

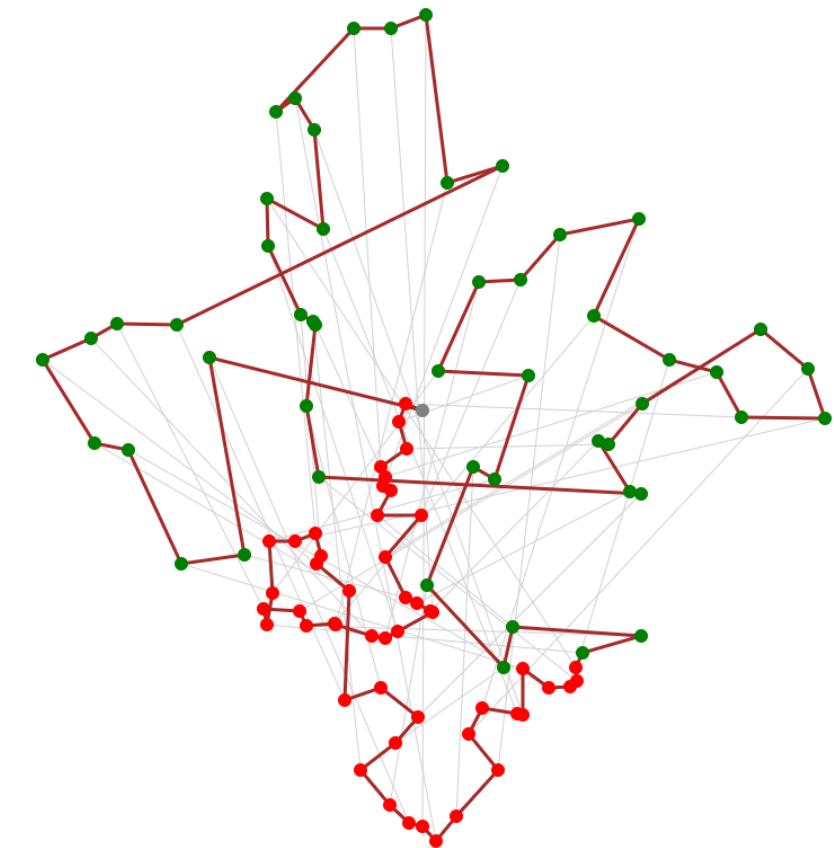
# AtCoder Heuristic First-step Vol.1

## ～山登り法・焼きなまし法編～

takumi152

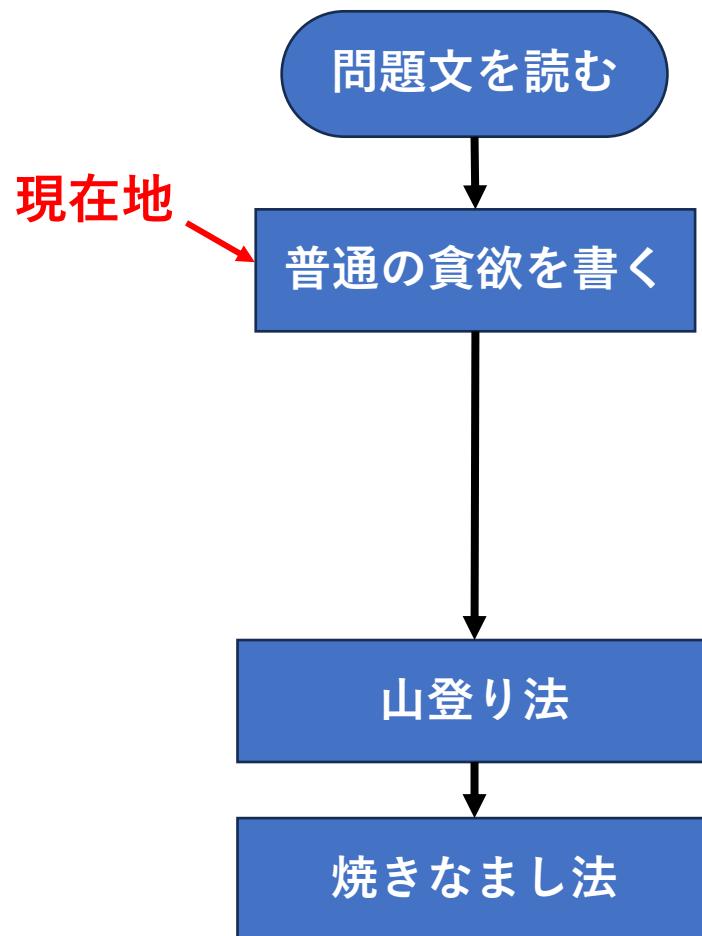
# 貪欲解から改善をしたい！

- ・貪欲でそこそこの解が求まった
- ・ここから何ができる？



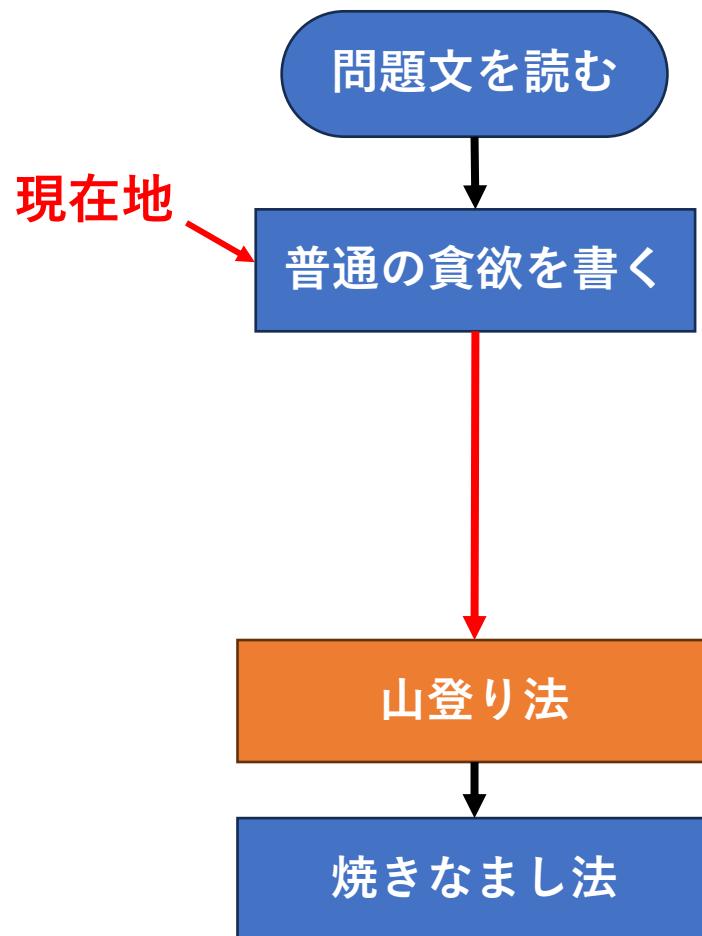
# 解法選択フローチャート

今回は普通の貪欲で出した経路を山登り・焼きなましで改善する



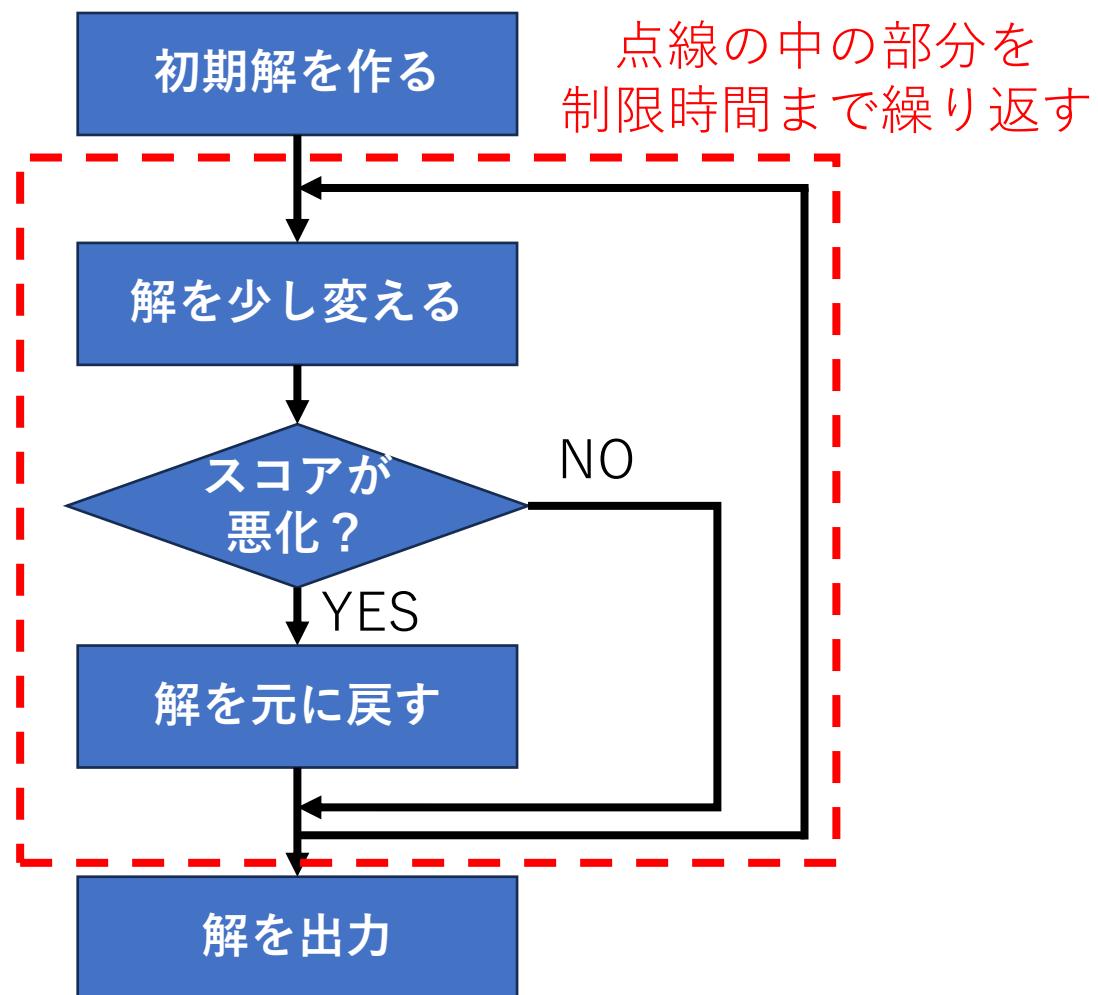
# 解法選択フローチャート

今回は普通の貪欲で出した経路を山登り・焼きなましで改善する



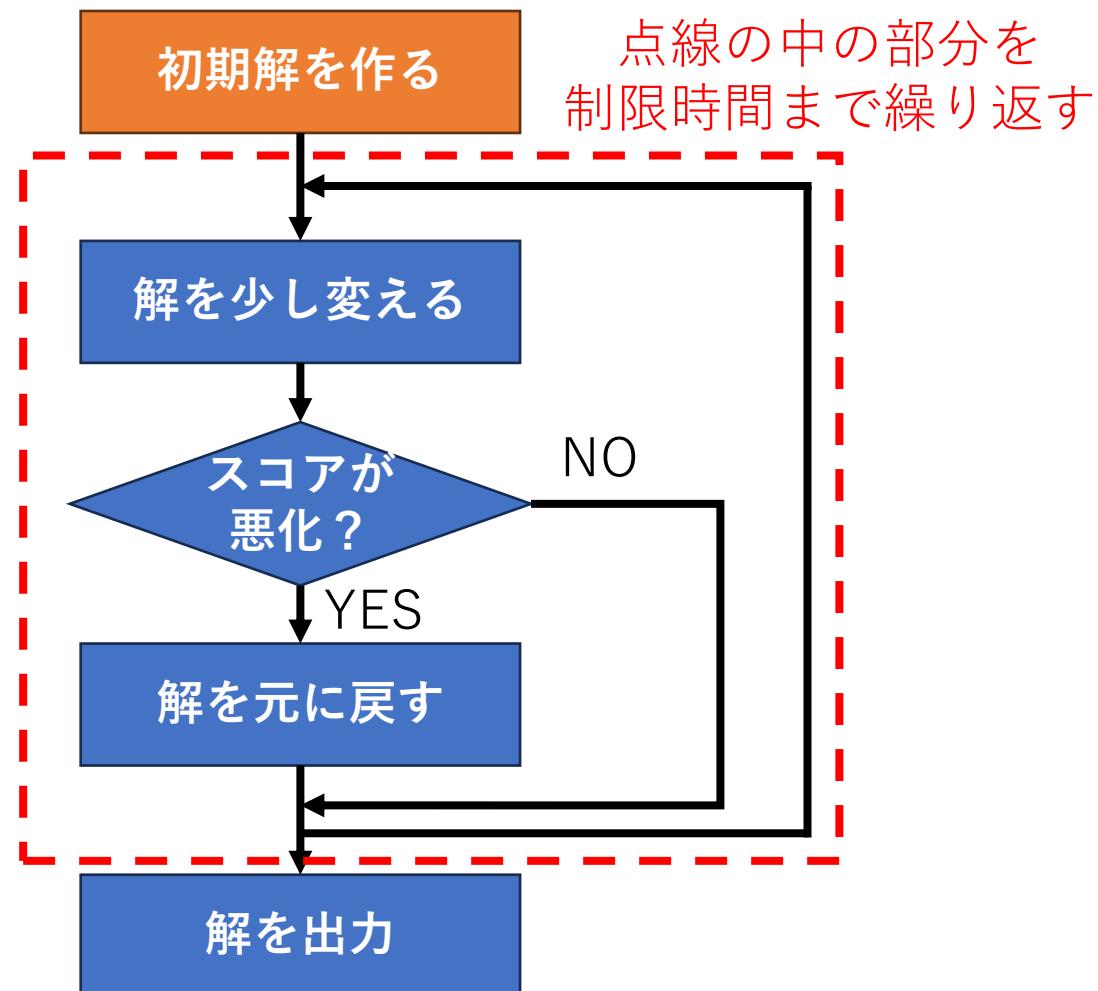
# 山登り法とは？

- 解を少し変えて、スコアが悪化しなかったら変えた後の解を採用することを繰り返す手法



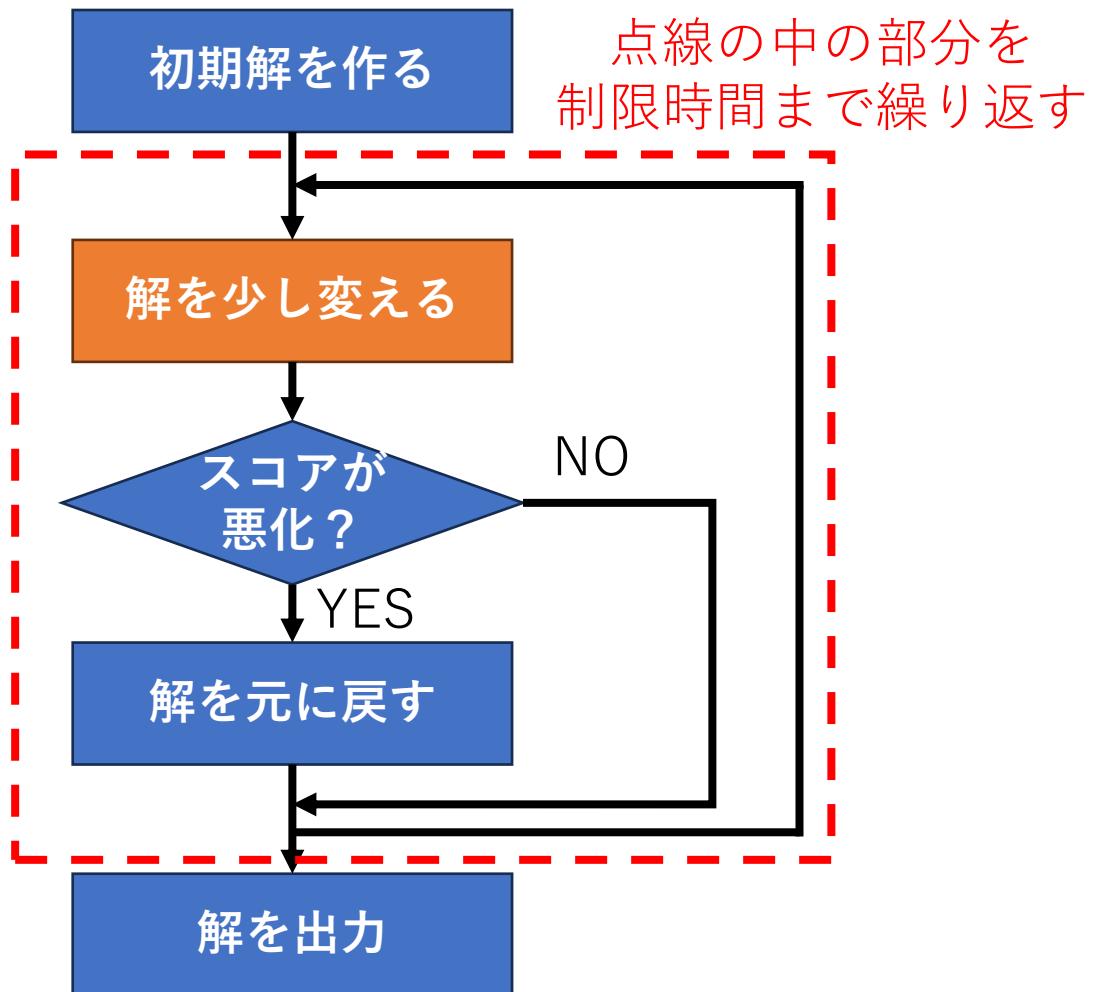
# 山登り法とは？

- まずは初期解を作る
- 今回は既に貪欲を書いたので、貪欲で出した解をそのまま初期解にできる



# 山登り法とは？

- 次に近傍（解の変え方）を用意し、それを使ってランダムに少し変化させる
- 近傍はどうやって考えればよい？

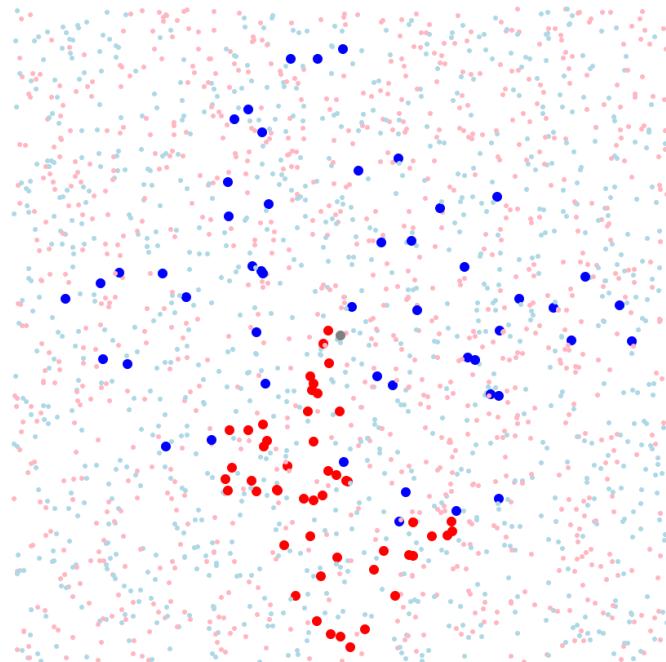


# 近傍をどうやって考えるか

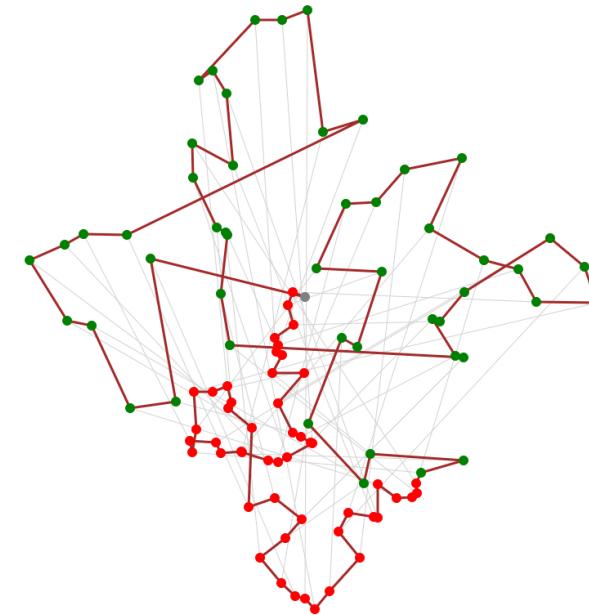
- まずは、解がどんなものだったか整理しましょう

# 近傍をどうやって考えるか

- まずは、解がどんなものだったか整理しましょう



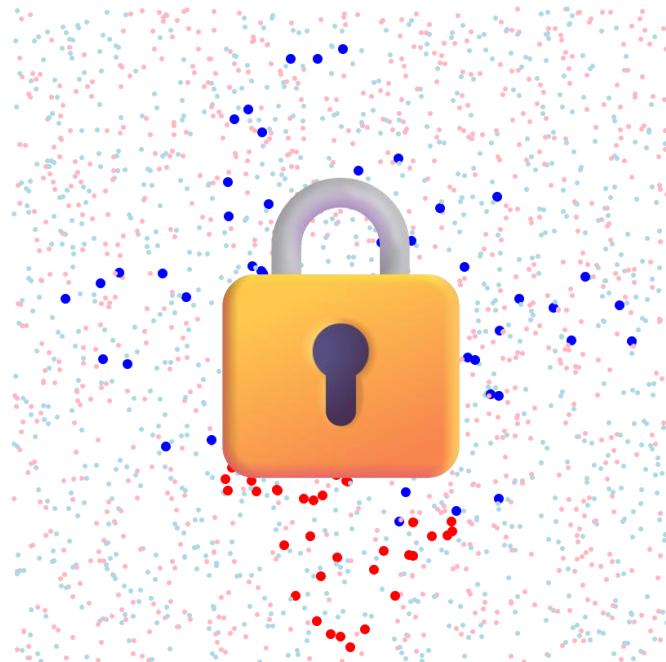
選んだ注文  
(レストラン・配達先の組)



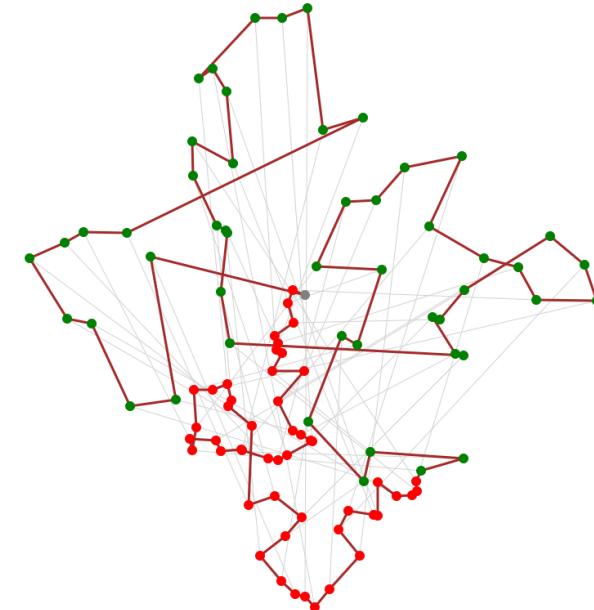
どのように移動するかを  
表すルート

# 近傍をどうやって考えるか

- ・今日は、簡単のために選んだ注文は変えないことにします



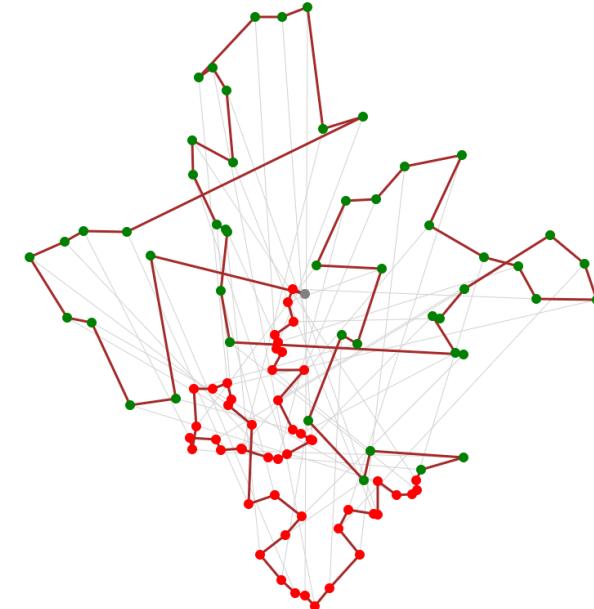
選んだ注文  
(レストラン・配達先の組)



どのように移動するかを  
表すルート

# 近傍をどうやって考えるか

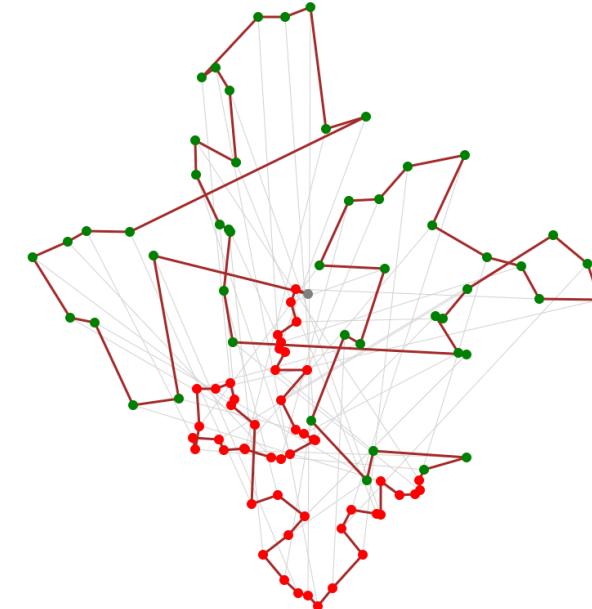
- ルートの中で何なら変えられそうか？



どのように移動するかを  
表すルート

# 近傍をどうやって考えるか

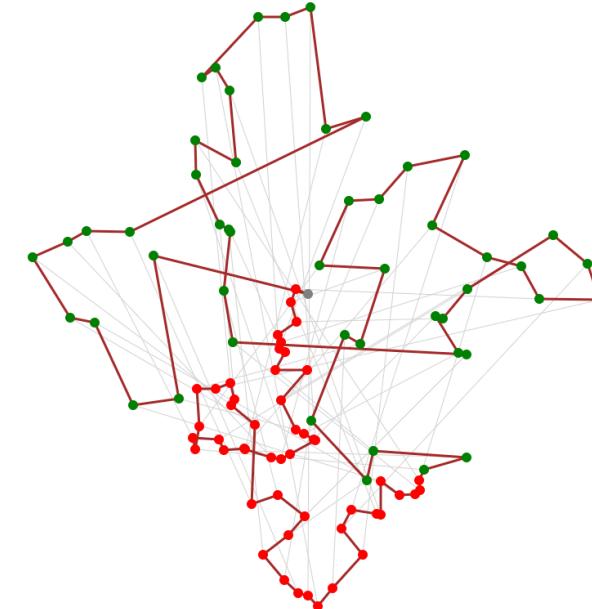
- ルートの中で何なら変えられそうか？
- 例えば、  
レストラン・配達先の位置は  
変えられない
  - 選んだ注文は変えないことにしたので
- レストラン・配達先を訪れる  
順序は変えられる



どのように移動するかを  
表すルート

# 近傍をどうやって考えるか

- ルートの中で何なら変えられそうか？
- 例えば、  
レストラン・配達先の位置は  
変えられない
  - 選んだ注文は変えないことにしたので
- レストラン・配達先を訪れる  
順序は変えられる  
→**今回は順序を変える  
近傍を考えます**

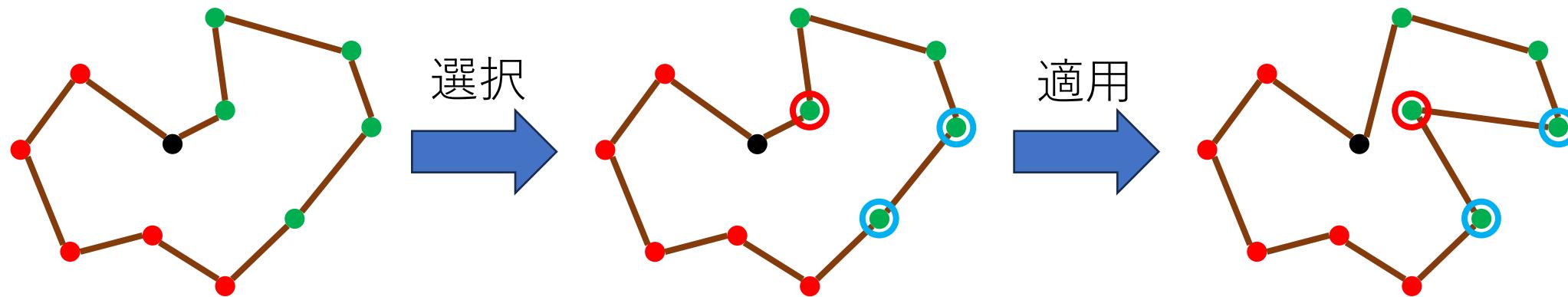


どのように移動するかを  
表すルート

# どんな近傍が考えられそう？

例えば、

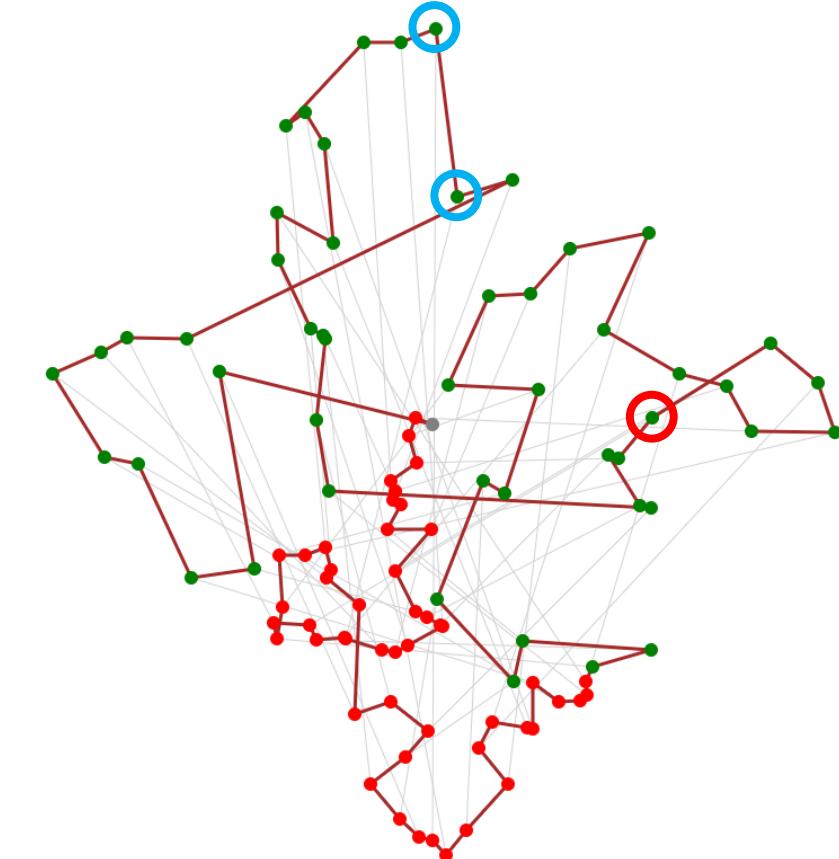
配達先をランダムに1個選んで、  
訪れる順序を他のランダムな配達先の間に変える近傍  
が考えられる



実は、この近傍は順序を決める問題では典型的な近傍のひとつです

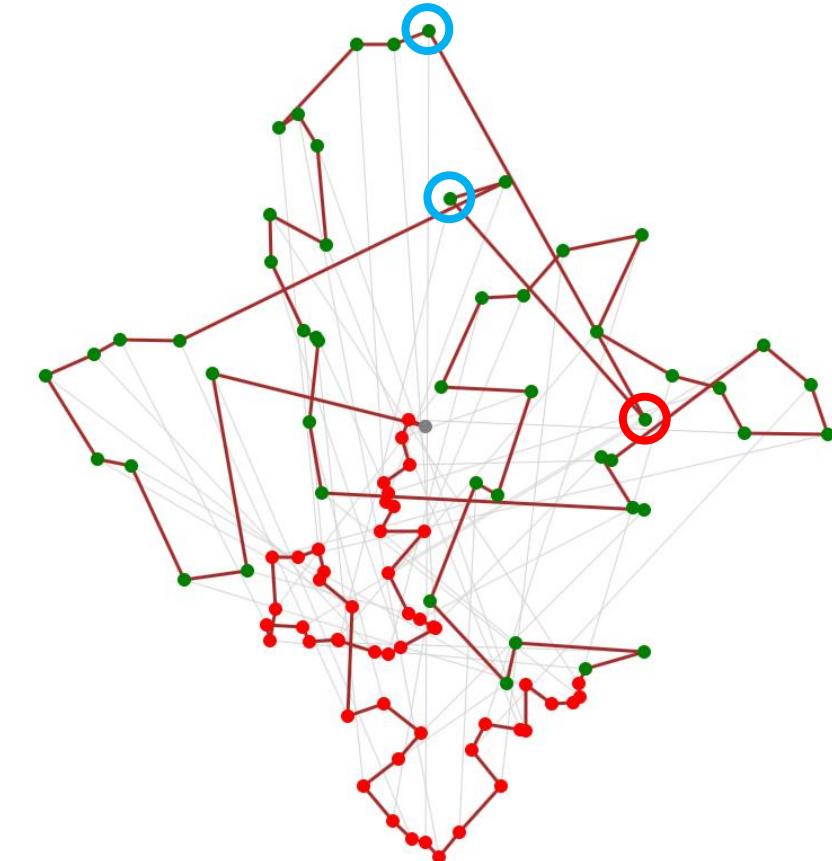
# 実際に近傍を適用してみよう

- 赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみる
- 移動距離 = 6308



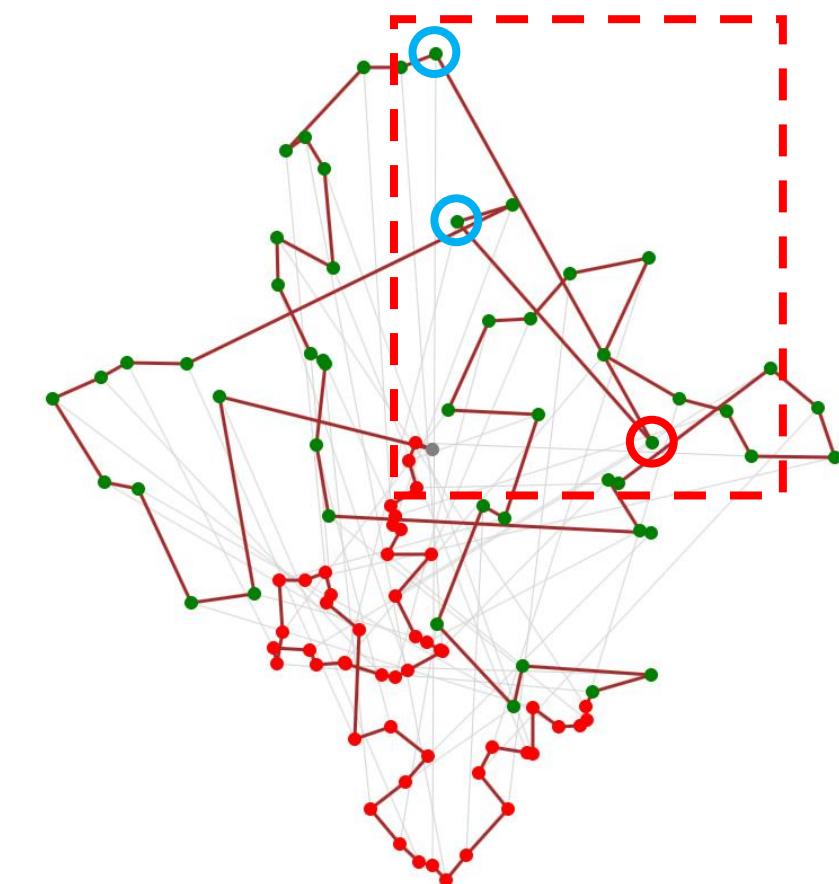
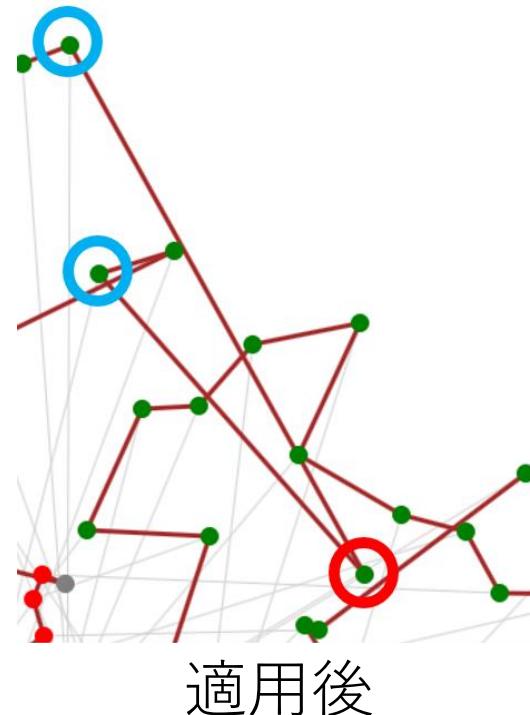
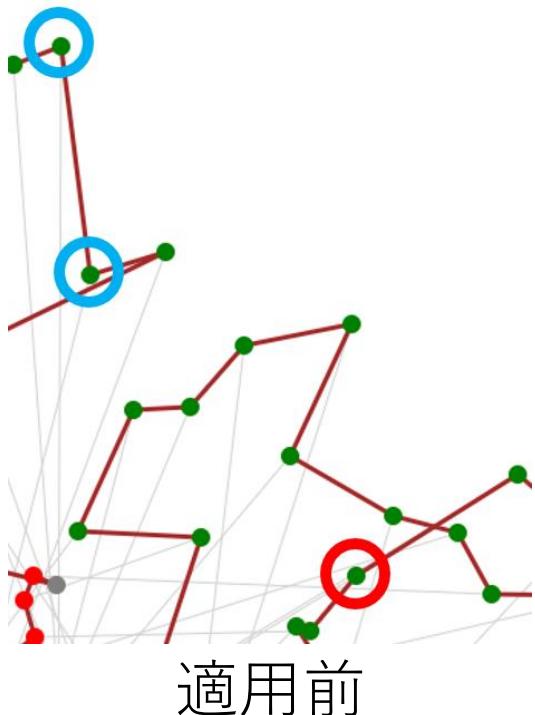
# 実際に近傍を適用してみよう

- 赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみる
- 移動距離 = 7046 (+738)



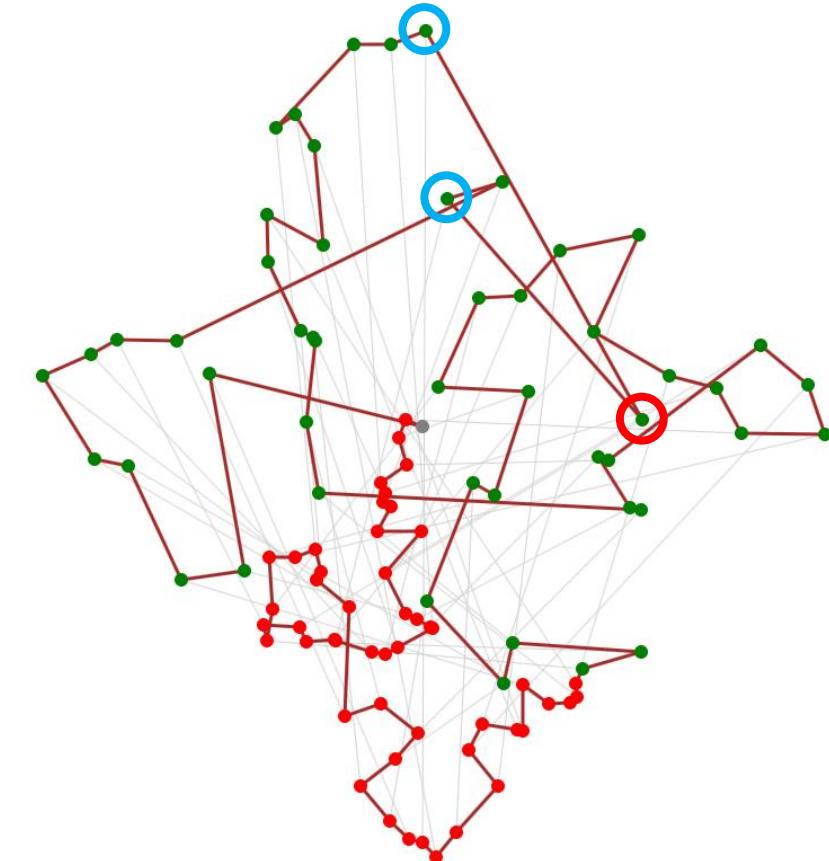
# 実際に近傍を適用してみよう

- 赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみる



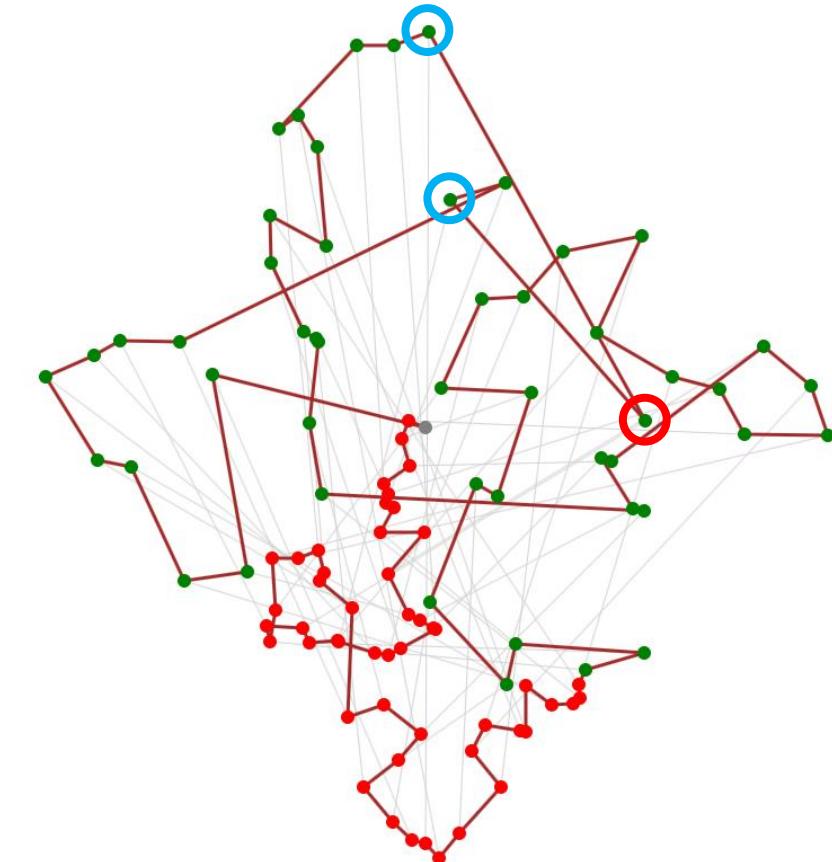
# 実際に近傍を適用してみよう

- 赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみる
- 移動距離 = 7046 (+738)



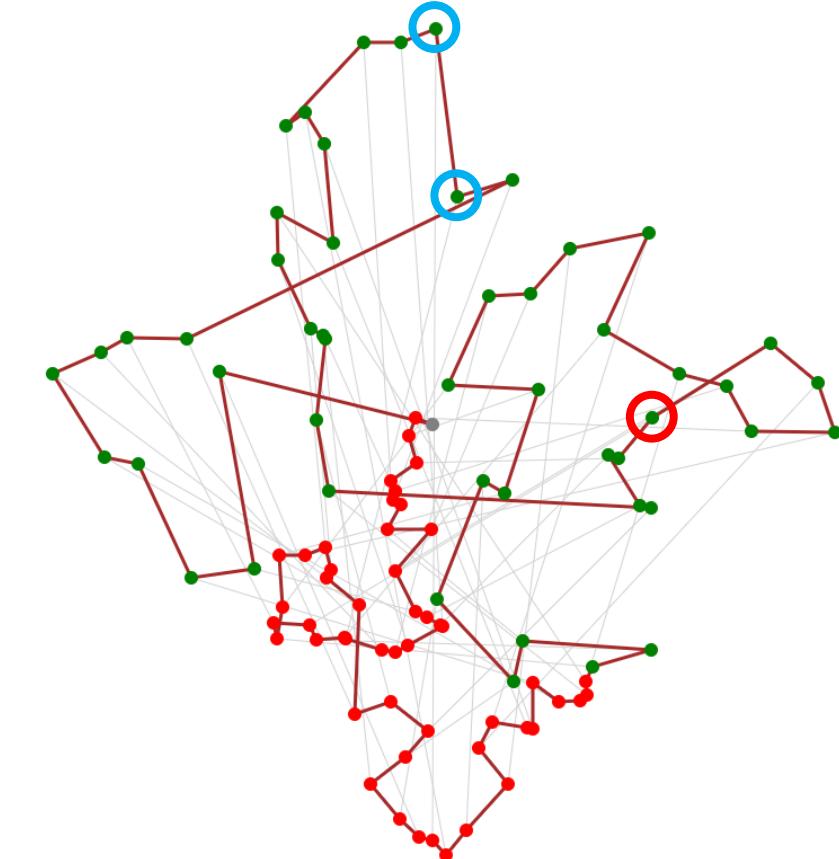
# 実際に近傍を適用してみよう

- 赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみる
- 移動距離 = 7046 (+738)  
→悪化したので元に戻す



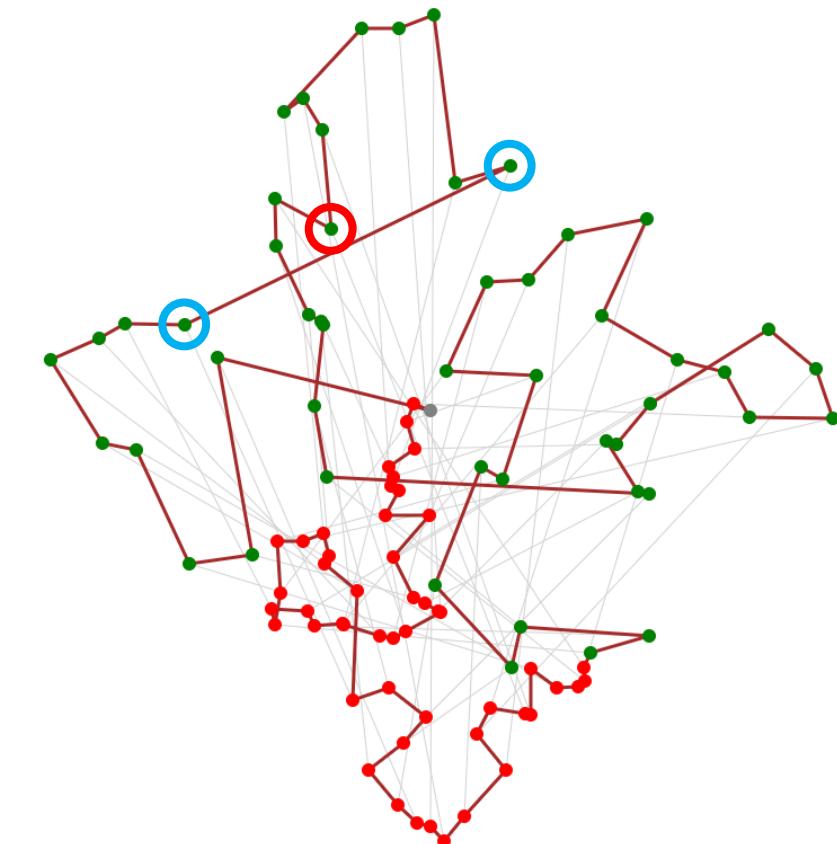
# 実際に近傍を適用してみよう

- 赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみる
- 移動距離 = 6308  
→悪化したので元に戻す



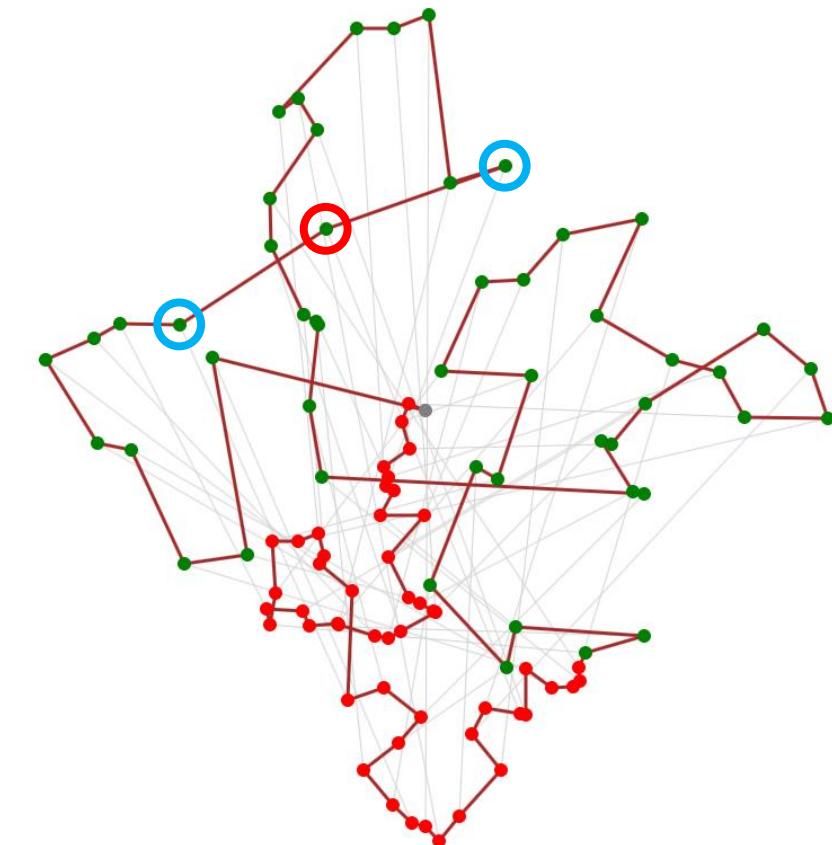
# 実際に近傍を適用してみよう

- 赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみる
- 移動距離 = 6308



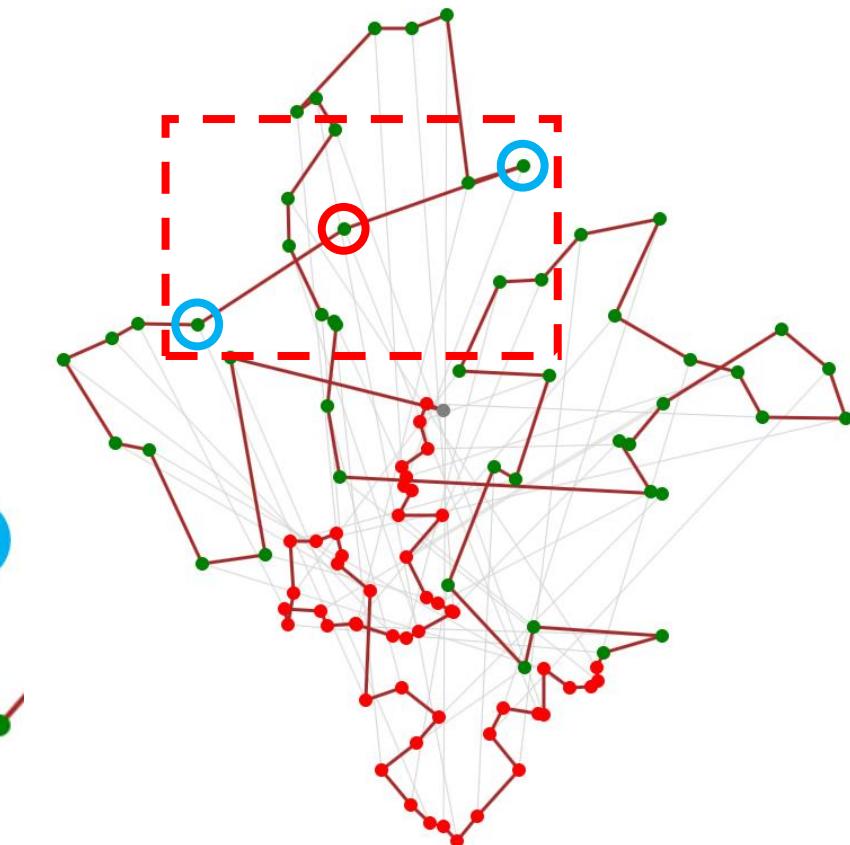
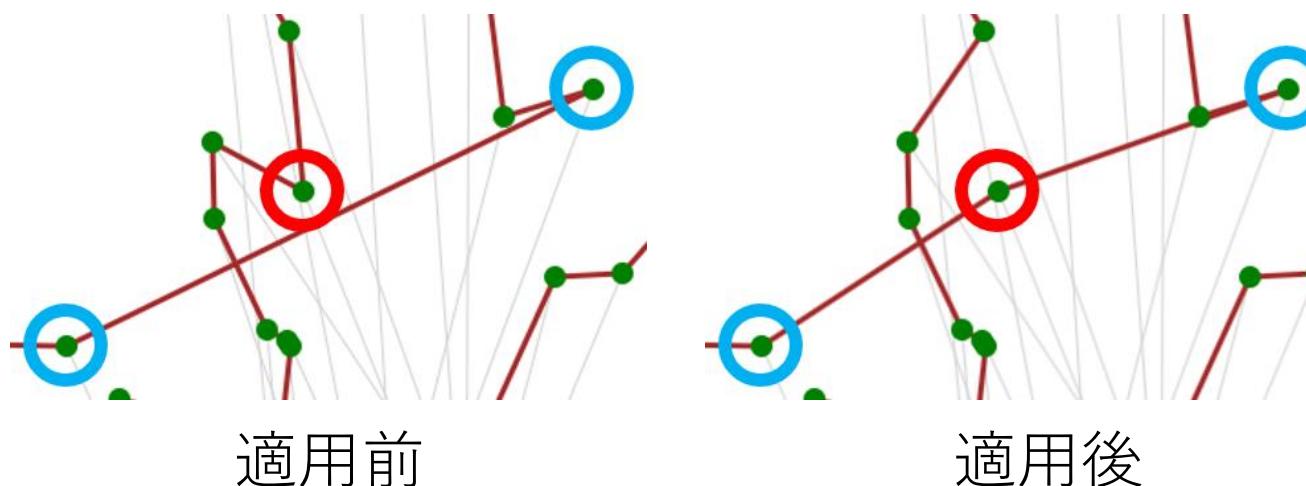
# 実際に近傍を適用してみよう

- 赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみる
- 移動距離 = 6238 (-70)



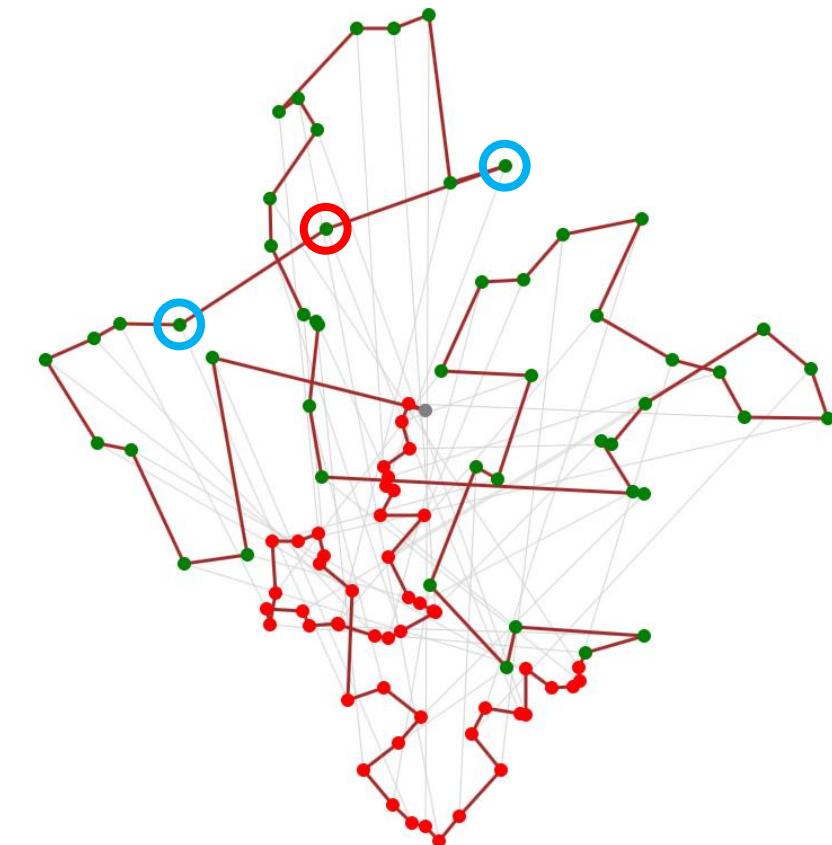
# 実際に近傍を適用してみよう

- 赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみる
- 移動距離 = 6238 (-70)



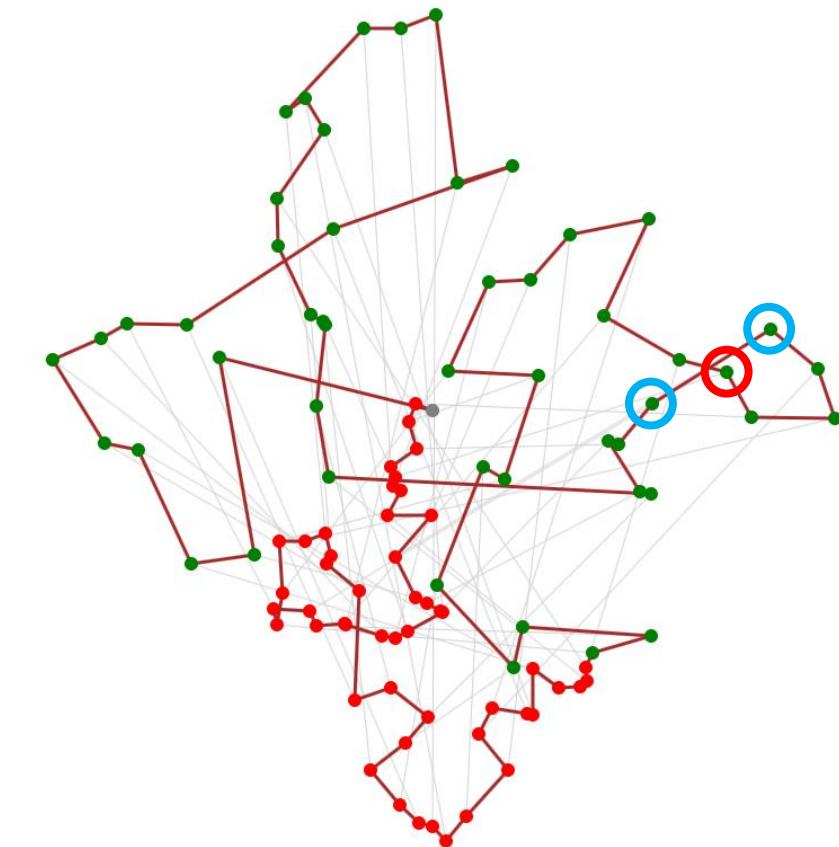
# 実際に近傍を適用してみよう

- 赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみる
- 移動距離 = 6238 (-70)  
→悪化していないので採用



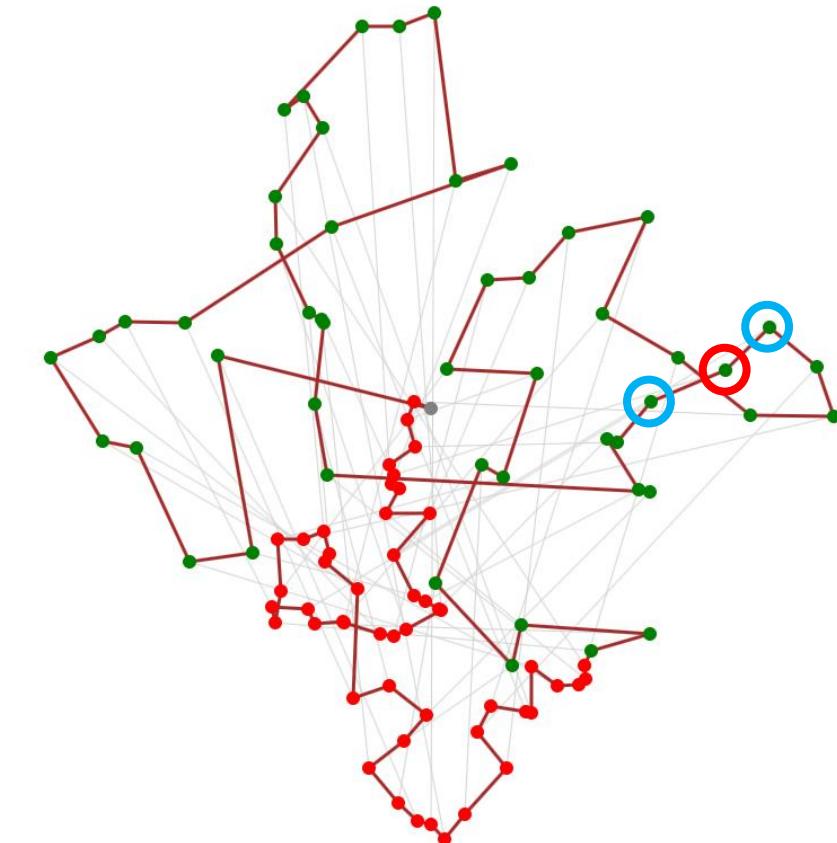
# 実際に近傍を適用してみよう

- 赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみる
- 移動距離 = 6238



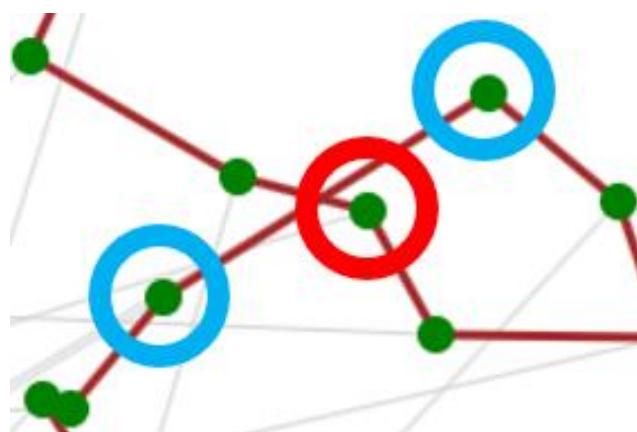
# 実際に近傍を適用してみよう

- 赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみる
- 移動距離 = 6238 (+0)

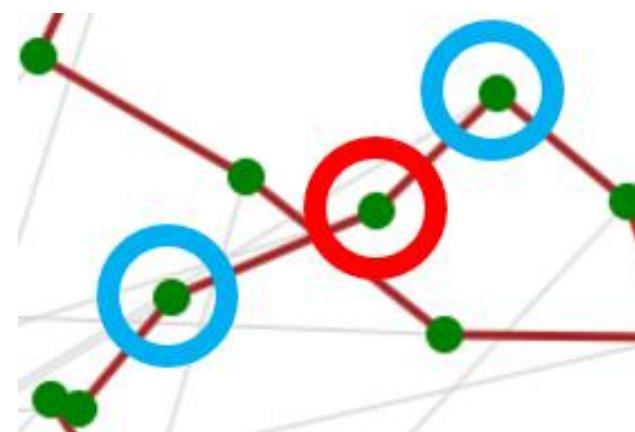


# 実際に近傍を適用してみよう

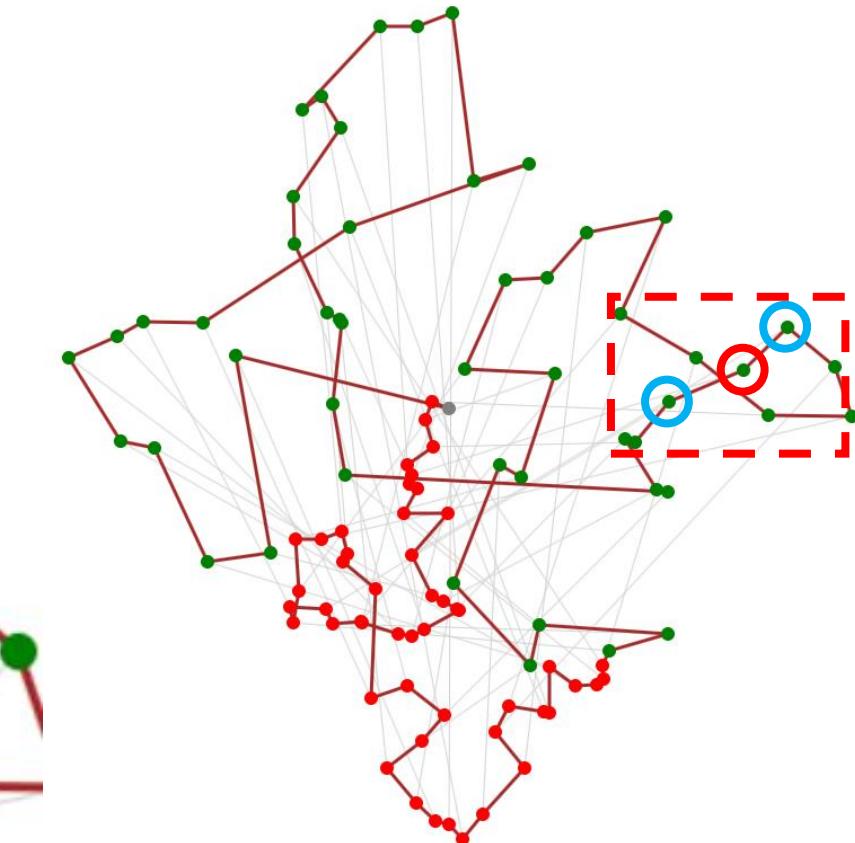
- 赤丸で示した配達先を訪れる順序を青丸で示した配達先の間に変更してみる
- 移動距離 = 6238 (+0)



適用前

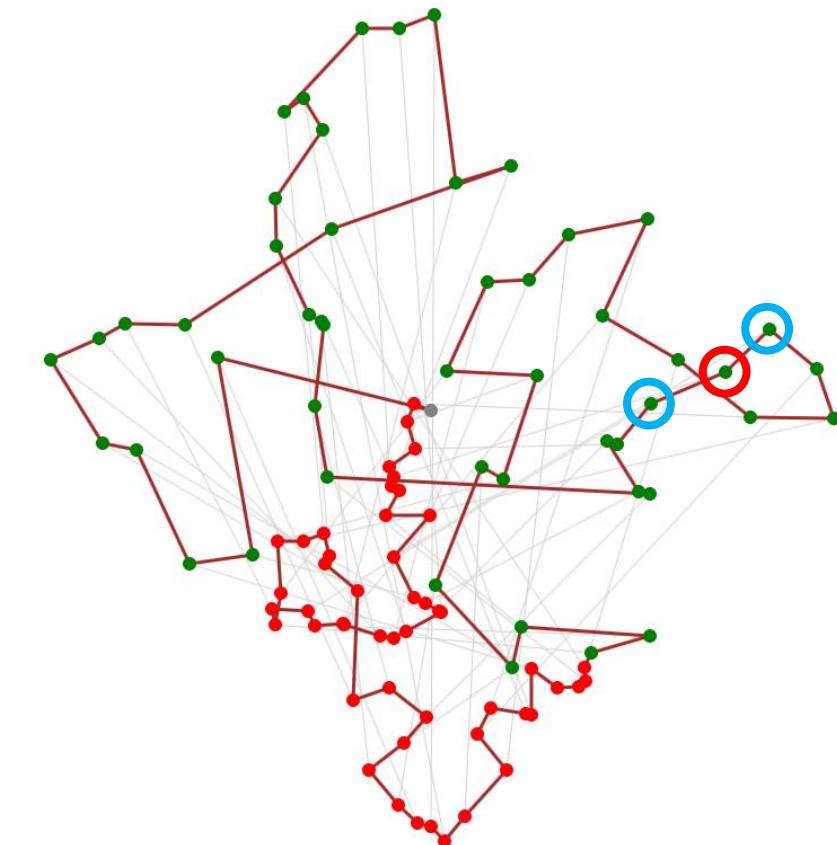


適用後



# 実際に近傍を適用してみよう

- 赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみる
- 移動距離 = 6238 (+0)  
→悪化していないので採用



# 実装タイム

ご自身の言語の「山登り」のコードに実装しましょう！

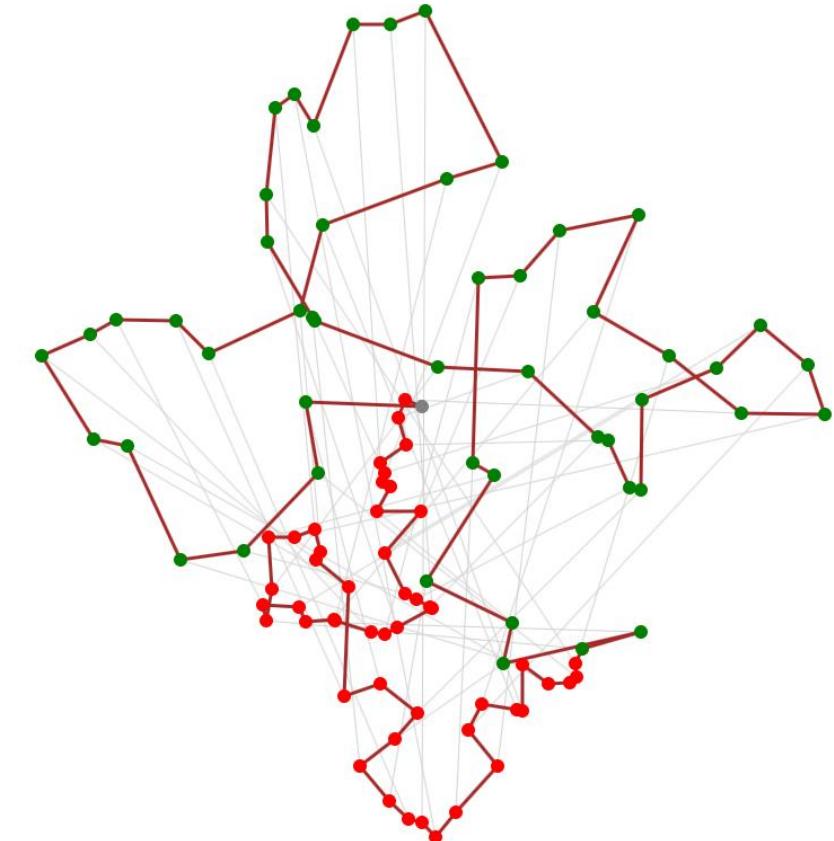
困ったときには問題ページの質問タブから受け付けますので必要事項をご記入ください。  
スタッフが順番に伺います。

- ①テーブル名 ②言語 ③質問内容



# 山登り法の実行結果

- ここまでの中を適切に実装すると、移動距離が5850～6050程度になります※

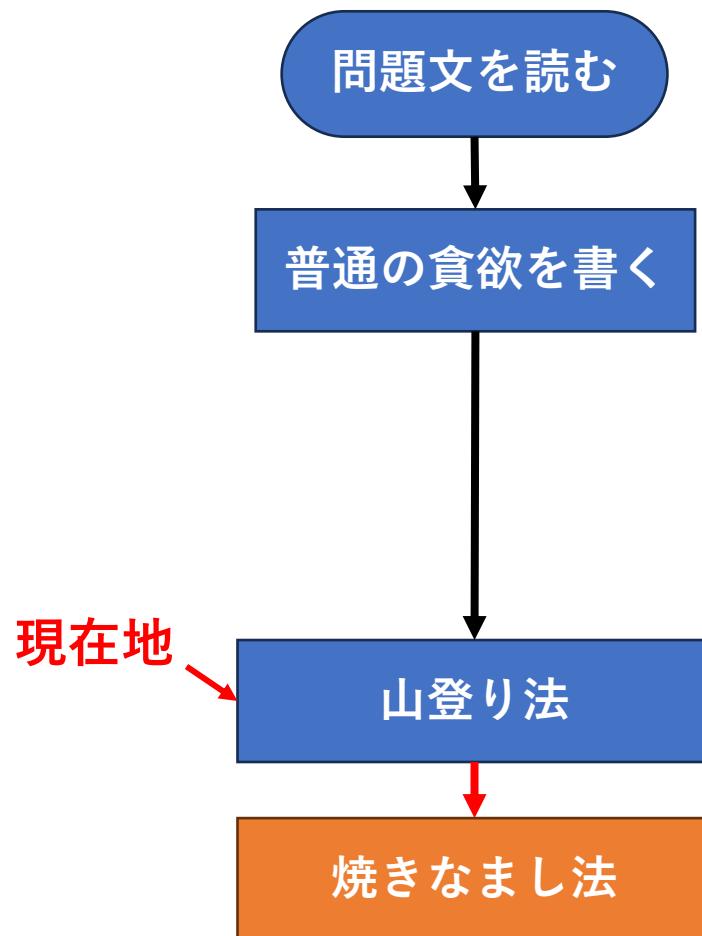


※乱数のシード値次第で結果がある程度変わります

解説者が本解法で確認した最良解  
(移動距離 = 5862)

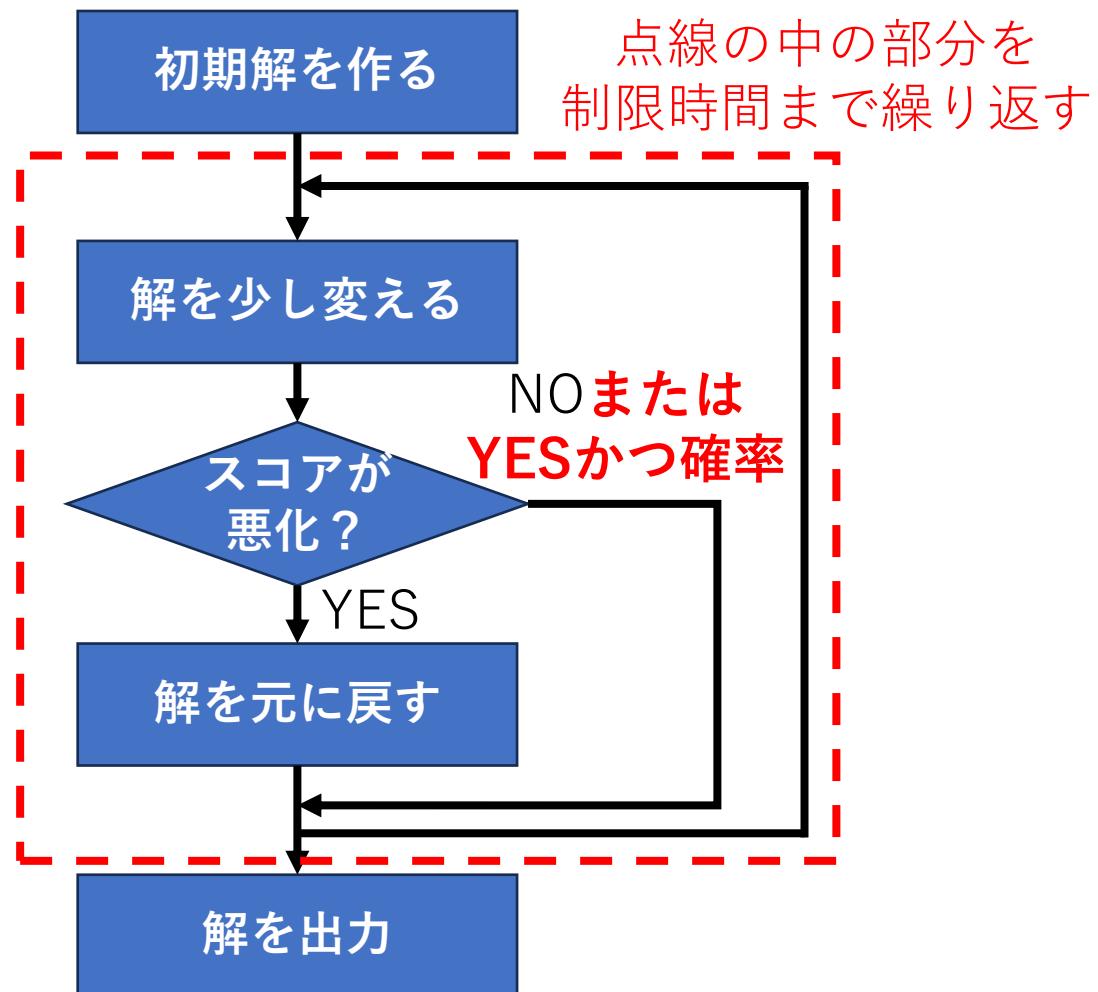
# 解法選択フローチャート

今回は普通の貪欲で出した経路を山登り・焼きなましで改善する

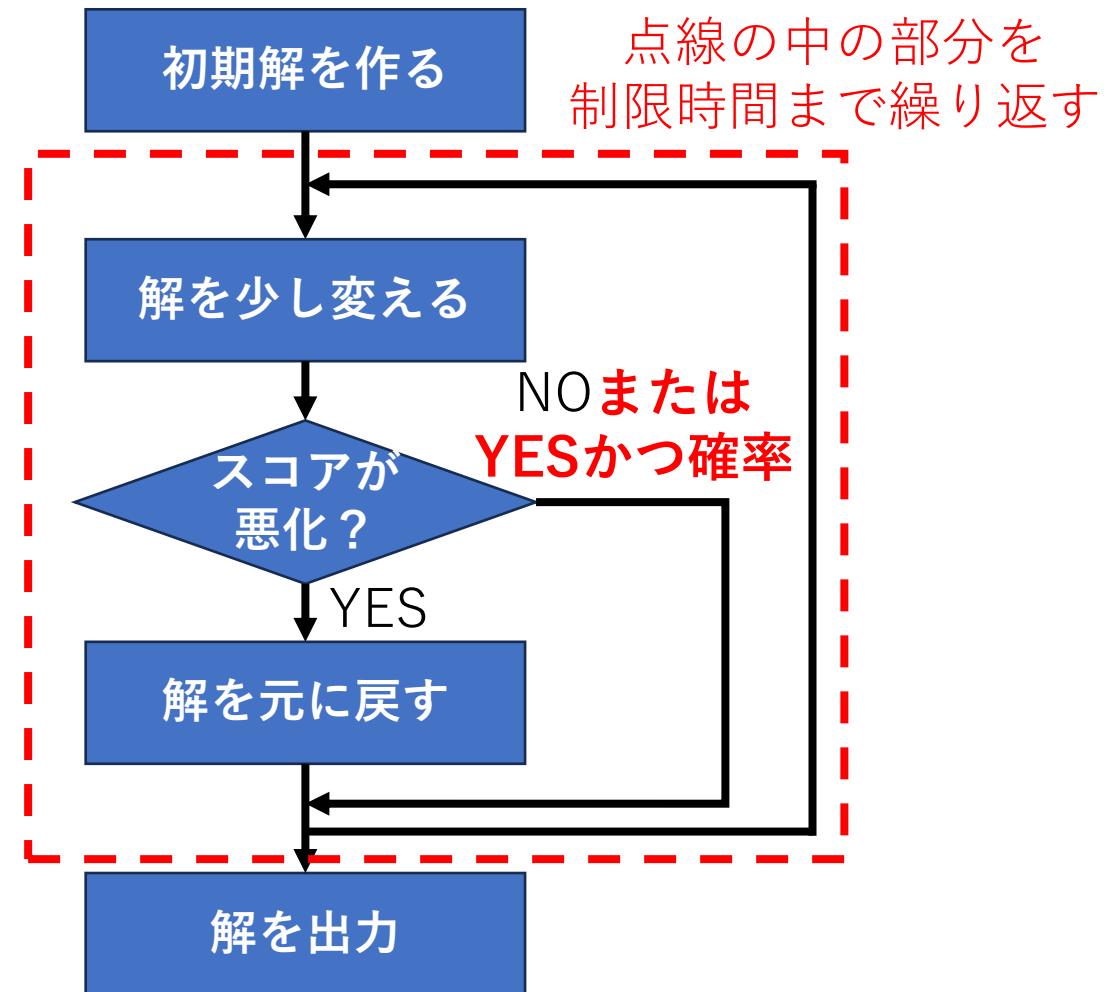
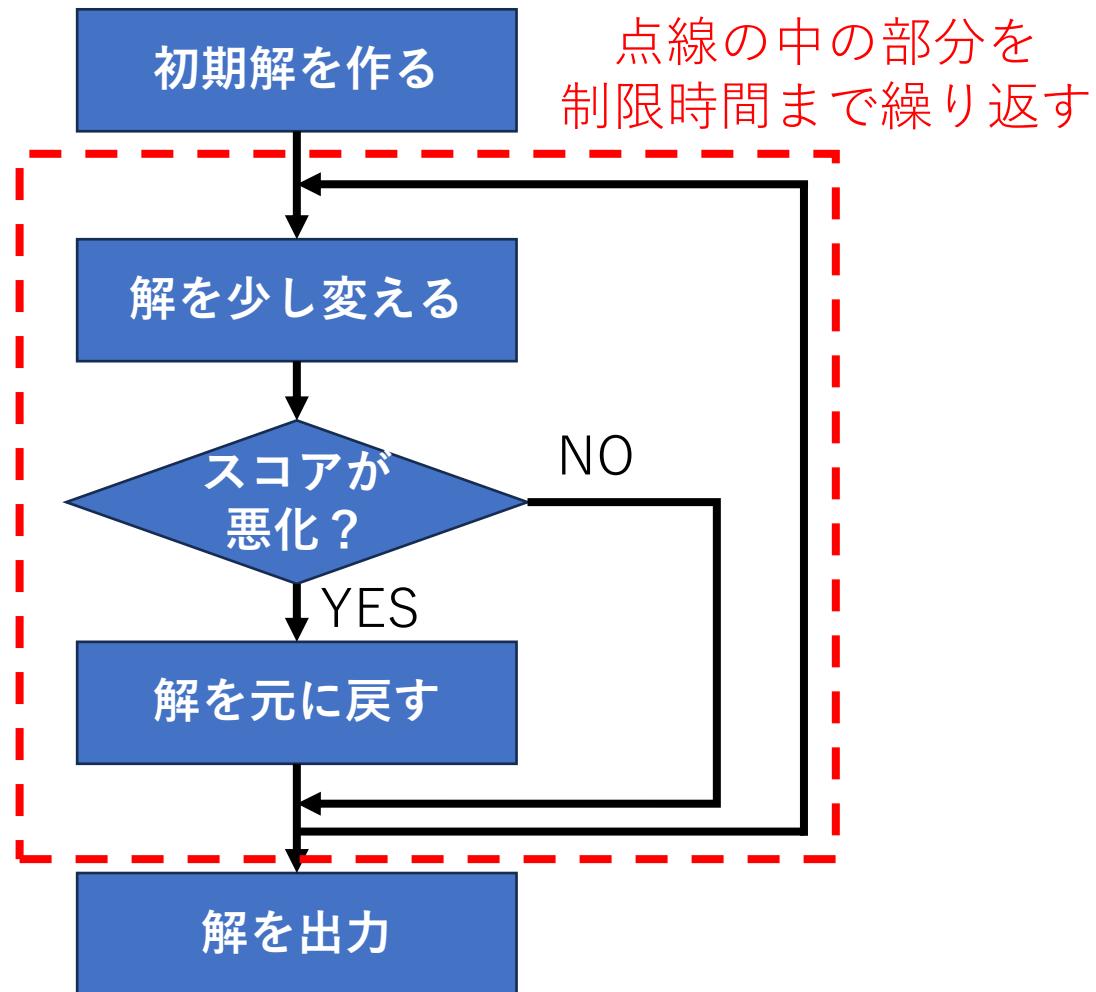


# 焼きなまし法とは？

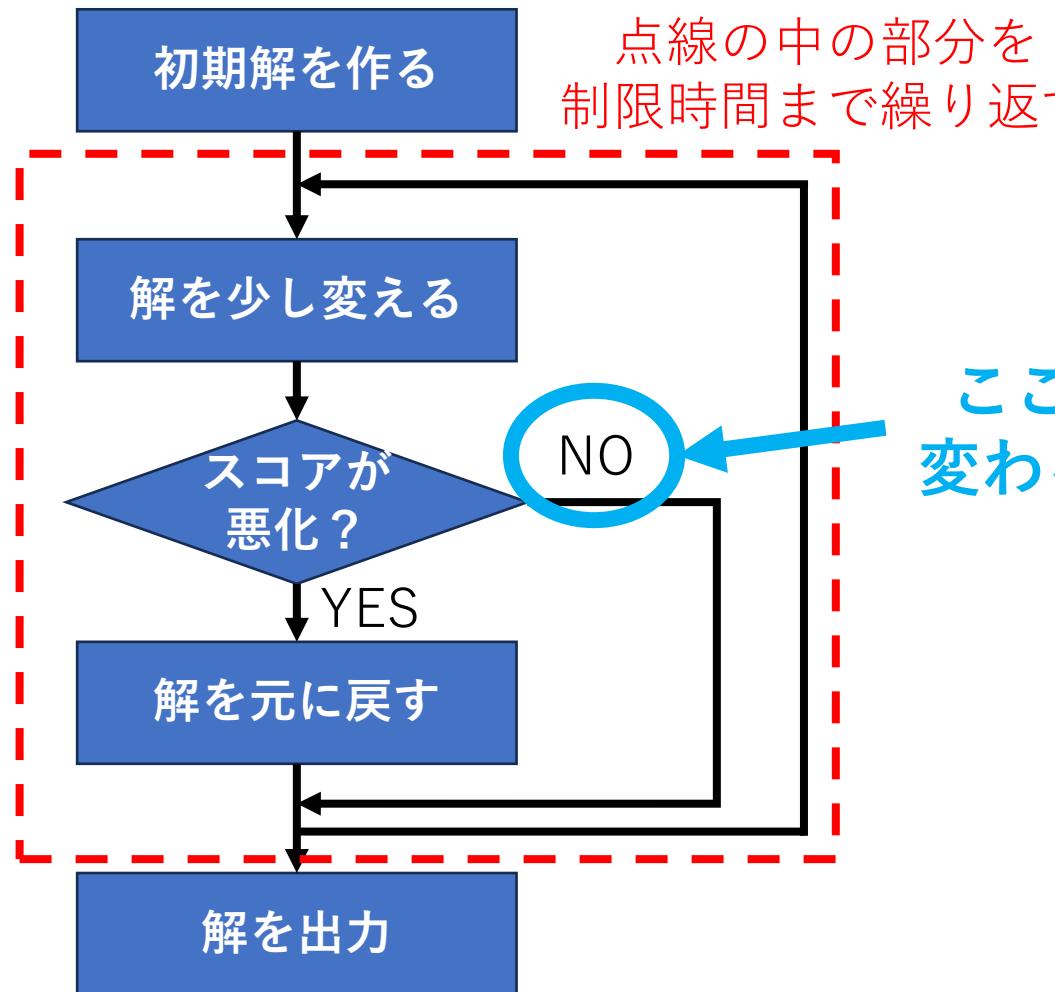
- 解を少し変えて、スコアが悪化しなかったら変えた後の解を採用することを繰り返す手法
- さらに、スコアが悪化していても確率的に変えた後の解を採用する



# 山登り法と焼きなまし法の比較

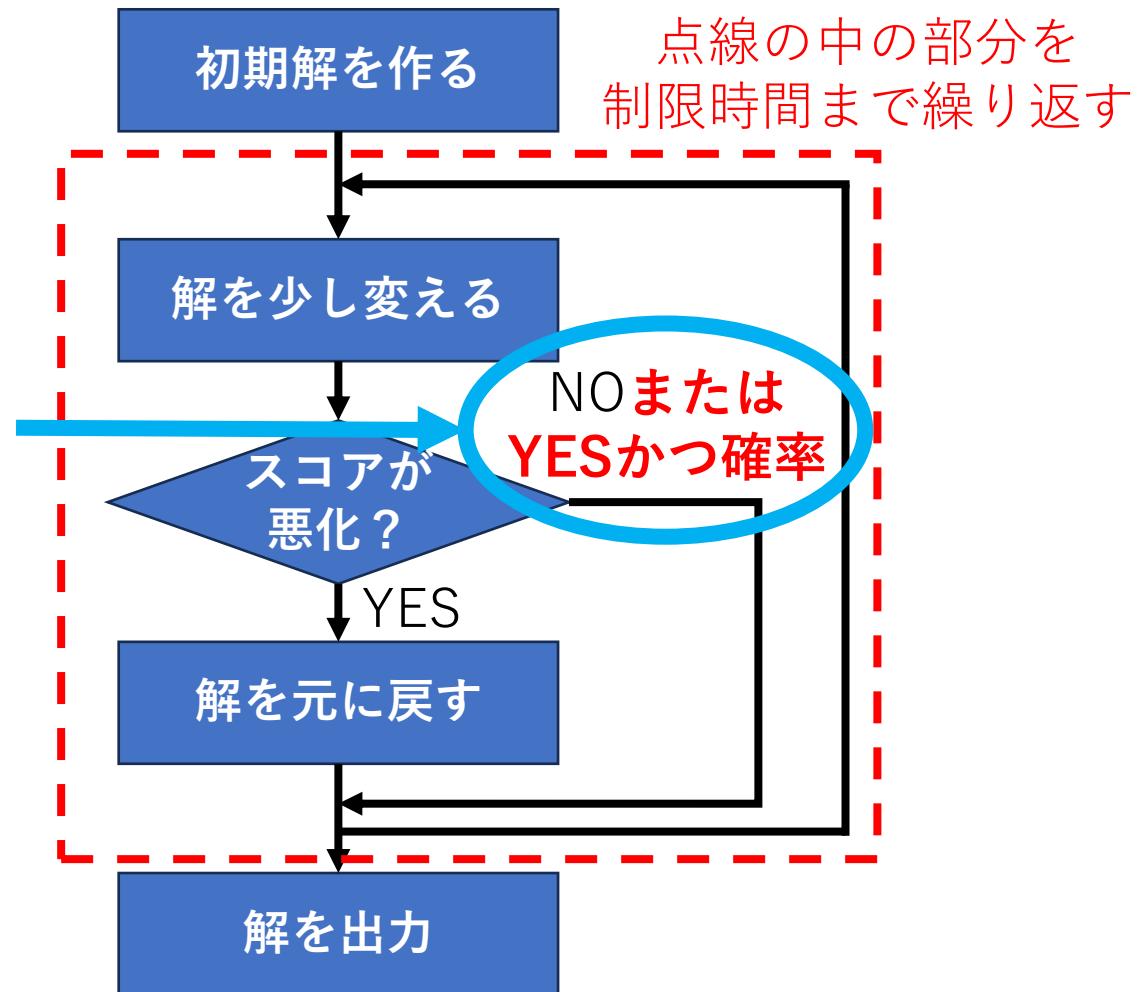


# 山登り法と焼きなまし法の比較



ここが  
変わる！

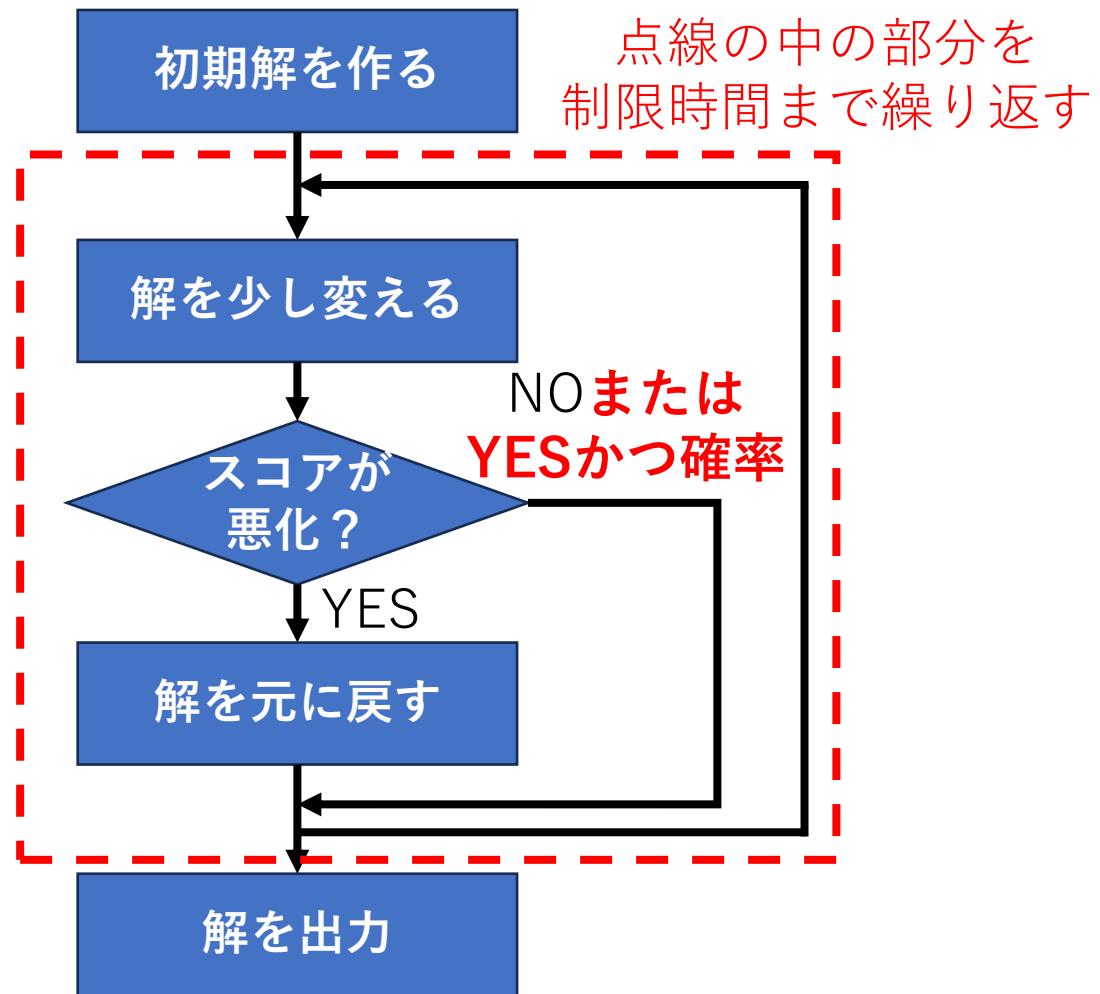
山登り法



焼きなまし法

# 悪化した解を確率的に採用する

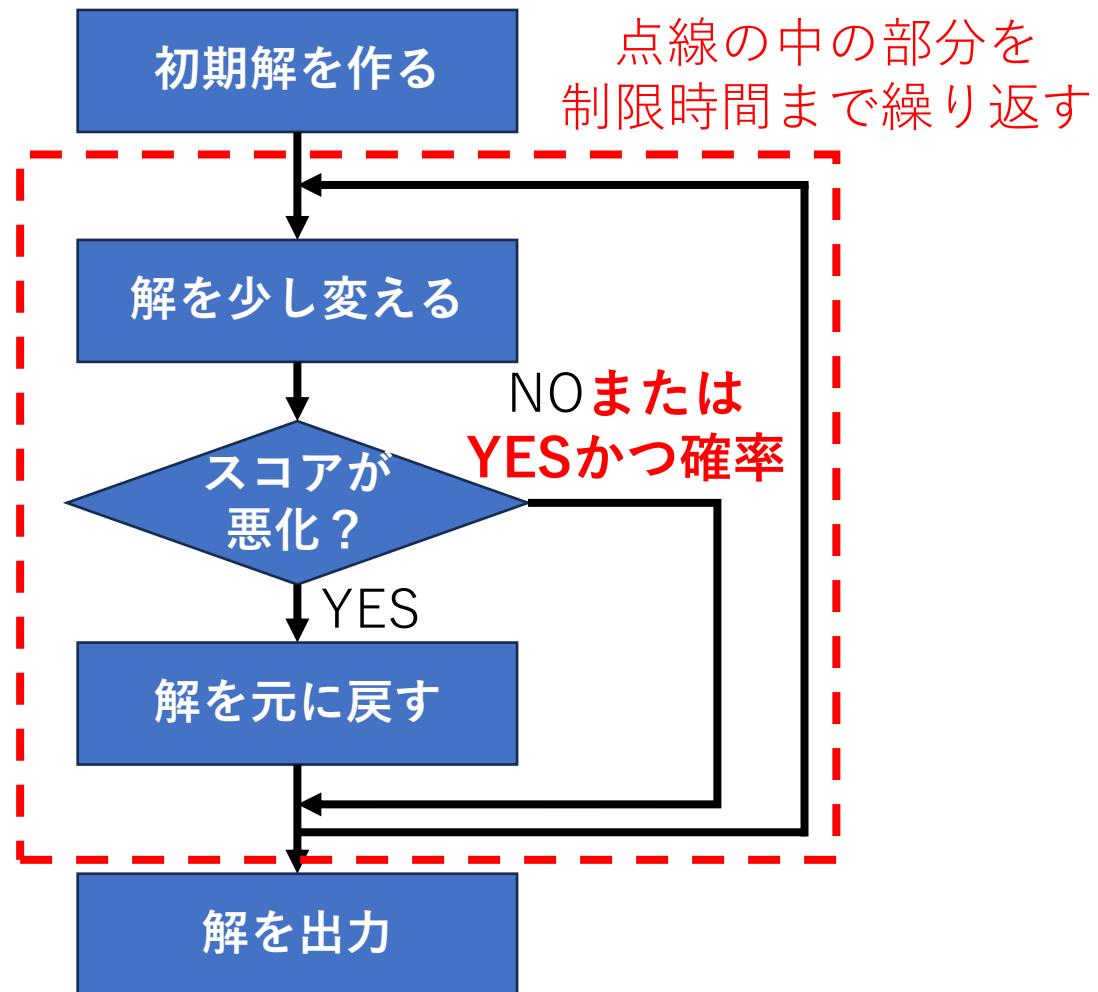
確率的に変えた後の解を採用する  
ってどういうこと？？



# 悪化した解を確率的に採用する

確率的に変えた後の解を採用する  
ってどういうこと？？

これから色々な数式が出てきますが・・・  
実際にAHCで焼きなまし法を使うときは  
意識しなくても大丈夫です

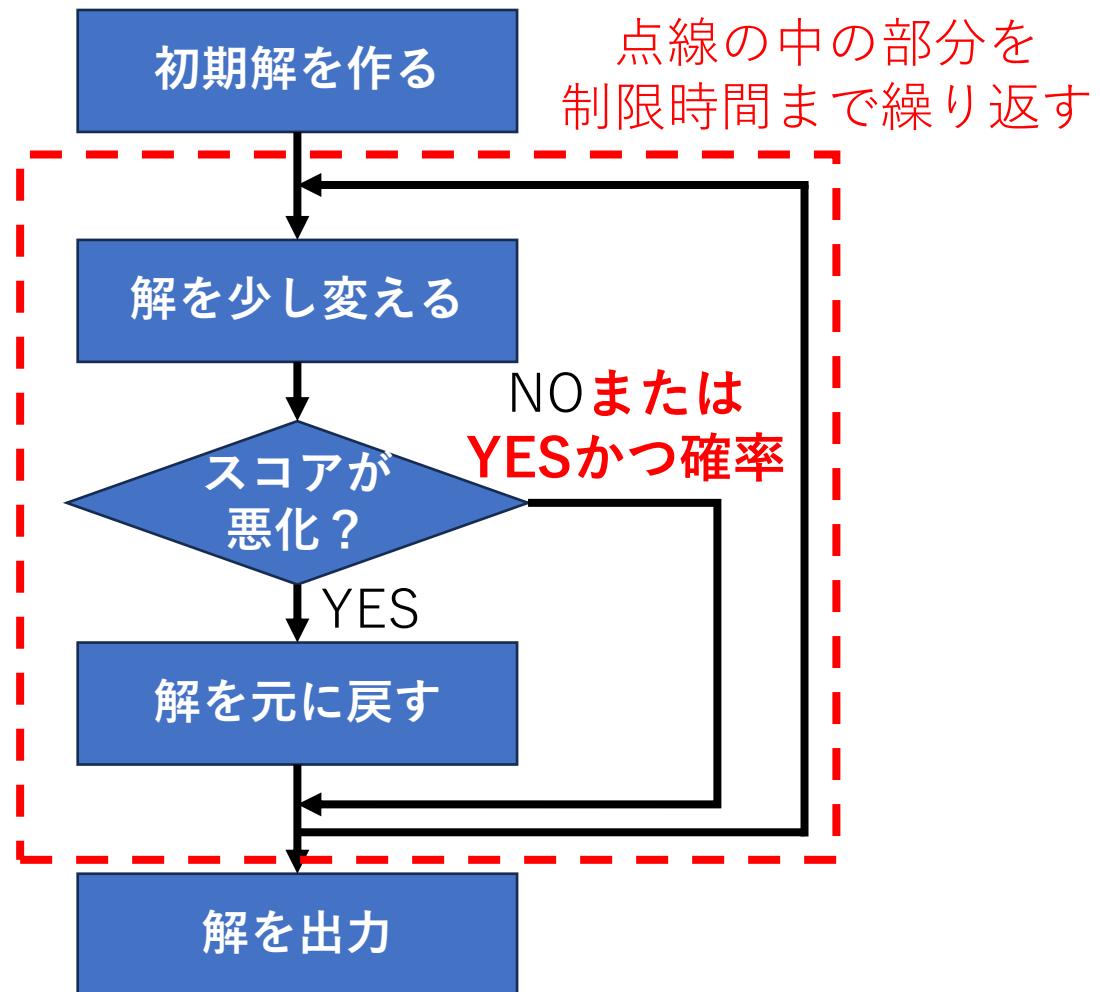


# 悪化した解を確率的に採用する

確率的に変えた後の解を採用する  
ってどういうこと？？

- 距離の悪化量を $\Delta S$
- 温度を $T$   
として、  
 $e^{\frac{-|\Delta S|}{T}}$  の確率で採用する

実際に採用するかは乱数で決める



# 悪化した解を確率的に採用する

確率的に変えた後の解を採用する  
ってどういうこと？？

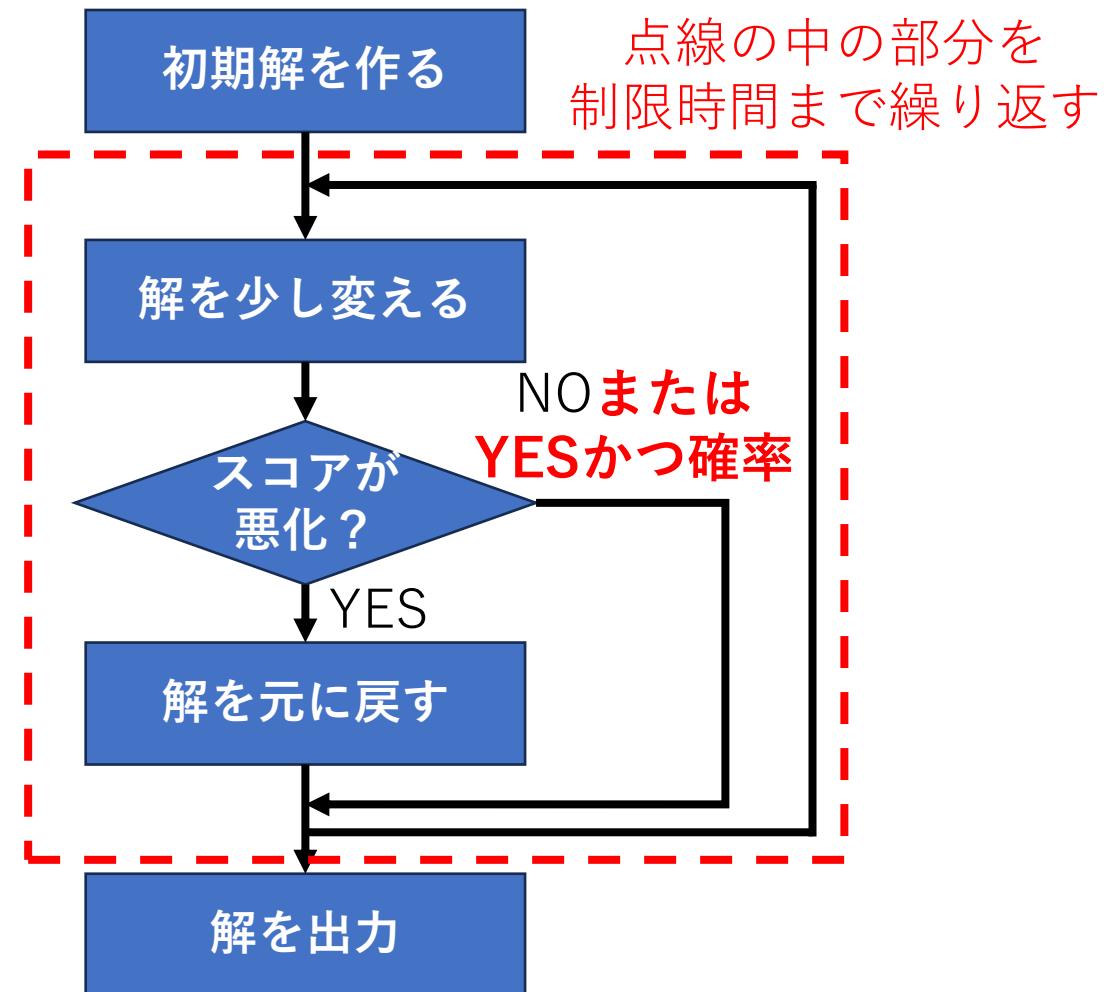
- 距離の悪化量を $\Delta S$
- 温度を $T$

として、  
 $e^{\frac{-|\Delta S|}{T}}$  の確率で採用する

悪化量が小さいほど  
採用されやすい

温度が高いほど  
採用されやすい

実際に採用するかは乱数で決める



# 悪化した解を確率的に採用する

確率的に変えた後の解を採用する

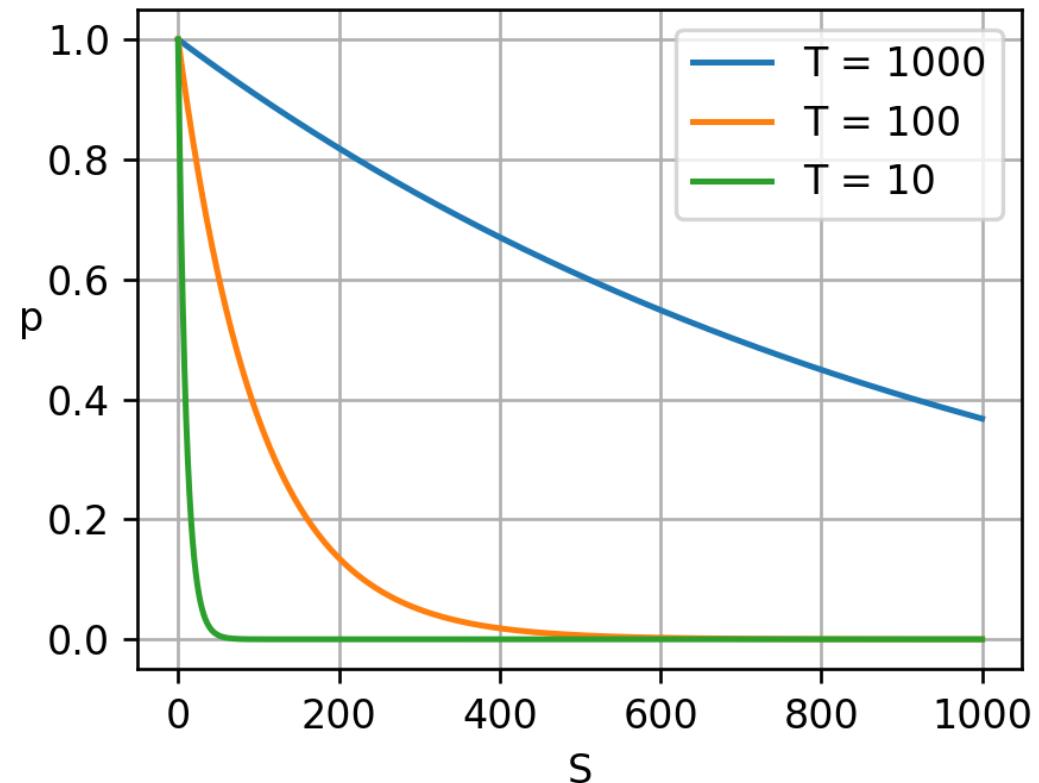
ってどういうこと？？

- 距離の悪化量を $\Delta S$
- 温度を $T$

として、悪化量が小さいほど  
採用されやすい

$e^{\frac{-|\Delta S|}{T}}$  の確率で採用する  
温度が高いほど  
採用されやすい

実際に採用するかは乱数で決める



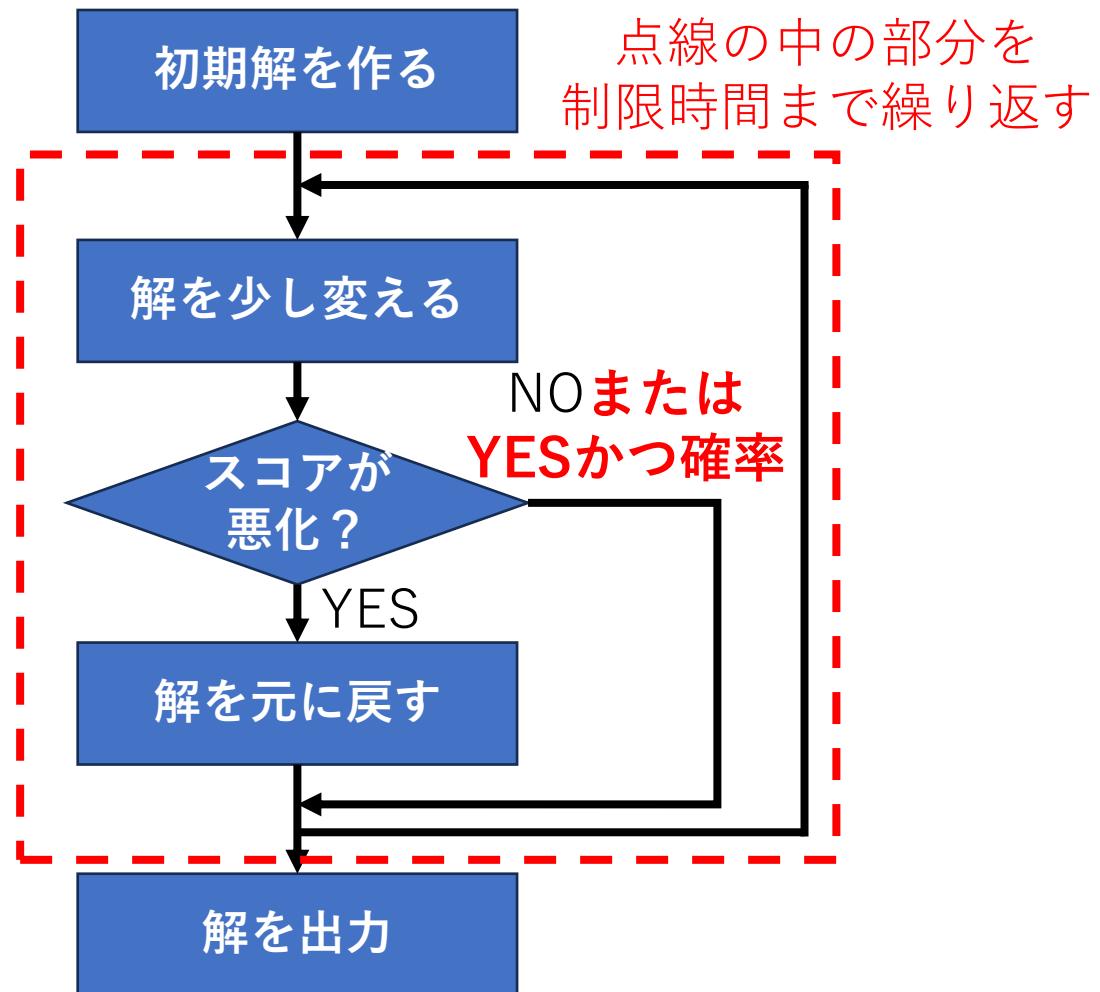
$T, \Delta S$ による採用確率の変化

# 悪化した解を確率的に採用する

確率的に変えた後の解を採用する  
ってどういうこと？？

- 距離の悪化量を $\Delta S$
- **温度**を $T$   
として、  
 $e^{\frac{-|\Delta S|}{T}}$  の確率で採用する

実際に採用するかは乱数で決める



# 悪化した解を確率的に採用する

## 温度って何？？

悪化したときの採用されやすさを決めるパラメータ

- 開始温度を  $T_{start}$
- 終了温度を  $T_{end}$
- 開始から終了までの経過時間の割合を  $t$  として、 $t$ だけ経過した時点での温度  $T$  は  $T = T_{start}^{1-t} \cdot T_{end}^t$  とする

- 距離の悪化量を  $\Delta S$
- 温度を  $T$  として、  
 $e^{-\frac{|\Delta S|}{T}}$  の確率で採用する

温度が高いほど  
採用されやすい

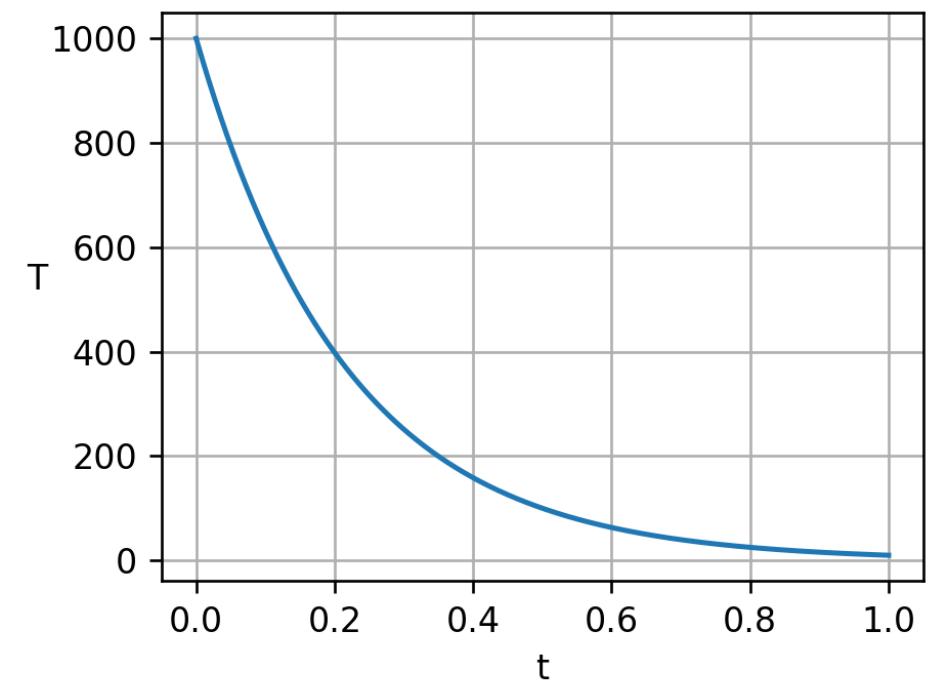
# 悪化した解を確率的に採用する

## 温度って何？？

悪化したときの採用されやすさを決めるパラメータ

- 開始温度を  $T_{start}$
- 終了温度を  $T_{end}$
- 開始から終了までの経過時間の割合を  $t$  として、  $t$ だけ経過した時点での温度  $T$  は

$$T = T_{start}^{1-t} \cdot T_{end}^t \text{ とする}$$



$T_{start} = 1000, T_{end} = 10$  の場合の  
温度変化

悪化した解を確率的に採用する

ここまで色々な数式が出てきましたが・・・

実際にAHCで焼きなまし法を使うときは  
意識しなくとも大丈夫です

開始温度と終了温度を変えると、  
実行途中のスコアの変化の仕方が変わる  
とだけ覚えておきましょう

# 実装タイム

ご自身の言語の「焼きなまし」のコードに実装しましょう！

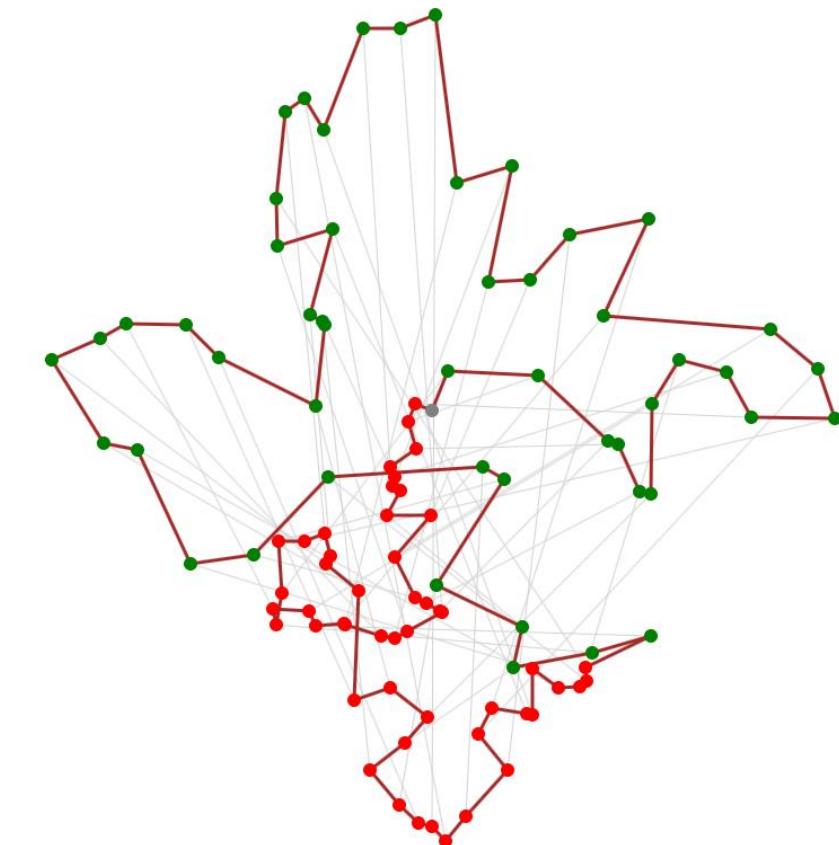
困ったときには問題ページの質問タブから受け付けますので必要事項をご記入ください。  
スタッフが順番に伺います。

- ①テーブル名 ②言語 ③質問内容

The screenshot shows a user interface for asking questions. At the top, there are navigation links: トップ (Top), 問題 (Problems), 質問 (Question) (highlighted with a red border), 提出 (Submission), and 提出結果 (Submission Result). Below these, the word "質問" (Question) is displayed in large black text. A red arrow points upwards from the bottom of the "質問" text towards the "質問" link in the header. Further down, a message says "質問がまだありません。" (No questions yet.) and a blue button labeled "質問する" (Ask a question) is shown, with a red arrow pointing to its left side.

# 焼きなまし法の実行結果

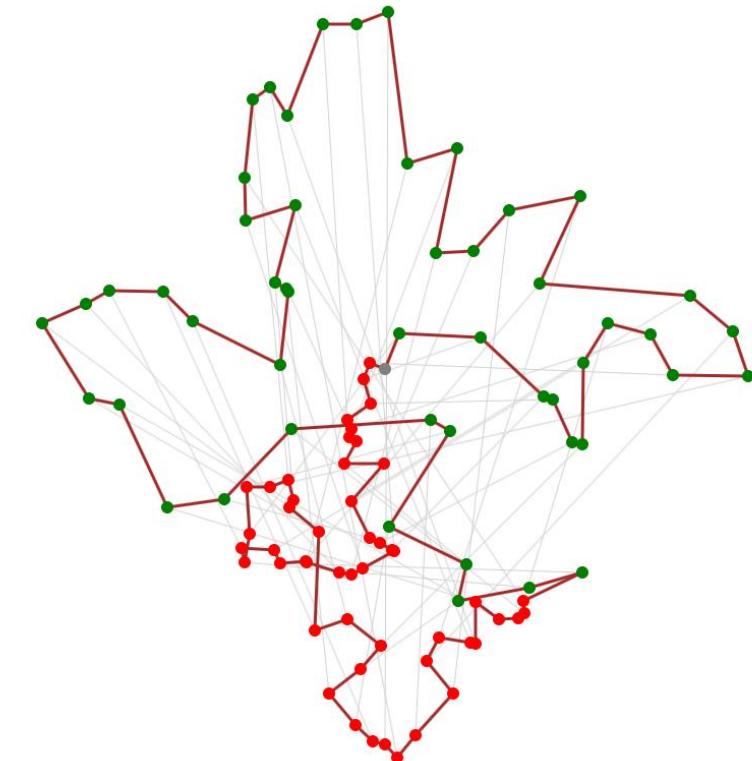
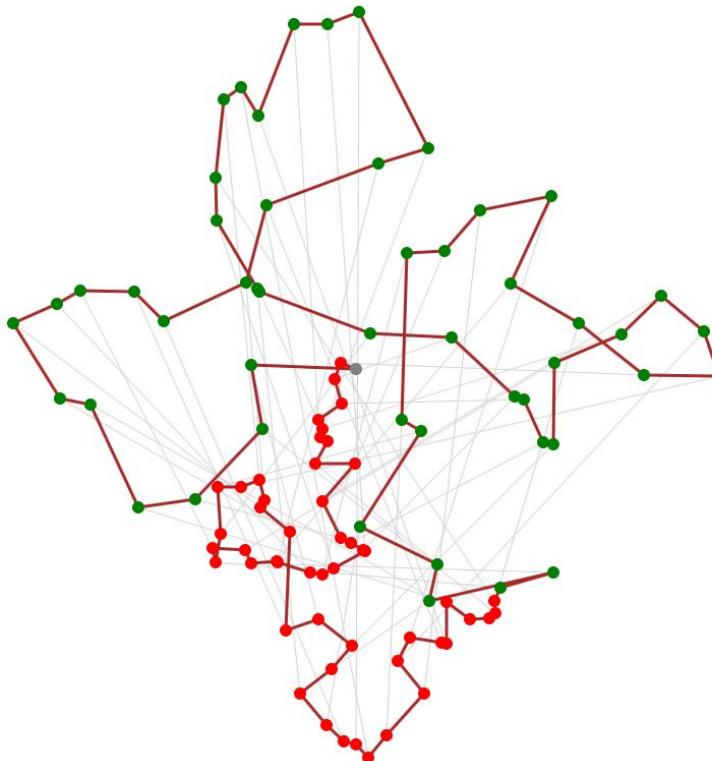
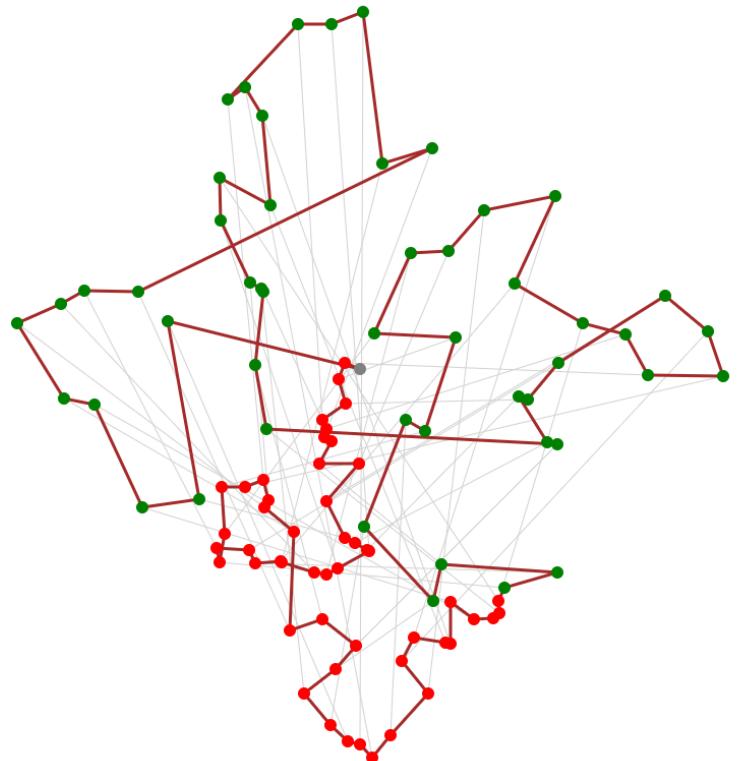
- ここまでの中を適切に実装すると、移動距離が5500~5900程度になります※



解説者が本解法で確認した最良解  
(移動距離 = 5506)

※温度などのパラメータ、試行回数、乱数次第で結果がある程度変わります

# ここまででの結果の比較

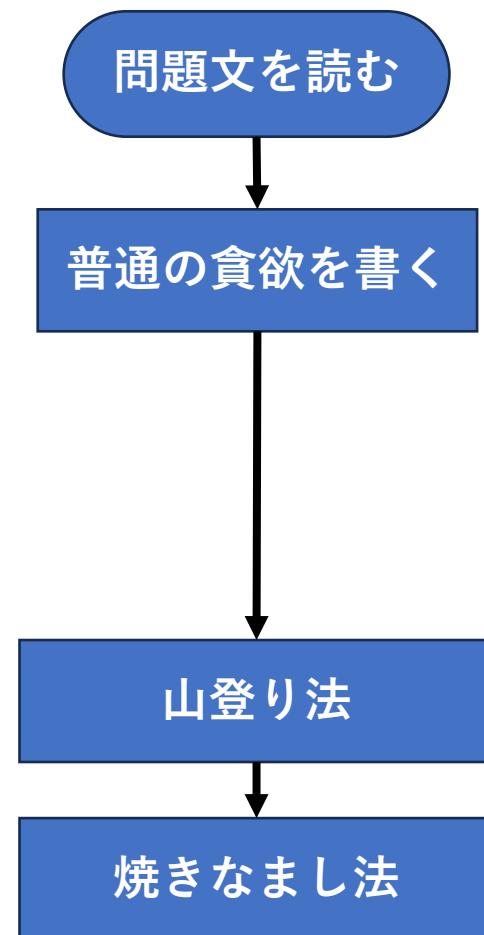


ところで

今回の問題では山登り法・焼きなまし法の解説をしましたが、  
なぜ山登り・焼きなましにしたのか？

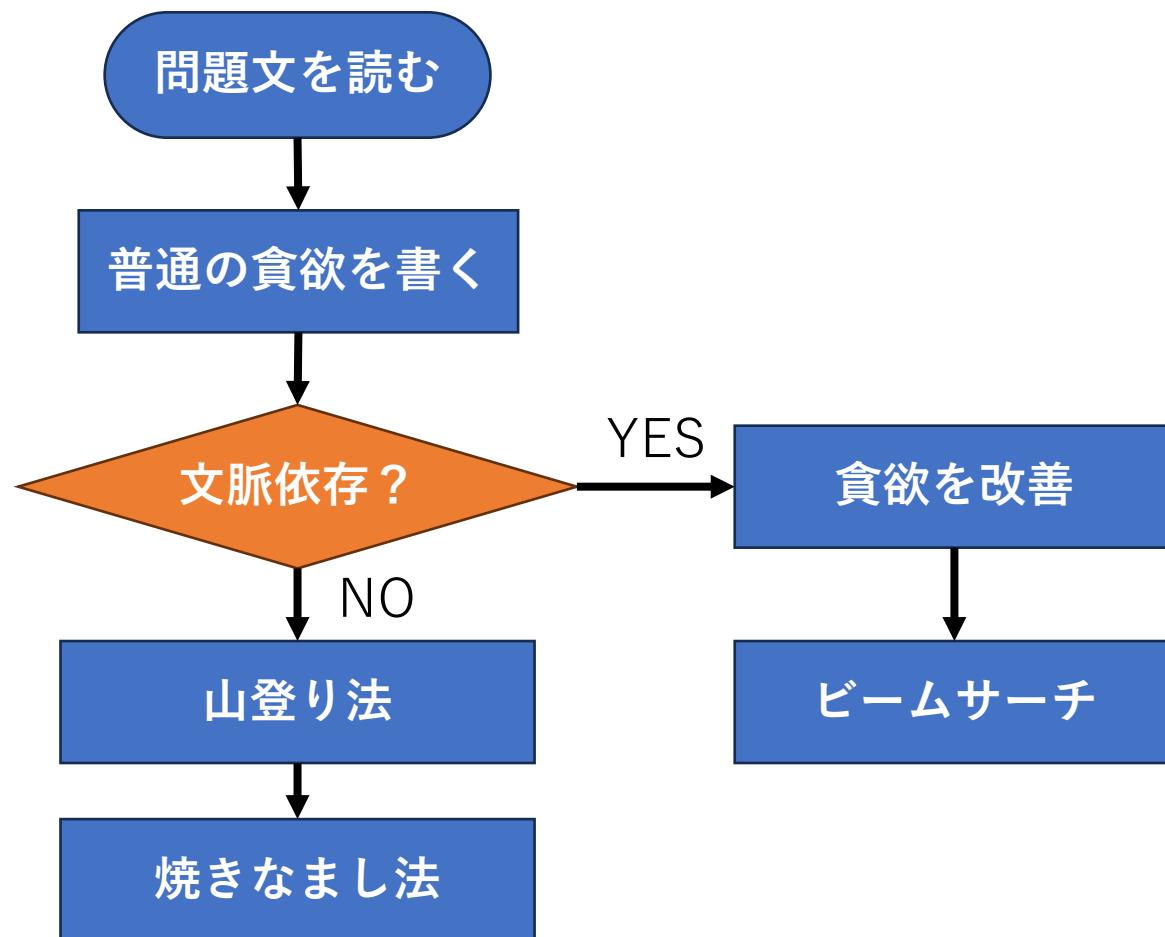
# 解法選択フローチャート

実は、焼きなましが良いかを判断するためのステップがある



# 解法選択フローチャート

実は、焼きなましが良いかを判断するためのステップがある



# 文脈依存かどうかを考えよう

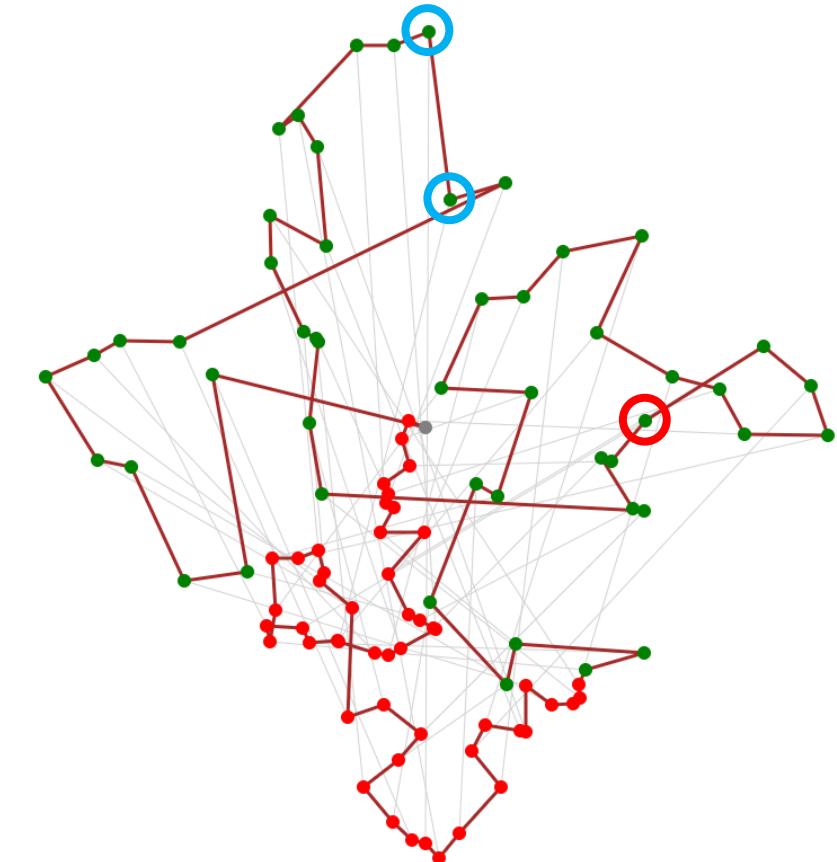
- 文脈とは？
  - 解の一部を変化させたときに他の部分にどれだけ影響するかのこと
- 変化させた部分以外が大きく影響を受けるなら**文脈依存である**（文脈が強い）
- 変化させた部分以外が影響をほとんど受けないなら**文脈依存ではない**（文脈が弱い）

参考資料: 短期AHCで勝つためのテクニック

[https://speakerdeck.com/shun\\_pi/duan-qи-ahcdesheng-tutamenotekunituku](https://speakerdeck.com/shun_pi/duan-qи-ahcdesheng-tutamenotekunituku)

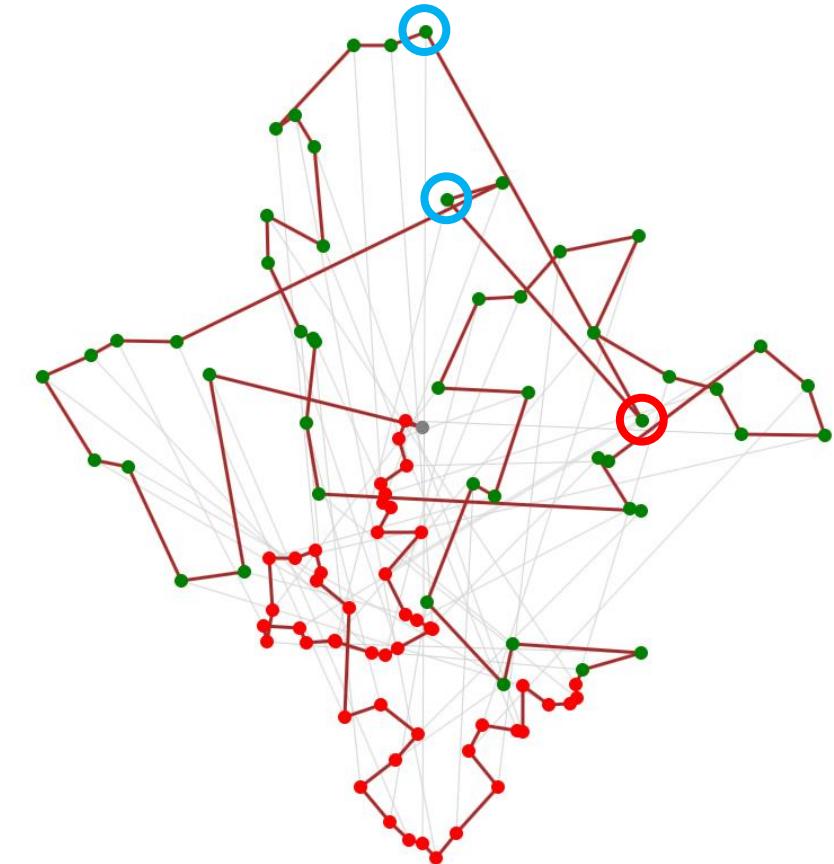
# 今回の問題は文脈依存か？

例えば、  
赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみると・・・



# 今回の問題は文脈依存か？

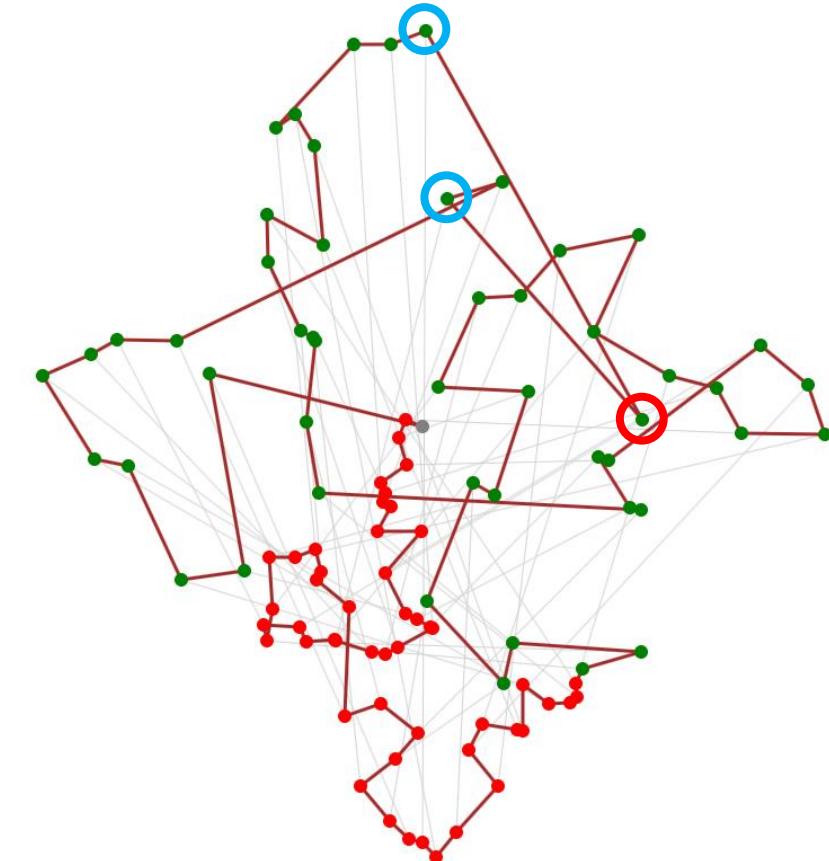
例えば、  
赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみると・・・



# 今回の問題は文脈依存か？

例えば、  
赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみると・・・

変更した部分が絡んでいるところ  
以外は変化していない

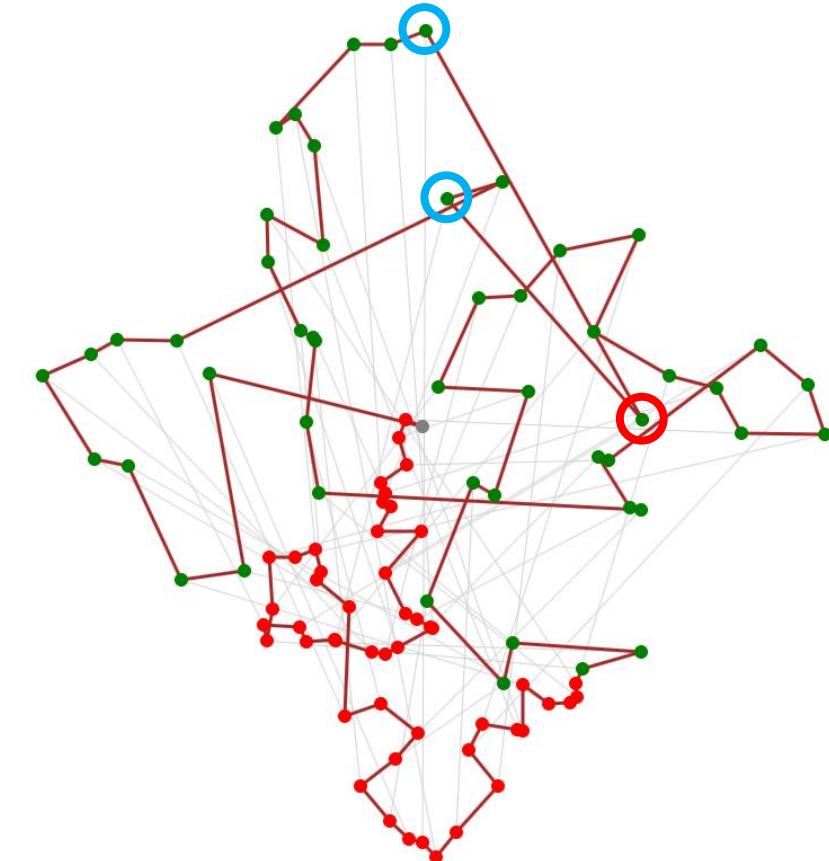


# 今回の問題は文脈依存か？

例えば、  
赤丸で示した配達先を訪れる順序を  
青丸で示した配達先の間に  
変更してみると・・・

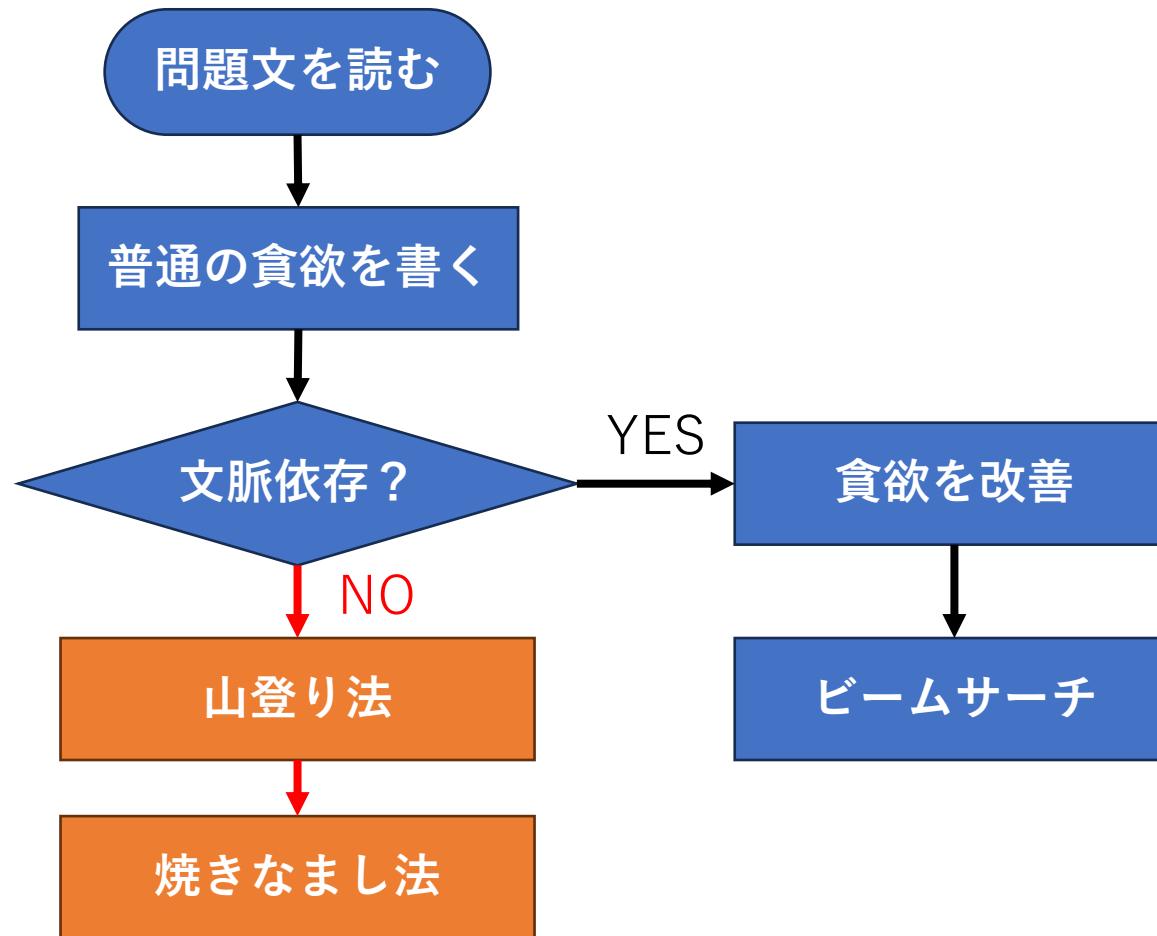
変更した部分が絡んでいるところ  
以外は変化していない

→文脈依存ではない！



# 解法選択フローチャート（再掲）

文脈依存ではないことが分かったので、焼きなましが良さそう



# ここからさらに改善するためのヒント

## 次のレストランに行く前に別の配達先に行くようにする

- ・今日は簡単のため、レストランを全部回ってから配達先に行くことにしていましたが、配達先に対応するレストランへ既に行っているなら先にその配達先に行ってもよいです
- ・対応するレストランより先に配達先に行ってしまうと条件を満たせなくなってしまうので、そこだけ注意しましょう

# ここからさらに改善するためのヒント

## 使うレストラン・配達先を変える

- レストラン・配達先をどうやって選ぶかを考えてみましょう
- 山登り・焼きなましの最中に  
レストラン・配達先を変えてみるのもよいです
  - 今回は簡単のために固定していましたが、  
他に950個のレストラン・配達先があります

# ここからさらに改善するためのヒント

## 焼きなまし法で使う近傍の種類を増やす

- 実は、配達先の部分だけを考えると巡回セールスマン問題（TSP）とほとんど同じ問題になります
- 典型的な近傍として2-optなどがあるので試してみるとよいです