移動時間の時刻依存性を考慮した配送計画システムの開発

高田　陽介，武内　優太

物流のラストワンマイルと呼ばれる街中での配送においては，現在も配送員や配車係が知識や経験に基づいて配送計画を考え配送が行われている．著者らは配送計画を自動で計算し提案するサービス“Loogia（ルージア）”を提供しており，現場で求められるさまざまな条件を考慮する汎用的な配送計画ソルバーを開発している．本稿では現場で求められる条件の一つである“移動時間の時刻依存性”を加味するための取り組みを紹介する．

キーワード：配送計画問題，局所探索法，SaaS

1.　はじめに

配送計画問題は，種々の制約条件のもとで，複数の車両を用いてすべての顧客をちょうど1度ずつ訪問するような経路の中で，コストが最小のものを求める問題の総称である．容量制約のみを考える基本的な問題でもNP困難であることが知られており，現実に現れる問題を配送計画問題として扱うにあたり古くからヒューリスティックな解法が提案されてきた．通常の配送計画問題では訪問点間の移動時間を定数で与える．しかし宅配便など実際の配送を考えると，朝夕のラッシュアワーや時間帯によって通行不可となる道路（スクールゾーンなど）があり，移動時間が時刻に依存することを考慮しなければ現実的な計画とならない場合がある．移動時間の時刻依存性を考慮した問題は古くから扱われており [1]，さらにそれらに対して局所探索法を適用する場合には近傍解をいかに効率的に評価するかということが課題の一つとして取り組まれてきた [2, 3].

本取り組みでは，2点間の移動時間がFIFO条件 [2] を満たす（つまり出発する時刻が遅くなることで到着が早まることはない）ような出発時刻に関する区分線形関数で表現される配送計画問題において，車両の移動時間の最小化を目的とする．文献 [3] では本取り組みでの設定に加え，各訪問点間の移動コストや各訪問点での作業開始時刻に依存したペナルティコストを区分線形関数で扱うことも考えており，近傍解評価においては最適な各訪問先の作業開始時刻を効率的に求めるアルゴリズムを提案している．本取り組みでは移動コストや作業開始時刻に対するペナルティは考えず移動時間にのみ着目することでよりシンプルなアルゴリズムを実装した．

2.　サービス紹介

著者らが開発および提供するサービス“Loogia（ルージア）”は，物流のラストワンマイルと呼ばれる街中での配送を行う事業者に対し配送計画を自動で計算し提案するサービスである．当該サービスは車両の台数やデポの情報，配送先の住所，時間枠などを入力として受け取る．それらを配送計画問題として解釈しアルゴリズムによって解を求めることで，すべての配送先を回る効率的な計画を提案する．

サービスの内部では配送計画問題を扱うアルゴリズムや地図上での最短経路問題を扱うアルゴリズムなどが実装されている．著者らは配送現場での実際の配送で必要とされる制約をなるべく加味し，移動時間をなるべく正確に見積もることで，“実際に実行できる計画”を提案することを目指しアルゴリズムの開発を進めている．1節で述べたとおり，“移動時間の時刻依存性”を加味しない計画は現実的に実行できない場合がある．そのため著者らは当該サービスにおいて移動時間の時刻依存性を加味することを目的とし本取り組みを実施した．

3.　実装

移動時間の時刻依存性を考慮したアルゴリズムの実装について，区分線形関数による移動時間の表現および近傍解の効率的評価という二つの側面から紹介する．

3.1　区分線形関数による移動時間の表現

現実においてはある2点間の移動時間は道路の混み

たかだ　ようすけ，たけうち　ゆうた
（株）オプティマインド
〒460–0008 愛知県名古屋市中区栄2丁目11–30
セントラルビル9階
takada.yosuke@optimind.tech
takeuchi.yuta@optimind.tech

具合や時間帯ごとの規制による影響などにより，出発時刻に依存して連続的に変化すると考えられる．ただし 2 点間の移動時間を連続関数で定義した配送計画問題を扱う最適化アルゴリズムの開発は非常に難易度が高いと思われる．また配送計画問題を解く段階において考える移動時間はあくまでも予測値であり，その予測値を正確に表現したうえで配送計画を立てたとしてもその計画が実際に配送をしたときの状況を正確に表せているわけではない．ベテランの配送員や配車係が見ても違和感のない範囲で時刻に依存した移動時間を近似的に表現することを考え，本取り組みにおいては時間を有限な個数の区分に分割しその時間帯の移動時間を線形関数として定義する（つまり区分線形関数として定義する）こととした．以下では 2 点 i, j 間の移動時間を表す区分線形関数の区分のうち時刻が早い方から k 番目の区分（これを区分 k と呼ぶ）を $[t_{ij}^k, t_{ij}^{k+1})$ と記す．また出発時刻 t における i, j 間の移動時間を $d_{ij}(t)$ とする．

3.2　区分ごとの移動時間の算出

次に現実の移動時間を近似した区分線形関数を実際のシステム上で作る方法を考える．一般的に見られる経路探索サービスやアルゴリズムはある出発時刻における移動時間を定数で出力するという仕様が多く，2 点間の移動時間を関数で出力するようなサービスは著者らの調査では見つかっていない．また著者らが開発する既存のプログラムも同じ仕様である．本取り組みにおいては移動時間の計算はなるべく既存の資産を利用することを前提とし，以下のような方針を考える．

- 通勤ラッシュや時間帯ごとの通行止めなど移動時間が大きく異なることが見込まれるような時間帯ごとに区分を分ける．
- 直前の区分と比較して移動時間が長いと思われる区分 k に対し，移動時間が長くなり始める時刻を t_{ij}^k とし，この区分の移動時間を移動時間が十分に長くなる時刻 t' における移動時間の値とする．
- 直前の区分と比較して移動時間が短いと思われる区分 k に対し，移動時間が十分短くなる時刻を t_{ij}^k とし，この区分の移動時間を移動時間が十分に短くなる時刻における移動時間の値とする．

ただし区分を決めるために事前に全時間帯における移動時間を連続的に求めることは現実的ではないため，ある程度の予測の元で “移動時間が長くなり始める時刻” や “移動時間が十分短くなる時刻” を決めている．たとえば，計画全体を 15 時から 22 時までとし，夕方にのみ道が混雑して移動時間が増加すると予想した場合を考える（図 1）．16 時から徐々に道が混み始めて移動時間が長くなると考え，t_{ij}^1 を 15 時，t_{ij}^2 を 16 時とする．夕方の混雑は 18 時ごろから解消されてくるが，混雑が十分に緩和されるのが 19 時と考え，t_{ij}^3 を 19 時とする．各区分の値はそれぞれ 15 時，17 時，19 時を出発時刻として計算した場合の移動時間を設定する．結果としてできる関数は図 1 の実線で示したものである．

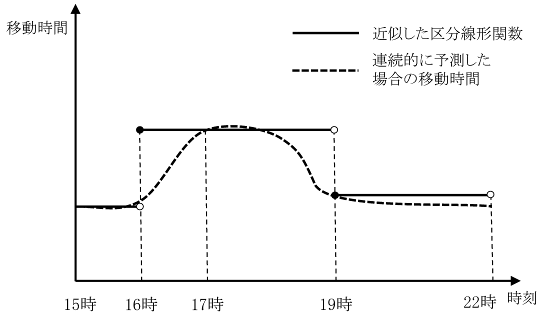

図 1　区分と移動時間の設定の例

このように移動時間の長い区間を広く取るという近似の方針を採用するのは，計画をなるべく “実行可能なもの” とするためである．計画で提示した移動時間が現実より短い場合，配送員が計画に沿って配送を行っている間に遅れが生じる可能性が高くなる．すると配送先が望む時刻や事前に連絡した時刻に間に合わない可能性が高まる．一方で計画で提示した移動時間が現実より長い場合は実際の配送が前倒しとなるが，その場合は待機することで簡単に計画どおりに修正することができる．計画で余裕をもちすぎることで，たとえば一日に配送できる荷物数が少なく見積もられてしまったり，必要な配送員の数を多く見積もったりしてしまう可能性もあるが，本取り組みにおいてはある程度のコストをかけてでも計画が実行できないリスクを軽減することを優先する．また，ある 2 点間において移動時間が非常に長くなるような時間帯があると見積もったうえで計画を立てる場合は，探索の時点でその 2 点間をその時間帯に通るような解が採用されにくくなる．結果として，出力される計画は本来より長い移動時間が含まれるものではなく，そういった経路を避けるようなものが出やすくなることが期待される．以上のことから，なるべく混雑する時間帯を広めに見積もるという方針をとることで，実際の配送が遅れてしまうリスクを抑えつつ現実に近い移動時間を見積もった計画を立てることができると考える．

3.3　区分ごとの移動時間の算出による課題と解決

このように時間の区分ごとに定数で移動時間を定義

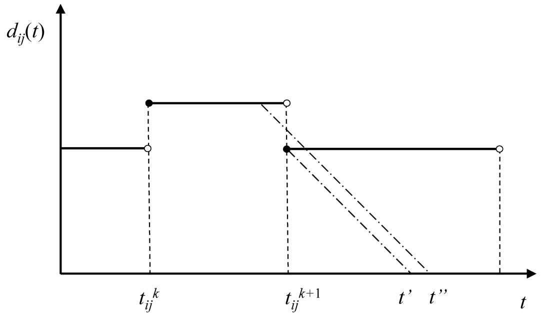

図 2　FIFO 条件を満たさない場合

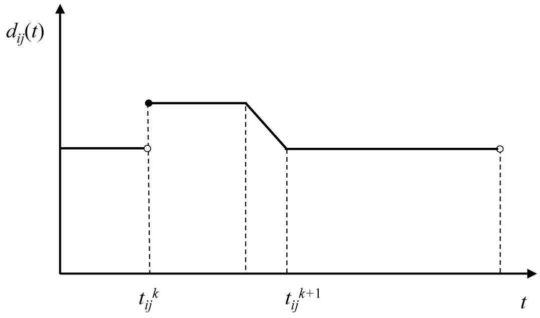

図 3　FIFO 条件を満たすように処理した場合

した場合，以下の二つの問題が発生する．まず，ある区分の値が隣接する次の区分の値よりも大きい場合，つまり

$$d_{ij}(t_{ij}^{k}) > d_{ij}(t_{ij}^{k+1}) \tag{1}$$

を満たす場合に発生する問題がある．イメージを図 2 に示す．このとき時刻 t_{ij}^{k+1} に出発した場合の j への到着時刻 t' は，$t_{ij}^{k+1} - (d_{ij}(t_{ij}^{k}) - d_{ij}(t_{ij}^{k+1})) < t < t_{ij}^{k+1}$ を満たす時刻 t に出発した場合の到着時刻 t'' よりも早くなり，FIFO 条件を満たさない．しかし実際の 2 点間の移動において常に所要時間が最短である経路を選ぶと仮定すると FIFO 条件は満たされる（つまり出発する時刻が遅くなることで到着が早まることはない）．また配送計画問題として扱う場合にも，早く到着することで実行不可能となったりペナルティがかかったりということがないような問題設定であれば，FIFO 条件を満たすと仮定することで常に計画を前詰めで評価すれば最適解を逃さないといえるため都合がよい．よって，式 (1) を満たす場合は FIFO 条件を満たすようにするために区分 k の値と区分 $k+1$ の値を連続に繋げるような傾き -1 の線形関数が定義された区分を区分 k の中に新たに生成する．処理後のイメージを図 3 に示す．各区分は 3.2 節に記載した方法で作られているため，この例においては t_{ij}^{k+1} は区分 k から見て十分に移動時間が短くなった時刻である．そのため区分 k の最後に傾き -1 の新しい区分を作ることで，本来の予測移動時間を下回ることなくより正確に近似するような関数を作ることができる．

時間の区分ごとに定数で移動時間を定義することによる二つ目の問題として，スクールゾーンなど特定の時間帯に通行不可であるような道路に訪問先がある場合にその時間帯のその点への移動時間を定義できないということが挙げられる．著者らが開発する 2 点間経路探索のプログラムではある時刻における規制情報が

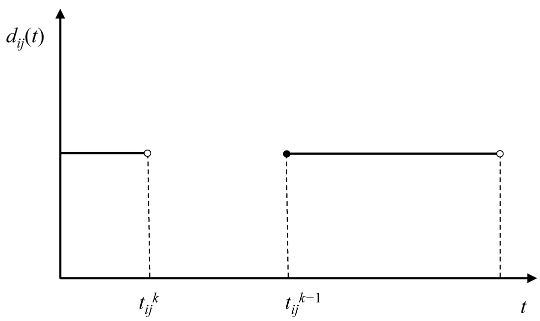

図 4　移動時間が無限大の区分を含む場合

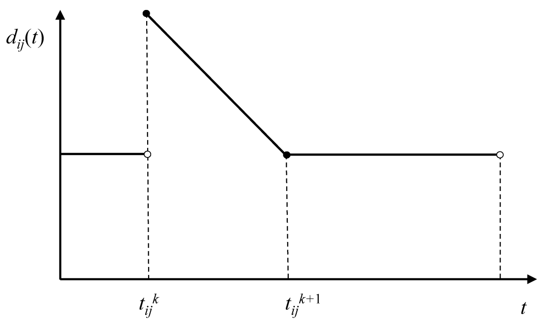

図 5　移動時間が無限大の区分がなくなるよう処理した場合

静的に定義されたグラフ上で最短経路を計算する．たとえば午前 7 時時点での規制情報を用いた場合，7 時に通行できない道路上に目的地（または出発地）が設定されているとどのような経路を使ってもたどり着くことができないため，7 時に出発した場合の移動時間を設定しようとした区分では移動時間が無限大となる．イメージを図 4 に示す．

本来は通行禁止である道路の直前まで先に移動し，そこで規制が解除されるまで待機してから移動を再開するのが最も移動時間を短くできる方法であるが，その場合の移動時間を求めるためにはある 2 点間に対して時間帯規制やそれが解除されるまでの待機時間など

を考えた最短経路問題を解く必要がある．ただし本取り組みは 2 点間経路の計算ロジックは既存のものを使うという考えに基づいているためこの方法を採ることはできない．なるべく現実に近い計画を出すことを目指しつつ計画がずれた場合になるべく配送員が臨機応変に対応できることを期待し，移動時間の予測は長くなる方向にすることを考え，無限大の区分を次の区分に接続するように傾き -1 の線形関数で補完するという処理を行う．これは通行不可の規制が解除されるまでの時間を解除直後の移動時間に足し合わせたものであり，規制が解除されるまで出発地点で待機した後に出発したものと解釈できる．処理後の移動時間関数のイメージを図 5 に示す．

3.4 近傍解の効率的評価

著者らが開発している最適化アルゴリズムはメタヒューリスティックの枠組みを用いており，内部で訪問点や辺に着目した近傍操作を採用している．時刻依存性を考慮しながらそれらの近傍操作における近傍解の目的関数値の評価（以降，近傍解評価と呼ぶ）を効率的に計算する方法を実装した．ここでの目的関数値とは車両の移動時間を指す．

以下では 2-opt*近傍 [4] のように，ルートの中の一つの辺が付け変わるような近傍操作を考える．2-opt* 近傍とは，ある解において異なるルートに属する任意の二つの訪問先を選び，それぞれの訪問先についてその訪問先までの部分ルートとその訪問先以降の部分ルートは変化させずその部分ルート同士を接続する辺を入れ替える操作により生成される解集合を指す．移動時間が時刻依存性をもたない場合，近傍操作に直接関わる辺のみに着目して近傍解評価を行うことができる．しかし移動時間が時刻依存性をもつ場合，ルートの変形が起きる部分以降の各訪問点間の移動時間も変動するため，計算の工夫をしなければ，変形が起きた部分以降のルート上の辺をすべて辿りながら時間を計算する必要がある．そのため，ルート上の訪問点の数を n とした場合，一度の近傍解評価に $O(n)$ の計算時間がかかり，n が大きくなるような問題例においては計算の効率が大きく下がる可能性がある．この問題を解決するため，近傍操作による解の移動が確定した際に，ルートの各点 v に対して「出発時刻 t に対するデポ帰着時刻」を表す関数 $F_v(t)$ を計算し保持しておくようにする．関数 $F_v(t)$ は，3.1 節で紹介した各辺に対する区分線形関数を，帰着デポからルートを逆順に辿りながら累積していくことで計算可能である．これにより，近傍解評価の際にはルートの変形の起こった辺の終点の訪問点 v に対して $F_v(t)$ を参照することでその点以降の移動時間の合計を計算することができる．結果 $F_v(t)$ の区分数を γ としたときの評価にかかる計算時間は，$F_v(t)$ を二分探索木として表現することで $O(\log \gamma)$ となる．関数 $F_v(t)$ は解が近傍解へ移動するたびに再構築する必要があるが，ここで想定するような近傍操作においては解の更新回数は近傍解の評価回数に比べて十分に少ないことが経験的にわかっており，評価に必要な計算を高速化することで全体として高速化されることが期待できる．

4. 応用結果

本取り組みの結果を可視化するため，サンプルの問題例を作成し実装前後での比較を行った．結果を図 6 および図 7 に示す．スクールゾーンが存在するエリアの周辺に三つの配送先を配置し計算した例である．時間帯の区分を (i) 7 時以前，(ii) 7 時から 9 時，(iii) 9 時以降の三つに分けている．破線で囲んだ道路がスクールゾーンによって (ii) に通行不可となる道路である．もともと (ii) にスクールゾーンを通るルート（図 6）を算出していたものが，移動時間の時刻依存性を考慮した場合では顧客 1 から顧客 2 への移動においてスクールゾーンを避け，さらに顧客 2 の地点で時間規制解除を待ってから顧客 3 に訪問する結果が得られており，時間帯 (ii) にスクールゾーンを回避するようなルート（図 7）を算出できていることが確認できる．本比較

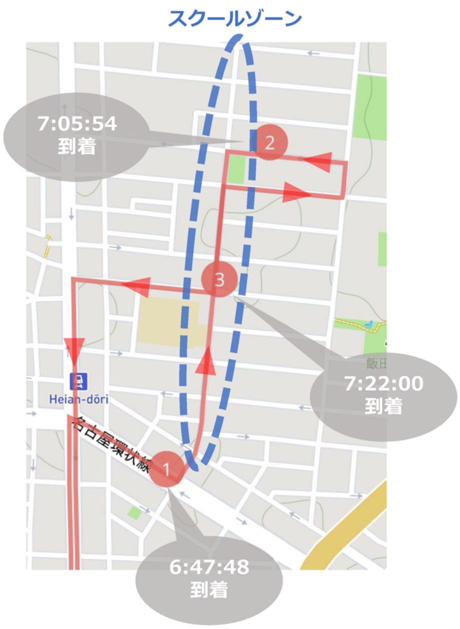

図 6 時刻依存性を考慮しない場合

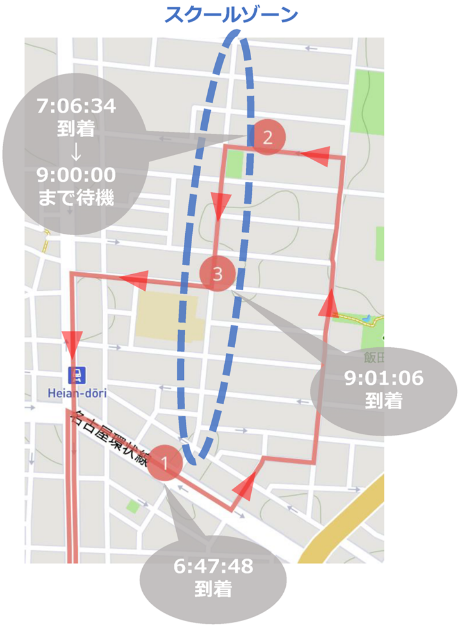

図 7　時刻依存性を考慮した場合

は結果をわかりやすくするために配送先の数を極端に少なくした例であり，実際の入力においてはスクールゾーンではない場所に多くの配送先があることが多く，その場合は本比較のように極端な待ち時間は発生しづらいことが期待できる．

5.　まとめ

本研究では，時刻依存性を考慮した配送計画問題に対するアプローチとして区分線形関数による移動時間の表現および近傍解の効率的評価について説明した．また実際の配送現場で問題となるスクールゾーン道路の回避について今回の実装が有効であることを確認した．

謝辞　名古屋大学の柳浦睦憲教授と東京海洋大学の橋本英樹准教授には普段から組合せ最適化の理論について多くのご知見をいただいており，また本稿の執筆においても有益なご助言を多くいただきました．心より感謝申し上げます．

参考文献

[1] L. Gouveia and S. Voß, “A classification of formulations for the (time-dependent) traveling salesman problem,” *European Journal of Operational Research*, **83**, pp. 69–82, 1995.

[2] S. Ichoua, M. Gendreau and J. Potvin, “Vehicle dispatching with time-dependent travel times,” *European Journal of Operational Research*, **144**, pp. 379–396, 2003.

[3] H. Hashimoto, M. Yagiura and T. Ibaraki, “An iterated local search algorithm for the time-dependent vehicle routing problem with time windows,” *Discrete Optimization*, **5**, pp. 434–456, 2008.

[4] J. Y. Potvin, T. Kervahut, B. L. Garcia and J. M. Rousseau, “The vehicle routing problem with time windows part I: Tabu search,” *Informs Journal on Computing*, **8**, pp. 158–164, 1996.