

# 配送路決定問題におけるデータ依存性 3

太田 佳孝 (沼田 一道 助教授)

## 1. はじめに

広域に展開して営業を行う事業体は、インターネット等の通信手段が発達した現在でも、各事業所間で、“書類”の物理的交換を必要としている。例えば、銀行においては日に何回か、事務センター（デポ）から出る複数台の配送車が各支店を巡回し、事務書類の集配を行っている。ここで広域に点在している支店をどのように分け、一台の車両が受け待つ支店をどのような順序で回るかが問題となる。このような問題を配送路決定問題 (Vehicle Routing Problem:VRP) と呼ぶ。VRP の解法としては、Saving 法や Sweep 法などが古くから知られており、また最近ではメタ戦略の一つである禁断探索法の適用も盛んに試みられている。

VRP を解いて実用に耐えうる解を得るためには、道のりデータ (所要時間データ) を用いる必要があると思われる。しかし、道のりデータ (所要時間データ) を得るには、渋滞状況、法定速度、道幅、交通規制、信号などの道路状況を熟知している必要がある。また時間帯によっても所要時間は変わってくる。よって実情を反映した道のりデータ (所要時間データ) を取得するのはかなり大変である。

そこで本研究では、道のりデータ (所要時間データ) の代わりに、容易に取得できるユークリッド距離をもとに推定した値を用いることの妥当性を検討する。ユークリッド距離をもとに推定した“道のりデータ”と地図上の最短経路から求めた道のりデータについて、実際に VRP を解き、両者の (準) 最適解の間ほどの程度の違いが生ずるかを観察する。

## 2. 配送路決定問題 (Vehicle Routing Problem:VRP)

$n$  をデポを含む配送地数、点 1 をデポ、 $V_0 = \{2, \dots, n\}$  を配送地の集合とする。 $V = V_0 \cup \{1\}$  を頂点集合とする。 $E$  を  $i, j (i, j \in V (i \neq j))$  間の辺集合とするグラフ  $G = (V, E)$  について考える。ただし、 $G$  は辺  $(i, j) \in E$  に対して  $ij$  間の移動距離  $t_{ij} (= t_{ji})$  が与えられた無向グラフとする。

このとき VRP は、グラフ  $G$  においてそれぞれ走行可能上限  $D$  をもった  $m$  台の車両に対して、実行可能なルートを確認する問題であるといえる。また、この問題における条件は以下の通りである。

- 各車両はデポを出発し、デポに戻る。
- 各配送地  $i \in V_0$  はちょうど 1 回ずつ、1 台の車両によって訪問される。
- 各車両については、得られたルートに含まれる配送地点間の距離  $t_{ij}$  の総和は車両の走行可能上限  $D$  を超えない。
- ルートの総距離を最小化する。

以上の条件より VRP を定式化すると次のようになる。

$$\text{Minimize } z = \sum_{i,j} t_{ij} \sum_k x_{ijk} \quad (1)$$

$$\text{Subject To } \sum_k y_{ik} = \begin{cases} 1, & i = 2, \dots, n, \\ m, & i = 1, \end{cases} \quad (2)$$

$$\sum_{i,j} t_{ij} x_{ijk} \leq D \quad k = 1, \dots, m, \quad (3)$$

$$\sum_j x_{ijk} = \sum_j x_{jik} = y_{ik}, \quad i = 1, \dots, n, \quad k = 1, \dots, m, \quad (4)$$

$$\sum_{i,j \in S} x_{ijk} \leq |S| - 1, \quad \text{for all } S \subseteq \{2, \dots, n\}, \quad k = 1, \dots, m, \quad (5)$$

$$y_{ik} \in \{0, 1\}, \quad i = 1, \dots, n, \quad k = 1, \dots, m, \quad (6a)$$

$$x_{ijk} \in \{0, 1\}, \quad i, j = 1, \dots, n, \quad k = 1, \dots, m. \quad (6b)$$

ただし、

$$x_{ijk} = \begin{cases} 1 & \text{車両 } k \text{ が配送地 } i \text{ の直後に配送地 } j \text{ を訪れる} \\ 0 & \text{それ以外} \end{cases}$$

$$y_{ik} = \begin{cases} 1 & \text{車両 } k \text{ が配送地 } i \text{ を訪れる} \\ 0 & \text{それ以外} \end{cases}$$

(1)はルート全体の総距離を最小化するという目的関数である。また、制約式(2)は、各配送地には必ずある車両が訪れることを示し、制約式(3)において、車両の走行可能距離制約を表す。制約式(4)において、車両は各配送地に訪れた後、そこを去ることを示している。また、式(5)は部分巡回路（デポ1に辺が接続していない閉路）を禁止するものである。

### 3. 本研究で用いる解法

本研究では、Saving法を用いる。Saving法は1964年にClark & WrightによってVRPに対する解法として開発された。この方法は、Saving値  $S_{ij}$  という指標を導入し、この指標に基づいて巡回路を決定する。

#### 3.1.1 Saving法

Saving法では距離データをもとに、任意の二つの配送先に対してデポと呼ばれる起点からそれらを個々に往復したときの距離と、デポから二つの配送先を連続して回ってデポに戻るときのルートの距離を比較して、その差をSaving値として計算する。全ての配送地を個別に配送するルートから始めて、車両の距離制約を満たすようにSaving値の大きい配送地の組み合わせの順に結んでいき、ルートを形成してゆく。

#### 3.1.2 Saving値

0をデポ、 $i$ と $j$ を配送先とし、相互間の距離を $d_{0i}$ ,  $d_{ij}$ ,  $d_{j0}$ とする。

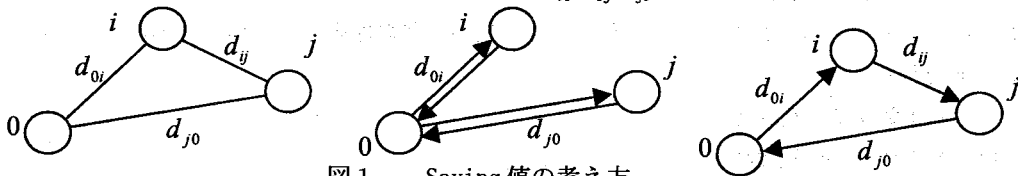


図1. Saving値の考え方

この場合のSaving値は以下のように計算できる。

$$\begin{aligned} S_{ij} &= 2d_{0i} + 2d_{j0} - (d_{0i} + d_{ij} + d_{j0}) \\ &= d_{0i} + d_{j0} - d_{ij} \end{aligned}$$

#### 3.1.3 アルゴリズム

以下にSaving法のアルゴリズムを示す。

- Step1.** 初期解として、デポと各配送地を往復するルートを設定する。そして、全ての配送地に対する Saving 値を求める。
- Step2.** 現在の全てのルートにおいて、直接デポにつながっている配送地を対象として、最も Saving 値の大きい組み合わせ（同じルートに存在しない）のルートを結合したとき、距離制約を満たすかどうかチェックする。
- Step3.** 距離制約を満たすルートの結合が存在しなくなったところで、アルゴリズムを終了。そのときのルートが解となる。

### 3. 2 Saving 法を使用した例

図2のような配送ネットワークを持つ物流センター P がある。A-J は配送先、リンク（道路）上の数字はユークリッド距離である。このとき、延べ走行距離（時間）を出来るだけ小さくする配送ルートを作成する。ただし、車両1運行の走行距離は30 km以内の制約があるものとする。

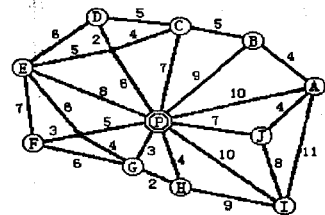


図2. 配送ネットワーク

配送ネットワーク図から物流センターと配送先の相互間及び配送相互間の最短距離を計算し、配送先の相互間の Saving 値を計算する（表1）。得られた Saving 値を降順に並び替え、セービングリストの作成（表2）。

表1. Saving 値

	A	B	D	E	F	G	H	I	J	
A		15	8	4	0	0	0	0	9	13
B			11	7	3	0	0	0	4	8
C				10	6	0	0	0	0	1
D					10	3	0	0	0	0
E						9	10	0	0	0
F							8	4	10	0
G								5	2	0
H									5	0
I										9
J										

表2. セービングリスト

1	A-D	16	13	F-C	6
2	A-I	13	13	G-B	6
3	B-C	11	13	B-I	5
4	C-D	10	16	A-D	4
5	D-E	10	16	B-I	4
6	A-I	9	16	F-E	4
7	E-F	9	19	B-E	3
8	F-I	9	19	D-F	3
9	A-C	8	21	G-I	2
10	B-J	8	22	C-I	1
11	B-D	7	22	H-F	1
12	C-E	6	22	F-I	1

Saving 値の分類表から、Saving 値の大きい順にルートを組み上げていく（図3）。

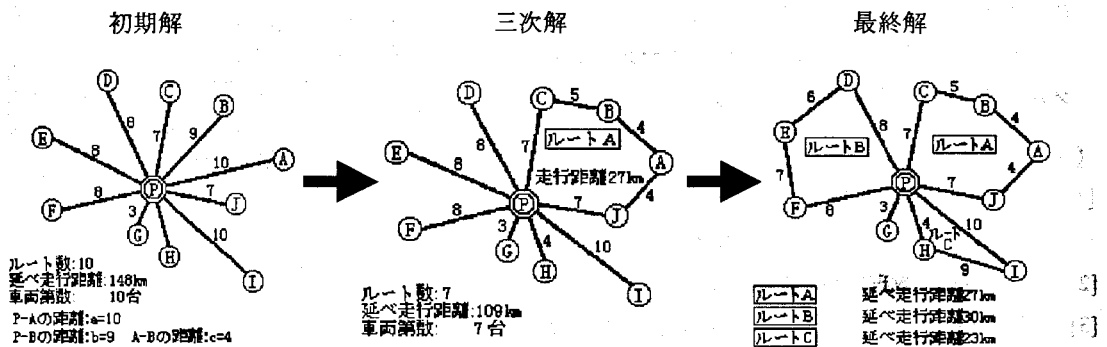


図3. 配送ルートの作成

#### 4. 扱うデータについて

本研究ではデジタル地図を用いユークリッド距離と道のりを測った。データは埼玉県  
南西部を中心に事業を展開しており、36の支店を持つH信用金庫の支店データを用いた。  
ユークリッド距離と道のりデータの相関を調べ、ユークリ  
ッド距離を回帰直線で補正したデータを算出し、それと実  
測値の道のりデータそれぞれについてVRPを解き、その両  
者を比較対照する(図4)。

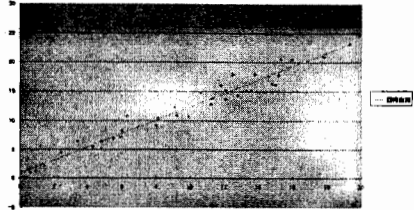


図4. 回帰直線によるデータの補正

#### 5. 数値実験および結果

計算機による実験結果を以  
下の表3に示す。使用した計  
算機は FMV-DeskPowerT II  
20(MMXPentium200Mhz, 32MB)。  
プログラムは、インプライズ  
社の Delphi3.1 を用いて作成  
した。表1には、補正後のユ  
ークリッド距離と実測した道  
のりについて、制約距離の変  
化に対する、車両台数、総走  
行距離である。

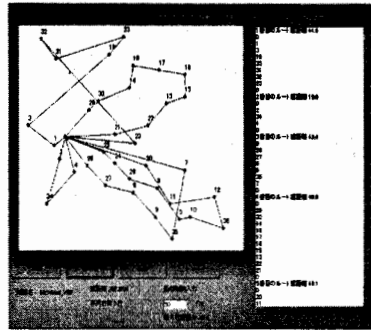


図5. 実行画面

表3. 実行結果

(制約 km 距離)	実測データ		補正データ	
	車両台数	総走行距離 (Km)	車両台数	総走行距離 (Km)
なし	1	139.7	1	136.5
5	35	890.9	35	831
10	34	888.5	34	824.1
15	33	880.2	31	804.5
20	29	828.7	27	756.3
25	24	742	24	707.9
30	20	656.7	19	605.9
35	14	493.9	13	438.9
40	9	337	8	297.5
45	7	277.4	6	227.7
50	5	202.9	4	188.8
55	4	187.6	4	172.9
60	4	186.9	4	156.7
65	3	169.9	3	156.7
70	3	163.9	3	154.6
75	3	161.6	3	150.6
80	2	159.9	2	145.7
85	2	154.3	2	145.7
90	2	154.3	2	145.7
95	2	154.3	2	145.7
100	2	154.3	2	145.7
105	2	154.3	2	145.7
110	2	152.2	2	145.7
115	2	152.2	2	145.7
120	2	151.4	2	145.6
125	2	148.1	2	142.8
130	2	148.1	2	140.6
135	2	148.1	2	140.6
140	1	139.7	1	136.5

#### 6. 考察および検討

制約距離がない場合、補正データの方が、3.2kmほど距離が短  
かった。また、車両数については両者にさほどの差異は見られな  
い。総走行距離について言えば、補正データの方が常に短かった。  
制約距離に対する両者の変化の傾向は同様である。

VRPを解いて、必要車両台数と大まかな総走行距離を知ると言  
う観点からは、補正したデータで十分であると言える。しかし、  
構成されたルートについては、両者には大きな違いがあった。実  
際の運行計画を立てるには道のりデータ、さらには所要時間デー  
タが大切であることが分かった。

#### 【参考文献】

- [1] 園川隆男、伊藤謙治：“生産マネジメントの手法”、朝倉書店、  
1996年
- [2] 福島雅夫：“数理計画入門”、朝倉書店、1996年
- [3] 田村 直：「時刻制約及び容量制約付き配送路決定問題  
—パラメータ最適化による解法」  
東京理科大学工学部経営工学科卒業論文(1990年)