Journal of the Operations Research
Society of Japan
Vol. 39, No. 3, September 1996

分割配送路問題 ―ラグランジュ緩和を利用した解法について―

毛利　裕昭　　久保　幹雄　　森　雅夫　　矢島　安敏
三菱総合研究所　　東京商船大学　　東京工業大学

(受理 1994 年 10 月 19 日 ; 再受理 1995 年 11 月 6 日)

和文概要　配送路問題 (Vehicle Routing Problem –VRP–) は,「デポ（配送拠点）から配送先のノードに商品等を配送する最小コストのルートを求める」という問題である．この問題は古くから研究がさかんに行なわれており，標準的 VRP の基本的な条件としては，以下のものが挙げられる．第１に１つのルートでの積載量が車両の容量を超えないこと，第２に車両数（ルートの数）が上限を超えないこと，第３に配達先のノードは１台の車両で１度だけ配送が行なわれること等である．本論文では VRP の基本条件である第３の条件を緩和した「分割配送路問題」と呼ばれる複数の車両によるノードの配送を許す問題を考える．このような条件を考えることにより車両の積載効率が上り，必要な車両の数（固定費用）を減少させる可能性が高まる．この分割配送路問題に関する研究は少なく，本論文では新たな定式化を行ない，Fisher and Jaikumar のアイディアにしたがって問題を，車両が配送を行なうノードおよびその配送量を決定する問題と担当ノードが決定した段階で各車両の配送経路を決定する問題に分解することにより数理計画ベースの新たな解法を与えている．

1.　はじめに

配送路問題 (Vehicle Routing Problem -VRP-) は、「デポ (配送拠点) から配送先のノードに商品等を配送する最小コストのルートを求める」（ここでルートとはデポを出発しデポに配送車両がもどるまでの経路である) という問題である。この問題は古くから研究がさかんに行なわれており、以下のようなものが標準的 VRP の基本的な条件として挙げられる Assad [1]。

1. １つのルートでの積載量が車両の容量を超えない。
2. 車両数 (ルートの数) が上限を超えない 。
3. 配送先のノードは１台の車両で１度だけ配送が行なわれる。
4. 配送コストは配送時間 (距離)、車両台数などの関数である。

このような標準的 VRP の研究が長年進められさまざまな社会効果をあげてきた。その一方、近年ではトラック不足、運転手不足、過小積載（もしくは空荷）トラックの大量発生といった輸送業硬直化の問題や NO_x 値問題に代表される環境問題が発生している。

こうした現実の要請に合わせて様々な VRP の問題が研究されるようになっている。本論文では積載量の少ない車の発生を防ぐことにより上記の物流問題の一部が解決されることを目的として VRP の基本条件である第３の条件を緩和した問題を考える。この問題は複数の車両で１つのノードが配送されることから「分割配送路問題（Split Delivery Vehicle Routing Problem ）」とよばれる。このような条件を考えることにより車両の積載効率が上り、必要な車両の数 (固定費用) を減少させる可能性が高まる。また、距離 (時間) コストに関して最適

解である場合には、その値は基本条件における解の値以下のコストになるはずである。とくに需要量がトラックの容量に対して比較的大きい場合に非常に有効であると考えられる。

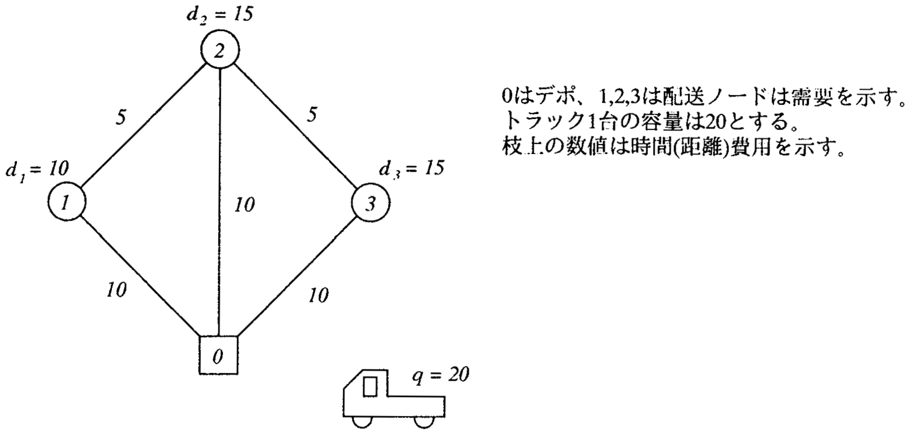

図 1: 問題例

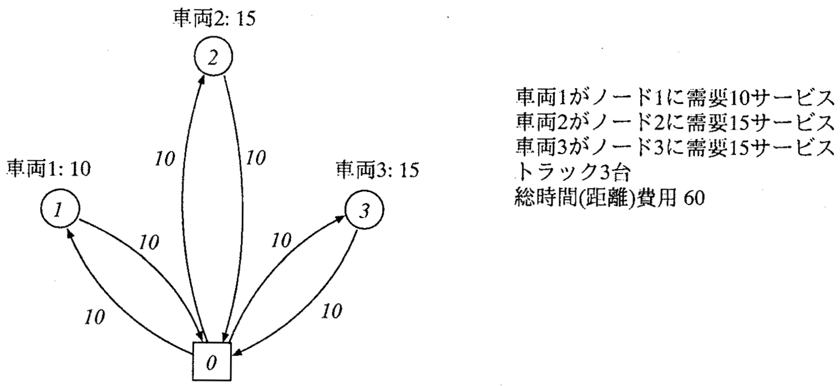

図 2: 基本条件のみを考慮した問題例の解

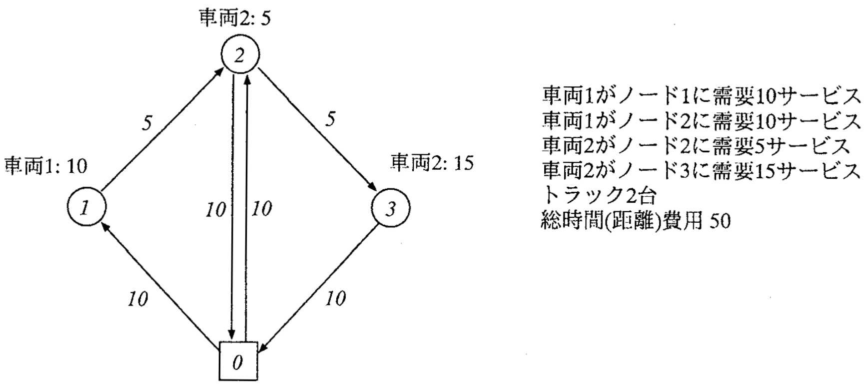

図 3: 分割配送を許した問題例の解

図 1の問題での例を見ると基本条件の解（図 2）ではすべての配送すべきノードに対して 1 台ずつ車を割当て、3 台を使用しなければならないのに対して分割配送の解（図 3）では 2 台

である。また、距離費用も分割配送を行なったものの方が10低い値を出している。

この分割配送に関する研究は少なく、現状では鈴木ら[9]による水売り行商人問題を利用した分枝限定法による厳密解法（この解法はノード数が少ない時は有効であるが、大規模な問題を解くことは不可能である）、Dror and Trudeau [2] のローカルサーチによる解法がある程度にとどまっている。本論文では、新たな定式化を行ない、Fisher and Jaikumar [3] のアイディアにしたがって問題を、車両が配送を行なうノードおよびその配送量を決定する問題 (Phase 1) と担当ノードが決定した段階で各車両の配送経路を決定する問題 (Phase 2) とに分解することにより数理計画ベースの新たな解法を与えている。（標準的 VRP ではこのような問題分割による解法が精度の良い解を出すことが知られている。）ただし、本論文では車両台数をできる限り最低にした上で、距離コストを最小化することを目的としている。

本論文は以下のような構成で記述されている。2章では、分割配送路問題に対する定式化を行ない、さらにどのようにして (Phase 1), (Phase 2) の問題に分割し、いかなるフレームワークでこの問題を解くのかを示す。以降の章では (Phase 2) はいわゆる巡回セールスマン問題（Travelling Salesman Problem -TSP-）であるため、本問題に固有の問題である (Phase 1) に対する議論を中心に行なう。3章では、(Phase 1) のラグランジュ緩和問題の性質と解法を述べる。4章では、ラグランジュ緩和問題の解を修正することにより実行可能解を作り、これを近似解として採用する方法について述べる。この方法は分枝限定法に比べ厳密性では劣るものの、大規模な問題に対して簡便に解を求めることができる。この考え方は Fisher [4] で述べられている。5章では、標準的な VRP と本論文で提案したアルゴリズムによる分割配送路問題の解を実験的に比較し分割配送による効果を考察する。

2. 分割配送路問題の定式化と解法のフレームワーク

本章では2.1で分割配送路問題の定式化を示し、2.2でこの問題を2つの問題に分割する方法について述べる。さらに、2.3でこの問題分割を利用しどのようなフレームワークで分割配送路問題を解くのかを示す。

2.1. 分割配送路問題の定式化

分割配送路問題は以下の様に定式化される。

(定数)

- m ： 車両の台数
- n ： 配送先のノードの数
- i, j ： ノードを示す添字 (0はデポを示し、その他は配送先を示す)
- k ： 車両を示す添字
- d_i ： ノード i における需要量
- c_{ij} ： ノード i からノード j への距離（時間）費用
- q_k ： 車両 k の容量

(変数)

- x_{ijk} ： ノード i からノード j へ車両 k が直接訪問したか否かを示す0−1変数
- y_{ik} ： ノード i へ車両 k によって配送が行なわれたか否かを示す0−1変数
- z_{ik} ： ノード i で車両 k による配送量を示す実数変数

(定式化)

(2.1) $$\min \quad \sum_{k=1}^{m}\sum_{i=0}^{n}\sum_{j=0}^{n} c_{ij}x_{ijk}$$

subject to

(2.2) $$\sum_{k=1}^{m} y_{0k} = m$$

$$1 \leq \sum_{k=1}^{m} y_{ik} \qquad i = 1,\ldots,n$$

(2.3) $$z_{ik} \leq d_i y_{ik} \qquad i = 0,\ldots,n,\ k = 1,\ldots,m$$

(2.4) $$\sum_{k=1}^{m} z_{ik} = d_i \qquad i = 1,\ldots,n$$

(2.5) $$\sum_{i=1}^{n} z_{ik} \leq q_k \qquad k = 1,\ldots,m$$

(2.6) $$\sum_{i=0}^{n} x_{ijk} = y_{jk} \qquad j = 0,\ldots,n,\ k = 1,\ldots,m$$

(2.7) $$\sum_{j=0}^{n} x_{ijk} = y_{ik} \qquad i = 0,\ldots,n,\ k = 1,\ldots,m$$

(2.8) $$\sum_{(i,j)\in S\times S} x_{ijk} \leq |S|-1 \qquad S \subset \{1,\ldots,n\}$$

$$2 \leq |S| \leq n-1$$

$$k = 1,\ldots,m$$

(2.9) $$y_{ik} = 0 \ \text{ or } \ y_{ik} = 1 \qquad i = 0,\ldots,n,\ k = 1,\ldots,m$$

(2.10) $$x_{ijk} = 0 \ \text{ or } \ x_{ijk} = 1 \qquad i = 0,\ldots,n,\ j = 1,\ldots,n,\ k = 1,\ldots,m$$

(2.11) $$z_{ik} \geq 0 \qquad i = 1,\ldots,n,\ k = 1,\ldots,m$$

(2.1) の目的関数は、距離費用最小化である。(2.2) は車の台数は m 台であり、各配送先は、1 台以上の車両により配送するという制約である。ここで車両台数 m は車両によらず容量が一定つまり $q_k = q \quad (k = 1,\ldots,m)$ のときは車両の最低台数は $\lceil \sum_{i=1}^{n} d_i/q \rceil$ で与えられ、その台数で配送が可能である。一般のＶＲＰではＮＰ困難な問題であるビンパッキングを解かなければ最低台数はわからない (Garey[5]) が、分割配送ではこのように簡単にわかる。(2.3) は、車両 k がノード i に配送する量は、y_{ik} が 0 であるなら 0 であり、配送が行なわれる場合は z_{ik} は d_i 以下であることを示す。(2.4) は、ノード i に対する、全車両の総配送量が需要量に一致することを示す。(2.5) は、車両 k による配送の総量は車両容量 q_k を超えることができないことを示している。(2.6) は、もしノード j が車両 k によって配送されている場合には、ノード j に関してどこかのノードから車両 k が来ることを意味している。(2.7) も同様で、ノード i が車両 k によって配送されている場合には、車両 k がノード i からどこかのノードへ行くことを意味している。(2.8) は部分巡回路を除く制約である。(2.9) は、y_{ik} は車両 k がノード i に配送するか否かを示す。(2.10) は車両 k がノード i からノード j へ直接配送をおこなかったか否かを示す。(2.11) は z_{ik} は配送量が非負であることを示す。

2.2. 2フェーズ分割配送路問題の定式化

ここでは、前節で行なった定式化を直接に解くことが困難であるため、

g_{ik}：車両 k がノード i へ配送を行なうために要する費用

なるものを導入することにより、以下のような2つのフェーズに問題を分割することを考える。

(Phase 1) 各車両が配送を行なうノードとそのノードでの配送量を決定する問題
(Phase 2) 担当ノードが決定した段階で各車両の配送経路を決定する問題

Fisher and Jaikumar [3] は、標準的な VRP をこのように2フェーズにし (Phase 1) に一般割当問題を適用することで精度の高い近似解を導くことに成功している。同様にして分割配送路問題に関する (Phase 1) を定式化すると以下のようになる。

$$\min \quad \sum_{k=1}^{m}\sum_{i=1}^{n} g_{ik}y_{ik}$$

subject to

$$
\begin{array}{lrl}
 & z_{ik} \le d_i y_{ik} & i=1,\ldots,n,\ k=1,\ldots,m \\
(2.12) & \displaystyle\sum_{k=1}^{m} z_{ik} = d_i & i=1,\ldots,n \\
 & \displaystyle\sum_{i=1}^{n} z_{ik} \le q_k & k=1,\ldots,m \\
 & y_{ik}=0 \ \text{ or } \ y_{ik}=1 & i=1,\ldots,n,\ k=1,\ldots,m \\
 & z_{ik} \ge 0 & i=1,\ldots,n,\ k=1,\ldots,m
\end{array}
$$

この問題における g_{ik} の決定方法に関して Fisher and Jaikumar [3] で述べられているように、各車両ごとにシードとなるノードをあらかじめ決定しておき (重心ノードのようなもの)、デポとシードノードに対して配送ノードを挿入した挿入距離を g_{ik} として利用している。具体的には、$l_k(k=1,\ldots,m)$ をシードノードとして

$$g_{ik} = \min[c_{0i}+c_{il_k}+c_{l_k0},\ c_{0l_k}+c_{l_ki}+c_{i0}] - (c_{0l_k}+c_{l_k0})$$

である。これが、ユークリッド空間上であるなら以下の様になる。（図 4 参照）

$$g_{ik} = c_{0i}+c_{il_k}-c_{0l_k}$$

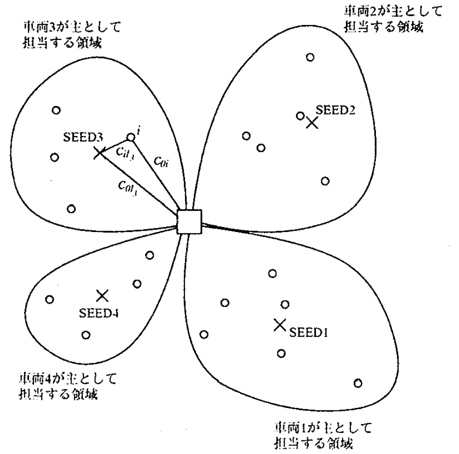

図 4: Fisher and Jaikumar の挿入距離計算のイメージ

一方、(Phase 2) の定式化は、以下の様になる。

$$\min \quad \sum_{k=1}^{m}\sum_{i=0}^{n}\sum_{j=0}^{n} c_{ij}x_{ijk}$$

subject to

$$\begin{aligned}
\sum_{i=0}^{n} x_{ijk} = y_{jk} \qquad & j=0,\ldots,n,\ k=1,\ldots,m \\
\sum_{j=0}^{n} x_{ijk} = y_{ik} \qquad & i=0,\ldots,n,\ k=1,\ldots,m \\
\sum_{(i,j)\in S\times S} x_{ijk} \le |S|-1 \quad & S\subset\{1,\ldots,n\} \\
& 2\le |S| \le n-1 \\
& k=1,\ldots,m \\
y_{ik}=0 \ \text{ or } \ y_{ik}=1 \quad & i=0,\ldots,n,\ k=1,\ldots,m \\
x_{ijk}=0 \ \text{ or } \ x_{ijk}=1 \quad & i=0,\ldots,n,\ j=0,\ldots,n,\ k=1,\ldots,m
\end{aligned}$$

これは、k について分解が可能であるので、m 個の TSP と考えることができる。TSP に対しては、様々な角度から研究が行なわれているが、本論文では既存の手法を利用するにとどめ、深い議論は行なわないものとする。

2.3. 解法のフレームワーク

(Phase 1) は、小規模の問題であれば分枝限定法を用いることにより厳密に解くことが可能である。(ちなみにこの連続緩和問題は輸送問題である) しかし、大規模の問題を解くことは難しいため、本論文ではラグランジュ緩和を用いそれから実行可能解を生成し近似解とする方法を採用した。この考え方は、Fisher [4] によるものである。そのフレームワークを図 5に示す。

このアルゴリズムは大きく４つの段階に分けられる。第１段階では劣勾配法により最良のラグランジュ緩和解を生成する。第２段階では、ラグランジュ緩和解の性質を利用して車両の担当ノードを変えずに配送量の調整を行なう。第３段階ではラグランジュ緩和解から実行可能解を生成する。 最終段階では各車両ごとに担当ノードについて TSP を解く。第１段階の詳細は次章で、第２段階と第３段階は第４章で詳細を述べる。

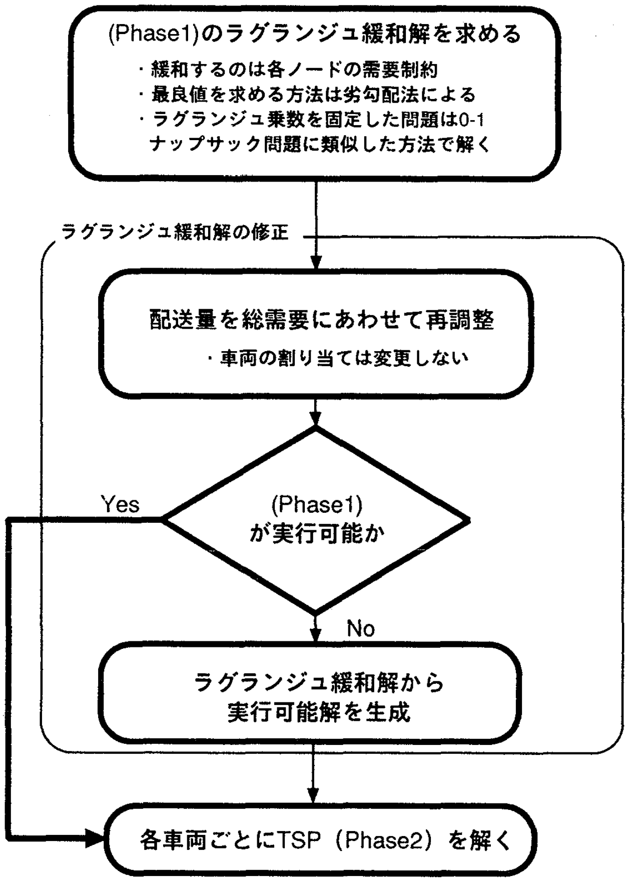

図 5: 解法のフレームワーク

3. 分割配送路問題 (Phase 1) のラグランジュ緩和問題

本章では、3.1で (Phase 1) のラグランジュ緩和問題とその特性を示し、3.2ではラグランジュ緩和問題の性質と解法を示す。3.3では、最良のラグランジュ緩和問題の値を得るための方法について述べる。

3.1. 分割配送路問題 (Phase 1) のラグランジュ緩和問題

ここでは、(Phase 1) の問題に対して容量制約である (2.12) をラグランジュ緩和する問題を考える。ラグランジュ乗数を $\{\pi_i\}$ として、この問題は以下の様に定式化できる。この問題を以降では $\mathrm{L}(\pi)$ と呼ぶ。

$\mathrm{L}(\pi)$:

$$(3.1)\quad f_L(\pi) = \min \sum_{k=1}^{m} \sum_{i=1}^{n} (g_{ik} y_{ik} - \pi_i z_{ik}) + \sum_{i=1}^{n} \pi_i d_i$$

subject to

$$(3.2)\qquad z_{ik} \le d_i y_{ik} \qquad i = 1, \ldots, n,\ k = 1, \ldots, m$$

$$(3.3)\qquad \sum_{i=1}^{n} z_{ik} \le q_k \qquad k = 1, \ldots, m$$

$$(3.4)\qquad y_{ik} = 0 \ \text{ or } \ y_{ik} = 1 \qquad i = 0, \ldots, n,\ k = 1, \ldots, m$$

$$(3.5)\qquad z_{ik} \ge 0 \qquad i = 1, \ldots, n,\ k = 1, \ldots, m$$

(3.1) で $\sum_{i=1}^{n} \pi_i d_i$ は定数であり、(3.1) ∼ (3.5) の問題は式の形により車両 k ごとに分解することが可能である。よって以降では必要に応じて車両 k ごとの問題で考え、添字 k を省略する。これをラグランジュ子問題 LS(π) と呼ぶことにする。

LS(π):

$$f_{LS}(\pi) = \min \quad \sum_{i=1}^{n}(g_i y_i - \pi_i z_i)$$

subject to

$$\begin{aligned} \sum_{i=1}^{n} z_i &\le q \\ z_i &\le d_i y_i && (i = 1, \ldots, n) \\ y_i = 0 \ \text{ or } \ y_i &= 1 && (i = 1, \ldots, n) \\ z_i &\ge 0 && (i = 1, \ldots, n) \end{aligned}$$

3.2. ラグランジュ子問題 LS(π) の解法

3.1で述べた混合整数計画問題 LS(π) が 0-1 ナップサック問題と類似した性質を持つことを以下で示し、その性質を利用して解を効率的に得ることができることを示す。ここで類似した性質とは、「連続緩和問題が連続ナップサック問題である」、「 問題を解く前処理で変数の固定が効果的に行なわれる」等の性質である。

まず、LS(π) の連続緩和問題がナップサック問題であることを示すため LS(π) の連続緩和問題に対して以下の補題が成立することを証明する。

補題 1 問題 LS(π) の連続緩和問題 CLS(π) の解を $(\overline{y}, \overline{z})$ とすると

$$\overline{z}_i = d_i \overline{y}_i$$

(証明)

$\overline{z}_i^* < d_i \overline{y}_i^*$ なるものが存在するとする。$\overline{y}_i^{**} = \overline{z}_i^* / d_i \ (i = 1, \ldots, n)$ とすると $(\overline{y}^{**}, \overline{z}^*)$ は実行可能解であり、$g_i > 0$ と、$g_i \overline{y}_i^{**} < g_i \overline{y}_i^*$ より目的関数をさらに小さくしうるので $(\overline{y}, \overline{z})$ が最適解であることに矛盾する。∎

この補題を用いて用いて問題 CLS(π) を書き直すと

CLS(π):

$$f_{CLS}(\pi) = \min \quad \sum_{i=1}^{n}(g_i/d_i - \pi_i) z_i$$

subject to

$$\begin{aligned} \sum_{i=1}^{n} z_i &\le q \\ 0 \le z_i &\le d_i \quad (i = 1, \ldots, n) \end{aligned}$$

となり連続ナップサック問題に変形できることが容易に理解される。以降では、上記の問題の効率を

$$\gamma_i = \frac{g_i}{d_i} - \pi_i$$

で定義する。また、$\{\gamma_i\}$ は、負のもののみを考慮して非減少の順に並んでいるものとして議論する。さらに

$$p = \min\{j : \sum_{i=1}^{j} d_i > q\}$$

としておく。この準備に基づいて以下の命題が導かれる。

命題 1

問題 LS(π) の連続緩和問題 CLS(π) の解を $(\overline{y}, \overline{z})$ とすると

$$\begin{array}{lll} \overline{y}_i = 1, & \overline{z}_i = d_i & (i = 1, \ldots, p-1) \\ \overline{y}_i = 0, & \overline{z}_i = 0 & (i = p+1, \ldots, n) \\ \overline{y}_p = (q - \sum_{j=1}^{p-1} d_j)/d_p, & \overline{z}_p = (q - \sum_{j=1}^{p-1} d_j) & \end{array}$$

(証明)

上記の議論より明らか。■

また、変数固定の前処理に関して以下の定理が成立する。これを利用することによって問題の規模の縮小ができる。

定理 1

LS(π) のある暫定値を $\tilde{v}$ とするならば、

(3.6) $J^* = \{j : j \in \{1, \ldots, n\}, f_{CLS}(\pi) - \tilde{v} \geq \mu_j^*\}$

(3.7) $\mu_j^* = \begin{cases} (\gamma_j - \gamma_p)d_j & (j = 1, \ldots, p) \\ -(\gamma_j - \gamma_p)d_j & (j = p+1, \ldots, n) \end{cases}$

とした時に、

(i) $j \in J^*$ かつ $j < p$ ならば暫定値よりよい実行可能解は $y_j = 1$ である。

(ii) $j \in J^*$ かつ $j > p$ ならば暫定値よりよい実行可能解は $y_i = 0$ である。

(証明)

以下では $f(\pi | y_j = a)$ は、y_j を a に固定した上での最適値を示すものとする。

(i) $j < p$ であるとき、

$$\begin{aligned} f_{LS}(\pi | y_j = 0) &\geq f_{CLS}(\pi | y_j = 0) \\ &\geq \sum_{i=1}^{p-1} \gamma_i d_i - \gamma_j d_j + \gamma_p(q + d_j - \sum_{i=1}^{p-1} d_i) \\ &= f_{CLS}(\pi) - (\gamma_j - \gamma_p)d_j \end{aligned}$$

(3.7)、(3.6) より

$$= f_{CLS}(\pi) - \mu_j^* \geq \tilde{v}$$

よって暫定解が改善される為には $y_j = 1$ であるべき。■

(ii) $j > p$ であるとき、

$$\begin{aligned}
f_{LS}(\pi|y_j=1) &\geq f_{CLS}(\pi|y_j=1) \\
&= \sum_{i=1}^{p'-1}\gamma_i d_i + \gamma_j d_j + \gamma_{p'}(q - d_j - \sum_{i=1}^{p'-1} d_i) \\
&\qquad\qquad (p' = \min\{k|\sum_{i=1}^{k} d_i > q - d_j\} \\
&= \sum_{i=1}^{p'}\gamma_i d_i + \gamma_j d_j + \gamma_{p'}(q - d_j - \sum_{i=1}^{p'} d_i) \\
&\geq \sum_{i=1}^{p'}\gamma_i d_i + \gamma_j d_j + \gamma_{p'+1}(q - d_j - \sum_{i=1}^{p'} d_i) \\
&\vdots \\
&\geq \sum_{i=1}^{p-1}\gamma_i d_i + \gamma_j d_j + \gamma_{p}(q - d_j - \sum_{i=1}^{p-1} d_i) \\
&= f_{CLS}(\pi) + (\gamma_j - \gamma_p) d_j
\end{aligned}$$

(3.7)、(3.6) より

$$= f_{CLS}(\pi) - \mu_j^* \geq \tilde{v}$$

よって暫定解が改善される為には $y_j = 0$ であるべき。∎

これらの命題、定理の性質を利用して分枝限定法をおこなえば大規模な LS(π) を解くことが可能である。

また、LS(π) を分枝限定法で連続緩和問題を利用して解く分枝戦略は 0-1 ナップサック問題との類推で本論文では以下のような方法を採用している。
・奥行き優先法則
・効率 γ_i 最小の分枝変数優先

3.3. 劣勾配法によるラグランジュ関数 $f_L(\pi)$ の最大化

ここでは、3.1で記述したラグランジュ緩和問題 L(π) の最良値を求める方法を示す。

$f_L(\pi)$ は、π に関して微分不可能な凹関数であり、その値は (Phase 1) の下界値になっている。しかし、最良の下界値を求めるには π に関してその値 $f_L(\pi)$ を π に関して最大化しなければならない。ここでは、劣勾配法を利用することによりそれを実現する。L(π) に対する劣勾配法のアルゴリズムを以下に示す。ここではイタレーション回数 i のラグランジュ乗数ベクトルを $\pi(i) \triangleq (\pi_1(i), \ldots, \pi_j(i), \ldots, \pi_n(i))$ とする。

(アルゴリズム 3-1)

STEP1 $\pi(0) := 0$ として L($\pi(0)$) を解き、その解を $(y_{(0)}, z_{(0)})$ とする。また、(Phase 1) に関する実行可能解の (例えば分割配送を許す sweep 法の過程等で求めた) 値を u^* として準備しておく。$i := 0$ とおく。

STEP2
$$\pi_j(i+1) := \pi_j(i) + \zeta_{(i)}(d_j - \sum_{k=1}^{m} z_{jk}) \quad (j = 1, \ldots, n)$$
$$\zeta_{(i)} \triangleq \frac{\beta_{(i)}(u^* - f_L(\pi(i)))}{\sqrt{\sum_{j=1}^{n}(d_j - \sum_{k=1}^{m} z_{jk})^2}}$$
とおく。ただし、$\beta_{(i)}$ は $0 < \beta_{(i)} \leq 2$ のパラメータである。ここで $i := i+1$ とする。

STEP3 $\mathrm{L}(\pi(i))$ を解き、その解を $(y_{(i)}, z_{(i)})$ とおく。ある k が存在し $|f_L(\pi(i)) - f_L(\pi(i+k))| < \varepsilon$ であれば終了。そうでなれば、STEP2 にもどる。

(注意) STEP3 の収束条件を満たすことは一般的に計算機上は困難であるので、ある回数このアルゴリズムを回し、その中での最大値をとるのが数値計算上は一般的であり本論文での計算機実験でもそのようにしている。

4. ラグランジュ緩和解の修正

本章では (Phase 1) 問題の劣勾配法によるラグランジュ緩和解を修正することにより実行可能解を得て近似解とする方法について述べる。4.1では、ラグランジュ緩和解の特性を利用して各車両の担当ノードを維持したまま配送量のみを調整するアルゴリズムについて述べる。また、4.1のアルゴリズムを適用しても実行可能解が得られない場合において実行可能解を得る方法を4.2で述べる。

4.1. ラグランジュ緩和解の配送量割当を調整するアルゴリズム

3.3で示されたラグランジュ緩和解に対する劣勾配法による解が (Phase 1) の問題に対して実行可能ではなくとも、配送量のみを調整することにより実行可能にすることができる場合がある。

下記の例の図 6の調整前の解ではノード 3 においては需要量が超過しており、ノード 4 では不足している。しかし、この配送量は図 7の様に調整し直すことが可能である。

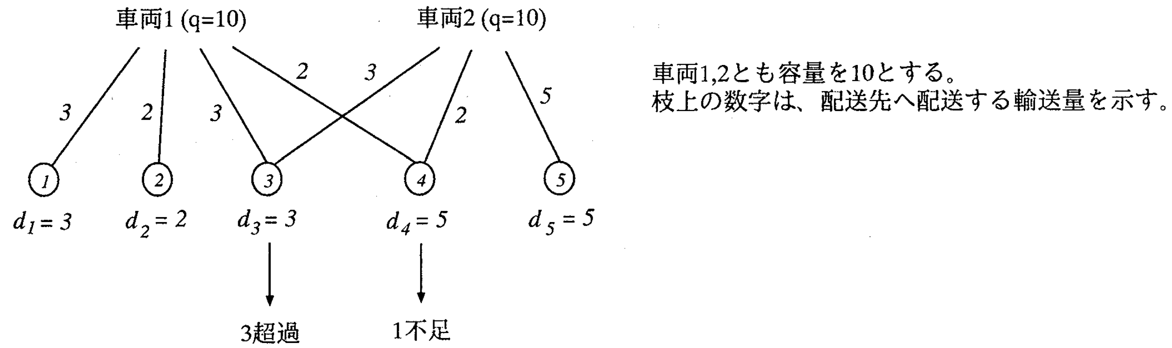

図 6: ラグランジュ緩和解の例

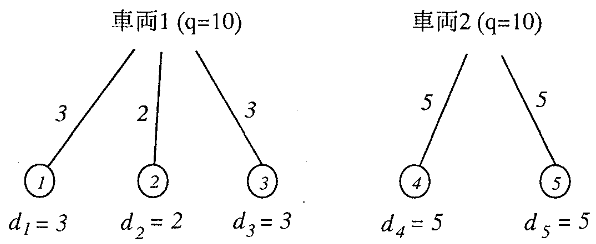

図 7: 割当調整後の解

さらに、先のラグランジュ緩和解のアルゴリズムによる解では、各車両ごとに不足する可能性があるノードが高々 1 つであることを利用することによって、効率的に再割当の処理を行なうことが可能である。以下にそのアルゴリズムを示す。記号を以下の様にさらに定義する。

以下のような二部グラフ $G(N \cup V, A)$ を定義する。

N : 配送ノードの集合

V : 車両の集合

(y^l, z^l) : ラグランジュ緩和問題の解

$A \triangleq \{(i,k) \mid y^l_{ik} = 1,\ i \in N,\ k \in V\}$ 車両と車両が配送するノードとの間にある枝集合

$SN \triangleq \left\{ i \;\middle|\; \sum_{k=1}^{m} z^l_{ik} < d_i,\ i \in N \right\}$: 需要不足のノード集合

$ON \triangleq \left\{ i \;\middle|\; \sum_{k=1}^{m} z^l_{ik} > d_i,\ i \in N \right\}$: 需要超過のノード集合

$JN \triangleq \left\{ i \;\middle|\; \sum_{k=1}^{m} z^l_{ik} = d_i,\ i \in N \right\}$: 需要をちょうど満たしているノード集合

$SN^1 \triangleq \left\{ i \;\middle|\; \sum_{k=1}^{m} y^l_{ik} = 1,\ i \in SN \right\}$: 車両一台で配送されている需要不足ノード集合

$V' \triangleq \{k \mid y^l_{ik} = 1,\ i \in SN \cup ON\}$: 需要不足・需要超過ノードを配送する車両集合

$A' \triangleq \{(i,k) \mid y^l_{ik} = 1,\ i \in SN \cup ON,\ k \in V\}$
: 需要不足・需要超過ノードとそれを配送する車両との間の枝集合

$A^1 \triangleq \{(i,k) \mid y^l_{ik} = 1,\ i \in SN^1,\ k \in V\}$
: 車両 1 台のみで配送されている需要不足ノードとそれをサービスする車両との間の枝集合

$H \triangleq \left\{ h_{ik} \mid h_{ik} = \dfrac{g_{ik}}{d_i} \text{ for } (i,k) \in A' \backslash A^1,\ h_{ik} = \infty \text{ for others} \right\}$
: 枝上の費用行列

(アルゴリズム 4-1)

STEP1 $(y, z) := (y^l, z^l)$ とし、 SN, ON の要素を N から検索して $Short, Over$ というリストに保存しておく。

STEP2 $$q_k := q_k - \sum_{i \in JN} z_{ik} \qquad \forall k \in V$$

とする。

STEP3 SN^1 の要素をリスト $Short$ から検索を行なって、$q_{\eta(i)} < d_i$ となる $i \in SN^1$ が存在するとき、y_{ik}^l, z_{ik}^l は元問題に対してこのアルゴリズムを適用しても実行不能であり終了する。(ただし $\eta(i)$ はノード i をサービスする車両を示す)。そうでなければ

$$\left.\begin{array}{lcl} z_{i\eta(i)} & := & d_i \\ q_{\eta(i)} & := & q_{\eta(i)} - d_i \end{array}\right\} \forall i \in SN^1$$

とする。

STEP4 $$G' \triangleq \left\{V' \cup \left((SN \cup ON) \backslash SN^1\right), A' \backslash A^1\right\}$$

上で輸送コストを H で定義できる輸送問題を考える。輸送問題が実行可能であるなら、その解を z'_{ik} として

$$z_{ik} := z'_{ik} \ \ \forall (i,k) \in A' \backslash A^1$$

とし

$$\left.\begin{array}{llcl} z_{ik} > 0 \ \text{のとき} & y_{ik} & := & 1 \\ z_{ik} = 0 \ \text{のとき} & y_{ik} & := & 0 \end{array}\right\} \forall (i,k) \in A' \backslash A^1$$

として終了。

実行不可能な場合は y_{ik}^l, z_{ik}^l は元問題に対してこのアルゴリズムを適用しても実行不可能であり終了する。

4.2. ラグランジュ緩和問題の解から実行可能解を得るアルゴリズム

4.1で示したアルゴリズムが効果的であるのは限られた場合であるので、ここではアルゴリズム 4-1 を適用しても実行可能解が得られない場合にラグランジュ緩和解から実行可能解を得るアルゴリズムについて述べる。

(アルゴリズム 4-2)

STEP1 $(y, z) := (y^l, z^l)$ (ラグランジュ緩和解) において容量不足のノード SN を探す。もし $SN = \phi$ であるなら終了。

STEP2 $y_{ik} := 0 \ \forall k \in V$ となるノード i が存在するならば、そのようなすべての i について $k' = \arg\min_k g_{ik}$ を求め $y_{ik'} := 1$ とする。

STEP3

$$\begin{cases} y_{ik} = 1 \text{のとき} & h_{ik} = \dfrac{g_{ik}}{d_i} \\ y_{ik} = 0 \text{のとき} & h_{ik} = \infty \end{cases}$$

とする費用行列 (h_{ik}) をもつ輸送問題を解き、実行可能であるならその解を改めて z_{ik} と置き換え、$z_{ik} > 0$ なる (i,k) に対して $y_{ik} := 1$、その他の場合を $y_{ik} := 0$ として終了する。実行不能なら STEP4 へ。ただし、STEP4 を一度実行している場合は STEP5 へ。

STEP4 G 上の連結グラフを $G_i = ((V_i \cup N_i), A_i)$ とし、

$$\sum_{k \in V_i} q_k < \sum_{i \in N_i} d_i$$

となるすべての G_i に対して

$$(i', k') = \arg \min_{i \in N_i, k \in V \backslash V_i} \{g_{ik} \mid y_{ik} = 0, i \in N_i, k \in V \backslash V_i\}$$

を求め、$y_{i'k'} := 1$ として STEP3 に戻る。

STEP5 $$(i', k') = \arg \min_{i \in N, k \in V} \{g_{ik} \mid y_{ik} = 0, i \in N, k \in V\}$$

を求め、$y_{i'k'} := 1$ として（またはこの STEP5 の操作を何度かおこなって）STEP3 へ戻る。

5. 計算機実験結果

本論文で提案した分割配送路問題に対するアルゴリズムによる効果を標準的な（分割しない）VRP と計算機実験により比較する。ただし、分割配送に関しては車両台数 m を分割配送可能な最低台数 $\lceil \sum_{i=1}^{n} d_i/q \rceil$ に固定して比較実験している。本論文で提案されたアルゴリズムは、UNIX 上 C 言語で記述され、SUN SPARC station IPX 上でインプリメントされた。巡回セールスマン問題を解くアルゴリズムとしては Held and Karp [7] の動的計画法による厳密解法を採用している。

ここでの問題は、ノードの数が 11、21、50、75、100 とした。作者らが作成した 11 ノードの場合を除き、ノードとデポの位置は、各々 TSPLIB から eil22.vrp, eil51.vrp, eil76.vrp, eil101.
vrp を利用している。各ノードの需要量については、車両の容量に対して小さければ距離減少の効果が小さく、比較的大きければ距離減少の効果が大きいと直観的に考えられる為そのことを検証するためにすべてのノードの需要が 30% の場合、60% の場合、30%、45%、60% をほぼ同数ずつ混ぜた場合についてテストを行なっている。

分割しない VRP の解としては、厳密解が明らかに分かるものを除き（ノードの需要が車両の 60% の場合）、近年有効な方法とされるタブーサーチのアルゴリズムで Gendreau ら [6] によるアルゴリズム（このアルゴリズムは様々なテスト問題の記録を更新している）を採用している。

パラメータに関しては、劣勾配法におけるイタレーション回数は 150 回で打ちきりとし、乗数 $\beta_{(i)}$ はイタレーションごとには変化させず、実験結果としては 0.01-0.06 のものを採用している。

実験結果を表 1～表 3 に示す。（但し、表中のノード数はデポを除いたものである）

表 1: 各ノードにおける需要量が車両容量の 60% の場合
（劣勾配法イタレーション 150 回）

	11 ノード	21 ノード	50 ノード	75 ノード	100 ノード
分割配送した場合の総距離	492	894	1752	2634	3474
分割配送しない場合の総距離	592	1166	2396	3622	4972
分割配送による距離減少率	16.89%	23.33%	26.88%	27.23%	30.13%
分割配送した場合の車両台数	7	13	30	45	60
分割配送しない場合の車両台数	11	21	50	75	100
分割配送による台数減少率	36.36%	38.10%	40.00%	40.00%	40.00%
CPU 時間 (sec)	0.90	2.43	27.95	202.00	425.02

表 2: 各ノードにおける需要量が車両容量の 30%、45%、60% をほぼ同数とした場合
(劣勾配法イタレーション 150 回)

	11 ノード	21 ノード	50 ノード	75 ノード	100 ノード
分割配送した場合の総距離	400	698	1430	2039	2744
分割配送しない場合の総距離	408	751	1460	2174	2966
分割配送による距離減少率	2.00%	7.06%	2.05%	6.21%	8.09%
分割配送した場合の車両台数	5	10	23	34	45
分割配送しない場合の車両台数	6	13	26	41	51
分割配送による台数減少率	16.67%	23.08%	11.54%	17.07%	11.76%
CPU 時間 (sec)	0.90	1.55	15.20	75.65	292.09

表 3: 各ノードにおける需要量が車両容量の 30% の場合
(劣勾配法イタレーション 150 回)

	11 ノード	21 ノード	50 ノード	75 ノード	100 ノード
分割配送した場合の総距離	296	529	1031	1526	1987
分割配送しない場合の総距離	296	530	1047	1527	2040
分割配送による距離減少率	0.00%	0.19%	1.53%	0.07%	2.60%
分割配送した場合の車両台数	4	7	15	23	30
分割配送しない場合の車両台数	4	7	17	25	34
分割配送による台数減少率	0.00%	0.00%	11.76%	8.00%	11.76%
CPU 時間 (sec)	2.08	5.02	14.83	57.28	118.45

ノードの需要が車両容量の 60% の場合では総距離の減少効果が 17% から 30% 程度で大きく、台数の減少も 37% から 40% である。一方、ノードの需要が車両容量の 30% の場合では、総距離における効果は 60% の場合とに比較してそれほどではない。100 ノードの場合で 2.6% である。11 ノードと 21 ノードの場合においてはこの場合では殆んど解全体として同程度である。しかし、車両の減少率に関しては 50 ノード以上の場合では 8 〜 12% 減少している。この 2 つの中間の場合として設定した 30%、45%、60% の需要を混ぜた場合では、総距離の減少率が 2% 〜 8%、車両台数の減少が 12% 〜 23% の効果をあげている。

また、CPU time においては、50 ノード以下では 1 分以内であるし、100 ノードの場合でも最高 7 分程度である。これは十分に実用的な計算時間であると思われる。

6. 結語

本論文では分割配送路問題に対して新しい定式化を与え、問題分割による数理計画ベースの新解法を提案した。分割された問題のうち本質的問題 (Phase 1) に対してラグランジュ緩和解を 0-1 ナップサック問題に似た特性を利用して効果的にもとめ、その解から実行可能解を求めるというアルゴリズムである。数値実験によりこの解法は、需要量が車両容量に対して比較的大きい場合とくにノード数が多い場合に有効であることが分かった。需要が小さくノード数も少ない場合には、分割しない VRP と解が同程度であるという直観のとおりの結果が示され

た。また、この解法は実用に十分耐えうる速さで機能することも確かめられた。

謝辞

本論文の実験における輸送問題のプログラムをご提供くださいました（株）サリオンシステムズリサーチの桧垣正浩氏、および有益なご指摘をいただきました審査員の方に深く感謝致します。

参考文献

[1] Assad, A. A. : Modeling and Implimentation Issues in Vehicle Routing. *Vehicle Routing: Methods and Studies* (Golden and Assad eds.), North-Holland, (1988), pp. 7 – 45.

[2] Dror, M. and Trudeau, P. : Split Delivery Routing. *Naval Research Logistics*, Vol.37 (1990), pp. 383 – 402.

[3] Fisher, M. and Jaikumar, R. : A Generalized Assignment Heuristic for Vehicle Routing. *Networks*, Vol.11 (1981), pp. 109 – 124.

[4] Fisher, M. : A Application Oriented Guide to Lagrangian Relaxation. *Interfaces*, Vol.15 (1985), pp. 10 – 21.

[5] Garey, M. R. and Johnson, D. S. : *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, (1979).

[6] Gendreau, M. , Hertz, A. and Laporte, G. : A Tabu Search Heuristic for the Vehicle Routing Problem. *Management Science*, Vol.40 (1994) , pp. 1276 – 1290.

[7] Held, M. and Karp, R. M. : A Dynamic Programming Approach to Seqencing Problems. *SIAMJ.Appl.Math*, Vol.10 (1962), pp. 196 – 210.

[8] 今野浩, 鈴木久敏 : 整数計画法と組合せ最適化, 日科技連 (1982).

[9] 鈴木久敏, 辻 真人, 平林隆一 : 分割配達を許す配送路問題. 東京工業大学経営工学科 Technical Report No.J-5, (1987) （日本オペレーションズリサーチ学会誌 1988 年 1 月号に要約版あり）.

毛利 裕昭
〒100 東京都千代田区大手町 2-3-6
株式会社 三菱総合研究所 情報技術開発部
E-mail : mohri@mri.co.jp

ABSTRACT

A SPLIT DELIVERY VEHICLE ROUTING PROBLEM

Hiroaki Mohri
Mitsubishi Research Institute, Inc.

Mikio Kubo
Tokyo University of Mercantile Marine

Masao Mori Yasutoshi Yajima
Tokyo Institute of Technology

We consider a standard vehicle routing problem, i.e. a routing problem where all nodes (customers) should be visited only once by only one vehicle and without exceeding a vehicle's capacity. The objective function minimizes total distance traveled. In this paper, we relax the first standard problem condition. We call this problem the Split Delivery Vehicle Routing Problem (SDVRP). The term "split delivery" means, as long as the total delivery equals the demand, the demand may be satisfied using more than one vehicle. Under the relaxed condition, we are able to reduce the number of vehicles and the total distribution cost (time). Hence, it will find practical application.

SDVRP has been little researched. On a mathematical programming base, Suzuki et al. (1987) suggested an exact algorithm using a Branch and Bound Method for this problem. But it is only useful for a small number of nodes. On a local search base, Doror and Trudeau (1990) proposed heuristic algorithm. In this paper, we take another formulation. Solving this problem on mathematical programming base, we decompose it into two problems: First, a problem of which vehicle serves the node and to what extent each vehicle serves each node. Second, the problem of the route that each vehicle takes. This idea is based on Fisher and Jaikumar (1981) whose solutions is very good one for a standard vehicle routing problem. The second problem is the Traveling Salesman Problem. Therefore, the first problem is the essential one. We suggest that by method of Fisher (1985) we can generate a heuristic solution from the Lagrangean Solution for the first problem. From this we are able to solve a large size SDVRP.