

デポを考えた配送計画の最適化 ～ 配送モデルを用いての近似解の研究 ～

M2004MM034 小栗 大樹

指導教員 鈴木 敦夫

1 はじめに

製品は、様々な原料から加工を積み重ねて1つの製品として出来上がる。その生産過程の中にはいくつものルートが存在する。その中には1ヶ所で加工して出来上がる物もある。しかし、その製品の多くは加工が行なわれる各工場を経由し、加工され製品として出来上がる。そこから物流センターや配送センターのなどに送られる。そこから小売店に経由され、我々のもとに届けられる。そんな様々な配送を繰り返し製品として出来上がる中で、その1つ1つの配送を如何に時間的にも金銭的にも効率的にする事が実現されれば、生産者や私達消費者にとっても、大変に喜ばしいことである。そんな様々な配送の中でも1番我々に身近である、配送センターからの小売店への配送を、この研究では取り上げていく。その物流センターから小売店までの配送を如何に効率的に計画する事、最適化が求められている。

この問題は配送だけに留まらず、小学校、中学校さらにはスイミングスクールなどのスクールバスの効率的な巡回経路の計画、郵便や新聞などの効率的な配達経路、または収集経路の決定、ゴミの最適な収集経路の計画など、近年は配送というものを如何に最適にもしくは効率的に行なう事が出来ないか、という事に、大いに関心がもたれていると思われる。

そうした配送計画問題は、最も普及しているロジスティクス・ツールである。また、モノの配送だけでなくモノの集荷にも使われるので現在は運搬経路問題 (vehicle routing problem) としても扱われている。

2 研究目的

今回の配送計画問題「運搬経路問題」の目的は、すべての顧客の需要を満たす運搬車の最適ルート(デポを出発して再びデポへ戻ってくる単純経路)を求めることである。

今回の研究では、小売店の場合における最適な配送計画を考える。配送の効率をより上げる事が出来ればその企業にとっても決してマイナスにはならず、大きな費用の削減などにも繋がっていくであろう。そんな問題を実際に様々な制約条件を踏まえながら、自らプログラムを作成し、より効率的な実行案を提案する。

各小売店の需要量をそれぞれ配送する運搬車が、その需要を最低限満たす事を前提とする。運搬車の積載能力を十分に発揮する。運搬車についての効率的な活用を目指している。確かに、ルートを最短に押さえるという事はもちろん効率的である。しかし、実際には運搬に使用できる運搬車

の台数というものは実際には決まっている。算出されたルートが4つあったとしても、3台の運搬車で配送を行わなければならないければ、その運搬車の積載能力を十分に生かす配送が必要になってくる。そうした点からも、運搬車の利用方法という項目を重要視し、運搬車の積載能力を十分に生かせよう、プログラムに改良した。最も重要なのは配送を行なう運搬車の総移動距離を、条件を満たす上で最小に押さえることである。運搬車の総移動距離の最小化、さらなる削減が総移動時間の最小にも繋がり、勤務時間の短縮にも大いに繋がっていくであろう。

研究の中では各小売店の配置は、適当に座標上に配置するものとする。デポ「配送センターなど」を(0,0)に設置し、各小売店を(,)「例(3,4)」と配置し、それぞれの距離も座標を利用することにより絶対値を用いて求める。実行する際には運搬車の最大積載重量を適当に入力する。あらかじめファイルの中には各小売店の配置を表わす座標、各需要量も適当に入力して作成しておく。そうする事により、様々な状況にも対応できる配送計画を提案する。

配送を必要とする小売店の位置、各小売店の需要量、配送に使用可能な運搬車の台数など、そうした状況は必ずしも、常に一定であるとは限らない。仮に小売店がある範囲に10店舗あるとする。しかし、配送が必要な小売店の位置は毎回10店舗であるとは限らない。少数の店舗によってはその製品の売れ行きが大変に良い。その店舗にとっては消耗が激しいわけになる。そんな時に急にそんな各店舗から配送要請があった。そんな場合に対処が可能であろうか。そんな事に大変大きな疑問、興味を持ったため最低限の配送計画を目指す。

そんな少数の店舗、各小売店の需要量、運搬車の最大積載重量などの条件を様々に変えた場合においても、その時その時のより良い近似解を求められるような発見的解法を考案し、その有効性を推進していく。

3 セーピング法

今は、以下に示すセーピング法をを上手くプログラム内でアルゴリズムとして完成させ、取り組んでいる。簡単な解法を次に順に示す。

セーピング法(節約法)の一般的な考え方。

- (1) すべての顧客「小売店」のペア i, j に対してセーピング値 S_{ij} を次式を用いて計算する。

$$S_{ij} = C_{i0} + C_{0j} - C_{ij}$$

- (2) セーピング値を大きい順に並べたリストを作成

する。

- (3) 以下の操作をリストが空になるまで繰り返す。

リストの順に顧客 (i, j) を連結した際の実行可能性 (小売店の需要量を満たしているかどうか) を調べる。もし、連結によって実行可能性が失われてないなら (i, j) 間を繋ぐ。そうでないなら、リストから現在の顧客ペアを削除し、次の顧客ペアを調べる。

ここから分かる事は、大変に高い確率でセービング法は実行可能であるという事である。一直線上にデポを含み左右に小売店が存在する場合においては、上の式からセービング値は 0 という事になる。

4 アルゴリズム

- (1) まず、ファイルを作成した。そのファイルには各小売店の座標を入力。上の段には x 座標である数字をそれぞれ「3, 2, -1, ..., 1, 3, 4」、下の段には「1, 3, 4, ..., -4, -3, -1」と入力する。入力したファイルを図を用いて示すと図 4 のようになる。簡単なモデル図を図 1 として示す。

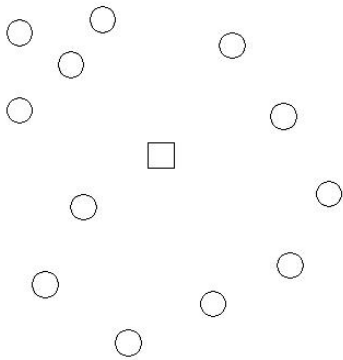


図 1 モデル図

- (2) そのファイルを次にプログラムに読み込ませ、その各座標を配列 $x[i], y[i]$ に、各小売店の需要量を $z[i]$ に入れる。
- (3) セービング法を行なう際に、まずある規則に小売店をプログラム内で並び替える。ここでは、 rad 「ラジアン」中心角を利用した。中心角は $0 \sim 2\pi$ で表わす事が出来る。その中心角の小さい順に番号を割り振る。
- (4) プログラム内で新たに割り振った番号の小さい小売店「中心角が最も小さい小売店」を基準点にし、左右の小売店を対象にセービング法を実行していく。ここでは様々な条件がある。まずはセービング値の非負を確かめる。左右の小売店とのセービング値が非負であるなら、それぞれの大小を比べる。セービング値の大きい小売店から優先的にルートの構築を行なっていく。しかしここで、いくらセー

ビング値が大きくても、その小売店の需要量が運搬車の積載能力を超えるようでは配送を行なう事は出来ない。そうした場合には、セービング値の小さい方の小売店を次に考える。このようにセービング値の大小の次には、その小売店の需要量が運搬車の積載能力の範囲内に納まるかを基準にして、セービング法を行ない、順にルートを構築していく。優先順位を付けると、以下のようになる。

- (i) セービング値が必ず非負である。
- (ii) セービング値の大きい小売店を優先的にし、ルートの構築を行なう。
- (iii) 小売店の需要量が、運搬車の積載能力を超えないようであれば、その小売店とのルートを構築する。もし、運搬車の積載能力を超えるようであれば、その小売店とはルートの構築は行なわず、別の小売店「この場合は、セービング値の小さかった小売店」との構築を考える。
- (iv) 運搬車の積載能力を超えない限りは、ルートの構築の作業を続ける。次のルートの構築の際に、どちらの小売店とルートを構築しても運搬車の積載能力を超えるようであれば、その最後にルートを構築した小売店を最後にそのセービング法は終了となり、デポとルートを構築する。
- (v) また新たな小売店を中心に左右の小売店を対象に条件に合う限り、ルートを構築していく。
- (vi) 小売店が無くなり次第終了する。
- (5) 出力の際には、プログラム内で割り振った小売店の番号を元のファイルの番号に変換させて出力させる。

5 簡単なモデルを用いての実行結果

- (i) 運搬車の積載能力が 10 の場合:
その運搬車の積載能力を入力して下さい:10
実行するファイル名を入力して下さい:data.dat
- | | | | | | |
|---|---|---|---|---|---|
| 0 | 3 | 2 | 1 | 6 | 0 |
|---|---|---|---|---|---|
- このルートでのセービング値:15.412584
- | | | | | |
|---|---|---|---|---|
| 0 | 5 | 4 | 9 | 0 |
|---|---|---|---|---|
- このルートでのセービング値:11.231643
- | | | | | |
|---|---|---|----|---|
| 0 | 8 | 7 | 12 | 0 |
|---|---|---|----|---|
- このルートでのセービング値:13.499290
- | | | | |
|---|----|----|---|
| 0 | 11 | 10 | 0 |
|---|----|----|---|
- このルートでのセービング値:6.129678

(ii) 運搬車の積載能力が 15 の場合:
 その運搬車の積載能力を入力して下さい:15
 実行するファイル名を入力して下さい:data.dat
 0 3 2 1 6 5 0
 このルートでのセービング値:22.438923
 0 4 9 8 7 12 0
 このルートでのセービング値:20.844584
 0 11 10 0
 このルートでのセービング値:6.129678

6 データを用いた研究

今回は、ある企業から配送&収集についてより現状より効率的に計画出来ないか、という研究を努めさせていただく事になり、現状の配送データを頂きました。データは以下のように記されている。

- 1 配送は分けられた区間によって現在は行なわれている。北方便、刈谷便、豊田便、春日井便、瀬戸便、稲沢便、市内便「名古屋市内」、大垣便、新三重便、三重便、東南便、小牧便と毎日それぞれ 12 便が配送を行なっている。
- 2 各便にはそれぞれ日にち毎の配送順が記されている。
- 3 それぞれの配送を行なう会社の情報は住所だけである。
- 4 毎日配送が行なわれる会社もあれば、週に 2 日、あるいは 3 日というような会社も存在する。

当初の予定では、需要量、供給量のデータを基に思考する予定であったが、配送ルートのデータしか入手出来なかったために、今回は巡回セールスマン問題と捉え最適なルートを探し出す。配送する企業の住所を基に、「ゼンリン電子地図」を用いて各企業間の道のりを算出し、距離行列を作成。その距離行列を基にこの研究を進めていく。

6.1 定式化

C_{ij} : 企業 i と企業 j 間の距離

$X_{ij} = 1$: 企業 i と企業 j がルートで結ばれるとき

$X_{ij} = 0$: その他「ルートで結ばれないとき」

i, j : 各企業の割り振り変数

y_i : 各ルート i に対する実変数 ($0 \leq y_i \leq n + 1$)

目的関数 (運搬車の移動距離の最小化)

$$\min \sum_{i,j} C_{ij} X_{ij}$$

制約条件

$$(1) \sum_i X_{ij} + \sum_j X_{ji} = 2$$

$$(2) 2 \leq \sum_i X_{0i} \leq 2n \quad (n = \text{ルートの数})$$

$$(3) X_{ii} = 0$$

$$(4) y_i - y_j + mX_{ij} \leq m - 1 \\ (1 \leq i \leq m; 1 \leq j \leq m; i \neq j)$$

式 (4) は閉路排除制約式である。これは、はじめに設定せずに解かせたところ、ルートが複数個、解として求められてしまった。この閉路排除制約式で、このデータに対して 1 つのルートを作成する事ができるようになった。

7 現状のデータの解析

市内便のデータを基に、現状の各ルートの分析を行なった。6 日間について各日ごとの配送を分析し、そのデータを用いて巡回セールスマン問題として定義し、最適であるその日ごとのルートを算出した。1 つの例を以下に示す。

1 日目 = 「0 1 2 3 4 5 6 7 8
 9 10 11 12 13 14 0」

1 日目の総移動距離は 77.8km である。ここで、1 日目の最適なルートを算出した。結果は以下である。

Route = 「0 4 1 3 9 8 7 6 5
 2 10 11 12 13 14 0」

このルートでは、総移動距離は 64.9km である。現状と比べると、12.9km もの距離の短縮、短縮率では 16.6 % に成功し、この率は大変に大きな値である。このように 6 日間分のそれぞれの最適なルートを算出し、各現状のルートと見比べたところ、平均で 1 日約 13.2 % の短縮率となった。

8 1 区間内の配送の最適化

データの中から、市内便「名古屋市内」を選択し、現在の行なわれている配送をより効率的に出来ないか提案する。配送先の企業のデータ「住所」を基にして、「ゼンリン電子地図」を用いて距離を求め距離行列を作成した。その距離行列を基に最適な配送計画を「What's Best! 7.0」を用いて解いた。毎日配送される企業もあれば、そうでない企業も存在する。今回はこの市内便で配送が行なわれる全ての企業を対象に配送計画を提案する。0 をデポに指定し、各数字を配送先の企業にする。

全ての企業を対象にしてルートを構築する利点は、その日に配送を行なわなくてもいい企業をその全てのルートから省いて配送を行なえば、その場合においても効率的な配送を行なえる点である。

簡単な例として、ある区間内の全企業を対象とした最適なルートを、

最適なルート = 「1 2 3 4 5 6 7 8 9
 10」であるとす。

実際に配送をする際に、その日は 3 と 8 の企業の配送を行わない日とする。そうすれば、最適なルートから 3 と 8 の企業をそのまま取り除き、

実行ルート = 「1 2 4 5 6 7 9 10」

と配送すれば、より効率的に近づく。もしここで、頻繁に配送を行なう企業だけで最適なルートを生成してしま

うと、新たにその日に配送が必要になった日にどのように配送したら良いのかわからなくなる。

各ルートにはそれぞれその日に応じて回る企業、さらには企業数が全て異なる。今回は全ての企業を1日で回る事を想定し、全ての企業を合わせて考えたルート、全ての企業への配送を行なうルートを求める。

8.1 最適なルート

・導き出した最適なルート

```
Route=「0 15 21 12 7 13 6 14
        5 17 9 11 8 10 16 4
        20 18 1 19 3 22 2 0」
```

8.2 考察

作成した距離行列の基、このようなルートを作成した。各ルートからも一目で分かるが、この企業の配送の現状は、まずその日によって回る企業は様々である。さらには規則性が見当たらない。そこで今回の研究では、全ての企業に回るようになった際の最大のルートを最適化で表す。総ルートの中で存在する企業数は全部で22件である。その全ての企業がこの市内便「名古屋市内」で配送が行なわれている。その最大のルートを基に、その日に回らない企業を削除すれば少なくともその日のルートが幾らか減らせるのではないかと思い、総数の合わせたルートの最適化を行なった。各ルートそれぞれ最適化を実行した際に、どれだけ現状のルートとの差が出るのかが非常に興味深い。総ルートを求めてみて現状でもそんなに効率の悪い配送を行なっていない事が伺える。

9 2区間内の最適なルートの算出

データの中から、隣接している春日井便と小牧便を選択し、現在の行なわれているそれぞれの配送を組み合わせ、新たに効率的な2つのルートが生成できないかを考える。先ほどの1区間内と同様に配送先の企業のデータ「住所」を基にして、「ゼンリン電子地図」を用いて距離を求め距離行列を作成した。その距離行列を基に最適な配送計画を「What's Best! 7.0」を用いて解いた。毎日配送される企業もあれば、そうでない企業も存在する。今回はこの市内便で配送が行なわれる全ての企業を対象に配送計画を提案する。0をデポに指定し、各数字を配送先の企業にする。

9.1 最適なルート

・導き出した最適な各ルート

```
Route1=「0 1 2 3 13 7 4 8
        12 10 11 9 14 0」
```

・導き出した最適なルート

```
Route2=「0 26 28 29 15 16 27
        17 18 19 21 22 20 24
        23 25 5 6 30 0」
```

9.2 考察

2, 3つの隣接する区間を1つの区間に考え、それぞれ2, 3つのルートを導き出したら、より現状よりも効率的なルートが生成できることこの結果から分かった。今度は全

ての12区間分の配送先の企業を対象に、最適な12ルートを生成させられればより効率的なルートが生まれる。またそこから、11ルートに減少させるなりしてこの企業の理想的な配送ルートを追求したい。

10 おわりに

今回の配送計画の研究を進めるにあたって、まず始めに簡単な図を作成しその図を例にしてプログラムを作成した。デポを原点である(0,0)に置き、各小売店を(,)とそれぞれ座標を用いて配置した。しかし、座標で小売店の位置を表した事によって、いくつかの問題点も多く表れた。1つは、座標を用いて作成した事による距離の不正確さである。例えば万が一、小売店間に大きな険しい山が聳え立っているとすると、その山を超えるには、明らかにプログラム上で座標によって求められた直線距離と実際の道のりには、大幅な誤差が生じる。しかし、プログラム内ではその距離は短いと判断される。実際の道のりはその数とはかけ離れたものであるかもしれない。

研究開始にあたっては、座標で配置図を表わす事により小売店間の距離、さらには方角まで確かなものになると思っていたが、今回企業のデータを基に研究をしたところ、重視するのはやはり距離では無く間違いなく道のりである。頂いたデータも、各企業の住所が記載されていた。頂いたデータをパソコンでソフトウェアの「ゼンリン電子地図帳」を用いて分析する際に、場合によっては距離と道のりの誤差が大きい場合も数多く見られた。この事実気付く事が遅くデータを分析中に明らかになった。以上から、配送や収集を考える際には不確かな距離よりも確かである道のりを重視する、道のりを利用する事が必須である。

最後に、企業からのデータの提供が大幅に遅れたためにもっと細かい部分まで研究が出来なかったのが、心残りである。

参考文献

- [1] G.Ghiani, G. Laporte and R. Musmanno, Introduction to Logistics Systems Planning and Control, John Wiley & Sons, Ltd, 2004.
- [2] 久保 幹雄, ロジスティクス工学, 朝倉書店, 2001.
- [3] 清水 忠昭, 菅田 一博, C言語のススメ, -Cで始めるプログラミング-, サイエンス社, 1994.
- [4] 浦 昭二, 原田 賢一, C入門, 培風館, 1994.