コンビニエンスストアにおける商品補充問題の最適化についての研究

2000MM052 三輪 修生　　2000MM106 山本 恒義
指導教員　鈴木 敦夫

1　はじめに

1.1　背景

コンビニエンスストアの商品は多品種. 配送頻度は高く，商品によっては時間指定もありますが，物流システムも進化し，今は仕分け，ピッキングなどのすべての作業がコンピュータ管理のもと自動化されている．そして各商品に合わせた最適な配送と効率化の実現のために，メーカー，ベンダーの系列を超えた配送網 (共配化) も整備されている．情報と商品は「店舗→本部→ベンダー→共配センター→店舗…」と流れている．各チェーンとも，店舗，本部，メーカー，ベンダー，物流部門をネットワークし，情報を共有している．各店舗は双方向の情報交換が可能なホストコンピュータを通じて，本部から配信される天気情報，イベント情報，過去のデータをもとに，商品を発注する．この発注情報に応じて，商品が共配センターやカテゴリー別の配送センターに集められ，必要な日時に各店舗に配送される．

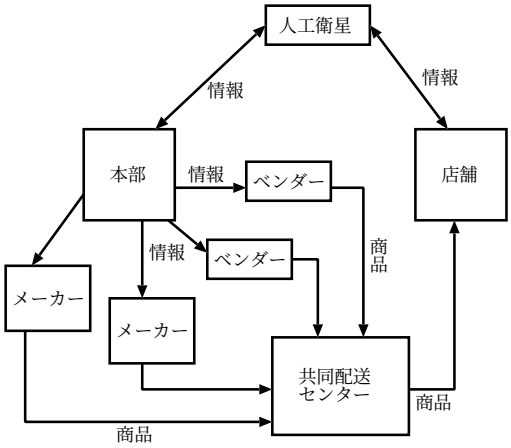

図 1　情報と商品の流れ

共配化が行われる前は温度帯ごとに，さらにメーカーごとに配送車が分かれており，配送車はどうしても多くなっていた．こうした問題を解決するために，共配が行われるようになった．配送システムの整備に伴い，コンビニの 1 店舗当たりの配送車両台数は，年々減少した．創成期は，1 店舗当たり 1 日 70 台だった車両台数は，今では 7 分の 1 の 10 台にまで減らすことができた．これは配送コストの削減はもちろんのこと，排気ガスによる二酸化炭素排出量の削減ができるなど，環境問題にも貢献している．

このように多くの効果をあげている共配にコンビニエンス業界は注目し，今後もさらに共配に力をいれるようである．しかし，1 つ 1 つの商品がどのようなルートで各店舗まで配達されるかについては，ほとんど考えられていないのが現実である．

1.2　問題へのアプローチ

我々はコンビニエンス業界ではあまり考えられていない共配センターから，各店舗を効率よく回る最適配送ルートについて考えることにする．それにより燃料費，必要トラック台数を最小化しそれにかかる維持費 (人件費を含む) を削減し，物流コスト最小化を最終目的として研究することにする．

そこで本研究では物流コスト最小化にあたり，まず最適配送ルートを配達路問題としてセービング法を用いて求める．とりあえずここではトラック 1 台は 1 トリップのみで，センターに戻り違うトリップを配送することは考えない．さらに時間的制約も無視する．しかし，ここで求めたものは最小ルートにすぎない．あくまで我々の最終目的は物流コストの最小化である．それにあたり，次に制限時間をいっぱいまで使い配送する配送ルートとその際必要とされるトラック台数を求める．以上を考慮して最終的には燃料費＋維持費 (人件費を含む) を最小化する最適配送ルートを求めていくこととする．

2　問題の説明

最適配送ルートを求め，必要トラック台数を求めるにあたり，条件として以下のものがある．

- 配送する際の条件

-トラックの積載可能量は 2 トンまで.
-何時から何時までに配達をし終えないといけない時間的制約が存在する.

3　配達路問題

各店舗をまわる最適配送ルートを配達路問題として解く．しかし，ここでは時間的制約を無視して，積載量のみの制約で考えることにする．

3.1　問題の定義

配達する場合、定められた地点をすべてまわって出発点に戻らなければならない。全ての点を通る閉路を巡回路という。各店に配る量 (a_i) が与えられていて、車の容量 (b) がそれら a_i の和よりも小さいとき

$$a_i \leqq b \qquad (i = 2, 3, 4, \cdots n)$$

$$\sum_{i=1}^{n} a_i > b$$

の場合は、配達量の和が車の容量をこえないように、グループ (トリップ) にわける。一つのグループを配達し終えるとセンターに戻る。その際、回数を台数として考える.

3.2 記号の定義

I：需要点の集合 (任意の i,j は i,$j \in I$)
a_i：各店舗への配達量
b：車の容量
トリップ：共同配送センターを含む閉路で、それに含まれる点への配達量の和が車の容量を越えないものとする。(一回の配達でまわることができる)
d_{ij}：点 i から点 j へ行くときの距離
Z：トリップの総走行距離の和
N_T：トリップの最初の配達地点の集合
T_i：最初の配達地点が $i \in N_T$ であるトリップ
r_i：T_i の最後の配達地点
q_i：T_i の配達量の和
p_i：点 i の次に通る点の番号

3.3 解法 (セービング法)

点 1(共配センター) と他の各点 (店舗) を往復する $(n-1)$ 個のトリップの組から出発し，2 個のトリップを連結することを繰り返して，よりよいトリップの組を見つける. つまり、トリップの距離の和が最小となるものを求める。このとき、各トリップに含まれる点への配達量の和が車の容量を越えてはいけない.
はじめは

$$N_T = \{2, 3, \cdots, n\}, r_i = i(i \in N_T), q_{1i} = a_{1i}(i \in N_T),$$

$$Z = \sum_{i=2}^{n} (d_{1i} + d_{i1})$$

である.
$\{q_{1i}+q_{1j}\} = \{a_{1i}+a_{1j}\} \leqq b(i \neq j)$ であれば点 i から 1(共配センター) に戻らず，点 j に直接行くことにより，二つのトリップを一つにすることができる．これを点 i に点 j を連結するという．このとき Z は,

$$S_{ij} = d_{i1} + d_{1j} - d_{ij}$$

だけ減少する．S_{ij} を点 j を点 i に連結させるときのセービング値という．点 j を点 i に連結すると，N_T から j が除かれ

$$r_i = j \qquad q_{1i} = a_{1i} + a_{1j}$$

となる.
$i \in N_T$，$j \in N_T$ が存在する限り，その中で T_i に T_j を連結するときのセービング値を求め，それらを連結させていく．この方法をセービング法という．セービング法は，一度連結したものは，その後切断せれることがないので，この解法は近似解を与える.

3.4 手順

1. 初期値の設定
   N_T = 2,3,$\cdots$,N
   $r_i = i \in N_T$，$q_{1l} = a_{1i}(i \in N_T)$，$p_i = 1(i \in N_T)$
   $Z = \sum_{i=2}^{n}(d_{1i}+d_{i1})$
2. セービング値の計算
   S_{ij}=d_{i1}+d_{1j} − $d_{ij}(i \in N_T,\ j \in N_T,\ i \neq j)$
3. T_i に T_j を連結するときのセービング値が最大であるものを求める.
$M_s = [(i,j) \mid i \in N_T,\ j \in N_T,\ i \neq j,\ q_{1i} + q_{1j} \leqq b]$
M_s が空であれば終了．空でなければ
   $\max(i,j) \in M_s\ S_{r_i j} = S_{r_i j}$
となる (i,j) を求める.
4. T_i に T_j を連結する.
   $N_T = N_T - i$，K=r_i，P_k=j，r_i=r_j，q_{1i}=q_{1i}+q_{ij}
   $Z = Z - S_{kj}$

3 の手順に戻る.
この手順に従って pascal で書かれたプログラムを c 言語変換ソフトを用いて c 言語に書き換え解く.

3.5 使用するデータ

某コンビニエンスストアの配送データをもとにセービング法を解く. このメーカーは自社で共配センターを保有していて, 全国のあらゆるところに存在する. 愛知県には中部定温センター, 豊川定温センターそして豊田常温センターの 3ヶ所が存在し, 東海 3 県に加え静岡県の一部を地区ごとに区分けして配送している. 本研究はその中の中部定温センターの一部の地区で考える.

表 1

中部定温センター	愛知県西部 107 店舗 岐阜県 68 店舗, 三重県 56 店舗
豊川定温センター	愛知県東部 79 店舗 静岡県一部の店舗
豊田常温センター	愛知県, 岐阜県, 三重県の全店舗 静岡県一部の店舗

使用するのは愛知県東部地区 107 店舗である. 現在の配送システムでは, 定温センターからはトラック 1 台平均 7 店舗, 常温センターからは平均 10 店舗配送している. 愛知県東部地区の 107 店舗の各店舗に配達する量は機密資料ということで入手することができなかったので一様乱数を用いて求めたものを使い考えることにする．さらにセンターからの距離, 店舗間の距離はゼンリン地図ソフトを用いて経度, 緯度を調べ,Miclosoft Excel 2000 を使って求めた直線距離である.

3.6 実行結果

表 2　実行結果

総配達量 (kg)	店舗	走行距離 (m)
1771.7	1-100-99-102-101-57-58-103-1	39487.85
1488.1	1-62-107-97-96-59-1	35995.27
1792.7	1-46-48-61-78-79-60-91-1	83446.80
1658.1	1-5-55-56-37-40-38-1	27111.74
1618.0	1-39-41-22-21-24-32-1	21152.4
1770.3	1-42-105-106-66-67-86-87-1	67710.66
1775.5	1-83-84-64-65-81-82-30-1	98451.58
1619.9	1-50-85-45-44-47-43-1	58125.12
1593.4	1-71-75-70-72-77-74-1	46755.59
1278.5	1-35-29-28-31-1	15200.07
1774.6	1-63-73-88-89-90-76-27-1	44657.48
1557.6	1-54-104-92-80-53-52-1	48311.17
1628.7	1-23-49-2-3-25-26-1	33419.25
1741.5	1-15-18-51-19-20-11-10-1	37346.03
1603.9	1-34-36-4-6-7-33-1	21046.54
1742.8	1-17-13-14-16-9-12-8-1	31012.95
1762.7	1-69-98-68-108-93-95-94-1	34884.72

総走行距離 = 744115.20 (m)

4　時間を考慮した最適配送ルートについて

最終目的である，コスト最小化にあたり，セービング法では，総走行距離を最小にするということをまず第一に考えたため時間的制約を一切無視して最適配送ルートを求めたが，そこで求めたものは，積載的制約は十分に満たし，無駄のないものであるが，時間的にはまだ空きがり改善の余地があるのではないかと考えられる．燃料費よりトラックの維持費の方が明らかにコストがかかるので，ここではセービング法で求めた配送ルートに時間的な枠の上限をいっぱいまで使って，トラックの台数を減らすことを目的として考えることにする．

4.1 問題の定義

配達する場合，ここでも 4.1 と同様に定められた地点をすべてまわり，かつ制限時間内に配達し終え出発点に戻らなければならない．各点に配る量 (a_i) が与えられていて，車の容量 (b) が a_i の和よりも大きくならないようにグループ (トリップ) に分けられている．4.1 ではこのトリップの数だけトラックの台数が必要であった．しかし，ここでは 1 つのトリップの要する時間の合計が時間内 (H 以下) であれば

$h_{1i}+h_{ij}+\cdots h_{n1} \leqq H$(任意の i,j,n)

トリップ同士をつなげていくことができる．しかし，荷物を積むために一度センターに戻らなければならない．そうして新たにつくられたトリップの数をトラックの台数とする．

4.2 記号の定義

3.2 に加え

H：配達完了の制限時間

h_{ij}：i から j に行くのに要する時間

4.3 解法と手順

トリップの総和 (Z) が小さいものから

trip(1),trip(2),trip(3),$\cdots$,trip($n-1$),trip(n) とする．

- trip(n) に trip(1) の点を連結させる場合…(1)

1.trip(n) の枝 (トリップの両端) の最小のものから連結

2.trip(n) の枝 (トリップの両端) の最大のものから連結

- trip(1) に trip(n) の点を連結させる場合…(2)

1.trip(1) の枝 (トリップの両端) の最小のものから連結

2.trip(1) の枝 (トリップの両端) の最大のものから連結

最小のトリップ trip(1) に最大のトリップ trip(n) を始点，終点の枝の小さい方を連結させる．

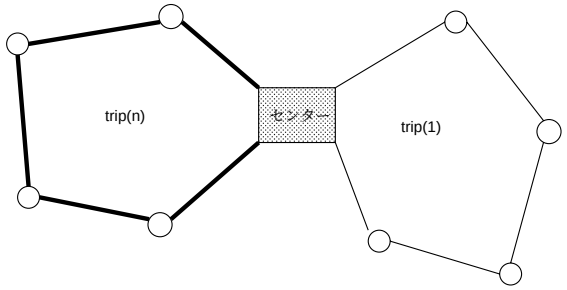

図 2

連結させた t 点は trip(1) に含まれ，trip(n) から除かれる．

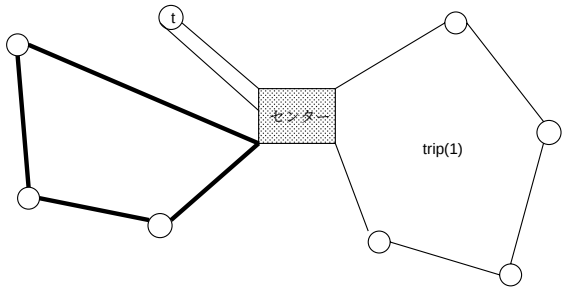

図 3

次の点に進むことが可能ならば，次の点に進みこれを繰り返し,trip(n) が空となれば終了．

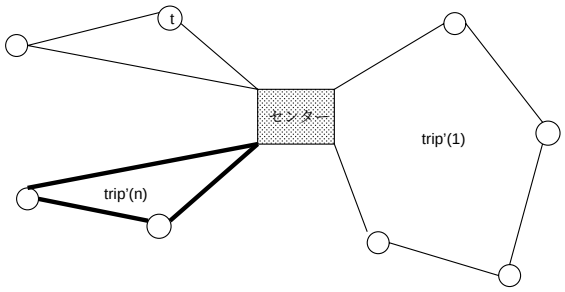

図 4

不可能ならば trip'(1) となり，そして trip(n) から取り除かれた点と連結していた点とセンターとを結び新たな trip'(n) を形成する．

次に小さなトリップ trip(2) に trip'(n) の連結を同様に行う．連結後 trip'(n) が空になれば終了．

一つ以上のトリップが空となるまで繰り返して終了．もしくは．trip^k(n) が他の全てのトリップと連結が不可能な場合，改善が困難であるとみなされるので終了．

4.4 実行結果

以上の方法で実際に 3 章でセービング法を用いて求めたルートとトラック台数の変更を行った. 解を求めるにあたり時間的な制約をその時間内で走ることのできる距離に変換して考えた. まずはじめに 1 台のトラックに対する配送距離の制限を 100000m〜120000m で考えた. その中の最良の結果は以下表のとおりである. その際, その制限の中には商品の荷降ろしの時間は加味していない.

表 3　結果 (1) の 1 (100,000m)

	ルート	走行距離 (m)
1	1-83-84-64-65-81-82-30-1 (他のトリップと連結不可)	98451.58
2	1-46-48-61-78-79-60-91-1- 35-29-28-31-1	98646.87
3	1-42-105-106-66-67-86-87-1- 33-7-6-4-36-34-1	88757.20
4	1-50-85-45-44-47-43-1-32-24-21- 22-41-39-1-5-1	95467.61
5	1-54-104-92-80-53-52-1-30-40- 37-56-55-1-17-13-1	99243.14
6	1-71-75-70-72-77-74-1-8-12- 9-16-14-1-26-25-1	98730.84
7	1-63-73-88-89-90-76-27-1- 23-49-2-3-1-94-1	91019.88
8	1-100-99-102-101-57-58-103-1-69- 98-68-108-93-95-1-62-107-97-96-1	98878.66
9	1-15-18-51-19-20-11-10-1-59-1	64808.17
		z=834003.78

4.5 考察

この実行結果中の 4 つの解法パターンの中で走行距離が最も少なく, トラックの台数が最も多く削減することができたのは,(1) の 1 の大きいトリップに小さいトリップを小さい枝から分割して連結させるパターンであった. 大きい枝, 小さい枝については様々な結果がでてしまいどちらが最良のものであるかということについては, 今後検証すべきものではありますが, 連結させるトリップについては, 小さいトリップを分割し他の大きいトリップに連結させていく方法が最良のものであった. これは連結させる際には, 一度配送センターに戻り再び配送センターからの配達を考えええると, 小さいトリップは配送センターから近い店舗が多いため走行距離の増加が最小限に抑えられたためこのような結果になったのではないであろうか. また連結させていった結果, 一度配送センターに戻り一店舗のみを配送して再び配送センターに戻るようなルートが結果として表れてしまった. これは明らかに新しくできたトリップには不適当なものであり, これについても改良すべきである.

5 おわりに

我々は本研究においてコスト削減を目指し、まずセービング法を用いて走行距離最小のルートを求めた. 次にそのルートをもとに時間的な制限の上限いっぱいまで使い, トラックの台数を減らすことによりさらなるコスト削減を目指した. その結果走行距離はセービング法によって求まったものより 15 %弱増加したもののトラックを大幅に減らすことができたのでコスト削減には役立つものであると考えられる. さらに今後, 環境保全のためのトラックに転換する場合においての初期投資も少なくすむであろうと思われる. しかし, 本研究ではまだ問題点も多く, これを改善することにより, より実用的な最適配送ルートが得られるはずである.

参考文献

[1] 勝浦 永江 :“ 配送ルートの研究 ”南山大学経営学部情報管理学科卒業論文,(1995).

[2] 久保 幹雄 ：ロジスティック工学，朝倉書店 (2001).

[3] 永野 三郎，長島 忍，吉村 伸：Pascal プログラミング，東京大学出版会 (1987).

[4] サークルケイ・ジャパン株式会社：物流に関する参考資料.

[5] 清水 忠昭，菅田 一博：C 言語のススメ，サイエンス社 (1994).

[6] プログラム変換ソフト：http://www.ee.t-kougei.ac.jp/ nisimiya/Linuxkanren/Settei/p2cinst.html