

最長径路法を用いた列車運行シミュレーション†

安部 恵 介^{††} 荒屋 真 二^{†††}

列車運行管理への自動化技術の導入において、事故等によるダイヤ乱れからの速やかな復旧を図る運転整理は、複雑な組合せ問題であり、定量的評価も難しいため最も自動化が困難な分野である。そのため運転整理における計算機支援方式は、シミュレーションをベースとした方式がとられており、緊急時における有効性の点からシミュレーションの効率化が強く要求されている。本論文では、まず運転整理における列車運行モデルに対し、その制約条件をグラフ表現する方法を示す。そして、シミュレーションをグラフ上の最長径路探索として効率的に行う最長径路方式を提案する。また、運転整理では部分的条件変更が繰り返されるという特徴に着目して、変更影響の波及部分のみを求めて最長径路長を更新することにより、実際の観点からいっそう効率化を図ったパラメトリック方式も提案する。これらのグラフ方式では、従来の一般的なイベントシミュレーション方式に比べて、計算時間が大幅に短縮される。また条件変更に対してはグラフの修正で済み、変更矛盾によるデッドロック発生は、グラフのサイクルとして発生箇所のみならず原因も速やかに提示できるという利点もある。数値実験によりこれらの方式の有効性を検討する。

1. ま え が き

列車運行管理は、運行監視、進路制御、運転整理、指令伝達等、鉄道システムにおいて中枢的な役割を果たしている¹⁾。近年、列車運行管理への自動化技術の導入が進み、CTC をはじめ進路制御、運行監視、旅客情報案内等で計算機による自動化が実用化されている。一方、事故等によるダイヤ乱れ時に、速やかな復旧を図るために運転計画の修正を行う運転整理は、社会的にも運営者側にとっても影響が大きい。そのため、運行管理においても非常に重要な部分を占めているが、反面最も自動化が困難な部分を含んでいる。列車運行の高速高密度化に伴い、運転整理の重要性が増すとともに、指令員の負担軽減、運転整理の質的向上のために、計算機支援が望まれている。

運転整理における計算機使用は指令員とのマンマシン方式がとられ、運行実績、遅延情報等の判断資料の提供、整理案の進路制御への反映等に利用されている。しかし整理案作成自体は複雑な組合せ問題であり、最適性の評価指標も明確でないため、計算機による最適化は困難であり、順序変更レベルでの最適化の研究²⁾がなされている程度である。そのため、整理案作成における計算機の最も有力な役割は予測シミュレーションにより指令員の判断を補助することであり、この方式はすでに実用化されてきた^{3), 4)}。この場合、計算機の予測結果に対し指令員が修正を行うとい

う形で、シミュレーションが対話形式で繰り返される。また、順序変更、番線変更レベルでは自動判断を伴ったシミュレーション方式³⁾もあるが、最適性の保証はなくやはり指令員の修正が必要である。運転整理においては、一度に全体的な最適案を作成することは困難であり、きめ細かい運転整理を行うためには、シミュレーションが試行錯誤的に頻繁に繰り返せることが望ましい。また、筆者らが自動化の試みとして開発してきた運転整理エキスパートシステム⁵⁾もシミュレーションをベースとしており、実用性の面からその効率化が強く要求されている。そのような理由から、筆者らは列車運行シミュレーションの高速化の研究を進めてきた⁶⁾⁻⁸⁾。

実際の列車運行は多くの要因から決定される連続的軌跡となるが、運転整理における予測シミュレーションでは、要求される精度と計算時間の点から駅単位の離散的方式がとられている⁹⁾。従来方式^{3), 4)}もオンライン使用という観点から効率化されたものであるが、基本的には処理イベントのイベントテーブルへの登録、時刻順の選択という一般的なイベントシミュレーション方式を踏襲したものであった。これに対して筆者らは列車運行がスケジュール制御されるシステムであることに着目して、スケジュールにより規定されるイベント間の位相的關係を表現した因果グラフを用いる方式⁶⁾を開発した。この方式はイベント処理を因果順に行うことにより、イベント選択、処理を効率化し、計算時間の短縮を図ったものである。さらに本論文では、運転条件等の時間的要素も枝の重みとして表すことにより、列車運行をグラフ表現する方法を示

† Train Traffic Simulation Using the Longest Path Method by KEISUKE ABE (Central Research Lab., Mitsubishi Electric Corp.) and SHINJI ARAYA (Fukuoka Institute of Technology).

†† 三菱電機(株)中央研究所

††† 福岡工業大学通信工学科

す。そして、このグラフ表現を用いてシミュレーションを最長径路アルゴリズムで行うことにより、いっそう効率化を図った最長径路方式^{7),8)}を提案する。また運転整理では、部分的変更に対しシミュレーションが繰り返されるという特徴に着目し、グラフの部分的修正に対する、最長径路長の更新アルゴリズムを考案し、これを用いてさらに効率化を図ったパラメトリック方式^{7),8)}も提案する。これは最長径路方式を基にして、部分的変更に対しては、変更影響の波及部分のみを求める方式であり、計算量がシミュレーション対象範囲よりむしろ、変更影響の波及範囲に依存している。ゆえに局所の変更に対しシミュレーションが繰り返される場合効果的である。

なお、これらのグラフ方式は、運転整理において重要なデッドロック検出にも有効であり、デッドロック発生箇所のみならず、その原因も速やかに提示することができる。実際の線区データに基づき、数値実験によりこれらの方式の有効性を検討した結果、計算時間の大幅な短縮が見られ、今後の運転整理の円滑化、自動化への有効性が確認された。以下、第2章で列車運行条件とそのグラフ表現について、第3章でシミュレーション方式を説明し、第4章で数値実験による検討を行う。

2. 列車運行条件のグラフ表現

2.1 列車運行モデル

実際の列車運行は、車両性能、設備条件、旅客行動等多くの要因により決定されるが、これらを忠実に模擬するには膨大な計算時間が必要となるため、シミュレーションでは用途に応じたモデル化が行われる。整理案作成上の補助手段としての予測シミュレーションでは、連続型モデルによる詳細な運行軌跡は必要なく、駅単位または閉そく単位の離散的モデル化が行われ、通常計算時間の点から駅単位方式がとられる⁹⁾。

駅単位の離散的モデルでは、列車走行を列車の駅への到着、出発というイベントで表現し、駅間走行は直線で近似される。列車走行は運転条件とスケジュールにより規定され、スケジュールは列車ダイヤにより規定されている。また、運転条件は、車両性能、設備状態等を考慮してあらかじめ算出されたものであり、各駅ごとに列車種別、編成両数、列車間の関係等に応じて定められている。

予測シミュレーションにおける列車走行は次の考え方に基いて決められる。

(i) 正常状態では常に列車ダイヤに従って運転する。

(ii) ダイヤ乱れにより列車が遅れているときは、制約条件（運転条件とスケジュール）を満たす範囲で極力遅延回復を図る。

2.2 列車運行の定式化

前節のモデルに基づき、列車運行を定式化する。本モデルでは列車運行は列車の各駅着発時刻により決定される。列車運行の制約条件は次のようになる。

〈個々の列車の走行条件〉

- (1) 駅間最小走行時間
- (2) 各駅最小停車時間
- (3) 各駅着発時刻、使用番線

〈列車間の運行条件〉

- (4) 各駅最小進入・進出時隔
- (5) 駅間許容列車数
- (6) 最小折返し時間
- (7) 各駅出発順序、番線使用順序
- (8) 車両運用

このうち、最小走行時間、最小停車時間は基本的に列車種別ごとに定められており、

$a(i, j)$: i 駅での j 列車の到着時刻

$d(i, j)$: i 駅での j 列車の出発時刻

$A(i, j)$: 列車ダイヤ上での i 駅 j 列車到着時刻

$D(i, j)$: 列車ダイヤ上での i 駅 j 列車出発時刻

$R(i, l)$: $i \sim i+1$ 駅間の l 種別列車の最小走行時間

$S(i, l)$: i 駅での l 種別列車の最小停車時間

$l(j)$: j 列車の種別

とすると、個々の列車走行条件は

$$\begin{aligned} a(i+1, j) - d(i, j) &\geq R(i, l(j)) \\ d(i, j) - a(i, j) &\geq S(i, l(j)) \\ d(i, j) &\geq D(i, j) \\ a(i, j) &\geq A(i, j) \end{aligned} \quad (1)$$

と表される。

また、駅進入・進出時の列車間の時隔は、実際には駅構内のポイント群の動きにより決まる。しかし一般には対象となる2列車の状態・関係によりいくつかのパターンを定め、そのパターンごとに各駅での最小時隔が決められる。時隔パターンは、駅構造、モデルの詳細度に応じていろいろ考えられるが、ここでは次のものを考える。

$h_a(l_1, l_2)$: 種別 l_1 列車に続いて種別 l_2 列車が到着する場合の時隔パターン

$h_d(l_1, l_2)$: 種別 l_1 列車に続いて種別 l_2 列車が出発する場合の時隔パターン

$h_b(l_1, l_2)$: 種別 l_1 列車に続いて種別 l_2 列車が同一番線を使用する場合の時隔パターン

このとき,

$H(i, h)$: i 駅での時隔パターン h のときの最小時隔

$j_a(i, j)$: j 列車の直前に i 駅に到着する列車

$j_d(i, j)$: j 列車の直前に i 駅を発車する列車

$j_b(i, j)$: j 列車の直前に j 列車の i 駅到着番線を使用した列車

とすると, 時隔による制約条件は,

$$\begin{aligned} a(i, j) &\geq a(i, j_a(i, j)) \\ &\quad + H(i, h_d(l(j_a(i, j)), l(j))) \\ a(i, j) &\geq d(i, j_b(i, j)) \\ &\quad + H(i, h_b(l(j_b(i, j)), l(j))) \quad (2) \\ d(i, j) &\geq d(i, j_d(i, j)) \\ &\quad + H(i, h_d(l(j_d(i, j)), l(j))) \end{aligned}$$

と表される.

また実際の列車走行は駅間の閉そく区間による制約も受けるが, ここでは駅単位のモデル化のため, これにかわるものとして, 駅間許容列車数を定める.

$k(i)$: $i \sim i+1$ 駅間許容列車数

$j_k(i, j, k)$: j 列車の k 本前に i 駅を出発した列車とすると, 駅間許容列車数による制約は,

$$d(i, j) \geq a(i+1, j_k(i, j, k(i))) \quad (3)$$

となる.

また, 最小折返し時間も, 駅構造, 列車種別, 編成両数等により異なり, 次のようなパターンを考える.

$u(l_1, l_2, n_1, n_2)$: n_1 両の l_1 種別列車が n_2 両の l_2 種別列車につながるときの折返しパターン

このとき,

$n(j)$: j 列車の編成両数

$j_u(j)$: j 列車の前運用列車

$U(i, u)$: i 駅でのパターン u の最小折返し時間とすると,

$$\begin{aligned} d(i, j) &\geq a(i, j_u(j)) \\ &\quad + U(i, u(l(j_u(j)), l(j), n(j_u(j)), n(j))) \quad (4) \end{aligned}$$

となる.

なお, 出発順序, 番線使用順序, 車両運用は(2), (4)式で表されており, 列車運行の制約条件は, (1) ~ (4)式で表現される. このとき, 列車運行は(1) ~

(4)式より次式で求められる着発時刻により表現される.

〈列車着発時刻〉

$$\begin{aligned} a(i, j) &= \max(A(i, j), \\ &\quad d(i-1, j) + R(i-1, l(j)), \\ &\quad a(i, j_a(i, j)) + H(i, h_d(l(j_a(i, j)), l(j))), \\ &\quad d(i, j_b(i, j)) + H(i, h_b(l(j_b(i, j)), l(j)))) \\ d(i, j) &= \max(D(i, j), \\ &\quad a(i, j) + S(i, l(j)), \quad (5) \\ &\quad d(i, j_d(i, j)) + H(i, h_d(l(j_d(i, j)), l(j))), \\ &\quad a(i+1, j_k(i, j, k(i))), a(i, j_u(j)) + U(i, u(l(j_u(j)), l(j), n(j_u(j)), n(j)))) \end{aligned}$$

ただし, (5)式は一般的な表現であり, 先行列車が存在しないとき, 折返し駅でないときなどはそれらに対応する項は除外される. なお, ここで述べた定式化はあくまで一例であり, より簡略化あるいは詳細化しても以下の議論は成立する.

2.3 グラフ表現

上記, 列車運行モデルにおいて, 列車の駅到着, 出発という各イベントを点で表し, 制約条件により規定されるイベント間の関係を枝で表現することにより, 列車運行を有向グラフで表現する. グラフの点を

$V_a(i, j)$: j 列車が i 駅に到着するというイベント

$V_d(i, j)$: j 列車が i 駅を出発するというイベント

とすると, 制約条件は次のように有向枝で表現することができる.

(1)式の走行時間, 停車時間に関する条件は, 列車ダイヤで規定される列車の走行ルートを表す次の走行ルート枝で表現される.

枝 $(v_a(i, j), v_a(i+1, j))$, 重み $R(i, l(j))$

枝 $(v_a(i, j), v_d(i, j))$, 重み $S(i, l(j))$

また, ダイヤ上の着発時刻による制約は, ダミーの基準点 v_0 を設定することにより次の基準時刻枝

枝 $(v_0, v_a(i, j))$, 重み $D(i, j)$

枝 $(v_0, v_d(i, j))$, 重み $A(i, j)$

で表される. さらに(2)式の時隔による制約は出発(到着)順序枝

枝 $(v_d(i, j_a(i, j)), v_d(i, j))$,

重み $H(i, h_d(l(j_a(i, j)), l(j)))$

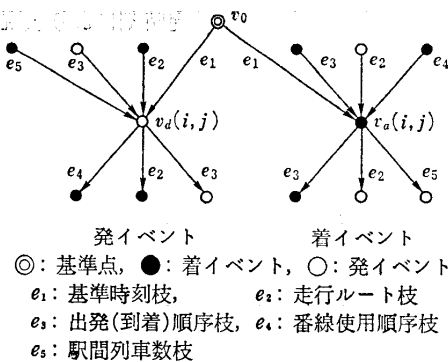


図 1 列車運行のグラフ表現

Fig. 1 Graph representation of train operation.

枝 $(v_d(i, j_a(i, j)), v_a(i, j))$,
重み $H(i, h_a(l(j_a(i, j)), l(j)))$

と番線使用順序枝

枝 $(v_d(i, j_b(i, j)), v_a(i, j))$,
重み $H(i, h_b(l(j_b(i, j)), l(j)))$

で表される。また、(3)式の駅間許容列車数による制約は次の駅間列車数枝

枝 $(v_d(i+1, j_k(i, j, k(i))), v_a(i, j))$, 重み 0

で表され、(4)式の折返し時間に関する制約は、次の車両運用枝

枝 $(v_d(i, j_u(j)), v_a(i, j))$,
重み $U(i, u(l(j_u(j)), l(j), n(j_u(j)), n(j)))$

で表される。

このようにして、図1のようなグラフが作成される。図1は一般的な形であり、折返し駅では車両運用枝が付加される等、状況に応じて接続関係は若干異なる。なお、ここで述べたグラフ表現は、次章で提案する最長径路方式、パラメトリック方式で用いられるものである。因果グラフ方式では、イベント間の位相的關係のみを対象としていたため、上記グラフから枝の重みと基準時刻枝を取り除いたグラフが用いられていた。

3. シミュレーション方式

3.1 グラフ方式の導入

運転整理における予測シミュレーションには以下のような特徴がある。

- 1) スケジュール制御されるシステムを対象とする。
- 2) 離散的な近似モデルに基づくイベントシミュレーションである。
- 3) 緊急時でのオンライン処理としての時間的制約を受ける。

4) 対話形式で、部分的条件変更に対し繰り返し用いられる。

5) ダイヤ変更に対するデッドロック検出機能が重要である。

上記特徴 3), 4) により、運転整理の円滑化、質的向上、さらには自動化のためにシミュレーションの高速化が強く望まれている。従来の列車運行シミュレーション方式^{3), 4)}も、特徴 3) の観点から駅単位の近似モデルによる効率化を図ったものであるが、処理方式自体は一般的なイベントシミュレーションの考え方に基づいている。すなわち、システム状態(列車群の位置)を離散的なイベント(列車が駅に到着、出発するという事象)の発生により変化させていく方式である。具体的にはイベント処理(イベントの発生およびそれに伴うシステムの状態変化)が可能となったイベントをその処理可能時刻とともにイベントテーブルに登録しておき、処理可能時刻の早い順に処理イベントを選択し、イベント処理を行っていく。このときイベントテーブル内のイベントの時刻順のサーチまたはソーティングが必要となる。またイベント間の因果関係が明確でないため、イベントテーブルへ登録する時点では、他イベントとの因果関係のチェックはなされていない。そのため、時刻順に選択されたイベントが、イベント処理の際に処理不能であることが判明し、再度処理可能時刻の計算、処理イベントの選択のやり直しが必要となる場合が多い。

これに対して筆者らは、上記特徴 1) に着目して、スケジュールにより規定されるイベント間の因果関係をあらかじめグラフ表現しておく因果グラフ方式⁶⁾を開発した。この方式では、因果グラフを用いることにより処理イベントの選択を因果順に行うことが可能となる。これにより従来方式における処理イベント選択のための時刻順のサーチや、イベント選択と因果関係チェックとの間の冗長な処理が不要となり、計算が効率化される。

また本論文では、さらに時間的要素も含めてすべての条件をグラフ表現し、シミュレーションにおける処理をすべてグラフ上の処理として行うことにより効率化を図った最長径路方式^{7), 8)}を提案する。さらに、最長径路方式を基に、特徴 4) に着目して実用的観点からいっそう効率化を図ったパラメトリック方式^{7), 8)}を提案する。

図2にこれらの3種類のグラフ方式の処理概要を示す。まず、列車ダイヤ、運転条件、設備条件等のデー

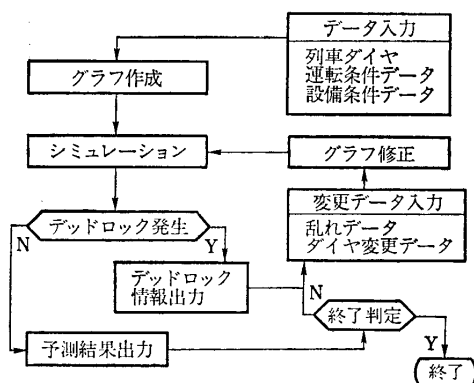


図2 シミュレーション処理概要

Fig. 2 Process flow of simulation.

タから、グラフを作成する。この部分はオフライン処理可能である。運転整理においては対話形式で処理が進められる。まず、列車遅延等の乱れ情報入力、予測シミュレーション、結果表示が行われる。それに対し、指令員によるダイヤ変更、予測シミュレーションが、最終的に満足のいく整理案が得られるまで繰り返される。ダイヤ変更の矛盾によるデッドロック発生は、シミュレーションの際に検出され、発生箇所、原因等の情報を表示し、指令員による矛盾箇所の修正が行われる。

これらのグラフ方式は以下のような特長を有する。

〈グラフ方式の特長〉

- i) グラフ表現を用いたシミュレーションの効率化。
- ii) グラフ作成後の条件変更に対しては、変更部分に関してのみグラフを修正すればよい。
- iii) デッドロック発生箇所のみならず、その原因も明確に提示できる。

特長 i) は因果グラフ方式、最長径路方式、パラメトリック方式の順により顕著となるものであり、特長 ii), iii) は上記 3 方式に共通である。特長 ii) は特徴 4) の観点から大きな時間節約となる。また特長 iii) は特徴 5) の観点から重要である。

3.2 最長径路方式

本モデルでは、列車運行はスケジュールと運転条件により決定され、これらの制約条件は枝の重みとして時間的要素も与えることにより、前章で示したように有向グラフで表現される。すなわち、運転整理における近似モデルでは、列車運行に対し、日程計画、生産計画等多くのスケジューリング問題と同様な、グラフ表現が適用できる。シミュレーションは各列車の各駅での着発時刻を求めることであり、これは、グラフの

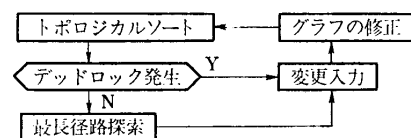


図3 最長径路方式の基本処理ループ

Fig. 3 The fundamental loop of longest path method.

点で表されるイベントの発生する時刻(イベント時刻)として与えられる。各列車が極力、計画ダイヤへの復帰を図るという列車走行の考え方は、PERT における最早開始時刻に対応しており、(5)式で与えられる各点のイベント時刻は、基準点から各点までの最長径路長として求められる。このようなシミュレーション方式を最長径路方式と呼ぶことにする。

シミュレーションは次の 2 段階より成る。

i) 制約条件(変更データ)からグラフを作成(修正)する。

ii) 基準点から各点までの最長径路長を求める。グラフにおけるサイクルは、列車運行におけるデッドロック現象に対応する¹⁰⁾。したがってスケジュールに整合性があれば、アサイクリックグラフとなる。グラフは一度作成すれば、以降は変更箇所のための修正で済む。最長径路を求める際にはまずグラフのトポロジカルソート¹¹⁾が行われる。これは、アサイクリックグラフ $G=(V, E)$ に対し、すべての枝 (u, v) ($(u, v) \in E$) において $\text{num}(u) < \text{num}(v)$ となるように、点 v ($v \in V$) に番号 $\text{num}(v)$ を割り当てることによる順序付けであり、 $\text{num}(v)$ は点 v のトポロジカルオーダーと呼ばれる。

グラフにサイクルが存在する場合は、トポロジカルソートの際に検出され、サイクルを構成する枝により、デッドロックの原因であるダイヤ上の矛盾箇所も検出される。このときいったん処理を中止し、まず矛盾箇所を修正し、その結果アサイクリックグラフとなり、トポロジカルソートが終了したならば、次に各点の最長径路長を求める。これはトポロジカルオーダーを用いることにより容易に求められる¹¹⁾。ここで、基準点 v_0 から点 v までの最長径路の長さを $\text{length}(v)$ とすると、

$$\text{length}(v) = \max[\text{length}(u) + (uv)] \quad (6)$$

$$u \in P(v)$$

$$\text{for all } v \in V - \{v_0\}$$

(uv) : 枝 (u, v) の重み, $P(v)$: 点 v の先行隣接点となる。最長径路を求める計算量は、トポロジカル

ソートも含めて、 $O(|V| + |E|)$ である¹¹⁾。最長径路方式の基本的処理部分は、図3のようなループとなる。

3.3 パラメトリック方式

運転整理では、一度に全体的な最良案を作成することは難しく、対話形式でシミュレーションを繰り返しながら徐々に改善されていく。とくに、きめ細かい運転整理を行うためには、細かい修正に対するシミュレーションが頻繁に繰り返されることが望ましい。このような部分的修正に対するシミュレーションでは、全体の処理をやり直すよりは、前回の結果を基にして、変更される部分のみを求めたほうが効率的である場合が多い。そこで、グラフの一部分が修正された場合に、その影響波及探索により、最長径路長の値が変化する部分のみを求めるアルゴリズム、ならびにそれを用いてシミュレーションを行うパラメトリック方式を提案する。

グラフの修正には次の3種類がある。

- (i) 枝の重み修正
- (ii) 枝の付加・除去
- (iii) 点（および枝）の付加・除去

列車遅延、臨時速度制限等の乱れ情報は(i)に、ダイヤ変更のうち、順序変更、運用変更、番線変更等は(ii)、運休、増発等は(iii)に対応している。

$\bar{E}$: 重み修正枝の集合、

$E^+(E^-)$: 付加(除去)枝の集合

$V^+(V^-)$: 付加(除去)点の集合

とし、元のグラフを $G=(V, E)$ 、修正グラフを $G'=(V', E')$ とする。まず、最長径路方式と同様に、乱れ情報、ダイヤ変更等の変更入力に対しグラフを修正する。処理の流れは、最長径路方式とほぼ同様であるが、トポロジカルオーダー、最長径路長の値を、変更される部分に関してのみ修正するという点が異なっている。図4にパラメトリック方式の基本的な処理の流れを示す。

$P'(v)(L'(v))$: 修正グラフ G' における点 v の先行(後続)隣接点

としたとき、修正グラフ G' において、

$$\text{num}(v) > \max[\text{num}(u)] \quad (7)$$

$$u \in P'(v)$$

が成立している点 v をトポロジカルオーダーに関する整合点、成立しない点を矛盾点と呼び、

$$\text{length}(v) = \max[\text{length}(u) + (uv)] \quad (8)$$

$$u \in P'(v)$$

が成立している点 v を最長径路長に関する整合点、成

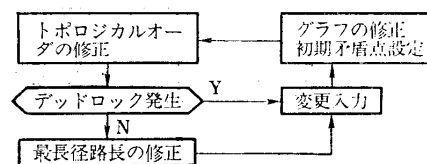


図4 パラメトリック方式の基本処理ループ
Fig. 4 The fundamental loop of parametric method.

立しない点を矛盾点と呼ぶことにする。グラフを修正する際に修正部分に関して矛盾点を検出し、それを初期矛盾点として設定しておく。ここで

$S_0(T_0)$: 最長径路長(トポロジカルオーダー)の初期矛盾点の集合

とする。初期矛盾点の設定に要する計算量は、グラフの修正部分も含めて $O(|\bar{E}| + |E^+| + |E^-| + |V^+| + |V^-|)$ である。ただし、計算量は $|P'(v)|$ に依存するが、2章で定義したグラフにおいて $|P'(v)| \leq 4$ であり、定数とみなせる。なお、以降の計算量の評価についても同様である。

次に初期修正の影響波及を探索し、修正グラフの最長径路長、トポロジカルオーダーを修正する。以下にトポロジカルオーダーの修正アルゴリズムを示す⁷⁾。

〈トポロジカルオーダーの修正アルゴリズム〉

1° $i \leftarrow \min[\text{num}(v)] - 1$;

$v \in T_0$

$T \leftarrow T_0$;

$(u, v) \in E^+$, $v \in T_0$ である点 v に対して

$F(v) \leftarrow \{v\}$ とする。

2° $T = \phi$ ならば終了。

3° $i \leftarrow i + 1$ 。

4° $\text{num}(u) = i$ である矛盾点 $u (u \in T)$ があれば、

$\text{num}(u) \leftarrow \max[\text{num}(z)] + 1$;

$z \in P'(u)$

$T \leftarrow T - \{u\}$

として、点 u を整合点にする。そうでなければ、2°へ。

5° 点 u の後続隣接点 $w (w \in L'(u))$ に対して、

$w \in F(u)$

であれば、枝 (u, w) を含むサイクルが存在し、終了する。そうでなければ、

$F(w) \leftarrow F(w) \cup F(u)$

とする。

6° 4°の結果、これまで整合点であった、点 u の後続隣接点 $w (w \in L'(u), w \in T)$ において、 $\text{num}(w) \leq \text{num}(u)$

となれば,

$T \leftarrow T + \{w\}$

として点 w を矛盾点とする. 4° へ戻る.

T : トポロジカルオーダの矛盾点の集合

$F(v)$: 点 v の先祖 (点 v 自身も含むものとする)

のうち, 付加枝の終点かつ初期矛盾点である
点の集合

上記アルゴリズムは, 修正グラフ G' に対するトポロジカルオーダの修正と, サイクルの検出を同時に行うものであり, 次の性質⁷⁾が基本となっている.

[性質 1] $\text{num}(v) < \min[\text{num}(u)]$ である点 $v (\in V')$
 $u \in T_0$

は修正の影響を受けない (整合点である).

[性質 2] 修正グラフ G' にサイクルが存在するならば, そのサイクルには, 終点が初期矛盾であるような付加枝 (u, v) ($(u, v) \in E^+$, $v \in T_0$) が少なくとも一つ含まれる.

性質 1 より,

$\text{num}(v) \geq \min[\text{num}(u)]$ である点 $v (\in V')$
 $u \in T_0$

に対して, トポロジカルオーダの値の小さい順に, 順次矛盾点の修正 (4°), 後続隣接点への修正影響の探索 (6°) を繰り返すことにより, トポロジカルオーダの修正処理が進められる⁷⁾. その結果, 矛盾点がなくなれば ($T = \phi$), すべての点のトポロジカルオーダが正しく求められ, 処理は終了する (2°).

また, グラフ G' におけるサイクル検出は, 性質 2 より, トポロジカルオーダの修正過程において, 点 v ($(u, v) \in E^+$, $v \in T_0$) を含むサイクルが構成されるかどうかを調べるにより行われる (5°).

トポロジカルオーダの修正の結果, 修正グラフ G' にサイクルが存在しなければ, 次に最長径路長の修正を行う. なお, サイクルが存在する場合は, 最長径路方式と同様に, デッドロック情報を提示し, 修正入力を行う. 最長径路長の修正においても, サイクル検出処理が不要という点以外は, トポロジカルオーダの修正の場合と同様の議論が成り立つ. 以下に最長径路長の修正アルゴリズムを示す⁷⁾.

〈最長径路長の修正アルゴリズム〉

1° $i \leftarrow \min[\text{num}(v)] - 1$;

$v \in S_0$

$S \leftarrow S_0$;

2° $S = \phi$ ならば終了.

3° $i \leftarrow i + 1$.

4° $\text{num}(u) = i$ である矛盾点 u ($u \in S$) があれば,

$\text{length}(u) \leftarrow \max[\text{length}(z) + (zu)]$;

$z \in P'(u)$

$S \leftarrow S - \{u\}$

として, 点 u を整合点にする. そうでなければ, 2° へ.

5° 4° の結果, これまで整合点であった, 点 u の後続隣接点 w ($w \in L'(u)$, $w \in S$) において,

$\text{length}(w) \neq \max[\text{length}(z) + (zw)]$

$z \in P'(w)$

となれば,

$S \leftarrow S + \{w\}$

として点 w を矛盾点とする. 4° へ戻る.

S : 最長径路長の矛盾点の集合

なお, トポロジカルオーダ, 最長径路長とも, 修正に要する計算量は, 修正過程において生じた矛盾点の集合を U とすると $O(|U|)$ である.

4. 数値実験による検討

実際の線区データを用いて, 数値実験により上記方式の有効性を検討した. 表 1 にシミュレーション対象規模と, グラフ作成, 修正に要する時間を示す. 対象規模はグラフの点数と枝数 (最長径路方式およびパラメトリック方式のグラフ) であり, グラフ作成時間はほぼこれらに比例している. 因果グラフ方式に比べてグラフで表現される情報量が多いため, グラフ作成時間が大きくなるが, この部分はオフラインで行える処理であり仮にオンラインで行っても実用上問題ない程度の時間である. 運転整理で行うのは, 変更部分に対するグラフの修正である. これに要する時間は変更の種類によるが, 0.02~0.3 秒程度, 平均で約 0.1 秒で済んでおり, またこれは対象規模に依存しない.

次に, 上記 3 方式のシミュレーション時間について

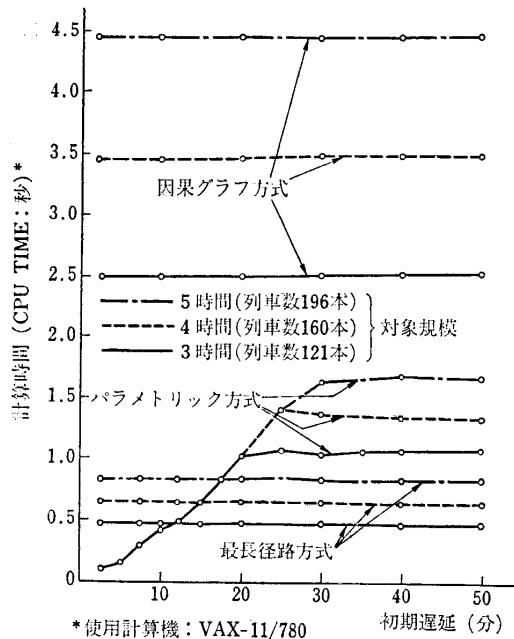
表 1 対象規模とグラフ処理時間

Table 1 Problem size and graph processing time.

シミュレーション 対 象 範 囲	3 時間	4 時間	5 時間
列車数 (駅数: 14)	121	160	196
点 数	2,619	3,645	4,275
枝 数	9,665	12,851	15,903
グラフ作成時間(秒)*1	2.6(0.8)	3.5(1.1)	4.5(1.3)
グラフ修正時間(秒)*2	0.1	0.1	0.1

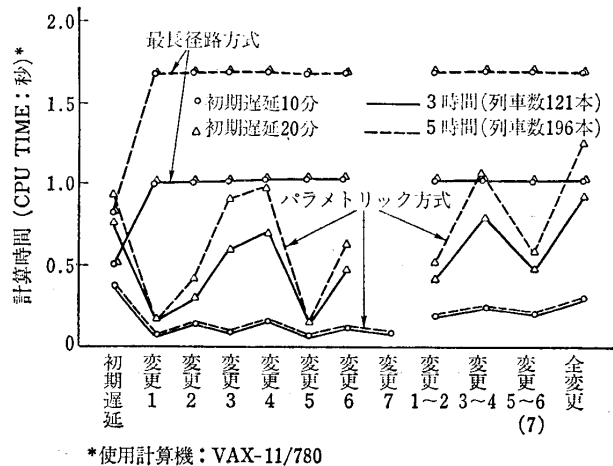
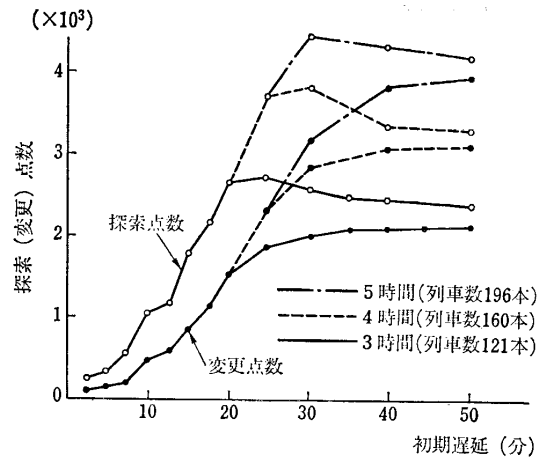
*1 () 内は因果グラフ作成時間 (使用計算機: VAX-11/780)

*2 平均値 (0.02~0.3 秒に分布)



検討する。図 5 にグラフ作成 (修正) 後の各方式のシミュレーションに要する計算時間を示す。これは初期遅延に対する運行予測を行った場合である。因果グラフ方式、最長経路方式の計算時間は対象規模に応じて一定であり、従来方式を改良した因果グラフ方式に対し、最長経路方式ではさらに効率化され、約 1/5 の計算時間で済んでいる。パラメトリック方式では変更影響の大きさにより計算時間が異なる。図 5 の例では、初期遅延 20 分くらいまでは、パラメトリック方式の計算時間は対象規模によらない。これは乱れの影響が 3 時間の範囲内に収束しているからであり、さらに乱れが大きくなると、対象規模により波及範囲が異なってくる。図 5 の例では、対象規模 3 時間の場合で、初期遅延 12.5 分、5 時間の場合で初期遅延 17.5 分までは、パラメトリック方式のほうが効率的であり、以降は最長経路方式のほうがよいことがわかる。これはパラメトリック方式のほうが 1 点の処理に要する計算量が多いためである。また、パラメトリック方式でもある程度乱れが大きくなると、計算時間は一定に収束し、このときの両方式の計算時間の比が、ほぼ 1 点当りの計算量の比に相当している。

パラメトリック方式の計算時間は、変更影響波及探索における、探索点数ならびに変更点数による。図 6 はパラメトリック方式における、初期遅延と計算量の関係



関係を示したものであり、図 5 の計算時間の変化はこれらの計算量の変化を反映したものである。

図 5 で示したように、パラメトリック方式の計算時間は変更影響の大きさにより変動し、最長経路方式との一般的な比較は難しい。ここでは運転整理における実用的観点から両方式の計算効率を比較する。図 7 に一連の運転整理における両方式の計算時間を示す。ダイヤ変更としては、順序変更と運用変更を行った。最長経路方式では、初期遅延、ダイヤ変更の種類によらず、対象範囲の大きさにより計算時間は一定 (初期遅延に対してはトポロジカルオーダーの計算が不要なので計算時間が約半分で済む) であるが、パラメトリック方式では初期遅延の大きさ、ダイヤ変更の種類により計算時間が大きく変動する。初期遅延 10 分の場合は

7個のダイヤ変更を行ったが、対象範囲3時間と5時間で計算時間は一致している。これは変更の影響が3時間の範囲内に収束しているためである。これに対し初期遅延20分の場合には計算時間が増加し、対象範囲により相違も見られる。しかし、いずれにしても、パラメトリック方式のほうが最長径路方式よりも計算時間が少なく、一連のダイヤ変更の合計で7~40%で済んでいる。また、いくつかのダイヤ変更をまとめて行う場合には、計算時間が若干増加し、最長径路方式とのトータルとしての計算時間の差もあまり大きくない。しかし、実際には少しずつ試行錯誤的にダイヤ変更を行う場合が多く、そのような場合には大きな時間節約となり、実用上パラメトリック方式の有効性は高い。

5. あとがき

最長径路法を用いた列車運行シミュレーション方式を提案し、数値実験により、従来方式に比べて計算時間が大きく短縮されることを示した。本論文では、駅単位のモデル化について述べたが、もっと細かく閉そく単位でも適用可能である。さらに、列車、線路、番線という概念を一般化して、複数の移動体が決められたスケジュールのもとに複数の設備を使用していくようなシステム全般に対して適用可能といえる。

また、対象システムがいったんグラフ表現されたならば、シミュレーションはグラフ上の処理として、対象の特性に依存しない一般的な形で行われるため、プログラムも簡潔であり、汎用性を有するという利点もある。

今後、あらかじめグラフ表現しておくことができない要因をも扱える形への拡張、運転整理の自動化、最適化への応用へと研究を進めていきたい。

謝辞 最後に、日頃よりご指導いただいている東京大学の曾根悟教授ならびに三菱電機(株)中央研究所の

上村勝彦部長、武田捷一課長に感謝する。

参考文献

- 1) 長谷川：鉄道における運行管理システム、情報処理, Vol. 25, No. 3, pp. 222-231 (1984).
- 2) 安部, 荒屋：分枝限定法を用いた列車順序最適化手法, 計測自動制御学会論文集, Vol. 21, No. 5, pp. 514-521 (1985).
- 3) 辻野, 他：東海道・山陽コムトラックにおける運転整理機能について, 第21回鉄道におけるサイバネティクス利用国内シンポジウム論文集, pp. 154-158 (1984).
- 4) 大川, 他：SIMTRACS のソフトウェア, 第4回シミュレーション・テクノロジー・コンファレンス論文集, pp. 165-168 (1984).
- 5) 荒屋, 福森：列車運行スケジュール制御のエキスパートシステム, 第23回 SICE 学術講演会 (1984).
- 6) 安部, 小西, 他：因果グラフを用いた汎用列車運行イベントシミュレータ, 昭和59年電気学会全国大会 (1984).
- 7) 安部, 荒屋：スケジューリンググラフを用いたパラメトリックイベントシミュレーション技法, 第10回システムシンポジウム, pp. 101-106 (1984).
- 8) 安部, 小西, 他：パラメトリックイベントシミュレーション技法の鉄道システムへの応用, 第21回鉄道におけるサイバネティクス利用国内シンポジウム論文集, pp. 134-138 (1984).
- 9) 列車群管理プロジェクト研究室：列車トラフィック・シミュレーション・システムの研究, 鉄道技術研究報告, No. 1140 (1980).
- 10) 佐藤, 他：移動体の流れのデッドロック検出法と列車群制御への応用, 第19回鉄道におけるサイバネティクス利用国内シンポジウム論文集, pp. 236-240 (1982).
- 11) 伊理, 他：演習グラフ理論—基礎と応用—, コロナ社, 東京 (1983).

(昭和60年1月31日受付)

(昭和60年9月19日採録)