

超大規模 Vehicle Routing Problem に対する単純化の概念を用いた 進化型多目的最適化アプローチの提案

佐藤哲也* 渡邊真也† 榊原一紀

(室蘭工業大学)‡ (室蘭工業大学)§ (富山県立大学)¶

1 はじめに

Vehicle Routing Problem (VRP) は、配送計画問題または運搬経路問題と呼ばれ、配送センター (Depot) から複数の需要地への配送を効率的に行おうとする配送ルート決定問題であり、実務的な問題として 1950 年代より数多くの研究がなされてきた [1, 2].

VRP は、制約として与えられた各ビークルの積載量制約を満たした上で複数のビークルを用いてすべてのカスタマーを訪れる経路決定問題であり、ビークルに対するカスタマーの割り当て、および各ビークルでのカスタマーの訪問順序を決定する問題である。これまで VRP に対しては、分枝限定法や数理計画法といった厳密解法から局所探索法 (Local Search, LS) や Simulated Annealing (SA), 遺伝的アルゴリズム (Genetic Algorithm, GA) といった様々な発見的解法が適用されており、数多くの適用例が報告されている [1]. また, VRP は総移動距離だけでなくビークルの台数, 各経路間の移動距離の分散度合いなど本質的に多目的の要素を内包していることから, 進化型多目的最適化 (Evolutionary Multi-criterion Optimization, EMO) を VRP へ応用した研究も数多く報告されている [2, 3, 4, 5].

しかし, 近年はコンピュータの高性能化及びクラウド技術の進歩にともなう情報のビッグデータ化に伴い幅広い分野で扱うデータや問題自体が大規模化しており, その対応が大きな課題となっている。VRP においても, 問題の大規模化により組み合わせ爆発, 計算コストの増加などの問題により, 従来手法現実的な時間内にユーザーが妥協できる品質の解を導出することが難しくなる。

そこで本研究では, 大規模問題に対しカスタマーのクラスタリングを用いた問題の単純化を試みた。本手法では, 探索が進むに従って段階的にその単純化を解消しながら解探索を行うことによる効率的な探索の実現を目指す。

提案手法の有効性を検証するため, 代表的なベンチマーク問題である Taillard らの問題に対して数値実験を行い, ある一定数以上のカスタマーを有するテスト問題を対象に, 問題の単純化を行わない場合との比較を行った。

2 Vehicle Routing Problem

VRP にはその制約の種類に応じて様々な派生形が存在するが, 本論文では最も単純な積載量制約付き VRP (Capacitated VRP: CVRP) を扱う。

本研究における VRP の定義を下記に示すとともに, その概念図を Fig. 1 に示す。

複数台のビークルを用いて N 人のカスタマーを巡る。各車はデポ (Depot) と呼ばれる出発地点から, 割り当てられたカスタマー集合を全てを巡り, デポに戻る。このとき車によるカスタマーの通過順をルート (巡回路) とよぶ。各車は i 番目のカスタマー地点 u_i において w_{u_i} の需要量 (重量) を積み込むものとする。カスタマー u_i とカスタマー u_j 間は, 距離コスト d_{u_i, u_j} で接続されているものとする。車がルートをめぐる間, 積載量が容量 W 以下となるといった制約が課せられる。

VRP の評価については総移動距離の最小化や各ルートごとの移動距離の分散, 各ルートごとの積載量の分散など様々なものが考えられるが, 本研究では総移動距離 F_{sum} および各ルートごとの移動距離の分散 F_{var} の 2 目的最適化問題として定式化した。本論文で扱う目的関数を以下に示す。

$$\min F_{\text{sum}} = \sum_{m=1}^M d^m \quad (1)$$

$$\min F_{\text{var}} = \frac{1}{M} \sum_{m=1}^M (\bar{d} - d^m)^2 \quad (2)$$

式 (1) における M は使用するルートの総数であり, d^m は m 番目のルートにおける距離コスト, \bar{d} は各ルートの平均距離を表す。 d^m の定義を式 (3) に示す。

$$d^m = d_{0, u_1^m}^m + \sum_{i=1}^{n_m-1} d_{u_i^m, u_{i+1}^m}^m + d_{u_{n_m}^m, 0}^m \quad (3)$$

式 (3) における $d_{u_i^m, u_j^m}^m$ は m 番目のルートにおけるカスタマー u_i から u_j までの距離コストを表している。また, u_i^m は m 番目のルートにおいて i 番目に巡るカスタマーを表しており 0 はデポを意味している。 n_m は m 番目のルートが巡回するカスタマーの総数を表しており, 本論文では巡回すべき全カスタマー数を $N = \sum_{m=1}^M n_m$ とする。

次に, 各ルートにおける積載量制約に関する定義を式 (4) に示す。本論文では, すべてのルートにおいて同一

†14043018@mmm.muroran-it.ac.jp

‡sin@csse.muroran-it.ac.jp

北海道室蘭市水元町 27-1 室蘭工業大学大学院 工学研究科情報電子工学系専攻

§北海道室蘭市水元町 27-1 室蘭工業大学大学院 しくみ情報系領域
¶富山県射水市黒河 5180 富山県立大学 工学部 情報システム工学科

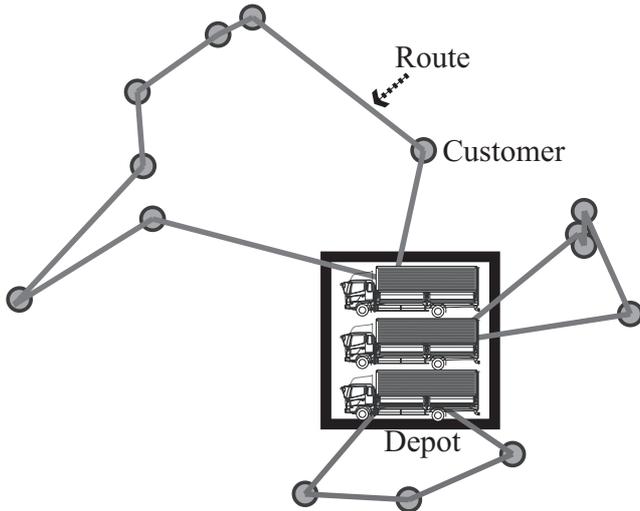


Fig. 1 The concept figure of VRP.

の積載量制約 W を使用した.

$$W \geq w^m = \sum_{i=1}^{n_m} w_{u_i^m}, \quad (m = 1, \dots, M) \quad (4)$$

式(4)における w^m は、 m 番目のルートにおける総積載量であり、 $w_{u_i^m}$ は m 番目のルートにおいて i 番目に巡る顧客の需要量を表している。

式(4)より VRP は容量に関する制約、およびすべての顧客を巡るといった制約を満たし、かつ評価が最小となるような配送計画、すなわち

- (1) ビークルへの顧客の割り当て
 - (2) 各ビークルが顧客を巡る順路
- を決定する問題であると捉えられる。

3 提案手法

提案手法は、将来的に超大規模 VRP と呼ばれる、顧客数が 1000 以上の問題に対して適用することを想定している。そのため、問題の大規模化にともなう組み合わせ爆発、計算コストの増大といった問題への対応が重要となる。

本研究では、割り当て問題を EMO を利用して探索を行い、順路問題を Lin- Kernighan Heuristic (LKH) [6] を利用して解くことにより、効率的な解探索の実現を目指した。LKH は k -opt 法におけるパラメータ k を可変にすることで、効率的に良質な巡回路を生成するアルゴリズムであり、総移動距離の改善量に応じて動的に k を増加させながらより良い改善を目指すという特徴を持つ。

以下に提案手法の全体の流れを示す。また、全体をフローチャートで表したものを Fig. 2 に示す。

Step 0: 初期パラメータの設定

- ・ 対象問題を設定
- ・ N 初期個体数
- ・ A アーカイブサイズ
- ・ G_S 解の停滞を許容する世代数

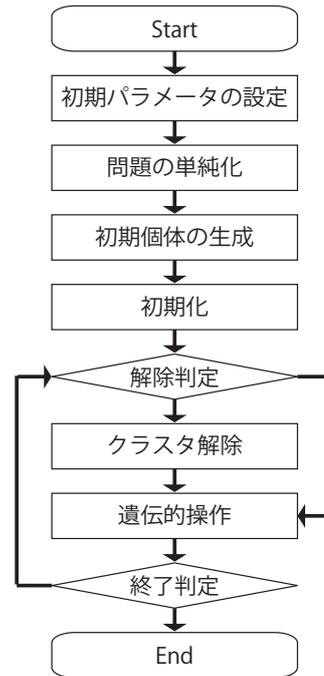


Fig. 2 Flowchart of Proposal Approach.

- ・ G_F オリジナル問題に回帰する世代
- ・ G_{Final} 終了世代

Step 1: 問題の単純化

対象問題における各顧客に対して、クラスタリングを利用したグルーピングを行い、グループ化した複数顧客を1つの顧客と見立てることで問題を単純化する。

Step 2: 初期個体の生成

N 個の初期個体を生成する。

Step 3: パラメータの初期化

$g = 0, F_{best} = 0, g_s = 0$ とする。ここで g は現在の世代数、 F_{best} は前世代の F_{sum} の最小値、 g_s は F_{best} の停滞世代数を表す。

Step 4: 評価値の比較

アーカイブ内の個体から F_{sum} が最小となる評価値と F_{best} を比較する。両者が異なるとき、 $F_{best} = \min F_{sum}$ 、 $g_s = 0$ と代入する。両者が等しいとき、 $g_s = g_s + 1$ とする。

Step 5: クラスタ解除

$G_S \leq g_s$ のとき、一部のクラスタを解除する。解除後 $g_s = 0$ とおく。

Step 6: 遺伝的操作

遺伝的操作により新たに N 個の子個体を生成し、アーカイブを更新する。

Step 7: 終了条件判定

$G_{Final} \leq g$ ならば終了し、そうでなければ $g = g + 1$ とし Step4 へ戻る。

本章では、上記の提案手法における Step1 のクラスタ形成アルゴリズム及び、Step5 クラスタ解除アルゴリズムを説明する。

3.1 クラスタ形成アルゴリズム

本提案手法は、ある程度距離に近い複数のカスタマーを1つのカスタマーと見立てることで、扱うカスタマー数を抑えるものである。一般的に、クラスタリングを行う際は距離の近いものから順に結合していくが、VRPではカスタマーが持つ需要量も考慮する必要がある。

実際に本研究で利用している代表的なベンチマーク問題である Taillard らの問題¹ では、真の解においてデポに近いカスタマーはカスタマー間距離が非常に近いカスタマー同士でも異なるビークルに割り当てられており、一方、デポから遠いカスタマーではカスタマー間距離が多少離れている場合でも同じビークルに割り当てられている。

そのため、クラスタリングを導入するにあたり、クラスタ内カスタマー数の上限と距離の閾値を設け、この点について考慮したグルーピングの実現を試みた。すなわち、デポから遠いカスタマーほど集団を形成しやすく、逆にデポから近いカスタマーほどより集団を形成しにくくなるようにクラスタリングを行った。

また、クラスタリングを行う際、より隣接距離の近いカスタマー同士から結合を行うが、新たに追加するカスタマーの需要量とクラスタに含まれているカスタマーの需要量の総和の和がビークルの最大重量値を超えてはならないこと、そしてある一定以上の距離のクラスタ同士は結合してはならないことなどを条件とした。

以下、アルゴリズムの詳細な手順を示す。

Step 1: 初期クラスタの生成

カスタマー数と同数のクラスタ $C_i (i = 1, \dots, N)$ を生成し、各クラスタ C_i に各カスタマーを1つずつ割り当て、各クラスタに含まれる総カスタマー積載量 $w^{C_i} (i = 1, \dots, N)$ を設定する。ここで、クラスタ内カスタマー数 P_i は、 $P_i = 1$ とし、現在の総クラスタ数 N^C を $N^C = N$ とする。

Step 2: クラスタ内カスタマー数上限の設定

デポと各クラスタ間距離の正規化を行い、正規化した距離に基づいて、各クラスタ内に取り入れることができる最大カスタマー数 P_i^{\max} を決定する。

Step 3: 代表クラスタの選択

未選択クラスタの中で式 (5) を満たすクラスタ C_i を選択する。このとき、該当するクラスタが存在しなければ終了する。

$$\max_{i=1, \dots, N^C} d_{(C_i, C_0)} \quad (5)$$

ここで C_0 はデポのことであり、 $d_{(C_i, C_j)}$ は最短距離法に基づくクラスタ間の距離を表す。

Step 4: 終了条件判定

$N^C \leq N^P$ のとき終了する。そうでなければ、Step5へ進む。ここで N^P は最小クラスタ数を表す。

Step 5: 結合クラスタの選択

式 (6) を満たすクラスタ C_j を選択する。このとき、該当するクラスタの組み合わせが存在しなければ、Step3に戻る。

$$\begin{aligned} D^T &\geq \min_{j=1, \dots, N^C | j \neq i} d_{(C_i, C_j)} \quad (6) \\ \text{s.t. } W &\geq w^{C_i} + w^{C_j} (j = 1, \dots, N^C | i \neq j) \\ \text{s.t. } P_i^{\max} &\geq P_i + P_j (j = 1, \dots, N^C | i \neq j) \end{aligned}$$

ここで、 D^T はクラスタの結合が可能な距離の上限(閾値)を表し、 W は積載量の上限、 $d_{(C_i, C_j)}$ はクラスタの代表点の距離を表す。

Step 6: クラスタの結合

クラスタ C^i と C^j を結合し、クラスタ内総重量とクラスタ内のカスタマー数を更新する。具体的には、 C^j に含まれるカスタマー情報を C^i に統合し、 C^j を削除するという処理を行う。そして $N^C = N^C - 1$ とし、Step4へ戻る。

上記のように、結合アルゴリズムでは、近接するクラスタ同士を目標となるクラスタ数 (N^P) まで結合するという操作の繰り返すことを基本としている。

3.2 クラスタ解除アルゴリズム

クラスタ解除アルゴリズムは、探索の停滞が認められた時に一部のクラスタを解除し、元の問題に近づけるといふ処理を行うためのものである。すべてのクラスタを一度に解除、つまり元の問題へ戻した場合、解除前後での問題の規模および性質が大きく異なるため、それまでの探索において得られた情報が必ずしも効果的に受け継がれない可能性がある。そのため、本論文では、段階的に一部のクラスタを解除することで、元の問題と単純化した問題のギャップへ対応することを試みた。

その際に重要になるのは、クラスタをどのタイミングで解除するのかという点であるが、本論文では下記に示す条件をクラスタ解除のタイミングとして用いた。

条件 1. 前回の解除から $G_r (0 \leq G_r \leq G_F)$ 世代以上経過している。

条件 2. F_{sum} の最良値が G_S 世代に渡り、変化がない。

上記の条件から分かる様に、本アルゴリズムではその時点の最良個体 F_{sum} が更新されなければ、現時点の問題設定での探索は概ね収束したと判断し、クラスタ解除を行うようなアプローチをとった。ここで、クラスタの解除数は $r_b \leq r \leq 2r_b$ の乱数より決定し、パラメータ r_b が大きいほど解除するクラスタの数が多くなり、逆に小さいほど少なくなるよう設定した。解除するクラスタの選択基準については、様々なものが考えられるが、本研究では下記に示す5つの基準およびこれらを組み合わせたものについて実装を試みた。

¹ VRP website
<http://www.bernabe.dorronsoro.es/vrp/>

Parameter	tai150a	tai385
N	50	
A	250	
G_r	3	
G_S	3	
G_F	160	240
G_{Final}	200	300

1. クラスタ内最大2点間距離が大きいもの
2. クラスタ内分散値が大きいもの
3. クラスタ重量が大きいもの
4. クラスタ内ヘビークラスタが重いもの
5. 他クラスタとの距離の分散

ただし、最終的には元のカスタマー情報を利用した解を求める必要があるため、探索が G^F ($0 \leq G^F \leq G$) 世代になったとき、強制的にクラスタを全て解除し、元の問題へ戻すという操作を行っている。

以下にアルゴリズムの詳細について述べる。

Step 1: オリジナル問題への帰化判定

現在の探索世代 g が $g \geq G_F$ を満たすとき、すべてのクラスタを全解除し、オリジナル問題に帰化して終了する。

Step 2: 部分解除のタイミング判定

前述のクラスタ解除を行うタイミングの判定を行い、満たしていれば Step3 へ、そうでなければ終了する。

Step 3: クラスタの解除数の決定

解除するクラスタ数 r を r_b および乱数を基に決定する。

Step 4: クラスタの解除

前述の解除するクラスタの基準に基づき評価値を算出し、その値の高いクラスタを r 個だけ選択し、クラスタを解除。

4 数値実験

本論文では、提案手法の有効性を検証するため、下記の2種類の数値実験を行った。

実験1 クラスタリング (問題の単純化) アプローチの有効性の検証

実験2 解除するクラスタの選択基準についての比較検証

テスト問題には代表的なベンチマーク問題である Tailard らの問題から tai150a と tai385 を用いた数値実験を行った。また、実験に使用したパラメータを Table. 1 に示す。今回の実験ではテスト問題に対してそれぞれの方法に各5試行ずつ適用し、最終結果の F_{sum} の最良値が中央値を取る結果の推移を出力した。

以下、実験結果及び考察について述べる。

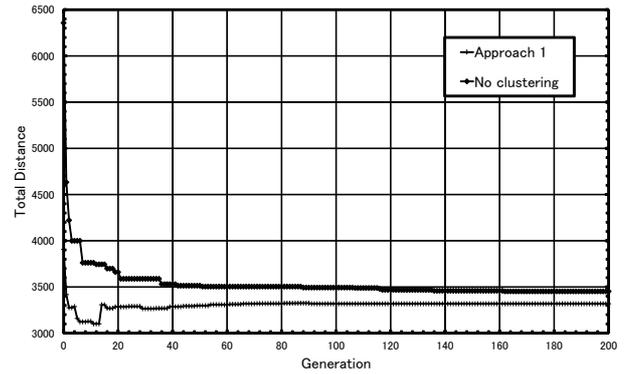


Fig. 3 Transitions of F_{sum} at tai150a

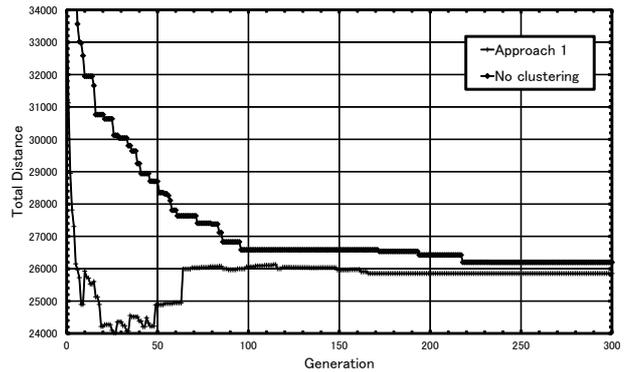


Fig. 4 Transitions of F_{sum} at tai385

4.1 実験1と考察

実験1では tai150a と tai385 の問題規模の異なる2つの問題に対し、クラスタリングを使用しない場合と、クラスタリングを使用した場合の数値実験を行った。ここでは3.2節で述べた5つの基準のうち1番目の基準である「クラスタ内最大2点間距離」に基づく結果をクラスタリングを使用した場合として用いた。tai150a と tai385 の数値実験の結果をそれぞれ Fig. 3 と Fig. 4 に示す。

問題の規模が比較的少ない tai150a の数値結果である Fig. 3 をみる。この問題では100世代前後ですべてのクラスタが解除されている。その地点での評価値を比較してもクラスタリングをしている方が評価値がよいことがわかる。

次に比較的問題の規模が大きい tai385 の数値結果である Fig. 4 をみる。この問題では200世代前ですべてのクラスタが解除されている。その地点での評価値を比較してもクラスタリングをしている方が評価値がよいことがわかる。

クラスタリングをした場合、問題が単純化されているため、少ない計算量で解探索が行われ、初期において収束されやすくなる。クラスタを解除後も、解除する前の情報を利用できるため、効率よく解探索が行われていると考えられる。

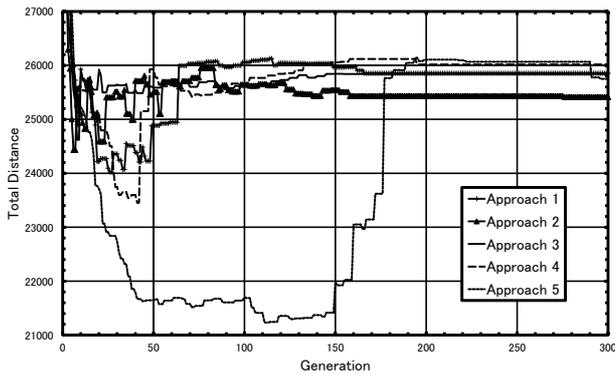


Fig. 5 Transitions of F_{sum}

4.2 実験 2 と考察

ここでは、解除するクラスタの選択基準に関して 3.2 節で述べた 5 つの基準の比較実験を行った。具体的には、この 5 つの基準単独の場合とそれらを組み合わせた場合について tai385 を対象に検証を行った。まず、5 つの基準ごとの実行結果を Fig. 5 に示す。

結果を比較すると、最終的な結果はいずれの手法も似たような結果を示しているものの、クラスタリングを行っている最中の探索推移を見ると、各手法で異なる推移であることがわかる。基準 1 と 4 は似たような探索推移を示しており、探索序盤は収束が強いが中盤にかけて改悪が目立つ。その後のクラスタ解除までの探索、解除後の探索は安定した推移を示している。基準 2 と 3 も似たような探索推移を示していることがわかった。これらは探索初期から終盤にかけて大幅な収束や大幅な改悪がなく、安定した推移を表している。最後に基準 5 は他の 4 手法と大きく異なり、序盤に大きく収束するが、後半のクラスタ解除で大幅な改悪をしていることがわかった。

次に複数の選択基準を組み合わせた考察を行うが、今回行う組み合わせとして、Fig. 5 で最良な結果を示した基準 2 と 5、探索全体で最良な結果を示した基準 4 と 5 で実験を行った。この結果を Fig. 6 と Fig. 7 に示す。

結果を見ると、基準単独と複数の選択基準を組み合わせた場合で明確な結果の違いが現れなかったが、単独の際と同様に、探索中の探索推移が特徴的な推移をしていることがわかった。Fig. 6 では基準 5 と同様初期に収束し、終盤にかけて改悪していることがわかる。しかし、基準 5 のように一度に大幅な改悪ではなく、段階的に改悪していることがわかる。これは基準 2 の推移を見れば安定した推移であることがわかる。この安定した推移という特長により、急激ではなく、段階的な推移になったのではないかと考えられる。また Fig. 7 では初期収束し、すぐに大幅改悪する基準 4 と後半に大幅改悪する基準 5 の中間をとるように、中盤に改悪していることもわかる。このように、複数の選択基準を組み合わせる場合、その基準の特徴を把握することが重要であることがわかった。

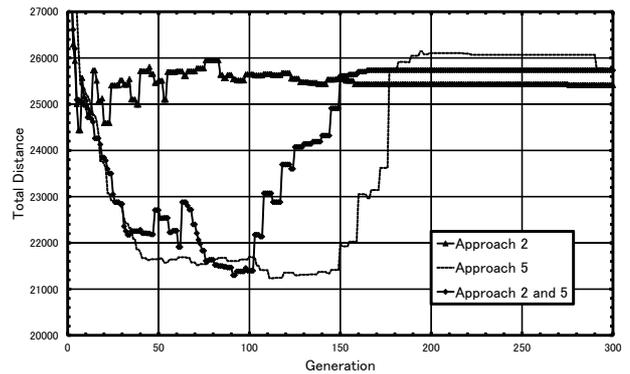


Fig. 6 The result of combination of Approach 2 and 5

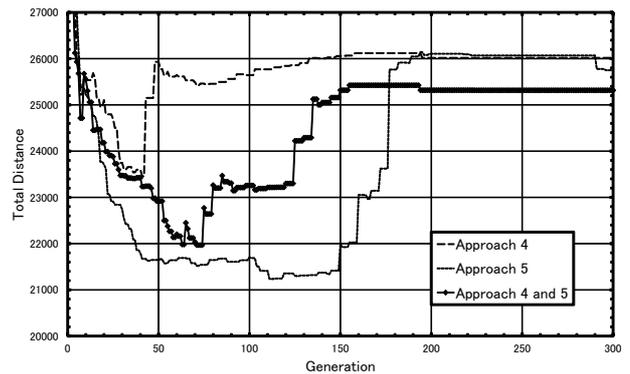


Fig. 7 The result of combination of Approach 4 and 5

5 おわりに

本論文では、大規模 VRP に対してクラスタリングを利用した問題の単純化と段階的詳細化手法を提案し、数値実験を通してその有効性について検証を行った。代表的なベンチマーク問題である Taillard らの問題の tai150a と tai385 への適用より、クラスタリングを使用した場合の探索性能が使用しない場合よりも優れていることが確認できた。

今後の展望としては、結果が真の解と離れているため、これに対応するため遺伝的操作の更なる改良と、大規模問題と呼べるようなカスタマー数が 1000 を超える問題への適用を目指していく。

参考文献

- [1] O. Braysy and M. Gendreau. Vehicle routing problem with time windows, part i: Route construction and local search algorithms . *Transportation Science*, Vol. 39, No. 1, pp. 104–118, 2005.
- [2] 渡邊真也, 榎原一紀. 多目的化の概念を用いた単目的 Vehicle Routing 問題へのアプローチ . 情報処理学会論文誌「数理モデル化と応用」Vol.48, No.SIG 2(TOM 16),158 - 166, 2007.
- [3] N.Jozefowicz, F.Semet, and E.Talbi. Parallel and Hybrid Models for Multi-objective Optimization: Application to the Vehicle Routing Problem

- . In *Parallel Problem Solving from Nature—PPSN VII*, pp. 271–280, 2002.
- [4] T.Murata and R.Itai. Multi-objective vehicle routing problems using two-fold emo algorithms to enhance solution similarity on non-dominated solutions
. In *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pp. 885–896, 2005.
- [5] J.Y. Potvin and S. Bengio. The vehicle routing problem with time windows - part ii: Genetic search
. In *INFORMS Journal on Computing*, Vol. 8, pp. 165–172, 1996.
- [6] S.Lin and B.Kernighan. *An effective heuristic algorithm for the traveling-salesman problem.*
. *Operations Research*, Vol. 21, pp. 498-516, 1973.