

最短経路問題の解決結果をユーザビリティ向上のために利用することの提案

[北海道函館中部高等学校 2年 吉田 陽貴]

1. 背景/目的

世の中には**複雑な機械/システム**がある

(例) 自動販売機(以下自販機)の購入時

| | |
|-------|------------------------------|
| 現金 | 現金を投入→飲料ボタンを押す |
| 電子マネー | 飲料ボタンを押す→電子マネー種類選択→ICカード等タッチ |

※操作方法は一例

[同じ操作に異なる行動が必要!]

次にすべき行動を目立たせる?

[課題]

大規模なシステムでは、手作業でそれらを求められない
→ フローチャートをグラフとみなすことによって、
最短経路問題を利用してはどうか?

2. 手法/開発

(1). 最短経路問題解決には**ダイクストラ法**を用いた [1]。
また、言語は**C++**を用いた。

(理由: 以下の要素を考えたため)

- 高速な処理が期待できる
- 計算量が少ない
- コストは必ず負ではないし、1とは限らない
- 操作を始める始点は1箇所である

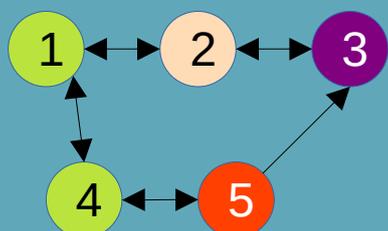
(*ここで、コストは利用者にとって必要な労力とみなす)

(2). その他に必要な要素を考えた。

- ①. 経路復元機能をつけた
-- **次に何をすべきかは最短経路から判断**するため。
- ②. 各目的頂点までに、各々経由されるべき複数頂点を経由したときの最短経路を1つにまとめるようにした。(図1参照)

- 目的となる頂点は複数の可能性があるため。
- 各々に目的頂点に達するまでに、経由されるべき頂点が複数ある可能性があるため。

[図1 2 - (2) - ②の内容について]



頂点1を始点とし、コストはすべて1とする

目的頂点(A): 3
解: 1,2,3 (コスト2)

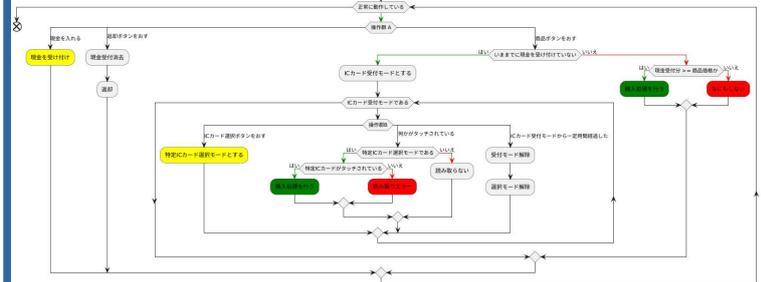
目的頂点(B): 5
経由されるべき点: 2

最短経路(頂点1,2間):
1,2 (同1)

最短経路(頂点2,5間):
2,1,4,5 (同3)

→ 解: 1,2,1,4,5 (同4)

図2 自販機のフローチャート例



上のような例 (図2 [注A]) を使ってテストした。この際フローチャートの各処理等に番号を順番に付してプログラムに入力した。このテストでは、コストは全て1として、始点頂点を0、現金及び電子マネー (左記の例参照) で購入完了となる条件が異なるから、それらを2つの目的頂点に分けて ((例)の赤色)、さらに各1つの経由されるべき頂点 ((例)の白色) を条件として考えた。

3. 結果

#1/2 Shortest: 8

-- Route --

1->2->3->1->2->6->18->19

#2/2 (以下略)

左記が出力となった。
※この例は、3を経由し、19を目的地としている。そのため、頂点3のあと、また頂点1,2を通過している。

→ このテストケースでは、目的頂点2箇所共に最短経路を求められていた。

計算量について、辺数を M とすると、特定頂点2点間の最短経路を求める計算量は $O(M \log M)$ [1] であったから、(経由されるべき頂点数+目的頂点数)の合計を X とすると $O(XM \log M)$ となることがわかった。

4. 考察

・ダイクストラ法を適用して、フローチャートから最短経路を求めて、次にすべき行動は求められているのではないか。

・このプログラムの計算量 $O(XM \log M)$ は、例えば、 $X=10^2$, $M=10^5$ といった複雑なシステムだとしても $10^2 * 10^5 * \log(10^5) = 5 * 10^7 < 10^8$ であるため、例えば家庭用PCでプログラムを実行しても1秒以内に計算が終わる高速なプログラムであると考えられる。

5. 今後の展望

・次にすべき行動を明確化することが、どれだけユーザビリティ改善につながるかを知る必要がある。

・最短経路のプログラムについて更に扱いやすいよう、手入力ではない方法でグラフを入力できるような仕組みを整えること等での点で改善が必要である。

参考文献/注

[1] 米田優峻, 競技プログラミングの鉄則, pp.362-369, 株式会社マイナビ出版 (2022)

[注A] <https://www.plantuml.com/plantuml/uml/> を用いて、plantUMLをPNGに変換した。