特集：輸送・交通計画技術

実績ダイヤを用いた運転整理における指令手配のノウハウ抽出手法

田中　峻一*　　坂口　　隆*　　加藤　　怜*　　瀧本　友晴*

Method for Extracting Knowledge of Train Rescheduling from Data of Operation Records

Shunichi TANAKA　　Takashi SAKAGUCHI　　Satoshi KATO　　Tomoharu TAKIMOTO

Train rescheduling is carried out for a delay recovery when train schedules are disrupted. Computational generation methods of the train rescheduling have been developed. However, it is an issue to be solved to incorporate the knowledge of experienced dispatchers in them. In our previous research, we proposed a method to extract knowhow as a set of association rules from past dispatcher decision data and to reflect it to train rescheduling algorithms using mathematical optimization. This paper presents the method and verification results of efficiency improvement on the extracting rules and of routinely update the rules.

キーワード：運転整理，数理最適化，運転整理ルール，相関ルール，PrefixSpan 法

1．はじめに

事故や気象災害等で列車のダイヤが乱れた際には，回復を図るための運転整理が実施される。運転整理は，ダイヤ乱れの発生状況や線区の特性などを把握したうえで，列車ダイヤをどのように変更するかを的確に判断するノウハウが必要となる。運転整理案の自動作成については，総遅延時分や旅客の不効用値などの単一の定量的評価指標に従って解を求める手法が提案されている[1)]。しかし，そのような単純な指標のみでは指令員が培ってきたノウハウや線区ごとの方針や特情（以下「ノウハウ等」という）を反映することが難しい。

そこで，鉄道総研では，これまでに運転整理のノウハウ等を過去の実績ダイヤから運転整理ルールとして抽出する手法と，抽出された運転整理ルールを用いて運転整理案を作成する方法を提案した[2)]。しかし，実用規模の線区への適用においては，同義の運転整理ルールが出現すること，運転整理ルールの抽出に処理時間がかかりすぎる等の課題があった。また，日常的に実績ダイヤが蓄積される中で運転整理ルールを更新する手法も必要である。本稿では，それらの課題に対する解決方法及び都市圏の線区における検証結果について報告する。

2．先行研究における手法と課題

2. 1　運転整理ルールを用いた整理案作成手法の概要

運転整理ルールを用いた運転整理案作成手法の全体像を図 1 示す。本手法では，指令員のノウハウ等を反映するために前準備としてノウハウ等をルールの形の形式知として抽出して蓄積する部分と，蓄積されたルールを運転整理アルゴリズム（例えば，混合整数計画法によるアルゴリズム）に制約条件として組み込んで運転整理案を作成する部分で構成される。

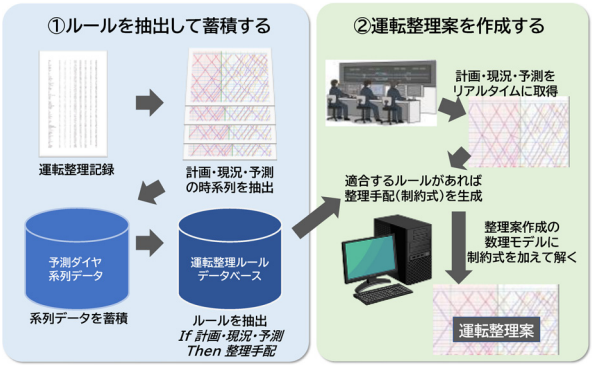


図 1　運転整理案作成へのノウハウの反映

2. 1. 1　運転整理ルールの抽出

運転整理において実施される個々の計画変更を「整理手配」と呼ぶ。運転整理ルールの抽出では，まず実績ダイヤを当初の計画に基づき系列順に比較し，矛盾が発生しないように実施された整理手配の推定を行う。その後，個々の整理手配を実施する前後の計画，遅延状況，その時点での運行予測の時系列を再現して，系列データとして蓄積する。次に，適用件数が一定数以上存在し，蓄積した系列データから，適用率の高いルール（相関ルール）を「運転整理ルール」として抽出する。抽出手法の詳細は 2.3 節にて述べる。

＊　信号・情報技術研究部　運転システム研究室

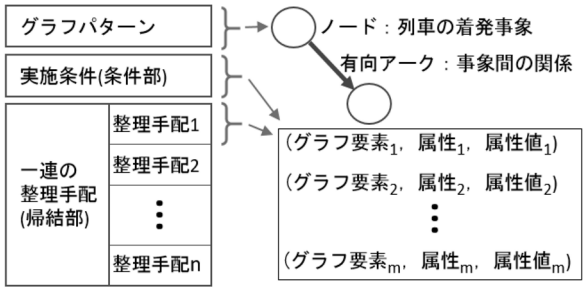


図２　運転整理ルールのデータ構成

2. 1. 2　運転整理案の作成

運行管理システムから，リアルタイムに現在の計画，運行状況を取得し，それに基づく運行予測を含めたダイヤ状況から，条件部が適合する運転整理ルールが存在する場合は，適合した場所（列車，駅，対象となる発着事象）に対して，帰結部を満たすように制約条件を生成する。次に，総遅延時分等を指標とした運転整理アルゴリズムに対して，制約条件を追加して解くことによって運転整理ルールが反映された運転整理案を生成する。

2. 2　運転整理ルールのデータ構成

運転整理ルールは，図 2 に示すように，グラフ構造を持つグラフパターン，実施条件を表す条件部，一連の整理手配を表す帰結部の 3 つの部分で構成される。以下に各部の構成を示す。

2. 2. 1　グラフパターン

実施条件と整理手配が対象とする各事象の相互関係を有向グラフで表す。ノードは，列車・駅での着発事象（通過を含む）を表し，アークはノード間の順序関係を表す。ダイヤ上に出現する一定ノード数以下で構成される部分グラフのそれぞれの形状を，グラフパターンと呼ぶ。各運転整理ルールは一つのグラフパターンに対して規定される。

2. 2. 2　実施条件（条件部）

グラフパターンの要素である個々のノードやアーク，それらに付属する属性，およびその属性値の組合せを一つの条件として，複数の条件の組合せによって整理手配を実施するトリガーとなる条件を表す。ノードの属性には，線区，駅，使用番線，着発識別，運休識別，遅延状態などが含まれる。アークの属性には，アークの種類やアークの発ノードから着ノードへの遅延伝播の有無などが含まれる。アークの種類は，同一の列車または車両間の順序関係であるか，異なる列車または車両間の順序関係であるかの識別を表す。

2. 2. 3　一連の整理手配（帰結部）

整理手配の順列により構成される。個々の整理手配のデータは，実施条件と同じ構成によって整理手配後のダイヤが満たすべき条件を表す。なお，順序変更や運用変更など，グラフ構造そのものが変化する整理手配がありえる。グラフパターンは一連の整理手配が実施される前の事象の順序関係を表し，順序変更や運用変更では整理手配後の順序関係（発ノード，アークの種類，着ノード）を記述することにより整理手配の内容を表す。

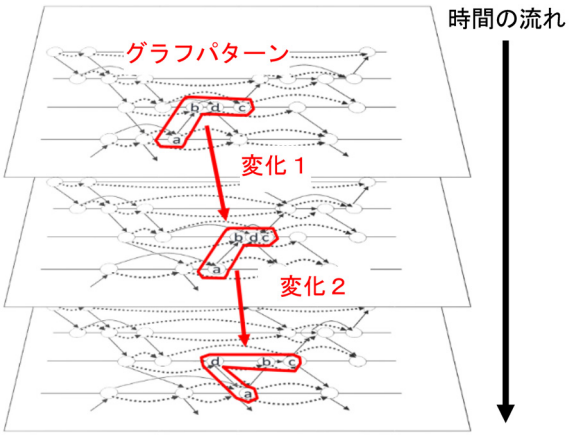


図３　グラフパターンによる系列データの切出し

2. 3　運転整理ルール抽出アルゴリズム

2. 3. 1　系列データと相関ルール

図 3 に示すように，過去の実績ダイヤから手配が実施されるたびに変化する予測ダイヤ（各時刻におけるそれまでの運行実績と今後予測される列車ダイヤ状況）に着目する。各時点における予測ダイヤを表すネットワーク上で，一定数のノードとアークで構成される小規模なグラフパターンで切り取った部分グラフの時系列変化を捉え，その時系列データを 1 件のデータ単位とする系列データ集合を作成する。系列データ集合の中に存在する系列 $x=(x_1, x_2,\cdots, x_n)$ の件数を，x の支持度といい，$\sup(x)$ と表記する。条件部を x_1，帰結部を $x_2,\cdots, x_n$ とする系列 $x=(x_1, x_2,\cdots, x_n)$ が運転整理ルールであるための成立要件は，同じグラフパターンを持つ系列データ集合において，次の 2 つの条件を満たすことである。

$$\sup(x) \geq \min\sup \tag{1}$$

$$\mathrm{conf}(x) = \frac{\sup(x)}{\sup(x_1)} \geq \min\mathrm{conf} \tag{2}$$

式 (1) は運転整理ルールの支持度が基準値である最小支持度（minsup）以上であることを表す。条件部 x_1 が出現する系列データのうち，条件部と帰結部の系列全体 x が出現する系列データの割合を系列 x の信頼度といい，$\mathrm{conf}(x)$ と表記する。式 (2) は系列 x の信頼度が基準値である最小信頼度（minconf）以上であることを表す。これらの条件を満たすルールは相関ルール[3)]と呼ばれる。

2. 3. 2　系列データ集合からの運転整理ルール抽出法

運転整理ルールを抽出する手法として，以下の手順に

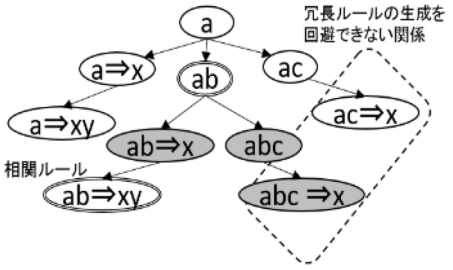


図４　PrefixSpan 法における探索木

よる運転整理ルール抽出アルゴリズムを構築した。

（1）与えられた最大ノード数 N 以下のノード数で構成される各グラフパターン p に対し，グラフパターン p を持つ系列データを検索し，それらの系列データから以下の手順によりグラフパターン p を持つ運転整理ルールを抽出する。

（2）系列パターンマイニング手法の 1 つである PrefixSpan 法[4] により，最小支持度の基準を満たす条件部を深さ優先で探索する。以下では，PrefixSpan 法により系列データから条件部に追加する要素を順次深さ優先で探索することを「条件部拡張」，帰結部に対する同様の探索を「帰結部拡張」と呼ぶ。

（3）（2）で条件部が検索される都度，同条件部（現時点では帰結部は空）に対する帰結部拡張を行う。このとき，得られた系列 $x \Rightarrow y$（x を条件部，y を帰結部とする系列を表す）が相関ルールの基準を満たすことを条件とする。

（4）（3）において基準を満たす系列 $x \Rightarrow y$ が 1 件得られたら，これをルール候補として一旦保留し，帰結部拡張を継続する。拡張先で（帰結部が拡張された）運転整理ルールが出力されたら，ルール候補は冗長であるため破棄する。なお，ルール A に対してルール B が冗長であるとは，ルール A の条件が成立すれば必ずルール B の条件も成立し，かつ，ルール A の帰結部に，ルール B の帰結部が含まれることを意味する（以下，このような関係にある運転整理ルールを「冗長ルール」という）。拡張先でルールが得られなければ，ルール候補を運転整理ルールとして出力する。

（5）（3）で条件部に続く帰結部拡張の結果，運転整理ルールが得られなかった場合，（2）の条件部拡張を再開する。帰結部拡張の結果，運転整理ルールが出力された場合は，冗長性を回避するため，以降の条件部拡張は行わずに探索経路を遡り，他の条件部拡張に移る。

2. 4　先行研究における課題

先行研究では運転整理ルールを抽出する場面において，以下の課題が存在し，実用規模の路線に適用するには不十分であった。

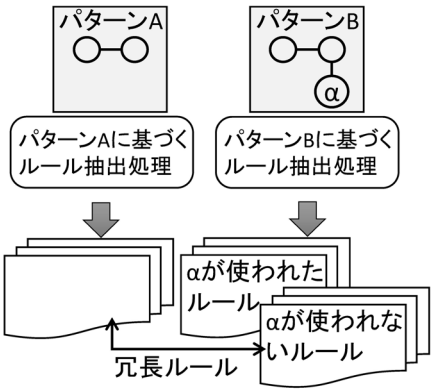


図５　異なるグラフパターンにおける冗長ルール

・2.3 節のアルゴリズムにおける（4），（5）の冗長ルールの対策を実施してもなお，冗長ルールが多数発生する。
・比較的本数の少ない線区においても，運転整理ルールの抽出に長い計算時間がかかる。
・抽出された運転整理ルールを日々の実績の蓄積に応じて経年変化するような更新手法が存在しない。

3．冗長ルールの対策

運転整理案を提案する上では冗長ルールは必ず削除しなくてもよいが，運転整理案作成以外に指令員の支援において運転整理ルールを活用する場面を考えると，排除できることが望ましい。本章では，冗長ルールを回避する手法について述べる。

3. 1　同一グラフパターン内での包含関係の対策

同一グラフパターンでは 2.3.2 項のアルゴリズムにおける（4），（5）にて冗長ルールの対策を行ったが，探索木上で先祖と子孫の関係にある運転整理ルールの包含関係をチェックしているにすぎない。図 4 に PrefixSpan 法における探索木の例を示す。$ab \Rightarrow xy$ が相関ルールとして検出されたとき，帰結部拡張での冗長性回避によって $ab \Rightarrow x$ は出力されない。また，条件部拡張の冗長性回避より，条件部 abc 以降の探索木は探索されない。

しかし，図 4 の点線で囲った $abc \Rightarrow x$ は，$ac \Rightarrow x$ に対して冗長であるが，このような関係による冗長ルールの生成は回避できない。

このような関係にある場合には，各グラフパターンに対する運転整理ルールの抽出終了後に，すべての運転整理ルールから冗長な運転整理ルールを調べて削除することで対策可能である。

3. 2　異なるグラフパターン間での包含関係の対策

図 5 に示すとおり，グラフパターンの中に運転整理ルールに関与しないノードが含まれた結果，異なるグラ

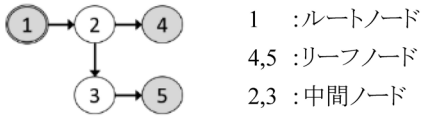


図６　ルートノードとリーフノード

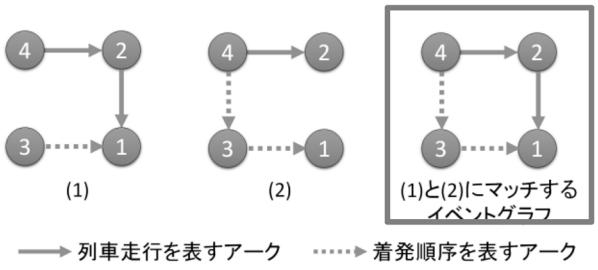


図７　同一事象を指す可能性のあるイベントグラフ

フパターン間で包含関係にある運転整理ルールや全く同一の運転整理ルールが抽出される場合がある。

このような場合，各グラフパターンについて，図６に示すようにグラフのルートノード，リーフノードとそれ以外のノード（中間ノード）に識別することで対策が可能である。ルートノードは運転整理ルールの属性（線区，駅など）を定義し，運転整理ルールの適用箇所を定める上で必要なノードである。一方，中間ノードは運転整理ルールで使用される離れた２個のノードの位置関係を表すために使用されるノードと考えられるので，ルートノード以外の全てのリーフノードが運転整理ルールの中で使用されていなければならないという制約を設けることで，冗長ルールを除くことができる。

3.3　グラフ構造から判別できない冗長ルール

異なるグラフパターン間の冗長ルールにおいて，グラフ構造としては包含関係に無いにもかかわらず冗長ルールとなるものが存在する。例えば，図７の（1）と（2）のグラフパターンはグラフ構造としての包含関係はないが，枠で囲った部分グラフには，（1）と（2）のどちらにもマッチする。従って，（1）のグラフパターンから生成された運転整理ルールと等価な運転整理ルールは，（2）のグラフパターンからも生成される。しかし，それらが同じ運転整理ルールを表していることの判別はグラフ構造や探索アルゴリズムから論理的に導くのは非常に困難である。

3.4　冗長ルールの対策による効果

単線を含む20駅程度の線区において過去に２日間，延べ72回の整理手配（対象とした手配は番線変更，単線使用順序変更）が実施された事例について，運転整理時の記録を元に予測ダイヤの時系列を再現することにより系列データ集合を生成し，冗長ルール対策の有無による抽出される運転整理ルール数の変化を確認した。実験パラメータは，N=3,4，minsup=2，minconf=1.0とした。実行結果を表１に示す。実際の整理手配数より運転整理ルールの数が多いのは，一つの整理手配系列に対し複数の実施条件が存在するためである。この結果より，対策前に抽出されたルールの多くが冗長ルールであることがわかり，冗長ルールの削除が機能していることを確認した。

表１　冗長ルール対策による効果検証結果

	抽出されたルール数（件）		減少率
	対策前	対策後	
最大ノード数：3	6,550	138	97.9%
最大ノード数：4	954,361	667	99.3%

４．運転整理ルール抽出アルゴリズムの効率化

２章に示した運転整理ルール抽出アルゴリズムでは，運転整理ルール抽出にかかる計算時間が莫大になることが確認され，大規模線区や大きな乱れが発生したときには不向きであることがわかっている。

本章では，PrefixSpan法を運転整理ルールの抽出に用いるうえで，探索を効率的に行う方法について述べる。

4.1　条件部拡張時の帰結部検索条件

2.3.2項のアルゴリズムにおける（2）において，条件部xの条件部拡張で得られた条件部（x^*とする）の信頼度の条件と包含関係から，次の定理を導ける。

「x^*の支持度が“xのルール候補の最大支持度を最小信頼度で割った値”（Rとする）より大きい場合は，x^*を条件部とする運転整理ルールは存在しない」

そこで，xの帰結部拡張において上記のRを求めておき，xの条件部拡張では，探索された条件部x^*に対し，上記の定理により運転整理ルールが存在しないと判定された場合は帰結部拡張を省略することで探索を高速化できる。

4.2　帰結部拡張時の条件部拡張再開条件

2.3.2項のアルゴリズムの（4）において，帰結部拡張時に記憶したルール候補の最大支持度の条件から，次の定理を導ける。

「条件部xの帰結部拡張の結果得られたルール候補の最大支持度が最小支持度より小さければ，それに続くxの条件部拡張には運転整理ルールが存在しない」

そのため，帰結部拡張を行った際に上記定理の条件を満たす場合は，条件部拡張を再開せず中断することで探索範囲を縮小することができ，高速化可能である。

表２　アルゴリズム効率化による効果検証結果

	ルール抽出の計算時間（秒）		減少率
	効率化前	効率化後	
最大ノード数：3	58	41	29%
最大ノード数：4	17,642	14,808	16%

4.3　アルゴリズム効率化の効果検証

3.4 節で検証を行った線区，条件において，アルゴリズム効率化の効果の確認を実施した。実装は Python で，RAM 8.0 GB の一般的な 64bit Windows PC を使用した。実行結果を表 2 に示す。この結果より，アルゴリズムの効率化によって計算時間を約 16% から 30% 程度改善できることを確認した。なお，検証は行っていないが，PrefixSpan 法は深さ優先探索で解くアルゴリズムであるため，並列化による更なる高速化も期待できる[5)]。

5．運転整理ルールを更新する手法

運転整理ルールを更新する際には，

① 信頼度や支持度が基準値に満たないもの

② 一度運転整理ルールになったものの，事例の追加に伴い，支持度と信頼度のいずれかが成立条件を満たさなくなったもの

のうち，今後の事例の追加によって再度運転整理ルールになりうるものは残しておくことが必要である。

そこで，運転整理ルールになりうるものを「正規の運転整理ルール」，「消去猶予中の運転整理ルール」，「運転整理ルール候補」というステータスを持たせてデータベース上に残しておく。また，ここではあるルール A に対して冗長なルール B があるとき，A を B の上位のルールと表現するが，各ルールについて上位のルールの有無を管理する。これらの情報を用いて，適切な事例を残しつつ，不要な事例を削除する仕組みを構築した。

信頼度や支持度を満たして上位の正規の運転整理ルールが無いものを正規の運転整理ルールとし，②の事例を消去猶予中の運転整理ルール，①の事例のうち信頼度・支持度が一定以上の値であるものを運転整理ルール候補と定義する。本研究では，運転整理ルールが消去と復活を繰り返すことを防ぐために，消去猶予の状況のものについては抽出すべき運転整理ルールとしている。また，以上のステータス区分とは別に，上位ルールの有無を区分してルールにステータスを持たせる。通常は上位の運転整理ルールが存在する場合は冗長なルールとなるためデータベースには登録しないが，上位の運転整理ルールが消去猶予のステータスである場合に限り，下位の運転整理ルールをデータベースに登録し，下位のルールであることが分かるように，「上位の消去猶予ルール有り」とする。下位の運転整理ルールをこの時点で登録する理由は，上位の運転整理ルールが消去猶予の状態から一定期間が経過することによって消去されたときに，その下位に位置する運転整理ルールの事例・反例（条件部を満たすときに帰結部が満たさない事例）の数を登録し，シームレスに下位の運転整理ルールを生成するためである。そのため，上位の消去猶予ルールがある場合は，抽出する運転整理ルールの対象外となる。運転整理ルール候補の各ステータスの条件とステータス間の遷移は図 8 のようになる。

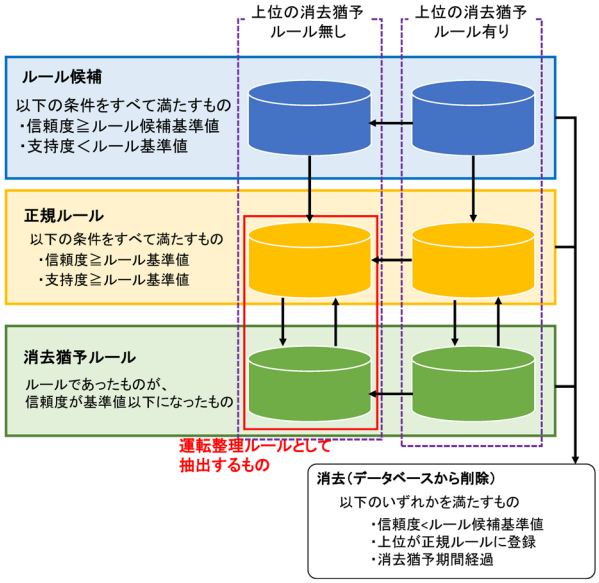


図８　ルールのステータスの推移

なお，ルールの変容に対応させるため，事例 1 件の重みを一律に評価せずに，古い事例ほど軽く，新しい事例ほど重く評価するように事例の重みを調整し，新旧の事例が混在する実績データから，新たな環境に即したルールの抽出を可能とした。

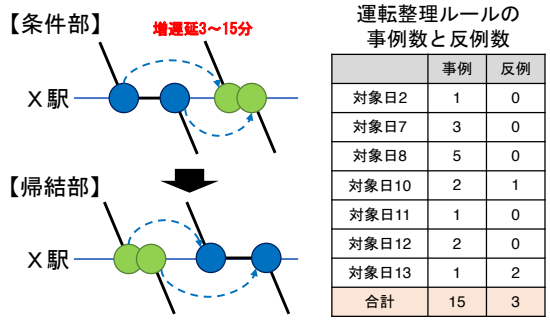
6．実路線によるルール抽出の試行

実在する都市圏の線区において，過去に実際にダイヤ乱れが発生した日 14 日分を対象として運転整理ルールの抽出のシミュレーションを行った。1 日ごとに運転整理ルール抽出作業を実施し，抽出のたびに運転整理ルールのステータスの更新作業を実施した。運転整理ルールの抽出と更新に用いるパラメータは，N=4，minsup=5，minconf=0.8，運転整理ルール候補の最小信頼度を 0.4 とした。なお，半年間の事例に対し，消去猶予期間を 1 年に設定したため，猶予期間を満了して消去されたルールは存在しない。

各事例の運転整理ルールの抽出・更新が終了時点での運転整理ルールの抽出数を表 3 に示す。事例 14 の運転整理ルールの抽出と更新を終了した時点で，運転整理

表３　運転整理ルールの抽出・更新結果

	変更命令数	時系列データ数（概数）	ルール候補数（概数）	正規ルール	消去猶予ルール	消去数	ルール抽出にかかる時間（分）	ルール更新にかかる時間（秒）
対象日1	55	8,100	290	0	0	0	47	2
対象日2	12	1,300	430	0	0	0	55	4
対象日3	58	10,000	1,100	0	0	13	108	15
対象日4	22	4,000	1,400	0	0	6	121	20
対象日5	36	3,700	1,500	0	0	24	94	27
対象日6	21	4,900	2,000	0	0	59	167	40
対象日7	53	9,200	2,500	0	0	138	233	62
対象日8	58	8,900	2,900	22	0	154	184	88
対象日9	27	2,900	3,000	22	0	83	19	97
対象日10	55	9,300	3,300	34	6	151	164	125
対象日11	57	9,000	4,500	37	6	182	97	171
対象日12	350	37,000	5,200	23	20	555	1212	317
対象日13	35	5,700	5,300	32	20	188	129	291
対象日14	130	13,000	5,700	40	25	322	177	299



運転整理ルールの事例数と反例数

	事例	反例
対象日2	1	0
対象日7	3	0
対象日8	5	0
対象日10	2	1
対象日11	1	0
対象日12	2	0
対象日13	1	2
合計	15	3

図９　抽出された正規ルールの例

ルールとして，65 件（正規の運転整理ルール 40 件，消去猶予中の運転整理ルール 25 件）が抽出された。今回の試行の範囲内では，上位の消去猶予ルールがあるルール候補に該当するものは無かった。

正規の運転整理ルールとして抽出された例を図 9 に示す。この運転整理ルールの条件部にあてはまる状況が 18 件あったうち，15 件において帰結部を満たす事例である。すべての対象グラフは，ノードが列車の着発（通過は同一の時刻を持つ 2 つのノードとして表現）を表し，点線で示したアークは着発事象の発生する順序を示している。変更が発生する条件を赤色の文字で示している。そのため，条件部・帰結部のそれぞれの解釈は以下のとおりとなる。

【条件部】

・　X 駅に停車する列車と X 駅を通過する列車が存在
・　X 駅において停車列車の発の後に通過列車が通過
・　通過列車が 3 ～ 15 分の範囲で遅延している

【帰結部】

・　X 駅において停車列車の到着前に通過列車が通過

なお，運転整理ルールの抽出にかかる時間は，19 分から 20 時間と大きなばらつきがあった。20 時間かかった事例は，1 日に 2 回人身事故が発生し，終日ダイヤが乱れた事例であり，その事例を除くと 4 時間以内に収まった。抽出にかかる時間は，変更命令数や時系列データ数だけではなく，探索するグラフパターンの形状や，ダイヤの状況により大きく変化することがわかった。運転整理ルールの更新にかかる時間は，事例数が多くなるとルール更新にかかる時間が増加する傾向が見られるものの，いずれも 10 分以内に計算できることが確認できた。ルールを抽出・更新するタイミングは，一連の運転整理が終了するごと（例えば，乱れが発生した日ごと）で十分であるため，N=4 の場合であれば実用に耐えうる時間で抽出できたことが確認できた。実験した線区の規模でより複雑な運転整理ルールを抽出するためには，先述したとおり並列化の処理等による対策が必要と思われる。

7．まとめ

運転整理において指令員のノウハウ等を抽出する手法について，主に実用規模の路線に適用可能とすることを目的として，冗長性の排除や高速化に関するアルゴリズムの高度化と日々の更新方法を提案した。これらを考慮することにより，日々の実績ダイヤのデータが取得可能である線区において，暗黙知である指令員のノウハウ等を形式知として取得することを可能とした。

今後の課題として，総遅延時分等を指標とした運転整理アルゴリズムとのインターフェースを整備した上で実効性について評価を行い，運転整理支援の全体の枠組みを構築することが挙げられる。

文　献

1）佐藤圭介，平井力：旅客視点の指標に基づくダイヤ乱れ時の列車順序・間隔整理手法，鉄道総研報告，Vol.30, No.8，pp.11-16, 2016
2）坂口隆，佐藤圭介：列車順序変更計画へのルール抽出技術の適用，鉄道総研報告，Vol.32，No.12，pp.11-16，2018
3）喜連川優：データマイニングにおける相関ルール抽出技法，人工知能学会誌，Vol.12，No.4，pp.513-520, 1997
4）Sharma, P., Balakrishna, G., “PrefixSpan: Mining sequential patterns by prefix-projected pattern,” International Journal of Computer Science & Engineering Survey, Vol.2, No.4, pp.111-122, 2011.
5）Jian, P., Jiawei, H., Behzad, M., Jianyong, W., et al. “Mining sequential patterns by pattern-growth: the PrefixSpan approach,” IEEE Transactions on Knowledge and Data Engineering, Vol.16, No.11, pp.1424−1440, 2014.