

荷重による飛行速度の変化を考慮したドローン配送計画

舟橋 勇佑¹ 谷口 一徹² 富山 宏之¹

概要：近年、荷物配送ドローンは有望な産業応用として期待されている。ドローンは蓄電池駆動のため、飛行可能な時間は限られている。また、ドローンの飛行速度は荷重の影響を受けるため、荷重の影響を考慮したドローンによる荷物配送計画は非常に重要である。そこで、本研究では荷重の影響を考慮した荷物配送計画問題を定義し、効率的に問題を解く動的計画法を提案する。実験結果から、提案アルゴリズムの有効性が確認できた。

キーワード： delivery drones, VRP, dynamic programming, FSVRP

1. はじめに

輸送システムはリアルタイム性が非常に重要なサイバーフィジカルシステムの一つである。近年、無人航空機(UAVs)やドローンは趣味やビジネスなどの幅広い用途で注目されている。その中でも特に、ドローンは渋滞などの交通状況に左右されないため、宅配サービスに適していると考えられている。また、CO₂排出量が少なく人件費もかからないといった点でも、トラックによる配送に比べて優れている。

本論文では、配送用ドローンによる配送計画問題を扱う。配送用ドローンによる配送計画問題では、配送すべき荷物が与えられると、デポから始まり、全ての荷物を配送したのちデポに戻る最適な経路を求める。これは、最短距離の経路を求める巡回セールスマン問題(TSP)に似ている。文献[3],[4],[5]では、消費エネルギーを最小化する配送計画問題(EMVRP)と呼ばれる配送用ドローンによる配送計画問題を対象としている。文献では、ドローンの消費エネルギーは移動距離だけでなく荷重にも依存していると仮定している。ただし、荷重に関係なく飛行速度は一定であると仮定されている。

本論文は、荷重による飛行速度の変化を考慮したドローン配送計画(FSVRP)という配送用ドローンによる配送計画問題を扱う。FSVRPでは配送にかかる総飛行時間が最小となる経路を求める。本論文では、ドローンの飛行時間は飛行距離だけでなく飛行速度にも依存し、飛行速度は荷重に依存すると仮定している。

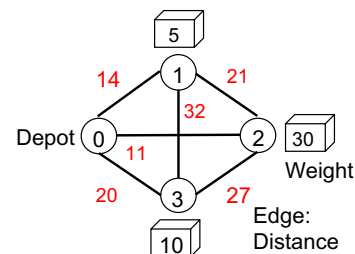


図 1 An example problem.

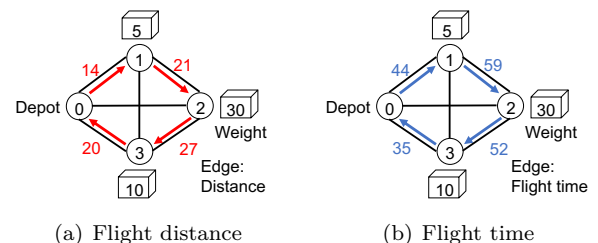


図 2 Optimal route of TSP.

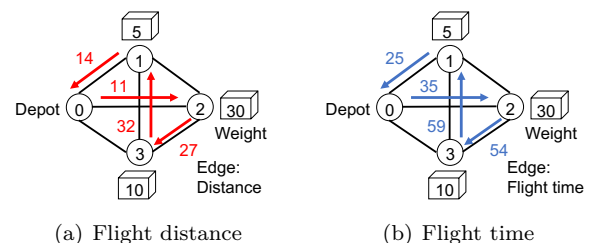


図 3 Optimal route of FSVRP.

図 1 に問題の例を示す。“0”と付いたノードはデポを示し、他の3つのノードは顧客(配送先)を示している。箱の中の数字は配送する荷物の荷重を示し、各ノードを結ぶエッジの数字は2地点間の距離を示している。TSPの最適な経路を図2に示す。経路の総飛行距離は82(=14+21+27+20)

¹ 立命館大学大学院理工学研究科 〒525-8577 滋賀県草津市野路東1-1-1
² 大阪大学大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘1-5

となる。しかし、この経路は総飛行時間の観点では最適ではない。この経路の総飛行時間*1 は 190 (= 44 + 59 + 52 + 35) である。

図 3 にこの例題における FSVRP の最適な経路を示す。FSVRP の最適な経路の総飛行距離は 84 (= 11 + 27 + 32 + 14) であり、TSP の最適な経路よりもわずかに長くなっている。しかし、FSVRP の最適な経路の総飛行時間は 173 (= 35 + 54 + 59 + 25) であり、TSP の最適な経路よりも短くなっている。一般的に、荷重が重ければ重いほど、ドローンの飛行速度は遅くなる。図 2 では、ドローンはデポから出発し、荷重 45 で顧客 1 に訪問し最も軽い荷重 5 の荷物を下ろす。その後、ドローンは荷重 40 で他の顧客を訪問する。一方、図 3 では、ドローンは最初に顧客 2 を訪問し、最も重い荷重 30 の荷物を下ろす。その後、ドローンは他の顧客を高速に飛行することができる。この例から、TSP の最適な経路と FSVRP の最適な経路が同じであるとは限らないことがわかる。

本論文は FSVRP を定義した最初の論文である。第 2 章では問題の定式化について説明し、第 3 章では FSVRP を解く動的計画法を提案する。第 4 章で提案手法の評価を示し、第 5 章で本論文のまとめと今後の課題について述べる。

2. 問題の定式化

本章では、最初に一般的な TSP について説明し、次に TSP を拡張した FSVRP について説明する。

2.1 一般的な配送計画問題

はじめに、配送する荷物が N 個与えられる。2 つ以上の荷物がある顧客に配送することはなく、各顧客は必ず一つだけ荷物を要求する。したがって、顧客数は荷物の総数である N となる。各荷物には 1 から N の番号が割り当てられている。荷物 i ($1 \leq i \leq N$) が配送される顧客を、顧客 i またはポイント i と呼ぶ。また、図 1 に示すように、デポは 0 が割り当てられる。

本論文では、ドローンが全ての荷物を一度の飛行で配達できることを前提としている。ドローンはデポを出発する際、全ての荷物を積み出発する。荷物の総重量がドローンの最大積載量を超える場合、配送計画の前に荷物をグループに分割する必要があるが、分割方法に関しては本論文では対象としていない。

$d(i1, i2)$ はポイント $i1$ と $i2$ の間の距離とする。 $x(j)$ は配送計画の決定変数であり、 j 番目に訪問した顧客とする。経路はデポが開始、終了地点であるため、以下のように定義される。

$$x(0) = x(N + 1) = 0 \quad (1)$$

また、全ての顧客が必ず一度訪問されるので、以下のように

*1 飛行時間は第 2 章 3 節の式に基づいて計算される。

定義される。

$$1 \leq x(j) \leq N \quad (1 \leq j \leq N) \quad (2)$$

$$x(j1) \neq x(j2) \quad (1 \leq j1, j2 \leq N, j1 \neq j2) \quad (3)$$

一般的な TSP は最も飛行距離が短い経路を求めるので、目的関数は以下のように定義される。

$$\text{minimize} \quad \sum_{j=0}^N d(x(j), x(j+1)) \quad (4)$$

2.2 荷重による飛行速度の変化を考慮したドローン配送計画

$w(i)$ が荷物 i の荷重を示し、 W_{drone} はドローン本体の重量を示す。 $W(j)$ がドローンが j 番目に訪問する顧客を出発する時の総重量とする。ドローンが配送を始めると、全ての荷物がドローンに積まれるので、次式が成立する。

$$W(0) = \sum_{i=1}^N w(i) \quad (5)$$

ドローンが j 番目のポイント $x(j)$ に訪問した時、ドローンは荷重 $w(x(j))$ の荷物を下ろす。したがって、ドローンがポイント $x(j)$ を訪問し終えた時、ドローンの総重量は以下のように定義される。

$$W(j) = W(j-1) - w(x(j)) \quad (6)$$

$t(i1, i2)$ がポイント $i1, i2$ 間の飛行時間を示しているとする。ドローンが $i1$ から $i2$ に飛行する時間は、 $i1, i2$ 間の距離を荷重を変数とした関数である飛行速度 v で割った値となる。したがって、ポイント $i1, i2$ 間の飛行時間は以下のように定義される。

$$t(x(j), x(j+1)) = d(x(j), x(j+1)) / v(W(j)) \quad (7)$$

FSVRP は総飛行時間が最も短い経路を求める。したがって、目的関数は以下のように定義される。

$$\text{minimize} \quad T = \sum_{j=0}^N d(x(j), x(j+1)) / v(W(j)) \quad (8)$$

本論文で扱う FSVRP は次のように定義される。 w, d と v が与えられた時、制約式 (1)-(3), (5)-(7) を満たす目的関数 (8) を最小化する x を求める。

2.3 荷重による飛行速度への影響

本節では、ドローンの荷重が飛行速度 v に及ぼす影響について、簡単な概算を述べる。前述の問題定義内の、飛行速度の関数である v はドローンに依存している。 v は与えられているとし、正確な v については本論文の対象外である。

図 4 (a) は水平飛行しているドローンを示している。 P はドローンの推進力を示し、 W_d はドローン本体の重量を

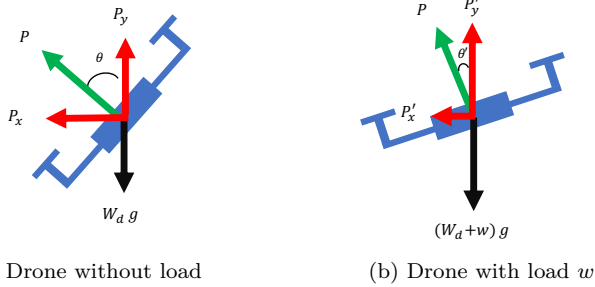


図 4 Forces on a drone.

示している。ドローンが落ちないために、 P の垂直成分 P_y と重力 $W_d \cdot g$ と等しくならなければならない。 θ をピッチ角とし、 P_y が重力と等しい時、以下の式が成立する。

$$P_y = P \cos(\theta) = W_d \cdot g \quad (9)$$

P の水平成分 P_x は目的地に向かう力であり、空気抵抗 $k \cdot v(0)$ に等しい。ここで k はドローン固有の係数である。

$$P_x = P \sin(\theta) = k \cdot v(0) \quad (10)$$

図 4 (b) は、荷重 w を運ぶ際のドローンを示している。墜落しないために、ピッチ角 θ' は θ より小さくしなければならない。よって、次式が成り立つ。

$$P'_y = P \cos(\theta') = (W_d + w) \cdot g \quad (11)$$

$$P'_x = P \sin(\theta') = k \cdot v(w) \quad (12)$$

したがって、以下の式が導出される：

$$v(w) = \frac{P}{k} \sin(\theta') = \frac{\sin(\theta')}{\sin(\theta)} v(0) \quad (13)$$

$$\theta' = \arccos\left(\frac{(W_d + w) \cdot g}{P}\right) \quad (14)$$

$$\theta = \arccos\left(\frac{W_d \cdot g}{P}\right) \quad (15)$$

3. 動的計画法

本章では、前章で定義された FSVRP を解くアルゴリズムを提案する。提案手法は、動的計画法 [6] に基づいている。

3.1 動作原理

一般的に、動的計画法 (DP) は数理計画問題に対する解法である。DP は、問題を再帰的に小さな部分問題として分割する。部分問題の最適解を利用することで、元の問題の最適解を求める手法である。DP は、同様な部分問題の再計算を回避することで、効率的に求解することができる。DP の設計において、元の問題と部分問題の間にある再帰関係を導出することが重要である。

S は訪問済みの顧客の集合を示し、 i が S の中で最後に訪問した顧客とする。 $state$ を S と i のペア (S, i) とする。初期状態は、訪問済みの顧客はおらず、デポにいる状態なので

$(\{0\}, 0)$ である。次に、初期状態から (S, i) に配送するための最小の飛行時間である $T(S, i)$ を求める問題を定義する。 $T(S, i)$ は以下のような再帰式を用いて計算される。

$$T(S, i) = \min\{ T(S \setminus i, i') + t(W'(\bar{S}) + w(i), d(i', i)) \mid i' \in S \setminus i \} \quad (16)$$

i は S の中で最後に訪問した顧客であり、 i' を i の前に訪問した顧客とする。 $T(S \setminus i, i')$ はデポから i' への最小飛行時間であり、 $t(W'(\bar{S}) + w(i), d(i', i))$ は i' から i への飛行時間である。 $W'(\bar{S})$ は、まだ配送されていない荷物の総重量であり、次式のように定義する。

$$W'(\bar{S}) = \sum_{k \notin \bar{S}} w(k) \quad (17)$$

式 (16) において、 i' から出発するとき荷物 i はドローンに積まれている状態である。したがって、 $w(i)$ は $W'(\bar{S})$ に含まれる。また、初期状態では飛行時間は 0 である。

$$T(\{0\}, 0) = 0 \quad (18)$$

元の問題は、ドローンがデポから出発し、 N 個全ての顧客に荷物を配送しデポに帰ってくるまでの最小飛行時間を求める問題である。したがって、元の問題は以下のように定義される。

$$T(\{0, 1, 2, \dots, N, 0\}, 0) \quad (19)$$

式 (19) の問題は式 (16) に従い再帰的に分割され、式 (18) に到達し、最小飛行時間が得られる。

3.2 アルゴリズム

第 3 章 1 節の動作原理に基づいて、本論文の動的計画法を実装する。アルゴリズムの疑似コードを Algorithm 1 に示す。

W_{init} は配送する全ての荷物の総重量であり、 V は顧客の集合である。 $state$ は訪問済みの顧客の集合である。 $state$ は長さ N のビットベクトルである。また、 N は顧客数である。顧客 i が訪問されると $(i - 1)$ 番目のビットが立つ。 $dp[state][customer]$ は飛行時間を格納する二次元配列であり、前節での $T(S, i)$ に対応している。例えば、 $dp[0011][2]$ はドローンが顧客 1, 2 に訪問済みであり、ドローンが現在いる顧客が 2 の時の総飛行時間を格納している。2-5 行目でデポから各顧客に訪問した時の飛行時間を計算し、6-17 行目で残り全ての顧客を訪問するのにかかる飛行時間を計算し、19-21 行目でドローンがデポに戻り最小飛行時間が計算される。6-17 行目が DP の主要部分である。再帰呼び出しの代わりに、三重ループによって飛行時間を計算する。本論文で提案する DP の計算量は、 $O(2^N \times N^2)$ である。これは $O(N!)$ の全探索より高速である。

Algorithm 1 Dynamic Programming for FSVRP

```

1:  $W_{init} \leftarrow \sum W$ 
2: for  $next\_customer \in \mathbb{V}$  do
3:    $dp[1 \ll (next\_customer - 1)][next\_customer] \leftarrow flight\_time(depot \text{ to } next\_customer \text{ with } W_{init})$ 
4:    $Weight[1 \ll (next\_customer - 1)] \leftarrow (W_{init} - W_{next\_customer})$ 
5: end for
6: for  $state \text{ in } (2^N - 1)$  do
7:   for  $next\_customer \in \mathbb{V}$  do
8:     if  $next\_customer$  has not been visited yet then
9:       for  $prev\_customer \in \mathbb{V}$  do
10:         $dp[state|(1 \ll (next\_customer - 1))[next\_customer] \leftarrow$ 
11:           $\min(dp[state|prev\_customer] + flight\_time(prev\_customer \text{ to } next\_customer \text{ with } Weight[state]),$ 
12:             $dp[state|(1 \ll (next\_customer - 1))[next\_customer])$ 
13:           $Weight[state|(1 \ll (next\_customer - 1))] \leftarrow Weight[state] - W_{next\_customer}$ 
14:        end for
15:      end if
16:    end for
17:  end for
18:  $min\_cost \leftarrow INFINITE$ 
19: for  $prev\_customer \in \mathbb{V}$  do
20:    $min\_cost \leftarrow \min(dp[2^N - 1][prev\_customer] + flight\_time(prev\_customer \text{ to } depot \text{ without } payload), min\_cost)$ 
21: end for

```

4. 評価実験

本章では、提案アルゴリズムの有効性を実験により示す。本論文では、提案アルゴリズムおよび比較対象となるアルゴリズムを Python により実装し、各アルゴリズムの実行時間と総飛行時間、総飛行距離を比較する。

4.1 実験手順

本節では、実験に用いた 6 つの探索アルゴリズムについて述べる。各手法の説明は以下の通りである。

- **TSP-NN**: TSP を解く最近傍探索。現在訪問している顧客から最も距離が近い顧客を選択する。
- **TSP-BF**: TSP を解く全探索。実行可能な経路 ($N!$ 通り) を全て探索し、最短経路を求める。
- **TSP-DP**: TSP を解く動的計画法。飛行時間を考慮せず、最短経路を探索する。
- **FSVRP-NN**: FSVRP を解く最近傍探索。現在訪問している顧客から最も飛行時間が短い顧客を選択する。TSP-NN に似ているが、最も距離が近い顧客を選択するか最も飛行時間が短い顧客を選択するかが異なる。
- **FSVRP-BF**: FSVRP を解く全探索。実行可能な経路 ($N!$ 通り) を全て探索し、最も総飛行時間が短い経路を求める。
- **FSVRP-DP**: 第 3 章で提案した FSVRP を解く動的計画法。

顧客数が 5 から 20 の FSVRP の問題計 320 個をランダムに生成した。また、各顧客数で 20 個の問題が存在する。

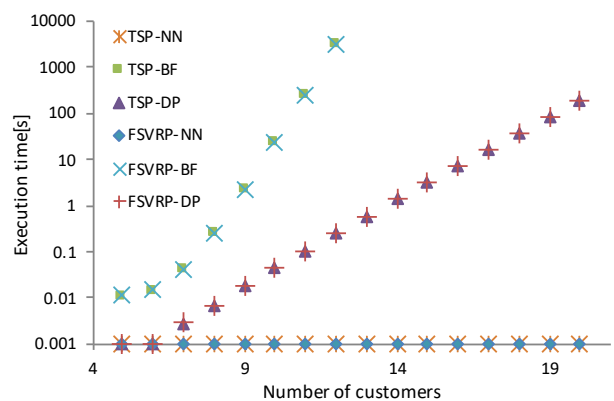


図 5 Runtime of the routing algorithms.

4.2 実験結果

図 5 は、6 つのアルゴリズムの実行時間を示している。TSP-BF と FSVRP-BF は、顧客数 12 を超えると 1 時間以内で最適解が得られない。TSP-DP と FSVRP-DP は計算量は指数関数的だが、顧客数が 20 程度であれば、十分実用的である。

図 6 は総飛行時間の結果を TSP-DP で正規化したものを示している。結果から、TSP-DP は飛行時間に関して最適でないことを示している。また、FSVRP-DP は TSP-DP よりも平均 3.94% 優れている。そして、FSVRP-DP は FSVRP-BF と同じ解を求めることができる。

図 7 は総飛行距離の結果を TSP-DP で正規化したものを示している。FSVRP の最適解は TSP の最適解よりも平均 0.63% 距離が長いことを示している。

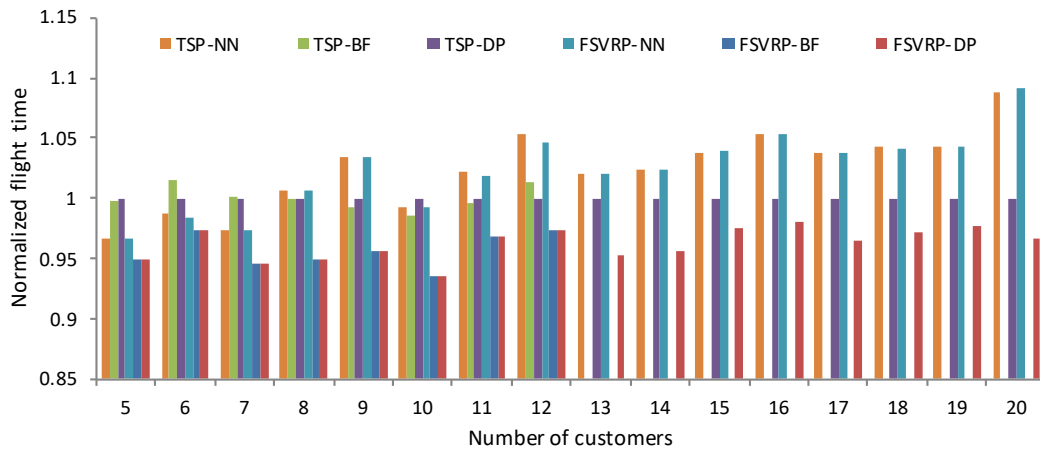


図 6 Total flight time of the obtained routes.

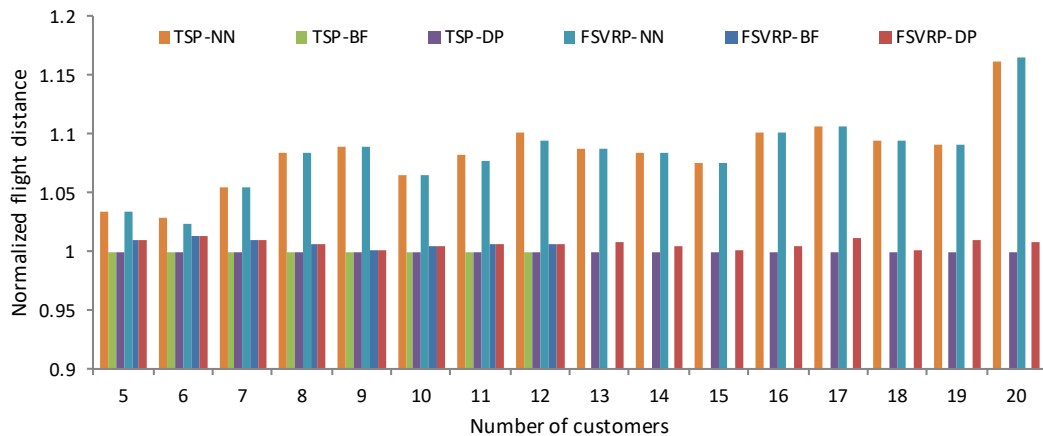


図 7 Total flight distance of the obtained routes.

5. おわりに

今後、ドローンは宅配サービスに普及されることが予想される。本論文では、荷重によってドローンの飛行速度が変化する配送計画を定式化し、効率的に FSVRP を解く動的計画法を提案した。実験結果から、提案手法が効果的であることを示した。今後、風などの動的に変化する要素を考慮した配送計画について検討する。

参考文献

[1] D. Zhang, T. He, “pCruise: Reducing Cruising Miles for Taxicab Networks,” Real-Time Systems Symposium, 2012.

[2] F. Kong, Q. Xiang, L. Kong, X. Liu, “On-line Event-driven Scheduling for Electric Vehicle Charging Via Park-and-charge,” Real-Time Systems Symposium, 2016.

[3] S. Negoro, I. Taniguchi, H. Tomiyama, “Fundamental Analysis of Low Energy Path Routing for Delivery Quadcopters,” International Technical Conference on Circuits/Systems, Computers and Communications,

2016.

[4] K. Dorling, J. Heinrichs, G. G. Messier, S. Magierowski, “Vehicle Routing Problems for Drone Delivery,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 1, pp. 70-85, 2017.

[5] Y. Funabashi, A. Shibata, S. Negoro, I. Taniguchi, H. Tomiyama, “A Dynamic Programming Algorithm for Energy-aware Routing of Delivery Drones,” International Conference Artificial Intelligence and Data Engineering, 2019.

[6] M. Held, R. M. Karp, “A Dynamic Programming Approach to Sequencing Problems,” Journal of the Society for Industrial and Applied Mathematics, vol. 10, no. 1, pp. 196-210, 1962.