

時間枠制約付き配送計画問題に対するヒューリスティック解法

岩 瀬 弘 和 *

A Heuristic Algorithm for Solving Vehicle Routing Problems with Time Windows

Hirokazu IWASE

1. はじめに

宅配便や郵便・新聞の配達、小売店への商品配送、ゴミの収集など、車両による配送・収集サービスは我々の生活において欠かせないものである。サービスを供給する側にとっては、車両（配送車）の数をできるだけ少なくし、また最短の移動距離で配送を済ませることが理想である。このように、より効率的な配送計画を求める問題を配送計画問題^[1]（VRP, Vehicle Routing Problem）という。また、サービスを受ける側にとっては都合により受け取りの時間枠を指定せざるを得ない場合もある（例えば9:00～11:00など）。このように各配送先（顧客）の時間枠を考慮する場合の問題を時間枠制約付き配送計画問題^[2]（VRPTW, Vehicle Routing Problem with Time Windows）という。VRPやVRPTWは組合せ最適化問題に属するが、これらは最悪の場合は指数オーダーの計算時間になってしまうNP-困難な問題であることが証明されており^[3]、顧客数が多くなると多項式時間では最適解を求めることができない。そのため、VRPTWの近似最適解を求めるための発見的手法としてタブーサーチ^{[4][5]}（Tabu Search）、焼きなまし法^[6]（Simulated Annealing）、蟻コロニー最適化^[7]（Ant Colony Optimization）、遺伝的アルゴリズム^[8]（Genetic Algorithms）などを適用した研究が盛んに行われている。

本論文では、比較的古典的であるがVRPTWに対する最近傍法^{[7][9]}（Nearest Neighbor algorithm）により得た解を初期解として利用し、従来手法である経路内操作、経路間操作に加え、新たな手法である経路分割・結合操作を適用することにより、解を改善するアルゴリズムを提案する。提案手法をベンチマーク問題^[10]に適用し、数値実験によりその有効性を確認する。

* Hirokazu IWASE 共通領域部（Department of General Studies）

2. 時間枠制約付き配送計画問題

配送計画問題（以下ではVRPと表記）は複数台の配送車がデポ（拠点、倉庫など）を出発して配送先（顧客など）を訪問し、デポに戻ってくる際の最適となる経路を求める問題である。ただし、配送先には複数台の配送車のうち1台が必ず1回のみ訪問する。また、ここでの最適とは経路の長さ（配送車の移動距離）を最短にすること及び使用する配送車の数を最少にすることである。デポは一般的には一つであるが、複数のデポを扱うモデルもある。VRPにおける英語論文では多くの場合において配送先を表す単語としてcustomerが用いられていることから、以下では配送先を「顧客」と呼ぶことにする。VRPでは配送車の最大数、配送車の最大積載量、顧客の総数、各顧客の位置（座標）・要求量・サービス時間があらかじめ与えられている。また、顧客間の距離は一般的にユークリッド距離を用いる。図1は顧客数=7、配送車数=3の場合の概念図である。 C_0 はデポであり、 C_1 から C_7 は顧客である。

VRPにおいて各顧客への訪問可能時刻（最早時刻から最遅時刻まで）が決められている問題を時間枠制約付き配送計画問題（Vehicle Routing Problem with Time Windows、以下ではVRPTWと表記）という。例えば図1の場合、VRPでは経路 $C_0 \rightarrow C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_0$ と逆順の経路 $C_0 \rightarrow C_3 \rightarrow C_2 \rightarrow C_1 \rightarrow C_0$ は同じ解として扱うことが可能であるが、VRPTWでは経路 $C_0 \rightarrow C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_0$ が時間枠制約を満たしても逆順の経路 $C_0 \rightarrow C_3 \rightarrow C_2 \rightarrow C_1 \rightarrow C_0$ が時間枠制約を満たすとは限らないため経路の順序が重要な意味を持つ。

VRPTWのベンチマーク問題^[10] C101（顧客数100、最大配送車数25）におけるデータの一部を表1に示す。顧客番号0はデポを意味する。全ての配送車はデポの最早時刻0に出発し、デポの最遅時刻1236までには必ず戻って来なければならない。顧客への到着時刻がその顧客の最早時刻よりも前の場合には最早時刻になってからサービスを開始する。一方、到着時刻が最遅時刻を超えることは許されない。また、サービス終了後は直ちに次の顧客へ移動する。

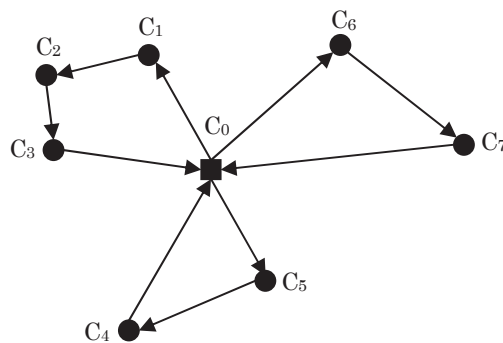


図1 配送計画問題（VRP）の概念図

表1 VRPTWのベンチマーク問題C101^[10]におけるデータの一部

顧客番号	位置座標 X	位置座標 Y	要求量	最早時刻	最遅時刻	サービス時間
0	40	50	0	0	1236	0
1	45	68	10	912	967	90
2	45	70	30	825	870	90
3	42	66	10	65	146	90
4	42	68	10	727	782	90
5	42	65	10	15	67	90

VRPTWを数学的モデルとして表現すると次のようになる。

入力情報：

Q ：配送車の最大積載量

N ：顧客の集合（0 はデポを表す）

V ：配送車の集合

r_i ：顧客 i の要求量， $i \in N$

s_i ：顧客 i に対するサービス時間

e_i ：顧客 i に対するサービス開始の最早時刻

l_i ：顧客 i に対するサービス開始の最遅時刻

d_{ij} ：顧客 i から顧客 j への移動距離， $i \in N, j \in N$

t_{ij} ：顧客 i から顧客 j への移動時間

b_{jk} ：配送車 k が顧客 j に対して行うサービスの開始時刻， $j \in N, k \in V, b_{jk} = \max\{e_j, b_{ik} + s_{ij} + t_{ij}\}$

決定変数：

$$x_{ijk} = \begin{cases} 1 : \text{配送車 } k \text{ が顧客 } i \text{ への訪問直後に顧客 } j \text{ へ訪問する場合, } i \neq j \\ 0 : \text{上記以外の場合} \end{cases}$$

モデル：

$$\text{Min} \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ijk} \quad (1)$$

$$\text{s.t.} \quad \sum_{k \in V} \sum_{i \in N} x_{ijk} = 1 \quad \forall j \in N \setminus \{0\} \quad (2)$$

$$\sum_{k \in V} \sum_{j \in N} x_{ijk} = 1 \quad \forall i \in N \setminus \{0\} \quad (3)$$

$$\sum_{i \in N} r_i \sum_{j \in N} x_{ijk} \leq Q \quad \forall k \in V \quad (4)$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0 \quad \forall h \in N \setminus \{0\}, \forall k \in V \quad (5)$$

$$\sum_{j \in N \setminus \{0\}} x_{0jk} = 1 \quad \forall k \in V \quad (6)$$

$$\sum_{i \in N \setminus \{0\}} x_{i0k} = 1 \quad \forall k \in V \quad (7)$$

$$x_{ijk}(b_{ik} + s_i + t_{ij}) \leq b_{jk} \quad \forall i \in N, \forall j \in N, \forall k \in V \quad (8)$$

$$e_i \leq b_{ik} \leq l_i \quad \forall i \in N, \forall k \in V \quad (9)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in N, \forall j \in N, \forall k \in V \quad (10)$$

式(1)は全配送車の総移動距離を最小化することを表している。式(2)から式(10)は制約条件である。式(2)と式(3)は、1台の配送車が各顧客に1回のみ訪問することを意味する。式(4)では、いずれの配送ルートにおいても顧客の総要求量が配送車の最大積載量以下に制限している。式(5)は、配送車が顧客を訪問した後（サービス終了後）、直ちに次の顧客に移動することを意味する。式(6)と式(7)はいずれの配送車もデポを出発してデポに帰着することを意味する。式(8)は、顧客*i*から顧客*j*への移動において時刻 $b_{ik} + s_i + t_{ij}$ の前に顧客*j*に到着できないことを表している。式(9)では、いずれの顧客*i*におけるサービス開始時間も時間枠 $[e_i, l_i]$ の間に制限している。式(10)は整数的制約である。

3. ヒューリスティック解法

VRPTWはNP-困難な問題であり、その厳密解（最適解）は問題サイズが大きくなると（顧客数が多くなると）求めることができない。そのため近似最適解を求めることが現実的な解決法となるが、従来手法の最近傍法^{[7][9]}のみでは最適解の得られる可能性は低く、その解に対して改良を施す必要がある。そこで本研究では、次の手順により近似最適解を求める。

- [Step1] 従来手法を利用して初期解を求める。これを現在の最良解 S_{better} とする。
- [Step2] S_{better} に対して改良操作を行い、得られた解を S とする。
- [Step3] Step2で得た解 S が S_{better} よりも良い解ならば、 S_{better} を S に置き換える。
- [Step4] 終了条件（最大繰り返し回数や最大計算時間など）を満たしていれば終了する。
そうでなければStep2に戻る。

Step1における初期解の求め方については3.1節で、Step2における改良操作の方法については3.2節でそれぞれ述べる。また、本研究ではStep4における最大繰り返し回数=50回を終了条件とする。

3.1 初期解

VRPTWの近似解法としてセービング（節約）法^{[11][12]}、挿入法^{[9][13]}、タブーサーチ^{[4][5]}、焼きなまし法^[6]、蟻コロニー最適化^[7]、遺伝的アルゴリズム^[8]など、様々な手法が提案・適用されている。ここでは文献^{[7][9]}に記されている最近傍法（Nearest Neighbor algorithm）を利用する。以下にそのアルゴリ

ズムを示す。

- [Step1-1] 配送車の番号 $k = 1$ とする。
- [Step1-2] 配送車 k の経路 $R_k = \emptyset$ (空集合) とする。
- [Step1-3] デポ C_0 を現在の顧客 C_{now} とする ($C_{\text{now}} \leftarrow C_0$)。
- [Step1-4] 全ての顧客が訪問済みならば終了する。
- [Step1-5] 経路 R_k に対して制約条件 (最大積載量及び時間枠) を満たす顧客が存在する場合には、その中から「最もコストの小さい (後述)」顧客 C_{next} を選び、経路 R_k に追加する。
存在しない場合には、Step1-7 に進む。
- [Step1-6] C_{now} を C_{next} に更新する ($C_{\text{now}} \leftarrow C_{\text{next}}$)。Step1-4 に戻る。
- [Step1-7] 配送車の番号 $k \leftarrow k+1$ とする。Step1-2 に戻る。

Step1-5の「コスト」については次の2種類を考える。ただし、式中の各変数は前節の数学的モデルにおける入力情報での定義と同じものとする。

(1) NN-S (Nearest Neighbor heuristic by Solomon ^[9])

$$\text{コスト} = \alpha \times d_{ij} + \beta \times \max \{e_j - (b_{ik} + s_i), 0\} + \gamma \times (l_j - (b_{ik} + s_i + t_{ij}))$$

ただし、三つのパラメータ α , β , γ は $\alpha + \beta + \gamma = 1.0$ を満たす0以上の実数である。

本研究では、 $\alpha = 0.2$, $\beta = 0.7$, $\gamma = 0.1$ とする。

(2) NN-G (Nearest Neighbor heuristic by Gambardella ^[7])

$$\text{コスト} = \max \{b_{jk} - (b_{ik} + s_i), d_{ij}\} \times (l_j - (b_{ik} + s_i))$$

4.1節で示すように、NN-S、NN-Gのいずれの方法においてもSolomonのベンチマーク問題^[10]に対して最適解を得ることはできない。そのため、これらの方法で得た解を初期解として次節で述べる操作により解の改良を図る。

3.2 改良操作

本研究では、[1]経路内操作、[2]経路間操作、[3]経路分割-結合操作の三つの改良操作を利用する。このうち[1][2]についてはVRPTWに関する多くの研究で使われているが、[3]については単純な操作であるものの他の研究では見られない操作方法である。

[1] 経路内操作

経路内操作として図2に示すような4種類の操作を利用する。図中の■はデポ、●は顧客である。また、実線の矢印は顧客同士が直接つながっている経路、点線の矢印は間に別の顧客が存在する可能性のある経路を意味する。経路内操作の場合は一つの経路の中で顧客の訪問順序が変わるだけなので、操作により得られる解候補に対して配送車の最大積載量による制約条件を考慮する必要はない。一方で時間枠による制約条件については解候補を得るたびにチェックする必要がある、特に訪問時刻

が最遅時刻を超える場合にはその時点で解候補から除外するものとする。

[2] 経路間操作

経路間操作として図3に示すような2種類の操作を利用する。経路間操作の場合は顧客の訪問順序が変わるだけでなく経路内の顧客の数や組合せが変わるので、時間枠及び最大積載量の制約条件をチェックする必要がある。

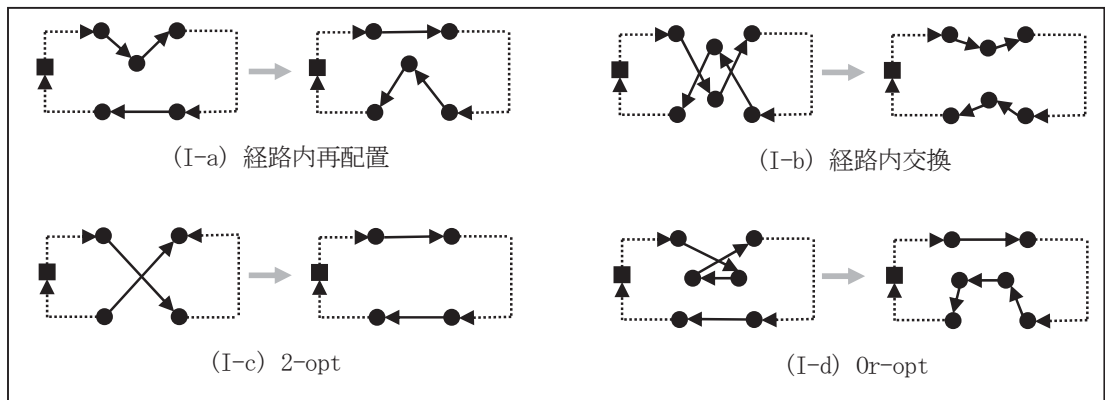


図2 経路内操作

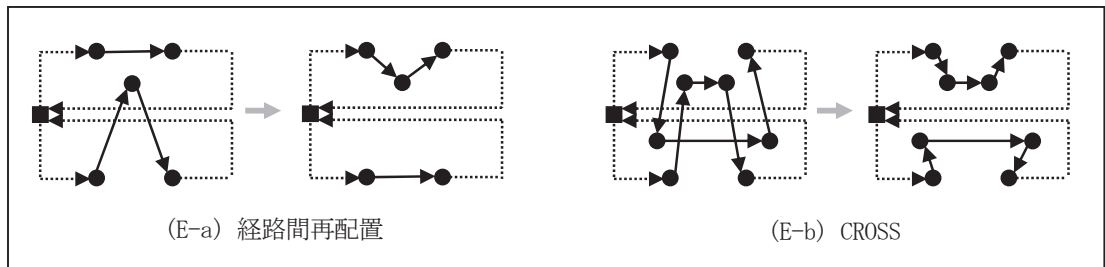


図3 経路間操作

[3] 経路分割-結合操作

図4に示すような経路分割-結合操作を提案する。一つの経路の中で顧客間距離が長いものを切断することにより二つの小さな経路を作り、その後、任意の二組の経路に対して結合可能なもの（制約条件を満たすもの）を結合するという操作である。切断する経路の選択方法としては、全ての配送車の経路の中から長い順に m (≥ 1)本を選ぶものとする。このとき、最適解を構成する経路についても切断されてしまう可能性があるため、結合操作の回数 n については m 以上の値に設定する必要がある。

Step2においてこれらの三つの操作を利用するが、それらの順序が改良操作の有効性に影響を及ぼすことも考えられる。例えば「経路内操作→経路間操作→経路分割-結合操作」の順で行う場合と「経路分割-結合操作→経路間操作→経路内操作」の順で行う場合とでは操作の効果が異なるかもしれない。また、経路内操作は4種類、経路間操作は2種類あるが、これらについても順序の影響が考え

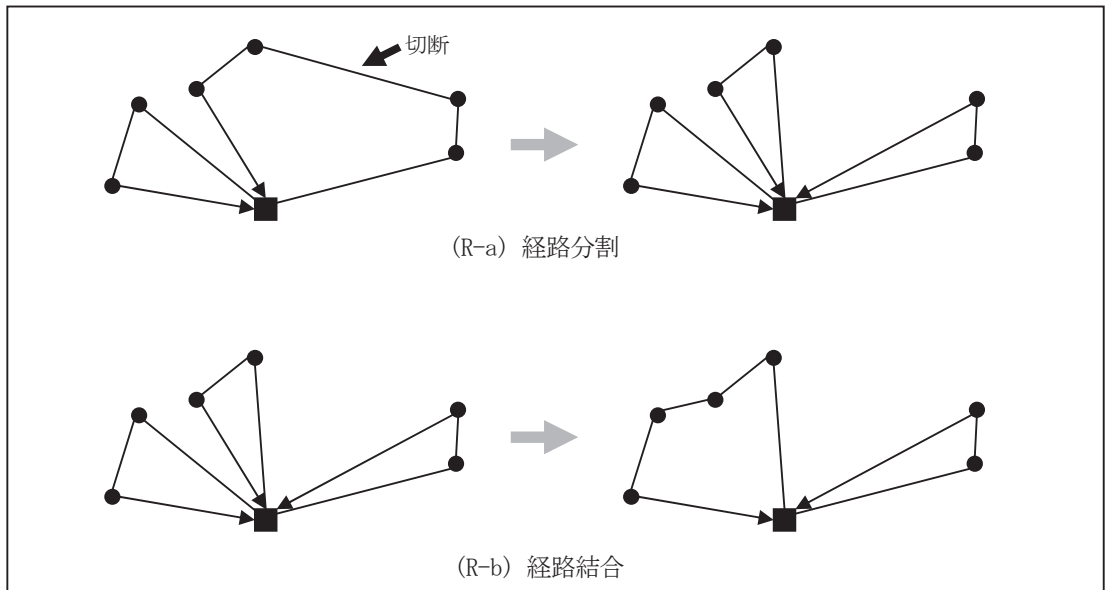


図4 経路分割-結合操作

られる。そこで、4.2節の数値実験において最適な順序に関する確認を行う。

4. 数値実験

本研究では、VRPTWのベンチマークテストとして有名なSolomonのベンチマーク問題^[10]のうち、C101～C109を利用する。いずれも顧客数=100、最大車両数=25であり、デポを中心として顧客の配置がクラスタ化された問題である。表1は問題C101のデータの一部である。他の問題に関しては顧客の配置（座標X,Y）、要求量及びサービス時間はC101と同じであるが、各顧客の時間枠（最早時刻・最遅時刻）がそれぞれの問題で異なり、解きやすい問題（C101）と解きにくい問題（C105など）が存在する。具体的には各顧客の時間枠の幅が狭い場合は有効な解を探す時点で解候補が大幅に絞られるので解きやすくなる。また、時間枠の幅だけでなく時刻も大きく影響する。例えば隣接する顧客であっても時間枠が[9:00, 10:00]と[17:00, 18:00]のように大幅にずれていれば、その顧客同士を結ぶ経路は解候補になりにくい。

4.1 初期解

C101～C109の問題に対してNN-SとNN-Gを適用した結果、得られた経路をそれぞれ図5、図6に示す。また、各解における配送車数と総移動距離（総経路長）、及びそれぞれの最適解を表2に示す。この結果から、NN-SとNN-Gを比較するとNN-Gの方が全ての問題において良い解を得ているが、それでも全ての問題において総移動距離の最適解は得られていない。

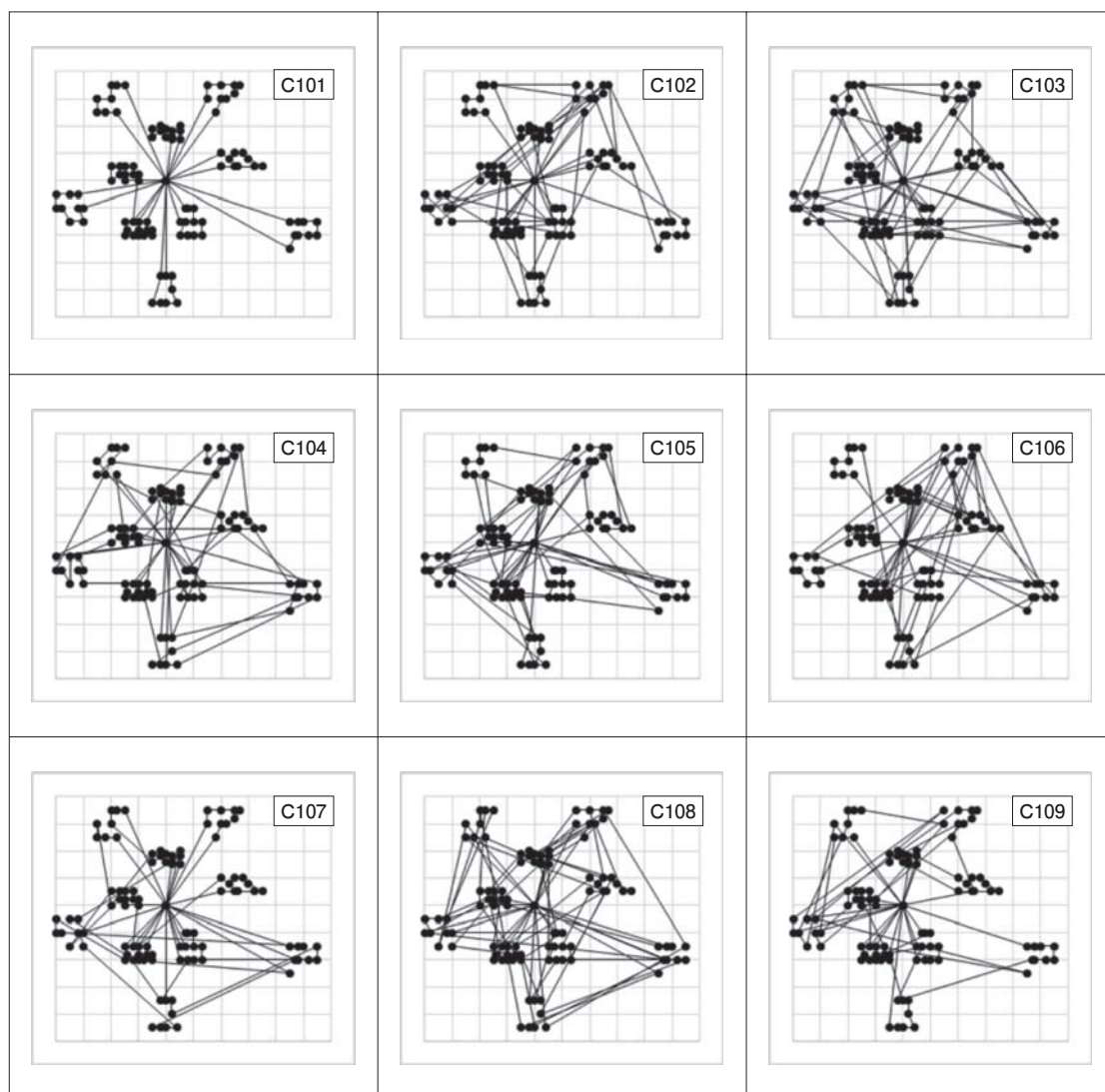


図5 NN-Sにより得た解

表2 各手法により得た解と最適解 (上段: 配送車数, 下段: 総移動距離)

	C101	C102	C103	C104	C105	C106	C107	C108	C109
NN-S	10 855.07	10 1744.66	10 2076.27	10 1743.37	12 1998.39	11 1940.28	10 1471.02	12 2619.07	10 1584.20
NN-G	10 855.07	10 1327.10	10 1478.92	10 1305.92	11 970.11	10 1017.11	10 1029.27	10 1193.48	10 1539.70
最適解	10 828.94	10 828.94	10 828.06	10 824.78	10 828.94	10 828.94	10 828.94	10 828.94	10 828.94

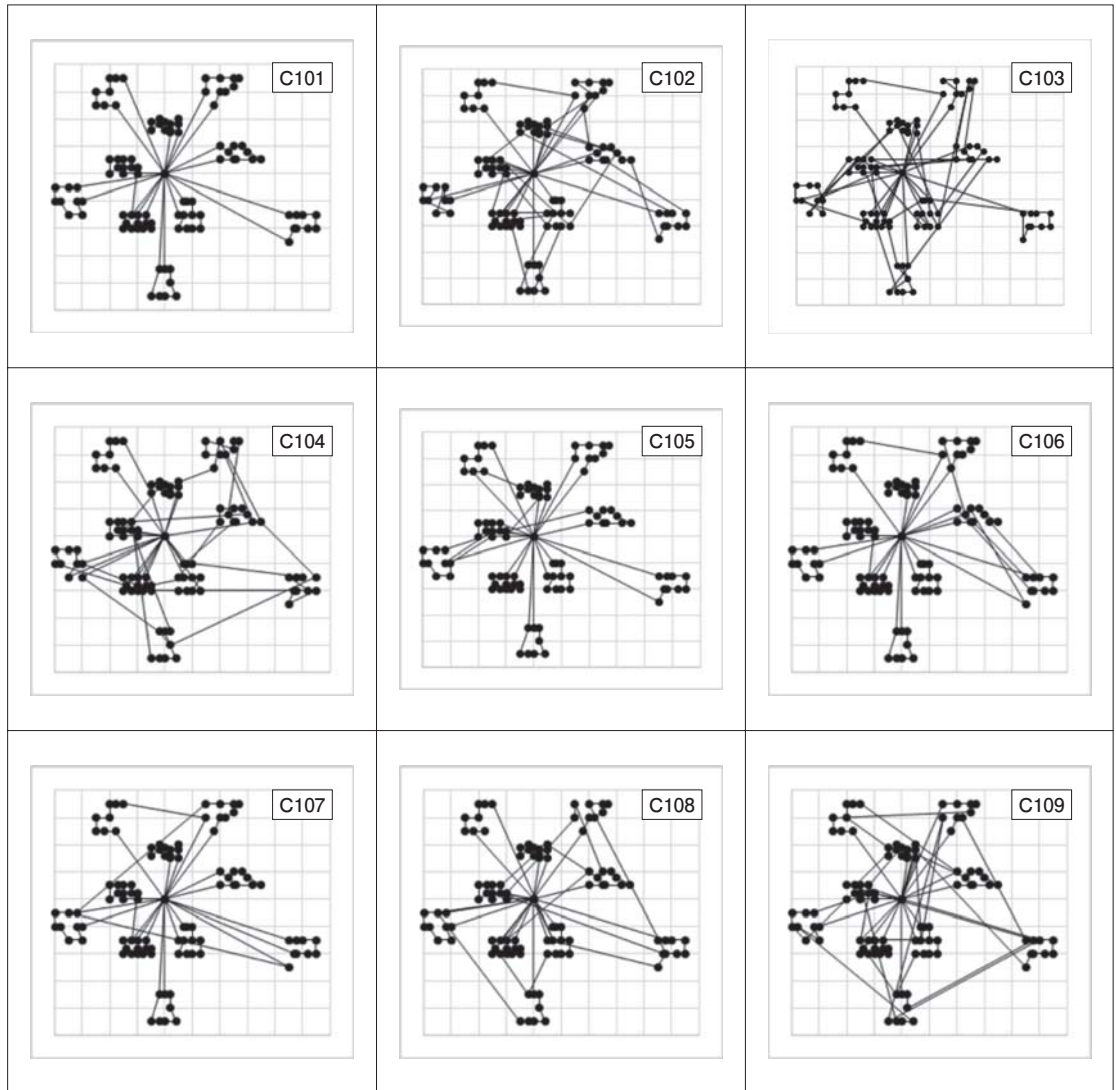


図6 NN-Gにより得た解

4.2 経路内操作・経路間操作の適用

3.2節で述べた通り、三つの操作の順序は改良操作の有効性に影響を及ぼすと考えられる。また、経路内操作の4種類、経路間操作の2種類についても、それぞれの操作内順序の影響は考慮する必要がある。そこで、「経路間操作→経路内操作」「経路内操作→経路間操作」の差異、及び経路間操作における2種類の順序の差異、さらには経路内操作において4種類の操作を全て行う場合（計24通り）と1回の経路内操作で4種類のうちランダムに選んだ一つのみを行う場合（Rnd4と表記）の差異について調べたところ、経路間操作の2種類については(E-b)→(E-a)の順が優れていることが確認できた。これを踏まえて、経路間操作については(E-b)→(E-a)に固定してその他の組み合わせについて調べたところ、表3の結果が得られた。ただし、ここでの改良操作（例えば「経路間→経路内」を1セットとして）の繰り返し回数は50回である。

表3の「経路間→経路内」「経路内→経路間」は経路内操作の4種類全てを実行した場合であり、その組み合わせは全部で24通りであるが、順序による影響は全くなく、C101では初期解に関わらず100%最適解が得られ、C106では初期解NN-Gで100%、C108では初期解NN-Gかつ「経路内→経路間」でのみ100%最適解が得られた。

一方、Rnd4の経路内操作に関しては100回のシミュレーションを行った結果であるが、「Rnd4→経路間」のほうが「経路間→Rnd4」よりも最適解が得られやすい傾向にある。また、「Rnd4→経路間」と「経路内→経路間」の比較についてはC108のみ後者のほうが優れているが、それ以外はC102、C109の最適解が得られる可能性を持っている点で前者のほうが優れているといえる。

表3 各改良操作において最適解の得られた割合 (%)

改良操作	初期解	C101	C102	C103	C104	C105	C106	C107	C108	C109
経路間→経路内	NN-S	100	0	0	0	0	0	0	0	0
	NN-G	100	0	0	0	0	100	0	0	0
経路内→経路間	NN-S	100	0	0	0	0	0	0	0	0
	NN-G	100	0	0	0	0	100	0	100	0
経路間→Rnd4	NN-S	100	3	0	0	0	0	0	0	0
	NN-G	100	4	0	0	0	100	0	54	2
Rnd4→経路間	NN-S	100	8	0	0	0	0	0	0	0
	NN-G	100	2	0	0	0	100	0	92	1

4.3 経路分割-結合操作の適用

初期解に対して「①経路分割-結合操作→②経路内操作 (Rnd4) →③経路間操作 ((E-b)→(E-a))」を最大サイクル数 (ここでは50サイクル) 実行して得られる解を最良解とする。各初期解 (NN-S、NN-G)、各分割結合条件におけるシミュレーションを全問題 (C101～C109) に対してそれぞれ100回ずつ実行した結果、最適解の得られた割合を表4に示す。表中の分割-結合欄の数値は[分割数, 結合数]である。

問題C101に関しては、初期解や分割-結合の条件によらず全ての場合において100%の割合で最適解が得られている。これは表3が示すように、いずれの初期解でも経路間操作及び経路内操作を行うことで (経路分割-結合操作がなくても) 必ず最適解が得られるためである。また、問題C107に関しても、初期解がNN-Gであれば全ての場合において最適解が得られることも同様の理由である。それ以外に100%の割合で最適解が得られるのは、問題C105に関しては初期解がNN-Gで分割結合が[3,4]以上の場合、問題C106に関しては初期解がNN-Gで分割結合が[2,3]以上の場合、問題C108に関しては初期解がNN-Gで分割数が2の場合のみである。その他の場合については、問題によって異なるものの、最適解が得られるのは確率的である。解が得られにくいものはC102、C103、C104であり、特にC104は非常に限定された条件でのみ (経路内操作における4種類の操作が特定の順序で行われる場合のみ) 最適解が得られる。各問題において得られた最適解を図7に示す。C103, C104以外はC101の最適解と同じである。

表4の網掛け部分は (確率的であるものも含めて) 全ての問題において最適解が得られた操作条件

である。この結果から、初期解NN-G、分割-結合[3,5]が比較的良好な操作条件に見えるが、問題の種類によってその良し悪しにバラツキがあるため、今後の詳細な研究が求められる。

表4 最適解の得られた割合 (%)

分割-結合	初期解	C101	C102	C103	C104	C105	C106	C107	C108	C109
[1, 1]	NN-S	100	9	0	0	0	0	0	0	3
	NN-G	100	1	0	0	0	27	100	96	0
[1, 2]	NN-S	100	7	0	0	31	23	0	4	1
	NN-G	100	2	0	0	0	23	100	96	1
[2, 3]	NN-S	100	8	4	0	71	92	100	15	69
	NN-G	100	15	4	0	28	100	100	100	0
[2, 4]	NN-S	100	5	4	0	74	91	100	9	68
	NN-G	100	15	5	1	31	100	100	100	1
[2, 5]	NN-S	100	5	2	0	75	94	100	10	69
	NN-G	100	27	1	1	23	100	100	100	0
[3, 4]	NN-S	100	5	0	0	77	69	100	19	18
	NN-G	100	16	9	1	100	100	100	96	15
[3, 5]	NN-S	100	2	3	0	72	73	100	24	27
	NN-G	100	18	12	2	100	100	100	90	34
[3, 6]	NN-S	100	5	7	0	76	77	100	16	21
	NN-G	100	17	5	0	100	100	100	93	26
[4, 5]	NN-S	100	33	6	0	74	55	100	32	12
	NN-G	100	23	31	0	100	100	100	86	17
[4, 6]	NN-S	100	34	13	0	77	50	100	49	9
	NN-G	100	17	26	0	100	100	100	94	9
[4, 7]	NN-S	100	37	12	0	72	55	100	32	15
	NN-G	100	17	26	0	100	100	100	91	12
[4, 8]	NN-S	100	22	5	0	78	58	100	25	15
	NN-G	100	19	36	0	100	100	100	89	17

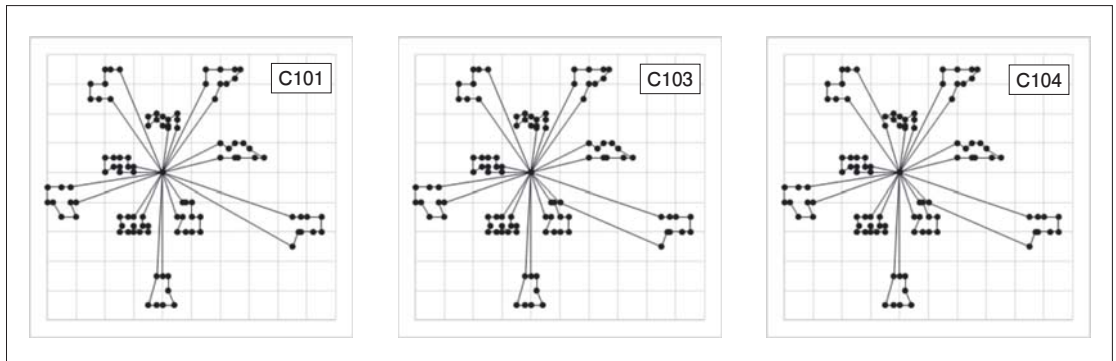


図7 各問題 (C101、C103、C104) の最適解

5. おわりに

本論文では、VRPTWに対して最近傍法 (Nearest Neighbor algorithm) により得た解を初期解として利用し、改良操作として三つの操作 (経路内操作、経路間操作、経路分割-結合操作) を適用して解を改善するアルゴリズムを提案した。ベンチマーク問題^[10]による数値実験の結果、その有効性を確認することができた。今回は顧客数が100で顧客配置がクラスタ化されたデータのみを扱ったが、今後はランダム配置のデータや顧客数のより多いデータに対する適用可能性についても調べていく予定である。

参考文献

- [1] T. L. Magnanti: “Combinatorial optimization and vehicle fleet planning: Perspectives and prospects”, Networks, Vol.11, No.2, pp.179–213, 1981.
- [2] M. Desrochers, J. Desrosiers, and M. Solomon: “A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows”, Operations Research, Vol.40, No.2, pp.342–354, 1992.
- [3] J. K. Lenstra, A.H.G. Rinnooy Kan, “Complexity of vehicle routing and scheduling problems”, Networks, Vol.11, pp.221–227, 1981.
- [4] B. L. Garcia, J. Y. Potvin, and J. M. Rousseau: “A Parallel Implementation of the Tabu Search Heuristic for Vehicle Routing Problems with Time Window Constraints”, Computers and Operations Research archive, Vol.21, No.9, pp.1025–1033, 1994.
- [5] J. Homberger and H. Gehring: “A two-phase hybrid metaheuristic for the vehicle routing problem with time windows”, European Journal of Operational Research, Vol.162, No.1, pp.220–238, 2005.
- [6] W. Chiang and R. Russell: “Simulated Annealing Metaheuristics for the Vehicle Routing Problem with Time Windows”, Annals of Operations Research, Vol.63, pp.3–27, 1996.
- [7] L. M. Gambardella, E. Taillard, and G. Agazzi: “MACS-VRPTW: A Multiple Colony System for Vehicle Routing Problems with Time Windows”, New Ideas in Optimization, D. Corne, M. Dorigo and F. Glover (eds.), pp. 63–76, McGraw-Hill, London, 1999.
- [8] J. Y. Potvin and S. Bengio: “The Vehicle Routing Problem with Time Windows, Part II: Genetic Search”, INFORMS Journal of Computing, Vol.8, No.2, pp.165–172, 1996.
- [9] M. M. Solomon: “Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints”, Operations Research, Vol.35, No.2, pp.254–265, 1987.
- [10] M. M. Solomon: Solomon’s Benchmark Problems, <http://web.cba.neu.edu/~msolomon/problems.htm>.
- [11] G. Clarke and J. W. Wright: “Scheduling of Vehicles from a Central Depot to a Number of Delivery Points”, Operations Research, Vol.12, No.4, pp.568–581, 1964.
- [12] İ. K. Altinel and T. Öncan: “A New Enhancement of the Clarke and Wright Savings Heuristic for the Capacitated Vehicle Routing Problem”, The Journal of the Operational Research Society, Vol. 56, No. 8, pp. 954–961, 2005.
- [13] M. A. Figliozzi: “Planning Approximations to the Average Length of Vehicle Routing Problems with Time Window Constraints”, Transportation Research part B - Methodological, Vol.43, No.4, pp.438–447, 2009.