



鉄道のダイヤ乱れ時への対応 その2

—現状と研究開発の状況—

富井規雄 (千葉工業大学 情報科学部)

佐藤圭介 ((公財) 鉄道総合技術研究所 信号・情報技術研究部)



前号では、鉄道におけるダイヤ乱れ時の対応について、その内容や難しさなどについて述べさせていただいた。その最後でも述べたように、ダイヤ乱れ時には、指令室の業務は急激に増大する。それを軽減し、また、利用者へのサービスを向上する上で大きな効果があるのは、ダイヤ乱れ時のダイヤ変更計画(運転整理案)作成作業の支援である。本号では、運転整理案を作成するアルゴリズムについて、現状の機能とその問題点、それに対する研究開発の状況を解説する。

運転整理の支援—現状

運転整理の支援機能の開発は、昭和47年(1972年)の新幹線岡山開業時まで遡ることができる。このときに、新幹線に運行管理システム(コムトラックと称する)が導入されたわけであるが、第一に開発すべき機能として、ダイヤ乱れ時に今後の列車運行状況を予測した結果(予測ダイヤ)をグラフィックディスプレイに表示し、指令員の入力を受け付けて運転整理案を作成する機能が掲げられていた。当時から、運転整理の支援に対する要望が強かったことが分かる。ただし、当時のコンピュータ(主記憶16kワード、32kバイト相当!)では、予測ダイヤの作成に処理時間がかかりすぎ、どちらかというオフラインでの運転整理方法の検討に用いられたようである。

その後のハードウェア、ソフトウェアの発達によって、運転整理の支援機能の研究開発も進んできた。その結果、予測ダイヤを作成する機能や、比較的単

純なダイヤの変更について、あらかじめ与えられている条件を満たす状況になったときに、ダイヤの変更を指令員に問いかけ、指令員がYesとしたときにはその変更を自動的に実行する機能などが実用化されている。後者については、たとえば、前号で説明した、列車の待避駅の変更(順序変更)を問いかける機能がある。図-1に示すように、所定のダイヤ(点線)で、駅3で普通列車が特急列車を待避する計画になっているときに、普通列車が遅れたとする(太線)。このとき、駅2での発車順序を変更して駅2で普通列車を待避させるかどうかを問いかける機能がある(逆に、特急列車が遅れたときに普通列車を先発させるかどうかを問いかける機能もある)。別の例としては、互いに交差する線路を走る2本の列車の交差点での優先順位を問いかける機能がある。図-2に示すように駅の中で2本の列車の進路が交差する場合に、どちらの列車を優先するのかを問いかける(図-2では、列車Aの進路と列車Bの進路が点線の○の個所で交差する。両方の列車がほぼ同時に○の場所を通過しようとする衝突しそうだが、もちろん、そんなことはない。手前に信号機があって、一方は赤になっている。優先されなかった方の列車はその手前で待たされる。すなわち、優先されなかった方は遅れることになる)。

しかし、現状においては、これらの機能は必ずしも有効に活用されているとは言いがたい。その理由は、次の通りである。

(1) 1つの駅での2本程度の列車のダイヤ変更を対象とした局所的な判断にとどまっている。しかし、ダイヤ乱れの規模が大きい場合には、こ

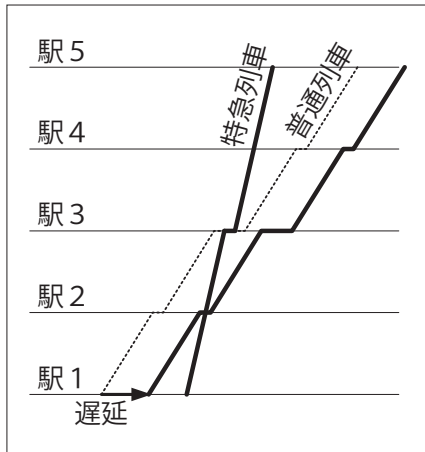


図-1 順序変更の問い合わせ：普通列車が遅れたとき、駅2で特急列車を待避するかどうかを問い合わせる

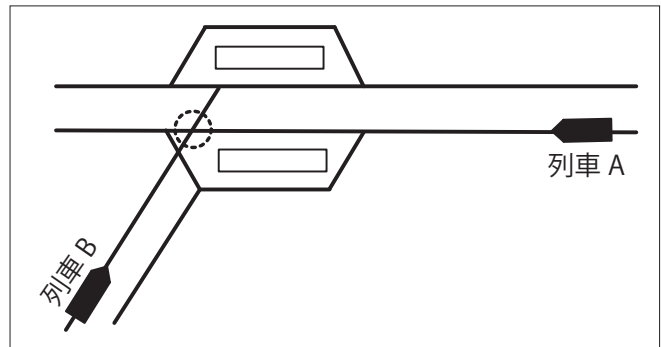


図-2 列車Aと列車B、どちらの列車を優先するかを問い合わせる

の程度の小規模なダイヤ変更の提案では大した労力軽減にはならない。より多数の列車を含む、より広範囲を対象とした運転整理案の作成の支援が望まれる。

(2) 局所的に見てよいと思われても、全体としてよいとは限らない。実際には、その変更がその後の列車運行に与える影響を加味しなければならないためである。したがって、このような近視眼的なアルゴリズムは、通常は破綻する。

(3) 利用者の動きを考慮していない。たくさんの旅客が待っている方向には、混雑を緩和するために、たくさんの列車を運転したい、あるいは列車を優先的に運転したいというニーズがある。しかし、現行の運転整理支援システムでは、そのような事情は考慮されていない。たとえば、図-2が朝ラッシュ時で、列車Bが都心へ向かう列車であれば、列車Aを少し遅らせても列車Bを優先するべきだろう。さもなければ、列車Bが駅の手前に止まることとなり、混雑した車内の乗客に迷惑がかかる。また、それによって列車Bの遅れが後続列車に波及するという問題もある。しかし、このような判断は、現行の運行管理システムには備わっていない。

(1) (2) の根本には、現状では「全体最適」の考えがないことが挙げられる。その理由としては、さまざまな状況を加味した評価関数を確立しにくいこと、および、多数の列車を対象にした複雑な最適

化問題を迅速に解くアルゴリズムの構築が容易ではないという事情がある。

(3) の根本には、現在の運行管理システムが、利用者の数や移動に関する要望（「利用者のデマンド」と言う）をきちんと扱っていないことがある。つまり、今どれくらいの利用者がそれぞれの列車に乗っているか、その人たちの目的地はどこか、あるいは、どれくらいの人が駅で待っているのか、どこに行こうとしているのかなどの情報は運転整理を行う上できわめて重要な情報であるが、現在の運転整理支援機能においては、これらの情報は考慮されていない。その理由は、少なくとも現時点においては、そのような情報をリアルタイムで取得することは非常に困難であるためである。運転整理案の作成支援という点では、コンピュータの自動判断とはせず、問い合わせ形式として、いちいちダイヤ変更の採用の可否を指令員に問い合わせる理由の1つはこれである。つまり、状況判断に必要な情報が不足しているためにコンピュータ単独では確実な判断ができず、それを指令員に委ねているということである。

■ 運転整理案作成アルゴリズム

◆ 運転整理案作成アルゴリズムへの期待

ダイヤ乱れ時には、指令室の業務は急激に増大する。その負荷を軽減する上で効果が大きいのは、運転整理案を自動的に作ることだろう。極端に言えば、

ダイヤ乱れを検知してコンピュータが自動的にダイヤの変更案を作ってくれる、あるいは、事故の規模が大きいときには、事故の復旧見込み時刻だけを入力すれば、ダイヤの変更案が自動的に生成されるというアルゴリズムがあれば、ダイヤ乱れ時の指令室の業務は大幅に軽減されるに違いない。また、運転整理案を迅速に作成して今後の運行計画を早期に確定することができれば、その情報を利用者に提供することができる。そうなれば、利用者は「今後どうなるのか」「どう行動すべきか」などの情報を得ることができる。

以下では、前述の「利用者の視点」「全体最適」をキーワードに、運転整理案の作成アルゴリズムに関するいくつかの研究事例を簡単に紹介させていただきたい。

運転整理案作成問題は、旅客の不満度を目的関数とし、列車運行上のさまざまな条件を制約として、目的関数の値を最小にするようなダイヤ変更計画を出力する組合せ最適化問題だと考えることができる。入力は、所定の列車ダイヤと列車の遅延状況（あるいは、運転再開見込み時刻）である。繰り返しになるが、2～3本の列車を対象とする局所的な変更だけを扱うのではなく、ある程度長い時間（たとえば、遅れがなくなって通常のダイヤで運行できるようになる時間まで、状況によるが数時間程度）を対象にすることが求められる。対象となる列車の数は多く、取り得るダイヤ変更手段の数は多い。非常に大規模で複雑な組合せ最適化問題になる。しかも、それを迅速に解くことが要求される。多数の利用者がその瞬間も駅で待っているからである。

事例 1 — 利用者の不満を少なくするアルゴリズム

文献 1) では、利用者が不満を感じるであろうと思われる「事象」をあらかじめ洗い出しておき、そのような事象がなるべく出現しないような運転整理案を作成するアルゴリズムを提案している。事象と

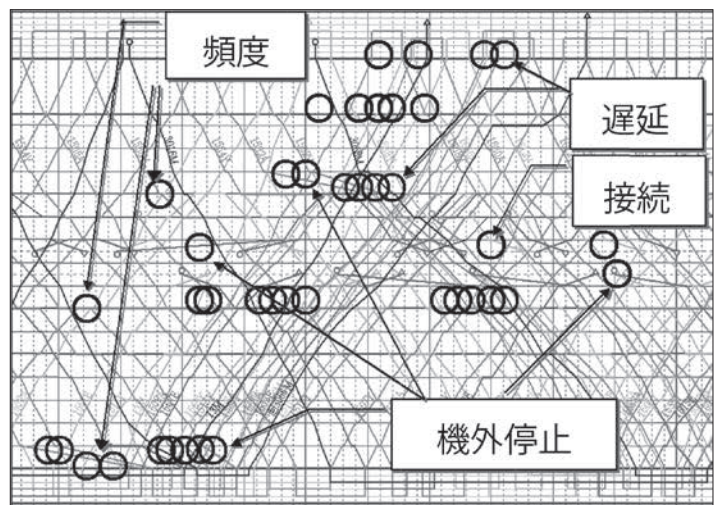


図-3 利用者の不満を最小にするアルゴリズムのイメージ (○が、利用者が不満を感じている個所)

しては、電車の遅れ、前の電車との運転間隔、接続の有無、機外停止などが考慮される。

図-3 にイメージを示す。このダイヤ図は、運転整理を行わないで放置しておいた場合のダイヤ図である。このままであれば、利用者はいろいろな不満を感じるであろう。○が、利用者が不満を感じている（とシステムが判断した）個所を示している。「頻度」「遅延」「接続」「機外停止」は、不満を感じると判断した理由である（煩雑になるので、主な個所についてのみ記載している）。頻度とは、列車の運転頻度が減少したための不満を指し、具体的には、前の列車との間隔があるしきい値を超えた場合に、不満と判断する。遅延は、主要駅において列車の到着の遅延がある値以上となった場合の不満を意味している。接続とは、通常のダイヤでは普通列車と快速列車などの間の接続がとられているにもかかわらず、ダイヤ乱れによって接続がとられなくなってしまった場合の不満を意味している。機外停止とは、駅に到着しようとしている列車が、番線がふさがっているために、駅の手前で待たされてしまうことを言う。具体的には、駅間の走行時分があるしきい値を超えた場合に機外停止が発生したと判断し、不満の件数にカウントすることとしている。

したがって、アルゴリズムとしては、これらの○がなるべく少なくなるような運転整理案を求めるこ

とが目的となる。

一方、どのような事象に不満を感じるかは、事故の規模、路線、方向、駅、時間帯など、その時の状況によって異なるであろう。たとえば、列車が1時間程度不通になるようなかなり大規模なダイヤ乱れの場合には、数分程度の列車の遅れは大きな不満にはならないだろう。しかし、列車の遅れが小さい場合には、数分程度の遅れでも不満になる。このような要求に対応するために、このアルゴリズムでは、それらの状況ごとに利用者が不満と感ずると考えられる事象を定義できるようにしている。また、状況ごとの事象は、経験等に基づいてあらかじめ人間が用意するとしている。最適化手法としては、メタヒューリスティクス^{☆1}の1つであるシミュレーテッド・アニーリングを用いている。

このアプローチでは、コンピュータには推定の難しい「利用者が不満を感じる事象」の定義を人間にまかせ、コンピュータは最適化に特化しようとしている。しかし、利用者が不満を感じる事象をさまざまな状況に応じてあらかじめ準備しておくことは容易ではなく、また労力を要するという点で課題を残している。

事例2 —利用者の行動シミュレーションと最適化を組み合わせたアルゴリズム

文献2)では、運転整理の一種である接続判断問題を対象として、利用者の視点から最適な接続判断を行うアルゴリズムを提案している。接続が考慮されているダイヤでは、ある電車が遅れた場合、その接続を確保するか、あるいは、接続をあきらめて電車を発車させるかを判断しなければならない。そして、その判断は、その後の列車に与える影響を考慮して行わなければならない。つまり、遅れている列車と接続をとると、接続をとった列車も遅れてしまう(それによって、以降の駅でその列車と別の列車との接続が問題になることもあり得る)、しかし、

^{☆1} 問題個別ではなく、汎用的に適用可能な発見的な最適化手法。生物や物理現象に観察される最適化のプロセスをアルゴリズム化している。ただし、最適解が求まるという保証はない。シミュレーテッド・アニーリングは、「焼きなまし」にヒントを得た手法である。

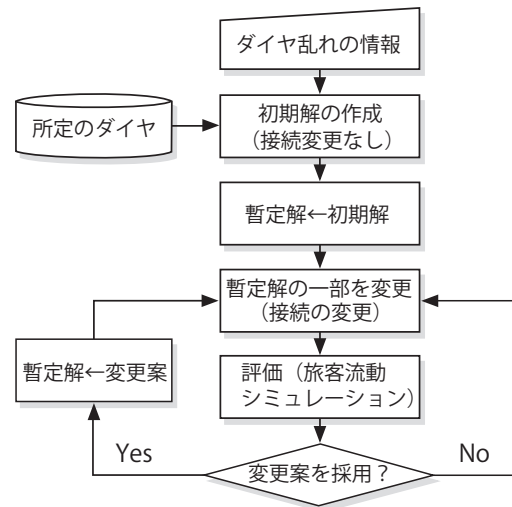


図-4 メタヒューリスティクスに基づく接続判断アルゴリズムの全体構成—改善と評価を繰り返す

接続をとらないとすれば、接続列車に乗ろうとしていた乗客が不便になってしまう、といった状況を考慮しなければならない。言い換えると、接続判断は、鉄道ネットワーク全体の列車と利用者を対象として行わなければならない。そして、その際に利用者がこうむる不便さを考慮しなければならない。文献2)では、利用者の視点からの評価値としての「利用者がこうむる不便さ」を算出するために、旅客流動シミュレーションを用いている。これは、利用者の動き(どの駅をいつ出発してどの列車に乗り、どこで乗り換えたかなど)を、一人ひとりについてシミュレーションによって推定し、その利用者がどの程度の「不便」をこうむったかを算出しようとするものである。不便としては、待ち時間や目的地までの所要時分の増加、乗換、車内の混雑などを考慮する。そして、改善(接続の変更)と旅客流動シミュレーションによる評価を繰り返し、全体として最適な接続変更案を得ようとしている(図-4に全体構成を示す)。最適化手法としては、シミュレーテッド・アニーリングを用いている。

このアルゴリズムは、メタヒューリスティクスの一般的な枠組みに基づいている。すなわち、1カ所の接続の変更と旅客流動シミュレーションによるその変更案の評価を繰り返して、全体として最適な接

統変更案を求めようとする。したがって、最適化の過程で旅客流動シミュレーションが何度も実行されることになる。そのため、特に列車本数や乗客数の多い路線では、この部分に時間を要するという問題がある。路線にもよるが、数万人以上の規模の旅客流動シミュレーションを何度も繰り返すことになる。しかし、解が出るまでの時間は長くても数分以内であることが求められる。このアルゴリズムでは、処理時間の多くの部分は、旅客流動シミュレーションに費やされるため、特に旅客流動シミュレーション部分の処理時間の短縮が課題となる。現時点で、大都市近郊の5路線190本の線区を対象として、3～4分程度で実用的な解を生成可能と報告されているが、実用性を高めるためには、さらなる高速化が望まれる。また、そもそも、そのときの利用者の数や行き先に関する正確な情報をどのようにして取得するのかという点も課題となる（文献2）では、自動改札機データ等から得られる平常時の情報を使用している）。

さらに、この手法を一般の運転整理案作成問題に拡張することも可能である。ただし、小規模のダイヤの変更を繰り返していく方法は、中規模以下のダイヤ乱れには向いているものの、数多くのダイヤ変更が必要となる大規模なダイヤ乱れ時への適用は、特に処理時間の点で適していないと考えられる。

事例3 厳密に最適な運転整理案を得る アルゴリズム

一方、近年になって、混合整数計画法を運転整理アルゴリズムに適用しようとする動きが、特にヨーロッパにおいて盛んである。混合整数計画法には、メタヒューリスティクスとは異なり、(ほぼ) 厳密に最適な解が得られるという特徴がある。運転整理においては、評価関数の定義にややあいまいさがあることや、迅速性が求められること、状況の変化があり得ることなどから、厳密な最適性が必須というわけではない。しかし、メタヒューリスティクスでは、最適解が得られるという保証がないだけでなく、そもそも得られた解がどれくらい最適解に近い

かすら分からない。したがって、最もよい解であるとして得られた解よりもずっとよい解が存在する可能性があり(しかも、そのような解が存在するのかどうかも分からない)、その点での不安が残る。一方、混合整数計画法には、得られた解の最適性に安心感があるというメリットがある。

混合整数計画法では、列車運行の制約を数式で表現することになる。利用者の視点を目的関数とするためには、利用者の行動も数式で表現しなければならず、難しさが増す。また、処理時間の点でもメタヒューリスティクスよりも時間を要するため、その点の工夫も必要である(本稿コラムを参照いただきたい)。

事例4 一定型的な「パターン」を利用する アルゴリズム

一方、これらとはまったく異なったアプローチとして、「パターン」を用いて運転整理案を作ろうとするアプローチがある³⁾。現実の指令室では、事故が発生して不通となった区間ごとに、とるべき運転整理手法をまとめたマニュアル(どの列車を運休にして、どこで折り返し運転を行うかなど)が準備され、それに沿った運転整理が行われることが多い。マニュアルに記載された運転整理のパターン(以下、運転整理パターン)に沿った運転整理を行うのは、迅速な対応を可能とすること、駅、車両基地・乗務員基地、乗務員などの関係者が運転整理の方針をあらかじめ共有して、その後の対応を容易にすることなどのメリットを考えてのことである。

運転整理パターンの例を図-5に示す。この路線では、系統1と系統2の2つの種類の列車が走っており、系統2の列車はD駅で分岐して支線に直通する(なお、実際には、中間にもっと多くの駅があるのだが、煩雑になるため記載していない)。今、D駅とE駅の間で事故が起こり、復旧までにそれなりの時間(たとえば、1時間程度)が見込まれるとする。深く考えなければ、不通になっている区間を通る系統1の列車を始発駅のA駅から運休することになる。しかし、そのようにすると、A駅から

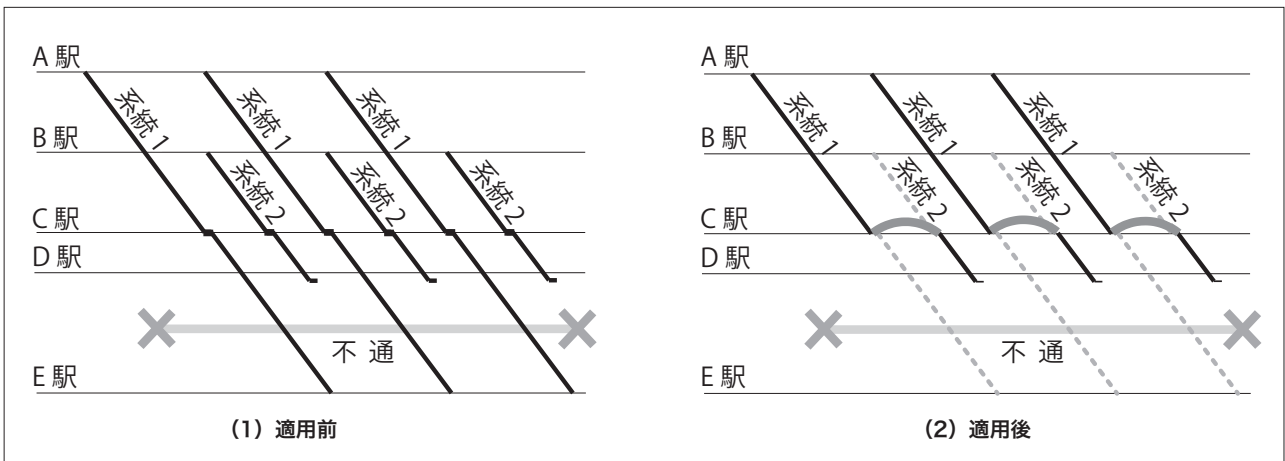


図-5 「パターン」を適用した運転整理案の作成

は列車がなくなる。実は、C 駅はこの路線の中心駅であって、A 駅（および A 駅と B 駅の間の駅）から C 駅に行く乗客が多い。したがって、A 駅からの列車がまったくなくなってしまうと、そういう利用者にとって非常に不便になる。そこで、図-5 (2) のような運転整理パターンを適用する。この運転整理パターンでは、系統 1 の列車の C 駅から E 駅間と、系統 2 の列車の B 駅から C 駅間を運休とする。そして、C 駅で車両運用の変更を行って、系統 1 の列車を D 駅から支線に行かせるようにする。こうすれば、A 駅や A 駅と B 駅の間の駅の利用者も C 駅や D 駅に行けることになる。さらに、裏の事情として、B 駅には車両基地が併設されているということがある。列車を運休すると、その列車の編成をどこに置いておくかが問題になる。駅に置いておく番線をふさいでしまうためにほかの列車を運転できないからである。その点、車両基地がある駅から列車を運休にすれば、その編成を車両基地に収容することができる。この運転整理パターンは、利用者の利便さはもちろんであるが、そういう事情も考慮して作られている。

文献 3) では、運転整理パターンをそのままコンピュータに入れてしまい、それにそって運転整理案を作ろうとするアルゴリズムを提案している。これは、大規模なダイヤ乱れ時にも対応できることを念頭においている。

実は、事例 1～3 は、中規模のダイヤ乱れを対

象としたアルゴリズムであった（列車密度等にもよるので、大・中・小の区別を厳密に定義することは難しいが、中規模とは、おおむね 15 分からせいぜい 30 分程度の遅れを想定している。詳細は、文献 4) 参照）。ダイヤ乱れが大きくなれば、利用者の数や動きが平常時とは異なるであろう。振替輸送が行われることもある。事例 1～3 は、いずれも利用者のデマンドを入力としているが、現実には大規模なダイヤ乱れ時の利用者のデマンドをリアルタイムで精度よく予測することは非常に難しい。また、それに基づいて、「運転整理の方針」をたてなければならない。区間折り返し運転など、大幅なダイヤの変更が必要になり、それに対応するアルゴリズムが必要になる。さらに、前述したように車両基地が併設されている駅からの運休はしやすいといった事情や、それに加えて過去の失敗体験など、なかなかアルゴリズムに明示的に反映しにくい事情も現実には存在し、それらを考慮した運転整理案としたいという要望もある。しかし、これらは、いずれも現状のコンピュータには骨の折れる仕事である（歯が立たないと言うべきだろう）。

このような事情から、大きなダイヤ乱れの場合には、とりあえず最適化にはこだわらず、とにかく迅速に実行可能な運転整理案を作って、利用者に早めにアナウンスして混乱を防ごうという考え方が生まれてくる。文献 3) のアルゴリズムは、そのような考え方に基づいて作られている。しかし、言うまで

【混合整数計画法による運転整理】



混合整数計画法とは、対象とする問題の制約条件と評価関数（目的関数と呼ばれる）を、実数値や整数値をとる変数を用いてモデル化し、制約条件を満足しつつ目的関数が最小ないし最大になる解を求める手法である。これを運転整理に当てはめる場合、列車の時刻や順序などの決定すべき事柄を変数とし、物理的な制約、リソースの制約を不等式で表現することになる。たとえば、列車のある駅から次の駅までの所要時間制約は、列車 r の駅 s の発車時刻を実数変数 d_r^s 、次駅 s' の到着時刻を実数変数 $a_{r'}^s$ 、所要時間を定数 I_{Run} とおくことで、以下ようになる。

$$a_{r'}^s - d_r^s \geq I_{Run}.$$

別の例として、駅 s で 2 本の列車 r_1, r_2 が続いて発車する場合に確保しなければならない時間の間隔 I_{DD} を考える。ここで、 r_1 と r_2 の順序変更が可能と想定すると、 r_1 が r_2 に先発するならば 1、そうでなければ 0 とするバイナリ変数 x_{r_1, r_2}^s を導入することになる。あわせて、任意の大きな正の定数 M を定義すると、発車間隔の制約の 1 つは、次のように表される。

$$d_{r_2}^s - d_{r_1}^s \geq I_{DD} - M(1 - x_{r_1, r_2}^s).$$

この式は、 r_1 が先発する ($x_{r_1, r_2}^s = 1$) ならば、 r_2 の発車時刻は r_1 の発車から I_{DD} ほど間隔を空けた時刻以降である ($d_{r_2}^s \geq d_{r_1}^s + I_{DD}$) ことを示す。先発しない ($x_{r_1, r_2}^s = 0$) ならば、右辺が小さな負の値になるため、不等式は意味を持たなくなる。これだけでは r_2 が先発するケースを網羅していないので、 r_2 が r_1 に先発するか否かのバイナリ変数 x_{r_2, r_1}^s を用いた

$$d_{r_1}^s - d_{r_2}^s \geq I_{DD} - M(1 - x_{r_2, r_1}^s)$$

という不等式と、どちらか一方の列車が必ず先発することを意味する、以下の式も必要になる。

$$x_{r_1, r_2}^s + x_{r_2, r_1}^s = 1.$$

運転整理の混合整数計画モデルのうち、列車運行の変数と制約条件は、国内外の研究でおおむねこのような表現となっている。

利用者のことを考えた運転整理にするために、モデルに利用者行動の変数を追加し、乗車する列車の選択ルールを制約条件と捉えて不等式で表すことで、利用者の不便を定量的に計れるようにすることが考えられる¹⁾。列車選択ルールを制約と捉える例には、利用者がある駅で乗車する列車は 1 つということがある。ほかにも、出発駅のホームに出現する時刻以前に発車する列車には乗車できないということがあ

もなく、このアプローチは、利用者の利便性については、最適性はおろか利用者のデマンドをはじめとするそのときの状況がまったく加味されていないという点で課題を残しており、さらなる発展が望まれる。

今後の発展

運転整理は難しい。筆者は、長年運転整理アルゴリズムの研究開発に従事しているが、追い求める理想的な運転整理には遠く及ばない。そもそも自分が求めるような運転整理が実現可能かすら定かではない。海外の友人には、おまえはまるで Holly Grail (聖杯) を求めるインディ・ジョーンズだなどと評されたことがある。

しかし、コラムにもあるように、アルゴリズムの研究は進んでいる。これまで解けなかった問題が解けるようになってきている。本稿では紹介できなかったが、その線区で日常的に行われる運転整理の内

容を分析した結果を反映して、途中駅での順序変更と終端駅での番線変更に特化したアルゴリズムなども実用的に用いられる段階にきている。

ただし、運転整理が難しい理由としては、特に大きなダイヤ乱れの場合、そもそもの前提条件を決めることが非常に難しいこと（たとえば、前号の「運転整理の方針」など）が挙げられる。また、最適な運転整理案が得られたとしても、復旧見込み時刻の変更など予期せぬ事態が発生して状況が変わるとそれが適用できなくなることがあり、その場合には、せっかく作って関係者に伝達した運転整理案を取り消し、再度作り直すことが必要になるという難しさもある。

そういう事情を考えると、今後は、アルゴリズムの設計において、人間がどの程度・どのように介入すべきかを再考することがまず必要である。また、現状では取得できていないデータ（乗客の動き、電車の混雑率、予備車両・予備乗務員の有無等）を取得して、それを反映した運転整理案を作成するアル

る。この状況は、利用者の出発駅と目的駅の組を (o, e) 、出現時刻を定数 t として、利用者が列車 r に乗車するならば 1、そうでなければ 0 とするバイナリ変数 $z_{t,r}^{(o,e)}$ を導入すると、次の式で表現できる。

$$z_{t,r}^{(o,e)} \leq d_r^o / t.$$

これは、列車の発車時刻が利用者の出現時刻以前 ($d_r^o < t$) ならば、右辺は 1 未満となり、左辺の変数値は 0、つまりその列車に乗車しないことを示す。

目的関数は、列車の遅れよりも、利用者の目的駅到着時刻の遅れとするのが適切である。出現時刻が t である (o, e) 間の利用者数を定数 $P_t^{(o,e)}$ 、それら利用者の平常時の到着時刻を定数 $A_t^{(o,e)}$ とおくと、利用者の到着時刻の遅れ総和最小化は、次のようになる。

$$\text{最小化} \sum_{(o,e)} \sum_t P_t^{(o,e)} \times \left(\sum_r z_{t,r}^{(o,e)} \times a_r^e - A_t^{(o,e)} \right).$$

これは、利用者の乗車列車 ($z_{t,r}^{(o,e)} = 1$) の目的駅到着時刻と平常時の到着時刻との差分 ($a_r^e - A_t^{(o,e)}$) を、全利用者について足し合わせたものである。

混合整数計画モデルの解は最適化ソルバーにより求められるが、上記の目的関数は 2 変数の積を含む非凸関数であり、未対応のソルバーに適用するには式変形が必要になる。また、バイナリ変数が多く計算時間がかかるため、モデル化の洗練や運転整理の構造に着目した解法の開発などが研究課題

である。

海外の研究では、鉄道ネットワークにおける列車の接続判定を主眼にして、利用者の到着時刻の遅れ最小化が議論されてきた²⁾。現在でも、列車運行や利用者行動の制約を取り込んで発展が続いている。

翌日、列車をダイヤ通りに運転するために必要な編成を始発駅に確保するために、折り返しの変更や回送列車の運転などを計画する業務を「運用整理」と呼ぶ（乗務員についても同様）。運用整理に（混合）整数計画法を利用することは、次号で紹介する平常時の運用計画作成が、古くから集合分割問題や集合被覆問題として捉えられていたことから、運転整理よりも盛んである。運用計画作成の制約に加えて、個々の車両や乗務員の整理開始時点での所在位置を考慮した、追加制約付き集合分割（被覆）問題とする研究が多い³⁾。

参考文献

- 1) Tamura, K., Tomii, N. and Sato, K. : An Optimal Rescheduling Algorithm from Passengers' Viewpoint based on Mixed Integer Programming Formulation, Proc. RailCopenhagen 2013 (2013).
- 2) Schöbel, A. : Optimization in Public Transportation, Springer, New York (2006).
- 3) Sato, K. and Fukumura, N. : Real-time Freight Locomotive Rescheduling and Uncovered Train Detection During Disruption, European Journal of O. R., Vol.221, No.3, pp.636-648 (2012).

ゴリズムを確立することも重要である。現在では、自動改札や IC カードのデータ、車両ごとの混雑率、経路案内システムの検索履歴など、さまざまなデータが取得できるようになっている。これらのデータを有効に活用することを考えるべきだろう。さらに、予期せぬ事態の発生に頑健な運転整理案—たとえば、運転再開見込み時刻の修正など、予期せぬ事態が発生しても再度の変更が少なく済むような運転整理案など—を作成するアルゴリズムを考案するなどの点での研究開発も必要である。

そして、このような研究開発の成果に基づいて、中規模以下の乱れの場合には、ほぼ自動でダイヤの変更が行えるアルゴリズムを確立する、また、大規模のダイヤ乱れの場合には、人間主導とはしながらも、特に運転整理案の作成や評価についてコンピュータアルゴリズムをフルに活用した支援機能を早期に実現していくことが望まれる。

また、そもそも、運転整理以前の課題として、予期せぬ事態がなるべく発生しないようにするための

さまざまな工夫や努力、また、運転整理がしやすいダイヤとすることなども重要である。それによって、利用者の不満や怒りを最小限にとどめ、かつ、利用者の納得の得られる運転整理を実現できるように努めていく必要がある。

参考文献

- 1) 鉄道総合技術研究所：鉄道のスケジューリングアルゴリズム、NTS 出版 (2005)。
- 2) Kanai, S., et al. : An Optimal Delay Management Algorithm from Passengers' Viewpoints Considering Whole Railway Network, Journal of Rail Transport Planning & Management, Vol.1, Issue 1 (2011)。
- 3) 平井 力 他：運転整理パターン記述言語 R による列車運転整理案作成アルゴリズム、情報科学技術レターズ (FIT2005), Vol.4 (2005)。
- 4) 富井規雄 編著：鉄道ダイヤ回復の技術、オーム社 (2011)。

(2013 年 8 月 14 日受付)

富井 規雄 (正会員) | tomii@cs.it-chiba.ac.jp

国鉄、(財) 鉄道総合技術研究所を経て、千葉工業大学情報科学部教授。運輸安全委員会委員 (非常勤)。著書として、「鉄道ダイヤ回復の技術」、「鉄道ダイヤのつくりかた」など。京都大学博士 (情報学)。

佐藤 圭介 | sato.keisuke.49@rtri.or.jp

(公財) 鉄道総合技術研究所 信号・情報技術研究部副主任研究員。鉄道の運転整理・運用整理の最適化の研究開発に従事。