

# 線形制約を持つ配送計画問題に対する効率的な巡回路評価法

## An Efficient Route Evaluation Method for the Vehicle Routing Problem with Linear Constraints

○ 岡本 優太<sup>1</sup>  
東京海洋大学

<sup>1</sup>Yuta Okamoto  
Tokyo University of Marine Science and Technology  
yutaokamoto1997@gmail.com

橋本 英樹<sup>2</sup>  
東京海洋大学

<sup>2</sup>Hideki Hashimoto  
Tokyo University of Marine Science and Technology  
hhashi0@kaiyodai.ac.jp

**Abstract** For many vehicle routing problems, a solution consists of not only vehicle routes but also the schedules of the vehicles along the routes. The problem of finding optimal schedule for a vehicle route can often be formulated as a linear programming problem. In this paper, we consider the vehicle routing problem with linear constraints that a vehicle route can be evaluated as a linear programming problem. Many heuristic algorithms for vehicle routing problems use local search methods. The standard neighborhoods are the 2-opt\* neighborhood, the cross exchange neighborhood and the Or-opt neighborhood. In this paper, we propose an efficient evaluation method for those neighborhoods for the vehicle routing problem with linear constraints. The computational results for generated instances showed the efficiency of the proposed method.

### 1 はじめに

近年、各業界の物流分野ではオンラインショッピングの需要増加や超高齢社会における労働者人口の減少によって人材不足が問題になっている。その中の重要な課題として荷物を各地点に配送する際の順番・時間を決める課題や在庫を最適化する課題がある。本論文では荷物を各地点に配送する際の順番・時間を決める課題について取り組む。この課題は一般的に配送計画問題と呼ばれ、代表的な組合せ最適化問題の1つとして知られる [5]。また、実在する課題の中には配送計画問題として扱うことができるものが多くあり実用上重要な問題である。

本論文では線形制約を持つ配送計画問題を考える。配送計画問題には様々なタイプの問題があり、

時間枠付き配送計画問題などのように、車両の巡回路だけでなく、その訪問スケジュールの決定も必要となる場合がある。車両の巡回路に対してその最適な訪問スケジュールを求める問題は線形計画問題として定式化できることが多い。このような車両の巡回路の評価を線形計画問題として定式化できる問題を本論文では線形制約を持つ配送計画問題と呼ぶ。本論文では、例として、電気自動車による時間枠付き配送計画問題を考える。

配送計画問題は巡回セールスマン問題の一般形であり、巡回セールスマン問題が NP 困難であるため配送計画問題も NP 困難であることが知られている。従って厳密解法の他に大規模な問題例を実用的な時間で解くための近似アルゴリズムが数多く提案されている [1, 2]。配送計画問題に対する近似解法の代表的な手法として局所探索法が挙

げられる。局所探索法とは、与えられた初期解から始めて、現在の解の近傍内により良い解があればそれに置き換える、という操作を反復するものである。最終的には、近傍内に自分より良い解が存在しない状態になるが、この解を局所最適と呼ぶ。局所探索法では近傍内の解の評価にかかる計算の効率性が局所探索法全体の計算時間に大きな影響を与える。

本論文では、線形制約を持つ配送計画問題に対し、局所探索法で近傍解を評価する際のサブルーチンである巡回路の評価を効率的に行う手法を提案する。配送計画問題に対する局所探索法でよく用いられる近傍として 2-opt\* 近傍, cross 交換近傍, Or-opt 近傍がある。これらの近傍解は全て現在の車両の巡回路を構成する枝を数本組み替えることで得られる解であり、このことを利用した効率的な近傍探索の枠組みが知られている [6]。この枠組みでは、現在の解の各客  $i$  に対して、客  $i$  までの巡回路の情報を格納したデータ構造  $\mathcal{F}_i$  と客  $i$  以降の巡回路の情報を格納した  $\mathcal{B}_i$  を保持する。本論文では、 $\mathcal{F}_i$  として客  $i$  までの巡回路の線形計画問題の双対問題の最適解、 $\mathcal{B}_i$  として客  $i$  以降の巡回路の線形計画問題の双対問題の最適解を用いた手法を提案する。計算実験では、電気自動車による時間枠付き配送計画問題の例に対して、単純に線形計画問題を解くことで巡回路を評価する方法と比較し、提案手法の有効性を確認する。

## 2 線形制約を持つ配送計画問題

この節では、線形制約を持つ配送計画問題の具体的な例として、電気自動車による時間枠付き配送計画問題を説明する。なお、本論文では巡回路は所与のものとし、その評価に論点を絞って議論を進める。

電気自動車による時間枠付き配送計画問題の線形な制約として、(1) 顧客の時間枠に関する制約、(2) 集荷から配達までの制限時間に関する制約、(3) 車両のバッテリー補充に関する制約がある。それぞれの制約について説明する。まず、(1) 顧客の時間枠に関する制約についてである。これは、顧客の都合により荷物を届けることができる時間が決まっているという制約である。例えば、オンラインショッピングで買い物をした際に決める希望配達時間をこの制約で扱うことができる。続いて、

(2) 集荷から配達までの制限時間に関する制約についてである。これは、荷物をある地点で集荷してから別の地点へ配達する際、その運搬時間に上限が定められているという制約である。例えば、食べ物を運ぶ際に料理が冷めないうちに配達して欲しいという状況が挙げられる。最後に、(3) 車両のバッテリー補充に関する制約についてである。これは、車両にバッテリー残量というパラメータを導入し、それが 0 以下でない限り走行できるという制約である。バッテリー残量は走行することによって減少し、各顧客にある補充地点で補充することで増加させることができる。この制約は例えば、電気自動車の充電残量を考慮した走行を扱うことができる。

### 2.1 定数定義

$N$  : 顧客数. 顧客は  $1, \dots, N$  の  $N$  人

0 : デポ

$[e_i, l_i]$ ,  $(e_i, l_i \in \mathbb{R}^+, e_i \leq l_i)$ : 顧客  $i$  の時間枠

$P$  : 荷物を集荷する顧客とそれを配達する顧客の対からなる集合

$t_{(i,j)} \in \mathbb{R}^+$ ,  $((i,j) \in P)$  : 顧客  $i$  で集荷した荷物を顧客  $j$  に配達するまでの制限時間

$d_{i,j} \in \mathbb{R}^+$  : 顧客  $i, j$  間の移動時間

$F \in \mathbb{R}^+$  : 車両が顧客 0 を出発する時のバッテリー残量値

$C \in \mathbb{R}^+$  : 車両のバッテリーの容量

$\alpha \in \mathbb{R}^+$  : 車両の走行距離におけるバッテリーの消費量

### 2.2 変数定義

$s_i \in \mathbb{R}^+$  : 顧客  $i$  に車両が到着する時刻を表す変数

$c_i \in \mathbb{R}^+$  : 顧客  $i$  における車両のバッテリー補充時間を表す変数

$r_i \in \mathbb{R}^+$  : 顧客  $i$  に到着した時のバッテリー残量を表す変数

$p_i \in \mathbb{R}^+$  : 顧客  $i$  における時間枠の違反量を表す変数

$q_i \in \mathbb{R}^+$  : 顧客  $i$  の荷物を配達するまでの制限時間に関する違反量を表す変数

## 2.3 定式化

$$\min \sum_{i=1}^N p_i + \sum_{i=1}^N q_i \quad (1)$$

$$\text{s.t. } s_i + d_{i,i+1} + c_i \leq s_{i+1} \quad i = 0, 1, \dots, N-1 \quad (2)$$

$$e_i - p_i \leq s_i \leq l_i + p_i \quad i = 0, 1, \dots, N-1 \quad (3)$$

$$s_j - s_i \leq t_{(i,j)} + q_i \quad \forall (i, j) \in P \quad (4)$$

$$r_0 = F \quad (5)$$

$$r_i = r_{i-1} + c_{i-1} - \alpha d_{i-1,i} \quad i = 1, 2, \dots, N-1 \quad (6)$$

$$0 \leq c_i \leq C - r_i \quad i = 0, 1, \dots, N-1 \quad (7)$$

$$s_i \geq 0 \quad i = 0, 1, \dots, N-1 \quad (8)$$

$$r_i \geq 0 \quad i = 0, 1, \dots, N-1 \quad (9)$$

$$p_i \geq 0 \quad i = 0, 1, \dots, N-1 \quad (10)$$

式 (1) は、巡回路内の顧客への到着時刻の和を表す関数である。これを最小化する問題となっている。

式 (2) は、顧客を訪問する順序と時間に関する制約である。巡回路内で次の顧客へ移動する際には、移動時間とバッテリー補充時間がかかる。

式 (3) は、各顧客の時間枠に関する制約である。各顧客の時間枠内に訪問されなければならない。

式 (4) は、顧客  $i$  で集荷した荷物を配達するまでの制限時間に関する制約である。

式 (5) は、顧客 0 におけるバッテリー残量に関する制約である。

式 (6) は、顧客 0 以外の顧客におけるバッテリー残量  $r_i$  に関する制約である。顧客  $r_i$  に到着した時点でのバッテリー残量は、顧客  $i-1$  に到着した時点でのバッテリー残量に顧客  $i-1$  におけるバッテリー補充量  $c_i$  を加え、 $d_{i-1,i}$  時間の走行で消費するバッテリーの量を差し引いたものである。

式 (7) は、各顧客においてバッテリーを補充する量  $c_i$  の下限と上限に関する制約である。 $c_i$  は負の

値はとれず、車両のバッテリーの容量を超えるように補充することはできない。

式 (8) は、変数  $s$  に関する非負制約である。

式 (9) は、変数  $r$  に関する非負制約である。

式 (10) は、変数  $p$  に関する非負制約である。

## 3 効率的な巡回路評価法

標準的な近傍に対する効率的な近傍探索の枠組み [6] を使用するためには、現在の解の各客  $i$  に対して保持するデータ構造  $\mathcal{F}_i$  と  $\mathcal{B}_i$  を設定する必要がある。本論文では、 $\mathcal{F}_i$  として客  $i$  までの巡回路に対応する線形計画問題の双対問題の最適解、 $\mathcal{B}_i$  として客  $i$  以降の巡回路に対応する線形計画問題の双対問題の最適解と設定する。この近傍探索の枠組みでは、 $\mathcal{F}_{i-1}$  から  $\mathcal{F}_i$  の計算 (forward 操作)、 $\mathcal{B}_{i+1}$  から  $\mathcal{B}_i$  の計算 (backward 操作)、 $\mathcal{F}_i$  と  $\mathcal{B}_{i+1}$  からその巡回路の評価の計算 (connect 操作) の 3 つの操作を  $O(T)$  時間で実行できるとき、近傍解を一つあたり  $O(T)$  時間で評価することができる。

この節では、巡回路の前半と後半を繋ぐことで評価する connect 操作について述べる。残りの forward 操作と backward 操作については同様に実行できるのでここでは割愛する。

### 3.1 巡回路を評価するための線形計画問題

与えられた 1 つの巡回路全体を評価するための線形計画問題  $P$  を (P)

$$\min \begin{pmatrix} c_f & c_b \end{pmatrix} \begin{pmatrix} x_f \\ x_b \end{pmatrix} \quad (11)$$

$$\text{s.t. } \begin{pmatrix} A_f & 0 \\ 0 & A_b \\ & A_c \end{pmatrix} \begin{pmatrix} x_f \\ x_b \end{pmatrix} \geq \begin{pmatrix} b_f \\ b_b \\ b_c \end{pmatrix} \quad (12)$$

$$x_f, x_b \geq 0 \quad (13)$$

とする。ここで、 $x_f, A_f, b_f, c_f$  は巡回路の前半に関する変数と定数であり、 $x_b, A_b, b_b, c_b$  は巡回路の後半に関する変数と定数、 $A_c$  と  $b_c$  は巡回路の前半と後半を跨ぐ制約に関する定数である。この問題  $P$  を解くことで巡回路を評価することができる。

主問題  $P$  の双対問題  $D$  は次のようになる。

(D)

$$\max \quad \begin{pmatrix} y_f & y_b & y_c \end{pmatrix} \begin{pmatrix} b_f \\ b_b \\ b_c \end{pmatrix} \quad (14)$$

$$\text{s.t.} \quad \begin{pmatrix} y_f & y_b & y_c \end{pmatrix} \begin{pmatrix} A_f & 0 \\ 0 & A_b \\ & A_c \end{pmatrix} \leq \begin{pmatrix} c_f & c_b \end{pmatrix} \quad (15)$$

$$y_f, y_b, y_c \geq 0. \quad (16)$$

### 3.2 connect 操作

巡回路の前半の情報を使った次の問題を  $P_f$  とする.

( $P_f$ )

$$\min \quad c_f x_f \quad (17)$$

$$\text{s.t.} \quad A_f x_f \geq b_f \quad (18)$$

$$x_f \geq 0. \quad (19)$$

問題  $P_f$  の双対問題  $D_f$  は次のようになる.

( $D_f$ )

$$\max \quad y_f b_f \quad (20)$$

$$\text{s.t.} \quad y_f A_f \leq c_f \quad (21)$$

$$y_f \geq 0. \quad (22)$$

巡回路の後半の情報を使った問題  $P_b$  と  $D_b$  も同様に定義する.

ここで,  $y_f^*$ ,  $y_b^*$  をそれぞれ  $D_f$ ,  $D_b$  の最適解とすると,

$$\begin{pmatrix} y_f^* & y_b^* & 0 \end{pmatrix} \begin{pmatrix} A_f & 0 \\ 0 & A_b \\ & A_c \end{pmatrix} = \begin{pmatrix} y_f^* A_f & y_b^* A_b \end{pmatrix} \leq \begin{pmatrix} c_f & c_b \end{pmatrix} \quad (23)$$

となる. 従って,  $\begin{pmatrix} y_f^* & y_b^* & 0 \end{pmatrix}$  は  $D$  の実行可能基底解となる.

connect 操作の実行時において,  $y_f^*$  と  $y_b^*$  は既に計算されているので,  $\begin{pmatrix} y_f^* & y_b^* & 0 \end{pmatrix}$  を初期解として  $D$  に対して単体法を実行することで巡回路の評価をすることができる. この方法では  $D$  の最適解にある程度近い解を初期解として単体法を開始できるため, 単純に問題  $P$  を一から解くよりも, 効率的に巡回路の評価計算ができると期待できる.

## 4 計算実験

実験の目的は, 巡回路を評価するための工夫を行う提案手法と工夫を行わない手法の2つを比較し, 工夫を行う手法の求解速度における優位性を示すことである.

実験環境は 2.3 GHz Intel Core i5 16 GB 2133 MHz LPDDR3, 使用言語は Python である. 線形計画問題の求解には MIP ソルバー Gurobi 9.0.2 を用いた. 単体法で初期解を利用した求解を行う際には, Gurobi の機能である VBasis/CBasis を用いた. これは Gurobi に入力したモデルの初期基底解を与える機能である [4, 3].

### 4.1 計算実験用データの生成

計算実験の対象として, 電気自動車による時間枠付き配送計画問題を用いた. 顧客数が  $N = 10 \sim 1000$  のデータを以下のように生成した. 車両は顧客を番号順に訪問するものとする. 生成された巡回路に対して, 巡回路の長さ  $N$  の半分  $\lceil N/2 \rceil$  を巡回路の前半, 残りを後半とし, これらに connect 操作を実行して巡回路に対して評価の計算時間を測定する.

顧客はランダムに定めた「座標, 時間枠」の情報を持している. 「座標」は, 各座標の最小値を  $-10$ , 最大値を  $10$  として, 一様分布に従う乱数によって定めた. 「時間枠」は開始時刻から終了時刻までの最小値を  $0$ , 最大値を  $10$  とし一様分布に従う乱数によって定めた. また, その時, 次に述べるの点に留意して定めた. 任意の2人の顧客  $i, j$  について, 先に訪問される顧客  $i$  の時間枠の開始時刻  $e_i$  が後に訪問される顧客  $j$  の時間枠の開始時刻  $e_j$  よりも常に等しいまたは小さくすることである. その他, 「集荷・配達のパア」はランダムに定め, 「集荷から配達までの制限時間」は集荷地点を訪問してから配達地点を訪問するまでに走行する距離に比例した値とした. また, 車両は「バッテリーの容量  $C$ , デポ出発時のバッテリー残量  $F$ 」の情報を保持している. 「バッテリーの容量  $C$ 」は  $20$  に, 「デポ出発時のバッテリー残量  $F$ 」は  $16$  とした. その他, 「走行距離におけるバッテリーの消費量  $\alpha$ 」は  $0.16$  とした.

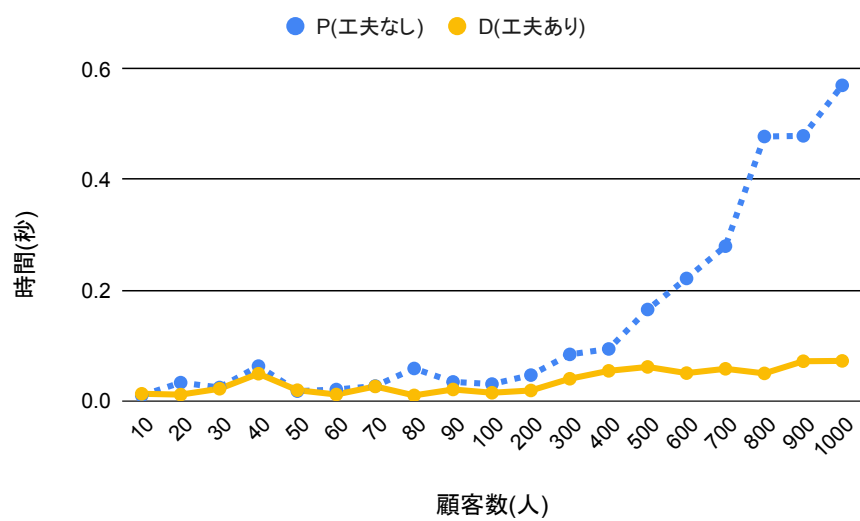


図 1: 実験結果：計算時間

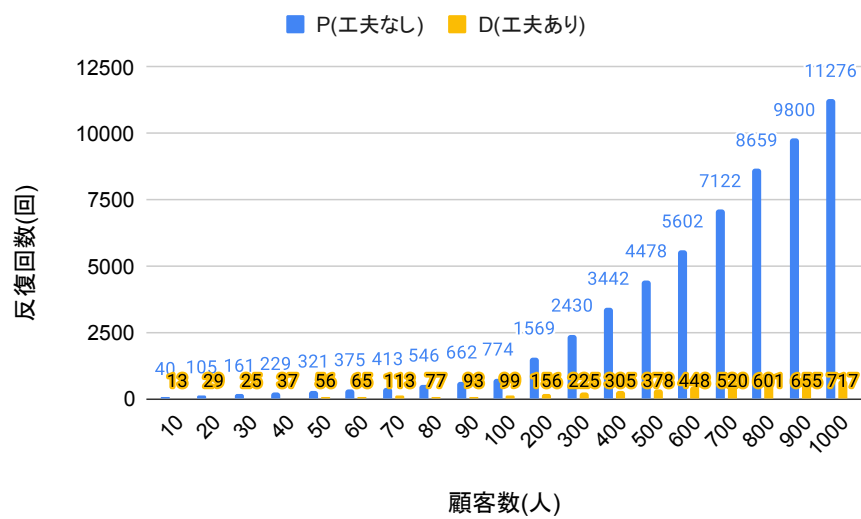


図 2: 実験結果：反復回数

表 1: 実験結果

顧客数	P(工夫あり)		D(工夫なし)	
	求解時間	反復回数	求解時間	反復回数
10	0.0094	40	0.0124	13
20	0.0326	105	0.0109	29
30	0.0238	161	0.0215	25
40	0.0625	229	0.0487	37
50	0.0172	321	0.0189	56
60	0.0201	375	0.0111	65
70	0.0264	413	0.0257	113
80	0.0582	546	0.0095	77
90	0.0339	662	0.0202	93
100	0.0300	774	0.0143	99
200	0.0462	1569	0.0184	156
300	0.0837	2430	0.0396	225
400	0.0935	3442	0.0539	305
500	0.1649	4478	0.0610	378
600	0.2211	5602	0.0498	448
700	0.2795	7122	0.0576	520
800	0.4779	8659	0.0493	601
900	0.4791	9800	0.0713	655
1000	0.5705	11276	0.0719	717

## 4.2 実験結果

実験結果を表 1 に示す。顧客数は巡回路に含まれる顧客数を表し、「P(工夫なし)」は巡回路の評価をするのに問題 P に対して単体法を一から実行した計算結果、「D(工夫あり)」は問題 D に対して初期解を与えて単体法を実行した提案手法の計算結果である。また、計算時間と反復回数をグラフにしたものをそれぞれ図 1 と 2 に示す。

実験結果より、反復回数については全てのデータにおいて、提案手法の反復回数が大幅に少なくなっており、その差は顧客数が多くなるとより顕著になっている。計算時間については顧客数が 100 程度まではほぼ同程度の時間であるが、それより大きい顧客数では提案手法の効果が現れており、顧客数 1000 では提案手法の方が約 8 倍程度高速に計算されていることが分かる。

## 5 まとめ

本論文では線形制約を持つ配送計画問題を提案した。この問題は様々なタイプの配送計画問題を表現できる一般性の高い問題である。また、この問題に対して、局所探索法の効率的な実装に不可欠である巡回路を効率的に評価する方法を提案した。電気自動車による時間枠付き配送計画問題に対する計算実験の結果、巡回路の評価について、

単純に線形計画問題を解いて評価する方法と比べて、提案手法の有効性の高さが示された。巡回路に含まれる顧客数が 1000 の場合は、提案手法が約 8 倍高速に実行できたことが確認できた。他の配送計画問題に本提案手法を適用する際、巡回路を評価するための線形計画問題が大規模あるいは複雑であれば、さらに提案手法の効果が高くなると期待される。

## 参考文献

- [1] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Science*, 39(1):104–118, 2005.
- [2] O. Bräysy and M. Gendreau. Vehicle routing problem with time windows, part II: Metaheuristics. *Transportation Science*, 39(1):119–139, 2005.
- [3] Gurobi Optimization, LLC. Cbasis. <https://www.gurobi.com/documentation/9.1/refman/cbasis.html>.
- [4] Gurobi Optimization, LLC. Vbasis. <https://www.gurobi.com/documentation/9.1/refman/vbasis.html>.
- [5] P. Toth and D. Vigo. *Vehicle routing: problems, methods, and applications*. SIAM, 2014.
- [6] 橋本, 胡. 配送計画問題に対する発見的解法. システム/制御/情報, 64(6):218—223, 2020.