

帰り荷と時間枠を考慮した選択型運搬経路問題に対する列生成法

胡 津閣^{*1}, 森山 弘海^{*2}

A Column Generation Algorithm for a Selective Vehicle Routing Problem with Backhauls and Time Windows

by

Jinge HU^{*1} and Hiroumi MORIYAMA^{*2}

(received on Mar. 27, 2020 & accepted on Aug. 3, 2020)

あらまし

運搬経路問題においては、通常、所与である全ての顧客の荷物を運搬するための各運搬車のルートを求める。しかし実務においては、所与の顧客の中から荷物を運搬すべき顧客を選択しなければならない場合も多い。また、実際の運搬現場においては、帰り荷や時間枠を考慮しなければならない場合も存在する。そこで本研究では、帰り荷と時間枠を考慮した選択型運搬経路問題を取り上げ、列生成法に基づくその近似解法を提案する。

Abstract

The classical vehicle routing problem consists of finding a set of routes to serve all given customers. However, in practice, it is often necessary to select some customers from a set of potential customers and to find a set of routes to serve them. Also, there exists practical cases where backhauls and time windows must be taken into account. In this paper, we consider a selective vehicle routing problem with backhauls and time windows and propose a column generation-based algorithm to solve the problem.

キーワード: 選択型運搬経路問題, 帰り荷, 時間枠, 列生成法

Keywords: Selective Vehicle Routing Problem, Backhauls, Time Windows, Column Generation

1. はじめに

運搬経路問題においては、通常、所与である全ての顧客の荷物を運搬するための各運搬車のルートを求める。しかし実務においては、所与である顧客の中から荷物を運搬すべき顧客を選択し、それらの顧客のみの荷物を運搬するための各運搬車のルートを求めなければならない場合も多い。例えば、物流会社においては、荷主からの自社の運搬規模を超える運搬依頼に対して、自社で荷物を運搬する顧客を選択しなければならない状況が頻繁に発生する。また、自社の運搬業務（の一部）を外部委託することを検討している企業においては、外部委託すべき（自社で運搬すべきではない）顧客を選択したいという要望がしばしば現れる。

他方、運搬経路問題に関するこれまでの研究の多くは、(a)デポで荷物の積み込みを行い、それを顧客に運搬して積み下ろしを行う場合（配送の場合）、もしくは、(b)顧客で荷物の積み込みを行い、それをデポに運搬して積み下ろしを行う場合（集荷の場合）

のいずれかを想定している¹⁾²⁾。しかし、実務においては、(a)と(b)が混在している場合も存在する。そして、そのような場合の運搬現場においては、各運搬車が荷物の積み下ろしを行う顧客（以下では単に配送先という）をすべて巡回したあとに、荷物の積み込みを行う顧客（以下では単に集荷先という）を巡回する運搬方法がしばしば用いられている³⁾⁴⁾。それゆえ、各運搬車が配送先をすべて巡回したあとに集荷先を巡回することを特に帰り荷とよぶと、この帰り荷を考慮することも実務上重要であると思われる。加えて、そのような実際の運搬現場においては、各顧客の荷物を運搬する時刻が所与の時間枠（何時から何時までの間）に入っていないなければならないという条件を考慮しなければならない場合⁵⁾も存在する。

そこで本研究では、顧客選択、帰り荷および時間枠を考慮した上で、応用範囲が広いと考えられる運搬収益から運搬費用を差し引いた利益の最大化を目的関数とした場合の運搬経路問題を特に、帰り荷と時間枠を考慮した選択型運搬経路問題とよび、列生成法に基づくその近似解法を提案する。

荷物を運搬すべき顧客の選択を考慮した運搬経路問題に関しては、これまでいくつかの研究がある（2014年までの研究は文献⁶⁾に、2016年度までの研究は文献⁷⁾にサーベイされている）。例えば、基本的な（各運搬車の最大積載量制約を考慮しない場合の）問題については、Boulyら⁸⁾が遺伝的局所探索法に基づく、Dangら⁹⁾が粒子群最適化に基づく、Lin¹⁰⁾がア

*1 情報通信学研究科情報通信学専攻 修士課程
Graduate School of Information and
Telecommunication Engineering, Course of
Information and Telecommunication Engineering,
Master's Program

*2 情報通信学部経営システム工学科 教授
School of Information and Telecommunication
Engineering, Department of Management Systems
Engineering, Professor

ニーリング法に基づく近似解法を, Keshtkaranら¹¹⁾が分枝価格法に基づく, El-Hajjら¹²⁾が切除平面法に基づく, Bianchessiら¹³⁾が分枝カット法に基づく最適解法を提案している. また, 各運搬車の最大積載量制約を考慮した場合の問題については, Archettiら¹⁴⁾がタブーサーチ法に基づく, Gunawanら¹⁵⁾が反復局所探索法に基づく近似解法を, Archettiら¹⁶⁾が分枝価格法に基づく最適解法を提案している. さらに, 各顧客の時間枠を考慮した場合の問題については, Linら¹⁷⁾がアニーリング法に基づく, Huら¹⁸⁾が局所探索法とアニーリング法に基づく, Cura¹⁹⁾が人工蜂群生法に基づく近似解法を提案している. しかしながら, これらを含む従来研究で取り上げている問題はいずれも帰り荷と時間枠を同時に考慮していない. これに対して当研究では, 帰り荷と時間枠を考慮した選択型運搬経路問題を取り上げる. そして, 列生成法に基づくその近似解法を構築し, それで求まる解の精度を数値実験で検証する.

2. 0-1整数計画問題への定式化

運搬車数を K , 顧客の集合を $N = \{1, 2, \dots, n\}$, ルート (デポを出発したあといくつかの顧客を巡回し, 再びデポに戻る閉路) の集合を $R = \{1, 2, \dots, r\}$ とする. ここで, ルートは通常の積載量制約とともに, 既述の帰り荷と時間枠の制約を満たしているものとする. また, ルート $h \in R$ の運搬費用 (固定費用と移動費用の総和) を c_h , 顧客 $i \in N$ の荷物を運搬した場合の運搬収益を f_i とする. さらに, ルートと顧客の関係を次で定義する.

$$\alpha_{hi} = \begin{cases} 1, & \text{ルート } h \text{ に顧客 } i \text{ が含まれる場合} \\ 0, & \text{その他} \end{cases}, \quad h \in R, i \in N \quad (1)$$

このとき, 変数 x_h を

$$x_h = \begin{cases} 1, & \text{ルート } h \text{ を選択する場合} \\ 0, & \text{その他} \end{cases}, \quad h \in R$$

とすると, 帰り荷と時間枠を考慮した選択型運搬経路問題は次のような 0-1整数計画問題に定式化される.

$$(P) \quad \max. \quad \sum_{h \in R} \left(\sum_{i \in N} f_i \alpha_{hi} - c_h \right) x_h \quad (2)$$

$$\text{s. t.} \quad \sum_{h \in R} \alpha_{hi} x_h \leq 1, \quad i \in N \quad (3)$$

$$\sum_{h \in R} x_h \leq K \quad (4)$$

$$x_h \in \{0, 1\}, \quad h \in R \quad (5)$$

ここで, 式(2)は目的関数で総運搬収益から総運搬費用を差し引いた総利益を最大化することを表す. また, 式(3)は運搬車が各顧客をたかだか 1 度巡回することを規定する. そして, 式(4)は使用する運搬車が所与の運搬車台数以下であることを規定する.

なお, 既述した外部委託すべき顧客を選択したいという要望においては, 外部委託費用と自社運搬費

用の総和を最小化することが望まれるが, その最適値は, 顧客 $i \in N$ の荷物を外部委託した場合の費用を e_i とおくと, $f_i = e_i$, $i \in N$ とした場合の (P) の最適値を求め, それを $\sum_{i \in N} e_i$ から差し引くことで定まることを付記しておく.

3. 列生成法に基づく近似解法

上述の (P) を明確に設定するためには, 考えられる全てのルートを列挙する必要があるが, その数は一般に膨大な数になる. そのため, 全てのルートを列挙することは, それ自身が極めて多くの計算量を必要とするため実際的でない. そこでここでは, ルートを逐次列挙しながら (P) の上界値を求める列生成法を提案し, その列生成法で列挙したルートを用いて (P) の下界値 (実行可能解) を求める近似解法を構築する.

いま, (P) の式(5)を $x_h \geq 0$, $h \in R$ に緩和した連続緩和問題:

$$(\bar{P}) \quad \max. \quad \sum_{h \in R} \left(\sum_{i \in N} f_i \alpha_{hi} - c_h \right) x_h \quad (6)$$

$$\text{s. t.} \quad \begin{aligned} & \text{式(3), (4)} \\ & x_h \geq 0, \quad h \in R \end{aligned} \quad (7)$$

を考える. ここで, 問題 (P) の最適値を $z(P)$ と記すと, 緩和法の原理より, $z(P) \leq z(\bar{P})$ である. よって, $(\bar{P})$ を解けば (P) の上界値が求まる. さらに, R のある部分集合 $\bar{R}$ を考え, $(\bar{P})$ におけるルートの集合 R を $\bar{R}$ に限定した限定主問題 $(\bar{P}_{\bar{R}})$ と, その双対問題:

$$(\bar{D}_{\bar{R}}) \quad \min. \quad \sum_{i \in N} u_i + Ku_0 \quad (8)$$

$$\text{s. t.} \quad \sum_{i \in N} \alpha_{hi} u_i + u_0 \geq \sum_{i \in N} f_i \alpha_{hi} - c_h, \quad h \in \bar{R} \quad (9)$$

$$u_i \geq 0, \quad i \in N \quad (10)$$

$$u_0 \geq 0 \quad (11)$$

を考える. ここで, $(\bar{D}_{\bar{R}})$ の u_i , $i \in N$ と u_0 はそれぞれ $(\bar{P}_{\bar{R}})$ の式(3)と式(4)に対応する双対変数であり, 双対定理より $z(\bar{P}_{\bar{R}}) = z(\bar{D}_{\bar{R}})$ である. そこで $(\bar{P}_{\bar{R}})$ の最適解を $(\bar{x}_h)$, $(\bar{D}_{\bar{R}})$ の最適解を $(\bar{u}_i), \bar{u}_0$ とし,

$$c_h^* = c_h + \sum_{i \in N} \alpha_{hi} (\bar{u}_i - f_i) + \bar{u}_0, \quad h \in R \quad (12)$$

とおく. このとき,

$$c^* = \min_{h \in R} c_h^* \quad (13)$$

である c^* を定めると, $c^* \geq 0$ ならば, $(\bar{D}_{\bar{R}})$ の最適解は $(\bar{P})$ の双対問題の最適解となり,

$$x_h^* = \begin{cases} \bar{x}_h, & h \in \bar{R} \\ 0, & \text{その他} \end{cases}, \quad h \in R \quad (14)$$

である (x_h^*) は $(\bar{P})$ の最適解となる.

以上の準備の下, 当研究では, 概略, 次のような手順で (P) の実行可能解を求めるものとする. ただし, 次の手順の S0, S2, S4 を実行する方法については後述する.

手順 列生成法に基づく近似解法

- S0. 初期ルートを生成し、それらの集合を $\bar{R}$ とおく。
 S1. $(\bar{P}_R)$ の最適解 $(\bar{x}_h)$ と $(\bar{D}_R)$ の最適解 $(\bar{u}_i)$, $\bar{u}_0$ を求める。
 S2. $c^* \geq 0$ ならば S4へ。
 S3. $c_h^* < 0$ であるルートの集合 $R_{\bar{u}\bar{u}_0}$ を求め、 $\bar{R} := \bar{R} \cup R_{\bar{u}\bar{u}_0}$ として S1へ。
 S4. $(\bar{P}_R)$ に $x_h \in \{0, 1\}, h \in \bar{R}$ を追加した問題 $(P_{\bar{R}})$ を解いて (P) の実行可能解 (x_h^*) を求める。

この手順の S0 においては、初期ルートを求める必要があるが、ここでは、 $(0 \rightarrow i \rightarrow 0), i \in N$ である n 個のルートを初期ルートとする。また、S4 においては $(P_{\bar{R}})$ の最適解を数値計画ソルバーで求めるものとする。さらに、この手順の S2 においては、式(13)の c^* を求める必要があるが、そのための問題（価格付け問題という）については次節で確認する。

4. 価格付け問題の定式化

ここでは、前述した価格付け問題（式(13)の c^* を求める問題）を混合 0-1 整数計画問題に定式化する。

配送先の集合を $L = \{1, 2, \dots, l\}$, 集荷先の集合を $M = \{l+1, l+2, \dots, l+m\}$, 配送先と集荷先を併せた顧客の集合を $N = L \cup M = \{1, 2, \dots, n\}$ とする。ただし、 $n = l+m$ である。また、デポ（運搬車の発地）を 0, ダミーのデポ（運搬車の着地）を $n+1$ で表し、 N に 0 を加えた集合を $N_0 = N \cup \{0\}$ とし、 N に $n+1$ を加えた集合を $N_{n+1} = N \cup \{n+1\}$ とする。さらに、

- a_0 : デポ 0 における運搬車の最早出発時刻
 b_0 : デポ 0 における運搬車の最遅到着時刻
 a_{n+1} : デポ 0 における運搬車の最早到着時刻
 b_{n+1} : デポ 0 における運搬車の最遅到着時刻
 a_i : 顧客 $i \in N$ の最早サービス開始時刻
 b_i : 顧客 $i \in N$ の最遅サービス開始時刻
 q_i : 顧客 $i \in N$ の需要量（配送量と集荷量）
 Q : 運搬車の最大積載量
 c_{ij} : 顧客（デポを含む） $i \in N_0$ から顧客（デポを含む） $j \in N_0$ への移動費用
 c_0 : 運搬車の固定費用
 t_{ij} : 顧客（デポを含む） $i \in N_0$ から顧客（デポを含む） $j \in N_0$ への移動時間
 s_i : 顧客 $i \in N$ のサービス時間

とする。加えて、 $A_1 = \{(0, j) \mid j \in N\}$, $A_2 = \{(i, j) \in N \times N \mid i \neq j\}$, $A_3 = \{(i, n+1) \mid i \in N\}$, $A = A_1 \cup A_2 \cup A_3$ とし、

$$\bar{c}_{ij} = \begin{cases} c_{ij} + \bar{u}_0/2 + (\bar{u}_j - \bar{f}_j)/2, & (0, j) \in A_1 \\ c_{ij} + (\bar{u}_i - \bar{f}_i)/2 + (\bar{u}_j - \bar{f}_j)/2, & (i, j) \in A_2 \\ c_{ij} + (\bar{u}_i - \bar{f}_i)/2 + \bar{u}_0/2, & (i, n+1) \in A_3 \end{cases}, \quad (i, j) \in A \quad (15)$$

$$d_{ij} = \begin{cases} t_{0j}, & (0, j) \in A_1 \\ s_i + t_{ij}, & (i, j) \in A_2 \\ s_i + t_{i0}, & (i, n+1) \in A_3 \end{cases}, \quad (i, j) \in A \quad (16)$$

とする。このとき、変数 x_{ij} と t_i を

$$x_{ij} = \begin{cases} 1, & \text{運搬車が顧客（デポを含む）} i \text{ から顧客（デポを含む）} j \text{ へ直接移動する場合,} \\ 0, & \text{その他} \end{cases} \quad (i, j) \in A$$

- t_0 : デポ 0 からの運搬車の出発時刻
 t_i : 顧客 $i \in N$ におけるサービス開始時刻
 t_{n+1} : デポ 0 への運搬車の到着時刻

と定義すると、前述した価格付け問題は、次のような混合 0-1 整数計画問題に定式化される。ただし、 B は十分に大きな数である。

$$(S_{\bar{u}\bar{u}_0}) \quad \min. \sum_{(i,j) \in A} \bar{c}_{ij} x_{ij} + c_0 \quad (17)$$

$$\text{s. t.} \quad \sum_{j \in N} x_{0j} = 1 \quad (18)$$

$$\sum_{j \in N_0 \setminus \{i\}} x_{ji} = \sum_{j \in N_{n+1} \setminus \{i\}} x_{ij}, \quad i \in N \quad (19)$$

$$\sum_{i \in N} x_{i, n+1} = 1 \quad (20)$$

$$\sum_{i \in L} \sum_{j \in M} x_{ij} \leq 1 \quad (21)$$

$$\sum_{i \in M} \sum_{j \in L} x_{ij} = 0 \quad (22)$$

$$\sum_{i \in N_0 \setminus \{j\}} \sum_{j \in L} q_j x_{ij} \leq Q \quad (23)$$

$$\sum_{i \in M} \sum_{j \in N_{n+1} \setminus \{i\}} q_i x_{ij} \leq Q \quad (24)$$

$$t_i + d_{ij} - B(1 - x_{ij}) \leq t_j, \quad (i, j) \in A \quad (25)$$

$$a_i \leq t_i \leq b_i, \quad i \in N \cup \{0, n+1\} \quad (26)$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A \quad (27)$$

ここで、式(17)は目的関数で最適値が式(13)の c^* になることを表す。そして式(18), (19), (20)はルートが生成されることを規定する。また、式(21), (22)は、ルートが (a) 配送先だけを巡回する, (b) 集荷先だけを巡回する, (c) 配送先と集荷先を巡回する場合に配送先をすべて巡回したあとに集荷先を巡回する, のいずれかになることを規定する。また、式(23)は運搬車が巡回する配送先の需要量の総和が最大積載量以下であることを、式(24)は運搬車が巡回する集荷先の需要量の総和が最大積載量以下であることを規定する。そして式(25), (26)は、デポ 0 における運搬車の出発時刻と到着時刻および巡回する顧客におけるサービス開始時刻が所与の時間枠を満たすことを規定する。

当研究では、この $(S_{\bar{u}\bar{u}_0})$ を数値計画ソルバーで解いて c^* を求めるが、式(25)のような十分に大きな数 B を含んだ定式化は一般に多くの計算時間を必要とするため、 B の値をできる限り小さい値に変更することで式(25)を強化することを考える。

式(25)は $x_{ij} = 0$ であるとき、 $t_i + d_{ij} - t_j \leq B, (i, j) \in A$ となるが、式(26)より $t_i \leq b_i$ であり $t_j \geq a_j$ であるため、式(25)の B の値を $b_i + d_{ij} - a_j$ に設定しても、式(25)はすべての実行可能解に対して成立する。よって、式(25)は $B > 0$ である場合のみ意味があることに注意すると、式(25)の B を $\max\{0, b_i + d_{ij} - a_j\}$ で置き換えることができる。さらに、式(25)の $(i, j) \in A_2$ に対応した制約式、式(26)の $i \in N_{n+1}$ に対応した下限制約式、式(26)の $i \in N_0$ に対応した上限制約式は、持ち上げ (lifting) 操作²⁰⁾を適用することにより、次のような強化した制約式にそれぞれ書き換えることができる。ただし、 $N_i^- = \{j \in N \mid (j, i) \in A\}$, $N_i^+ = \{j \in N \mid (i, j) \in A\}$ である。

$$t_i + d_{ij} - \max\{0, b_i + d_{ij} - a_j\}(1 - x_{ij}) + \max\{0, \min\{a_j - a_i, -d_{ji}\} + b_i - a_j\}x_{ji} \leq t_j, \quad (i, j) \in A_2 \quad (28)$$

$$a_i + \sum_{j \in N_i^-} \max\{0, a_j + d_{ji} - a_i\}x_{ji} \leq t_i, \quad i \in N_{n+1} \quad (29)$$

$$t_i \leq b_i - \sum_{j \in N_i^+} \max\{0, b_i - b_j + d_{ij}\}x_{ij}, \quad i \in N_0 \quad (30)$$

そこで当研究では、以上のように強化された問題 ($S_{\bar{u}u_0}$) を数理計画ソルバーで解いて c^* を求めるが、列生成法に基づく近似解法の S2 においては、常に c^* を求める必要はなく、 $c_h^* < 0, \exists h \in R$ であるルート h が求まれば S3 へ計算を進めることができる。また、S3 においても、常に $c_h^* < 0$ である全てのルートの集合を求める必要はなく、 $c_h^* < 0, \exists h \in R$ であるルート h を $R_{\bar{u}u_0} = \{h\}$ とすることができる。そこでここでは、($S_{\bar{u}u_0}$) の最適解を常に求めるのではなく、目的関数値が負の実行可能解が求まった時点で ($S_{\bar{u}u_0}$) の計算を終了するものとする。

5. 数値実験による検証

以上、帰り荷と時間枠を考慮した選択型運搬経路問題に対する列生成法を提案したが、その有効性を検証するために数値実験を実施した。ただし、数値実験で使用した実験データの作成方法と実験環境に関しては以下の通りである。

(1) 通常的时间枠付き運搬経路問題に対する代表的なベンチマーク問題²¹⁾を基に、帰り荷と時間枠を考慮した選択型運搬経路問題の問題例を作成した。使用したベンチマーク問題に関しては以下の通りである。

- 顧客数 n は $n = 100$ である。
- 顧客（デポを含む） $i \in N_0$ から顧客（デポを含む） $j \in N_0$ への移動費用 c_{ij} と移動時間 t_{ij} は、顧客（デポを含む） $i \in N_0$ の2次元平面上の座標 (X_i, Y_i) を与え、次式で定めている。

$$c_{ij} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}, \quad i, j \in N_0 \quad (31)$$

$$t_{ij} = c_{ij}, \quad i, j \in N_0 \quad (32)$$

- 顧客 $i \in N$ の最早サービス開始時刻 a_i と最遅サービス開始時刻 b_i は、パラメータ $f (0 < f \leq 1)$ を与えて $|\bar{N}| = nf$ である顧客の部分集合 $\bar{N} (\subseteq N)$ をランダムに選択するとともに、デポ 0 における運搬車の最早出発（到着）時刻 $a_0 (= a_{n+1})$ と最遅出発（到着）時刻 $b_0 (= b_{n+1})$ とを与え、次式で定めている。

$$a_i = \begin{cases} U[a_0 + t_{0i}, b_0 - t_{i0} - s_i] - h_i, & i \in \bar{N} \\ 0, & \text{その他} \end{cases}, \quad i \in N \quad (33)$$

$$b_i = \begin{cases} U[a_0 + t_{0i}, b_0 - t_{i0} - s_i] + h_i, & i \in \bar{N} \\ b_0 - t_{i0} - s_i, & \text{その他} \end{cases}, \quad i \in N \quad (34)$$

ただし、 $U[a_0 + t_{0i}, b_0 - t_{i0} - s_i]$ は $[a_0 + t_{0i}, b_0 - t_{i0} - s_i]$ の一様整数乱数であり、 h_i はある正

Table 1 Characteristics of the benchmark problems

Type	b_0	$\max q_i$	$\sum q_i$	Q	s
R1	1236	50	1458	200	10
R2	3390	50	1458	1000	10
C1	230	41	1810	200	90
C2	1000	41	1810	700	90
RC1	240	40	1724	200	10
RC2	960	40	1724	1000	10

規分布（文献²¹⁾ではこの正規分布の平均と分散を明示していない）で定めた値である。

- デポ 0 における運搬車の最早出発（到着）時刻 $a_0 (= a_{n+1})$ は 0 である。
- サービス時間 $s_i, i \in N$ はすべて等しく、 $s_i = s, i \in N$ である s を定めている。
- ベンチマーク問題は Table 1 の 6 タイプに分類されている。ただし、Table 1 の $\max q_i$ は $\max_{i \in N} q_i$ であり、 $\sum q_i$ は $\sum_{i \in N} q_i$ である。

(2) 運搬車数 K は $K = 5, 10$ の 2 通りとした。

(3) 上記ベンチマーク問題の最初の $l = n/2$ 個の顧客を配送先、残りの $m = n - l$ 個の顧客を集荷先とし、最大積載量 Q は $Q/2$ とした。

(4) 顧客 $i \in N$ の運搬収益 f_i は次式で定めた。

$$f_i = 0.1c_{0i}q_i, \quad i \in N \quad (35)$$

(5) 運搬車の固定費用 c_0 は $c_0 = 10$ とした。

(6) 使用計算機は Intel® Core™ i7-7700T CPU 2.90GHz（4-core）の CPU と 16GB のメモリを搭載した PC（OS：Windows 10 professional 64bit）で、使用言語は C#（処理系：Microsoft Visual Studio Community 2017）である。

(7) 列生成法に基づく近似解法に使用した数理計画ソルバーは Gurobi Optimizer Version 8.1.0²²⁾ である。

(8) 列生成法（手順の S0～S3）の計算は、S3 の終了時点の計算時間が 1800 秒以上となれば打ち切りとした。

以上の実験データと実験環境の下で実施した数値実験の結果をベンチマーク問題の 6 タイプごとにそれぞれ Table 2～7 に示す。ただし、Table 2～7 に関しては以下の通りである。

- Table 2～7 の **Name** はベンチマーク問題の問題名である。
- Table 2～7 の f は上述した顧客 $i \in N$ の最早サービス開始時刻 a_i と最遅サービス開始時刻 b_i を定める場合のパラメータで $0 < f \leq 1$ である。
- Table 2～7 の z_L と z_U はそれぞれ求めた下界値と上界値である。ただし、「-」は計算打ち切りまでに上界値が求まらなかった場合である。
- Table 2～7 の n_c は求めた実行可能解の運搬顧客数である。
- Table 2～7 の **Cols** は生成した列数である。
- Table 2～7 の **time**⁽¹⁾ は列生成法（手順の S0～S3）の計算時間（単位は秒）で、**time**⁽²⁾ は列生成法に基づく近似解法（手順の S0～S4）の計算時間（単

位は秒)である。

また、上界値が求まった問題例については、相対誤差 $\varepsilon = (z_U - z_L)/z_U$ を Table 8 に示す。

この数値実験では、全ての問題例で実行可能解（下界値）を求めることができたが、多くの問題例においてその精度を（計算打ち切りまでに上界値が求まらなかったため）検証できなかった。しかしながら、上界値が求まった問題例においては、相対誤差 0.035 以下の解が求まっており（Table 8 参照）、提案法を用いて高い精度の実行可能解が求まることを検証できた。

また、この数値実験の結果は、 f が 1.00 より小さい全ての問題例と、タイプが R2, C2, RC2 の多くの問題例は、上界値を求めることができなかったことを示している。この理由としては、 f が小さい問題例ほど時間枠制限のある顧客数が少なくなるとともに、タイプが R2, C2, RC2 の問題例は（R1, C1, RC1 の問題例よりも）時間枠の幅が広くなることから、そのような問題例は、実行可能なルートの組み合わせが増加し、上界値を求めるのに必要な列の求解に時間を要するためと考えられる。さらに、この数値実験の結果は、同じ問題名ならば、 $K = 5$ よりも $K = 10$ の

Table 2 Computational results (Type R1)

Name	f	$K = 5$						$K = 10$					
		z_L	z_U	n_C	Cols	$time^{(1)}$	$time^{(2)}$	z_L	z_U	n_C	Cols	$time^{(1)}$	$time^{(2)}$
R101	1.00	1112.760	1112.760	25	131	10.0	10.0	1685.504	1696.128	52	251	56.6	56.7
R102	0.75	1428.724	—	34	244	1822.1	1822.1	1879.747	—	65	482	1821.1	1821.4
R103	0.50	1579.006	—	40	244	2059.7	2059.7	2073.624	—	68	482	1813.9	1814.0
R104	0.25	1727.620	—	45	298	2733.2	2733.2	2118.298	—	73	649	1854.5	1855.5
R105	1.00	1329.330	1376.203	34	201	59.7	59.7	1943.958	1950.706	60	433	136.8	136.9
R106	0.75	1557.884	—	39	270	2138.6	2138.7	2065.480	—	70	572	2011.4	2011.6
R107	0.50	1721.797	—	42	303	1823.6	1823.6	2158.181	—	75	668	1841.4	1841.8
R108	0.25	1751.254	—	43	338	1803.7	1803.7	2234.728	—	78	736	1820.8	1821.6
R109	1.00	1366.204	—	32	225	1975.1	1975.1	1979.373	—	61	505	1959.5	1959.6
R110	1.00	1544.040	—	38	240	2157.2	2157.2	2091.193	—	69	485	1829.0	1829.1
R111	1.00	1637.657	—	38	228	1839.1	1839.1	2028.195	—	68	462	1831.3	1831.5
R112	1.00	1731.566	—	39	262	2036.6	2036.7	2169.861	—	71	585	1802.4	1803.0

Table 3 Computational results (Type R2)

Name	f	$K = 5$						$K = 10$					
		z_L	z_U	n_C	Cols	$time^{(1)}$	$time^{(2)}$	z_L	z_U	n_C	Cols	$time^{(1)}$	$time^{(2)}$
R201	1.00	2143.158	—	78	1282	1800.1	1800.5	2242.014	—	86	842	778.3	778.4
R202	0.75	2137.523	—	73	909	2355.7	2356.9	2260.984	—	79	1020	1915.4	1918.3
R203	0.50	2143.594	—	70	978	1801.5	1803.2	2422.646	—	89	881	1802.2	1802.7
R204	0.25	2359.658	—	75	1182	1801.4	1803.1	2411.985	—	81	1152	1802.2	1805.9
R205	1.00	2220.226	—	75	1088	1804.3	1804.8	2419.569	—	88	1180	1967.2	1967.5
R206	0.75	2232.034	—	67	1227	1800.4	1802.1	2453.697	—	84	1234	1800.9	1802.0
R207	0.50	2198.664	—	68	1289	1800.7	1803.6	2372.486	—	83	995	1858.5	1862.5
R208	0.25	2359.849	—	78	1400	1801.4	1812.1	2506.566	—	80	1336	1801.8	1805.2
R209	1.00	2199.352	—	70	881	1896.2	1896.9	2355.743	—	87	1057	1836.2	1838.0
R210	1.00	2157.304	—	76	926	2043.1	2044.9	2400.948	—	77	974	1807.5	1809.2
R211	1.00	2199.943	—	67	938	1832.3	1835.2	2431.080	—	83	978	1951.9	1955.1

Table 4 Computational results (Type C1)

Name	f	$K = 5$						$K = 10$					
		z_L	z_U	n_C	Cols	$time^{(1)}$	$time^{(2)}$	z_L	z_U	n_C	Cols	$time^{(1)}$	$time^{(2)}$
C101	1.00	2215.401	2245.713	37	249	65.1	65.2	3353.590	3457.379	66	466	125.1	125.2
C102	0.75	2318.126	—	40	268	2328.7	2328.7	3425.226	—	71	654	3513.2	3513.4
C103	0.50	2577.873	—	39	340	2003.9	2003.9	3436.250	—	66	711	1875.7	1876.8
C104	0.25	2569.014	—	44	499	1927.5	1927.6	3517.126	—	78	718	1823.7	1824.2
C105	1.00	2282.907	2365.199	38	285	140.1	140.2	3470.680	3556.030	68	589	346.3	346.5
C106	1.00	2341.559	2393.869	38	407	1642.9	1642.9	3523.252	—	73	695	1822.2	1822.4
C107	1.00	2459.705	2511.391	40	323	318.6	318.7	3552.776	3630.186	69	599	706.5	706.6
C108	1.00	2397.715	—	40	360	1823.8	1823.9	3632.999	—	74	679	1808.7	1808.8
C109	1.00	2482.150	—	39	377	1834.3	1834.4	3438.845	—	71	643	1827.6	1827.8

Table 5 Computational results (Type C2)

		$K = 5$						$K = 10$					
<i>Name</i>	<i>f</i>	z_L	z_U	n_C	<i>Cols</i>	$time^{(1)}$	$time^{(2)}$	z_L	z_U	n_C	<i>Cols</i>	$time^{(1)}$	$time^{(2)}$
C201	1.00	4616.510	4698.352	92	2341	481.5	482.1	4633.582	4732.639	95	2802	491.8	492.8
C202	0.75	4145.106	—	82	1388	1802.0	1802.7	4518.105	—	94	1785	1803.6	1804.7
C203	0.50	3910.374	—	79	1114	1802.1	1802.7	4419.673	—	94	1562	1834.4	1835.0
C204	0.25	3815.727	—	81	1294	1800.2	1800.9	4320.233	—	94	1593	1800.6	1801.9
C205	1.00	4598.378	—	92	1823	1824.7	1825.1	4574.330	—	96	2182	1922.4	1923.0
C206	1.00	4297.118	—	87	1649	1802.5	1802.9	4485.557	—	91	2053	1800.4	1801.0
C207	1.00	4151.891	—	83	1574	1800.2	1800.6	4332.178	—	96	1799	1800.5	1802.1
C208	1.00	4073.884	—	86	1327	1800.7	1801.1	4466.735	—	91	1643	1800.4	1800.9

Table 6 Computational results (Type RC1)

		$K = 5$						$K = 10$					
<i>Name</i>	<i>f</i>	z_L	z_U	n_C	<i>Cols</i>	$time^{(1)}$	$time^{(2)}$	z_L	z_U	n_C	<i>Cols</i>	$time^{(1)}$	$time^{(2)}$
RC101	1.00	2119.297	2121.167	34	224	98.4	98.4	3356.711	3455.795	64	480	253.8	253.9
RC102	0.75	2306.489	—	36	258	2447.1	2447.1	3557.083	—	69	485	1846.7	1846.8
RC103	0.50	2388.170	—	44	337	1940.4	1940.4	3637.563	—	72	558	1806.4	1806.5
RC104	0.25	2252.773	—	38	271	1931.6	1931.6	3769.909	—	68	528	1808.7	1808.8
RC105	1.00	2118.420	—	38	286	1806.9	1806.9	3420.146	—	62	449	2588.5	2588.6
RC106	1.00	2265.120	—	32	305	1946.6	1946.7	3559.426	—	65	530	1808.7	1808.7
RC107	1.00	2393.268	—	37	749	1874.4	1874.4	3756.876	—	63	537	1960.4	1960.5
RC108	1.00	2370.963	—	39	292	1802.1	1802.1	3855.973	—	69	529	2450.6	2450.6

Table 7 Computational results (Type RC2)

		$K = 5$						$K = 10$					
<i>Name</i>	<i>f</i>	z_L	z_U	n_C	<i>Cols</i>	$time^{(1)}$	$time^{(2)}$	z_L	z_U	n_C	<i>Cols</i>	$time^{(1)}$	$time^{(2)}$
RC201	1.00	4238.314	—	82	1783	1814.7	1814.9	4498.474	—	94	1500	1824.4	1824.5
RC202	0.75	3915.139	—	71	1067	1856.0	1856.9	4610.419	—	96	1183	2069.3	2069.4
RC203	0.50	3948.842	—	66	1064	1865.0	1865.6	4316.583	—	90	1121	1827.0	1827.3
RC204	0.25	4263.031	—	82	1126	1803.9	1804.4	4250.040	—	93	1403	1802.4	1803.5
RC205	1.00	4060.455	—	74	1059	1829.2	1829.5	4465.257	—	95	1116	1800.1	1800.3
RC206	1.00	4022.818	—	71	1303	1825.0	1826.8	4575.669	—	95	1356	1835.1	1835.5
RC207	1.00	4047.205	—	77	1181	1941.8	1942.6	4508.778	—	93	1250	1801.8	1802.1
RC208	1.00	4010.867	—	70	855	1802.6	1803.0	4372.123	—	87	1292	1802.3	1802.9

Table 8 Relative errors

<i>Name</i>	K	z_L	z_U	ε
R101	5	1112.760	1112.760	0.000
R101	10	3915.139	3915.139	0.000
R105	5	1329.330	1376.203	0.034
R105	10	1943.958	1950.706	0.003
C101	5	2215.401	2245.713	0.013
C101	10	3353.590	3457.379	0.030
C105	5	2282.907	2365.199	0.035
C105	10	3470.680	3556.030	0.024
C106	5	2341.559	2393.869	0.022
C107	5	2459.705	2511.391	0.021
C107	10	3552.776	3630.186	0.021
C201	5	4616.510	4698.352	0.019
C201	10	4612.626	4732.639	0.021
RC101	5	2119.297	2121.167	0.001
RC101	10	3356.711	3455.795	0.029

問題例の方が（多くの顧客を巡回できる可能性が高まるため当然のことであるが）求まった実行可能解の運

搬顧客数 n_C が増加することを示している。

加えて、この数値実験の結果は、提案法の計算時の大部分が列生成法（手順の S0～S3）の計算時間 $time^{(1)}$ であることを示しているため、計算時間の短縮化のためには、今後、効果的な列の生成法や価格付け問題に対する効率的な解法の開発が必要であることを示唆している。

6. おわりに

当研究では、帰り荷と時間枠を考慮した選択型運搬経路問題を取り上げ、列生成法に基づくその近似解法を提案した。そして、提案した解法の有効性を検証するための数値実験を実施した。数値実験では、提案した解法を用いて全ての問題例で実行可能解を求めることができた。また、上界値が求まった問題例においては、相対誤差0.035以下の解が求まることを確認し、提案法を用いて高い精度の実行可能解が求まることを検証した。

しかしながら，当研究では価格付け問題の求解に数理計画ソルバーを用いていることもあり，多くの問題例で上界値を求めることができなかった．それゆえ，今後，効果的な初期列の生成方法や，価格付け問題に対する効率的な解法を開発し，提案法の一層の高速化を図り，求まる解の精度を検証する必要がある．

謝辞

本研究は JSPS 科研費 JP18K04617 の助成を受けたものです．

参考文献

- 1) G. Laporte, “The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms,” *European Journal of Operational research*, Vol. 59, pp. 345-358, 1992
- 2) P. Toth and D. Vigo, “Models Relaxation and Exact Approaches for the Capacitated Vehicle Routing Problem,” *Discrete Applied Mathematics*, Vol. 123, pp. 345-358, 2002
- 3) M. O. Yano, T. J. Chan, L. K. Richter, T. Cutler, K. G. Murty and D. McGettigan, “Vehicle Routing at Quality Stores,” *Interfaces*, Vol. 17, No. 2, pp. 52-63, 1987
- 4) Ç. Koç, and G. Laporte, “Vehicle Routing with Backhauls: Review and Research Perspectives,” *Computers and Operations Research*, Vol. 91, pp. 79-91, 2018
- 5) M. M. Solomon and J. Desrosiers, “Time Windows Constrained Routing and Scheduling Problems,” *Transportation Science*, Vol. 22, No. 1, pp. 1-13, 1988
- 6) C. Archetti, M. G. Speranza and D. Vigo, “Vehicle Routing Problems with Profits”, In: P. Toth and D. Vigo (Eds.), *Vehicle Routing: Problems, Methods, and Applications*, pp. 273-298, Philadelphia: SIAM, 2014
- 7) A. Gunawan, H. C. Lau and P. Vansteenwegen, “Orienteering Problem: A Survey of Recent Variants, Solution Approaches and Applications,” *European Journal of Operational Research*, Vol. 255, pp. 345-358, 2016
- 8) H. Bouly, D. -C. Dang and A. Moukrim, “A Memetic Algorithm for the Team Orienteering Problem,” *4OR: A Quarterly Journal of Operations Research*, Vol. 8, Issue 1, pp. 49-70, 2010
- 9) D. -C. Dang, R. N. Guibadj and A. Moukrim, “An Effective PSO-inspired Algorithm for the Team Orienteering Problem,” *European Journal of Operational Research*, Vol. 229, Issue 2, pp. 332-344, 2013
- 10) S. W. Lin, “Solving the Team Orienteering Problem Using Effective Multi-start Simulated Annealing,” *Applied Soft Computing*, Vol. 13, Issue 2, pp. 1064-1073, 2013
- 11) M. Keshtkaran, K. Ziarati, A. Bettinelli and D. Vigo, “Enhanced Exact Solution Methods for the Team Orienteering Problem,” *International Journal of Production Research*, Vol. 54, Issue 2, pp. 591-601, 2016
- 12) R. El-Hajj, D. -C. Dang and A. Moukrim, “Solving the Team Orienteering Problem with Cutting Planes,” *Computers and Operations Research*, Vol. 74, pp. 21-30, 2016
- 13) N. Bianchessi, R. Mansini and M. G. Speranza, “A Branch-and-cut Algorithm for the Team Orienteering Problem,” *International Transactions in Operational Research*, Vol. 25, Issue 2, pp. 627-635, 2018
- 14) C. Archetti, D. Feillet, A. Hertz and M. G. Speranza, “The Capacitated Team Orienteering and Profitable Tour Problems,” *Journal of the Operational Research Society*, Vol. 60, Issue 6, pp. 831-842, 2009
- 15) A. Gunawan, K. M. Ng, V. F. Yu, G. Adiprasetyo and H. C. Lau, “The Capacitated Team Orienteering Problem,” *Proceedings of the International Conference on Industrial Engineering and Operations Management, Bangkok, Thailand, March 5-7*, pp. 1630-1638, 2019
- 16) C. Archetti, N. Bianchessi, M.G. Speranza, “Optimal Solutions for Routing Problems with Profits,” *Discrete Applied Mathematics*, Vol. 161, Issues 4-5, pp. 547-557, 2013
- 17) S. W. Lin and V. F. Yu, “A Simulated Annealing Heuristic for the Team Orienteering Problem with Time Windows,” *European Journal of Operational Research*, Vol. 217, Issue 1, pp. 94-107, 2012
- 18) Q. Hu and A. Lim, “An Iterative Three-component Heuristic for the Team Orienteering Problem with Time Windows,” *European Journal of Operational Research*, Vol. 232, Issue 2, pp. 276-286, 2014
- 19) T. Cura, “An Artificial Bee Colony Algorithm Approach for the Team Orienteering Problem with Time Windows,” *Computers and Industrial Engineering*, Vol. 74, pp. 270-290, 2014
- 20) M. Desrochers and G. Laporte, “Improvements and Extensions to the Miller-Tucker-Zemlin Subtour Elimination Constraints,” *Operations Research Letters*, Vol.10, pp.27-36, 1991
- 21) M. M. Solomon, “Algorithm for the Vehicle Routing and Scheduling Problems with Time Windows Constraints,” *Operations Research*, Vol. 35, No. 2, pp. 254-265, 1987
- 22) Gurobi Optimization, Inc. Gurobi Optimizer Reference Manual Version 8.1.0, 2018