

# 複数経路を考慮した 運転整理計算機支援のための乗客流推定法

田中 峻一\*, 熊澤 一将, 古関 隆章 (東京大学)

Passenger Flow Estimation with Multi Paths for a Computer-aided Train Rescheduling System  
Shunichi Tanaka, Kazumasa Kumazawa, Takafumi Koseki (The University of Tokyo)

## 1. まえがき

天災や列車事故、車両故障などにより、列車のダイヤに乱れが生じた際には、計画ダイヤに変更を加えて、ダイヤの乱れの波及を防ぐ運転整理が行われる。現在は運転整理を作成するアルゴリズムが確立されておらず、指令員の経験による定性的な判断により行われている。

そのため、指令員の運転整理業務を支援するシステムが必要とされており、乗客の立場に立った定量的な評価を行うアルゴリズム<sup>(1)</sup>と、運転整理案を作成するアルゴリズムを組み合わせたシステム<sup>(2)</sup>の開発が進められている。

本研究では、乗客の立場に立った定量的評価を行う際の乗客流推定について、従来の手法では表現できなかった場合の解決法を提案する。

## 2. 運転整理案の評価

〈2.1〉乗客の立場に立った運転整理案の評価 本研究では乗客の立場に立った運転整理案の評価を行う。従来の研究では、次節における乗客行動シミュレーションに基づいて、乗客の所要時間、乗換、混雑の3要素の不効用を考え、それらを時間換算した和が小さくなる運転整理案がより良い整理案として扱う。

〈2.2〉乗客行動シミュレーション 本システムでは、出発駅に均等な間隔で出現し、目的駅までの所要時間・乗換による損失が小さくなるよう行動する乗客を仮定している。このモデルは、時刻表を見ずに駅にやってきてその路線の情報をよく知っている乗客に対応し、都市鉄道でよく見られるものである。乗客行動推定にはグラフ構造を用いる。図1に示すように、各列車の各停車駅での到着・出発をノードで表し、走行・停車・乗換をリンクで表現する。このグラフに対し各駅のノード間の最短経路を、ダイクストラ法を用いて探索することで損失が最小となる経路を求めることが出来る。経路ごとにODデータから乗客を割り付け各列車の乗客数を推定する。

## 3. 列車運転乱れ時における乗客流推定

〈3.1〉従来の乗客流推定における問題点 先行研究では、すべての乗客が所要時間と乗換の不効用が最小となる経路を選択する行動をとっていた。この方法で行うと、優等列

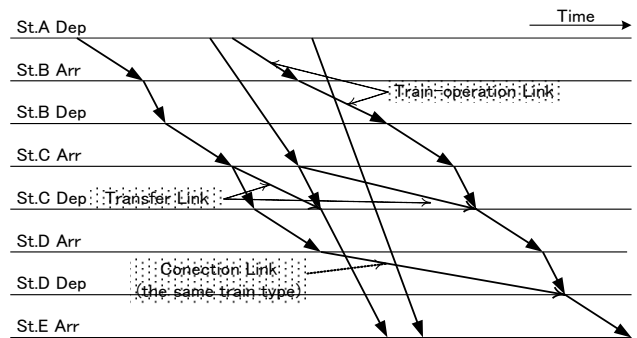


図1 乗客流推定グラフ

Fig.1. Graph of passenger flow estimation

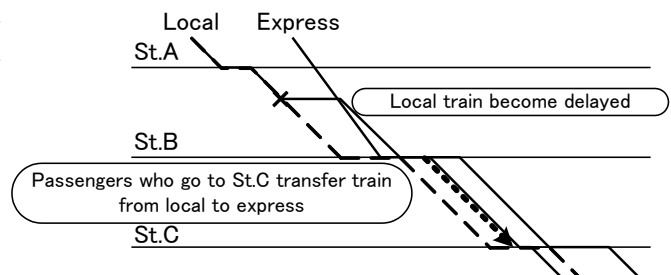


図2 乗客流の変化

Fig.2. Change of passenger flow

車に人が偏ってしまい、実際の状況よりも積み残しの発生しやすい状況になってしまい、複数の積み残しが依存関係をもつような大規模な乱れに対しては何らかの非現実解しかでてこないことになる。

また、乗客は混雑に関係せず、評価値に従って全員の行動を決定させているため、図2に示すようにわずかな評価値の違いによって全員の行動が変更するということが起こりうる。

これらの問題を解決するために、最短経路の他に評価値が比較的小さい経路を探索し、それらの経路にも評価値に応じた乗客を流す方法を提案する。

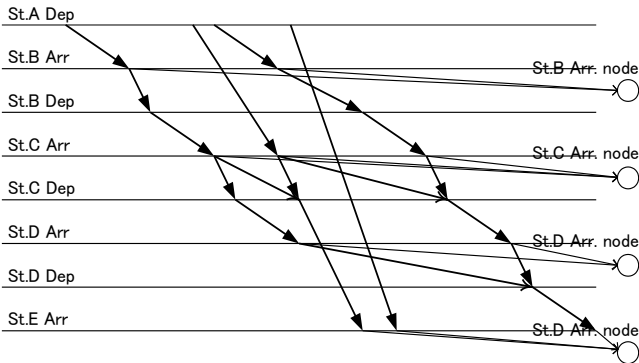


図3 駅到着ノードの導入  
Fig.3. Introduction of station-arrival nodes

〈3.2〉 k-th Shortest Path アルゴリズムの導入 グラフ理論において k 番目にコストが小さくなる経路を求めるアルゴリズムを k-th Shortest Path アルゴリズムといい、Martins et al. のアルゴリズム<sup>(3)</sup>などが知られている。この k-th Shortest Path アルゴリズムを用いて、比較的評価値の小さい経路を保持する経路を考える。

従来の乗客流グラフでは、ノードは各列車の着発であり、各列車発ノードから各列車着ノードまでの最短経路を探索し、到着駅において、それぞれの着ノードまでの不効用値を考え、最小となる経路を取っていた。

そのため、従来のグラフにおいて k-th Shortest Path アルゴリズムを導入しても、出発の列車と到着の列車が同じ経路の中で複数の経路を求めることになり、有効な解をもとめることができない。

そこで、図3のように、各駅のノードを1つにするような駅到着ノードを導入し、各列車着ノードから駅到着ノードまで重み0(または重み一定)なリンクを追加したグラフを考え、各列車発ノードから駅到着ノードまでの複数の経路を求めることによって、k-th Shortest Path アルゴリズムを導入することができる。

このようなグラフの変更を加えることによって、出てくる経路においては、優等列車に乗り換えた後に元乗っていた列車に乗り換えるような経路や、接続している駅で乗り換えずに後の優等列車に乗った後に別の駅で先ほど接続していた列車に乗る経路(図4において駅Aで普通列車に乗車した客が駅Bで快速に乗換を行わず、駅Cにおいて急行に乗り換え、駅Eで急行から快速に乗り換えるような経路)といった不適切な経路が多く探索されることがある。さらに、ある列車から別の列車への乗換駅の候補が複数ある場合、現在のグラフでは待ち時間と乗車時間は同じ重みをもつ時間として扱っているため、評価値は同等となり、従来のグラフと同様に、乗換駅の異なるものばかりが出てくることがある。

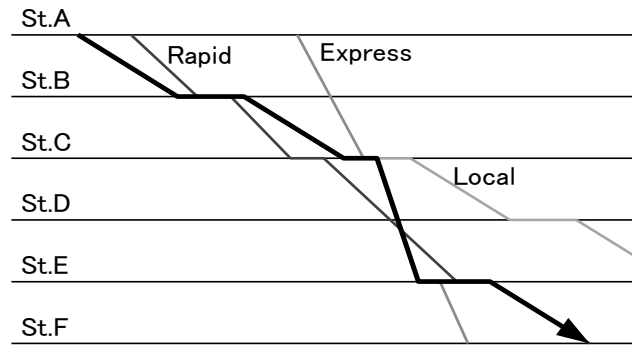


図4 不適切な経路の例  
Fig.4. Example of bad paths

また、k-th Shortest Path アルゴリズムは、一般的に最短経路を求めた上で、2番目、3番目、…の経路を求めるアルゴリズムであるため、最短経路を探すのみに比べると計算時間が増大してしまう問題がある。

〈3.3〉 ダイクストラ法の結果の保持 前節では k-th Shortest Path アルゴリズムの導入を考えた一方、従来のグラフにおいては、列車発ノード1つに対して、同じ駅における列車着ノードが複数あるため、最短経路を同一始点最短経路探索アルゴリズムであるダイクストラ法で求めることにより、探索をおこなう計算量は一定のまま、着ノードが違う経路、すなわち、目的駅へ到達するときに乗っている列車が異なる経路を求めることができる。

しかし、この方法であると、乗換駅の違いによるような不効用値が近くなるような経路を探索することが難しくなる。

#### 4. まとめ

運転整理業務を支援するためのシステムに乗客流推定に基づく評価を導入する考え方を提案しその乗客流推定において複数の経路を考慮する方法について詳述した。本研究では2つの方法を紹介したが、両方法とも一長一短がある。今後は、乗客流推定に待ち時間の考慮を行うことを考え、より現実に即した乗客流推定の提案・実装を行う。

#### 文献

- (1)原和弘, 熊澤一将, 古関隆章:「運転整理計算機支援のための列車運転乱れ時における乗客流推定法」, 電気学会全国大会, vol.5, pp.30-31 (2008)
- (2)Y. Nagasaki, M. Eguchi, T. Koseki, “Automatic Generation and Evaluation of Urban Railway Rescheduling Plan”, International Symposium in Speed-up and Service Technology for Railway and Maglev System, pp.301-306 (2003)
- (3) E. Q. V. Martins, M. M. B. Pascoal, J. L. E. Santos, “An algorithm for ranking loopless paths”, Res. Rep., Univ. de Coimbra (1999)