日本オペレーションズ・リサーチ学会和文論文誌
Transactions of the Operations Research Society of Japan
Vol. 64, 2021, pp. 101–125

移動需要に基づく鉄道運行計画決定問題に対する分解アルゴリズム

松永 龍弥　　　柴田 秀哉
三菱電機株式会社

(受理 2020 年 4 月 1 日; 再受理 2021 年 5 月 21 日)

和文概要　現状の鉄道は，基本的に固定の列車ダイヤに従って運行されている．しかし，移動需要データを活用し動的にダイヤを立案できれば，利用者の利便性は向上する．動的な列車ダイヤ立案の実現に向けた研究例は多く，小規模な路線網については実用化に向けた取組みも始まっている．しかしながら，都市部に求められるような大規模な路線網を対象とする場合，解くべき最適化問題の求解時間が飛躍的に増大し，実用的な時間でのダイヤ立案が困難であるという課題があった．そこで，本論文では，計算量を路線数の線形オーダに抑制することを目標としてアルゴリズムを提案する．提案手法では，複数の路線に跨る移動需要を複数の単一路線の移動需要へと置き換えることで，解くべき最適化問題の規模を低減させ，高速化を実現する．また，解の精度低下を防止するために反復的に解を更新する手法を組み合わせることで，速度性能と解の精度維持を両立する．数値実験により，典型的な解法である局所探索法と提案手法を比較した結果，路線規模が拡大したときに提案手法は局所探索法よりも短時間で高精度の解を導出し，その有効性を確認した．一方で，提案手法の計算時間は路線数に対して線形オーダに抑えられることを期待したが，各反復における解の更新処理の計算時間が線形オーダより大きく，大規模な路線網に対しては計算時間のボトルネックになるという課題が判明した．今後は，この課題を解決し，大規模な路線網に対しても列車ダイヤを短時間で立案するための検討，評価を実施していく．

キーワード: 最適化, アルゴリズム, 鉄道, スケジューリング, 混合整数計画

1. はじめに

1.1. 背景と目的

近年の IoT 技術の発展により，ユーザの持つ端末から移動需要に関するデータを収集し，交通の利便性を高める試みが盛んになっている．例として，乗換案内サービスの経路検索実績データから鉄道の混雑率を推定し，ユーザに空いているルートを提示するサービスがある [13]．また，一部の地域では，移動需要に応じて動的に運行計画を変更するデマンド型交通も普及しつつある [10]．

一方で，主要な交通機関の一つである鉄道においては，基本的に固定の列車ダイヤの下で，日々の運行が行われている．鉄道においても，移動需要データに基づいて動的に列車ダイヤを変更することができれば，日々の移動需要の変動に応じた適切な計画を立案できるため，利用者に提供可能なサービスの質が向上する．

移動需要に応じた鉄道運行を実現するためには，短時間で列車ダイヤが立案できる必要がある．現状の列車ダイヤは，基本的に人手で作成している [20]．しかしながら，人手でオンデマンドに列車ダイヤを立案することは困難であるため，動的な列車ダイヤ立案の実現には短時間での列車ダイヤの自動作成技術が必要となる．列車ダイヤを自動作成する手法として，数理最適化手法によるアプローチがあり，様々な研究が行われている．小規模な路線網については，実用化に向けた取組みも始まっている [23]．

しかしながら，都市部に求められるような，数十路線，数百駅規模の大規模な路線網に対しては，自動作成手法が確立されていない．計画対象の路線規模が拡大すると，列車ダイヤを決定するために解くべき最適化問題の求解時間が飛躍的に増大し，オンデマンドの列車ダイヤ立案が困難となるためである．

そこで，本研究では，路線規模拡大時に生じる課題を解決し，大規模な路線網に対しても高速に列車ダイヤを立案可能とすることを目的とする．具体的には，列車ダイヤの計算時間が，対象路線数に対して線形オーダとなることを目標とする．複数系に跨るスケジューリング問題を対象とした部分問題分解手法が柴田ら [18] により提案されているが，本手法を鉄道運行計画決定問題に適用することで，アルゴリズムを構築する．提案手法では，複数の路線に跨る移動需要を，複数の単一路線の移動需要へと分解し，解くべき最適化問題の規模を低減させることで高速化を実現する．また，解の精度低下を防止するために反復的に解を更新する手法を組み合わせることで，速度性能と解の精度維持を両立する．また，数値実験により局所探索法との比較評価を行い，提案手法の有効性を検証する．

1.2. 関連研究

数理最適化手法による列車ダイヤの自動作成に関する従来研究を紹介し，本研究の位置づけを整理する．まず，単一路線を対象とした研究について紹介する．今田らは，移動者の移動時間の総和が最小になるような列車ダイヤを求める問題を混合整数計画問題として定式化し，分枝限定法により求解している [5]．國松らは，利用者の満足度と事業者の利益の双方を考慮した問題を扱い，メタヒューリスティクスとシミュレーションを組み合わせた解法を提案している [11]．Sun らは，停車駅における乗車人数に応じた停車時間モデルを用いた列車ダイヤの最適化手法を提案している [19]．解くべき最適化問題が大規模な凸計画問題となるため，ラグランジュ緩和法を利用した手法により求解している．Yang らは，移動需要に不確実性が含まれる状況を想定し，シミュレーションを活用した解法を提案している [24]．Barrena らは，メタヒューリスティクスの 1 つである適応的巨大近傍探索法を列車ダイヤ最適化に適用している [1]．Shen らは，移動時間の最小化と車両混雑解消の双方を目指す多目的最適化モデルを提案している [17]．

複数の路線で構成される路線網を対象とした研究も広く行われている．路線が複数となる場合，移動者が効率的に乗換ができるように，移動者の乗換待ち時間に着目した研究が多い．Wu らは，移動者の乗換待ち時間を最小にする列車ダイヤを求める問題を混合整数非線形計画問題として定式化し，遺伝的アルゴリズムによる解法を提案し，小規模の路線網に対して評価を行っている [22]．Wong らは，移動者の乗り換え待ち時間の最小化を目的とした列車ダイヤを決定する問題を混合整数計画問題として定式化し，緩和問題の解を用いた変数低減と分枝限定法を組み合わせた解法を提案している [21]．Jansen らは，移動者の乗換待ち時間最小化問題をタブーサーチを用いて求解している [6]．

周期的な列車ダイヤを想定した研究として，Sun ら [19]，Zhong ら [27]，Nachtigall ら [12] による研究がある．これらの研究では，移動者の待ち時間が最小になるように時間帯ごとに運行間隔を決定するためのメタヒューリスティクスを提案している．始発列車や終列車を対象とした研究として，Kang ら [8] や Yang ら [25] がある．Kang らは，始発列車を対象として，乗換駅において上手く接続できるような発車時刻を決定する問題を局所探索法で求解している [8]．Yang らは，終列車を対象として，移動需要の不確実性を考慮したタブーサーチによる解法を提案している [25]．

その他，様々な状況を想定した研究が行われている．坂口らは，列車ダイヤを，毎日運行

することが保証されている定期列車と日によって運行有無が変わる非定期列車に分け，非定期列車の運行有無を1日単位で決定するモデルを提案している [16]．Niu らは，2路線を対象に動的計画法と遺伝的アルゴリズムを組み合わせた解法を提案している [14]．Bertsimas らは，列車の運行頻度と価格を調整することで移動者の流れを制御して，効率の良い移動を実現するためのモデルを提案している [2]．

これらの先行研究では，単一路線を対象としているか，あるいは，複数路線を対象とする場合は特定の状況を仮定していることが多い．具体的には，周期的な列車ダイヤを仮定して運行頻度のみを決定する，始発・終電の接続のみを考慮する，などの状況である．そのような状況を限定していない研究は，計画対象の路線網が拡大すると計算時間が急速に増大するため，数十路線，数百駅規模の問題を実用的な時間内に求解できない．運行計画を立案する問題は混合整数計画問題であり，路線規模の拡大に伴い整数変数の数が増大するためである．これに対し，本研究では，列車ダイヤが非周期的であり，かつ計画期間が始発・終電付近などの特定の期間に限定されない状況において，複数路線を対象として，路線数の線形オーダの計算時間で列車ダイヤを立案することを目指す．

1.3. 論文構成

本稿の構成を以下に示す．2章では，本研究で扱う鉄道の運行計画決定問題を定式化し，路線規模拡大時の課題について述べる．3章では，提案手法の基となる部分問題分解手法について説明する．4章では，部分問題分解手法を鉄道運行計画決定問題に適用する．5章では，数値実験により提案手法の評価を行う．6章では，結論と今後の展望について述べる．

2. 鉄道運行計画決定問題

本章では，本研究で扱う鉄道の運行計画決定問題の定式化，及び路線規模拡大時の課題について説明する．なお，運行計画決定問題の定式化にあたっては，Bertsimas ら [2] の定式化を参考にした．

2.1. 問題設定

鉄道運行計画決定問題の目的は，列車運行に係る事業者コストを一定以下に抑えるという制約の下で，移動者の移動需要が与えられたときに，移動者の総移動時間が最小になるような列車ダイヤを求めることである．ここで，移動需要とは，出発時刻，出発駅，到着駅の組を指す．以降，出発時刻は移動者の出発駅への到着時刻に等しいものとする．また，出発駅と到着駅の組を OD ペアと呼ぶ．

本研究では，路線規模拡大時の求解方法に着目するため，複数路線の扱いとは独立に考えうる要素について，単純化した問題を扱う方針とする．すなわち，単一路線に閉じる部分の鉄道運行に関する単純化を行う．以下，移動者，及び鉄道運行に関する仮定を示す．移動者に関する仮定は以下の通りである．

- 各 OD ペアに対して経路候補は予め定められ，各移動者は経路候補の中からいずれかの経路を選択する．
- 経路の選択基準は移動時間にのみ依存するものとし，運賃や乗換回数などの他の要素は考慮しない．

鉄道運行に関する仮定は以下の通りである．

- 駅間所要時間は全列車で同一とし，各駅における停車時間は全列車，全駅で一定とする．更に簡単化して，各駅における停車時間は0分であるとする．

- 列車ダイヤは分単位で決定される．
- 全ての駅間所要時間は分単位で決まり，状況や時間帯に依らず一定である．
- 列車の収容可能人数は全列車で同一である．
- 全ての列車はその路線の始発駅を出発し，終着駅まで到達する．すなわち，途中駅を始発駅や終着駅とするような列車は考慮しない．
- 急行や特急は考慮しない. 全ての列車はその路線上の全ての駅に停車する．従って，通過待ちなどに必要な番線のやりくりについても考慮しない．
- 決定するのは列車ダイヤのみであり，車両や乗務員の運用計画は考慮しない．従って，車両基地や回送列車についても考慮しない．
- 運行間隔に関する制約は設けない．すなわち，列車は最短 1 分間隔で運行可能である．

2.2. 記号の定義

本節では，定式化に必要な記号を定義する．

2.2.1. 入力データ

$\mathcal{L}$	:	路線の集合．上りと下りは別路線として区別する．
$\mathcal{S}$	:	駅の集合．
$\mathcal{S}_l$	:	路線 $l \in \mathcal{L}$ 上の駅集合．$\mathcal{S}$ の部分集合．
$\lvert l \rvert$	:	路線 $l \in \mathcal{L}$ 上の，始発駅を除く駅数．
$n(l,s)$	:	路線 $l \in \mathcal{L}$ 上の駅 $s \in \mathcal{S}_l$ の順番．始発駅を 0 とする．終着駅は $\lvert l \rvert$ となる．
$t(l,s)$	:	路線 $l \in \mathcal{L}$ 上の始発駅から駅 $s \in \mathcal{S}_l$ への所要時間．
$\mathcal{OD}$	:	出発駅と到着駅の組の集合．$\mathcal{OD} = \mathcal{S}^2 \setminus \Delta_{\mathcal{S}}$. 但し，$\Delta_{\mathcal{S}}$ は $\mathcal{S}$ の対角集合を表す．本集合の要素を移動と同一視する．
$\mathcal{R}$	:	経路の集合．経路は出発駅，到着駅，全ての乗換駅，全ての経由路線から成る．
$\mathcal{R}_{(s_o,s_d)}$	:	移動 $(s_o,s_d) \in \mathcal{OD}$ を実現する移動経路候補の集合．$\mathcal{R}$ の部分集合．
$\lvert r \rvert$	:	経路 $r \in \mathcal{R}$ で経由する路線数．乗換回数に 1 を加算した数に等しい．
$l(r,i)$	:	経路 $r \in \mathcal{R}$ における i 番目の経由路線．$1 \le i \le \lvert r \rvert$.
$s^{\mathrm{tfr}}(r,i)$	:	経路 $r \in \mathcal{R}$ における i 番目の乗換駅．$0 \le i \le \lvert r \rvert$. $i=0$ は出発駅，$i=\lvert r \rvert$ は到着駅を表す．
$\tilde{s}(l,n)$	:	路線 $l \in \mathcal{L}$ 上の n 番目の駅．$n=0$ のとき路線 l の始発駅．
$\mathcal{T}$	:	運行計画対象期間，すなわち，路線の始発駅を列車が出発し得る時刻の集合．$\mathcal{T} = \{0,\dots,T-1\}$. 但し，$T$ は時刻の個数である．
$\mathcal{T}_{(r,i)}$	:	路線 $l(r,i)$ の駅 $s^{\mathrm{tfr}}(r,i)$ に列車が停車する可能性がある時間帯．

$$\mathcal{T}_{(r,i)} = \left\{ t \in \mathcal{T} \;\middle|\; t(l(r,i), s^{\mathrm{tfr}}(r,i)) \le t \le T + t(l(r,i), s^{\mathrm{tfr}}(r,i)) - 1 \right\} \tag{2.1}$$

但し， $(s_o,s_d) \in \mathcal{OD}$, $r \in \mathcal{R}_{(s_o,s_d)}$, $0 \le i \le \lvert r \rvert$.

B_l	:	路線 $l \in \mathcal{L}$ の列車本数の上限値．
K	:	各列車の収容可能人数．
$d_{t,(s_o,s_d)}$	:	出発駅 s_o, 到着駅 s_d, 出発時刻 t の移動需要を持つ人数．

2.2.2. 決定変数

$x_{t,l}$: 0-1 変数．列車が時刻 $t \in \mathcal{T}$ に路線 $l \in \mathcal{L}$ の始発駅を出発するとき 1, そうでないとき 0.

$y_{t,r,i}$: 非負整数変数．移動 $(s^{\mathrm{tfr}}(r,0), s^{\mathrm{tfr}}(r,|r|)) \in \mathcal{OD}$ を行う移動者のうち，経路 r を選択し，かつ時刻 t に経由駅 $s^{\mathrm{tfr}}(r,i)$ に到達する人数．但し，$t \in \mathcal{T}_{(r,i)}$, $(s_o, s_d) \in \mathcal{OD}$, $r \in \mathcal{R}_{(s_o,s_d)}$, $0 \leq i \leq |r|$.

$z_{t,r,i}$: 非負整数変数．移動 $(s^{\mathrm{tfr}}(r,0), s^{\mathrm{tfr}}(r,|r|)) \in \mathcal{OD}$ を行う移動者のうち，時刻 t に始発駅を出発し路線 $l(r,i)$ 上を移動する列車に乗車する人数．但し，$t \in \mathcal{T}$, $(s_o, s_d) \in \mathcal{OD}$, $r \in \mathcal{R}_{(s_o,s_d)}$, $1 \leq i \leq |r|$.

決定変数 $x_{t,l}$ は列車ダイヤを，$y_{t,r,i}$ は移動者の移動計画を，$z_{t,r,i}$ は移動者の列車への割当をそれぞれ決定する．

2.3. 制約条件

制約条件は，予算制約，経路選択制約，乗車時刻制約，降車時刻制約，容量制約から成る．なお，本節及び2.4節では，総和記号の上の添字が下の添字より小さい場合は，その項を0とみなすこととする．

2.3.1. 予算制約

各路線の運行する列車本数は，上限値を超えてはならない．本制約は次式で表される．

$$\sum_{t \in \mathcal{T}} x_{t,l} \leq B_l, \quad {}^{\forall} l \in \mathcal{L}. \tag{2.2}$$

2.3.2. 経路選択制約

各移動者は，経路候補集合からいずれか 1 つの経路を選択しなければならない．本制約は次式で表される．

$$\sum_{r \in \mathcal{R}_{(s_o,s_d)}} y_{t,r,0} = d_{t,(s_o,s_d)}, \quad {}^{\forall} t \in \mathcal{T}_{(r,0)}, {}^{\forall}(s_o, s_d) \in \mathcal{OD}. \tag{2.3}$$

$$\sum_{t \in \mathcal{T}_{(r,i)}} y_{t,r,i} \leq \sum_{t \in \mathcal{T}_{(r,i-1)}} y_{t,r,i-1}, \quad {}^{\forall}(s_o, s_d) \in \mathcal{OD}, {}^{\forall} r \in \mathcal{R}_{(s_o,s_d)}, {}^{\forall} i \in \{1, \ldots, |r|\}. \tag{2.4}$$

2.3.3. 乗車時刻制約

移動者が i 番目の経由路線の列車に乗るためには，その列車が $(i-1)$ 番目の乗換駅に到着するより前に，移動者が経由駅に到着しなければならない．本制約は次式で表される．

$$\sum_{\tau=0}^{t} z_{\tau,r,i} \leq \sum_{\tau=t(l(r,i-1),s^{\mathrm{tfr}}(r,i-1))}^{\min\{t+t(l(r,i),s^{\mathrm{tfr}}(r,i-1)),\, T+t(l(r,i-1),s^{\mathrm{tfr}}(r,i-1))-1\}} y_{\tau,r,i-1},$$
$$\quad {}^{\forall} t \in \mathcal{T}, {}^{\forall}(s_o, s_d) \in \mathcal{OD}, {}^{\forall} r \in \mathcal{R}_{(s_o,s_d)}, {}^{\forall} i \in \{1, \ldots, |r|\}. \tag{2.5}$$

2.3.4. 降車時刻制約

移動者が i 番目の経由路線を走行する列車から降車する時刻と，i 番目の乗換駅への到達時刻は同一である．本制約は次式で表される．

$$y_{t+t(l(r,i),s^{\mathrm{tfr}}(r,i)),r,i} = z_{t,r,i},$$
$$\quad {}^{\forall} t \in \mathcal{T}, {}^{\forall}(s_o, s_d) \in \mathcal{OD}, {}^{\forall} r \in \mathcal{R}_{(s_o,s_d)}, {}^{\forall} i \in \{1, \ldots, |r|\}. \tag{2.6}$$

2.3.5. 容量制約

各列車の乗車人数は，最大乗車可能人数以下でなければならない．本制約は次式で表される．

$$\sum_{(r,i)\in X(l,s)} z_{t,r,i} \leq K x_{t,l}, \quad {}^{\forall} l \in \mathcal{L}, {}^{\forall} s \in \mathcal{S}_l, {}^{\forall} t \in \mathcal{T}. \tag{2.7}$$

但し，ここで $X(l,s)$ は

$$X(l,s) = \left\{ (r,i) \;\middle|\; \begin{array}{l} l(r,i) = l,\, n(l, s^{\mathrm{tfr}}(r,i-1)) \leq n(l,s) < n(l, s^{\mathrm{tfr}}(r,i)), \\ i \in \{1,\dots,|r|\},\ (s_o,s_d) \in \mathcal{OD},\ r \in \mathcal{R}_{(s_o,s_d)} \end{array} \right\} \tag{2.8}$$

で定義される集合であり，路線 $l \in \mathcal{L}$ 上の駅 $s \in \mathcal{S}$ を含むような経路とその旅程の組を表す．

2.4. 目的関数

本問題の目的は，移動者の総移動時間を最小化することである．移動時間は乗車時間と待ち時間の和で表されるため，それぞれの総和を考える．

2.4.1. 待ち時間

移動 $(s_o, s_d) \in \mathcal{OD}$ を行い，かつ経路 $r \in \mathcal{R}_{(s_o,s_d)}$ を選択する全ての移動者の中で，時刻 $t \in \mathcal{T}_{(r,i)}$ において $(i-1)$ 番目 $(1 \leq i \leq |r|)$ の乗換駅で列車を待っている人数を $\mathbf{wait}(t,r,i)$ とおく．$\mathbf{wait}(t,r,i)$ は，

$$\mathbf{wait}(t,r,i) = \sum_{\tau=t(l(r,i-1),s^{\mathrm{tfr}}(r,i-1))}^{t} y_{\tau,r,i-1} - \sum_{\tau=0}^{t-t(l(r,i),s^{\mathrm{tfr}}(r,i-1))} z_{\tau,r,i} \tag{2.9}$$

と表される．ここで，右辺第 1 項は時刻 t までに駅 $s^{\mathrm{tfr}}(r,i-1)$ に到着する移動者の総数を，右辺第 2 項は時刻 t までに駅 $s^{\mathrm{tfr}}(r,i-1)$ に到着する列車に乗車する移動者の総数を表す．この式は，乗車時刻制約 (2.5) において，時刻を表す添え字 t を $t - t(l(r,i), s^{\mathrm{tfr}}(r,i-1))$ に置き換え，右辺から左辺を引いたものに等しい．

$\mathbf{wait}(t,r,i)$ を用いると，移動者の総待ち時間は次式で表される．

$$\sum_{(s_o,s_d)\in\mathcal{OD}} \sum_{r\in\mathcal{R}_{(s_o,s_d)}} \sum_{i=1}^{|r|} \sum_{t\in\mathcal{T}_{(r,i)}} \mathbf{wait}(t,r,i) \tag{2.10}$$

ここで，待ち時間としてカウントするのは，列車がその駅に停車する可能性のある時間帯のみとする．列車が停車しない時間帯に待っている移動者は，計画対象の期間外に発車する列車によって移動者は移動するので，待ち時間に含めなくても問題無い．

2.4.2. 乗車時間

移動経路が定まれば，その経路における乗車時間は一意に定まる．移動者の総乗車時間は次式で表される．

$$\sum_{(s_o,s_d)\in\mathcal{OD}} \sum_{r\in\mathcal{R}_{(s_o,s_d)}} \sum_{t\in\mathcal{T}_{(r,0)}} t^{\mathrm{ride}}(r)\, y_{t,r,0} \tag{2.11}$$

ここで，$t^{\mathrm{ride}}(r)$ は

$$t^{\mathrm{ride}}(r) = \sum_{i=1}^{|r|} \bigl(t(l(r,i), s^{\mathrm{tfr}}(r,i)) - t(l(r,i), s^{\mathrm{tfr}}(r,i-1)) \bigr) \tag{2.12}$$

で定義される定数であり，経路 $r \in \mathcal{R}$ の総乗車時間を表す．列車が始発駅を出発する最大の時刻である $T-1$ に近い時刻に出発し，かつ乗換を含む移動を行う移動者は，乗換先の路線に乗車することが出来ず，目的地まで到達できない場合がある．この場合も，運行計画対象期間外に目的地まで移動できたとみなし，目的地までの総乗車時間をカウントする．

2.5. 最適化問題のクラス

決定変数は 0-1 変数及び整数変数であり，目的関数，制約条件ともに線形結合で表される．従って，本稿で扱う運行計画決定問題は整数線形計画問題に属する．また，簡単化のため，移動者を表す整数変数 $y_{t,r,i}$, $z_{t,r,i}$ を連続緩和すると，問題のクラスは 0-1 混合整数線形計画問題となる．

移動者を表す整数変数を連続緩和することによる精度低下は小さいと考えられるため，以後の議論では，移動者を表す決定変数 $y_{t,r,i}$, $z_{t,r,i}$ を連続変数として扱う．

2.6. 路線規模拡大時の課題

運行計画決定問題の変数，制約条件の数は，ともに計画対象路線における OD ペアの数に比例する．OD ペアは出発駅と到着駅の組み合わせであるため，その数は駅数の 2 乗に比例して増加する．そのため，計画対象とする路線規模が拡大すると，解くべき最適化問題の規模が大幅に増大する．運行計画決定問題は 0-1 混合整数計画問題であるため，問題規模が増大すると求解に掛かる時間が急速に増大し，実時間で運行計画が導出できなくなる．従って，計画対象の路線規模が大きい場合にも実時間で運行計画を導出するための手法が必要である．

3. 部分問題分解手法

本章では，文献 [18] を基に，柴田らが提案した部分問題分解手法を説明する．また，運行計画決定問題との関係性についても説明する．

3.1. 問題定義

3.1.1. 対象とする問題の特性

部分問題分解手法は，次の 3 つの性質を持つ問題を対象とする．

複数系の疎結合性

最適化対象は，複数の系の集合体である．各系は完全には独立しておらず，互いに疎結合している．運行計画決定問題において，系は路線に相当する．各路線は乗換駅を介して，複数路線に跨る移動に関する制約によって互いに結合している．

2 種類の決定変数の存在

最適化対象は，主に個別の系に閉じた制約を課された決定変数（以下，個別変数）と，本質的に複数系に跨った制約を課された決定変数（以下，全体変数）を持つ．個別変数は，系に閉じた手段により決定可能である．一方，全体変数は，系が完全に独立することを防ぐ役割を持ち，全体最適の余地を表している．運行計画決定問題において，列車ダイヤを表す $x_{t,l}$ が個別変数に，移動者の移動計画を表す $y_{t,r,i}$，及び移動者の列車への割当を表す $z_{t,r,i}$ が全体変数に対応する．

時刻間の依存性

最適化対象は，時刻の要素を持つ問題であり，複数時刻に跨った制約条件を持つ．運行計画決定問題において，予算制約，経路選択制約，乗車時刻制約は複数時刻に跨った制約条件である．

3.1.2. 最適化問題の定式化

定式化に必要な記号は以下の通りである．

集合

$\mathcal{N}$： 系の集合．

$\mathcal{T}$： 計画対象期間．$\mathcal{T}=\{0,\ldots,T-1\}$．但し，$T$ は離散化された時刻の個数である．

決定変数

$\boldsymbol{x}_{n,t}$： 個別変数．時刻 $t\in\mathcal{T}$, 系 $n\in\mathcal{N}$ に関する 0-1 ベクトル変数．系に関する独立性が強く，複数系に跨った制約条件に強くは縛られない．

$\boldsymbol{y}_{n,t}$： 全体変数．時刻 $t\in\mathcal{T}$, 系 $n\in\mathcal{N}$ に関する連続ベクトル変数．特定の系に帰属するが，複数系に跨った制約条件に本質的に縛られる．

以上の定義の下，前述した 3 性質を持つような問題として，次の最適化問題を対象とする．

$$\min \sum_{n\in\mathcal{N}}\sum_{t\in\mathcal{T}} f_{n,t}(\boldsymbol{x}_{n,t},\boldsymbol{y}_{n,t}) \tag{3.1}$$

$$\text{s.t.} \sum_{t\in\mathcal{T}}\left(\boldsymbol{A}_{n,t}\boldsymbol{x}_{n,t}+\boldsymbol{B}_{n,t}\boldsymbol{y}_{n,t}\right)\le \boldsymbol{c}_n,\ {}^{\forall}n\in\mathcal{N}, \tag{3.2}$$

$$\sum_{n\in\mathcal{N}}\sum_{t\in\mathcal{T}} \boldsymbol{D}_{n,t}\boldsymbol{y}_{n,t}\le \boldsymbol{e}. \tag{3.3}$$

ここで，$f_{n,t}$（$t\in\mathcal{T}, n\in\mathcal{N}$）はベクトルを引数に持つ実数値関数である．また，$\boldsymbol{A}_{n,t}$, $\boldsymbol{B}_{n,t}$, $\boldsymbol{D}_{n,t}$（$t\in\mathcal{T}, n\in\mathcal{N}$）は適当な行数及び列数を持った行列，$\boldsymbol{c}_n$（$n\in\mathcal{N}$），$\boldsymbol{e}$ は適当な長さの列ベクトルである．

式 (3.2) は系に閉じた制約条件を，式 (3.3) は複数系に跨る制約条件をそれぞれ表している．式 (3.3) において異なる系に関する変数が足し合わされており，系と系との間の相互依存を表している．運行計画決定問題において，系に閉じた制約条件に対応するのは予算制約，降車時刻制約，容量制約であり，複数系に跨る制約条件に対応するのは経路選択制約，乗車時刻制約である．なお，ここでは不等号制約のみを表記したが，等号制約が混在しても良い．

3.2. 部分問題分解手法

部分問題分解手法は，次の 3 点を基本方針とする．具体的な手順は 4 章での鉄道への適用の中で説明する．

部分問題への分解

問題を系毎に分解することで，問題規模を低減し，求解速度を向上させる．

反復的交互決定

個別変数と全体変数を交互に決定することで，一度に決定する変数を削減し，求解速度を向上させる．この工程を反復し，分解による精度低下を防止する．

逐次確定

時刻が早い順に決定変数を逐次的に確定する．これにより，反復回数を削減し，収束速度を向上させる．

4. 路線網分解手法

本章では，3 章で説明した部分問題分解手法を鉄道運行計画問題に適用し，路線数に対して線形オーダの計算時間で求解可能なアルゴリズムを構築する．

4.1. 概要

本研究で提案するアルゴリズムの流れを図1に示す．このアルゴリズムを以後，路線網分解手法と呼ぶこととする．以後の節では，ステップ毎に路線網分解手法の動作を説明する．なお，以後の説明において，列車ダイヤを表す変数 $x_{t,l}$ をまとめたベクトルを $\boldsymbol{x}$ と表記する．同様に，移動計画を表す変数 $y_{t,r,i}, z_{t,r,i}$ についても，同様に $\boldsymbol{y}, \boldsymbol{z}$ と表記する．

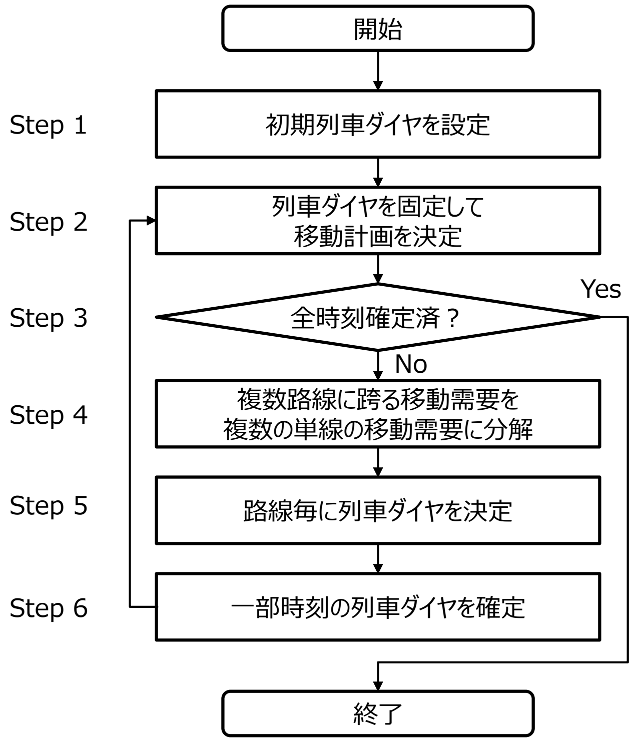


図 1: 路線網分解手法の流れ

4.2. 初期列車ダイヤの設定

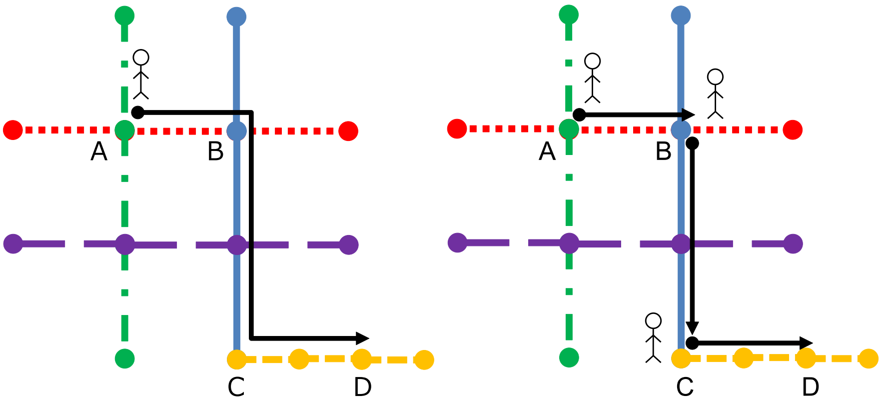
Step1では，初期列車ダイヤを定める．すなわち，列車ダイヤを表す0-1変数 $\boldsymbol{x}$ の初期解を設定する．ここで，初期解として，全ての $x_{t,l}$ を1とした解を与える．これは，予算を度外視して全路線全時刻で列車を運行させる初期列車ダイヤを設定することに等しい．この初期解は，列車ダイヤに関する制約条件である予算制約 (2.2) を必ずしも満たさないが，ここで定める初期解はStep2で移動計画を決定するためだけに使用されるため，制約条件を満たす必要は無い．初期解をこのように定める理由は，4.5節で説明する．

4.3. 移動計画の決定

Step2では，運行計画決定問題の変数のうち，変数 $\boldsymbol{x}$ を固定した問題を解くことで，移動者の移動計画を表す変数 $\boldsymbol{y}$, $\boldsymbol{z}$ を求める．変数 $\boldsymbol{y}$, $\boldsymbol{z}$ は連続変数であるため，移動計画を求める問題は線形計画問題となる．

4.4. 収束判定

Step3では，全時刻の列車ダイヤが確定済みか否かを判定する．確定済みの場合は，列車ダイヤを出力し，停止する．確定済みで無い場合はStep4に進む．



(a) 分解前 (b) 分解後

図 2: 移動需要分解

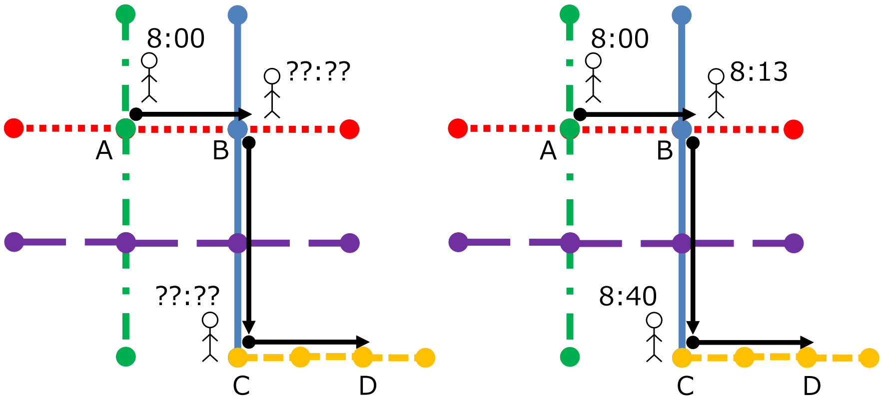
4.5. 移動需要の分解

Step4 では，Step2 で求めた移動計画を基に，複数路線に跨る移動需要を，複数の単一路線上の移動需要へと分解する．移動需要の分解の様子を図 2 に示す．図 2 では，駅 A →駅 B →駅 C →駅 D と 2 回の乗換を含む移動を行う移動需要を，駅 A →駅 B，駅 B →駅 C，駅 C →駅 D の，乗換を含まない 3 個の移動需要へと分解する．このように，移動需要の分解とは，$|r| \geq 2$ を満たし，経路 $r \in \mathcal{R}$ で移動する各移動を，出発駅 $s^{\mathrm{tfr}}(r, i-1)$ から到着駅 $s^{\mathrm{tfr}}(r, i)$ へと路線 $l(r, i)$ 上の列車に乗って移動する $|r|(1 \leq i \leq |r|)$ 個の移動へと置換することを表す．複数路線に跨る移動需要を分解することで，全ての移動需要が単一路線上のものとなるため，運行計画決定問題も路線毎に独立に解くことができる．そのため，元問題を直接解くのに比べ，高速に解くことができる．

移動需要の分解において，分解後の単一路線の移動需要がその出発駅に到着する時刻を推定する必要がある．この到着時刻の推定精度が低いと，実際の到着時刻との誤差によって移動者が乗換駅で長時間待つような質の低い列車ダイヤが生成されてしまう．本手法では，予め定めた列車ダイヤによって移動を行うときの各駅への到着時刻を，その推定値とすることで，移動需要分解による精度低下を防止する．精度低下を防止できる理由は 4.7 節で説明する．

図 3 に時刻推定の様子を示す．図 3a は図 2 と同じ移動需要を表しており，駅 A を 8:00 に出発する．暫定ダイヤの下で，駅 A から駅 B，駅 B から駅 C までの所要時間がそれぞれ 13 分，27 分と算出された場合，図 3b に示すように，駅 B の出発時刻を駅 A の出発時刻（8:00）に 13 分を加算した 8:13，駅 C の出発時刻を駅 B の出発時刻（8:13）に 27 分を加算した 8:40 と推定する．

反復回数が少ないときは到着時刻の推定精度が低く，移動者が乗換駅で長時間待つような列車ダイヤが生成され，1 つ前の反復より到着時刻が遅くなる可能性が高い．そのため，反復による到着時刻の遅延を考慮して，実際の到着時刻より早い時刻が推定値となるような初期解を与える必要がある．このため，Step1 で全ての $x_{t,l}$ を 1 とした初期解を与えている．全路線，全時刻で列車を運行させる状況では，実際の到着時刻より早い時刻に移動者は乗換駅に到着するという推定結果となる．



(a) 推定前　(b) 推定後

図 3: 時刻推定

4.6. 列車ダイヤの決定

Step5 では，Step4 で分解した移動需要を入力として，路線毎に列車ダイヤを決定する．複数路線に跨る移動需要は存在しないので，運行計画決定問題は路線毎に独立に求解できる．分解後の単一路線の列車ダイヤを決定する問題の定式化については，4.8 節で説明する．

4.7. 列車ダイヤの逐次確定

Step6 では，Step5 で求めた列車ダイヤに対して，確定していない期間の始点から一定期間の列車ダイヤを確定させ，Step2 に戻る．

移動需要分解と列車ダイヤの逐次確定の関係について説明する．始めの反復では，適当に定めた列車ダイヤを基に移動計画を決定するため，乗換駅への到着時刻の推定精度は低い．しかし，出発駅への到着時刻は確定しているため，始点に近い時刻では移動需要の分解による推定誤差の影響は小さい．そのため，始点に近い時刻の列車ダイヤは精度が高く，列車ダイヤを確定しても精度低下は発生しにくい．

2 回目以降の反復では，複数路線に跨る移動需要を分解したとき，計画対象期間の始点に近い時刻の列車ダイヤは確定済である．列車ダイヤが確定している期間内に発着する列車に移動者が乗車するとき，その移動者の乗換駅への到着時刻は正確に分かる．そのため，列車ダイヤが未確定の期間のうち，始点に近い時刻においては移動需要の分解による推定誤差の影響は小さい．従って，分解した移動需要に基づいて導出した列車ダイヤの中で，始点に近い時刻のものは精度が高い．始点に近い時刻の列車ダイヤを確定させるため，次の反復では移動需要分解時の時刻推定の精度が前の反復時より向上する．このように，移動需要の分解と列車ダイヤの逐次確定を組合わせることで推定精度が向上し，精度の高い列車ダイヤが立案可能となる．

本手順で確定する時刻の長さは特に限定されない．最も単純な方法は，各反復で 1 分だけ確定させていく方法である．一度に確定させる時刻が少なければ，最終的に導出される解の精度は高まる．その一方で反復回数が増え，求解時間が増加する．逆に，一度に確定させる時刻を多くすれば，求解時間を短縮させられるが精度低下を招く．すなわち，一度に確定させる時刻の数は，精度と速度のトレードオフを調整するパラメータの役割を果たす．確定させる時刻の長さと精度，速度との関係は，5 章において検証する．

4.8. 単一路線の運行計画決定問題の求解

移動需要を分解した後に，単一路線の運行計画決定問題を解く．本節では，単一路線の運行計画決定問題，及びその求解手法である局所探索法について説明する．計画対象の路線を $l \in \mathcal{L}$ とする．

4.8.1. 記号の定義

定式化に必要な記号を定義する．但し，2.2 節と同一の記号については省略する．

鉄道路線に関する入力データ

- $\mathcal{OD}_l$: 路線 l 上の出発駅と到着駅の組の集合．$\mathcal{OD}_l = \mathcal{S}_l^2 \setminus \Delta_{\mathcal{S}_l}$.
- $\mathcal{T}_{(l,s)}$: 路線 l 上の駅 $s \in \mathcal{S}$ に列車が停車する可能性がある時間帯.

$$\mathcal{T}_{(l,s)} = \{t(l,s), \dots, T + t(l,s) - 1\} \tag{4.1}$$

決定変数

- $y_{t,(s_o,s_d)}$: 非負整数変数．移動 $(s_o, s_d) \in \mathcal{OD}_l$ を行う移動者のうち，時刻 t に到着駅 s_d に到達する人数．但し，$t \in \mathcal{T}_{(l,s_o)}$, $(s_o, s_d) \in \mathcal{OD}_l$.
- $z_{t,(s_o,s_d)}$: 非負整数変数．移動 $(s_o, s_d) \in \mathcal{OD}_l$ を行う移動者のうち，時刻 t に始発駅を出発した列車に乗車する人数．但し，$t \in \mathcal{T}$, $(s_o, s_d) \in \mathcal{OD}_l$.

4.8.2. 制約条件

制約条件のうち，予算制約，乗車時刻制約，降車時刻制約，容量制約は，単一路線においても存在する．単一路線においては，経路は一意に定まるため，経路選択制約は存在しない．

予算制約

運行する列車本数は，上限値を超えてはならない．本制約は次式で表される．

$$\sum_{t \in \mathcal{T}} x_{t,l} \leq B_l. \tag{4.2}$$

乗車時刻制約

移動者が列車に乗るためには，その列車が出発駅に到着するより前に，移動者が出発駅に到着しなければならない．本制約は次式で表される．

$$\sum_{\tau=0}^{t} z_{\tau,(s_o,s_d)} \leq \sum_{\tau=t(l,s_o)}^{t+t(l,s_o)} d_{\tau,(s_o,s_d)}, \quad {}^{\forall} t \in \mathcal{T},\ {}^{\forall}(s_o, s_d) \in \mathcal{OD}_l. \tag{4.3}$$

降車時刻制約

移動者が列車から降車する時刻と，到着駅への到達時刻は同一である．本制約は次式で表される．

$$y_{t+t(l,s_d),(s_o,s_d)} = z_{t,(s_o,s_d)}, \quad {}^{\forall} t \in \mathcal{T},\ {}^{\forall}(s_o, s_d) \in \mathcal{OD}_l. \tag{4.4}$$

容量制約

各列車の乗車人数は，最大乗車可能人数以下でなければならない．本制約は次式で表される．

$$\sum_{(s_o,s_d)\in X(l,s)} z_{t,(s_o,s_d)} \le K x_{t,l}, \quad {}^{\forall} s \in \mathcal{S}_l, {}^{\forall} t \in \mathcal{T}. \tag{4.5}$$

但し，ここで $X(l,s)$ は

$$X(l,s) = \{\, (s_o,s_d) \mid n(l,s_o) \le n(l,s) < n(l,s_d),\ (s_o,s_d) \in \mathcal{OD}_l \,\}. \tag{4.6}$$

で定義される集合であり，駅 $s \in \mathcal{S}_l$ を含むような移動の集合を表している．

4.8.3. 目的関数

目的関数は，元問題と同じく，移動者の総移動時間の最小化である．但し，単一路線の場合，移動経路が列車ダイヤに依らず一意に定まるため乗車時間は定数である．そのため，目的関数は実質的に，待ち時間の最小化に等しい．

待ち時間

移動者の総待ち時間は，次式で表される．

$$\sum_{(s_o,s_d)\in\mathcal{OD}_l} \sum_{t\in\mathcal{T}_{(l,s_o)}} \left\{ \sum_{\tau=t(l,s_o)}^{t} d_{\tau,(s_o,s_d)} - \sum_{\tau=0}^{t-t(l,s_o)} z_{\tau,(s_o,s_d)} \right\} \tag{4.7}$$

乗車時間

単一路線の場合，乗車時間は定数である．移動者の総乗車時間は次式で表される．

$$\sum_{(s_o,s_d)\in\mathcal{OD}_l} \sum_{t\in\mathcal{T}_{(l,s_o)}} \left(t(l,s_d) - t(l,s_o)\right) d_{t,(s_o,s_d)} \tag{4.8}$$

4.8.4. 局所探索法

分解後の単一路線の運行計画決定問題は，局所探索法により求解する．局所探索法とは，暫定解の一部を変更して得られる近傍解の集合を探索し，現在の解より良い解が存在すれば置き換えるという操作を，指定した終了条件が満たされるまで繰り返すアルゴリズムである．本節では，局所探索法を用いて，運行計画決定問題を解くための手法について述べる．5 章では，元問題を分解せずに局所探索法により解く手法を路線網分解手法の比較対象とする．そのため，複数路線を対象として説明する．

局所探索法を用いて運行計画決定問題を解くために，まず本問題を組合せ最適化問題として捉えなおす．列車ダイヤを表す 0-1 変数 $\boldsymbol{x}$ を固定したときの移動者の移動計画を決定する最適化問題の最適値を $f(\boldsymbol{x})$ とおく．すると，運行計画決定問題は予算制約 (4.2) を満たす中で $f(\boldsymbol{x})$ を最小化する問題として表される．$\boldsymbol{x}$ の値が与えられたときの $f(\boldsymbol{x})$ の値は，線形計画問題を解くことで得られる．なお，本問題には予算制約 (4.2) が等号で成り立つ最適解が必ず存在するため，そのような解のみを探索対象とする．

次に，0-1 変数 $\boldsymbol{x}$ の近傍を定義する．ある 1 つの路線 l を選び，その中で $x_{t,l} = 1$ となる時刻と，$x_{t,l} = 0$ となる時刻をそれぞれ選ぶ．これらの時刻における $x_{t,l}$ の値を交換して得られる $\boldsymbol{x}$ を，元の $\boldsymbol{x}$ の近傍とする．これは，ある路線における 1 本の列車の発車時刻を変更することと同じである．図 4 に，$L = 3, T = 10$ における近傍解の例を示す．図 4a に示す暫

定解 $\boldsymbol{x}$ において，$x_{2,1}=1, x_{4,1}=0$ であるので，交換対象として選び，その値を交換する．すると，図 4 に示す近傍解となる．

t	0	1	2	3	4	5	6	7	8	9
路線1	0	0	1	0	0	0	1	0	0	1
路線2	1	0	0	1	0	1	0	0	1	0
路線3	0	1	1	0	0	0	1	0	0	1

(a) 暫定解

t	0	1	2	3	4	5	6	7	8	9
路線1	0	0	0	0	1	0	1	0	0	1
路線2	1	0	0	1	0	1	0	0	1	0
路線3	0	1	1	0	0	0	1	0	0	1

(b) 近傍解

図 4: 近傍解

局所探索法の動作について説明する．i 回目の反復開始時の 0-1 変数 $\boldsymbol{x}$ の値を $\boldsymbol{x}^{(i-1)}$，$\boldsymbol{x}^{(i-1)}$ の近傍全体を表す集合を $\mathcal{X}^{(i-1)}$ とおく．ランダムに選んだ $\boldsymbol{x}' \in \mathcal{X}^{(i-1)}$ に対し，$f(\boldsymbol{x}') < f(\boldsymbol{x}^{(i-1)})$ であれば $\boldsymbol{x}^{(i)} = \boldsymbol{x}'$ として，i 回目の反復を終了し，$(i+1)$ 回目の反復へと移行する．全ての $\boldsymbol{x}' \in \mathcal{X}^{(i-1)}$ に対し，$f(\boldsymbol{x}') \geq f(\boldsymbol{x}^{(i-1)})$ であるとき，反復を終了し，$\boldsymbol{x}^{(i-1)}$ を解として出力する．初期解 $\boldsymbol{x}^{(0)}$ は，予算制約 (4.2) が等号で成り立つ $\boldsymbol{x}$ をランダムに与える．

路線網分解手法では，各反復において一定期間の列車ダイヤを確定させる．そのため，局所探索法において，列車ダイヤを確定させた時刻に対応する変数 $\boldsymbol{x}$ は局所探索の交換対象から外す．具体的な動作を説明する．1 反復における確定期間を T^{fix} とすると，まず，1 反復目においては，局所探索法による解探索の終了後，$x_{t,l}\,(t=0,\ldots,T^{\text{fix}}-1)$ を確定する．$k\,(\geq 2)$ 反復目には，$(k-1)$ 反復目以前に確定したものは局所探索の交換対象から外して探索を行い，解探索の終了後，$x_{t,l}\,(t=(k-1)T^{\text{fix}},\ldots,kT^{\text{fix}}-1)$ を確定する．

4.9. 移動需要分解による問題規模低減効果

移動需要の分解による問題規模の低減効果について考察する．表 1 に，元の最適化問題と，分解後の最適化問題における変数と制約条件の数を示す．ここで，n_l は路線 l の駅数，N は全駅数を表している．簡単のため，複数路線に跨る駅を重複して数える．このとき，$N=\sum_{l\in\mathcal{L}} n_l$ が成り立つ．移動需要分解により，列車ダイヤを表す変数 $\boldsymbol{x}$ の数は $1/L$ まで減少する．移動計画を表す変数 $\boldsymbol{y}, \boldsymbol{z}$ の数，制約条件の数は，総駅数の 2 乗オーダから単一路線の駅数の 2 乗オーダまで減少する．混合整数計画問題は問題規模が増大すると求解時間が急速に増大する傾向を持つ．そのため，解くべき問題数は L 倍になっているものの，全体としては計算時間の短縮が期待できる．

路線間の駅数のばらつきと問題規模低減効果の関係について考える．ある路線の駅数が他の路線よりも極端に多いとき，その路線単体の問題の規模が元の問題と比較してそれほど小さくならない．そのため，問題規模低減効果は小さい．逆に，全路線が同程度の駅数であるとき，問題規模低減効果は大きい．表 2 に，各路線の駅数が等しい場合における問題規模低減効果を示す．列車ダイヤを表す変数 $\boldsymbol{x}$ の数は表 1 と同じ $1/L$ であるが，移動計画を表す変数 $\boldsymbol{y}, \boldsymbol{z}$ の数，制約条件の数は $1/L^2$ まで減少しており，問題規模の低減効果が大きいことが分かる．

表 1: 移動需要分解による問題規模の低減効果

	列車ダイヤ変数の数	移動計画変数の数	制約条件の数	問題数
元の問題	LT	$O(N^2T)$	$O(N^2T)$	1
分解後の問題 (路線 l)	T	$O(n_l^2T)$	$O(n_l^2T)$	L

表 2: 各路線の駅数が等しい場合の問題規模の低減効果

	列車ダイヤ変数の数	移動計画変数の数	制約条件の数	問題数
元の問題	LT	$O(L^2T)$	$O(L^2T)$	1
分解後の問題	T	$O(T)$	$O(T)$	L

5. 評価

本章では，4 章で提案した路線網分解手法の有効性を検証するための評価を行う．2 路線から 5 路線までの様々な路線網を対象とし，路線網分解手法を適用せずに局所探索法で求解する場合との比較を行う．この比較対象手法を以後，非分解手法と呼ぶこととする．

5.1. 評価条件

評価に使用するデータ，評価環境について説明する．

5.1.1. 評価用データ

評価用データは，鉄道事業者 [7, 15, 26]，国土交通省 [9] が公開している路線データ，移動需要データを元に，5 路線 57 駅分作成する．参考にしたデータは複数の鉄道事業者のものであるが，本評価では，全ての路線は仮想的な同一事業者に属するものとして扱う．なお，各路線は上り方向，下り方向の双方を含む．今回の評価で計画対象とする路線情報を表 3 に，路線図を図 5 に示す．後述する通り，計画対象期間を 1 時間，1.5 時間として評価を実施するが，表 3 の運行本数における「$T = 60$」は計画対象期間が 1 時間,「$T = 90$」は 1.5 時間のときの値を示す．スラッシュで区切られている値は，左側が上り，右側が下りの運行本数を表す．

移動者について，計画対象期間が 1 時間のときの出発駅毎の移動者数を表 4 に，出発時刻毎の移動者数を図 6 に示す．全移動者数は 114,780 人である．図 6 について，横軸は出発時刻，縦軸は全駅に亘る移動者の総数である．各駅を出発する移動者として，その駅を車両が通過し得る時刻に出発する移動者のみを対象としている．最も所要時間の長い路線 5 では始発駅から終点まで 51 分掛かるため，最大で $t = 110\,(= 60 + 51 - 1)$ まで移動者は出発する．路線上の各駅を車両が通過し得る時刻に出発を希望する移動者のみを対象としているため，計画対象期間の始めの期間と，計画対象期間以後における移動者数は少ない．表 4 について，全体的に複数の路線に跨る駅，すなわち乗換駅における移動者数が多い傾向がある．計画対象期間が 1.5 時間のとき，全移動者は 210,970 人である．詳細は省略するが，1 時間のときと同じ傾向である．

5.1.2. 評価環境

評価環境を表 5 に示す．線形計画問題の求解は数理最適化ソルバ Gurobi[4] により行う．Gurobi の設定は全て既定値とする．

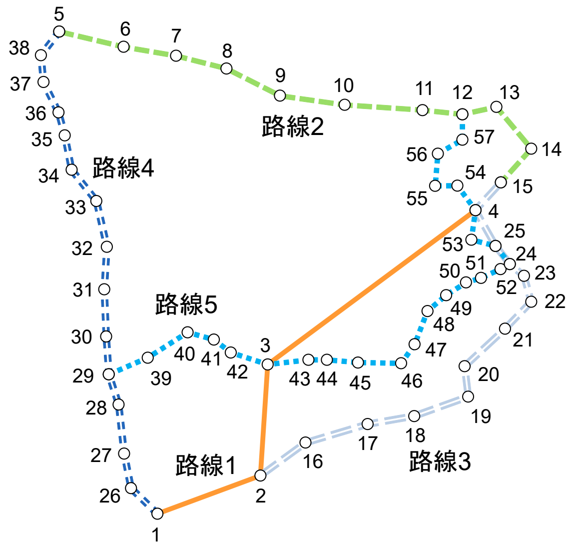


図 5: 路線図

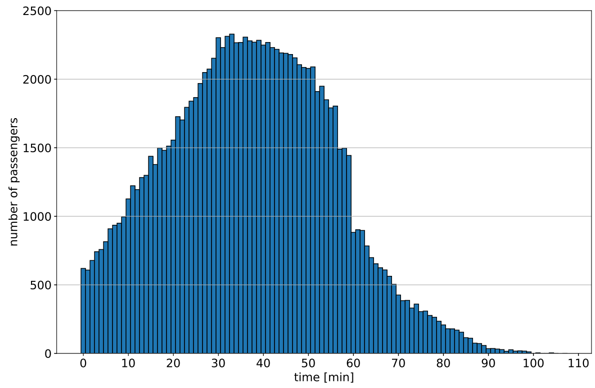


図 6: 出発時刻毎の移動者数 ($T = 60$)

表 4: 出発駅毎の移動者数 ($T = 60$)

駅番号	人数 [人]	駅番号	人数 [人]	駅番号	人数 [人]
1	10901	20	968	39	118
2	5190	21	1076	40	527
3	7287	22	851	41	398
4	22223	23	1460	42	444
5	11996	24	3590	43	81
6	789	25	3923	44	214
7	2434	26	379	45	688
8	879	27	561	46	373
9	1781	28	776	47	1720
10	1697	29	3723	48	394
11	455	30	854	49	549
12	4238	31	542	50	333
13	2185	32	447	51	396
14	734	33	2377	52	376
15	2026	34	598	53	147
16	689	35	607	54	256
17	893	36	1682	55	325
18	1484	37	326	56	458
19	1883	38	2209	57	270

表 3: 路線情報

路線番号	駅数	運行本数	
		$T = 60$	$T = 90$
1	4	10/6	14/11
2	11	9/9	14/13
3	13	15/14	22/20
4	15	9/9	14/14
5	25	9/10	14/15

表 5: 評価環境

項目	内容
CPU	Intel Core i5-6500 3.20 GHz
メモリ	64 GB
OS	Windows 7 Professional
プログラミング言語	Python 3.7.1
ソルバ	Gurobi 8.1.1

5.2. パラメータ依存性評価

5.2.1. 実施内容

路線網分解手法において，一度の反復で運行計画を確定させる期間はパラメータである．本評価では，このパラメータが提案手法の性能に及ぼす影響を検証する．

運行計画を確定させる期間を変更しつつ求解を行い，設定した計算時間内に導出した計画の評価値を比較する．計画対象の路線は 5.1.1 項で述べた全路線，計画対象期間は 1 時間とする．1 反復での確定期間は，1, 2, 3, 4, 5, 6, 10, 15, 30, 60 分とする．確定期間を T^{fix} としたとき，反復回数は $\lceil \frac{T}{T^{\text{fix}}} \rceil$ となる．$T^{\text{fix}} = 60$ の場合，1 回目の反復で計画期間全ての時刻の計画を確定するため，2 回目の反復は行われない．これは，単純に移動需要を分解して求解することと等しい．計算時間の上限は，計画対象期間と同じ 1 時間とする．実運用を考える場合，オンデマンドに遅延なく運行計画を立案するには，1 時間分の計画を 1 時間以内に求めなければならない．この観点から，計算時間の上限を設定した．

また，同じ計算を 5 回計算行い，各計算の評価値の平均で比較する．これは，局所探索法が乱択アルゴリズムであるためである．次節の評価においても同様である．

なお，評価値は，一人当たりの移動時間から，移動者が理想的に移動を行うときの移動時間，すなわち，全移動者が最も短い乗車時間で移動可能な経路を選択し，かつ待ち時間 0 分で移動するときの移動時間を差し引く．列車ダイヤの良否に依らない部分を除くためにこの補正を行う．

5.2.2. 結果と考察

評価結果を図 7 に示す．図より，$T^{\text{fix}} = 3$ のとき，2.93 分と評価値が最も良い．$T^{\text{fix}} \leq 2$ では計算時間上限までに全時刻の列車ダイヤが確定するまで計算が進まず，$T^{\text{fix}} \geq 4$ では計算時間上限までに全時刻の列車ダイヤが確定し，計算が収束した．$T^{\text{fix}} = 3$ では，計算が収束するケース，収束しないケースの双方があった．このことから，与えられた時間の中で計算が収束する最も大きい T^{fix} を選ぶことが，良い解の導出に繋がるといえる．

T^{fix} の変化に対する評価値の変動について，$T^{\text{fix}} = 60$，すなわち反復回数が 1 回のときは最も評価値が悪く，3.33 分であるが，$T^{\text{fix}} = 30, 20, 15, 10$ と小さくなるにつれて，評価値が良くなる．$T^{\text{fix}} = 10$ では 2.98 分と，$T^{\text{fix}} = 60$ のときと比較して 0.35 分だけ評価値が小さい．単純に移動需要を分解するだけでなく，反復最適化を組み合わせることで，精度が向上することを表している．

5.3. 耐規模性評価

5.3.1. 実施内容

路線網分解手法の耐規模性を評価するために，様々な路線規模の路線網に対し，路線網分解手法，非分解手法により求解し，導出した解の精度を比較する．路線網分解手法のパラメータ T^{fix} として，5.2 節にて最も良い評価値となった $T^{\text{fix}} = 3$ を採用する．

計画対象の路線網について，5.1.1 項で説明した 5 路線から，幾つかの路線を抽出したものを計画対象とする．なお，路線網分解手法は複数路線に跨る移動需要が存在する状況を想定した手法であるため，そのような移動需要が存在する路線の組合せを対象とする．そのため，全組合せ $2^5 - 1 = 31$ 通りのうち，1 路線のみの場合，また，路線同士が接続していない，路線 1 と路線 2，路線 3 と路線 4 の計 7 通りを除いた 24 通りの組合せに対して比較を行う．計算開始から 1 時間経過したときの，各手法により導出された解における一人当たり移動時間を比較する．また，計算時間の上限を 3 時間として，非分解手法と路線網分解手法が

表 6: 路線網分解手法と非分解手法との比較結果

路線数	路線の組合せ	駅数	1 時間計算時の移動時間の差 [分]	路線網分解手法の解を上回るのに掛かる時間の比率
2	1,3	15	-0.01	0.43
	1,4	18	0.04	-
	2,3	23	-0.04	0.15
	2,4	25	0.05	-
	1,5	27	-0.10	0.15
	3,5	35	0.01	-
	2,5	35	0.03	-
	4,5	39	-0.01	0.47
3	1,2,3	25	-0.04	0.38
	1,2,4	28	0.04	-
	1,3,4	29	-0.03	0.60
	1,3,5	36	-0.06	0.71
	1,2,5	37	-0.07	0.42
	2,3,5	37	-0.02	0.43
	1,4,5	40	-0.07	0.62
	2,3,5	44	-0.02	0.68
	2,4,5	48	0.00	-
	3,4,5	49	0.02	-
4	1,2,3,4	38	-0.02	1.37
	1,3,4,5	49	-0.01	1.27
	1,2,3,5	49	-0.09	0.67
	1,2,4,5	49	0.00	-
	2,3,4,5	57	0.02	-
5	1,2,3,4,5	57	0.04	-

同じ評価値の解を導出するのに掛かる時間の比較を行う．路線規模が最も大きい 5 路線の路線網については，計画対象期間を 1.5 時間とした比較評価も併せて実施する．

5.3.2. 比較結果と考察

路線網分解手法と非分解手法の比較結果を表 6 に示す．「1 時間計算時の移動時間の差」は，路線網分解手法，非分解手法のそれぞれの手法が導出した解における移動時間の差分を取ったものである．路線網分解手法のほうが移動時間が短いとき，すなわち良い解を導出しているときに正の値を取る．「路線網分解手法の解を上回るのに掛かる時間の比率」は，非分解手法が路線網分解手法を上回る解を初めて導出した時間を，路線網分解手法が収束するまでに掛かった時間で割った値である．この値は，1 時間計算時に非分解手法が路線網分解手法より移動時間の短い解を導出した路線の組合せに対して算出している．

路線網分解手法のほうが移動時間が短い解を導出する場合と，非分解手法のほうが移動時間が短い場合の双方がある．非分解手法が最も有利になる路線の組合せは，路線 1,5 の組合せで，その差分は 0.10 分，路線網分解手法が最も有利になる路線の組合せは，路線 $1, 2, 3, 4, 5$ の組合せで，その差分は 0.04 分であり，大きな差ではないといえる．

非分解手法が路線網分解手法を上回るのに掛かる時間の比率について，路線数毎の平均値は，2 路線では 0.30 倍，3 路線では 0.55 倍，4 路線では 1.10 倍と，路線数が増加するにつれて大きくなる．これは，路線規模の拡大により解空間が拡大し，非分解手法の解探索に時間が掛かるようになっていることを表している．

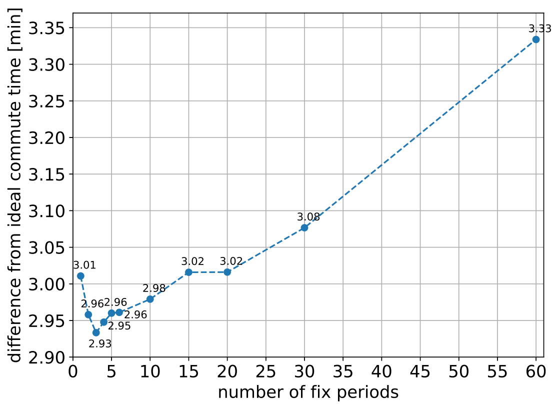


図 7: 1 反復での列車ダイヤ確定期間と評価値との関係

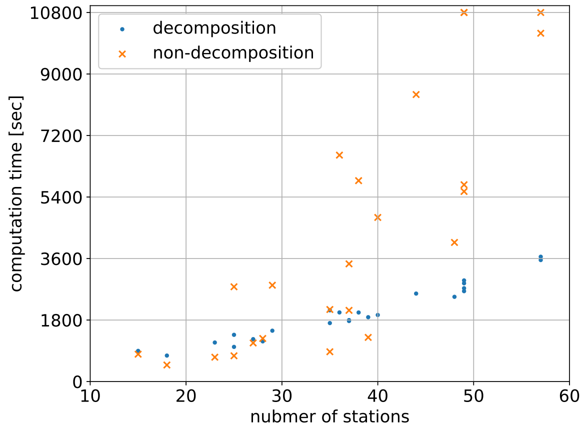


図 8: 同じ評価値の解を導出するまでに掛かる時間の比較

同じ評価値の解を導出するまでに掛かる時間を比較した結果を図 8 に示す．路線網分解手法と非分解手法のそれぞれで計算が収束したときの解のうち，移動時間が長いほうを基準として，その移動時間以下の解を導出するまでに掛かる時間を比較している．縦軸は計算時間を，横軸は各路線の組合せにおける駅数を表している．駅数が増加するときの計算時間の変化について，非分解手法に比べ，路線網分解手法のほうが増大度合いが緩やかである．路線網分解手法において，駅数に対して計算時間は概ね線形で増加している．駅数が 40 以下のとき，非分解手法のほうが計算時間が短い場合が多いが，駅数が 50 程度となると路線網分解手法のほうが計算時間が短くなる場合が多く，駅数が 60 程度では路線網分解手法のほうが計算時間が短い．4.9 節で述べた通り，非分解手法では，解くべき問題の規模が総駅数の 2 乗オーダであるのに対し，路線網分解手法では各路線の駅数の 2 乗オーダまで小さくなるためである．

5 路線における計算時間と解の精度の関係を図 9 に示す．なお，移動者が理想的な移動を行うときの平均移動時間は 19.28 分である．路線網分解手法では，計算時間が約 51 分経過したときに，最も移動時間の短い計画を導出しており，評価値は 2.93 分である．これに対し，非分解手法は，計算時間が 1 時間の時点では 0.04 分移動時間の長い計画を導出している．路線網分解手法が導出した計画より移動時間の短い計画を非分解手法が導出するには，約 81 分掛かり，6 時間計算を行うと路線網分解手法より 0.09 分移動時間の短い計画を導出する．

路線網分解手法と非分解手法の速度性能の違いは，計画対象期間を長くしたときに顕著に現れる．図 10 は，計画対象期間を 1.5 時間としたときの，5 路線における計算時間と解の精度との関係を示した図である．移動者が理想的な移動を行うときの平均移動時間は 20.68 分である．路線網分解手法は，計算時間が約 89 分経過したときに，最も移動時間の短い計画を導出しており，評価値は 3.03 分である．計画対象期間が 1.5 時間であるため，1.5 時間計算したときの評価値で比較をすると，非分解手法では評価値 3.45 分であり，路線網分解手法のほうが 0.42 分移動時間の短い計画を導出している．非分解手法で 6 時間計算を行ったとしても，評価値は 3.12 分であり，路線網分解手法のほうが 0.09 分移動時間の短い計画を導出している．

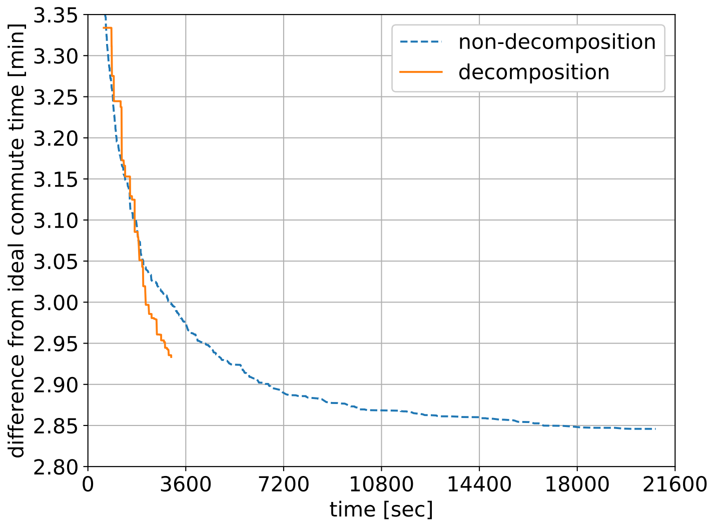


図 9: 5 路線における計算時間と評価値との関係 ($T = 60$)

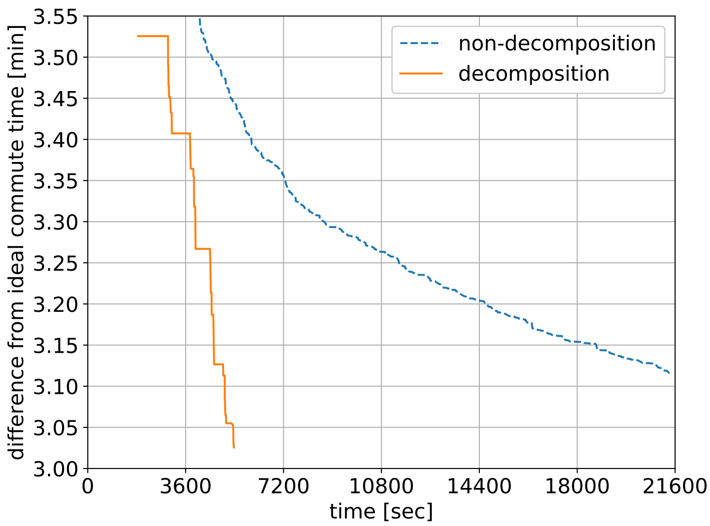


図 10: 5 路線における計算時間と評価値との関係 ($T = 90$)

路線	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
路線1（駅1→駅4）	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
路線1（駅4→駅1）	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
路線2（駅5→駅15）	0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0
路線2（駅15→駅5）	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
路線3（駅2→駅15）	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	1
路線3（駅15→駅2）	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0
路線4（駅1→駅5）	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0
路線4（駅5→駅1）	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0
路線5（駅29→駅12）	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0
路線5（駅12→駅29）	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0

路線	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
路線1（駅1→駅4）	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0
路線1（駅4→駅1）	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
路線2（駅5→駅15）	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0
路線2（駅15→駅5）	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
路線3（駅2→駅15）	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	0
路線3（駅15→駅2）	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	1	0
路線4（駅1→駅5）	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
路線4（駅5→駅1）	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
路線5（駅29→駅12）	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
路線5（駅12→駅29）	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0

路線	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
路線1（駅1→駅4）	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
路線1（駅4→駅1）	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
路線2（駅5→駅15）	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
路線2（駅15→駅5）	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0
路線3（駅2→駅15）	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0
路線3（駅15→駅2）	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	1
路線4（駅1→駅5）	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0
路線4（駅5→駅1）	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
路線5（駅29→駅12）	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
路線5（駅12→駅29）	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0

図 11: 導出した列車ダイヤ

表 7: 駅毎の総利用者数及び平均待ち時間

駅番号	人数 [人]	待ち時間 [分]	駅番号	人数 [人]	待ち時間 [分]	駅番号	人数 [人]	待ち時間 [分]
1	14745	2.24	20	968	1.93	39	118	2.47
2	9291	2.02	21	1076	1.87	40	512	2.81
3	13073	2.00	22	850	1.96	41	393	2.89
4	32087	2.18	23	1457	1.87	42	440	3.23
5	14997	2.23	24	4113	1.79	43	81	3.01
6	789	2.84	25	3941	1.93	44	213	2.70
7	2431	2.73	26	376	2.56	45	688	2.77
8	871	2.88	27	561	2.80	46	373	2.81
9	1781	2.69	28	773	2.80	47	1720	2.67
10	1697	2.84	29	8395	1.57	48	394	2.55
11	455	2.70	30	854	2.57	49	548	2.55
12	4951	2.26	31	542	2.57	50	333	2.68
13	2185	2.79	32	447	2.62	51	396	2.78
14	734	2.74	33	2377	2.79	52	376	2.60
15	19022	0.36	34	598	2.81	53	147	2.73
16	688	2.67	35	607	2.81	54	255	2.85
17	893	2.09	36	1682	2.98	55	324	2.66
18	1484	1.88	37	326	2.85	56	454	2.81
19	1883	2.01	38	2209	3.11	57	269	2.73

5.3.3. 列車ダイヤに関する考察

路線網分解手法が導出した列車ダイヤを図 11 に，列車ダイヤに基づいて移動を行うときの各駅の総利用者数及び平均待ち時間を表 7 に示す．また，総利用者数と平均待ち時間の関係を表す散布図を図 12 に示す．表 7 に示す総利用者は，その駅を出発駅とする移動者だけでなく，移動途中の乗換駅として利用する場合も含む数である．図 11 より，各路線で概ね均等な運行間隔となっている．これは，計画対象期間の始めの期間と，計画対象期間以後を除く期間では移動需要に時間的な偏りが無いことから，妥当な結果であるといえる．表 7，図 12 より，利用者数が 5,000 人以上の駅では平均待ち時間は 2.5 分以下と，利用者数が 5,000 人未満の駅と比較して平均待ち時間が短い傾向にある．これは，効率的な移動が実現していることを表す．

また，駅 15 の平均待ち時間は 0.36 分と他の駅と比較して短い．駅 15 は路線 2 と路線 3 を接続する駅であり，駅 15 を出発する総移動者が 2,026 人であるのに対し，駅 15 の総利用者数は 19,022 人と，約 9.4 倍である．そのため，路線 2 と路線 3 の接続が良いと，駅 15 の利用者の大半を占める乗換客の待ち時間が短くなり，待ち時間の低減に繋がる．実際に，路線 3（駅 2 →駅 15）から路線 2（駅 15 →駅 5）への接続状況を確認する．路線 3（駅 2 →駅 15）の始めの 5 本の列車は，$t = 3, 7, 11, 13, 16$ に始発駅を出発する．路線 3 は始発から終点までに 33 分掛かるため，駅 15 に到着する時刻は $t = 36, 40, 44, 46, 49$ となる．これに対し，路線 2（駅 15 →駅 5）では，$t = 36, 40, 44, 49$ において始発駅を列車が出発している．つまり，路線 3（駅 2 →駅 15）の始めの 5 本の列車のうち，$t = 46$ を除く 4 本の列車に乗車した移動者は，待ち時間 0 分で路線 2（駅 15 →駅 5）に乗換可能となる．このことからも，路線間の乗換がスムーズに可能な，質の高い列車ダイヤであるといえる．

5.3.4. 路線網分解手法における課題

最後に，路線網分解手法の課題と，その解決策を示す．本研究では，離散変数を含む元問題を部分問題に分解することで計算量の低減を目指していた．一方，連続変数のみから成る問題については，離散変数を含む問題と比較して短時間で解けることから，特に計算量の

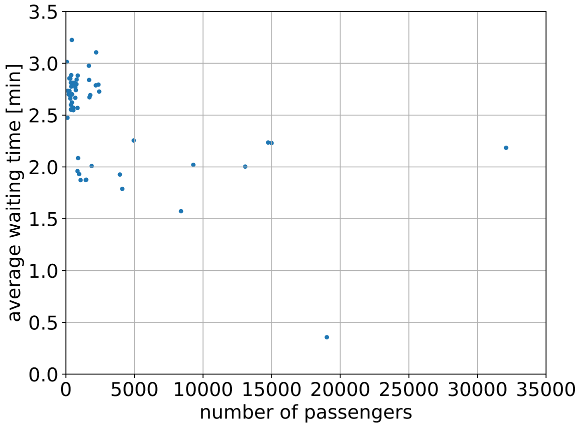


図 12: 総利用者数と平均待ち時間との関係

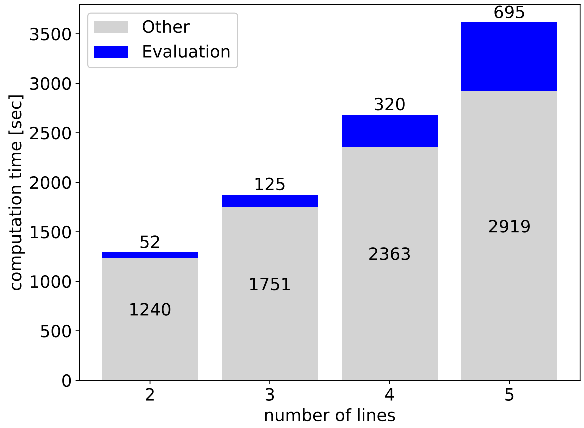


図 13: 路線数と移動計画決定に掛かる時間との関係

削減を考慮していなかった．具体的には，アルゴリズムの Step2（図 1 参照）において，列車ダイヤを固定して移動計画を決定する際，全路線に跨る大規模な線形計画問題を解く必要がある．ところが，提案手法の計算時間を確認したところ，路線数が増加するにつれて移動計画を決定するのに掛かる時間が，全体の計算時間に対して無視できない長さとなることが判明した．図 13 に，移動計画決定に掛かる時間と全体の計算時間との関係を示す．横軸は路線数であり，縦軸は各路線数に対応する全組合せの計算時間を平均したものである．「Evaluation」が移動計画の決定,「Other」がその他の処理に要する計算時間を示す．移動計画を決定するのに掛かる時間が，2 路線では 1,240 秒中 52 秒と全体の約 4%を占めるだけであったが，5 路線では 3,614 秒中 695 秒と全体の約 19%を占めている．これは，全体の計算時間からは無視できない割合であり，路線数の増加に伴いその割合は増加する．

本研究で目標とする，数十路線，数百駅規模の路線網を計画対象としたとき，列車ダイヤの導出は並列処理が可能であるため，十分に並列化できれば，数十路線であっても単一路線の列車ダイヤを導出する時間と変わらない時間で計算を行うことが可能である．一方で，移動計画の決定は並列処理ができない．そのため，移動計画の決定がボトルネックとなり，実時間での求解が実現できない．そのため，大規模な路線網に対して移動計画の決定を短時間で行うことが課題となる．この課題を解決する方法として，最適化問題を解くのではなく，移動者の移動をシミュレーションし，それを移動計画とみなすことが考えられる．藤原ら [3] は数千万人規模の旅客の移動を推定するシミュレータを提案しており，この手法を応用することで，短時間でのシミュレーションが実現できる可能性がある．

6. おわりに

本研究では，移動需要に基づく鉄道の列車ダイヤ自動作成に向けて，計算量を路線数の線形オーダに抑制することを目標として，アルゴリズムを提案した．提案手法では，複数の路線に跨る移動需要を単一路線の移動需要へと置き換えることで，解くべき最適化問題の規模を低減させ，高速化を実現する．また，解の精度低下を防止するための反復最適化手法を組み合わせることで，速度性能と解の精度維持を両立する．数値実験により提案手法を評価した結果，5 路線 57 駅規模の路線網に対して，1 時間以内に高精度の解を導出し，その有効性を確認した．その一方で，提案手法における，各反復で導出した列車ダイヤに従って移動

者が移動を行う際の移動計画を求める処理に掛かる計算時間が，路線数が増加するにつれて支配的となり，実用的な時間内に列車ダイヤが立案できなくなるという課題が判明した．今後は，この課題を解決し，数十路線，数百駅規模の列車ダイヤを短時間で立案するための検討，評価を実施していく．

参考文献

[1] E. Barrena, D. Canca, L.C. Coelho, and G. Laporte: Single-line rail rapid transit timetabling under dynamic passenger demand. *Transportation Research Part B*, **70** (2014), 134–150.

[2] D. Bertsimas, Y.S. Ng, and J. Yan: Joint frequency-setting and pricing optimization on multimodal transit networks at scale, *Transportation Science*, **54-3** (2020), 839–853.

[3] 藤原正康, 加藤学, 伏木匠: 鉄道向け旅客流動シミュレータの開発. 平成 25 年電気学会全国大会講演論文集, (2013), 123–124.

[4] Gurobi Optimization Inc: Gurobi optimizer reference manual. URL: `https://www.gurobi.com/`（アクセス日：2019 年 9 月 4 日）.

[5] 今田京介, 富井規雄: 旅行時間総和を最小にする混合整数計画法による計画ダイヤ作成アルゴリズム. 第 23 回鉄道技術連合シンポジウム講演論文集, (2016), 471–474.

[6] L.N. Jansen, M.B. Pedersen, and O.A. Nielsen: Minimizing passenger transfer times in public transport timetables. *Proceedings of the 7th Conference of the Hong Kong Society for Transportation Studies: Transportation in the Information Age*, (2002), 229–239.

[7] JR 東日本: 各駅の乗車人員. URL: `https://www.jreast.co.jp/passenger/`（アクセス日：2019 年 10 月 17 日）.

[8] L. Kang, X. Zhu, H. Sun, J. Puchinger, M. Ruthmair, and B. Hu: Modeling the first train timetabling problem with minimal missed trains and synchronization time differences in subway networks. *Transportation Research Part B: Methodological*, **93** (2016), 17–36.

[9] 国土交通省: 利用時刻別にみた鉄道利用者数の推計. URL: `https://www.mlit.go.jp/common/001001547.pdf`（アクセス日：2019 年 10 月 17 日）.

[10] 国土交通省: 地域公共交通の活性化・再生への事例集. URL: `https://www.mlit.go.jp/sogoseisaku/transport/htm/all.html`（アクセス日：2020 年 3 月 23 日）.

[11] 國松武敏, 平井力, 富井規雄: 利用者デマンドを反映した列車ダイヤ作成アルゴリズム. 電気学会論文誌 D, **129-1** (2009), 10–20.

[12] K. Nachtigall and S. Voget: A genetic algorithm approach to periodic railway synchronization. *Computers Operations Research*, **23-5** (1996), 453–463.

[13] NAVITIME: 電車混雑回避ナビゲーション. URL: `https://www.navitime.co.jp/lp/predict_congestion/`（アクセス日：2019 年 9 月 6 日）.

[14] H. Niu, X. Tian, and X. Zhou: Demand-driven train schedule synchronization for high-speed rail lines. *IEEE Transactions on Intelligent Transportation Systems*, **16-5** (2015), 2642–2652.

[15] 小田急電鉄: 鉄道部門：1 日平均駅別乗降人員. `https://www.odakyu.jp/company/railroad/users/`（アクセス日：2019 年 10 月 17 日）.

[16] 坂口隆, 佐藤圭介, 加藤怜, 福村直昇: 日別需要に基づく鉄道輸送計画作成手法の開発. 鉄道総研報告, **24-10** (2010), 5–10.

[17] Y. Shen, G. Ren, and Y Liu: Timetable design for minimizing passenger travel time and congestion for a single metro line. *Promet - Traffic and Transportation*, **30-1** (2018), 21–33.

[18] 柴田秀哉, 松永龍弥: 複数系に跨るスケジューリング問題を対象とした部分問題分解に基づく最適化手法. オペレーションズ・リサーチ学会 2020 年春季研究発表会アブストラクト集, (2020), 1-B-4.

[19] X. Sun, S. Zhang, H. Dong, Y. Chen, and H. Zhu: Optimization of metro train schedules with a dwell time model using the lagrangian duality theory. *IEEE Transactions of Intelligent Transportation Systems*, **16-3** (2015), 1285–1293.

[20] （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（株式会社エヌ・ティー・エス, 2005）.

[21] R.C.W. Wong, T.W.Y. Yuen, K.W. Fung, and J.M.Y. Leung: Optimizing timetable synchronization for rail mass transit. *Transportation Science*, **42-1** (2008), 57–69.

[22] Y. Wu and J. Tang: Optimizing timetable synchronization for regional public transit with minimum transfer waiting times. *Proceedings of 24th Chinese Control and Decision Conference*, (2012), 3782–3785.

[23] 山口恵, 柳生大輔, 狩集美智, 國母政仁, 大塚理恵子, 木村恵二: ダイナミックヘッドウェイが実現する鉄道の未来像. 日立評論, **100-5** (2018), 540–544.

[24] L. Yang, K. Li, and Z. Gao: Train timetable problem on a single-line railway with fuzzy passenger demand. *IEEE Transactions on Fuzzy Systems*, **17-3** (2009), 617–629.

[25] S. Yang, K. Yang, Z. Gao, L. Yang, and J. Yang: Last-train timetabling under transfer demand uncertainty: mean-variance model and heuristic solution. *Journal of Advanced Transportation*, **2017-3** (2017), 1–13.

[26] 横浜市: 横浜市統計書　第 9 章　道路、運輸及び通信. `https://www.city.yokohama.lg.jp/city-info/yokohamashi/tokei-chosa/portal/tokeisho/09.html`（アクセス日：2019 年 10 月 17 日）.

[27] J. Zhong, M. Shen, J. Zhang, H.S. Chung, Y. Shi, and Y. Li: A differental evolution algorithm with dual populations for solving periodic railway timetable scheduling problem. *IEEE Transactions on Evolutionalry Computation*, **17-4** (2013), 512–527.

柴田秀哉
三菱電機株式会社
情報技術総合研究所
〒247-8501 神奈川県鎌倉市大船 5-1-1
E-mail:
Shibata.Hideya@cb.MitsubishiElectric.co.jp

ABSTRACT

A DECOMPOSITION-BASED ALGORITHM FOR TRAIN TIMETABLING PROBLEM WITH DYNAMIC PASSENGERS' DEMANDS

Ryuya Matsunaga Hideya Shibata
Mitsubishi Electric Corporation

In many current railway systems, trains are operated under the fixed timetable. Dynamic planning of train timetable with demands can make passengers more comfortable. There are many researches for demand-driven train timetabling by solving optimization problem and efforts for practical use have begun for small railway networks. However, it is difficult to plan train timetable on-demand for large scale networks in urban area, because it takes too much time to solve train timetabling problem (TTP). In this paper, we propose an algorithm which aims to solve TTP in $O(L)$ time where L is number of lines. Our method decomposes the original problem into several single line problems by converting a multiple transfer demand to several ones with no transfer. This makes the problem size small and the computation time short. Also, our method updates the solution iteratively in order to prevent declining the solution quality. We evaluate our method by comparing with a local search method. The result shows that our method can compute the solution more quickly and accurately than local search for large scale networks. On the other hand, the result reveals the problem that the computation time of updating the solution is larger than $O(L)$ and becomes a bottleneck for large scale networks. Future research is hence needed to resolve this task to solve TTP for large scale network in short time.