

# 最短経路探索ゼミ

---

交通研B4

飯塚卓哉 出原昇馬 清水大暉

# 目次

---

- Dijkstra法
- Bellman-Ford法
- A\*法
- 各アルゴリズムの比較

## Dijkstra法：最短経路探索法のひとつ

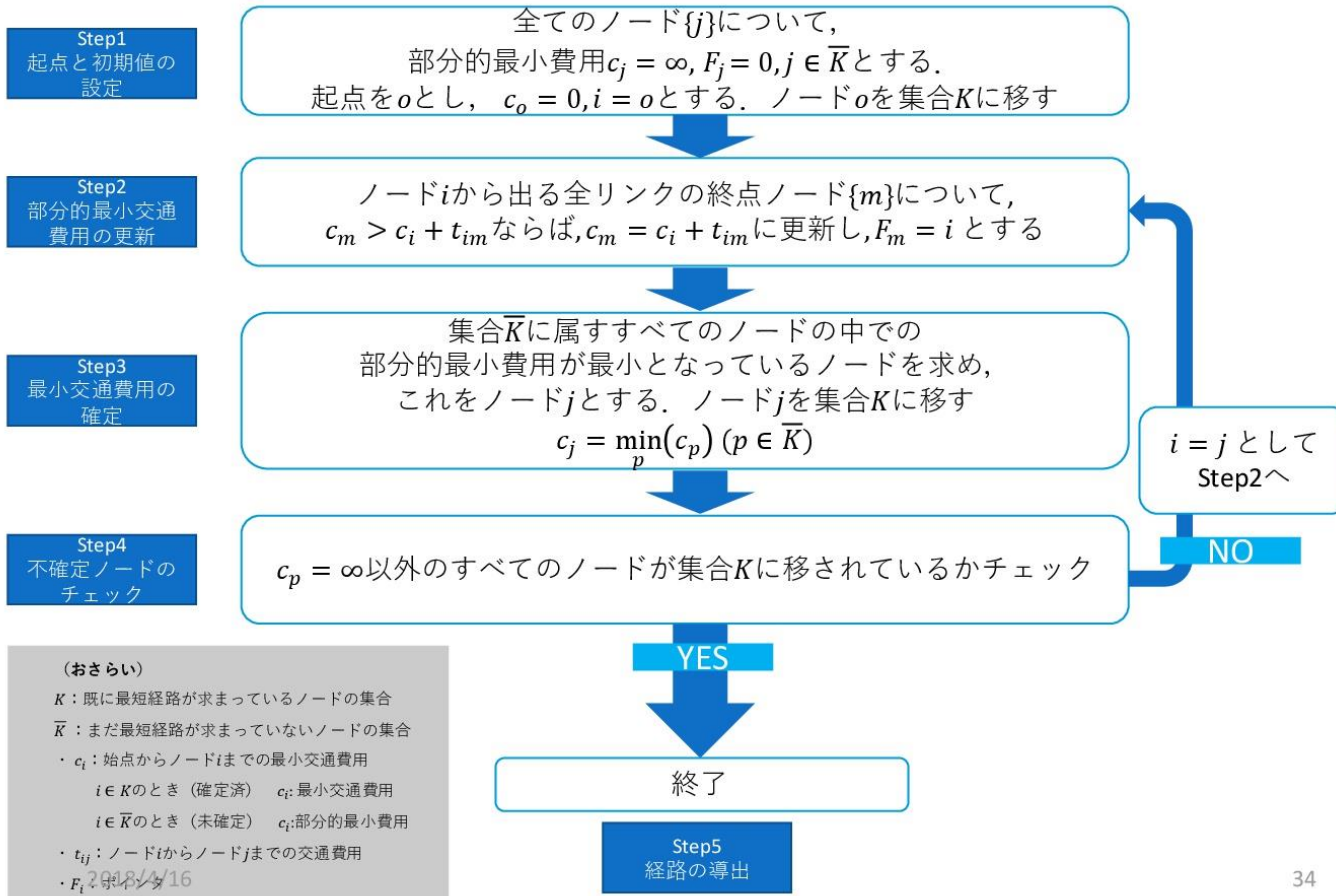
「最短とわかっているところから確定していく」

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 $\bar{K}$ のどちらかに分類できる

各ノード $i$ について以下の変数を定めます。

- $c_i$ ：始点からノード $i$ までの最小交通費用  
 $i \in K$ のとき（確定済）  $c_i$ : 最小交通費用  
 $i \in \bar{K}$ のとき（未確定）  $c_i$ : 部分的最小費用...ここまでの最小費用
- $t_{ij}$ ：ノード $i$ からノード $j$ までの交通費用
- $F_i$ ：ポインタ（最短経路を列挙する際に使う。後述）

## アルゴリズム

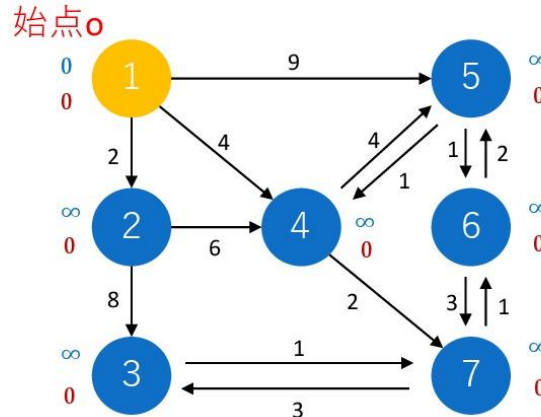


# Dijkstra法

## Step1 起点と初期値の設定

全てのノード $\{j\}$ について、  
 部分的最小費用 $c_j = \infty, F_j = 0, j \in \bar{K}$ 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

<Step1>  
 $i = 1$



(おさらい)

- $K$ : 既に最短経路が求まっているノードの集合
- $\bar{K}$ : まだ最短経路が求まっていないノードの集合
- $c_i$ : 起点からノード $i$ までの最小交通費用
  - $i \in K$ のとき (確定済)  $c_i$ : 最小交通費用
  - $i \in \bar{K}$ のとき (未確定)  $c_i$ : 部分的最小費用
- $t_{ij}$ : ノード $i$ からノード $j$ までの交通費用
- $F_i$ : ポインタ

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	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$	0	0	0	0	0	0	0

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

# Dijkstra法

## Step2 部分的最小交通費用の更新

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

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 $c'_m < c_m$ なので, すべて更新  
 また,  $F_m = i = 1$ とする

(おさらい)

$K$ : 既に最短経路が求まっているノードの集合

$\bar{K}$ : まだ最短経路が求まっていないノードの集合

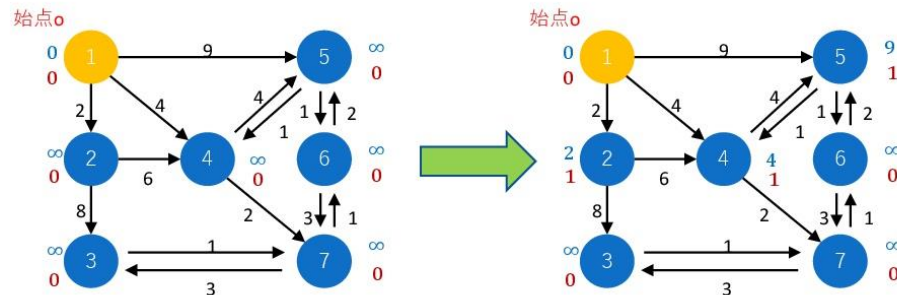
・  $c_i$ : 始点からノード*i*までの最小交通費用

$i \in K$ のとき (確定済)  $c_i$ : 最小交通費用

$i \in \bar{K}$ のとき (未確定)  $c_i$ : 部分的最小費用

・  $t_{ij}$ : ノード*i*からノード*j*までの交通費用

・  $F_i$ : ポインタ



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)

## Step3 最小交通費用の確定

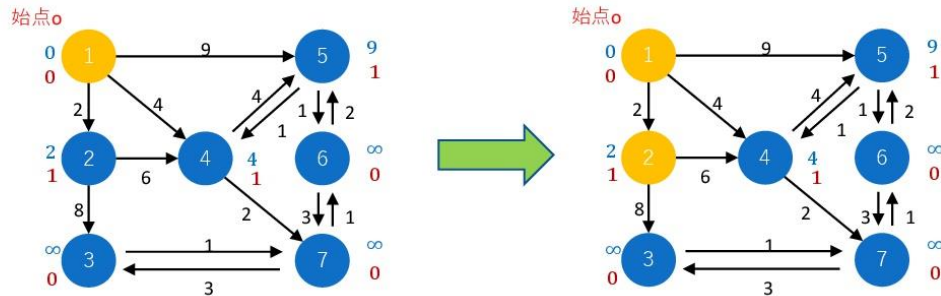
集合 $\bar{K}$ に属すすべてのノードの中での部分的最小費用が最小となっているノードを求め、これをノード $j$ とする。ノード $j$ を集合 $K$ に移す

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

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

→  $j = 2$ として、ノード2を集合 $K$ へ移す

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



(おさらい)

- $K$  : 既に最短経路が求まっているノードの集合
- $\bar{K}$  : まだ最短経路が求まっていないノードの集合
- $c_i$  : 始点からノード $i$ までの最小交通費用
  - $i \in K$  のとき (確定済)  $c_i$  : 最小交通費用
  - $i \in \bar{K}$  のとき (未確定)  $c_i$  : 部分的最小費用
- $t_{ij}$  : ノード $i$ からノード $j$ までの交通費用
- $E_i$  : ポインタ

# Dijkstra法

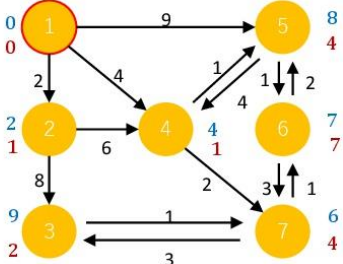
## Step4 不確定ノードのチェック

$c_p = \infty$ 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

$c_p \neq \infty$ かつ $p \in \bar{K}$ なる $p$ が残っている

→次は $i = 2$ でstep2~step4を行う

→  $i = 4 \rightarrow i = 7 \rightarrow i = 6 \rightarrow i = 5 \rightarrow$ 終了



## Step5 最短経路の導出

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

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

(おさらい)

- $K$ : 既に最短経路が求まっているノードの集合
- $\bar{K}$ : まだ最短経路が求まっていないノードの集合
- $c_i$ : 始点からノード $i$ までの最小交通費用
  - $i \in K$ のとき (確定済)  $c_i$ : 最小交通費用
  - $i \in \bar{K}$ のとき (未確定)  $c_i$ : 部分的な最小費用
- $t_{ij}$ : ノード $i$ からノード $j$ までの交通費用
- $F_i$ : ポインタ



# Bellman-Ford法

---

## Bellman-Ford法とは

- ・ 重み付き有向グラフにおいて単一始点の最短経路を解くアルゴリズム
- ・ リンクコストが負でも計算可
  - ・ リンクコストが負の場合ダイクストラでは誤った結果が出る

# Bellman-Ford法

## 計算の流れ

### 1. 初期設定

- ・  $d$  : 始点から各ノードへの暫定最短距離を格納. 初期は  $[0, \infty, \dots, \infty]$
- ・  $p$  : 始点から各ノードへ至るとき直前に通るノード(親ノード)を格納. 初期は  $[None, \dots, None]$

### 2. 繰り返し計算(計算量: $O(|Vertex| |Edge|)$ )

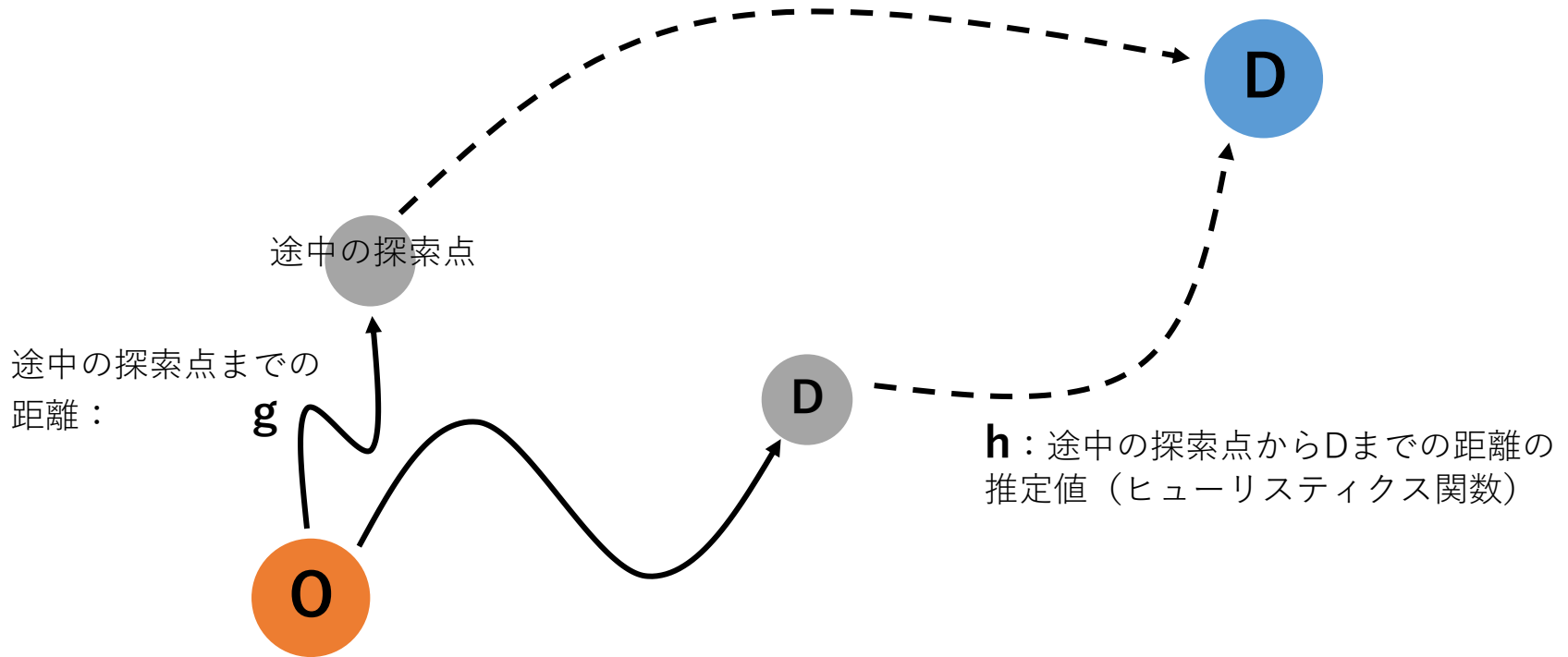
- ・ あるノード  $u$  から1回で到達可能な全てのノード  $v$  に対し,  
$$d[v] > d[u] + (u \rightarrow v \text{間コスト})$$

なら  $d[v]$  を  $d[u] + (u \rightarrow v \text{間コスト})$  に,  $p[v]$  を  $u$  に更新

- ・ それを全てのノードについて繰り返し
- ・ このループを高々  $(V - 1)$  回行えば全てのノードの最短距離, 最短経路が求まる
  - ・ 始点  $u_0$  から終点  $u_V$  まで最短経路を辿る際に, 全てのノードを経由し, かつ全ての  $i$  に対し  $i$  回目のループで順次  $u_i$  が更新されてゆく場合, ループは  $(V - 1)$  回行われることになる

# A\*法

: Dijkstra法の改良版. 探索順をより効率良くする



$f = g + h$  が小さい点を取りだして探索する

→  $h$  をある程度妥当性の高い関数として与えると, これはDijkstra法よりも効率の良いアルゴリズムとなる

都市においては, ユークリッド距離やマンハッタン距離が $h$ の値としてよく用いられる

# 各アルゴリズムの比較

渋谷の実ネットワークをもとに計算

A\*法 2.59 s

```
([562, 563, 564, 565, 559, 558, 557, 185, 929, 517, 1153, 921, 920, 540, 514, 924, 780, 473, 1037, 444, 445, 446,  
448, 449, 451, 452, 453, 719, 1320, 1321, 1322, 1324, 1323, 1330, 1331, 1335, 1336, 1338, 1339, 1353, 1354], 1957.0)  
2.5937299728393555
```

Bellmanford法 3.23 s

```
(出発地, 到着地, 最短距離, 経路)  
(562, 1354, 1957, ['562', '563', '564', '565', '559', '558', '557', '185', '929', '517', '1153', '921',  
'920', '540', '514', '924', '780', '473', '1037', '444', '445', '446', '448', '449', '451', '452', '453', '719',  
'1320', '1321', '1322', '1324', '1323', '1330', '1331', '1335', '1336', '1338', '1339', '1353', '1354'])  
3.234361171722412
```

Dijkstra法 3.68 s

```
562 1354  
([562, 563, 564, 565, 559, 558, 557, 185, 929, 517, 1153, 921, 920, 540, 514, 924, 780, 473, 1037, 444, 445, 446,  
448, 449, 451, 452, 453, 719, 1320, 1321, 1322, 1324, 1323, 1330, 1331, 1335, 1336, 1338, 1339, 1353, 1354], 1957.0)  
3.687467575073242
```

A\*の勝利！！