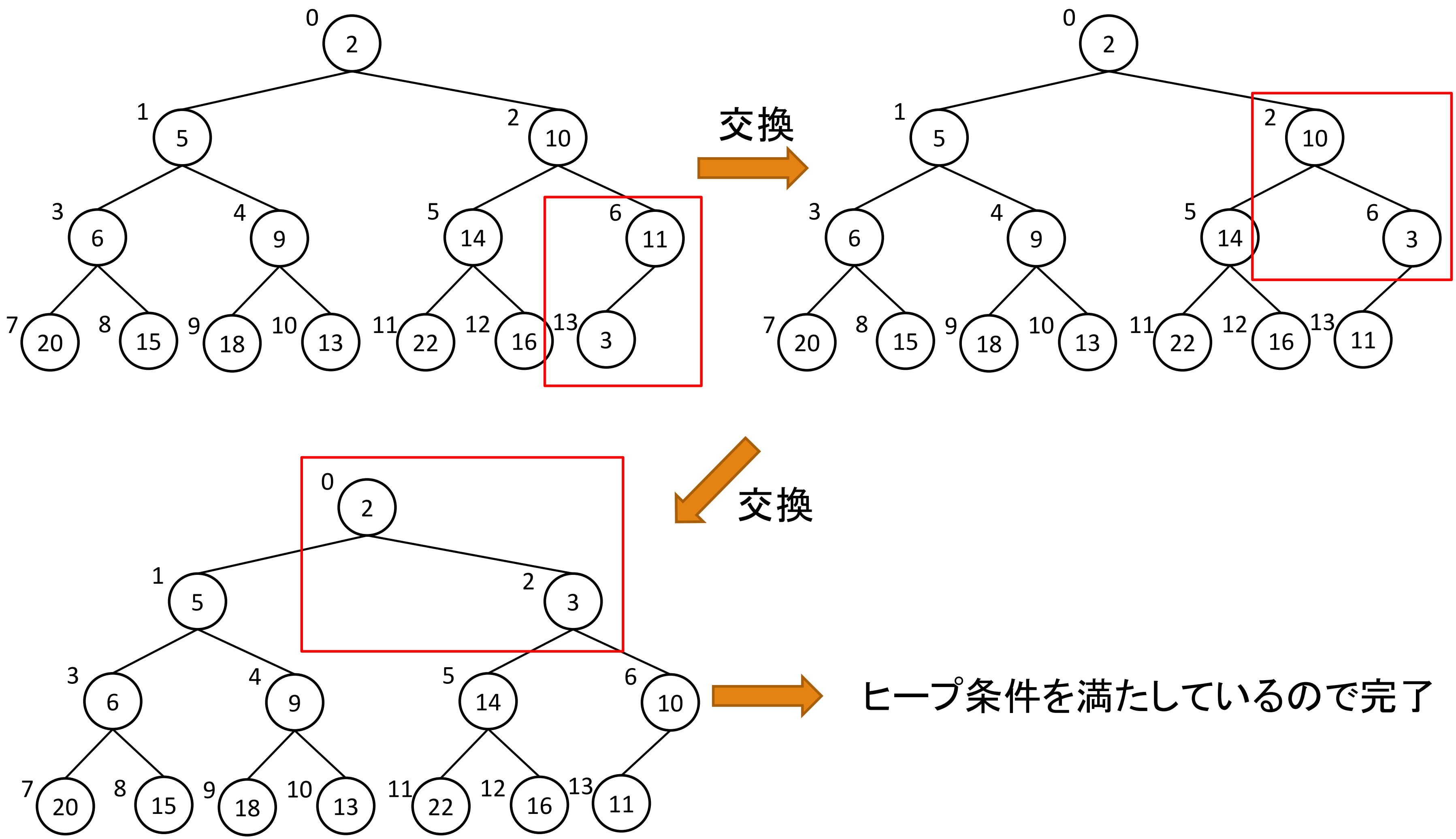
・数字の追加、削除が起きてもヒープ構造を維持

☆数字の追加

インデックスn	0	1	2	3	4	5	6	7	8	9	10	11	12	13
配列B(n)	2	5	10	6	9	14	11	20	15	18	13	22	16	3



2.4 ヒープ構造

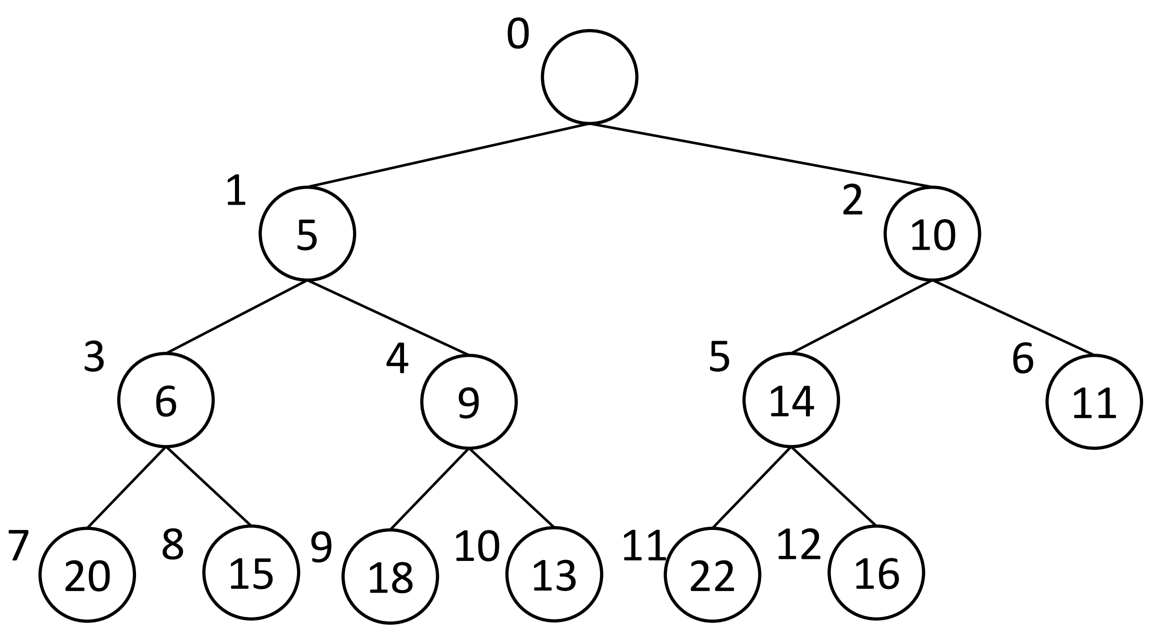


2.4 ヒープ構造

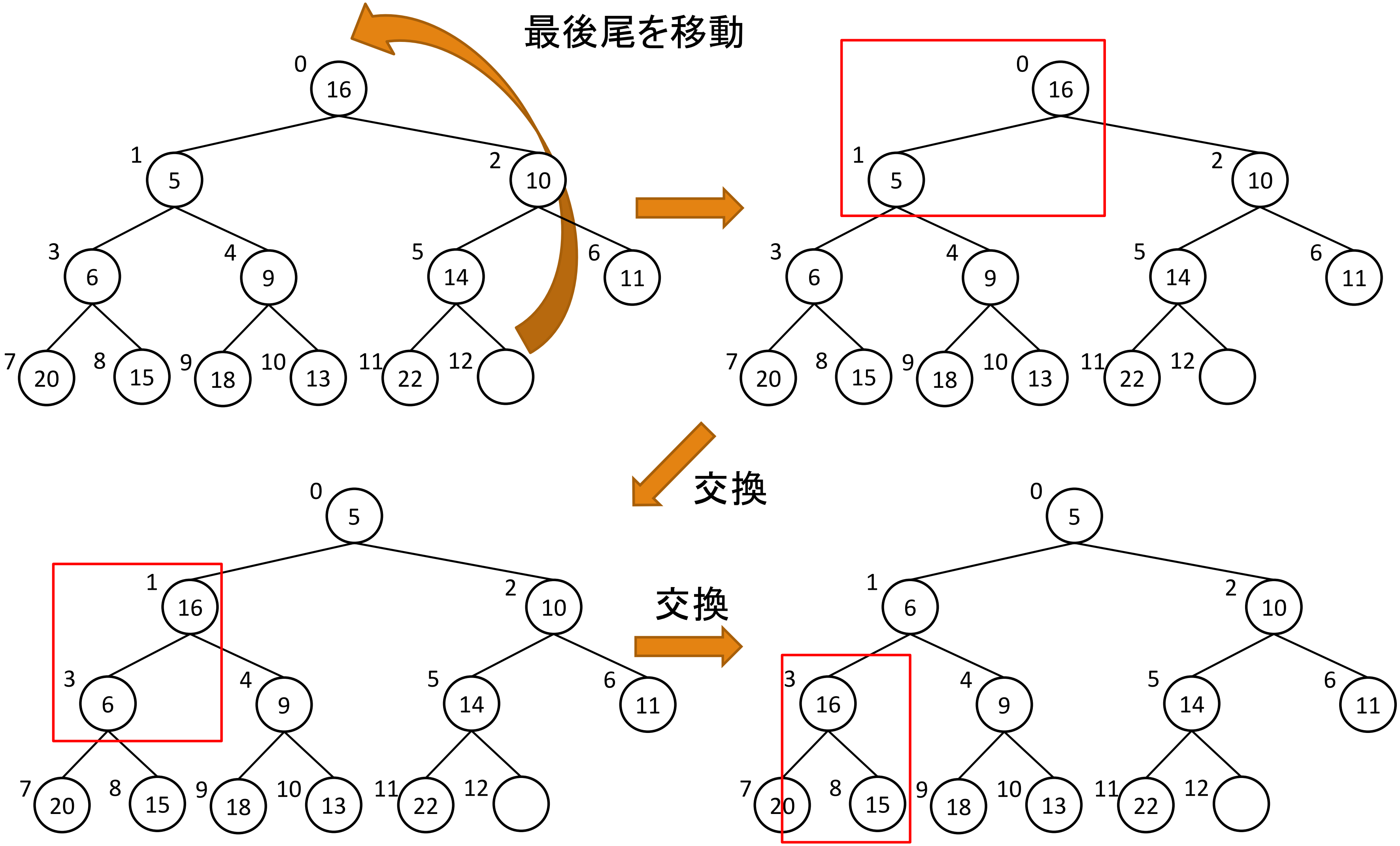
・数字の追加、削除が起きてもヒープ構造を維持

☆数字の削除（最小値データの取り出し）

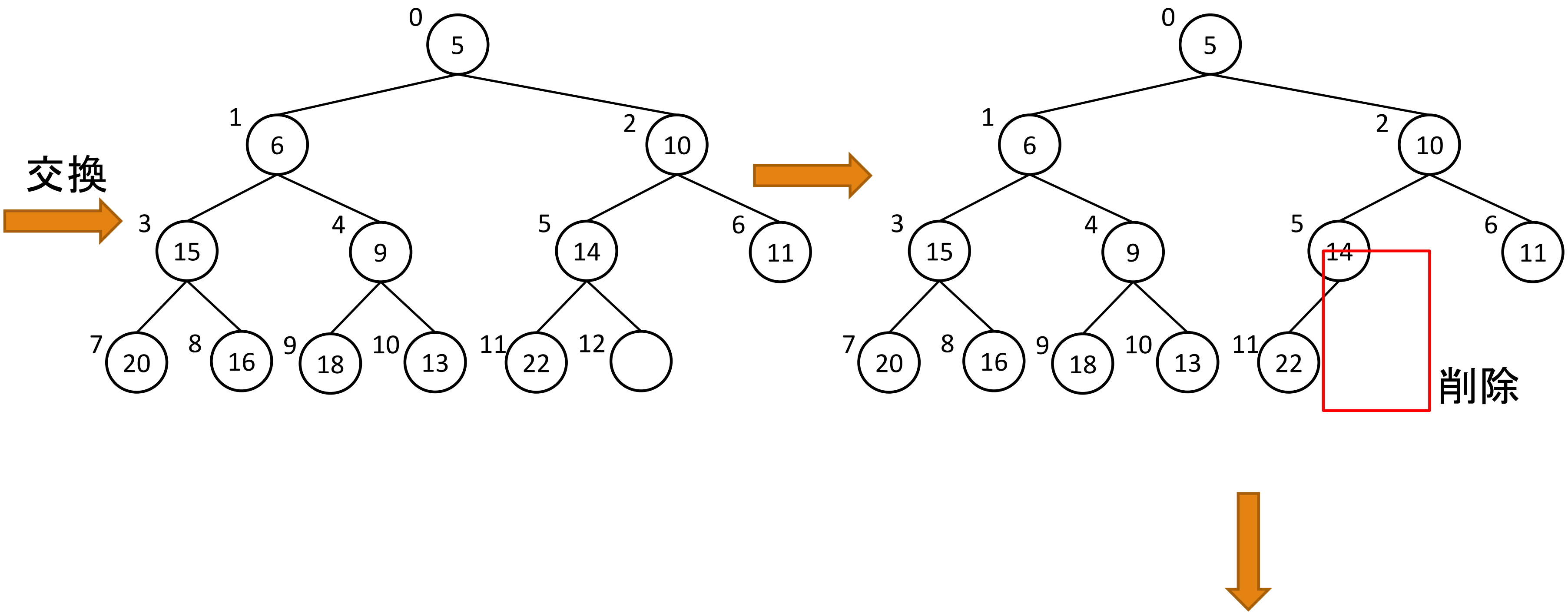
インデックスn	0	1	2	3	4	5	6	7	8	9	10	11	12
配列B(n)	~~2~~	5	10	6	9	14	11	20	15	18	13	22	16



2.4 ヒープ構造



2.4 ヒープ構造



ヒープ条件を満たしているので完了

今日の内容

1. 最適化問題（均衡配分モデルの導出における数学的手法）
2. 最短経路探索とダイクストラ法
3. 均衡配分法のアルゴリズム

3.1 非線形最適化問題の解法

・n回目の探索の結果、点$\boldsymbol{x_n}$が得られたとする

さらに降下した点$\boldsymbol{x_{n+1}}$を求めるにあたり必要なことは2つ

①どの方向に降下方向ベクトル$\boldsymbol{d}$を設定するか

降下方向ベクトル $\boldsymbol{d} = \boldsymbol{y} - \boldsymbol{x_n}$

（点$\boldsymbol{y}$: 点$\boldsymbol{x_n}$からさらに$Z(\boldsymbol{x})$を小さくする方向に存在する点）

②降下方向にどれくらい進むか

$$\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + \underbrace{\alpha}_{\text{ステップサイズ}} \boldsymbol{d} = \boldsymbol{x_n} + \boldsymbol{\alpha}(\boldsymbol{y} - \boldsymbol{x_n})$$

3.2 Frank-Wolfe法

・利用者均衡配分モデルの解法として広く用いられる

特徴

◎ 必要な記憶容量が少なくてすむ

◎ 手順が簡単なため、プログラムが作成しやすい

× 収束が進むと収束スピードが緩慢になる

3.3 Frank-Wolfe法のアルゴリズム

Step0

全リンクの交通量$x_a^0 = 0$とする。リンク旅行時間t_a^0を求める。
t_a^0をもとにしてall-or-nothing配分を行い、x_a^1を設定。
リンク交通量x_a^1に対するリンク旅行時間t_a^1を計算する。
ODペアごとの最小旅行時間u_{rs}^1も求める。

Step1

リンク旅行時間がt_a^nの状態でall-or-nothing配分。そのときの交通量：y_a^n
降下方向ベクトルd_a^nを計算する・・・$d_a^n = y_a^n - x_a^n$

Step2

次ステップの交通量：$x_a^{n+1} = x_a^n + \alpha^n d_a^n = x_a^n + \alpha^n (y_a^n - x_a^n)$

α^nの探索：$\displaystyle \min_{0 \le \alpha \le 1} \sum_a \int_0^{x_a^{n+1}} t_a(\omega) d\omega$

求まったx_a^{n+1}に対してt_a^{n+1}とu_{rs}^{n+1}を求める。

Step3

収束条件を満たしているか確認

$$\sum_{rs} \frac{|u_{rs}^{n+1} - u_{rs}^n|}{u_{rs}^{n+1}} \le \kappa$$ （あらかじめ定めた定数）

No：$n = n + 1$（Step1へ）

Yes：x_a^{n+1}を解とする

3.3 参考 : all-or-nothing配分のアルゴリズム

Step1 ノード番号iを始点とした最短経路探索を行う。

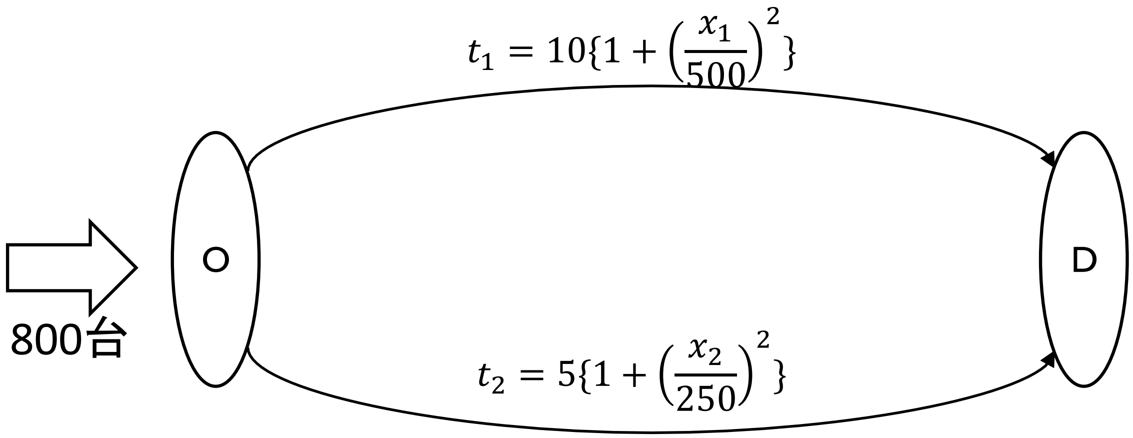
Step2 全OD間交通量のうち、ノードiをOriginとする交通量を、Step1の探索結果に基づき、各Destinationまでに通過するリンクに負荷。

Step3 始点ノードをすべて網羅したか確認

No 別のノードを始点にする

Yes:終了

3.4 Frank-Wolfe法を用いた配分例



3.4 Frank-Wolfe法を用いた配分例（$\kappa = 0.001$とする）

Step0 $u_{od}^1 = 10$

$x_1^1 = 0, t_1^1 = 10$

$x_2^1 = 800, t_2^1 = 56.2$

Step1

$y_1^1 = 800$

$d_1^1 = y_1^1 - x_1^1 = 800$

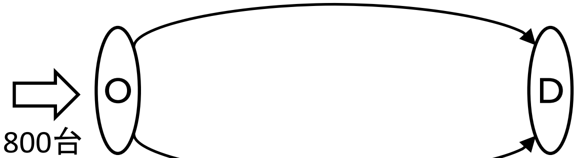
$y_2^1 = 0$

$d_2^1 = y_2^1 - x_2^1 = -800$

Step2 $u_{od}^2 = 16.8$

$x_1^2 = x_1^1 + \alpha^1 d_1^1 = 416$

$t_1^2 = 16.9$



$x_2^2 = x_2^1 + \alpha^1 d_2^1 = 384$

$t_2^2 = 16.8$

$$\boxed{\min_{0 \le \alpha \le 1} \sum_a \int_0^{x_a^1 + \alpha^1 d_a^1} t_a(w) dw} \Rightarrow \boxed{\alpha^1 = 0.52}$$

$$t_1 = 10\{1 + \left(\frac{x_1}{500}\right)^2\}$$

$$t_2 = 5\{1 + \left(\frac{x_2}{250}\right)^2\}$$

Step3

$$\frac{\left|u_{OD}^2 - u_{OD}^1\right|}{u_{OD}^1} = 0.68 > 0.001$$

⇒Step1からもう1回

3.4 Frank-Wolfe法を用いた配分例（$\kappa = 0.001$とする）

Step0

$u_{od}^1 = 10$

$x_1^1 = 0, t_1^1 = 10$

$x_2^1 = 800, t_2^1 = 56.2$

Step1

$y_1^1 = 800$

$d_1^1 = y_1^1 - x_1^1 = 800$

$y_2^1 = 0$

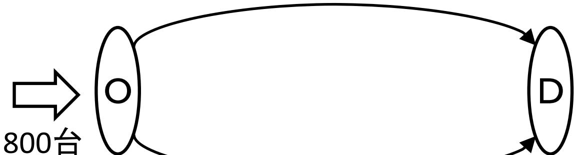
$d_2^1 = y_2^1 - x_2^1 = -800$

Step2

$u_{od}^2 = 16.8$

$x_1^2 = x_1^1 + \alpha^1 d_1^1 = 416$

$t_1^2 = 16.9$



$x_2^2 = x_2^1 + \alpha^1 d_2^1 = 384$

$t_2^2 = 16.8$

$$\boxed{\min_{0 \le \alpha \le 1} \sum_a \int_0^{x_a^1 + \alpha^1 d_a^1} t_a(w)\,dw} \Rightarrow \boxed{\alpha^1 = 0.52}$$

$$t_1 = 10\{1 + \left(\frac{x_1}{500}\right)^2\}$$

$$t_2 = 5\{1 + \left(\frac{x_2}{250}\right)^2\}$$

Step3

$$\frac{\left|u_{OD}^2 - u_{OD}^1\right|}{u_{OD}^1} = 0.68 > 0.001$$

⇒Step1からもう1回

3.4 Frank-Wolfe法を用いた配分例（$\kappa = 0.001$とする）

Step0 $u_{od}^1 = 10$

$x_1^1 = 0, t_1^1 = 10$

$x_2^1 = 800, t_2^1 = 56.2$

Step1

$y_1^1 = 800$

$d_1^1 = y_1^1 - x_1^1 = 800$

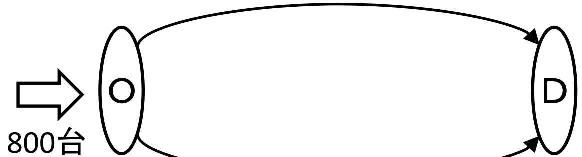
$y_2^1 = 0$

$d_2^1 = y_2^1 - x_2^1 = -800$

Step2 $u_{od}^2 = 16.8$

$x_1^2 = x_1^1 + \alpha^1 d_1^1 = 416$

$t_1^2 = 16.9$



$x_2^2 = x_2^1 + \alpha^1 d_2^1 = 384$

$t_2^2 = 16.8$

$$\boxed{\min_{0 \le \alpha \le 1} \sum_a \int_0^{x_a^1 + \alpha^1 d_a^1} t_a(w) dw} \Rightarrow \boxed{\alpha^1 = 0.52}$$

$$t_1 = 10\{1 + \left(\frac{x_1}{500}\right)^2\}$$

$$t_2 = 5\{1 + \left(\frac{x_2}{250}\right)^2\}$$

Step3

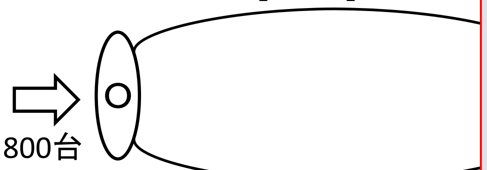
$$\frac{\left|u_{OD}^2 - u_{OD}^1\right|}{u_{OD}^1} = 0.68 > 0.001$$

⇒Step1からもう1回

3.4 Frank-Wolfe法を用いた配分例（$\kappa = 0.001$とする）

Step0 $u_{od}^1 = 10$

$x_1^1 = 0, t_1^1 = 10$

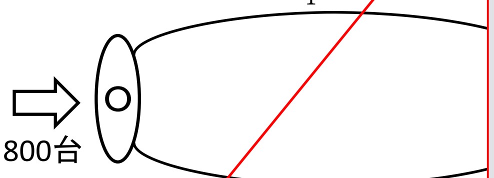


$x_2^1 = 800, t_2^1 = 56.2$

Step2 $u_{od}^2 = 16.8$

$x_1^2 = x_1^1 + \alpha^1 d_1^1 = 416$

$t_1^2 = 16.9$



$x_2^2 = x_2^1 + \alpha^1 d_2^1 = 384$

$t_2^2 = 16.8$

```
> f<-function(s){
+   g1<-function(x){
+     10*(1+(x/500)^2)
+   }
+   g2<-function(x){
+     5*(1+(x/250)^2)
+   }
+   a<-integrate(g1,0,800*s)
+   names(a)<-c('value','abs.error','subdivisions','message','call')
+   a1<-a$value
+   b<-integrate(g2,0,800*(1-s))
+   names(b)<-c('value','abs.error','subdivisions','message','call')
+   b1<-b$value
+   a1+b1
+ }
> optim(0.5,f,NULL,method="L-BFGS-B",lower=0,upper=1)
$par
[1] 0.5183413

$value
[1] 8549.75

$counts
function gradient
       4        4

$convergence
[1] 0

$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"

> |
```

0.001

$$\min_{0 \le \alpha \le 1} \sum_a \int_0^{x_a^1 + \alpha^1 d_a^1} t_a(w)dw \Rightarrow \boxed{\alpha^1 = 0.52}$$

$$t_1 = 10\{1 + \left(\frac{x_1}{500}\right)^2\}$$

$$t_2 = 5\{1 + \left(\frac{x_2}{250}\right)^2\}$$