

過疎地におけるバス時刻最適化問題

鳥取大学大学院・工学研究科 大本 高志 (Takashi Oomoto)
小柳 淳二 (Junji Koyanagi), 河合 一 (Hajime Kawai)
Graduate School of Engineering,
Tottori University

1 序論

過疎地におけるバス事業者の経営は赤字であることが多く、県や市町から補助金を受けている場合がほとんどである。近年、補助金が縮小傾向にあることから赤字が深刻化し、路線廃止を余儀なくされる場合もある。この問題を解決すべく、バス路線の抜本的な見直しが必要とされている。

本研究におけるバス時刻最適化問題は、コスト制約条件の下、利用者の時間ペナルティの総和が最小となるバス運行時刻を求める問題である。コストの内容としては、バス本数などを考慮する。次に、利用者の時間ペナルティとは、利用者が要求する経路および時刻を満たさないことに対するペナルティ値である。ペナルティ値は、利用者が要求する経路が存在しない場合に最も高い値が適応され、要求する経路は存在するが要求する時刻がバス運行時刻と合わない場合は、時刻のずれに応じてペナルティ値が適応される。なお、この問題は計算量が多く現実的な時間内に厳密解を求めることが困難であるため、本研究ではこの問題を解く手法としてタブーサーチを使用している。タブーサーチはタブーリストと呼ばれる枠に、今までに発見した改善解を登録するという手法をとっており、そのため最終的な改善解の他にも候補となる解を出力することが可能である。制約として組み込むことが難しい条件がある場合であっても、最終的な解を人の手で選択することにより解決しうると考えタブーサーチを用いた手法を提案する。

本研究において、特に過疎地を対象とした理由は二つあり、一つは入力データとして利用者の要求する経路と時刻が分かることを前提としたモデルを提案するからである。利用者のデータから時間ペナルティ値を決定しているが、これはつまりバス運行時刻の良し悪しを利用者のデータで判断しているということである。よってなるべくむらが無く、かつ多くの利用者データを得る必要があるため、比較的アンケート調査がしやすいと思われる過疎地を対象としている。もう一つは、道路構造の単純性である。本モデルを拡張した問題では、経路探索をする場合に、分かれ道の少ない単純な道路構造をしている場合に適した手法をとっている。複雑な道路構造の地域に関しても探索は可能であるが、そのような条件下では効率的に探索可能であるとは言えないため過疎地を対象としている。ただし、本研究ではバス時刻最適化問題のみを扱うため経路探索に関しての記述は行っていない。

2 問題形式化

本節では、本研究で扱うバス時刻表最適化問題の形式化を行う。ノード集合 $V = \{0, 1, \dots, n\}$ とリンク集合 $E = \{(i, j) | i, j \in N, i \neq j\}$ で与えられる有向グラフ $G = (V, E)$ を考える。ここで、ノードはバス停、リンクはバス停間を繋ぐ道路を表すものとする。次に、バス車両 $M = \{1, 2, \dots, m\}$ はグラフ G 上を走行するものとし、走行する経路を $\sigma = \{\sigma_1, \sigma_2, \dots, \sigma_s\}$ とする。ただし、バス車両本数 m はバス制約本数 U 以下とし、経路 σ に含まれるノードについては、同じノードを二つ以上含むことを許さないものとする。さらに、バス車両 k は走行する経路 σ 、および出発ノードの時刻 t_0 を含むものとする。なお、リンク集合 $(i, j) \in E$ にはノード間の移動時間 $d_{ij} (\geq 0)$ が与えられており、初期ノードの時刻 t_0 に d_{ij} を加えることで次のノードの出発時刻 t_1 を求めることができる。

また、利用者 $L = \{1, 2, \dots, h\}$ には利用者 $g \in L$ の乗降要求 $D_g = (r_g, q_g, w_g)$ が与えられ、 r_g は要求乗車ノード、 q_g は要求降車ノード、 w_g は要求乗車時刻を表す。ここで、利用者 g は要求乗車ノード r_g と要求降車ノード q_g を通るバス車両 k を選択する行動をとる。ただし、バス車両 k が辿る経路 σ は方向性があるため、要求乗車ノード r_g を通った後に要求降車ノード q_g を通るような経路のみを選択するものとする。さらに、利用者 g には要求乗車時刻 w_g に対する時間ペナルティ関数 $p_g(t) (\geq 0)$ を与える。利用者が選択するバス車両 k に対して要求乗車ノード r_g にバス車両が時刻 t で到着したとすると、 $t - w_g$ が 0 に近づくほど小さい時間ペナルティ値を、 $t - w_g$ が $\pm\infty$ に近づくほど大きい時間ペナルティ値を返すような時間ペナルティ関数を設定する。また、利用者が選択するバス車両が複数台存在する場合は、必ず時間ペナルティ値が小さいものを選択する行動をとるものとする。

以上を踏まえた上で、利用者 g が利用するバス車両 k に対する時間ペナルティ関数の集合を $P = p_g^1, p_g^2, \dots, p_g^m$ とすると、バス時刻表最適化問題は以下のように定式化できる。

$$\text{minimize} \quad \sum_{g=0}^h \min\{p_g^1, p_g^2, \dots, p_g^m\} \quad (1)$$

$$\text{subject to} \quad m \leq U \quad (2)$$

ここでは制約条件をバス本数のみとしたが、必要であればある程度容易に条件を追加できる手法をとっている。そのことについては4.2節で説明する。

3 解の出力

3.1 解の導出

バス路線の構成要素として、大きくバス経路とバス時刻表の二つに分けることができると考えられるが、本節ではバス時刻表に着目し、導出法について説明する。2節において、バス車両 k は走行する経路 σ 、および出発ノードの時刻 t_0 を含むことから、利用者の時間ペナルティの総和を最小化したときに選ばれたバス車両における、それぞれの経路とその出発ノード時刻に着目すると、最適化されたバス時刻表を得ることが可能である。また、バス時刻最適化問題を解くにあたり経路群のデータを入力する必要があるが、妥当な経路群のデータを最初に与えることで、入力した経路群に対する改善されたバス時刻表を導出することが可能である。バス経路を探索する必

要がある場合は、入力データとして与える経路群を変えながら前節で示した問題を繰り返し解く必要がある。

3.2 解の表現方法

この節では改善解を探索する手法であるタブーサーチの説明を行う上で必要となる、計算機上での解表現方法を示す。全てのバス車両 k について、走行する経路 σ および出発時刻 t_0 を表現できればバス時刻表を表すことが可能である。さらに、解を導出する過程でタブーサーチを用いていることから、探索が容易に可能となるような解表現をする必要があるため、0 と 1 からなる一次元配列でバス時刻表を表現する。一次元配列について、経路 $\sigma_z (z = 1, 2, \dots, s)$ におけるバス車両を出発させることが可能な時刻群を $T_z = \{T_z^1, T_z^2, \dots\}$ (有限個) とすると、一次元配列のサイズは $\sum_{z=1}^s |T_z|$ 個となる。ここで $|T_z|$ は集合 T_z の要素の個数を表すものとする。したがって、一次元配列を並べて作った 2 進数の各桁は経路の種類と出発時刻を表しており、各桁で 0 はバス車両を出発させないことを表し、1 はバス車両を出発させることを表している。例えば、経路 $\sigma_1, \sigma_2, \sigma_3$ におけるバス車両出発時刻をそれぞれ $t = \{0, 1, 2, 3\}$ とし、経路 σ_1 において時刻 $t = 2$ 、経路 σ_2 において時刻 $t = 1$ 、経路 σ_3 において時刻 $t = 3$ にバス車両を出発させるバス時刻表を上記の方法を用いて表現すると、 $s = \{0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1\}$ となる。このとき解 s の 1 番目の要素から 4 番目の要素にかけては経路 σ_1 について表しており、5 番目から 8 番目にかけては σ_2 について表しており、9 番目から 12 番目は経路 σ_3 について表していることになる。なお、例においては各経路におけるバス車両出発時刻の設定数をそれぞれ $t = \{0, 1, 2, 3\}$ としたが、時刻の設定数は経路ごとに異なってもよい。また、時刻の間隔についても一定でなくてよい。特に例のような時刻の設定数が同一で時刻の間隔が一定の場合においては、時刻の間隔を時刻表の最小時間単位と呼ぶこととする。

4 タブーサーチ

タブーサーチはメタヒューリスティック解法の一つであり、1989 年にフレッド・グローバーにより考案された。メタヒューリスティック解法とは最適解を求めることが困難である問題に対して、最適近似解を求める手法で、タブーサーチの他に遺伝的アルゴリズムや焼きなまし法などがある。以下に一般的なタブーサーチの手法を示す。

1. 初期解を任意で決定し、 s とおく。また s をタブーリストに加える。
2. 解 s に近傍操作を行い作成した近傍解群 $N(s)$ の中から最も良い近傍解 s' を探し、 $s := s'$ とする。また s をタブーリストに加える。このときタブーリストの枠の上限を超えるならば、一番古い解を削除する。
3. 終了条件が満たされるまで 2. を繰り返し行う。

タブーサーチにおいては、近傍操作をどのように行うかによって解の探索効率が異なる。次の節では本研究で用いる近傍操作について記す。

4.1 近傍操作

本研究では効率的に解を求めるため、解の探索状況によって二通りの近傍操作を切り替えて使用している。まずはそれぞれの近傍操作の方法を説明する。

一近傍操作

解 s からハミング距離が $\lambda = 1$ 以内に存在するものを近傍解とする操作である。例えば、 $s = \{0, 0, 0, 0, 0\}$ であるならば、近傍解 s' はハミング距離が $\lambda = 1$ 以内である $s' = \{1, 0, 0, 0, 0\}$, $s' = \{0, 1, 0, 0, 0\}$, $s' = \{0, 0, 1, 0, 0\}$, $s' = \{0, 0, 0, 1, 0\}$, $s' = \{0, 0, 0, 0, 1\}$ の五つとなる。

二近傍操作

解 s からハミング距離が $\lambda = 2$ 以内に存在するものの中で、特に解 s の変数 1 の総数を変化させないものを近傍解とする操作である。例えば、 $s = \{0, 0, 1, 0, 1\}$ であるならば、近傍解 $s' = \{1, 1, 0, 0, 0\}$, $s' = \{1, 0, 1, 0, 0\}$, $s' = \{1, 0, 0, 1, 0\}$, $s' = \{1, 0, 0, 0, 1\}$, $s' = \{0, 1, 1, 0, 0\}$, $s' = \{0, 1, 0, 1, 0\}$, $s' = \{0, 1, 0, 0, 1\}$, $s' = \{0, 0, 1, 1, 0\}$, $s' = \{0, 0, 0, 1, 1\}$ の 9 つとなる。

4.2 本研究モデルにおけるタブーサーチ法

この節では本研究で行っている、探索状況により近傍操作の方法を変更するタブーサーチの方法を説明する。

1. 全桁に変数 0 が入っている一次元配列を初期解 s とおく。
2. 解 s に一近傍操作を行い作成した近傍解群 $N_1(s)$ の中から最も良い近傍解 s' を探す。ここで、解 s と近傍解 s' を比較し、近傍解 s' の方が良い解であるならば $s := s'$ とする。また s をタブーリストに加える。このときタブーリストの枠の上限を超えるならば、一番古い解を削除する。近傍解 s' の方が良い解でないならば、解 s に二近傍操作を行い作成した近傍解群 $N_2(s)$ の中から最も良い近傍解 s'' を探す。解 s と近傍解 s'' を比較し、近傍解 s'' の方が良い解であるならば $s := s''$ とする。また s をタブーリストに加える。このときタブーリストの枠の上限を超えるならば、一番古い解を削除する。解 s と近傍解 s'' を比較し、近傍解 s'' の方が良い解でないならば近傍解群 $N_1(s)$ と $N_2(s)$ の中から最も良い近傍解 s''' を探し、 $s := s'''$ とする。また s をタブーリストに加える。このときタブーリストの枠の上限を超えるならば、一番古い解を削除する。
3. 終了条件が満たされるまで 2. を繰り返す。

近傍解の数はハミング距離 λ に対して指数的に増加するため、一近傍操作に比べ二近傍操作は設定する一次配列解の桁数が多いほど近傍解の数が大幅に増加してしまう。近傍解の数が増えすぎると解の遷移回数が減り、探索効率が落ちると考えられたため、二近傍を多く行う操作は避けたほうがよいと考えられる。しかし、二近傍操作は一近傍操作で探索しきれない範囲を探索するため、厳密解により近い近似解を導出できる可能性が高くなると考えられる。よって、一近傍操作では近傍解の中に改善解を見つけることができなかつた場合のみ二近傍操作をおこなうような近

傍操作を提案する。5節で紹介する実験においても一近傍操作のみをおこない、解を探索する場合と、本研究提案手法である一近傍操作と二近傍操作を併用して解を探索する場合とでは、後者のほうが効率が良いという結果でた。次節ではその実験方法と結果について説明する。なお、一近傍操作と二近傍操作を併用して解を探索する手法を以後、一、二併用近傍探索と呼ぶ。

また、本モデルでは改善解を導出したときに、制約条件を満たしているかのチェックを行っている。例えば、バスの本数制約であれば一次元配列内の1の個数がバス制約本数を上回った場合、解の候補から外すといった操作をおこなっている。バス移動コストや利用者定員等の制約が必要な場合であっても、新たなルーチンは必要となるが十分拡張可能であると考える。

5 数値実験

5.1 提案アルゴリズムによる実験

4.2節で説明した一、二併用近傍探索によるタブーサーチと、一近傍操作のみを行うタブーサーチで得られる利用者の時間ペナルティの総和と計算時間を導出する実験を行う。入力データである利用者の要求乗車ノード、要求降車ノード、および要求乗車時刻はそれぞれ仮想データ230個を用意し、経路 σ は2, 5, 8, 11, 14個と順に増やして実験を行う。続いて、タブーサーチによる解探索は15分が経過した時点で終了とし、その中で発見した最も良い時間ペナルティ値の総和と、その値が出たときの時間を記録する。結果は以下ようになった。

入力経路数	一近傍探索	一、二併用近傍探索
2	41818.0(1.02s)	41818.0(1.02s)
5	40014.0(5.86)	40014.0(5.88)
8	26948.0(96.03s)	26916.0(291.19s)
11	19440.0(342.58s)	19421.0(216.69s)
14	18518.5(539.50s)	18504.5(444.38s)

入力経路数が多くなるほど利用者の要求する経路を満たしやすくなるため、時間ペナルティ値の総和は減少するが、一次元配列数が増加するため計算量が増え、また解の探索範囲が広がるため最適改善解導出に時間がかかるようになる。一近傍探索と一、二併用近傍探索の結果を比較すると、一、二併用近傍探索の方が時間ペナルティ値の総和が低く、最適改善解導出時間が短いことが分かる。したがって、一、二併用近傍探索は一近傍探索より効率よく解を探索できているといえる。

5.2 実験結果による時刻表と時間ペナルティの変化

この節では、仮想入力データを用意し実際にバス時刻表を導出する手順を紹介する。まず、入力データである利用者は通学もしくは通勤者であり朝の6時から9時にかけてバスを利用するものとする。また、現在運行しているバス時刻表においては利用者58名がバスを利用しており、残りの173名は、現在バスを利用していないが、都合がよい時間帯にバスが走行していれば利用する利用者であるとする。したがって、前者の利用者を維持しつつ、後者の利用者も利用するような改善バス時刻表を導出すればよいことになる。次に、入力データである経路 σ についてはノード数119個のグラフ上に含まれる12経路を設定する。ノード間のつながり方や移動時間などの詳

細なデータは量が多くなるため割愛する．次に，時刻表の最小時間単位は5分とし，入力経路において現在運行しているバス時刻表を以下のように設定した．

経路 0	経路 1	経路 2	経路 3	経路 4	経路 5	経路 6	経路 7	経路 8	経路 9	経路 10	経路 11
7:30	7:10	8:15	7:20	7:45	7:20	6:55	7:40	7:30	8:20	8:10	7:30
7:40				8:40	7:45	8:30	8:30				
					8:05						

このときの時間ペナルティ値の総和を計算すると 23481.5 であった．改善バス時刻表においてこの数値よりも低い値が出れば，より良いバス時刻表が導出できたと言えるであろう．最後に，バス制約本数を現在の時刻表で走行しているバス本数と同じ 18 本とし，15 分間タブーサーチを実行した．結果導出されたバス時刻表は以下の通りになった．

経路 0	経路 1	経路 2	経路 3	経路 4	経路 5	経路 6	経路 7	経路 8	経路 9	経路 10	経路 11
6:30	7:15	7:45	6:00	6:35	6:05	6:30	6:20	6:45	6:55	7:20	6:25
			7:05	7:30	7:15	7:05	7:05				
							7:50				

このときの時間ペナルティ値の総和を計算すると 18690.0 であった．改善前の時間ペナルティ値より良い値が出ていることからバス時刻表は改善されたといえる．次のグラフは利用者一人ひとりの時間ペナルティ値をいくつかの区分に分け，利用者数を数えたヒストグラムである．

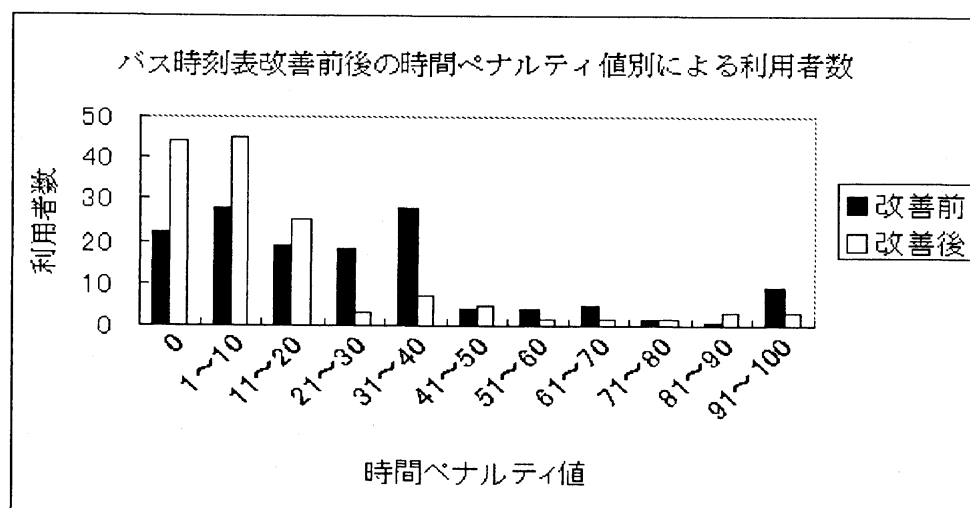


図 1: バス時刻表改善前後の時間ペナルティ値別による利用者数

バス時刻表改善前と比べ改善後は時間ペナルティ値が 0 である利用者が増加していることから，バスを利用する可能性が高い利用者が増えたということがわかる．また，バスを利用する可能性が低いと考えられる時間ペナルティ値が高い利用者についてもあまり増加していないことから，多くの利用者にとって改善前より良いバス時刻表になっている．

6 まとめ

本論文は，現在運営が赤字傾向である過疎地におけるバス事業に着目し，バス運用本数を維持しつつ利用者の効用を上げるという考えでバス時刻表を改善する手法を提案した．計算機上で改

善バス時刻表を導出するため、バス時刻最適化問題の形式化を行い、利用者の需要を時間ペナルティ関数として定義した。改善バス時刻表の導出においてはタブーサーチを用い、効率よく解を探索できる一、二併用近傍操作を提案した。最後に、仮想データを用いた改善バス時刻表の導出手順を示した。

今後の課題として、公共交通の観点からバス運用における制約を理解し、モデルに導入する必要があると考える。制約を増やした結果、除外する近傍解が増えすぎると効率的に解を探索することができなくなる可能性も考えられることから、多くなりすぎるようならば、ある程度ペナルティ関数法を適用する必要があると考える。

謝辞

この研究は日本学術振興会科学研究費補助金、基盤研究 (C)、課題番号 21510145 の補助を受けている。

参考文献

- [1] 橋本英樹, 今堀慎治, 柳浦睦憲, 茨木俊秀. 移動時間コスト関数を考慮した時間枠つき配送計画問題 に対する局所探索法 (数理最適化から見た「凸性の深み, 非凸性の魅惑」). 数理解析研究所講究録, 1349: 94-112, 2004
- [2] 野々部 宏司, 柳浦 睦憲. 局所探索法とその拡張 — タブー探索法を中心として計測と制御, Vol.47, No.6, pp.493-499, 2008
- [3] 柳浦 睦憲. 局所探索法 — 反復改善に基づく最適化の基本戦略オペレーションズ・リサーチ, Vol. 52, No. 9, pp. 538-542, 2007
- [4] 向直人, 馮鈞, 渡邊豊英. 拡張 *R-Tree* を用いたバスの分散管理方式. DEWS2004 7-C-03, 2004
- [5] Yoichiro Shimizu, Hisafumi Kokubugata, Syuichi Matsumoto, Hironao Kawashima. *A Fundamental Study on Operation Plans of On-Demand Buses by using Simulated Annealing*. 土木計画学研究・講演集 Vol.37, 2008.6
- [6] Naoto Mukai, Jun Feng, Toyohide Watanabe. *CVTPR-Tree Approach for Customer Satisfaction on Temporal Vehicle Routing Problem with Time Windows*. DBSJ Letters Vol.2, No.3