スタートアップゼミ課題発表 #1

最短経路探索

B4 渡邉 葵

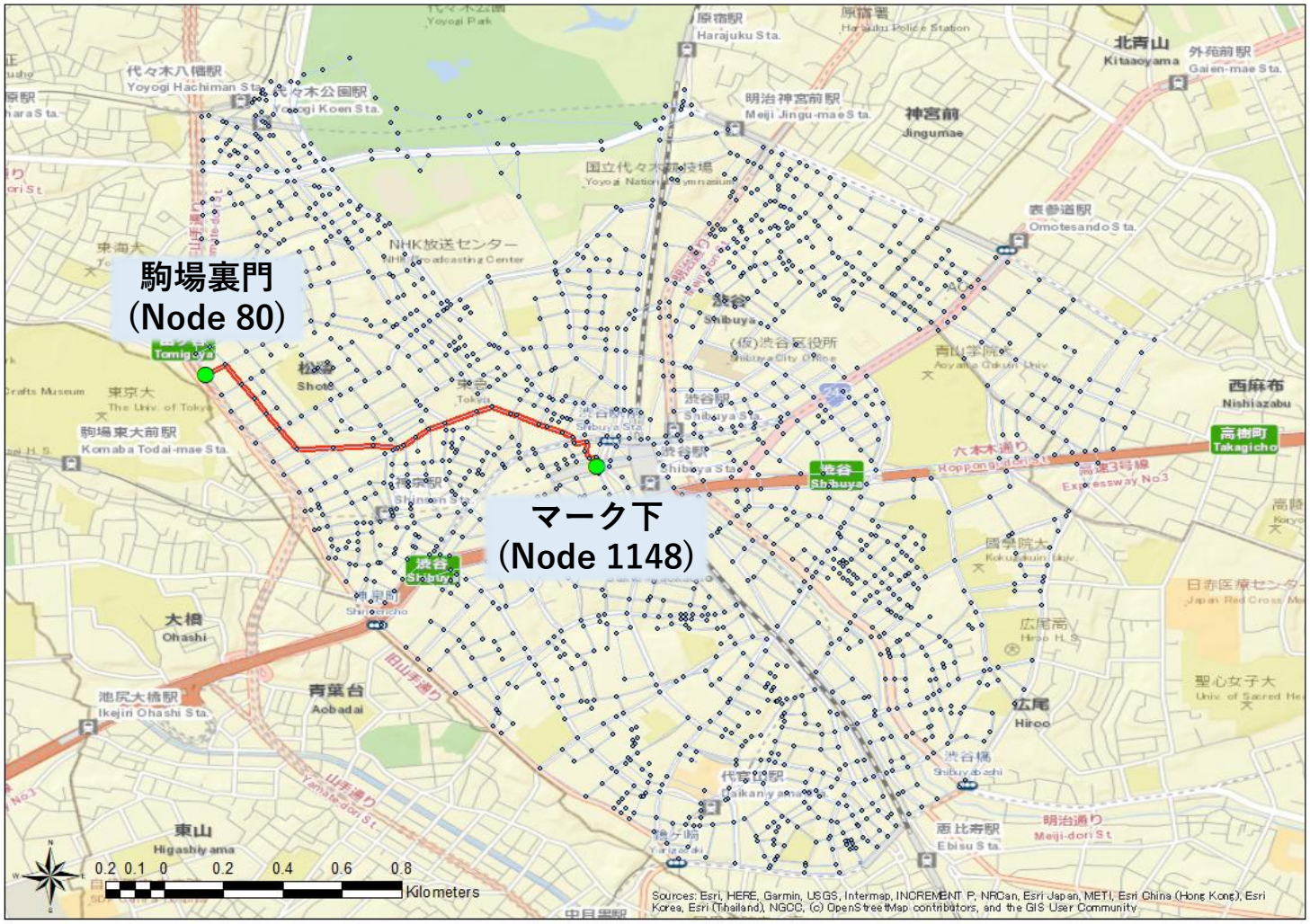
目次

1. 結果の可視化
2. アルゴリズムの比較
3. 言語の比較

1. 結果の可視化

1. 結果の可視化

- 最短経路探索によって得られた結果をGIS上で可視化した



1. 結果の可視化

- 最短経路探索によって得られた結果をGIS上で可視化した

1. 結果の可視化

- 最短経路探索によって得られた結果をGIS上で可視化した

2. アルゴリズムの比較

2. アルゴリズムの比較

- Bellman-Ford法について

概要　1点から他のすべての点への最短距離を計算
長所　リンクコストが負でも計算できる
短所　計算が遅い $O(V \times E)$

グラフ　$G = (V, E)$

初期化　始点から各ノードへの暫定最短距離　$d = [0, \infty, \cdots, \infty]$
始点から各ノードへ至るときの直前ノード　$p = [None, \cdots, None]$

ループ　ノードuに着目し，$u \to v$ と見ていく

uと隣接するノードの集合

$$
\begin{aligned}
&{}^{\forall}u \in V,\ {}^{\forall}v \in U';\\
&\quad \textbf{if} \quad d[v] > d[u] + c_{uv}\\
&\quad \textbf{then} \quad d[v] = d[u] + c_{uv}\\
&\qquad\qquad\ \ p[v] = u
\end{aligned}
$$

$u \to v$ のコスト

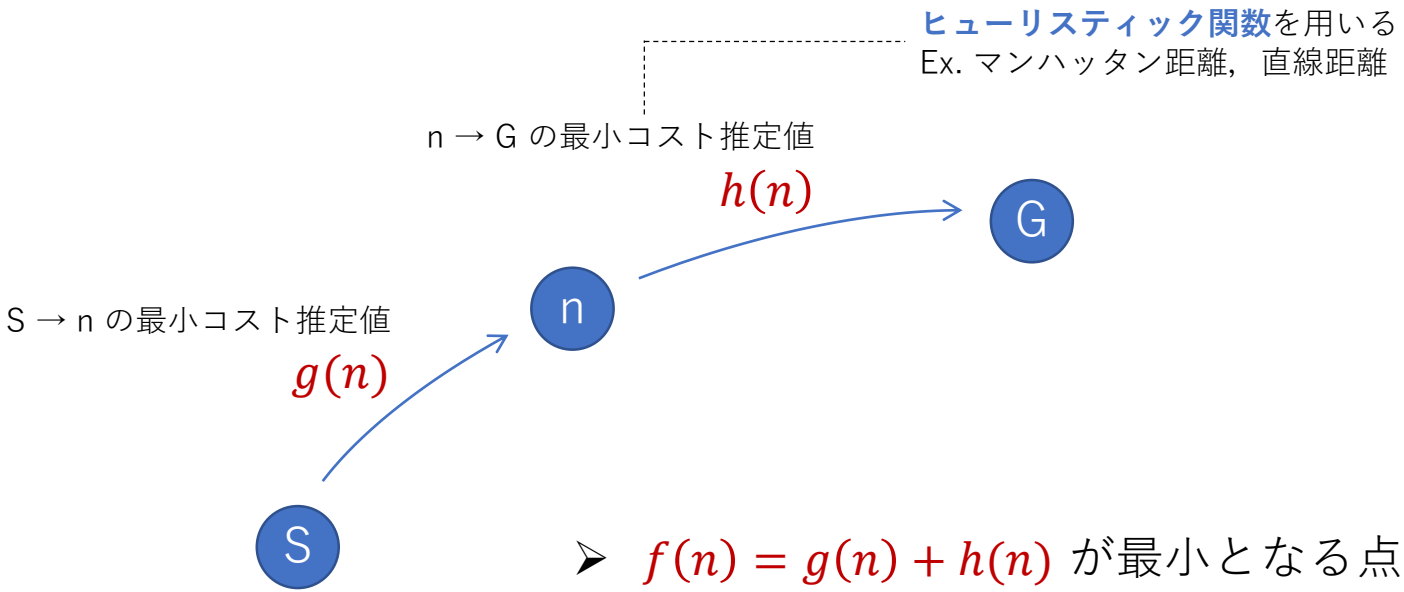
高々 $(V-1)$ 回

計算量　$O(V \times E)$

2. アルゴリズムの比較

- A*法について

概要　1点からある点($S \to \mathrm{G}$)への最短距離を計算
長所　計算が速い
短所　負のリンクコストは計算できない



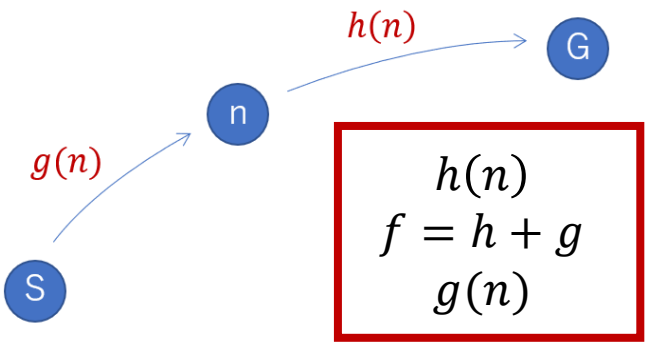
➢ $f(n) = g(n) + h(n)$ が最小となる点を探していくことでGに到達

2. アルゴリズムの比較

- A*法について

具体例：始点ノード S → 終点ノード G，上下移動のみ
計算中ノード$OPEN$，計算済みノード$CLOSED$

$OPEN$　　$CLOSED$



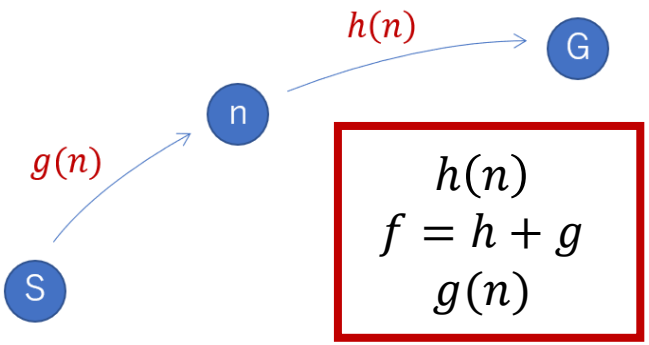
			G
	S		

2. アルゴリズムの比較

- A*法について

具体例：始点ノード S → 終点ノード G，上下移動のみ
計算中ノード$OPEN$，計算済みノード$CLOSED$

$OPEN$　　$CLOSED$



			G
	3 3 0		

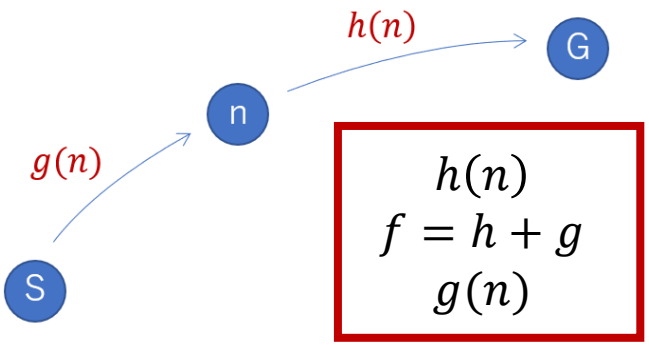
始点ノードを$OPEN$にうつす

2. アルゴリズムの比較

- A*法について

具体例：始点ノード S → 終点ノード G，上下移動のみ
計算中ノード$OPEN$，計算済みノード$CLOSED$

$OPEN$　　$CLOSED$



	2 3 1		G
4 5 1	3 3 0	2 3 1	
	4 5 1		

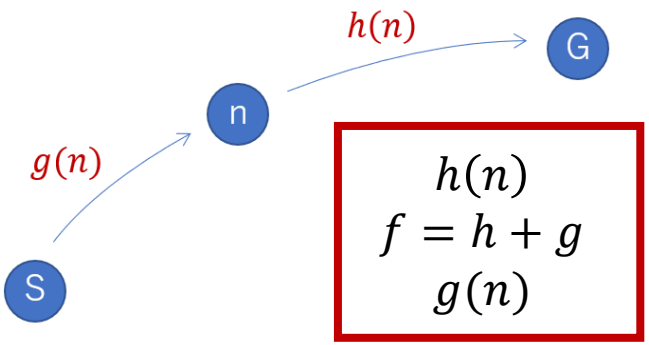
$OPEN$内で最小の$f(n)$をもつノードnを$CLOSED$にうつし，隣接ノードの $f(n)$を計算

2. アルゴリズムの比較

- A*法について

具体例：始点ノード S → 終点ノード G，上下移動のみ
計算中ノード$OPEN$，計算済みノード$CLOSED$

$OPEN$　　$CLOSED$



	2 3 1		G
4 5 1	3 3 0	2 **3** 1	
	4 5 1		

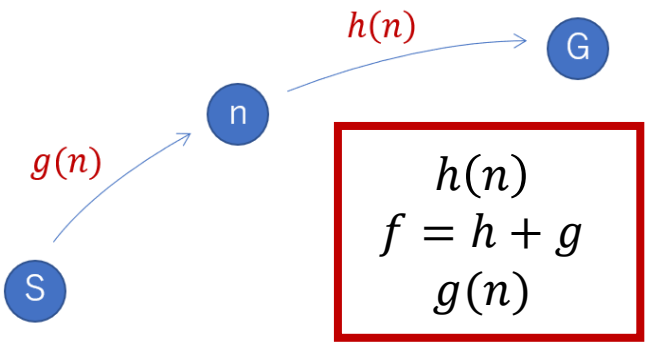
隣接ノード内で，$f(n)$が最小のノードを探し，$CLOSED$にうつして移動

2. アルゴリズムの比較

- A*法について

具体例：始点ノード S → 終点ノード G，上下移動のみ
計算中ノード$OPEN$，計算済みノード$CLOSED$

$OPEN$　　$CLOSED$



	2 3 1	1 3 2	G
4 5 1	3 3 0	2 3 1	1 3 2
	4 5 1	3 5 2	

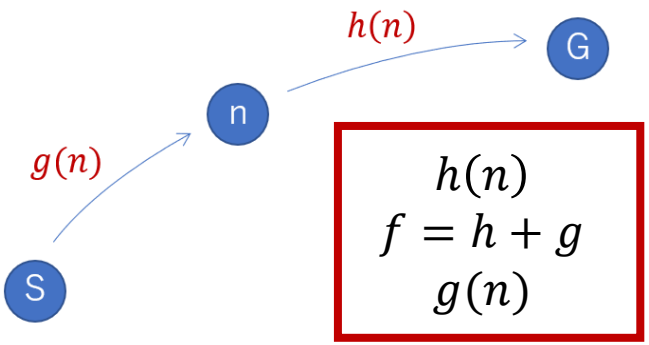
移動先のノードの隣接ノードの $f(n)$を計算

2. アルゴリズムの比較

- A*法について

具体例：始点ノード S → 終点ノード G，上下移動のみ
計算中ノード$OPEN$，計算済みノード$CLOSED$

$OPEN$　　$CLOSED$



	2 3 1	1 3 2	G
4 5 1	3 3 0	2 3 1	1 **3** 2
	4 5 1	3 5 2	

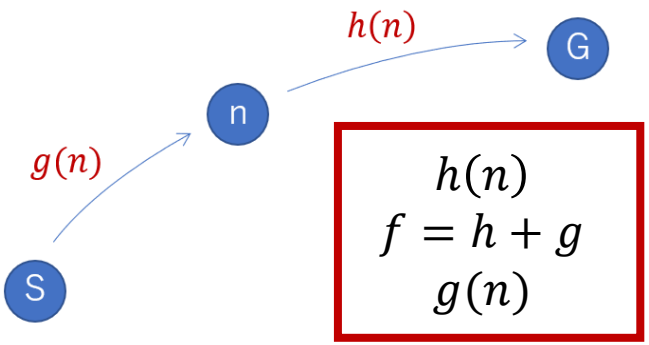
$OPEN$内で，$f(n)$が最小のノードを探し，$CLOSED$にうつして移動

2. アルゴリズムの比較

- A*法について

具体例：始点ノード S → 終点ノード G，上下移動のみ
計算中ノード$OPEN$，計算済みノード$CLOSED$

$OPEN$　　$CLOSED$



	2 3 1	1 3 2	0 3 3
4 5 1	3 3 0	2 3 1	1 3 2
	4 5 1	3 5 2	2 5 3

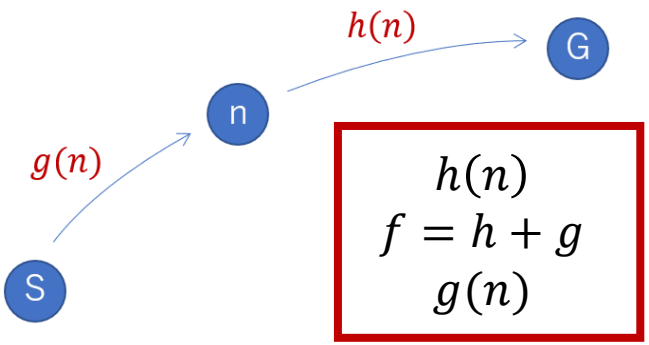
移動先のノードの隣接ノードの $f(n)$を計算

2. アルゴリズムの比較

- A*法について

具体例：始点ノード S → 終点ノード G，上下移動のみ
計算中ノード$OPEN$，計算済みノード$CLOSED$

$OPEN$　　$CLOSED$



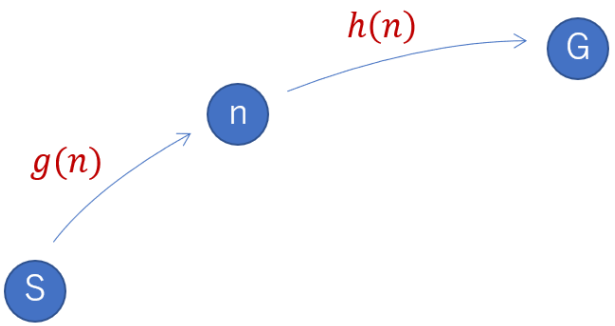
	2 3 1	1 3 2	0 **3** 3
4 5 1	3 3 0	2 3 1	1 3 2
	4 5 1	3 5 2	2 5 3

$OPEN$内で，$f(n)$が最小のノードを探し，$CLOSED$にうつして移動．これでゴール．

2. アルゴリズムの比較

- A*法について

始点ノード S → 終点ノード G
計算中ノード$OPEN$，計算済みノード$CLOSED$



初期化　$OPEN = \{S\}$,　$CLOSED = \phi$
$g(S) = 0$より，　$f(S) = h(S)$

ループ　1. $OPEN$内で最小の$f(n)$をもつノードnを取り出す → $n = D$なら終了
→ nを$CLOSED$にうつす

2. nに隣接するすべてのノードmに対して，　$f'(m) = g(n) + cost(n, m) + h(m)$ を計算

・$m \notin (OPEN \cup CLOSED)$
$f(m)$を代入し，mを$OPEN$に追加して，mの親をnに

・$m \in OPEN$
$f(m)$が小さくなっていたら更新し，mの親をnに

・$m \in CLOSED$
$f(m)$が小さくなっていたら更新し，　m を$OPEN$に移し，mの親をnに

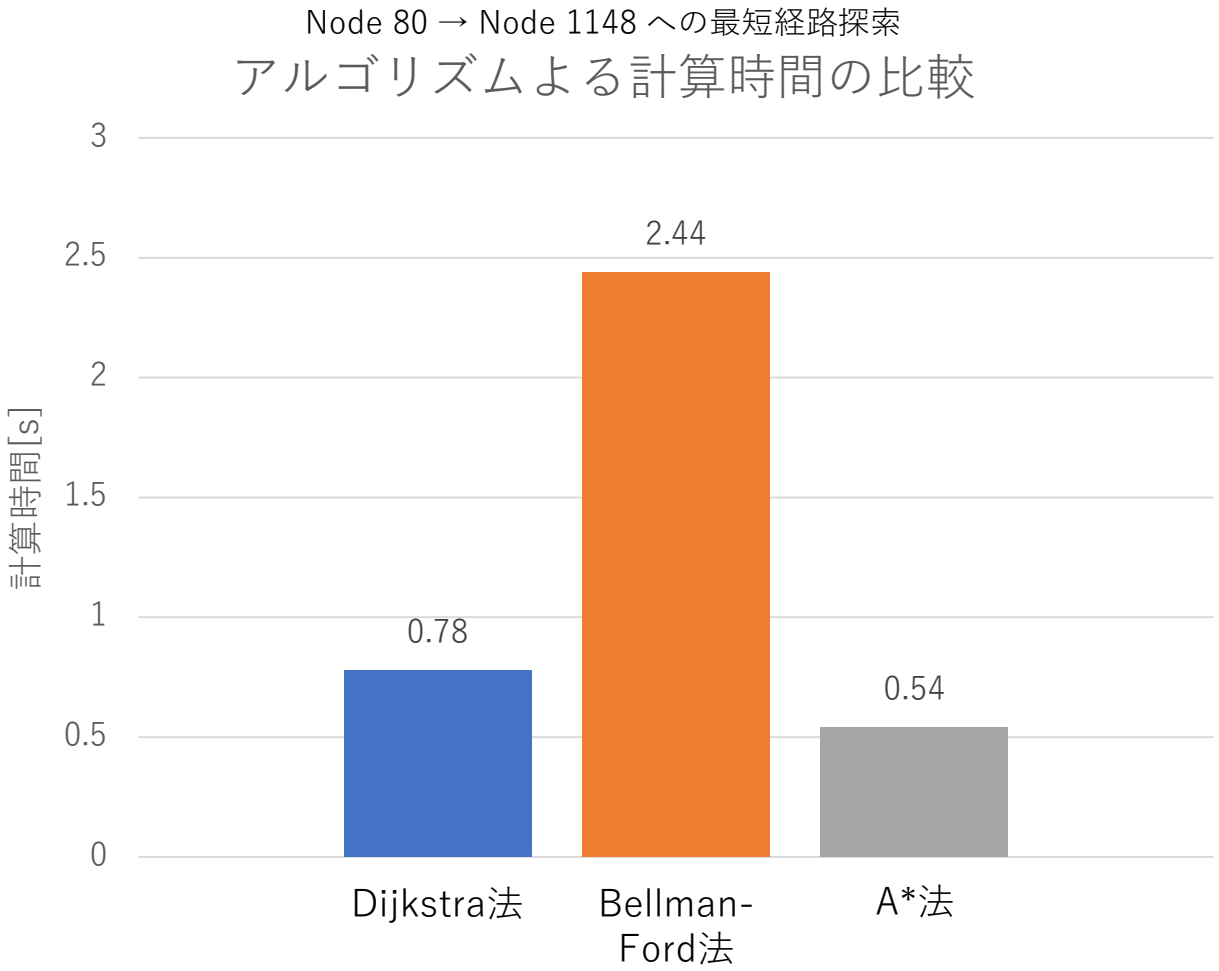
2. アルゴリズムの比較

- 各アルゴリズムの比較

	Dijkstra法	Bellman-Ford法	A*法
計算速度	△	×	○
負のリンクコスト	×	○	×
その他			特定のOD組に対しての 最短経路を出す

2. アルゴリズムの比較

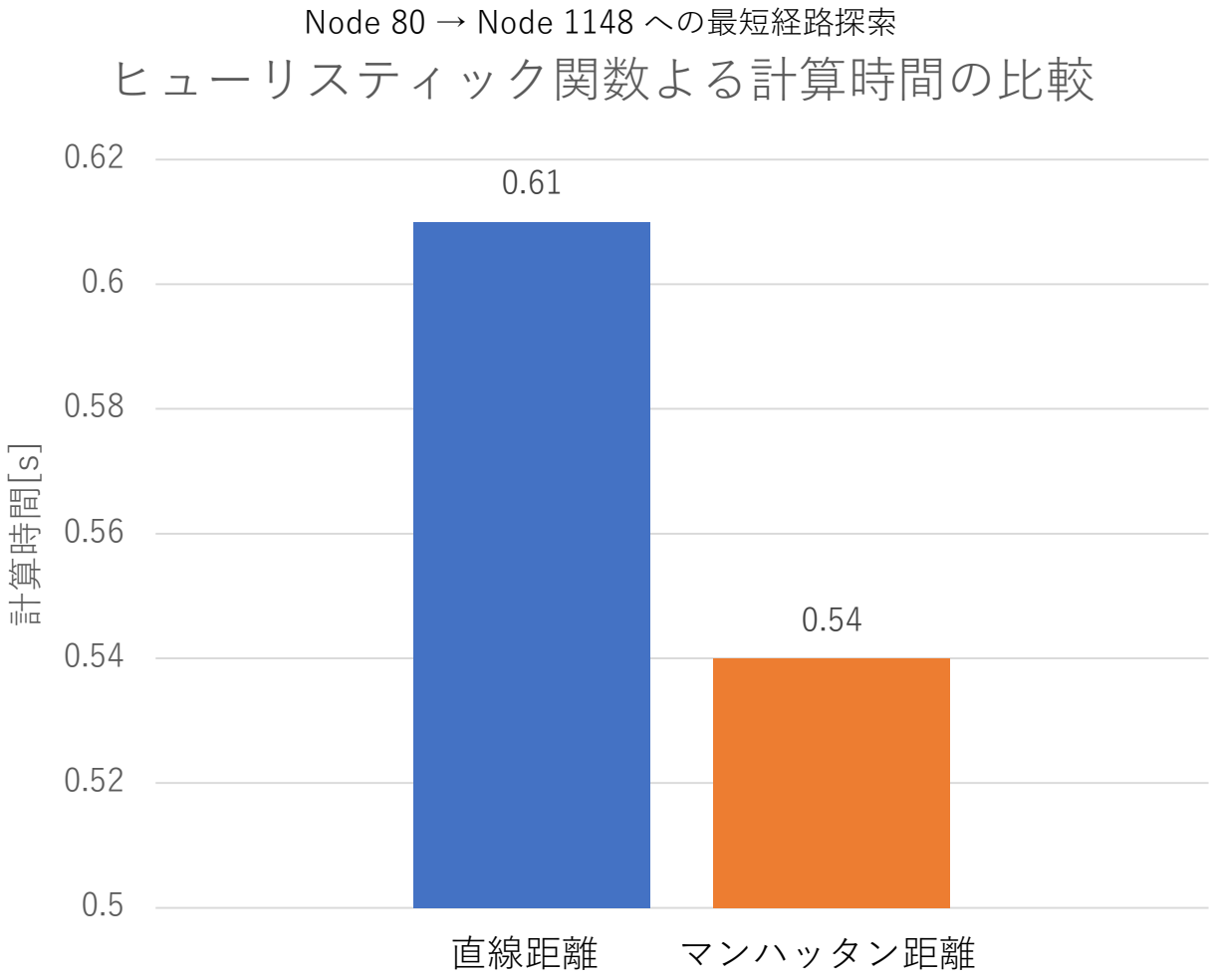
- アルゴリズムによる計算時間の比較



ここでの計算時間は，ODノードや隣接行列の取り込みなどは済ませた以降の，実際のアルゴリズムの部分の計算時間を指している．

2. アルゴリズムの比較

- A*法におけるヒューリスティック関数の設定



ここでの計算時間は，ODノードや隣接行列の取り込みなどは済ませた以降の，実際のアルゴリズムの部分の計算時間を指している．

2. アルゴリズムの比較

- 負のリンクコスト

2. アルゴリズムの比較

- 負のリンクコスト

2. アルゴリズムの比較

- 負のリンクコスト

- 災害が発生し，駒場からひとまず渋谷駅まで歩くことに(通常の最短経路は赤色)
- 途中で，松濤中学校で配給をやっているという情報を目にする
- **松濤中学校へのリンクコストが負の値に**

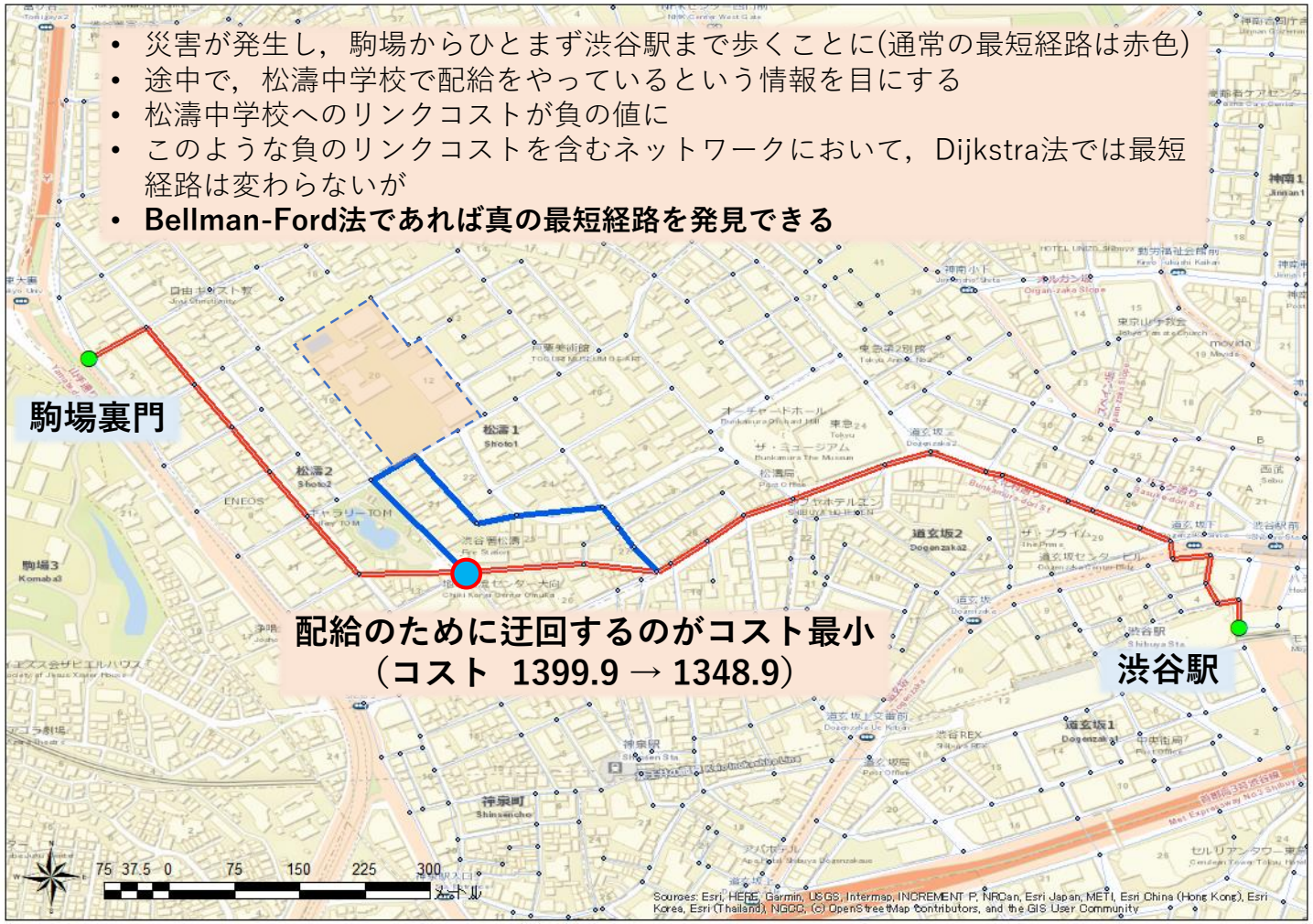
2. アルゴリズムの比較

- 負のリンクコスト

- 災害が発生し，駒場からひとまず渋谷駅まで歩くことに(通常の最短経路は赤色)
- 途中で，松濤中学校で配給をやっているという情報を目にする
- 松濤中学校へのリンクコストが負の値に
- **このような負のリンクコストを含むネットワークにおいて，Dijkstra法では最短経路は変わらないが**

2. アルゴリズムの比較

- 負のリンクコスト



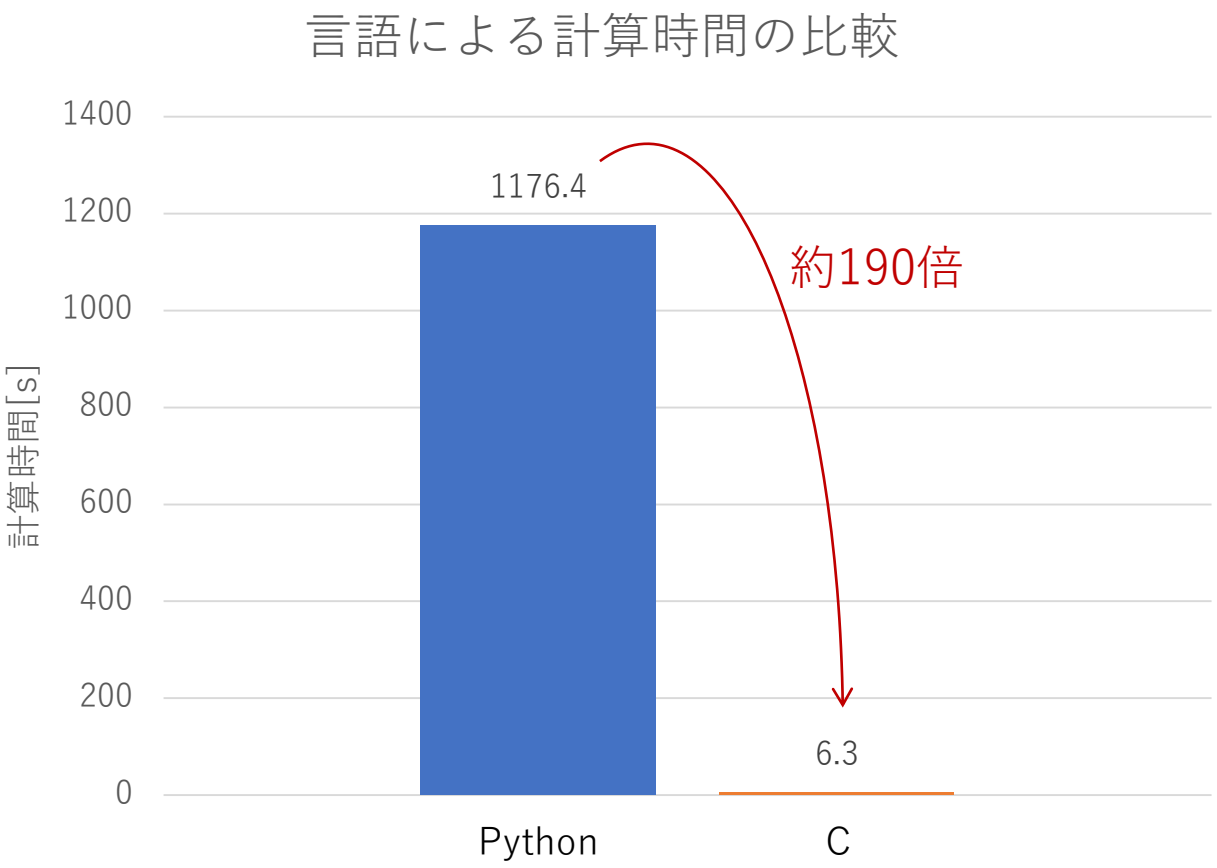
- 災害が発生し，駒場からひとまず渋谷駅まで歩くことに(通常の最短経路は赤色)
- 途中で，松濤中学校で配給をやっているという情報を目にする
- 松濤中学校へのリンクコストが負の値に
- このような負のリンクコストを含むネットワークにおいて，Dijkstra法では最短経路は変わらないが
- **Bellman-Ford法であれば真の最短経路を発見できる**

配給のために迂回するのがコスト最小
（コスト 1399.9 → 1348.9）

3. 言語の比較

3. 言語の比較

- Dijkstra法で，Node 931 から他のすべてのノードへの最短経路を列挙するのに要する時間



ここでの計算時間は，プログラム全体の計算時間を指している．