

物流コスト削減のための配送集荷経路問題の最適化

加来佑一朗^{†1} 内村圭一^{†1} 上瀧剛^{†1} 大村悦彰^{†1} 福本ゆり^{†1}

車両の巡回経路や積載率によって生じる総走行距離の増加に伴う輸送コストの削減のために、本稿では配送集荷経路問題の最適化手法について報告する。配送集荷経路問題に対して列生成法とヒューリスティックスを併用することによって、ベンチマークテストとの比較の結果は総走行距離と計算時間ともに短縮された。

Optimize Vehicle Routing Problem Simultaneous Delivery and Pick-up to Reduce Logistics Costs

Yuuichirou Kaku^{†1} Keiichi Uchimura^{†1}
Gou Koutaki^{†1} Nobuaki Ohmura^{†1} Yuri Fukumoto^{†1}

In order to reduce transport costs due to the increase in the total traveling distance and loading rate of the vehicle, we report on the optimization method for Vehicle Routing Problem simultaneous Delivery and Pick-up. We obtained the result that the total traveling distance and computation time is reduced by combination of heuristics and column generation method for benchmark test.

1 はじめに

高齢化・過疎化に伴う通信販売利用による宅配便の増加がみられる^[1]。一方、経済のグローバル化に伴い国際物流が飛躍的に伸びている^[2]。これらに対して商品や製品の輸送・配達ルーティング・スケジューリング最適化は重要課題の一つに位置付けられている。そこで、最適な配車計画を立案する配車計画支援システムの早急な導入が望まれている。これには配送経路問題（VRP：Vehicle Routing Problem）という問題を最適化することが必要である。VRPとはデポ（特定の施設）に待機する車両によって、顧客の位置、需要量、顧客ごとのタイムウインドウ（作業開始時刻から終了時刻）、作業時間、移動時間、利用可能な配送車両数ならびに車両の最大積載量などが与えられたとき、これらの様々な制約を守るように総移動距離、必要な運搬車台数が最適である顧客訪問順を求める問題である。筆者らは、より現実に近い配車計画を解決するために、VRPに集荷の要素を含めた配送集荷経路問題（VRPSDP：Vehicle Routing Problem with Simultaneously Deliveries and Pickups）を最適化することを目指している。VRPSDPはNP困難問題であるVRPにさらに制約を付加したNP完全問題であり、最適化することが難しく、計算に時間のかかる問題である。

このVRPSDPに対してSalhiらはセービング法とヒューリスティックスを基にした挿入法を用いている^[3]。Dethloffは3つの異なった挿入技法を用いて物流管理の計画のためにこの問題を解決している^[4]。Nagyらは粒子群最適化を用いた最適化を提案している^[5]。

このようにVRPSDPを解決する手法は様々提案されている。本稿でも最適解を探索することのできる列生成法に

ヒューリスティックスである挿入法とtwo-opt法を併用することで、高速に最良解を得るようなVRPSDPに対するルートの最適化手法を提案する。この提案手法の有用性を示すため、本手法をベンチマークテスト^{[6][7]}に適用し、この結果をうけてVRPSDPに適用する。

2 最適化手法の提案

VRPSDPを解決するために列生成法にヒューリスティックスである挿入法とtwo-opt法を組み込む。まず挿入法で暫定解（タイムウインドウなどの制約を守っていて最適ではないがそれなりに良い解）を複数個作成し、その解をtwo-opt法により改善した後、列生成法により最適解を探索する。

2.1 挿入法

暫定解を複数生成するために本稿では挿入法を選択した。暫定解を生成する他の手法としてセービング法が代表的であるが、ソロモンのベンチマークテストによる顧客数が100である時間枠付きVRPに対する実験結果^[6]は、挿入法がセービング法に優越すると結論づけている^[8]。VRPSDPはVRPの発展系であり、本稿では顧客数が100以上の場合を対象にしたいので、挿入法を用いることにする。

2.1.1 挿入法のアルゴリズム

いま、あるルート上で点*i*の次に点*j*を訪問していたとする。この時、顧客*k*を点*i*, *j*間に挿入したとき、2点間の距離を*d*とし、評価を表す基準 Δ_{ij}^k を、

$$\Delta_{ij}^k = d_{ik} + d_{jk} - d_{ij} \quad (1)$$

と定義する（図1参照）。 Δ_{ij}^k を用いると、挿入法の一般形は以下のように書ける。

^{†1} 熊本大学
Kumamoto University.

挿入法

- 1) 運搬車の台数分の顧客（種顧客）を適当に選択する。
- 2) デポと種顧客を往復するようなルートを作り，それらを初期ルートとする。
- 3) 種顧客以外の顧客 k に対して，適当な順番に挿入時に実行可能かつ Δ_{ij}^k が最小になる点 i, j の間に，顧客 k を挿入する。

2.1.2 挿入時の制約

暫定解は列生成法で用いるため，実行可能解である必要がある．よって挿入法で顧客 k を挿入した時の評価値を算出する過程で，以下の (1)，(2)の条件のときは評価値に大きなペナルティを与えて挿入しない．

- (1) 車両の最大積載量を超える．
- (2) タイムウインドウを守っていない．

まず (1) について説明する．挿入法により，最大車両台数の数だけ種顧客を選択し，初期ルートが完成する．そのルートごとに配送用の最大積載量と集荷用の最大積載量が設定してあるので，図 2 に示すように顧客 i と顧客 j の間に顧客 k を挿入するとき最大積載量を超えるようであれば，その評価値にペナルティを与える．

次に，(2) について説明する．タイムウインドウとは作業開始時刻 e_i と終了時刻 l_i の間に配送集荷を行わなければならないという設定である．本稿でも顧客ごとに設定された配送と集荷は，その顧客のタイムウインドウを守るように設定する．ここで，顧客 i から j への移動時間を t_{ij} ，顧客 i での作業時間を s_i とすると，配送集荷開始時刻 b_j は次のようになる．なお， m はそのルートの顧客数である．

$$b_j = \max\{e_j, b_i + t_{ij} + s_i\} \quad (i, j \in m) \quad (2)$$

条件 (1)，(2) および式 (2) より，式(3)を満たすルートが実行可能解となる．

$$b_k < l_k \quad (0 < k < m) \quad (3)$$

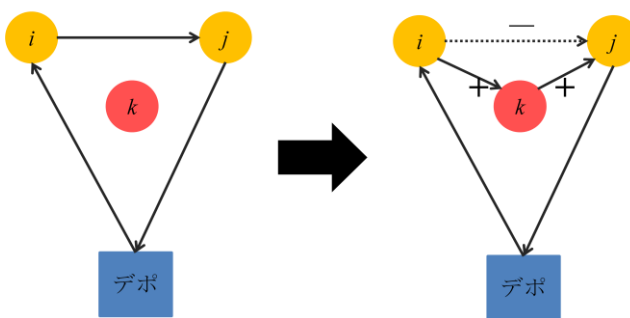


図1 挿入法における評価の計算

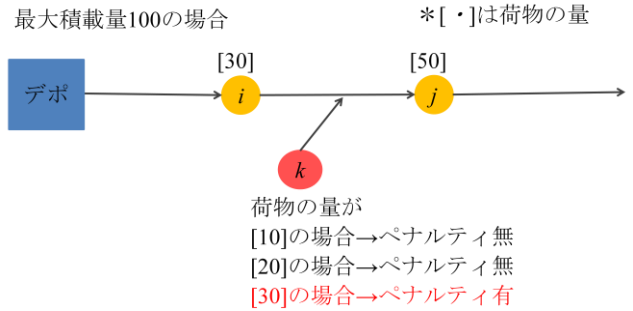


図2 積載量によるペナルティの有無

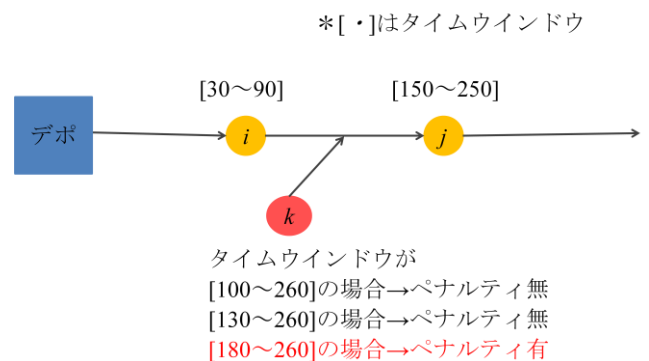


図3 タイムウインドウによるペナルティの有無

図3に作業時間 90，移動時間 10 の場合のタイムウインドウのペナルティ例を示す。

2.2 two-opt 法

挿入法により作成された暫定解に対して，ルートの総走行距離を削減することのできる two-opt 法を用いる．これはルート上に存在する枝の交差をなくす手法である．two-opt 法の一般形は以下のように書ける．

two-opt 法

- 1) 適当なルート上において，枝 (i, j) ， (k, l) で次の式となる枝の対がなければ終了．

$$d_{ik} + d_{jl} < d_{ij} + d_{kl}$$

- 2) そのような枝の対があれば枝 (i, j) ， (k, l) を枝 (i, k) ， (j, l) につなぎかえる．

図4に示すように 1) により交差箇所を探索し，2) により交差を削除することができる．この操作を任意の回数指定することで，そのルート上の交差を削除する．このとき，交差を取り除くことでタイムウインドウが守られなくなる場合は，交差は削除されない．

2.3 列生成法

列生成法の一般形は以下のようになる．

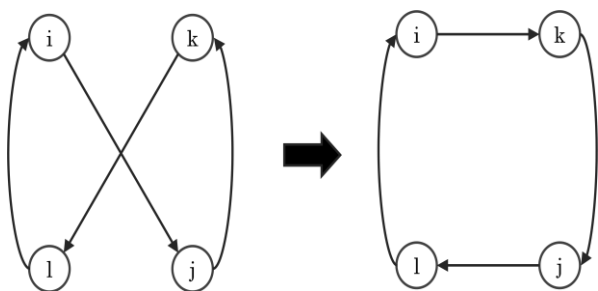


図4 two-opt 法

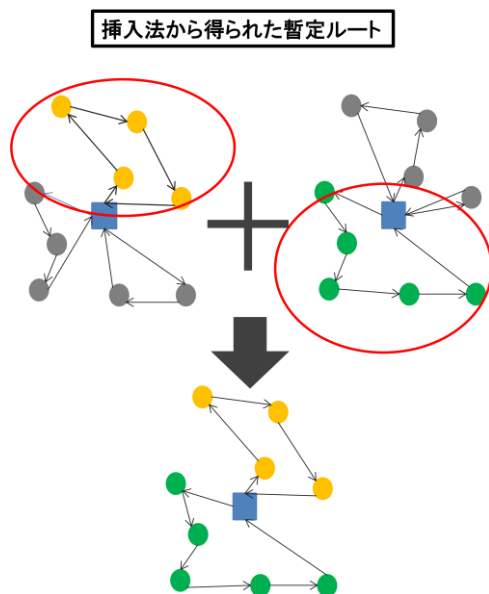


図5 列生成法

列生成法

- 1) 実行可能解を取得する。
- 2) プライシングを行い、双対変数の値を求める。
- 3) 双対変数の値を用いて、ラグランジュ緩和問題を解く。下界値を得る。
- 4) ラグランジュ緩和問題の解を実行可能解の生成系に追加し、上界値・実行可能解を得る。
- 5) 上下界値のギャップが 0 であれば終了する。そうでなければ 2) に戻る。

1)の実行可能解は挿入法及びtwo-opt 法により取得する。列生成法では図5のように暫定解の個々のルート（図5の朱丸参照）を組合せることで最適解を探索する。組合せの候補が増えるほど解は良くなるため、暫定解は最大車両数を変えながら複数作成する必要がある。

3 シミュレーション

VRPSDP を解決するにあたり、まず、提案手法（挿入法、two-opt 法と列生成法の組合せ）の有用性を確かめるために、VRP の実験対象として用いられるベンチマークテストで

シミュレーションを行う。次に、そのベンチマークテストの顧客情報を基に作成した配送集荷モデルを用いて VRPSDP のシミュレーションを行う。使用した PC の CPU は Xeon(R) E5-2620 2.00GHz, OS は Windows 7, プログラミング言語は C である。

3.1 VRP でのシミュレーション

VRP に対してはベンチマークテスト C101, C201, R101, R201, RC201^[6], C1_2_1, C2_2_1^[7]を用いて、顧客数 100 や 200 の場合でのシミュレーションを行う。C101, C201, C1_2_1, C2_2_1 の顧客パターンは、顧客の配置がタイムウインドウによりクラスターされている。R101, R201 のパターンは顧客の配置がランダムである。RC201 は顧客配置がランダムとクラスターの組合せである。シミュレーション結果がベンチマークテストの解に近いほど最適解（本稿では、最適解と称す）といえる。表1にベンチマークテストに示されている最大積載量と作業時間を掲載する。

ところで、列生成法に用いる暫定解の数（列数）の最大数 c_{max} は式 (4) に示すように挿入法のループ回数 i に依存する。

$$c_{max} = 24i \sum_{k=s}^l k \quad (4)$$

ここで、 k は使用車両台数、 l は最大使用車両台数、 s は最小使用車両台数である。また、最適化するための計算時間は列の組合せのパリエーションが大きいくほど増加する。本稿を通して $s=10$, $l=40$ とした。また、ベンチマークテストでは $i=20$ とした。

図6～12にルートのシミュレーション結果を示す。図6,7及び11のルートは交差がなく、表2より最適化できたといえる。図8,9,10,及び13には交差が見られるが、表2から、ベンチマークよりも良い結果が算出できていることが分かる。また、計算時間については、表3から従来手法[9]よりも高速であることが確認できた。また、提案手法の計算時間が顧客分布によって差がある。計算時間は列生成法での組み合わせ数に大きく依存するので、それが原因と考えられる、

表1 シミュレーション設定

顧客分布	顧客数	最大積載量	作業時間
C101	100	200	10
C201	100	700	90
R101	100	200	10
R201	100	1000	10
RC201	100	1000	10
C1_2_1	200	200	90
C2_2_1	200	700	90

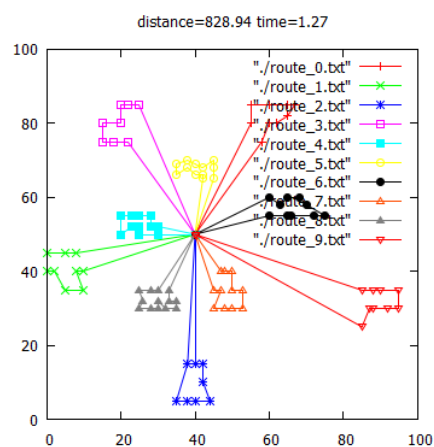


図 6 C101 のルート

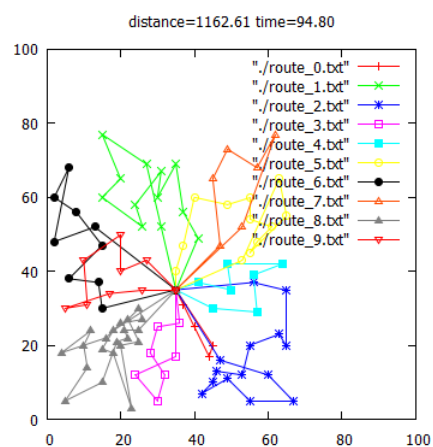


図 9 R201 のルート

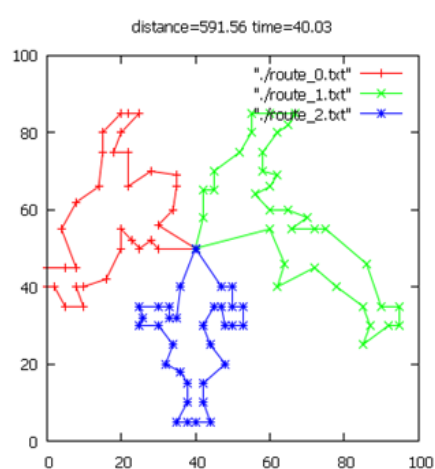


図 7 C201 のルート

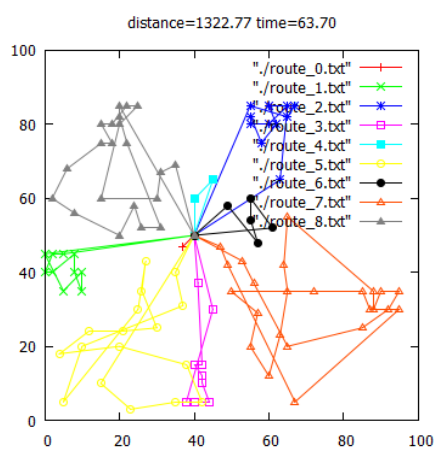


図 10 RC201 のルート

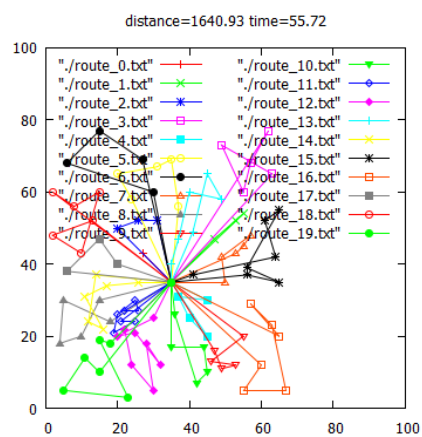


図 8 R101 のルート

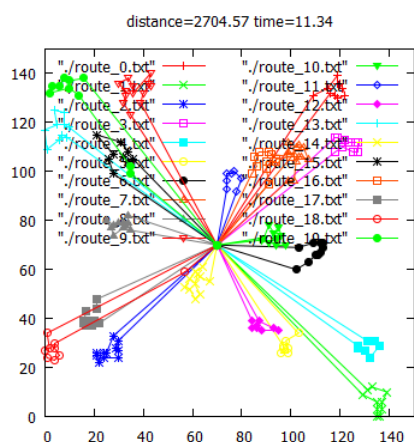


図 11 C1_2_1 のルート

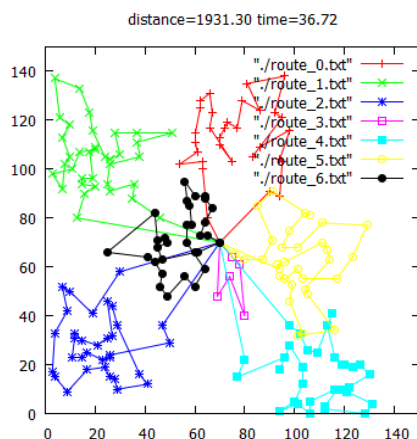


図 12 C2_2_1 のルート

表 2 総走行距離の比較

顧客分布	ベンチマークテスト ^{[6][7]}	提案手法
C101	828.94	828.94
C201	591.56	591.56
R101	1645.79	1640.93
R201	1252.37	1162.61
RC201	1406.91	1322.77
C1_2_1	2704.57	2704.57
C2_2_1	1931.44	1931.30

表 3 計算時間の比較(sec)

顧客分布	C101	C201	R101
提案手法	1.27	12.56	55.72
文献[9]*	41	145	138

*他の顧客分布での記載なし

3.2 VRPSDP でのシミュレーション

前節での VRP でのシミュレーション結果から、提案手法の VRPSDP への適用は有効であると考えられる。従って、ベンチマークテスト C101, C201, R101 を参考に 3 つの顧客パターン A~C (表 4 参照) において、VRPSDP に対するシミュレーションを行った。顧客のタイムウインドウ、配送量はベンチマークテストと同じ設定であるが、集荷量については、任意の値を設定した。表 5 にパターン A での設定例を示す。

パターン A, B, C に対するシミュレーション結果をそれぞれ図 13, 14, 15 に示す。また、図 16, 17, 18 にループ回数に対する総移動距離の推移を、図 19 にループ回数に対する計算時間の推移を示す。

表 4 顧客パターン

パターン番号	顧客分布
A	C101
B	C201
C	R101

表 5 パターン A の設定例

顧客	x 座標	y 座標	配送量	集荷量	開始時刻	終了時刻
1	45	68	10	30	912	967
2	45	70	30	10	825	870
3	42	66	10	10	65	146
			.			
			.			
			.			
99	55	80	10	20	743	820
100	55	85	20	10	647	726

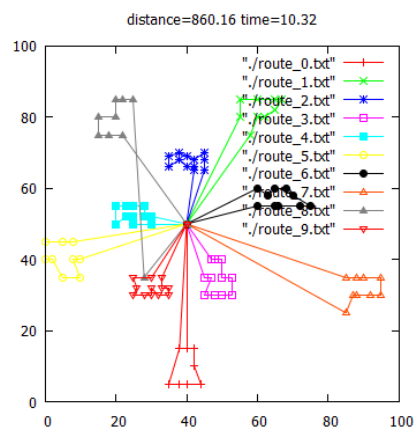


図 13 パターン A のルート

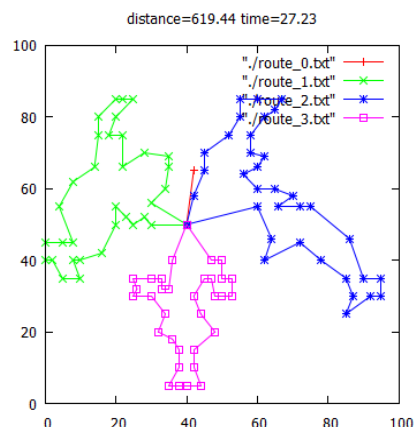


図 14 パターン B のルート

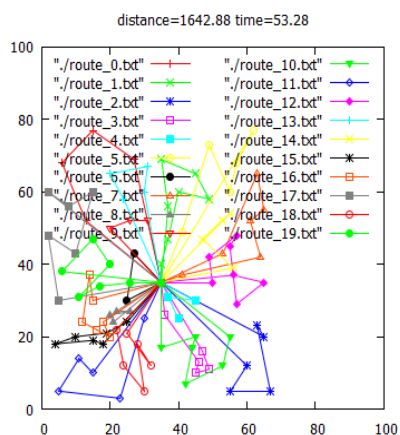


図 15 パターン C のルート

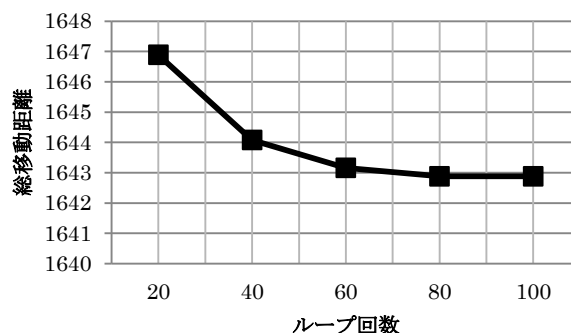


図 18 パターン C のループ回数による
 総移動距離の推移

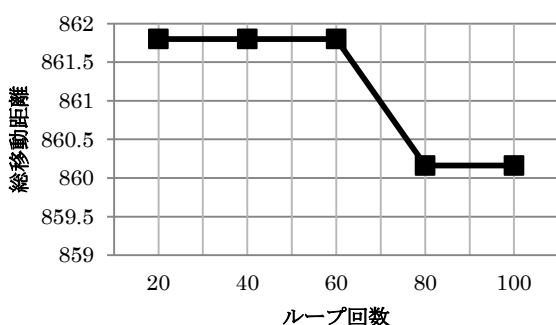


図 16 パターン A のループ回数による
 総移動距離の推移

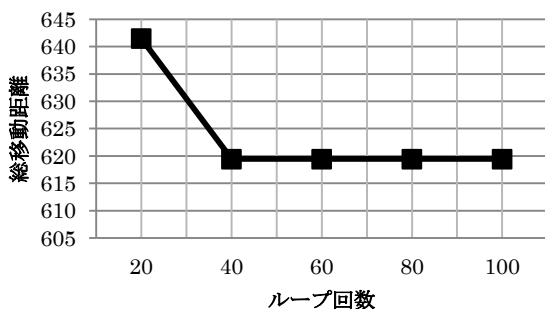


図 17 パターン B のループ回数による
 総移動距離の推移

4 考察

図 13 および図 15 においては、ルート上に交差が見られるなどのため、改善の余地が見込める。対応策として、暫定解の質を向上させることが考えられる。このためには、他の局所探索法を用いることが挙げられる。例えば、or-opt 法を用いてルート同士を組合せることで、暫定解の総走行距離の削減が見込める。

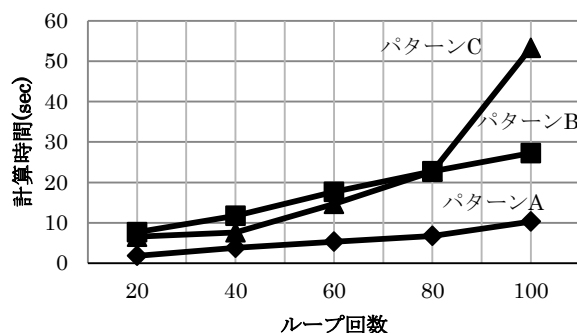


図 19 ループ回数による計算時間の推移(sec)

総移動距離に関しては、図 16, 17 および 18 より、想定通り列数の増加により解が良くなることが確認できた。

計算時間については、VRP のシミュレーションにおいて、従来の結果よりも高速であることが確認できたが、VRPSDP については、ベンチマークテストのような比較対象がないので、今後検討していく必要がある。また、図 19 より、パターン C の計算時間の増加が激しい。パターン C の顧客配置は、作成された列数が多かったため、列生成法での最適化する時間が増大したためである。対応策として、タブーリストを用いて、一度できた暫定解は破棄する手法などが挙げられる。

5 まとめ

本稿では、まず列生成法に挿入法と two-opt 法を用いることで VRP を解決し、その有用性を示したあと VRPSDP への適用を図った。結果として、VRP では最適解が導出され、VRPSDP では列数増加による解の改善が見られた。

今後は、顧客配置を現実的なものに設定して実験を行っていく。また、更なる解の改善および計算時間の改善を図っていく。

謝辞

本研究の一部は、A-STEP探索タイプ（課題番号：AS242Z02811H）の助成を受け実施したものである。記して敬意を表する

6 参考文献

- [1] <http://www.meti.go.jp/statistics/index.html>
- [2] 林克彦：国際物流の動向と商社の役割，日本貿易会月報，7,8月号，No.694, pp. 15-17, 2011
- [3] Salhi.S, and Nagy.G : A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling, Journal of Operational Research Society, Vol. 50, No. 10, pp. 1034-1042, 1999
- [4] Dethloff. J : Vehicle routing and reverse logistics : The vehicle routing problem with simultaneous delivery and pickup, OR Spektrum, Vol. 23, Issue 1, pp. 79-96, 2001
- [5] Nagy.G, and Salhi.S. : Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries, European Journal of Operational Research, Vol.162, Issue 1, pp. 126-141, 2005
- [6] <http://web.cba.neu.edu/~msolomon/problems.htm>
- [7] <http://www.sintef.no/Projectweb/TOP/VRPTW/Homberger-benchmark>
- [8] 久保幹雄：ロジスティック工学，朝倉書店，2003
- [9] S. Masrom, Siti. Z. Z. Abidin, A. M. Nasir, A. S. Abd. Rahman : Hybrid particle swarm optimization for vehicle routing problem with time windows, Recent Researches in Computational Techniques, Non-Linear Systems and Control , pp. 142-147 , 2011