

配送計画問題に対するデータベース付きメタ戦略

名古屋大学大学院・工学研究科 棚橋 優 (Yu Tanahashi)
今堀 慎治 (Shinji Imahori)
Graduate School of Engineering,
Nagoya University

概要

本稿では、配送計画問題に対するメタ戦略アルゴリズムの計算時間短縮について考える。配送計画問題とは、ある1つの拠点から複数の顧客へトラックを用いて物を配送する際、そのコストが最小となる配送経路を求める問題である。この問題は古くから研究されている問題であり、応用例は日常生活において頻繁かつ身近に存在する。本稿では、探索を繰り返すごとに得られる局所最適解の情報をデータベースに記録し、それを以降の探索に活用するメタ戦略アルゴリズムを提案する。提案法では、同じ計算を何度も行う必要がなくなり、計算時間を削減することができる。また、提案法の特徴として、一度データベースを作成すれば、同じ顧客集合の問題に対して、情報の使い回しができることが挙げられる。この点も踏まえたうえで、数値実験による評価を通して提案法の有効性を検証する。

1 はじめに

日常生活において、運搬という作業は絶え間なく行われている。例えば、宅配便や出前等のデリバリーサービス、毎日決められた時間に行われる郵便配達、幾度となく発注を繰り返す小売店。本稿では、このような場面に現れる配送計画問題について扱う。配送計画問題の目的は、運搬車による費用最小の配送経路集合を求めることである。我が国の国内貨物輸送は、その大部分をトラック輸送に頼っており、経済社会にとって不可欠の構成要素である。運搬の効率化は、物流に直接携わる関係者にとってはもちろんのこと、企業、一般国民、ひいては我が国全体にとって重要な課題であり、配送計画問題における費用削減の期待はきわめて大きい。

この問題を解くにあたって考えられる最も単純な方法は、全ての可能性について考える方法、すなわち全探索であろう。しかし、全探索は計算量が顧客数に対して指数関数的に増加するため、極めて非効率な解法となってしまふ。従って工夫が必要である。

配送計画問題に対する解法の多くは構築法とメタ戦略からなる。構築法とは何も無いところから解を組み立てる方法のことである。しかしここで得られる解は、多くの場合、精度の良い解とは言えない。よって得られた解を初期解とし、解の近傍を探索して良い解が見つければそちらへ移るといった手段を取る。この方法を局所探索法といい、局所探索法で得られる良質な解を局所最適解という。局所最適解は初期解に依存し、初期解によって良くも悪くも異なる解となる。従って、一度の局所探索法の適用のみで満足するのではなく、初期解を変えて何度も局所探索を行うことで、より精度の高い解を得ることが出来る。本稿では、局所探索法を複数回繰り返す反復局所探索法を用いる。このように、ある初期解から良い解を探す手法を総称してメタ戦略とよぶ。メタ戦略は、高精度な解を近似的に求めることができるが、良い解を見つけるためにはそれなりの計算時間がかかってしまうという課題がある。

本稿では、現実問題をモデルとした、同じ顧客配置でその需要のみ異なる問題が何度も繰り返されるような状況を想定する。毎日決まった時間に異なる量を発注する複数の小売店と、それらへの配送を担う運搬車を思い浮かべると良い。このとき、一つ一つの問題はそれぞれ独立した異なる問題であるが、顧客同士の相対距離が変わらないことから、最適解に含まれるそれぞれの巡回路（1台の運搬車による配送経路）は大変よく似ていると予想される。この状況において、それぞれの問題に対してメタ戦略アルゴリズムを実行すると、同じ計算が何度も繰り返し行われることになる。それらは無駄な時間であり、このような計算を削減することによって計算の効率化が期待できる。

現実社会では、同じ地域の配送を何度も経験しているドライバーがいる。彼らはこれまでの経験から、どういう経路を進めばより早くより楽に仕事をこなせるかを知っている。本稿では、熟練のドライバーと同じことを計算機でもできないだろうか考えた。過去に行った計算結果を次の計算に役立てることで、効率良く探索を行うのである。解の精度を保ったまま計算時間を短くすることを目的とし、データベースを用いたメタ戦略アルゴリズムを提案する。

2 配送計画問題

2.1 配送計画問題とは

配送計画問題とは、ある配送拠点から n 人の顧客へ、複数の運搬車を用いて物を配送するとき、さまざまな制約条件を満たしながら、そのコストが最小となる配送経路集合を求める問題 [4][6] である（図 1 参照）。この問題は代表的な組合せ最適化問題であり、実用性が高い問題である。

実社会での応用を考えたとき、配送計画問題は大規模な問題となることが多い。この場合、手計算はもちろんのこと、計算機を用いたとしても厳密解を求めることは難しい。配送計画問題は NP 困難（NP-hard）であり、多項式時間で厳密解を見つけることは極めて困難であると考えられている。そこで、この問題に対する妥協策として、現実的な時間で出来る限り厳密解に近い解を求める近似解法を設計する。

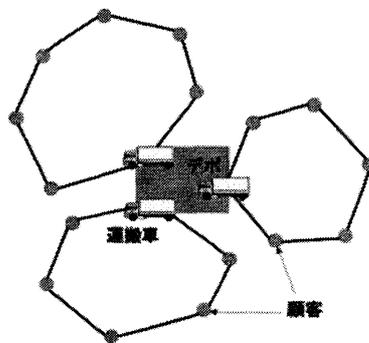


図 1: 配送計画問題の例

2.2 配送計画問題の定式化

本稿で取り扱う配送計画問題について述べる。

本稿におけるコストとは、全運搬車の総移動距離と定義する。また運搬車には容量制約を設け、必要ならば運搬車は何台でも使用してよいことにする。本稿で扱う配送計画問題は次の制約条件

を持つ。

- デポと呼ばれる点を出発した運搬車が、顧客を経由し、再びデポに戻る。
- 各顧客の需要は、ちょうど1台の運搬車によって運ばれる。
- 1台の運搬車が運ぶ顧客の需要量の合計は、運搬車の最大積載量を超えない。

本稿で用いる変数および用語を以下のように定義する。

変数および用語	定義
二次元座標 (x_0, y_0)	デポの位置
二次元座標 (x_i, y_i)	顧客 i の位置 ($i = 1, 2, \dots, n$)
相対距離 d_{ij}	点 i, j 間の距離 ($d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$)
需要量 q_i	顧客 i が要望する荷量 ($i = 1, 2, \dots, n$)
容量 r	運搬車1台が運ぶことができる最大積載量
台数 v	運搬車の台数
移動距離 D_i	運搬車 i が移動する距離 ($i = 1, 2, \dots, v$)
枝	顧客から顧客およびデポへの経路
巡回路	運搬車1台が巡る経路

これらの変数を用いると、配送計画問題の入力、評価値、出力は以下ようになる。

- 入力 : 顧客数 n , 運搬車の容量 r , デポの座標 (x_0, y_0) , 各顧客の座標 (x_i, y_i) , 各顧客の需要量 q_i ($i = 1, 2, \dots, n$)
- 評価値 : 総移動距離 $\sum_{i=1}^v D_i$
- 出力 : 評価値が最小となる配送経路

2.3 先行研究

この節では配送計画問題の先行研究について述べる。

配送計画問題は、制約条件の違いにより、本論文で扱う問題以外にもさまざまなバリエーションが存在する。例えば顧客への到着時刻が決められている時間枠付き配送計画問題 [4] や、運搬車の種類がいくつか存在し区別している問題 [6] などがある。

そしてこれらの配送計画問題も、その多くがNP困難であり、多項式時間の厳密解法は絶望視されている [7]。そこで、長時間かけて厳密解を求めるのではなく、近似解を求める方法、すなわち近似解法の研究が古くからなされてきた。

近似解法として、古典的なものに1964年に発表されたセービング法 [3] が挙げられる。また、現在における多くの企業およびソフトウェアに採用実績がある近似解法は、顧客を方面別のグループ（クラスター）に分割した後に、個々のクラスターに対してルートを設定する解法である。この解法はクラスター先・ルート後法（cluster-first/route-second method）と呼ばれる。クラスターへの分割に関しては、1970年代には領域分割法（region partitioning method）が中心に用いられ

ており、1980年代以降には一般化割当法 (generalized assignment method), 漸近的最適性をもつ施設配置ヒューリスティック (location based heuristic) [2] の有効性が認識されている [5].

また1990年代には、得られる解の精度だけでなく、現実問題に対する頑健性の面でも高い性能を持つメタ戦略 (metaheuristics) が注目される。これは計算機の計算速度の向上によって、ある程度の計算量の負荷をかけても実務上許容されるようになってきたことに起因する。解法の実装に関しては、地理情報システム (GIS: Geographic Information System) を中心とした情報技術との連携による構築・導入事例が評価されている [1].

3 既存アルゴリズム

この章では、本稿で用いた既存アルゴリズムについて述べる。

3.1 配送計画問題へのアプローチ

本稿では2つのステップからなる近似解法を用いる。第1ステップで解の構築を行い、第2ステップでより良い (評価値の小さい) 解の探索を行う。第1ステップにおける解の構築は、ランダム生成もしくは既存の構築法を用いる。構築法については次の3.2節で述べる。ここでの目的は、確実に実行可能解をつくり上げることである。このとき、評価値の意味では決して良い解とはいええず不十分である。そこで第2ステップで、第1ステップで得られた解を初期解とし局所探索法を用いることで、より良い解への改善を行う。3.3節では、本稿で改善法として使用した局所探索法について述べる。さらに本稿では、局所探索法を拡張し、2つのステップを何度も繰り返す反復局所探索法を採用した。この手法については3.4節で述べる。

3.2 構築法

すでに実行可能解が1つ存在するとき、新しく解を生成するには、存在している解を少し変えてやれば良い (3.4節参照)。しかし、解をゼロから構築するためには、構築法と呼ばれる方法を用いる。構築法 (construction method) とは、何も無いところから徐々に巡回路を拡大していき、解を構成していく解法の総称である。ここでは、配送計画問題に対する古典的な構築法であるセービング法と、ビンパッキング問題の解法であるファーストフィット法を元とする方法を紹介する。

3.2.1 セービング法

セービング法 (saving method, 節約法とも呼ぶ) [3] は1964年に Clarke と Wright によって提案された手法であり、その単純さとある程度の実用性のため、配送計画問題に対するアルゴリズムの代名詞といえる近似解法である。

セービング法の考え方は、初めにデポから全ての顧客へ往復する n 個の巡回路を生成し、2つの巡回路を結合することを繰り返して、総移動距離を減少させていくことである。その際、2点間の距離 d_{ij} を用いて次のセービング値 s_{ij} という値を計算し、移動距離の節約効果が高い組合せから順に結合させていく (図2参照)。

$$s_{ij} = d_{i0} + d_{0j} - d_{ij} \quad (1)$$

計算するセービング値の数は、デポと顧客の数を考慮すると $(n+1)^2$ 個である。また、結合の操作は、1つの大きな巡回路が形成された場合を考えると、最大で $(n-1)$ 回行われる。従って、セービング法の計算時間は $O(n^2)$ に収まる。

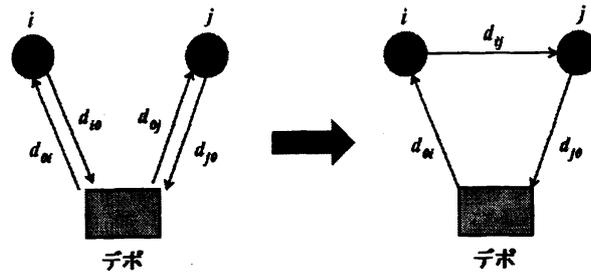


図 2: セービング法における巡回路の結合

3.2.2 ファーストフィット法

ファーストフィット法 (first fit method) は、ビンパッキング問題に対する代表的な近似解法である。ビンパッキング問題 (bin-packing problem) とは、重さが異なる n 個の荷物を、 W の重さを入れることのできる T 個の箱に、すべて入れられるかどうかを決定する問題である。本稿の言葉で表すと、荷物は顧客が求める積み荷であり、荷物の重さは顧客の需要量、 W の重さを入れることのできる T 個の箱は、 W の容量を持つ T 台の運搬車である。

ファーストフィット法の考え方は次の通りである。

初めに顧客をランダムな順に並べる。そして並べた列の先頭の顧客から順番に、用意していた運搬車に割り当てる。ある顧客を割り当てる際、もし運搬車の容量を超えてしまうのならば、次の運搬車へ配属させる。これ以降、顧客を最初の運搬車に積むことが出来るのならば初めの運搬車へ、出来ないのなら次の運搬車へ、さらに出来なければ次の次へ割り当てる。以上の操作を繰り返し、全ての顧客を運搬車へ割り当てる事が出来たら完了である (図 3 参照)。

この手法を用いて全ての顧客を運搬車に割り当て、各運搬車での配送順序は割り当てられた順番とする。こうすると、配送計画問題としての評価値 (移動距離) を無視し、運搬車の容量のみを考慮した初期解を生成することになる。

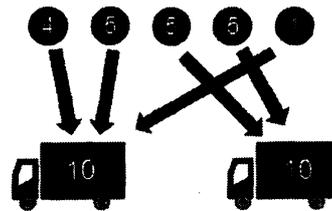


図 3: ファーストフィット法 (●内の数字は需要量, 運搬車内の数字は容量)

3.3 改善法

この節では、本稿で改善法として用いた局所探索法について述べる。まず局所探索法の一般的な説明を行い、次に本稿で実際に使用した近傍について説明する。

3.3.1 局所探索法

局所探索法とは、ある解に少しの変更を加えて、元の解よりも良くなったとき、変更後の解に移動するという操作を、それ以上良い解が見つからなくなるまで繰り返す方法である。

局所探索法では、現在の解 x の近傍 $N(x)$ の中から次の解を選択する。このとき評価値を改善する解が存在する場合には、その中から1つの解 x' を選択する。近傍の中に評価値を改善する解が存在しない場合には探索を終了する。この時に得られた解を局所最適解という。

以下、配送計画問題のように評価値をできるだけ小さくする最小化問題の場合を考える。

3.3.2 配送計画問題における近傍

ここでは配送計画問題に焦点を絞り、本稿で実際に用いた近傍について述べる。本稿では、以下で説明する 2-opt 近傍および 3-opt 近傍を用いる。これらの近傍は巡回セールスマン問題に対する古典的な近傍である。

2-opt 近傍

2-opt 近傍は以下のように定義することができる。

- 1つ以上の巡回路による解が構築されている。
- 解における2本の枝 $(i, i+1)$ $(j, j+1)$ を $(i, j+1)$ $(j, i+1)$ につなぎ替える。2本の枝を交換する組み合わせは、1つの巡回路のみを対象とする場合と2つの巡回路を対象とする場合がある。これら全ての組み合わせを合わせたものが 2-opt 近傍である。

本稿では、1つの巡回路を対象とした場合と、2つの巡回路を対象とした場合の、2種類に分けて考える (図4, 図5参照)。

3-opt 近傍

3-opt 近傍は以下のように定義することができる。

- 1つ以上の巡回路による解が構築されている。
- 巡回路の3本の枝 $(i, i+1)$ $(j, j+1)$ $(k, k+1)$ を $(i, j+1)$ $(j, k+1)$ $(k, i+1)$ につなぎ替える。3本の枝を交換する組み合わせは1つの巡回路, 2つの巡回路, 3つの巡回路を対象とした場合があり、これら全ての組み合わせを合わせたものが 3-opt 近傍である。

本稿では、1つの巡回路を対象とした場合と、2つの巡回路を対象とした場合の、2種類を考える (図6, 図7参照)。3つの巡回路を対象とする場合も考えられるが、計算量が大幅に増加することと、その割に得られる恩恵が少ないことから、この場合は除外した。

枝の選択

上記のとおり、解の近傍を探索する際には2本または3本の枝の選択を行う。このとき、まず枝交換の対象とする巡回路を1つもしくは2つ決定し、次に巡回路に沿ってデポから生じる枝から順に枝を選択する。しかし、2つの巡回路を対象とした枝交換を考える場合、運搬車の容量を超えてはならないという制約から、どれだけ枝の組合せを考えても実行可能解が成立しなくなることがある。このような実行不可能となる領域は出来る限り探索せず、再び実行可能となる枝の組合せまで近傍探索を省略することで、無駄な計算時間を大いに削ることができる。

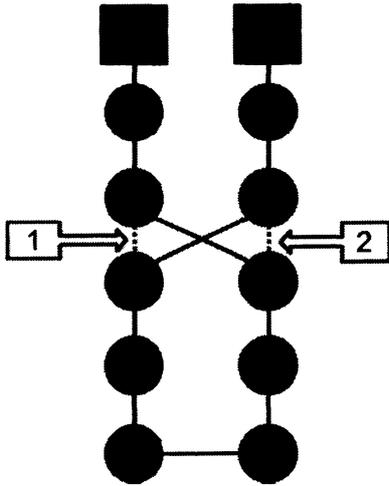


図 4: 1つの巡回路に対する 2-opt 近傍

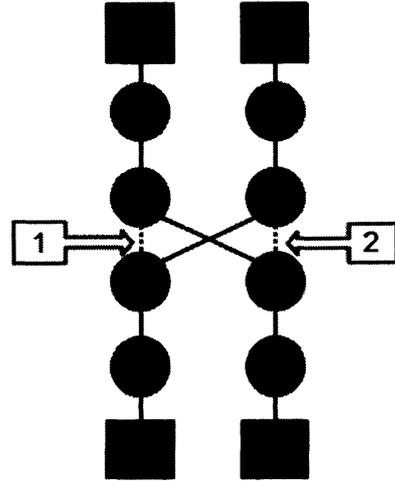


図 5: 2つの巡回路に対する 2-opt 近傍

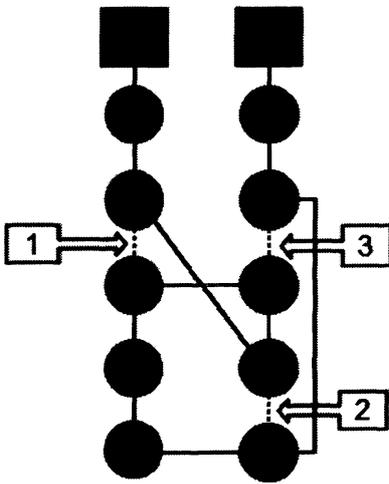


図 6: 1つの巡回路に対する 3-opt 近傍

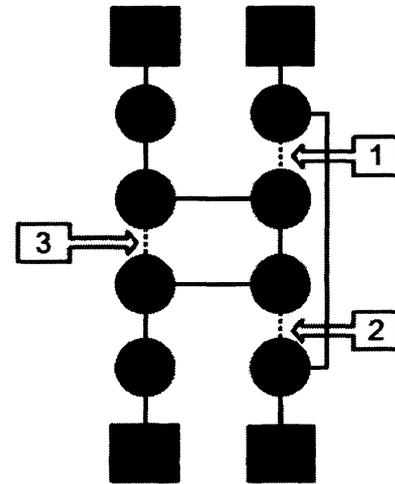


図 7: 2つの巡回路に対する 3-opt 近傍

3.4 反復探索

局所探索法で最終的に得られた解（局所最適解）はあくまで局所的なものであり、必ずしもその問題における厳密な最適解（大域的最適解）とは限らない。局所探索法は理論的には遅い（指数オーダーの反復を要する）場合があるが、実際の速度はほとんどの問題に対して非常に速い。従って計算時間に余裕があるときは、さらに時間をかけて、より良い解を探索しようという要求が自然に出てくる。

3.4.1 反復局所探索法

より良い解を探索するために、局所探索法を初期解を変えて繰り返し行う方法がある。この方法は多スタート局所探索法と呼ばれる。代表的な手法に、それまでの探索で得られた良い解を用いて初期解を生成する、反復局所探索法（iterated local search）がある。近接最適性（proximate

optimality property, 良い解に近いところには良い解が存在するという性質)によれば, この方法によりさらに良好な解を探索することができる.

反復局所探索法を実行するには, 局所最適解に陥ったときにそこから脱出するための(通常の近傍解とは異なる)「拡張された近傍解」が必要になる.

3.4.2 拡張された近傍解

1回の局所探索が終了したとき, 次回の局所探索のための初期解を再び生成する必要がある. そこで, それまでの探索で得られた最も良い局所最適解である暫定解に対し, ランダムな枝の組み替え操作を行う. 具体的な手順は次のとおりである.

1. 試行回数を, 5回から10回の範囲でランダムに決定する.
2. 2つの巡回路に対する近傍を, 前述の2-opt近傍または3-opt近傍からランダムに決定する.
3. 入れ替えの対象となる枝をランダムに決定する.
4. 入れ替え後の巡回路の総需要量がトラックの容量を超えていたら, 枝を入れ替えずに3.に戻る. 容量制約を満たしていれば, 枝を入れ替えてから2.へ戻る. 2.へ戻る操作を1.で決定した回数だけ行う.

脱出先である解は, 暫定解から比較的近い解といえる. これを新たな初期解とすることで, 近接最適性を生かしながら反復探索をすることができる.

4 提案アルゴリズム

ここでは, 本論文における提案法であるデータベース付きメタ戦略について詳述する. 初めに提案法の目的について述べる. 次に4.1節でデータに関する説明を行い, その有用性を明らかにする. その後4.2節でデータベースの利用方法, 4.3節でデータベースの構造および計算量について説明する.

本稿で扱う配送計画問題の背景として, デポの座標 (x_0, y_0) および顧客の座標 (x_i, y_i) は変わらず, 需要量 q_i が異なる問題が何度も繰り返されるような状況がある. このとき, それぞれの問題に対する局所最適解は類似の構造をもち, その結果として, 同じ計算が何度も繰り返し行われると予想される.

同じことが1つの問題に対する反復局所探索法の中でも起きている. その解法の性質上, 同じ顧客集合の問題に対し局所探索を何度も繰り返している. 以前の局所最適解に含まれた巡回路と全く同じ巡回路が次の探索でも現れる, という事は決して珍しくない. つまり, 一つ一つの局所探索の計算を比較してみると, 同じ計算を行っている箇所が数多く存在することになる. これは明らかに無駄である. そこで, 局所最適解の情報をデータベースへ記録し, 反復局所探索法を効率よく行う工夫を提案する. 過去の情報を活用し余計な計算を省くことで, 計算時間の短縮を目指す.

4.1 データとは

ここでは, 提案法の鍵となるデータについて述べる.

4.1.1 局所最適解の性質

メタ戦略では局所探索を何度も繰り返す、その度に局所最適解が得られる。例えば反復回数を100回として計算を行うと、最大100通りの相異なる局所最適解が得られる。同問題に対するこれらの局所最適解を比較すると、いくつかの特徴が明らかになる。図8を用いてそれを説明する。

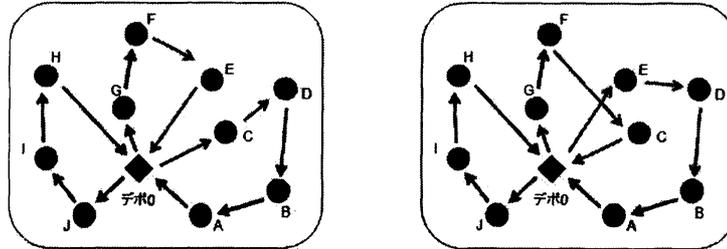


図8: 異なる局所最適解同士の比較例

1つ目の特徴は、局所最適解同士の解の構造が似ている、すなわち異なる局所最適解に同じ巡回路が現れることである。図8における、デポ、J、I、H、デポの順で回る巡回路がそれにあたる。

図8では、各局所最適解に3つの巡回路が存在する。言い換えると、3つの小さい顧客集合が存在する。本稿において、これらを部分顧客集合と呼ぶ。すると多くの場合、異なる局所最適解には全く同じ部分顧客集合が存在すると言えらる。反復局所探索で良い解を探そうとすると、複数の局所探索において同じ部分顧客集合が生じることが多い。位置的に近い顧客達が同じトラックで配送されることは、現実でも大いにありえることである。

2つ目の特徴は、それぞれの巡回路は配送順序という意味で良い順序になっていることである。各巡回路は、それぞれの部分顧客集合に対しほぼ最小の移動距離となっている。これは局所最適解に至る過程で、3.3節で述べた1つの巡回路に対する近傍探索を行っているためである。巡回路とは1台のトラックが回る経路であるが、部分顧客集合に対する順序付け問題、すなわち巡回セールスマン問題であるとも言える。

巡回セールスマン問題 (traveling salesman problem)

あるセールスマンが訪問予定されているすべての都市を1回ずつ訪れて出発地へ戻る際、総移動距離を最小にする訪問順序を決定する問題 [10].

局所最適解に含まれる巡回路は、この巡回セールスマン問題を解いた結果となっている。ただし、これは2-opt近傍と3-opt近傍を用いた局所探索法で得られた結果であるため、厳密解である保証はどこにもない。この点に対する対抗策は4.2.3節で述べることにし、ここでは厳密解が得られるという仮定のもと話を進める。

4.1.2 巡回路の記録

局所最適解に含まれる巡回路とは、部分顧客集合に対する巡回セールスマン問題を解いた結果である。そしてこれと等しい部分顧客集合は、それ以降の探索中にも数多く現れる。つまり、毎回同じ集合に対する巡回セールスマン問題を解くことになる。ここに計算を省略する余地が生まれる。局所最適解に含まれる巡回路をデータとしてデータベースに記録しておく。そして以降の

計算において記録したデータと照合し、同じ部分顧客集合が生じた場合には、改めて計算をしなくても移動距離最小の巡回路を得ることが出来る。この方法ならば、解の精度を保ったまま計算時間を短くすることができると考えられる。

4.2 データベースの利用

ここでは、データの保存や参照といった、データベースの具体的な使い方について説明する。またデータの精度に関することも本節で述べる。

4.2.1 データの保存

前述の通りデータベースには、局所最適解に含まれる巡回路をデータとして記録する。ある回の局所探索が完了し、局所最適解が得られたとき、直ちにそこに含まれる巡回路を全てデータベースに書き込む。具体的に記録する情報は、巡回路に含まれる顧客数、巡回路に含まれている顧客とその回る順番、移動距離、そして巡回路を検索する際に鍵となる特殊な値（ハッシュ値）である。ハッシュ値については4.3節で述べる。暫定解だけではなく得られた局所最適解を全て対象とすることで、より多くの種類のデータを記録する。

4.2.2 データの参照

データベースの参照は近傍探索を始める前に行う。局所探索の初期解は一般に良い評価値を示していない。ここでデータベースを参照し、解に含まれる部分顧客集合を検索して、この集合における理想の巡回路を探す。該当データが存在しないとき、従来通り近傍探索を行う。該当データが存在するときは、すぐに解の部分顧客集合をデータベースに保存されている巡回路に置き換える。こうすることで、近傍探索を行う前に、この巡回路は最適な形となり、これ以降、この巡回路1つに対する近傍探索は行わずに済む。また局所探索中、近傍操作によって部分顧客集合に変更が生じた場合、その都度変更の生じた部分顧客集合に対して再びデータベースの参照を行う。このようにして、出来る限り1つの巡回路に対する近傍探索を省略する。これらの手順をまとめると図9のようになる。

4.2.3 データの見直し

ここまでの議論は、扱うデータが正しいことを前提として行ってきた。しかし、データすなわち局所最適解に含まれる巡回路は、それぞれの部分顧客集合に対して厳密に移動距離最小であるとは言えない。なぜならば、局所最適解はそれ自体が局所探索法という近似解法を用いて得られたものだからである。

データベースを信頼して使うためには、データを可能な限り厳密解としておくことが望ましい。提案法を用いて配送計画問題を解き終えたとき、一つのデータベースが出来上がる。解く過程で蓄えた、巡回路のデータベースである。何度も述べている通り、これらはそれぞれの部分顧客集合における巡回セールスマン問題を、配送計画問題中の2-opt近傍および3-opt近傍を用いた局所探索法で解いた結果である。ここで、より良い（移動距離が少ない）結果を得るために、もう一度これらの顧客集合に対する巡回路を求めることにする。対象となる顧客集合を入力とした巡回セールスマン問題に対して、2-opt近傍と3-opt近傍を用いた反復局所探索を行う。この場合も厳

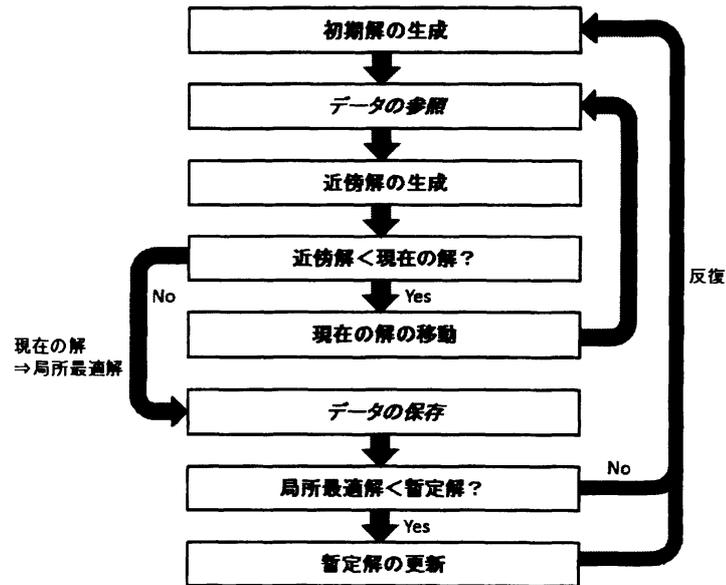


図 9: データベースを導入した反復局所探索法

密解を求めることは保証出来ないが、ある1つの巡回路に着目した反復探索は、配送計画問題を解く過程では行っていないので、より良い巡回路の獲得が期待できる。

このように、配送計画問題を解いた後、データの見直し作業を課す。そして、データベースのデータをより正しい情報にして、再び本来の配送計画問題を解く。ただし見直しに要する時間は評価の対象外とする。

あらかじめ用意されたデータベースを読み込んで使うことが出来ることも、提案法の利点のひとつである。こうして、配送計画問題の計算と、見直しの計算を繰り返し、データベースを何度も使いまわすことで、より情報量が多く、かつ正確なデータベースが出来上がると考えられる。まさに、何度も経験を積む熟練のドライバーのように、データベースも賢くなる。

4.3 データベースの構造

この節では、データベースの構造として用いたハッシュテーブル (hash table) について述べる。またそれに伴い、データの記録や参照に要する計算時間について述べる。

4.3.1 ハッシュテーブル

ハッシュテーブルとは、ハッシュ法にもとづくデータ構造のことである [8][9]。データベースの中に特定のデータが存在するか否かを探索する手法は幾つかあるが、ハッシュ法では検索コストが集合中のデータ数に依存せず、常に一定のコストで検索が可能である。

基本的な考え方は、ある任意の関数 (ハッシュ関数) を用いた、記録する値 (データ) のデータベースへの写像である。ハッシュ関数により検索名 (キー) からハッシュ値と呼ばれる値を計算し、データの格納個所を決定する。

4.3.2 ハッシュ関数

本稿で扱うキーとは、部分顧客集合であり、データとは記録すべき巡回路の情報である。出来る限り単純かつハッシュ値の衝突を防ぐようなハッシュ関数を考える。

まず全ての顧客に対して適当な ID を与える。このとき ID は互いに異なるランダムな大きい値とする。次に、欲しい部分顧客集合における顧客の ID の和を算出する。データベースのサイズを N としたとき、ハッシュ値がとりうる範囲を 0 から $N - 1$ とし、それよりも大きな値となった場合には、 N で割った余りをハッシュ値とする。

各顧客の ID はランダムな大きい数値であるため、得られるハッシュ値は 0 から $N - 1$ まで偏りなく分布することになる。またハッシュ値の計算量は O (部分顧客集合における顧客数) であり、実質的にデータの記録および参照にかかる計算量となる。

5 数値実験

提案法を用いた場合と用いない場合を比較し、データベースの利用効果を評価する。

5.1 実験概要

使用した顧客の配置例として、ベンチマーク問題 RC1_4_1 を用いた [11]。図 10 に記す。

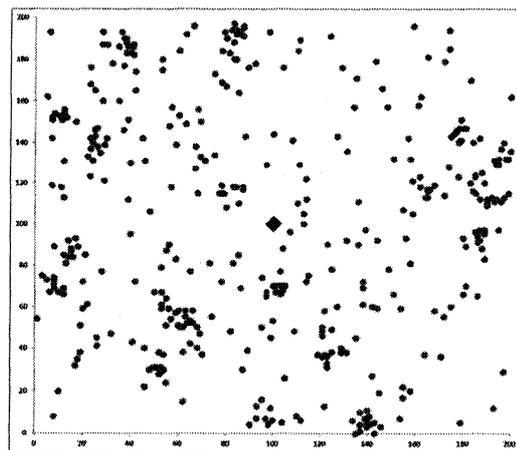


図 10: RC1_4_1 配置例

ただし、中央のひし形がデポ、丸が顧客を表す。客数は 400 人、需要量はそれぞれ 1 から 40 までの一様乱数とし、各運搬車の容量は 800 である。

この問題に対し、データベースを使わずに解く場合と、データベースを使って解く場合の実験を、セービング法およびファーストフィット法を構築法として用いたときの 2 通りに対して行った。ただし、提案法で用いるデータベースとして次のデータベースをあらかじめ用意した。

- 提案法を用いて問題 RC1_4_1 を解き、その後データの見直しを行ったデータベース。

データを蓄えることを目的に、反復回数 1 万回として RC1_4_1 を解く作業を、異なる乱数を用いて 10 回行う。(10 万個の局所最適解を得る。) その後一つ一つのデータに対し、反復回数を 1 万

回とした反復局所探索法を行った。こうすることで、始めから信頼できるデータが十分に存在している状況をつくり、提案法の効果がより顕著に現れるようにすることが狙いである。

実験で行う反復局所探索法の停止条件は、反復回数1万回とする。また、反復局所探索法には乱数を用いているため、1つの状況につき100回、乱数の条件を変えながら同じ実験を行った。

最後に、数値実験に用いた計算機環境を表1にまとめる。

表 1: 計算機環境

CPU	Intel Xeon X5690 Westmere-EP, 3.46GHz
メモリ	4GB
OS	Cent OS 5.4

5.2 セービング法を構築法とした場合

総移動距離（評価値）および計算時間について、100回行った平均と最大値、最小値を表2、3に記載する。また、セービング法の構築時の解と、提案法で得られた出力解の一例を図11、図12に載せる。

表 2: 総移動距離について

	平均	最大値	最小値
従来法	3238.82	3316.78	3210.43
提案法	3233.61	3314.37	3210.43

表 3: 計算時間（秒）について

	平均	最大値	最小値
従来法	112.21	294.44	62.63
提案法	51.59	186.84	12.24

まず表2に注目する。総移動距離の平均値は、提案法の方が若干良い結果となった。しかし、この平均値の差と、それぞれの最大値と最小値の差を比較すると、平均値の差はごく僅かなものである。よって、総移動距離に関しては両者に際立った違いはないと言える。一方で表3に注目すると、計算時間の平均値は従来法に比べ、提案法は半分以下となった。従ってこの結果を見ると、提案法の目的である、解の精度をそのままに計算時間を短くすることが達成された。

しかし気になる点が一か所ある。従来法と提案法のどちらにおいても、計算時間の最大値と最小値に非常に大きな差があることである。この理由について検証を行う。

図11によると、セービング法によって構築された初期解は10の巡回路からなる。しかし、改善が施され最終的に得られた出力解は、多くの場合が図12のように9つの巡回路で成り立っていた。これは、反復局所探索法の途中で巡回路の数が減っているからである。図13のように、2-opt

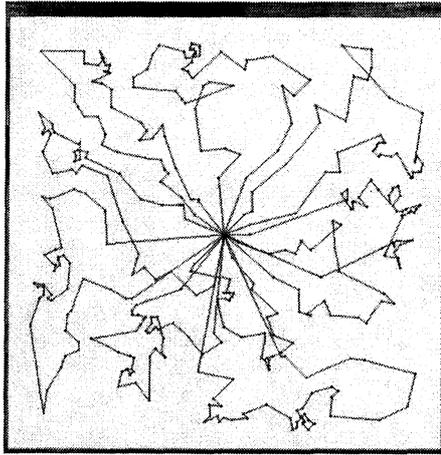


図 11: セービング法による構築

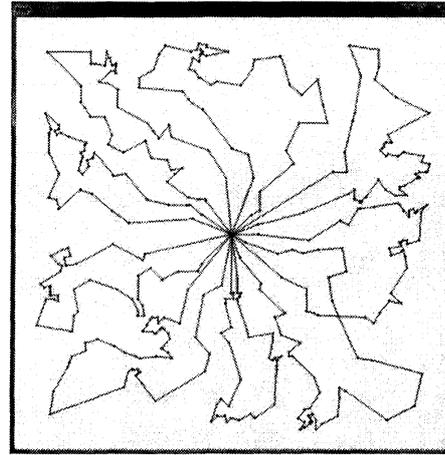


図 12: セービング法利用時の出力解の一例

近傍および 3-opt 近傍を定める枝交換操作によって、デポからデポという経路が生じるとき、巡回路は 1 つ消滅する。

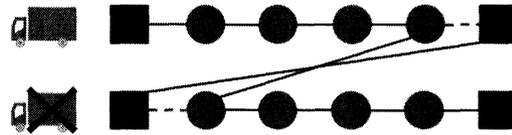


図 13: 巡回路の減少

反復局所探索中の近傍探索にかかる計算時間は、巡回路の数、巡回路に含まれる顧客数、そして巡回路に含まれる顧客の総需要量に依存する。巡回路の数が少ないと、逆に巡回路に含まれる顧客の数が多くなるため、運搬車の台数と計算時間間に大きな相関はないと考えられる。しかし容量制約を考えると状況は変わる。巡回路の数が少なければ、運搬車 1 台の容量が運搬する総需要量に対して余裕がなくなるため、容量制約によって実行不可能となる領域が広がり、3.3.2 節の枝選択の項で述べた工夫を加えることにより計算時間は短くなる。つまり巡回路が 10 あるときと 9 あるときでは、かかる計算時間が大きく違うのである。

実験結果にバラつきが生じたのは、反復局所探索のどのタイミングで巡回路数の減少が起こるかが乱数任せであるためである。反復の初期で巡回路が減ると、全体の計算時間は短くなり、反復の終わりまで巡回路が減らないままならば、全体の計算時間は長くなる。

実験結果を評価する際、計算時間にバラつきが生じるのは望ましくない。よって次節ではファーストフィット法を用いて、巡回路数が 9 の初期解を構築したうえで反復局所探索法を実行する。

5.3 ファーストフィット法を構築法とした場合

セービング法と同様に、総移動距離および計算時間について、100 回行った平均と最大値、最小値を表 4、5 に記載する。また、ファーストフィット法の構築時の解と、提案法で得られた出力解の一例を図 14、図 15 に載せる。

表 4 より、総移動距離はセービング法のとくと比べると全体的に大きくなってしまった。しか

表 4: 総移動距離について

	平均	最大値	最小値
従来法	3283.27	3363.87	3213.10
提案法	3285.40	3348.46	3213.10

表 5: 計算時間 (秒) について

	平均	最大値	最小値
従来法	67.66	80.84	60.08
提案法	15.53	19.72	12.21

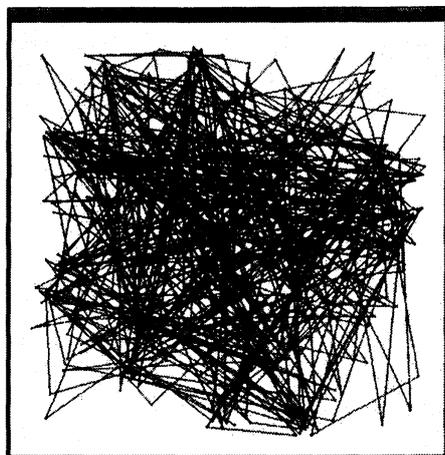
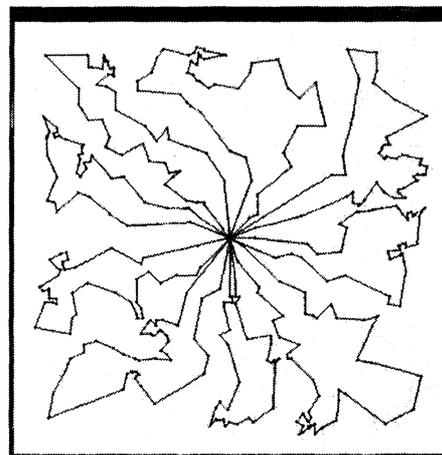


図 14: ファーストフィット法による構築

図 15: ファーストフィット法利用時の
出力解の一例

し従来法と提案法の差については、セービング法のとおり同様、両者に目立った違いはない。

次に表 5 に注目する。計算時間の平均は、提案法は従来法よりも 4 分の 1 程度まで短縮することに成功した。また最大値と最小値を見ると、セービング法のとおり差はなく、どちらにおいても提案法が従来法の 4 分の 1 程度となった。

3.2.2 節で述べたとおり、ファーストフィット法は配送計画問題の評価値を決める移動距離を一切考えていない。よって構築時の初期解は、図 14 のようにとても乱れている。しかし、この時の巡回路の数は 9 つであることを確認した。よって図 15 のような出力解が得られるまで、巡回路の減少は見られず、常に 9 つのままであったため、計算時間のバラつきがなくなり、提案法は効果的に働いているという評価をわかりやすく行うことが出来た。

6 まとめと今後の課題

本稿では、現代社会において非常に身近で高頻度に現れる、配送計画問題を取り扱った。配送計画問題に対してより効率的な計算を行うべく、データベースを用いた反復局所探索法を提案し

た。さらに数値実験により、提案法が効果的に動作していることを確認した。

配送計画問題は2つの性質を持っている。割り当て問題の性質と、順序付け問題の性質である。今回、提案法であるデータベースが活躍したフィールドは、巡回セールスマン問題すなわち順序付け問題の分野である。何らかの方法で割り当てさえ行ってしまうと、あとはデータベースが最適な順番を教えてくれる。1つの巡回路の顧客数が多ければ多いほど、巡回セールスマン問題としては難しくなる。また、本稿では時間枠を考えなかったが、時間枠制約が生じれば、これもまた巡回セールスマン問題として難しくなる。こういった、巡回セールスマン問題の良質な解を得るために多大な時間を要する場合、本稿の提案法はより活かせると考える。難しく時間がかかるプロセスを、データベースの活用によって省くことができるからである。

今回行った実験では2つの構築法を採用した。セービング法では、計算時間に大きなバラつきが生じた。また、ファーストフィット法では、計算時間は安定したものの、セービング法に比べて解の精度が悪くなってしまった。しかしこれは問題によって状況が変わる。需要と容量のバランスによって、セービング法における計算時間のバラつきが消える例や、ファーストフィット法とセービング法の解精度の差がなくなる場合があることを確認した。従って、各問題に対し提案法と相性の良い構築法を見つけ、誰もが納得する形で配送計画問題を解いていくことが今後の課題である。

参考文献

- [1] J. Braca, J. Bramel, B. Posner and D. Simchi-Levi: A Computerized Approach to the New York City School Bus Routing Problem, IIE Transactions, 29, 1997, 693-702.
- [2] J. Bramel, D. Simchi-Levi: A Location Based Heuristic for General Routing Problems, Operations Research, 43, 1995, 649-660.
- [3] G. Clarke, J. W. Wright: Scheduling of Vehicles from a Central Depot to a Number of Delivery Points, Operations Research, 1, 1964, 568-581.
- [4] T. Ibaraki, S. Imahori, M. Kubo, T. Masuda, T. Uno, M. Yagiura: Effective Local Search Algorithms for Routing and Scheduling Problems with General Time Window Constraints, Transportation Science, 39, 2005, 206-232.
- [5] 久保幹雄, 田村明久, 松井知己 編集: 応用数理計画ハンドブック, 朝倉書店, 2002.
- [6] G. Laporte, I. H. Osman: Routing Problems : A Bibliography, Annals of Operations Research, 61, 1995, 227-262.
- [7] J. K. Lenstra, A. H. G. Rinnooy Kan: Complexity of Vehicle Routing and Scheduling Problems, Networks, 11, 1981, 221-227.
- [8] K. Mehlhorn, P. Sanders 著, 浅野哲夫 訳: Algorithms and Data Structures - The Basic Toolbox, シュプリンガー・ジャパン, 2009.
- [9] 大沢裕 著: データ構造とアルゴリズムのエッセンス, 昭晃堂, 2006.
- [10] 山本 芳嗣, 久保幹雄 著: 巡回セールスマン問題への招待, 朝倉書店, 1997.
- [11] Transportation Optimization Portal - TOP. <http://www.sintef.no/vrptw>