

地図データを使った戸別配達経路最適化

倉田 是

流通経済大学 流通情報学部

商品を購入して戸別に配達する傾向が増加している。この最短時間配達経路の最適化問題を扱った。地図データは MapInfo をもとにした。道路の走行時間と交差点のロス時間を考慮した戸別配達最短時間経路選択問題をアルゴリズム化し、これをコンピュータのプログラミングによって実現して実行させ、良好な結果を得た。アルゴリズムの特徴は、デポ及び配達箇所ごとに有限の高さ（6を選んだ）の進出・進入木構造を設け、経路探索の計算量を大幅に少なくしたことである。また、この木構造のポインタを利用して、最適解の交差点を表示できる。

A delivery routing problem using map data

Tadashi KURATA

Faculty of Distribution and Logistics Systems
Ryutsu Keizai University

In recent year, many families buy foods and daily goods from deliver men. Then, we propose some algorithms for solving a delivery routing problem minimizing delivery time and make algorithms into computer programs. For the shortest delivery time form door to door, the delivery truck must run the shortest route. Two trees of routing from a destination and into same destination are adopted. Using these trees, computation time and work are minimized for searching minimum route form a destination to another destination. A depot and eight destinations are set on a real residential area map. Computer programs realized these algorithms are successful in the problem on the residential area.

1. はじめに

商店が住宅区域になくなり、共稼ぎや子育て、老人世帯など買い物に不便を来すようになった。そのため、戸別に生鮮や冷凍を配達してもらう傾向が増加している。生活協同組合やコンビニエンスストアなどでこのサービスをしているが、取扱高は年に2倍くらいの伸びである。以下この配達を個配と呼ぶ。

本報告は、この個配の配達時間を短縮できるように、MapInfo[®]の地図データをもとにして、MapInfo のデータの加工を含めたアルゴリズムを開発し、プログラム化したものである。

2. 本報告の特徴

本報告も最短経路探索問題の一種である。この分野では言うまでもなく、

- (1) 巡回セールスマン問題
- (2) 配送計画問題
- (3) カーナビゲーション

等で広くまた深く扱われている。この中で本報告と同分野の(1)は現実問題ではなく、問題解法の問題である。また(2)は主要道路を利用して多量に荷物を配達する場合の問題が多い。

一方、本報告で扱う個配は、都市の住宅街で、道路幅員が3~4mの道路をトラックで戸別に配達する場合の問題を扱う。道路が狭いこともあって、大きな交通事故にはならないかもしれないが、事故の心配が多い。全国的生活協同組合の調査によると、配達に際しての事故のうち、バック事故は交通事故全体の10~50%になる⁽²⁾。したがって、配達箇所でUターンして向きを変えて最短時間で配達する経路探索問題には馴染まない。自由に道が選べる(1)の巡回セールスマン問題では、これは問題にはならなかった。

また、不況の影響によるものと思われるが、個配の配達員には、家庭の主婦のアルバイトが増加している。アルバイトであるから、必ずしも長続きせず、道順に慣れない上、トラックの運転にも慣れないこともあって、バック運転の少ない、確かな道路地図案内が必要と思われる。

本報告の特徴の一つは、道路で自由にUターンして、向きを変えることの難しい場合の最短経路探索問題である。

本報告では、住宅地域で配達物の最短配達時間を求めることを問題としている。そこで、道路車線幅の狭く、一方通行路などの住宅街に多くある道路に合わせた問題を設定した。これらの道路状況に合わせて走行速度を変え、信号機のある交差点、狭い住宅道路の交差点の徐行や一旦停止を考慮して配達時間を計算している。

本報告では、図1に示す東京近郊の約1km²の住宅地に8カ所の配達場所(①より⑧まで)を設定して、0をデポ(depot)として最短時間となる経路を求める。この区画には約70箇所の交差点がある。

このすべてのデポと配達箇所との間の経路を探索する方法では計算量が多くなる。本報告では、デポと配達箇所のそれぞれに、進出と進入経路の有向部分木を作り、その葉が重なり合う程度の高さの木構造とすることにより計算量を少なくしようとしていることも特徴の一つである。

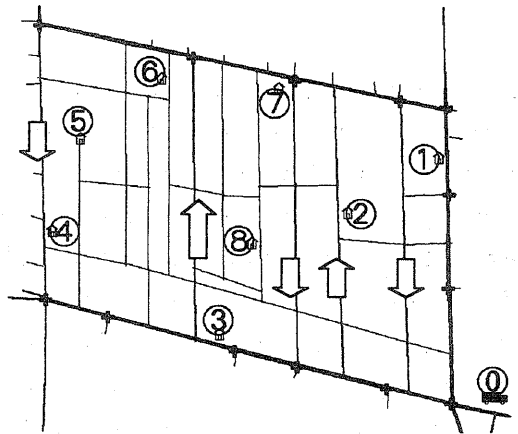


図1 デポと配達箇所(矢印は一方通行)

3. アルゴリズム

本報告で使う MapInfo の道路データは次の2種類である。

- (1) 道路点データ

(2) 道路区間データ

前者は、都道府県コード・道路点ID・隣接道路点ID・接続道路数・区間(1)ID・区間(2)ID・・・等よりなる。後者は区間ID・区間長・車線数コード・接続点ID・・・等よりなる。各ID(たまたまこの地域では5~6桁の整数値)はMapInfoのデータ中でユニークなものである。このデータを合体させ、図2に示す交差点の基本有向木構造を作る。

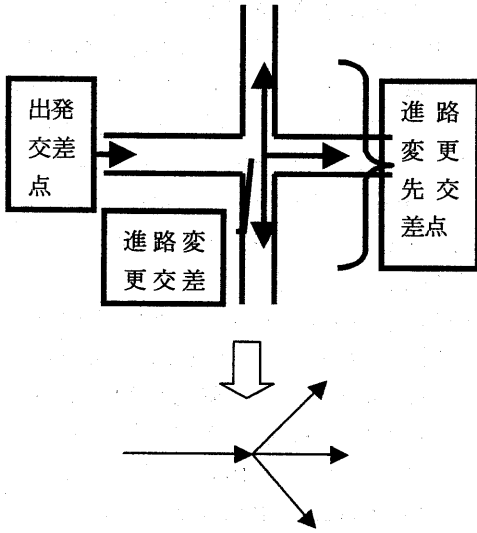


図2 基本有向木構造

この各ノードは交差点(道路点ID)であり、各枝の属性は区間長・路線番号・道路幅員・道路車線数である。MapInfoのデータに加えて、信号機設置の有無と一方通行路の指数を設けた。幹ばかりでなく、すべての枝に路線番号や道路車線数を付加したのは、左折・直進・右折の識別をプログラムにより自動的に行うためである。特にT字路などの3差路交差点には有効であった。

一方通行路を作成する際には、注意する必要がある。一方通行路に逆向する基本有向木構造の出発交差点と進路変更交差点を削除するばかりでなく、他の木の枝にも注意して切り取らなければならないことである。

図2の基本有向木構造を使って左折・直進・右

折の別を求めた後に、次の処理には少ないデータ数で行えるように工夫した。その結果、基本有向木構造を表1のものとした。ここで、分岐数を3としたのは、この地域に3以上の交差点がなかったからである。

表1 基本有向木構造

出発交差点	進路変更交差点	区間距離	車線数	一方通行路指数	信号機の有無	進路変更先交差点			進路変更方向			
						1	2	3	1	2	3	

ここまでのアルゴリズム実現のためのプログラミング言語にはMap Basicを用いた。Map Basicは独立したBasic言語ではなく、MapInfoデータ処理のためにある言語である。したがって、MapInfoからデータを引き出し、使いやすいように加工するには便利な言語である。たとえば、一方通行路、信号機及び仮称の路線番号の付与はすべて地図から選択して行うことができるので間違いが少なく、便利である。

しかし、配列は1次元配列までで、Type文を使ってかろうじて仮想的な2次元配列ができる程度である。やや高度なアルゴリズムの実現には不向きである。また、統合環境にあるエディタも使いにくい。製作したプログラム本数は6本である。

この地域には約70の交差点がある。デポまたは配達箇所から隣接するデポまたは配達箇所まで経路を計算する必要がある。交差点から交差点を1マッハタン距離として、デポと配達箇所相互間の距離を求めると、10マッハタン距離程度である。

もし、ある箇所から隣接箇所までのすべての経路を探索するとすると、すべての交差点が3差路として仮定して、 $3^{10}=59,049$ となる。一方、6マッハタン距離の有向部分木を双方から伸ばせば、木の葉の部分は重なることになることを利用

すると、 $3^6 \times 2 = 1,458$ となる。計算量は前者の 2.4% ときわめて少ない。

このために、デポまたは配達箇所から進出する方向の進出有向部分木と、デポまたは配達箇所に進入する進入有向部分木を作ればよいことになる。B地点からA地点への進行の繋がりの様子を図示すると図3である。

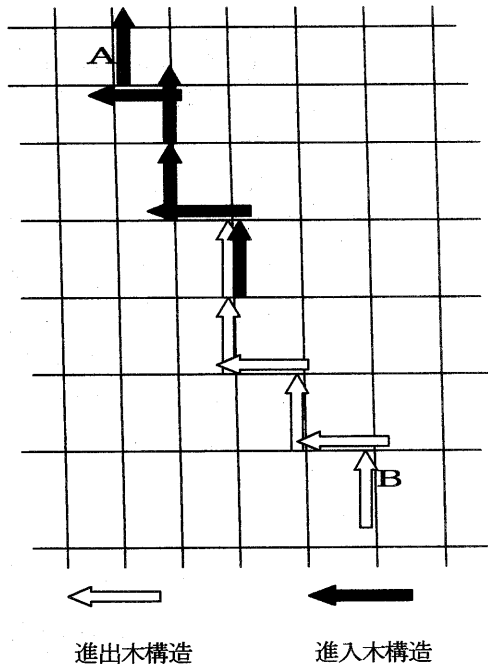


図3 2本の木構造による経路の接続

このアルゴリズムは、出発交差点・進路変更交差点・進路変更交差点をそれぞれ一つづつずらしながら繋げて行くことを行えばよい。進出木構造は、図2の基本木構造の枝にさらに次の基本木構造の幹を重ねて伸ばして行くことに相当する。進入木構造は、基本木構造の枝を手前にして、幹に次の基本木構造の幹を繋げて行くことになる。

ここで問題となるのは、繋げて行く枝や幹が内側に回り込んで、何本の枝が同一道路区間に重なることである。このために、出来上がった同一道路区間の枝(幹)を同一方向のもの内でもっとも時間が短いもののみを残すアルゴリズムを加えた。

いわば庭師が枝の込み入った木を剪定する作業である。この結果、剪定を行わなかった枝の数が総計で約 1,500 本あったのに対し、約 800 本と 40% 減少した。先の計算値より遙かに少ないのは、一方通行路や、他地域の計算しなくてよい枝があるからである。

最短時間を求めるアルゴリズムは、相互間のすべての枝を重ねて、枝があったものを候補としてすべて出力する。このとき、それぞれの進出木構造と進入木構造の位置を記憶させておく。これは後から木の枝をたどりながら、最短時間経路の交差点の連結した表を出力させるためである。このために、木構造には、ルートをもとして、枝を伸ばすごとに1つづつ増える連結ポイントが用意してある。6が最後の葉となる。このポイントを辿れば、容易に葉やノードからルートの経路を表示することができる。

次にこれらの候補の中から、デポから出発してデポに帰るまでの最短時間となる経路を捜し出す。特に、このアルゴリズムの最適化は考慮しなかった。他のプログラムは開始と同時に終了するが、このプログラムだけは、10sec ほどかかるので (CPU は Pentium II 400MHz), 計算量が多いアルゴリズムである。

最後に、この結果に含まれる進出木構造と進入木構造のポイント情報を使って、各交差点を連ねる最短時間のルートが出来上がる。

なお、Uターンも可能であり、配達箇所を指定すれば、Uターンを選んで、もっとも最短時間の経路を選ぶことも可能である。

このアルゴリズムの実現のために、C言語でプログラミングを行った。プログラム本数は6本である。なお、Map Basic と異なって、地図を操作する必要がないので、すべてをまとめて1本のプログラムとすることも可能である。

4. アルゴリズムの検証

図1に示した地区で検証を行った。

MapInfo の地図データには、一方通行路も交差点の信号機の有無も入っていない。このために、

プログラムの力を借りて、それぞれのデータを入力した。

デポ及び配達箇所データのデータはC言語のプログラムで入力する。一般に交差点と交差点の間にあり、この両者の交差点IDと、どちらか一方の交差点からの距離を入力する。このとき、同時に他方の交差点からの距離を計算して、デポ及び配達箇所前をどちらの方向から進入してもよいように、進出・進入木構造の準備をしている。

今回は交差点の進路変更（左折・直進・右折）による交差点通過時間ロスを考慮しなかったが、交差点における時間的ロスを仮定し、次のように単純な場合分けを行った。

表2 交差点通過ロス時間

進入道路	進出道路	信号機	通過時間 (0.01 分)
1車線住宅 地道路	1車線住宅 地道路	無し	20
		有り	40
	一方通行 路	無し	40
		有り	60
	2車線道路	無し	60
		有り	80
一方通行 路	1車線住宅 地道路	無し	20
		有り	40
	一方通行 路	無し	0
		有り	40
	2車線道路	無し	60
		有り	80
2車線道路	1車線住宅 地道路	無し	0
		有り	40
	一方通行 路	無し	0
		有り	40
	2車線道路	無し	20
		有り	40

配達車の速度は、道路の種類により変化させることにし、次に示す値を仮定した。

- ① 2車線道路 40km/h
- ② 一方通行道路（1車線） 30km/h
- ③ 住宅地域内1車線道路 20km/h

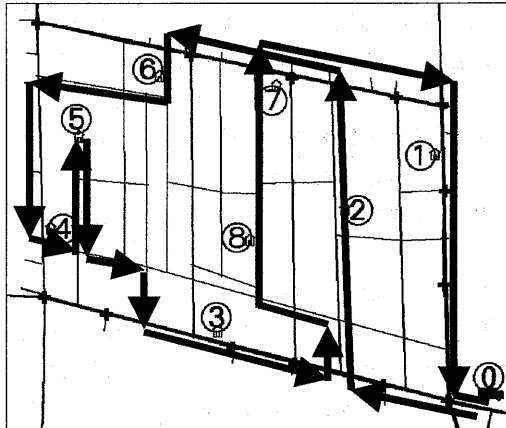


図4 最短経路

この結果を図4に経路で表示した。この経路を利用すると、最短時間 17.71 分で配達できる。なお、周辺部の道路を一周すると、約5分かかる。

⑤のUターンのところは、同一経路を往復するのは致し方ないと思うが、その他にも同一経路がある道路区間では同方向で、ある道路区間では逆方向で通過する。これは、都会の住宅地に多い一方通行路の影響と思われる。

5. 考察と今後の課題

MapInfo の地図データの処理・加工と、一方通行道路の存在する住宅地での最短時間経路を求めるアルゴリズムはほぼ成功したと考えられる。

本報告のプログラミングに限って、難易について述べると、進出木構造と進入木構造の作成であった。前者は比較的考えやすいが、特に後者は感覚的に逆なので苦労した。また、交差点通過ロス時間の計算は、交差点の種類と進入進出道路の種類によって異なるので、プログラミングで工夫の必要な部分であった。

現在の木の高さは6で固定している。それより

増加させることは、プログラムの長さを増やせばよいので簡単であるが、プログラミングに手を入れないで済み、自由に高さを調節できる再帰型関数のアルゴリズムが望ましい。これは今後の課題である。

本アルゴリズムを実現したプログラムは、まだ研究者の扱う段階である。一般に使えるためには、利用者のインターフェースを工夫しなければならない。

- (1) デポと配達箇所をディスプレイ地図画面で入力可能であること
- (2) 結果のルートを表示する地図画面で表示すること

これも今後の課題である。

最近でも都市近郊の住宅が伸びている。これは新たな道路の建設が行われていることにもなる。また個配のお客もこの地域に増える。したがって、絶えず地図データを新しくしなければならない。MapInfo は 1998 年版であるが、本報告で扱った地域でも、新しい道路ができていて、本来の実用には適さない。最新の道路地図を MapInfo の地図画面に重ねて、新しく道路を設定することも可能であるが、まだできていない。これも課題の一つである。

本報告は一連の研究⁹⁾の到達点の一つである。今回初めて交差点の左折・直進・右折の自動設定ができた。しかしながら、これを活用できなかった。左折・直進・右折のロス時間を実測したが、たまたま郊外の交通量の少ない地域なので、左折・直進・右折のロス時間に有意な結果がでなかった。本報告で扱った地域は、交通量の多い道路もあるので、有意な結果が出るのが予想される。さらなる調査によって、交差点の種類による左折・直進・右折のロス時間を導入することも今後の課題である。

一般に、都市や近郊を問わずに、午前の交通量と午後あるいは夕方の交通量は異なっている。一般に、午後と夕方が多い。このために、配達の運用管理者は配達時間帯と配達件数の配分に気を使う。ある時間帯にある道路が渋滞する場合に

は、別の経路を選択する方が最適経路となる。このような時間帯による道路の変更は、区間 ID があるので可能である。道路の混雑度に応じた最適経路選択も今後の課題である。

家庭に届ける品物であるから、多くても数 kg で、工業製品やコンビニの商品配達と異なって積載荷物の重量と容積は考慮しなかった。また、車両も中・小型が多い。これらが、一般に扱われている配送問題と異なるところである。しかしながら、大気汚染や省エネルギーを考えると、出発は満車で、帰路は空車であるから、極端に燃料消費が異なる。したがって、これを考慮した経路選択も問題とすべきであろう。

6. むすび

MapInfo の地図データをもとにして、道路の走行時間と交差点のロス時間を考慮した戸別配達の最短時間経路選択問題をアルゴリズム化し、これをコンピュータのプログラミングによって実現して、検証できた。

参考文献など

- (1) MapInfo は USA に本社があり、日本総代理店は三井造船システム技研株式会社
- (2) “特集生協の総合力で実現する車両事故削減”，生協運動，No. 581，2000 年 8 月号，pp. 5-17
- (3) 以下著者はすべて倉田是（単著）である
“配達ルートの最適化問題”，日本シミュレーション&ゲーミング学会第 10 回全国大会，1998 年 11 月
“配達経路最適化の一方法”，流通経済大学流通情報学部紀要，Vol.4，No. 1，1999 年 10 月
“配達経路の最適化問題について”，日本シミュレーション&ゲーミング学会第 11 回全国大会，1999 年 10 月
“地図データを使った配達経路の最適化アルゴリズム” 流通経済大学流通情報学部紀要，Vol.4，No.2，2000 年 4 月