

# 知能プログラミング演習 II 課題 1

グループ 8

29114060 後藤 拓也

2019 年 10 月 14 日

■提出物 repl

■グループ グループ 8

■グループメンバー

| 学生番号     | 氏名   | 貢献度比率  |
|----------|------|--------|
| 29114003 | 青山周平 | NoData |
| 29114060 | 後藤拓也 | NoData |
| 29114116 | 増田大輝 | NoData |
| 29114142 | 湯浅範子 | NoData |
| 29119016 | 小中祐希 | NoData |

■自分の役割 最良優先探索と A\*アルゴリズムのパラメータ調整

■追加機能 csv ファイル出力による親ノードと対応する子ノードの整理

# 1 必須課題 1

## 1.1 最良優先探索

最良推定法は各ノードにおけるヒューリスティック値をもとに探索を進めていく方法で、直前のノードから次のヒューリスティック値だけを見て、行き当たりばったり探索を進める山登り法とは異なり、過去のデータ（これから訪れる可能性をもったノード）のヒューリスティック値を OpenList に保存するので、山登り法よりも最適な探索が可能である。初期のパラメータを用いて探索を行うと下図 1 のような木構造を持つ探索となる。

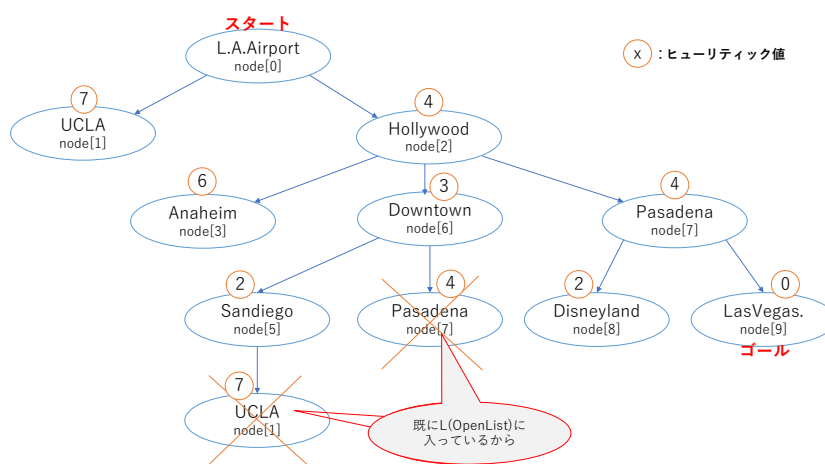


図 1 初期パラメータにおける木構造モデル

探索経路の解としては、

[L.A.Airport → Hollywood → Pasadena → Las Vegas]

と 4STEP で進めるはずが、探索ノード（親ノードの推移）としては、

[L.A.Airport → Hollywood → DownTown → Sandiego → Pasadena → LasVegas]

と、7STEP を踏んでいる。Hollywood から次に進む際に Pasadena へ進むのではなく、1 度 Downtown へ進んでいるのである。そのため、この改善策として、「Pasadena:node[7] のヒューリスティック値:h(7) を Downtown:node[6]:h(6) よりも小さくする」方法を取る。そのパラメータ調整後の木構造が下図 2 のようになる。

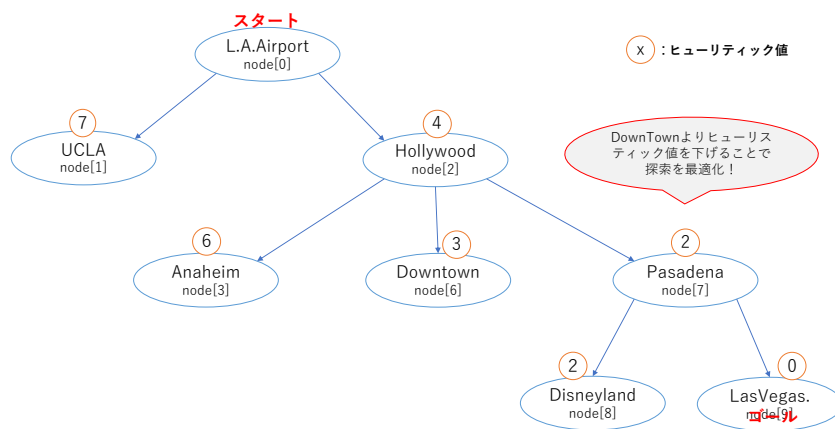


図 2 パラメータ調整後の木構造モデル

探索ノード (親ノードの推移) として,

[L.A.Airport → Hollywood → Pasadena → Las Vegas]

と最適パスとなった.

欠点としては, そのノードに固有なヒューリスティックな値を使って探索を進めていくので, すべてのノードのうちで最小のヒューリスティックの値をもつノードが無限に生成される場合は, 目標ノードに到達できない.

## 1.2 A\*アルゴリズム

A\*アルゴリズムはそのノードまでのコストの合計とそのノードのヒューリスティック値の和をとる. 最良優先探索の弱点でさえも初期ノードからのコストを考慮することで, 正解にはたどり着ける.

初期のパラメータを用いて探索を行うと下図 3 のような木構造を持つ探索となる.

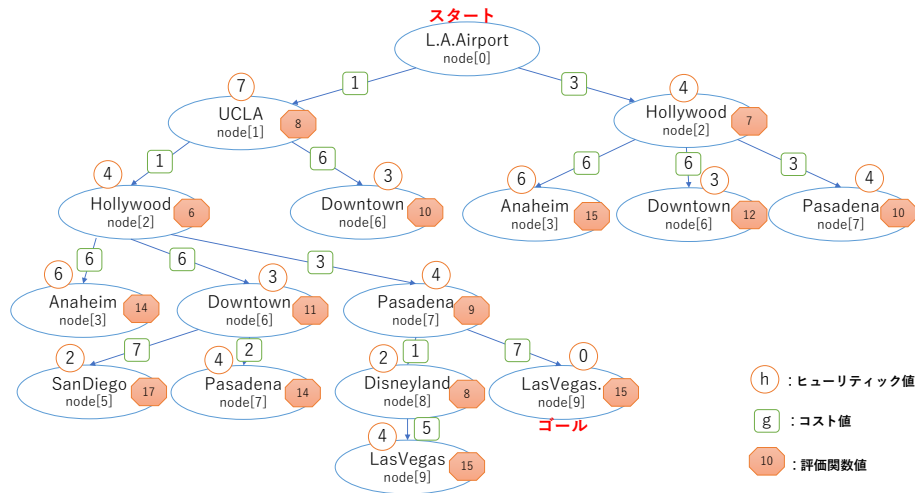


図3 初期パラメータにおける木構造モデル

探索経路の解としては,

[L.A.Airport → UCLA → Hollywood → Pasadena → Las Vegas]

と 5STEP で進めるはずが, 探索ノード (親ノードの推移) としては,

[L.A.Airport → Hollywood → UCLA → Hollywood → Pasadena → Disneyland →  
Dawntown → LasVegas]

と, 8STEP を踏んでいる. 余分な探索として, 2 箇所が上げられる.

1. 始めに Hollywood:node[2] に進まずに, UCLA:node[1] に進んでほしい.
2. 最後に Pasadena:node[7] から Disneyland:node[8] に進まず, LasVegas:node[8] に進んでほしい.

よって, これらを改善するために以下の 2 つを変更してみる.

1. UCLA:node[1] のヒューリスティック値:h(1)=7 を Hollywood:node[2] のヒューリスティック値:h(2)=4 よりも小さくする. (既にコスト g の値は小さいので.)
2. Pasadena:node[7] から Disneyland:node[8] のコストを Pasadena:node[7] から Disneyland:node[8] のコストよりも小さくする. (既にヒューリスティック値は h(9) よりも小さいので.)

その結果, 下図 4 のようになった.

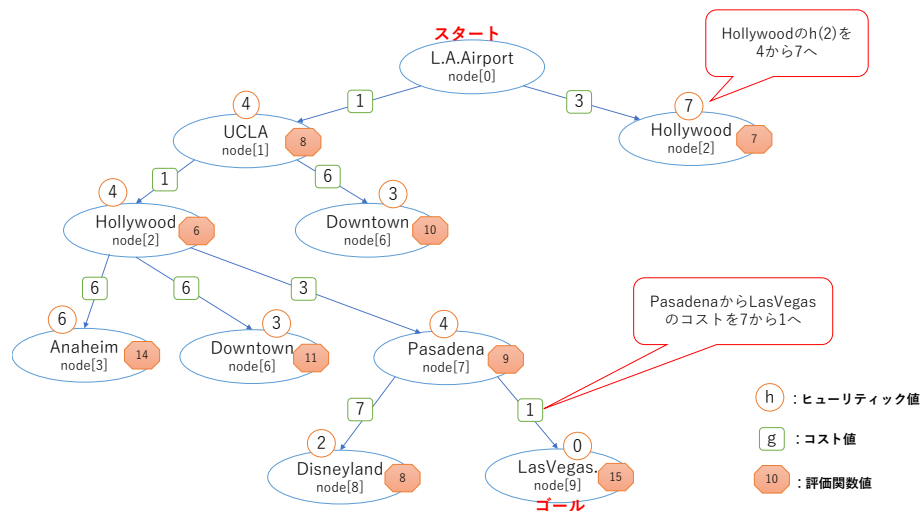


図 4 2つのパラメータ変更後の木構造モデル

これによって、探索経路の解と同じ 5STEP の探索ノード (親ノード) の推移を作ることができた。

なお、始めの分岐で UCLA:node[1] を選ばず、Hollywood:node[2] を選ぶようにパラメータを調整することで、4STEP 探索も可能となる。(木構造は最良優先探索と同様になるので省略。)

### 1.3 CSV 出力

実際に探索手順を追っていくために、“知能処理学”で学んだような親ノード、子ノードと展開されるリストとそれぞれのコストやヒューリスティックの値を表を実際に手で書いていた。それをプログラムで出力させてみようとする前は自分の担当だった A\*アルゴリズムを用いてプログラムを書いてみた。Search クラスの中の aStar メソッドの中で、該当する親、子ノードを取得し、リストに格納していく。探索がすべて終わったところで、用意した exportCSV メソッドを呼び出し、出力させる。プログラムのポイントは、全てのノードデータを集めてから、1 枚の csv ファイルを出力させなければならないが、各 STEP ごとに子ノードの数も違い、またどこまでが 1STEP の情報なのか判別しないといけない。その工夫としては、1STEP が終了するごとに子ノードリストに null を入れることで解決した。exportCSV メソッド内でループを回して順にリストの中身を取る際に、子ノードリストに含まれている null の有無で条件判別ができ、STEP ごとに csv ファイルを改行させることに成功した。

試して分かったことだが、Node クラスでノードインスタンスを作る際に、そのノードにコスト:g, ヒューリスティック値:h, 評価関数値:f がすでに含まれているので、わざわざ別個でそれらのデータを取る必要はなかった。それらの結果、ただ親ノードとそれに続く子ノードを出力し、csv ファイルに保存するだけのお粗末な仕上がりとなってしまった。ただそのおかげで、コストやヒューリスティック値の使用有無によらず、どの探索でも使用できる汎用性の高いプログラムとなった。

## 2 必須課題 2

### 2.1 GitHub の導入

増田君の提案と実行力のおかげで、GitHub を用いたプログラム・レポート共有ができた。GitHub は自分は初めて使うツールだったので、branch や merge などわからないところが多く、苦労した。特に苦労した点は、レポジトリをリモートに登録する作業である。今回は増田君がベースで作ってくれた master の内容を自分のローカルにクローンし、その中に含まれている master へのコネクト情報(隠れディレクトリ.gitに含まれている内容)を使って作成しないといけないことに、なかなか気づけなかったことだ。自分はネットで GitHub の使い方を調べているときに、レポジトリを作成させる方法のもう1つ、クローンではなく、新規作成させる方法を見て何度も試していたので、なかなか成功しなかった。そこさえわかれば、add, commit 操作やブランチ移動の checkout 操作など、そのほかにも上手く使いこなせることができた。

### 2.2 Slack の導入

また、グループの進捗確認を行う際に、始めは LINE を駆使していたが、それでは誰がいつ言った内容に回答しているのか、樹形列順でごちゃごちゃしてしまうということで、これまた増田君の提案のもと、Slack を使い始めた。これは Git よりも直観的でわかりやすく、また、1人の発言に対しての回答が取りやすく、整理されていてわかりやすかった。

## 参考文献

- [1] Java による知能プログラミング入門 ー著：新谷 虎松
- [2] 知識システムの実装基礎 ー著：新谷 虎松/大園 忠親/白松 俊

- [3] java CSV 出力 –著 : TECH Pin  
[https://tech.pjin.jp/blog/2017/10/17/\[java\] csv 出力のサンプルコード](https://tech.pjin.jp/blog/2017/10/17/[java] csv 出力のサンプルコード)
- [4] java List 配列処理と変換 –著 : Samurai Blog  
<https://www.sejuku.net/blog/16155>
- [5] Git でブランチを作成する方法 –著 : ProEngineer  
<https://proengineer.internous.co.jp/content.columnfeature/7633>
- [6] Git レポジトリの変更と取得 –著 : GitHub ヘルプ  
<https://help.github.com/ja/articles/getting-changes-from-a-remote-repository>
- [7] LaTeX 箇条書き –著 : LaTeX コマンド集  
<http://www.latex-cmd.com/struct/list.html>