

運転整理に対するヒューリスティック解法の提案

Proposal of Rescheduling Heuristic Algorithm

森下 翔葵[†] 金川 明弘^{††} 滝本 裕則^{††}

Shoki Morishita[†] Akihiro Kanagawa^{††} Hironori Takimoto^{††}

[†]岡山県立大学大学院 情報系工学研究科 ^{††}岡山県立大学 情報工学部

1 はじめに

自然災害や事故などにより列車のダイヤが乱れた際、運転整理と呼ばれるダイヤの乱れ回復を試みる作業を行わなければならない。この運転整理は対象の列車数が多いかつ、安全運行のための様々な制約を考慮しなければならない。加えて、できる限り乗客の利便性を損なってはいけないなどという理由から、非常に複雑な問題となっている。また、リアルタイムに走行している列車に対するダイヤの変更業務であるため、迅速性も要求されている。指令員はダイヤが乱れた際に、情報の収集・整理・伝達を同時に行わなければならないが、通常は指令員の手作業によって行われている。指令員の負担軽減また、将来的な労働人口の減少などという観点からコンピュータによる支援が求められている。

ここ 10 年運転整理に関する研究が多く報告されるようになった [3][2]。近年、混合整数計画法を用いた運転整理に関する研究が多く報告されている [4][5]。これは、計算機性能の向上や混合整数計画法に対する研究が進化したこと、混合整数計画法ではプログラミング量が少なく、解の精度が高いことが大きな要因と考える。その中でも今田ら [6] は、初めて折り返し運転を考慮した運転整理アルゴリズムを提案した。これは、長時間にわたる支障区間が発生した場合に、実際の運転整理でよく用いられている、折り返し運転を考慮している。しかし駅数や列車数が増加したときに、処理時間が増大してしまうという問題がある。

そこで本研究では、より大規模な問題にも対応できるようにするため、運転整理を駅に対して列車を割り当てる組合せ問題としてモデル化を行い、実行時間短縮のためのアルゴリズムを提案する。

2 運転整理

運転整理を行う際には、列車の運転を取りやめる運休や、列車の一部区間の運転を取りやめる部分運休といったダイヤ変更手段を組合せることが多い。具体的なダイヤ変更手段は文献 [7] を参照されたい。そのダイヤ変更手段を組合せた運転整理法は、主に以下のようものが挙げられる。

- **順序変更** 列車の運転順序を変更する手段である。ただ単に列車の運転順序を変更するだけでなく、特急などの優等列車が追い抜くための待避の解除なども行われる。

- **折り返し運転** 列車を途中駅で運転打ち切りとし、反対線への列車として折り返す手段である。支障区間が発生した際、反対線への列車が来ないためよく用いられる手段である。
- **遅延** 列車を駅で待機させる手段である。支障区間が解除された際に、その区間の運転本数を確保することができるため、折り返し運転と組合せて用いられる。

通常は、列車同士の衝突回避などの安全を確保することを重視し、上記の運転整理法を適切に組み合わせなければならない。また、公共的な使命を持った輸送機関であるため、できる限り乗客の利便性を損なわない運転整理を行う必要がある。以上の理由から、運転整理は非常に複雑な問題であり指令員の仕事量、負担ともに大きく増加する。

3 提案手法

3.1 制約条件

モデル化を行うにあたって、本研究で考慮した運転整理に対する条件を表 1 に示す。

表 1: 制約条件

条件
列車の到着から折り返しの発車まで最低 3 分必要
ホーム数を超える数の列車を遅延させることはできない
列車の発車は先行列車の発車から最低 3 分あける
列車の早発は行わない
1 つの番線に複数車両入線できない

1 番目は折り返し運転を行う際に、乗客の乗降等を考慮した制約である。2 番目は遅延を行う際に、列車同士の安全確保を考慮した制約である。3, 4 番目の条件は運転整理全体に対する制約であり、より現実的な運転整理を行うために考慮している。駅の番線数以上の列車が進入してきた時、ホームの手前で停車し待機するという方法があるが、乗客を長時間列車の中で待たせる状態になる。本研究では、乗客への被害を最小

限に抑えることも考慮するため、番線数以上の列車が進入するような状態となった場合、直前の駅の出発時刻を延長するという条件を追加する。それが5番目の条件の意味である。

3.2 モデル化

運転整理に対する研究の多くは、列車の運行状況の変更を行うということで、列車ごとの運行状況でモデル化を行っていた。列車ごとの運行状況でモデル化を行った場合、折り返し運転時に別の車両として認識する処理ができないため、折り返し運転を考慮しない運転整理アルゴリズムが提案されている [8]。今田ら [6] は、これに対し折り返し運転を考慮した運転整理アルゴリズムを提案した。本研究でも折り返し運転を考慮するため、全ての駅のホームに対して、総運行時分分の配列を用意し、そこに列車を割り当てる組合せ問題としてモデル化を行う。各配列に対して表 2 に示す数値表現を用いることでダイヤを再現する。

運転整理の表現法について述べる。遅延させて運行する場合は、遅延している間そのホームに待機している状態に変更する。待機中に同じ方向の列車が駅の番線数を超えて入ってくる状況になった場合は、以前の駅での出発を該当の番線が空くまで待機させるようにする。遅延させる列車数が増加し、始発駅の番線数を超える状態になった場合は不可能な運転整理とし、遅延させる列車数を減らす。

折り返し運転を行う場合は、対象の反対列車の出発時刻まで折り返し待ち状態に変更し、待機している番線に他の列車が進入しないようにする。また、折り返し運転を行う場合乗客の入れ替えなどの処理に時間を要するため、到着から対象列車の出発時刻までの間隔が指定した時間よりも短かった場合、出発時刻を必要な間隔が確保できるまで遅延させる。

表 2: 配列の表現法

数値	意味
0	空
1	列車の到着
2	列車の発車
3	列車の待機
4	列車の折り返し

3.3 運転整理アルゴリズム

以下に折り返し処理と遅延処理のアルゴリズムを示す。アルゴリズムに用いた記号は表 3 に示す。折り返し処理では、到着から発車まで 3 分未満の場合発車を遅らせ調整している。遅延処理では、1 つのホームに複数の列車が進入することを防ぐため、すでに列車が待機している場合は 1 つ手前の駅で待機するようにする。

Algorithm 1 折り返し処理

```

if  $TD_s^n - TU_s^n \leq 2$  then
     $up_s[TU_s^n] = 0$ 
     $up_s[TU_s^n - 1] = 0$ 
     $TU_s^n = TD_s^n + 3$ 
else
     $up_s[TU_s^n - 1] = 0$ 
end if
for  $i = TD_s^n \dots TU_s^n$  do
     $down_s[i] = 4$ 
end for

```

Algorithm 2 遅延処理

```

 $flag = 0$ 
while  $i \neq TF$  do
    for  $i = TD_s^n \dots TF$  do
        if  $down_{s-flag}[i] \neq 0$  then
             $flag++$ 
            for  $j = TD_s^n \dots i - 1$  do
                 $down_{s-flag}[j] = 0$ 
            end for
            break
        else
             $down_{s-flag}[i] = 3$ 
        end if
    end for
end while

```

表 3: 各記号の定義

記号	意味
TU_s^n	駅 s における n 番目の上り列車発車時刻
TD_s^n	駅 s における n 番目の下り列車発車時刻
up_s	駅 s の上り番線専用配列
$down_s$	駅 s の下り番線専用配列
TF	支障終了時刻

3.4 ヒューリスティック解法

各運転整理の手段について、以下で述べるような特徴がある。遅延の処理を行うと、到着時間の遅れは大きくなるものの、計画通りの列車の運行本数を確保することができる。それに対し折り返しの処理を行うと、支障区間の先における運行本数を減らすことになるが、支障区間までの運行本数を確保できる。

既存研究において運転整理手段の決定は、上り列車

と下り列車それぞれで考えられている。図1に始発から3本目の上り列車と下り列車で、異なる運転整理を行った例を示す。縦軸を駅、横軸を時間(分)とし、上から下に向かう電車を下り列車、その逆を上り列車とする。図1において、3番目の下り列車は駅AB間において折り返し処理をしたため、当初の運行計画を取りやめ、3本目の上り列車として運行を行っている。それに対し3本目の上り列車は駅CD間において遅延処理をしたため、出発時刻を遅らせ元々の計画通りの運行を行っている。

ここで図1中の太線部はともに、元々は3本目の上り列車が運行する予定だったものであることがわかる。このことから、上り列車と下り列車で異なる処理を行った場合、被りが生じ駅AB間と比較して駅CD間での乗客の輸送力が減少していることが分かる。運転整理において、できるだけ乗客への影響を減らすことが重要でありこの運転整理が好ましいとは言いがたい。このような状態を回避するため、本研究では上り列車と下り列車の始発からn番目の列車では、同じ運転整理手段を用いることで各駅間での輸送力の差に大きな影響がでないようにする。

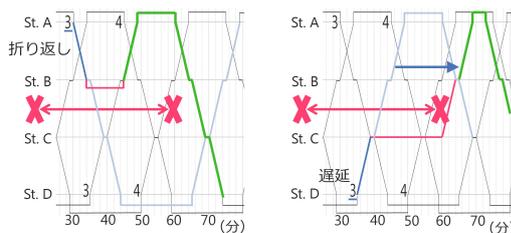


図1: 異なる運転整理手段の選択

3.5 目的関数

目的関数については、既存研究 [6] と同じく着遅延時分の総和を最小にすることを目的とし評価を行う。また、運休時の遅延時分についてはパラメータ α として、指令員が任意に決めることができるように設定する。列車の運行本数が多く、ある程度の運休を許容し運転整理を行う場合は α を高めに設定し、運行本数が少なく運休をすることで乗客に大きな支障を与える場合は低めに設定することを想定している。着遅延時分の計算法について述べる。まず作成したダイヤの全駅全ホームにおける到着時刻の総和を求める。その総和から、元のダイヤの全駅全ホームにおける到着時刻の総和の差分を取る。その差分に運休数と重み α の積を加算することで着遅延時分の総和とする。

4 実験

4.1 実験方法

本アルゴリズムの高速性、有効性を確認するため、既存研究 [6] で解が求められている、駅数4、列車本数14本で実験を行う。図2にダイヤを示す。縦軸は駅、横軸は時間を表し1マスの時間は2分とする。この路線は首都圏近郊の実際の路線を参考に作成されたものである。ここで示している駅数は折り返し運転に対応している規模の大きい駅数のことであり、駅間の運行時間を長くすることによって、折り返し運転に非対応の規模の小さい駅も考慮することができる。そのため全ての駅を考慮すると15から20駅くらいの規模の問題である。途中駅には上り、下り列車専用の番線だけでなく、折り返し列車用の番線も用意されているものとする。中央の駅間で60分の支障を発生させ、重み $\alpha = 30$ とする。なお、配線やダイヤなどの詳細は文献 [6] にて、ただ一例のみが明示されているため、比較においてもこの一例のみで行ったこととした。

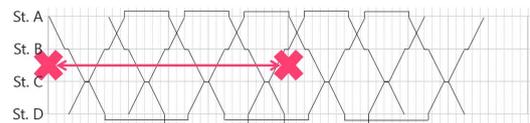


図2: 実験に用いたダイヤ

4.2 実験結果

実験結果を以下の表4に、また得られた運転整理案を図3に示す。

表4: 実験結果

	既存研究	本研究
目的関数 (分)	216	216
実行時間 (秒)	約 55	約 2

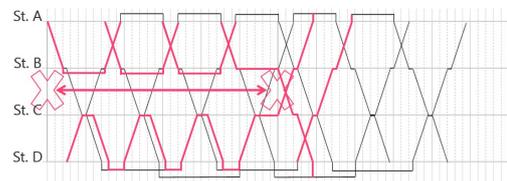


図3: 得られた運転整理案

表4より、本解法でも混合整数計画法と同等の求解が可能であることが分かる。また実行結果においては、制約式が少ない本解法のほうが早く求解できている。また図3から運休と遅延を適切に使い分け、速やかに元々のダイヤへ回復していることが分かる。

5 むすび

本研究では、運転整理問題に対してヒューリスティックを用いた解法を提案した。本解法では、混合整数計画法より早い時間で同等の求解が可能であることが確認できた。既存研究の混合整数計画法と比較し、配列のバイナリ表現のみで運転整理を行っているため、短時間での求解が可能になったと考えられる。また、ヒューリスティック解法により下り列車のみを考慮しているため、支障対象数の半分だけで求解できている点も時間短縮に効果があったと考える。また、遅延や運休だけでなく折り返し運転にも対応しており、より汎用的な解法であるといえる。

今後の課題としては、臨時列車の設定や、多路線との接続を考慮することがあげられる。また、混合整数計画法で求解できなかった駅数や時間を増加させた大規模な問題を作成し、取り組んでいきたい。

参考文献

- [1] S. Kanai, K. Shiina, S. Harada, and N. Tomii, "An optimal delay management algorithm from passengers' viewpoints considering the whole railway network," *Journal of Rail transport Planning & Management*, Vol.1, No.1, pp.25-37, 2011.
- [2] F. Coman, A. D'Ariano, D. Pacciarelli, and M. Pranzo, "A tabu search algorithm for rerouting trains during rail operations," *Transportation Research Part B, Methodological*, Vol.44, No.1, pp.175-192, 2010.
- [3] X. Meng, L. Jia, Y. Qin, "Train Time Optimizing and Rescheduling Based on Improved Particle Swarm Algorithm," *Transportation Research Record*, No.2197, pp.71-79, 2010.
- [4] 千種健二, 佐藤圭介, 古関隆章, "混合整数計画法に基づく列車運行乱れ時の旅行時間増大量に主眼を置いた運転整理最適化", *電気学会論文誌 D*, Vol.132, No.2, pp.170-177, 2012.
- [5] 田村 啓, 佐藤圭介, 富井規雄, "混合整数計画法による旅客損失を最小化する鉄道の運転整理と計算時間短縮式", *電子情報通信学会論文誌 D*, Vol.J97-D, No.3, pp.393-404, 2014.
- [6] 今田京介, 富井規雄, "途中折り返し運転を考慮した混合整数計画法による運転整理アルゴリズム", *電気学会論文誌 D*, Vol.137, No.6, pp.484-491, 2017.
- [7] (財) 鉄道総合技術研究所 運転システム研究室, 鉄道のスケジューリングアルゴリズム - コンピュータで運行計画をつくる, (株) エヌ・ティー・エス, 2005.
- [8] V. Cacchiani, D. Huisman, et al. "An overview of recovery models and algorithms for realtime rail-
way rescheduling," *Transportation Research Part B, Methodological*, Vol.63, pp.15-37, 2014.