

K 番目経路探索

ここでは、K 番目経路探索を行う方法を解説します。配布プログラムである K 番目経路探索アルゴリズムは Java 言語で書かれており、ここでは Eclipse により動かす方法を説明します。

1 K 番目経路探索とは

K 番目最短経路探索では、最短経路探索を拡張して、K 番目までの短い経路を順に列挙します。確定的に経路選択枝を列挙する方法のひとつです。

OD 交通量をネットワーク配分するとき、OD 間の経路の設定が必要になります。大規模なネットワーク上の経路選択では、OD 間で考えられる経路は膨大であり、その中から適切な経路を抽出し、フローを流す手法が求められます。

その手法の一つとして、経路選択モデルの構築があります。経路選択モデルの構築にあたっては、OD 間の経路集合の中から選択枝集合を形成し、その選択枝集合と実際の選択経路を用いて、経路選択モデルのパラメータを求めます。その選択枝集合の形成の際に K 番目最短経路探索のアルゴリズムを用います。例えば、アルゴリズムにより抽出した K 番目までの最短経路全てを選択枝集合とする方法や抽出した選択枝集合から類似性の低い経路のみを再び抽出して選択枝集合とする方法があります。詳しくは、「4.3.1 確定的に経路選択枝を列挙する方法」を読んでください。

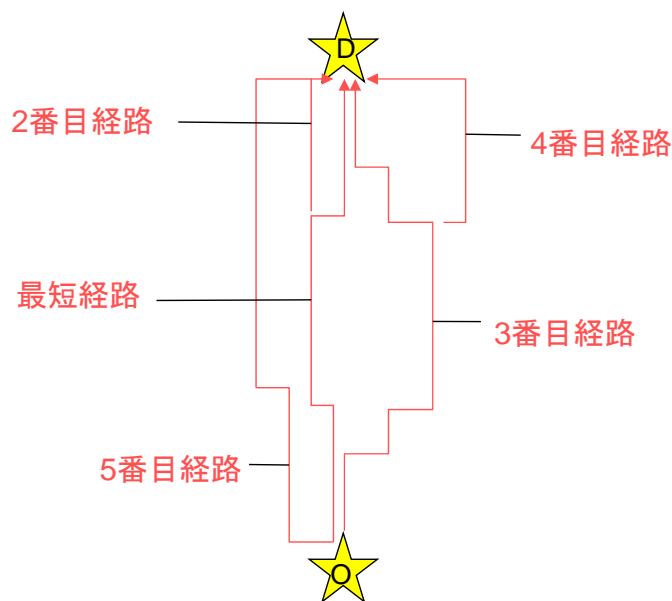


図 K 番目経路列挙のイメージ ※ 山川佳洋, 羽藤英二(2008)を参考に作成

2 事前準備

配布プログラム ksp.zip の中には、y-path と y-path1 のフォルダがあります。二つを workspace フォルダに保存して下さい。y-path が Eclipse でインポートするプロジェクトになります。Javainstall.docx で説明している方法でインポートしてください。

Input データとして、node.csv, link.csv, oddata.csv を利用します。全て y-path1 フォルダ内の Input フォルダに保存されています。Input データは次の構造になっています。

node.csv は 1 行目に総ノード数, 2 行目以降にノード番号とノードの XY 座標が示されています。

48	・・・全ノード数 ※「48,,」と余分なカンマが入らないように注意
1,13,142	Excel を用いて, csv を作成すると入ってしまう
3,15,182	・・・ノード番号, X 座標, Y 座標 ※ X 座標, Y 座標は整数を用いること

図 ノードデータ(node.csv)

link.csv は 1 行目からリンクを指定します。「(リンクナンバー), (発側ノード NO), (着側ノード NO), (リンクコスト)」で与える。リンクコストは単純な距離として一般に与えられますが、独自に設定してもよいです。この値を元に最短経路探索を行います。なお、コスト値は整数としてください。また、ネットワークのリンクはすべて単方向リンクとして扱われます。下図のように、ノード 1-ノード 16 に双方向のリンクがある場合は、1→16 と 16→1 の 2 本の単方向リンクが必要になります。

0,1,16,103	・・・ (リンクナンバー), (発側ノード NO), (着側ノード NO), (リンクコスト)
1,16,1,103	を与える ※方向別リンクとみなされるので, 方向別に全リンクを含める
2,3,1415,408	ように注意

図 リンクデータ(link.csv)

oddata.csv ファイルで、経路探索を行う始点(O)と終点(D)を指定します。1 行目が指定する OD の総数を示し、2 行目以降で OD を示します。OD は、「(OD ペアのナンバー),(O のノード番号),(D のノード番号)」と示します。

3	・・・「3,,」のように余分なカンマがつくとエラーが起きるので注意
1,16,1	
2,1116,2425	・・・カンマの前後にスペースは不要
3,2831,1034	

図 OD 作成の注意 (oddata.csv)

3 計算実行

まず、経路列挙範囲の指定を行います。プロジェクトの `y-path1` の中の(デフォルト・パッケージ)の中に `ksp.java` があります。その中の 14,15 行目で `totalNum` と `Asp1` が指定されています。 `totalNum` が上限経路列挙数、 `Asp1` が経路列挙の範囲を指定しています。これは、“最短経路のリンクコストの `Asp1` 倍までの経路を列挙する。ただし、列挙数は最大で `totalNum` 本までとする”という意味です。この二つの数字は自由に設定してください。 `Asp1` とすると最短経路のみが列挙されます。

```
int totalNum    = 10000; //経路数
double Asp1     = 1.2;   //最短経路を基準とした条件 (Asp1>1)
```

図 経路列挙範囲の指定 (`ksp.java`)

以上で設定は終わりです。プログラムの実行は、メニューバーの”実行(R)”→ ”実行(S)”→”1 Java アプリケーション(1)”から行ってください。コンソールに「k 番目最短経路探索 1,2,...」と書き込まれ、しばらく待てば計算終了となります。ここで書き込まれる数字は全 OD に対して、列挙された経路数を示します。

`y-path1` フォルダの中の `Output` フォルダに下記のような `path.dat` ファイルが作成されます。一番左側が経路番号、次の二つの数が O ノードと D ノードを示します。{}に囲まれた 4 つ目以降の数が OD 間経路を示し、数はリンク番号を示します。一番右側の数字は経路上のリンクコストの和です。なお、`path.dat` ファイルが `output` フォルダに存在している状態でプログラムを回すと、現在のデータの下から結果が追加されます。ご注意ください。

{ 1 1 9 { 1 2 5 10 } 7 }
{ 2 1 9 { 1 4 9 12 } 7 }
{ 3 1 9 { 3 6 9 12 } 8 }

経路 NO O ノード D ノード 経路上のリンク NO 経路コスト

図 経路列挙の結果(`path.dat`)