

事例研究 [論文]

ボロノイ分割を用いた加盟店舗への 動的な注文割り当て手法

川本 卓樹, 蓮池 隆

1. 研究背景および目的

スマートフォンやPCを代表としたインターネットデバイスが地域・世代問わず広範にわたって普及した現代において、ECサイトを通じて商品やサービスの恩恵を受ける消費者は年々増加傾向にある [1]。経済産業省 [2] が行った EC サイトに関する市場調査によれば、企業も消費者のニーズに呼応する形で、ECサイトを新規に開設し、ECサイト市場全体の規模もますます大きくなってきている。特に、店頭販売による売り上げは市場の成熟化によって近年伸び悩んでしまっているため、企業が販売の場を実店舗からECサイトへと拡大することによって、新しい客層を確保することが狙いであろう。

ECサイトを導入することで新規の顧客を獲得することが期待されるが、その反面、サイトを導入するに至るまで、あるいは導入後にもさまざまな障壁が存在する。たとえば、サイトを導入するために必要な初期費用や、導入した後も安定してサイトを運営するための運用コストなどのコストの問題が挙げられる。また、サイトを運営していくために必要なノウハウがその企業に備わっているかどうかという問題も存在する。こうしたさまざまな障壁を乗り越えるために、昨今ある一つのECサイト運営企業がECサイトの開発や運営を担当し、その企業に対して、これまで店頭販売しか行ってこなかった個人店舗や企業などが「加盟」し、開発されたプラットフォームを利用することによって、ECサイトで商品を取り扱うことを可能にするような販売形態をしている組織も存在しつつある。本研究では、特定の商材を取り扱うECサイトについて考える。なお、各加盟店で商材ごとに品質の違いが生まれない

ものとする。

このような形態で商品を販売している組織では、ECサイト企業が顧客から注文を受けた後、どの加盟店に対して依頼された注文を割り当てておくべきかを考える必要があるが、この割り当てを考える際にはさまざまなことを考慮しなければならない。代表的な例として、割り当てた先の加盟店が注文内容をすべて満たすことができるか、割り当てた先の加盟店と配送先までの距離は現実的に配送可能であるか、加盟店がロイヤリティに対して不平等を感じないように割り当てることができるかなどが挙げられる。

現在、特に配送距離に関して考える際に、郵便番号によって地域区分を行い、その区分けによって注文の割り当てを行っている事例が存在する。これは、ECサイトで注文を依頼する顧客が入力する顧客情報の中に、郵便番号が含まれており、この情報が大きな地域のみをまとめるのに適しているためである。しかし、郵便番号での地域の区分けでは、県や市区町村をまたいだ割り当てが優先されづらくなってしまい、市境や県境など、郵便番号の変化する境の近辺を配送先とする注文が、非効率な配送になってしまっている可能性が考えられる。また、配送店舗を決める際に、諸制約を鑑みながら注文を各加盟店へ割り当てる業務の一部をオペレータが手動で行っているケースも少なくないが、オペレータが手動で割り当ててしまうと、より配送先との距離が近い加盟店が存在するにもかかわらず、異なる店舗に注文を依頼してしまうなど、人的な判断によるロスが発生している可能性も考えられる。

また、注文割り当ての公平性については、オペレータによる割り当てによって、加盟店従業員との間の親密さなどから、意図せずオペレータが同じ加盟店にばかり割り当ててしまう場合も存在する。

そこで本研究では、ボロノイ分割を利用した提案手法による新たな配送地域の区分けによって、より効率的な地域区分を実現する。さらに、この提案手法を含んだ注文割り当てアルゴリズムを構築することによ

かわもと たかき, はすいけ たかし
早稲田大学大学院創造理工学研究科
〒169-8555 東京都新宿区大久保3-4-1
taka1023@suou.waseda.jp
thasuike@waseda.jp
受付 22.4.14 採択 22.8.19

て、現状オペレータによってマンパワーで行われている注文割り当て業務に代わるような、自動で加盟店へ注文の割り当てを行うアルゴリズムを構築する。以上を実現することで、マンパワーに依存しない注文の割り当て方や、より効率的な割り当て方の模索を行い、オペレータの注文割り当て業務の支援や、オペレータや加盟店の負担軽減への貢献につなげることを本研究の目的とする。

2. 問題状況および関連研究

2.1 問題状況

関連研究を述べる前に、本研究が想定する問題状況について記述を行う。本研究では、EC サイト運営企業が、店舗ごとに商材の品質に違いが生まれないような、ある特定の商材を取り扱う EC サイトの開発や運営を担当し、その企業に対し実際に商材を取り扱う店舗が「加盟」することにより、EC サイトでの販売を実現しているような組織・販売形態に焦点をあてる。よって、注文を依頼する加盟店を顧客ではなく EC サイト企業側で選択できるものとする。

そのような販売形態をとっている組織では、まず EC サイト運営企業が顧客から注文を依頼された際に、提携している加盟店に対して、さまざまなことを考慮しながら注文を割り当てるという手続きが発生する。

EC サイト運営企業から各加盟店へ注文が割り当てられた後、各加盟店がそれぞれの注文を満足する商品を用意し、指定された配送先へ指定された日にその商品を配送する。この際に発生する「配送料」は、商品代以外に顧客が事前に支払っている一律の手数料の内に含まれているものとする。したがって、配送距離の伸び縮みによって、配送料は変化しない。

商品の配送が完了すると、一定期間ごとに各加盟店へ割り当てられた注文の金額に応じて、各加盟店に対してロイヤリティが支払われる。

2.2 関連研究

2.1 節で述べたような問題状況は、注文を配送する店舗である配送元が複数存在する複数デポの物流形態であることから、共同物流における物流コスト最小化を目的とした最適化問題として書き表すことのできる可能性がある。このようなモデルは一般に共同物流モデルとして知られ、研究が行われている。

丹下 [3] は、日本の共同物流の端緒や最近の動向について編んだ。丹下によると、共同物流はすでに日本においては相当に普及した概念であり、その端緒は 1970 年初頭といわれている。その後長らく、共同物流の効果

は定性的にしか示されてこなかったが、丹下は単独物流と共同物流を比較し、積載率がどれほど向上したか、作業の効率化がどれほどなされたかを定量的に示し、共同物流形態の有用性を示した。

また、共同物流のモデル化についてもすでにくつつかの研究が行われている。共同物流は巡回路型のモデルで書き表すことができることから、巡回セールスマン問題や、より複雑な問題状況を表現することができる配送計画問題で定式化されることがある。小野山ら [4] は、マルチステージ GA を用いることで、共同物流網配送計画問題を実用的な時間内に求解まで導いている。また、内田ら [5] は、共同物流センターを建設せず、配送中に他の企業（他デポ）に立ち寄ることを許すことで共同配送計画を実現させる「他企業デポ立寄り型共同配送」の数理モデルを、Simulated Annealing と Tabu Search を組み合わせたヒューリスティック探索によって求解まで導いている。

また、2.1 節の問題状況は、すでに配置されている各加盟店を、施設配置可能な候補の集合と見立てることで、施設配置問題の p-メディアン問題に置き換える可能性も考えられる。

以上の共同物流や p-メディアン問題を、本研究のような問題状況に適用するにはいくつか問題点が挙げられる。まず、共同物流は、巡回路型のモデルであるため、距離の最小化を目的関数へ考慮する際には、加盟店と注文の配送先の組合せに応じて距離を列挙した距離行列を考える必要がある。しかし、本研究では数多くの店舗が加盟し、店舗だけで何千店舗と存在しているため、注文一つ一つに対してすべての加盟店との距離を列挙していくことは、問題の規模が大きくなるほど困難になっていくと考えられる。

また、共同物流と p-メディアン問題は、双方とも注文（需要）が確定している状況で最適化問題を解かなければならないか、あるいは注文内容や届け先に関する情報が所与である状況から最適化問題を解くことが前提となるが、本研究はそのような前提が成り立たない問題状況である。本研究の問題状況では、顧客から次々と時系列で注文が依頼される（需要が時間を追うごとに徐々に判明していく）状況において、その先どのような需要が発生するかわからない状態で注文を割り当てなければならない。また、逐次発生する需要の中には、当日配送のように喫緊に配送しなければならない注文も含まれている。以上から、需要がすべて判明している状況で問題を解くことができず、当日配送のような突発的な需要も比較的頻繁に発生するため、

注文をある程度溜める際の基準を設けることも難しい。加えて、需要が逐次発生することで、巡回路型の問題を解く際にも、需要が新たに追加されるごとに再求解が行われるため、大きな計算コストをかけてしまう。

以上を踏まえると、最適化問題を定式化し、そこへある程度溜まった注文を入力するという方法ではなく、その段階においての最適な加盟店へ逐次動的に注文を割り当てていくような手法であることが望ましい。

3. 提案手法

3.1 考慮する制約

注文を各加盟店へ割り当てる際、距離だけを考えるのではなく、さまざまな制約を考慮しながら割り当て先の加盟店を決める必要がある。

本研究では、以下の制約を割り当て先加盟店選定時の主な制約として定める。

1. 取り扱い品目を割り当て先加盟店が満たすことができるかどうか
2. 架電する日（＝注文を受けた日）に割り当て先加盟店が営業日かどうか
3. 注文を配送する日に割り当て先加盟店が営業日である、あるいは営業日ではない場合に配送することが可能な日であるかどうか
4. その他（組織脱退制約・手配の一時停止制約など）
5. （繁忙期用商品のみ）各加盟店の繁忙期間中における繁忙期用商品の注文割り当て数が上限に至っているかどうか

本研究では、注文に対して比較的余裕をもって対処することができる「通常期」と、注文が殺到している「繁忙期」についてそれぞれ考え、各期間に対して制約を設けることとする。

通常期は、制約 1～4 のみを考慮して注文を割り当てる。通常期は顧客からの注文に対して比較的余裕をもって対処できるため、どの加盟店でも商品の在庫を切らすほど注文が殺到することはない。そのため、加盟店ごとの注文割り当て数の上限は設定せず、注文を割り当てる。

一方、繁忙期は制約 1～5 を考慮して注文を割り当てる。繁忙期は、通常期にも取り扱っている商品に加えて、繁忙期にのみ取り扱っている商品も存在する。本論文では通常期にも繁忙期にも取り扱っている商品を「通常商品」と呼び、繁忙期にのみ取り扱っている商品を「繁忙期用商品」と呼ぶこととする。繁忙期には、特に繁忙期用商品の注文が殺到するため、注文が依頼された場合は、注文を溜める間もなく次々と注文

を加盟店へ割り当てる必要がある。そのため、規模の小さな店舗では次第に商品の在庫が減少し、注文を割り当てられても、その注文の商品を準備することができないために、配送できない事態が生じてしまう。以上の事態を未然に防ぐため、繁忙期用商品に関しては注文割り当て数の上限を設定し、繁忙期間中に受け入れられる上限を超えて、注文が各加盟店へと割り当てられないことがないように制約を設ける。ただし、繁忙期間中も取り扱っている通常商品に関しては、繁忙期用商品とは別の商品という扱いであるため、特に上限の対象とはならず、もし繁忙期用商品の上限に達してしまっている店舗に対しても、通常商品は引き続き割り当て可能であることとする。

以上のような制約を各期間で設け、注文の割り当てを自動で行うアルゴリズムを構築していく。

3.2 用いる既存手法

本研究では、従来の郵便番号による地域区分に代わる新たな地域区分を生成する分割方法として、ボロノイ領域を生成するボロノイ分割 [6] と呼ばれる分割手法を用いる。ボロノイ領域は、それに対応するボロノイ母点が最寄りとなる点の集合であるため、各加盟店を母点とみなすことで、各注文の最寄りの加盟店に割り当てることを可能とする。また、前述した制約を満たさなかった際、隣接するボロノイ領域とボロノイ辺を共有するボロノイ領域をたどることで、近くの加盟店を効率的に見発することに寄与する。

なお、本研究においては、緯度・経度を直交座標として用いてボロノイ分割を行っているため、球面上（地球楕円体上）では必ずしも正しい分割とはならない。よって、ボロノイ領域の境界付近の注文に関しては、必ずしも最短距離の店舗を選択できていない可能性も考えられる。これによる影響は大きくないものと考えられるが、上記の点に関して注意されたい。

また、各注文の点位置決定に際しては、Even-Odd rule (Ray Casting Algorithm) [7] を用いる。

3.3 提案アルゴリズム

2.2 節で述べた関連研究を本研究へ適用することは、需要があらかじめ判明していないことなどから困難である。そこで、3.1 節の制約を考慮しながらボロノイ分割を用いて注文を時系列で逐次割り当てるアルゴリズムの構築を行う。

特に本研究では、

- 加盟店の総配送距離がより短くなることに重点を置いた「距離優先アルゴリズム」
- 加盟店ごとに割り当てられた注文の総額がより等

しくなることに重点を置いた「利益平準化優先アルゴリズム」

の二つを提案する。

以下に、二つのアルゴリズムの手続きを具体的に記述していく。

3.4 距離優先アルゴリズム

距離優先アルゴリズムは以下の手順を踏んで、依頼された注文を時系列で逐次動的に処理する。

1. 対象加盟店でボロノイ分割を行い、それぞれのボロノイ領域が隣接しているボロノイ領域をあらかじめ特定する。
2. 多角形の内外判定アルゴリズムにより、配送先がどのボロノイ領域の加盟店に最も近いかが特定する。
3. Step2 で特定された加盟店が制約をすべて満たしているかどうかを判定する。
4. Step3 の判定により以下の分岐を行う。
 - (a) すべて満たした場合はその加盟店へ注文を割り当て、Step6 へ。
 - (b) 満たせない場合は、Step1 であらかじめ特定した、その加盟店のボロノイ領域に隣接しているボロノイ領域のうち、すべて制約を満たしている加盟店を抽出する。
5. Step4-(b) の抽出結果により以下の分岐を行う。
 - (a) 抽出された加盟店のうち、最も配送先との直線距離が近い加盟店へ注文を割り当てる。
 - (b) 抽出された加盟店がない場合は、未配送とする。

6. 次の注文が依頼されたら Step2 へ。

3.5 利益平準化優先アルゴリズム

利益平準化優先アルゴリズムは以下の手順を踏んで、依頼された注文を時系列で逐次動的に処理する。

1. 対象加盟店でボロノイ分割を行い、それぞれのボロノイ領域が隣接しているボロノイ領域をあらかじめ特定する。
2. 多角形の内外判定アルゴリズムにより、配送先がどのボロノイ領域の加盟店に最も近いかが特定する。
3. Step2 で特定された加盟店と、Step1 であらかじめ特定した、その加盟店に隣接しているボロノイ領域の加盟店のうち、制約をすべて満たしている加盟店のみを抽出する。
4. Step3 の抽出結果により以下の分岐を行う。
 - (a) 抽出された加盟店のうち、その段階で注文の総額が最も少ない加盟店へ注文を割り当てる。ただし、注文の総額が等しい加盟店が複数存在する場合、注文の総額が等しい

加盟店のうち、最も配送先との直線距離が近い加盟店へ注文を割り当てる。

- (b) 抽出された加盟店がない場合は、未配送とする。

5. 次の注文が依頼されたら Step2 へ。

4. 数値実験

提案アルゴリズムの実用性を検証するための数値実験を実施する。実験に用いるデータは、実際に 2.1 節で述べた問題状況のような販売形態で運営を行っている組織において、3.1 節で述べた制約を考慮しながらマンパワーで注文を割り振っている実績がある実データを用いる。提案した二つのアルゴリズムにより得られた注文の割り当て結果と、オペレータが実際に割り当てた実績の結果を比較することにより、提案アルゴリズムが実用に耐えうるものであるかの検証を行う。

4.1 通常期の実験条件

通常期の数値実験では、東京都・北海道の二つの地域に対してそれぞれ提案した二つのアルゴリズムを適用し、検証を行う。なお、どの地域においても通常期の実験対象期間はある年の 11 月 1 日～11 月 30 日の 1 ヶ月間とし、その期間に顧客から依頼された注文を実験対象の注文とする。また、考慮する制約は、3.1 節の制約 1～4 の制約とする。加盟店や注文の配送先の緯度経度に関するデータの一部は実データに存在しないため、Google Maps Platform [8] の API を利用することで緯度経度の取得を行った。

4.1.1 東京都における実験条件（通常期）

[対象注文数] 7,425 件（配送先の郵便番号が東京都に該当している注文）

[対象加盟店] 3,082 店（企業に加盟している全国の加盟店）

一度注文された後にキャンセルされた注文や、加盟店ではなくほかの配送業者に配送が委託された注文、顧客が注文の際入力した住所に何らかの大きな不備があり、Google Maps Platform からエラーが返ってきた注文、Google Maps Platform から、明らかに異なる都道府県の緯度経度が返ってきた注文などは実験で対象とする注文から除いている。除いて残った注文のうち、顧客が注文の際入力した配送先の郵便番号が東京都に該当している注文を東京都の対象注文とした。以下すべての地域の対象注文に対し、上述した方法によって前処理が行われている。

東京都における数値実験では、全国の加盟店を対象の加盟店としてボロノイ分割を行い、注文割り当ての

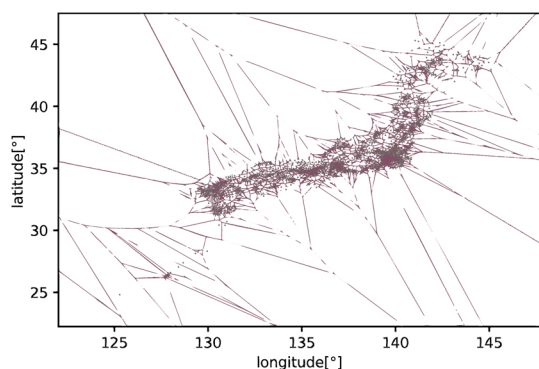


図1 全国の加盟店を用いたボロノイ分割

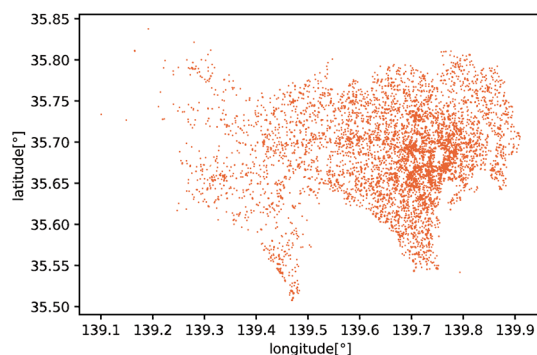


図3 通常期東京都内の配送先の位置

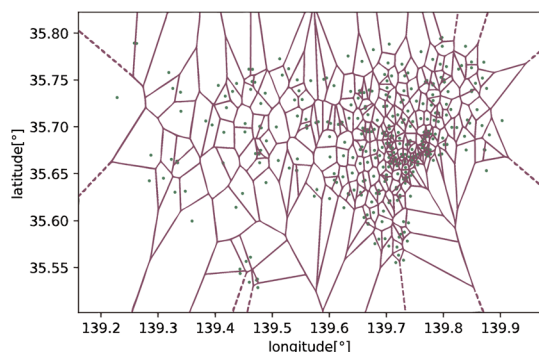


図2 東京都内加盟店のボロノイ図

対象とした。対象を東京都内の加盟店のみとしない理由は、東京都の場合、都外の加盟店に対して割り当てることが最も効率的なケースが存在するためである。たとえば、東京都町田市の注文を、町田市に隣接する神奈川県川崎市や相模原市の加盟店が配送することは往々にしてあるだろう。

全国の加盟店でボロノイ分割を行ったボロノイ図を図1に示す。実験ではこのボロノイ図を用いて割り当てを行っている。なお、図1では東京近郊のボロノイ領域が不明瞭であるため、東京都内のみの加盟店でボロノイ分割を行ったボロノイ図を図2に示す。縦軸は緯度、横軸は経度である。

また、対象期間の注文すべての配送先の位置を図3に示す。

4.1.2 北海道における実験条件（通常期）

[対象注文数] 2,991 件（配送先の郵便番号が北海道に該当している注文）

[対象加盟店] 245 店（加盟店のうち北海道内に位置する加盟店（220 店）＋ダミー店舗（25 店））

北海道における数値実験では、道内加盟店のみを注文割り当ての対象加盟店として実験を行う。これは、

配送先住所が北海道である注文が北海道以外の地域に属する加盟店へ割り当てられることは、配送距離や他県と隣接していないという特殊な地理状況を考えると現実的でないためである。実際、実データのオペレータ実績において、「配送先住所が北海道である注文が割り当てられた加盟店」に他県の加盟店が存在するかどうかを事前に調べたところ、すべての注文が道内の加盟店へ割り当てられていることが確認できた。よって、提案アルゴリズムにおいても、北海道では道内加盟店のみを割り当ての対象加盟店とした。

道内加盟店のみを注文の割り当て対象とする際に、ダミー店舗を導入した。ダミー店舗の詳細は、付録を参照されたい。

なお、隣接するボロノイ領域の判定を行う際に、配送先店舗候補としてダミー店舗が選ばれることを防ぐため、ダミー店舗は取り扱うことのできる商品がなく、常に定休日であり配送可能な日も存在しないとあらかじめ設定し、ダミー店舗へ注文が一切割り当てられないよう施す。

ダミー店舗を導入してボロノイ分割を行った図を図4に示す。

北海道における対象期間の注文すべての配送先の位置を図5に示す。

4.2 繁忙期の実験条件

繁忙期の数値実験においても、東京都・北海道の二地域に対してそれぞれ提案した二つのアルゴリズムを適用し、検証を行う。なお、実データをご提供いただいている企業の繁忙期が4月初旬～5月中旬であるため、対象期間を通常期実験期間翌年の4月1日～5月31日の2ヵ月間として実験する。

3.1 節で述べたように、繁忙期間中には大きく分けて二つの商品が存在しており、本論文ではそのうちの一つを「通常商品」と呼び、もう一つを「繁忙期用商

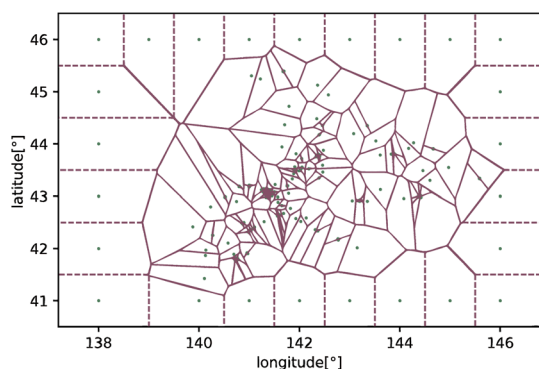


図 4 ダミー店舗を追加した際のボロノイ図

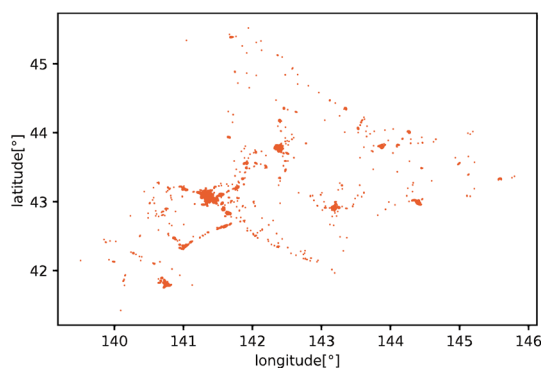


図 5 通常期北海道の配送先の位置

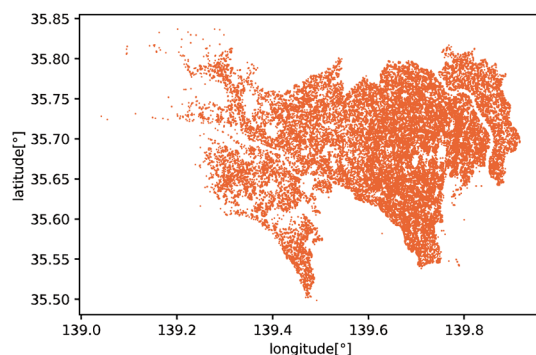


図 6 繁忙期東京都の配送先の位置

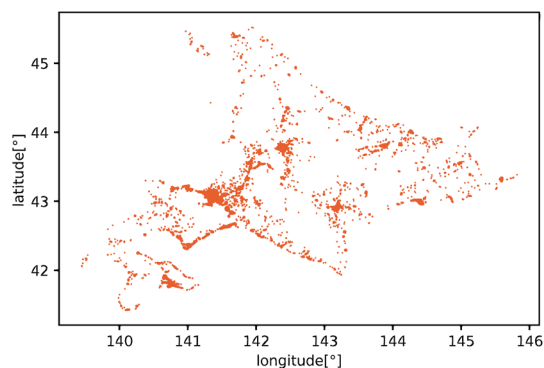


図 7 繁忙期北海道の配送先の位置

品」と呼ぶこととする。通常商品は、繁忙期間中でも繁忙期の制約ではなく通常期の制約を考慮して注文の割り当てを行う。すなわち、通常商品は 2.2 節の制約 1～4 までの制約を考慮しながら注文を割り当てる。一方、繁忙期用商品は加盟店ごとに「繁忙期間全体を通して注文を受け入れることが可能な件数」という上限件数があるため、制約 1～5 までの制約を考慮しながら注文を割り当てる。

4.2.1 東京都における実験条件（繁忙期）

[対象注文数] 41,978 件（配送先の郵便番号が東京都に該当している注文）

[対象加盟店] 3,005 店（企業に加盟している全国の加盟店）

対象期間が 2 ヶ月であることに加えて繁忙期間であることから、通常期の実験条件と比較して対象注文数が著しく増加していることがわかる。

加盟店は、対象期間が通常期の実験を行った翌年になったことで加盟店の組織への加盟・脱退があったものの、おおよそ通常期と同様である。そのため、加盟店の位置やボロノイ図については、図 1 および図 2 を参照されたい。また、対象期間の注文すべての配送先

の位置を図 6 に示す。

4.2.2 北海道における実験条件（繁忙期）

[対象注文数] 21,773 件（配送先の郵便番号が北海道に該当している注文）

[対象加盟店] 240 店（加盟店のうち北海道内に位置する加盟店（215 店）+ ダミー店舗（25 店））

繁忙期の北海道における実験条件も、通常期同様、道内加盟店のみを対象とし、ダミー店舗を適当な場所へ追加する。東京都同様、加盟店の位置およびボロノイ図は、通常期の図 4 とおおよそ同様であるため、そちらを参照されたい。また、対象期間の注文すべての配送先の位置を図 7 に示す。

4.3 実験環境

研究における数値実験は、OS: Windows10 64bit, CPU: Intel (R) Core (TM) i5-8250U Processor@ 1.60GHz, 8.00GB RAM を搭載した PC で行った。

また、実データを実験に供する際の前処理、ボロノイ分割や提案アルゴリズムの構築などはすべて Python3.7.4 を用いた。特にボロノイ分割を提案アルゴリズムで用いる際には、scipy.spatial.Voronoi [9] を利用してボロノイ分割を行った。

4.4 評価指標

距離優先アルゴリズム、利益平準化優先アルゴリズム、オペレータが実際に割り当てた実績それぞれの比較を行うための評価指標を設定する。評価指標には、距離と利益のばらつき具合の二つを用いる。

- 距離：注文の配送先とそれを配送する加盟店との直線距離の合計
 - 利益のばらつき具合：対象期間、対象都道府県の加盟店ごとに割り当てられた注文の総額の標準偏差
- 距離は、注文の配送先の座標と、その注文を配送する加盟店の座標の緯度と経度を用いて、測地線距離と呼ばれる、楕円体における二点間の最短距離を評価値としている。なお、測地線距離の計算には楕円体モデルに WGS-84、計算アルゴリズムに Karney 法を使用した `geopy.distance` [10] を用いている。利益のばらつき具合は、対象都道府県の加盟店ごとに割り当てられた注文の総額の標準偏差を評価値としている。すなわち、東京都の通常期では 3,082 店を対象に注文を割り当てているが、利益のばらつき具合の評価の際には、分母を東京都内の加盟店 (343 店) として計算を行っている。北海道は、実験条件の対象加盟店のうちダミー店舗を除いた加盟店数で計算を行っている。

前述したように、東京都の場合、神奈川県や埼玉県など隣接する加盟店に注文を割り当てる場合が存在する。そのため、対象都道府県である「東京都内のみの加盟店」を評価の対象とすると、オペレータによる割り当てとアルゴリズムによる割り当てでは、注文全体の総額が一致しなくなるデメリットが生じる。しかし、加盟店の対象を「注文が割り当てられた加盟店」とすると、東京都内の加盟店にもかかわらず、注文が一度も割り当てられなかった加盟店が対象とならない。すなわち、最も利益を享受できていない加盟店が対象とならないため、注文の総額が一致しないというデメリットが生じるが、「注文が割り当てられた加盟店」ではなく、「対象都道府県の加盟店」に割り当てられた注文に関して、加盟店ごとの総額の標準偏差を求めるよう設定した。そのため、東京都内では注文の総額が一致していないということに注意されたい。一方で、北海道は、必ず北海道内に位置する加盟店に割り当たるため、注文の総額は一致している。よって、利益のばらつき具合を比較する際には、特に北海道を参照されたい。

また、提案手法にて未配送となった注文が発生した場合は、オペレータの評価の不利とならないよう、その注文に関してオペレータの実績からも評価対象から除外する。

表 1 通常期東京都の割り当て結果

	距離 [km]	利益のばらつき [円]	未配送 [件]
距離優先	6,700.10	132,145.28	3
利益優先	17,300.28	38,389.24	3
オペレータ	16,399.84	296,392.55	0

表 2 通常期北海道の割り当て結果

	距離 [km]	利益のばらつき [円]	未配送 [件]
距離優先	10,402.39	85,976.78	2
利益優先	44,007.31	28,734.50	2
オペレータ	15,802.71	142,574.06	0

5. 実験結果

5.1 通常期の割り当て結果

4.3 節で述べた実験環境のもと、距離優先アルゴリズム・利益平準化優先アルゴリズムを実データに適用した。東京都の結果を表 1、北海道の結果を表 2 としてそれぞれ示す。なお、表では距離優先アルゴリズムを「距離優先」、利益平準化優先アルゴリズムを「利益優先」、オペレータ実績を「オペレータ」と表記する。また、評価指標の単位はそれぞれ、距離を「km」、利益のばらつき具合を「円」で表し、未配送となった注文件数とともに示す。

両表より、どちらの地域でも距離優先アルゴリズムの距離・利益のばらつき具合はオペレータ実績と比較して、より優れた結果を示した。利益のばらつき具合に関しては、距離優先アルゴリズム内では特に考慮していないが、オペレータ実績よりも小さくなる結果となった。これは、オペレータ実績において明示されていない潜在的な制約も考慮して割り当てている可能性や、エリアごとにオペレータが思い浮かべる加盟店が存在し、そこへ無意識のうちに偏って割り当ててしまっている可能性が考えられる。

利益平準化優先アルゴリズムでは、特に利益のばらつき具合がどちらの地域でもオペレータ実績や距離優先と比較して優れた結果を示した。一方、距離の指標は東京ではオペレータ実績と比べやや劣った結果を示し、特に北海道では 3 倍程度劣った結果を示した。これは、東京など面積が小さな地域に加盟店が密に存在した状況では、隣接するポロノイ領域の加盟店に注文を割り当てても距離が大きくは伸びない一方、北海道などの面積が大きな地域に加盟店が疎に存在した状況

表 3 繁忙期東京都の距離優先アルゴリズム割り当て結果

	距離 [km]	利益のばらつき [円]	未配送 [件]
距離優先	50,014.54	376,731.14	432
オペレータ	94,333.05	742,227.57	0

表 4 繁忙期東京都の利益平準化優先アルゴリズム割り当て結果

	距離 [km]	利益のばらつき [円]	未配送 [件]
利益優先	116,747.75	125,713.70	239
オペレータ	94,836.60	744,686.10	0

表 5 繁忙期北海道の距離優先アルゴリズム割り当て結果

	距離 [km]	利益のばらつき [円]	未配送 [件]
距離優先	97,761.64	417,473.65	117
オペレータ	123,567.99	449,665.17	0

表 6 繁忙期北海道の利益平準化優先アルゴリズム割り当て結果

	距離 [km]	利益のばらつき [円]	未配送 [件]
利益優先	349,959.92	142,647.98	53
オペレータ	123,506.81	452,006.20	0

では、隣接するボロノイ領域に割り当てると距離が一気に伸びてしまうためと考えられる。

また、未配送注文は両地域でほとんど発生することがなかった。隣接するボロノイ領域までを候補となる店舗群として考えれば、注文の多くを未配送とせずとも、提案アルゴリズムにより割り当てることができる。なお、未配送注文に関して、両地域において提案手法の未配送件数がそれぞれ一致しているが、これらは未配送となった注文内容そのものに関しても、いずれも一致している。これは、提案アルゴリズム上、両アルゴリズムともその注文が属するボロノイ領域とそれに隣接するボロノイ領域までが配達可能範囲である点は同様であるため、特に未配送注文が少数のうちは、未配送となる注文が同様となることであると考えられる。

5.2 繁忙期の割り当て結果

繁忙期における、両地域の割り当て結果も通常期と同様に示していく。なお、繁忙期は距離優先アルゴリズムと利益平準化優先アルゴリズムで未配送注文が異

なり、それによってオペレータ実績で評価の対象外とする注文も異なるため、アルゴリズムごとに結果を示す。東京都の結果を表 3、表 4、北海道の結果を表 5、表 6 としてそれぞれ示す。

結果より、距離優先アルゴリズムでは、両評価指標で優れた結果を示していることがわかる。また、利益平準化優先アルゴリズムでは、利益のばらつき具合が両地域で大きく改善された結果を示した。一方、北海道のように面積の大きな地域では、距離の指標が著しく劣った結果を示した。面積の大きな地域では、利益平準化優先アルゴリズムを用いて割り当てると、配送距離が非常に長くなってしまい、却って各加盟店の負担になってしまうことが推察される。

また、北海道では、利益平準化優先アルゴリズムの未配送件数が距離優先アルゴリズムと比較して少ないにもかかわらず、オペレータ実績の距離が利益平準化優先アルゴリズムと比べて長い結果となった。これは、利益平準化優先アルゴリズムで未配送となった注文のうち、配送先との距離が非常に長い注文がオペレータ実績に多く含まれていたためであった。

両アルゴリズムとも、対象注文数の増加に伴い、未配送となる注文件数が増加傾向にあることがわかる。繁忙期における提案アルゴリズムでは、繁忙期用商品の割り当て数に上限が存在することによって、繁忙期の終盤になるにつれ割り当て可能な加盟店が減少していく。そのため、注文の配送先を縄張りとするボロノイ領域の加盟店、そのボロノイ領域に隣接するボロノイ領域の加盟店すべてが、3.1 節 5 番目の制約を満たすことができなくなる。その結果、未配送となる注文の件数が通常期と比較して増加したと考えられる。

以上から、提案アルゴリズムの適用によって、通常商品のように注文の上限が存在しない商品は、隣接するボロノイ領域の加盟店と上手く連携をとりながら、お互いの領域をカバーできることが明らかになった。一方、繁忙期における繁忙期用商品のように、加盟店ごとに注文の割り当て上限を設定しなければならない商品は、隣接する加盟店と連携をとったとしてもカバーできず、未配送注文の件数が増加してしまうことが明らかになった。

6. 追加実験

5.2 節で述べたように、繁忙期において提案アルゴリズムでは多くの未配送注文を発生させてしまう。しかし、顧客から依頼された注文が未配送となることは、現実では避けなければならない事態である。そこで、

3.3 節のアルゴリズムに改良を施し、未配送注文を可能な限り発生させないアルゴリズムの構築を行う。ただし、利益平準化優先アルゴリズムと比較して、距離優先アルゴリズムの方が両評価指標で安定した結果を示すことが明らかになったため、追加実験では特に 3.4 節の距離優先アルゴリズムを改良する。

距離優先 + 全探索アルゴリズムは以下の手順を踏んで、EC サイトで依頼された注文を時系列で逐次動的に処理する。

1. 対象加盟店でボロノイ分割を行い、それぞれのボロノイ領域が隣接しているボロノイ領域をあらかじめ特定する。
2. 多角形の内外判定アルゴリズムにより、注文がどのボロノイ領域の加盟店に最も近いか特定する。
3. Step2 で特定された加盟店が制約をすべて満たしているかどうかを判定する。
4. Step3 の判定により以下の分岐を行う。
 - (a) すべて満たした場合はその加盟店へ注文を割り当て、Step7 へ。
 - (b) 満たせない場合は、Step1 であらかじめ特定した、その加盟店のボロノイ領域に隣接しているボロノイ領域のうち、すべて制約を満たしている加盟店を抽出する。
5. Step4-(b) の抽出結果により以下の分岐を行う。
 - (a) 抽出された加盟店のうち、最も配送先との直線距離が近い加盟店へ注文を割り当て、Step7 へ。
 - (b) 抽出された加盟店がない場合は、全探索アルゴリズムを用いて、すべて制約を満たしている加盟店のうち、最も配送先との直線距離が近い加盟店を特定する。
6. Step5-(b) の特定結果により以下の分岐を行う。
 - (a) 加盟店が特定できた場合は、その加盟店へ注文を割り当てる。
 - (b) 加盟店が一店舗も特定されない場合は、未配送とする。
7. 次の注文が依頼されたら Step2 へ。

3.4 節の距離優先アルゴリズムと異なる手続きは、Step5-(b) 以降である。距離優先アルゴリズムでは、隣接するボロノイ領域ですべて制約を満たすことのできる加盟店が存在しない場合、未配送として注文を処理していた。一方、距離優先 + 全探索アルゴリズムでは、隣接するボロノイ領域のさらに外側に存在する加盟店を含めたすべての加盟店に対して、制約をすべて満たすことができるか判定を行う。その後、制約をす

表 7 繁忙期東京都の距離優先 + 全探索アルゴリズム割り当て結果

	距離 [km]	利益のばらつき [円]	未配送 [件]
提案	51,095.37	379,127.02	0
オペレータ	95,449.32	747,868.95	0

表 8 繁忙期北海道の距離優先 + 全探索アルゴリズム割り当て結果

	距離 [km]	利益のばらつき [円]	未配送 [件]
提案	99,578.28	418,267.27	0
オペレータ	124,846.11	451,912.80	0

べて満たすことのできた加盟店のうち、最も直線距離が近い加盟店へ注文を割り当てるように処理する。このような手続きを追加することによって、ほとんどのケースで注文をいずれかの加盟店へ割り当てることが可能となる。ただし、非常に稀なケースではあるものの、注文数が対象加盟店すべての上限の合計を超えてしまい、すべての加盟店が注文受け入れ不可能となる場合は未配送となるほか、すべての加盟店が満たせない制約を含んだ注文も、未配送となる。

距離優先 + 全探索アルゴリズムがどの程度実用に耐えるかを検証するため、実験を行った。対象地域はこれまで同様、東京都、北海道の二地域において、実験環境は 4.3 節と同様の環境で実験を行った。

追加実験における東京都の結果を表 7、北海道の結果を表 8 に示す。

表より、どの地域においても距離・利益のばらつき具合の評価指標ともに、オペレータ実績と比較してより優れた結果を示した。また、多く発生していた未配送注文も、件数を 0 件に抑えることができた。

7. 結論

本研究では、主に EC サイト運営企業とそこに加盟している加盟店が成す組織の販売形態において、顧客から注文を依頼された際に、どの加盟店へ注文を割り当てることがより効率的であるかについて、従来の郵便番号による地域区分やオペレータによる手動の割り当てに代わる、ボロノイ分割を用いた自動での注文割り当てアルゴリズムの構築を行った。

提案アルゴリズムでは、距離優先アルゴリズム、利益平準化優先アルゴリズム、距離優先 + 全探索アルゴリズムによるハイブリッドアルゴリズムの三つを構築

した。構築したアルゴリズムについて、2.1 節の形態で販売を行っている組織の実データを用いて数値実験を行い、オペレータ実績と比較してどの程度実用的な結果を示すか検証した。

その結果、通常期の距離優先アルゴリズムでは、距離と利益のばらつき具合それぞれの指標において、オペレータ実績と比較して安定して優れた結果を示すことができた。また、利益平準化優先アルゴリズムでは、注文数や加盟店数が少なく面積の大きい地域に対しては、配送距離が延びてしまう結果を示したが、東京都のように面積が小さく加盟店や注文の配送先が密集した地域に対しては、距離の指標もオペレータの実績と大きくは劣らない結果を示し、利益のばらつき具合の指標において、優れた結果を示すことができた。加えて、有用性の検証を行うことで、ボロノイ分割を用いた提案アルゴリズムが全探索アルゴリズムと比較して、注文の割り当てを 15 倍程度速く行うことができることも確認できた。また、通常商品に加えて加盟店ごとに上限の存在する繁忙期用商品を取り扱っている繁忙期においては、距離優先 + 全探索アルゴリズムを適用することで、距離優先アルゴリズムでは多く生じる未配送注文の件数を可能な限り低く抑えつつ、オペレータ実績と両評価指標を比較して、一定の改善効果を示すことができた。

最後に本研究の今後の課題を挙げる。本研究は、計算機上で実験を行ったため、制約を満たす加盟店に割り当てられた場合は必ず注文が受け入れられる。しかし、実現場では、制約以外の諸般の事情から断られるケースも多々存在するため、実現場での検証を行う必要があると考えられる。

また、本研究は、直線距離によって距離を比較したが、河や山など自然による障害があることで、実際には直線距離とは大きく乖離した配送距離となるケースも存在する。したがって、直線距離ではなく実際の道路状況に即した道路距離での比較を行い、ボロノイ分割に地理的な情報を組み合わせることで、より現実の地形に即した地域区分を行う必要があるだろう。

また、有界なポリゴン生成手法の際に用いるダミー店舗の設置に関して、よりシンプルで汎用的な方法をとる必要も考えられる。

また、本研究では、ある程度注文を溜めることはせず、どの注文も需要が発生した段階で割り当てが行われている。この発展として、顧客から注文を依頼された際、逐次加盟店へ動的に注文を割り当てることを原則として行いながら、注文日から配送日までに大きく

日の空いている注文に対しては、一度 EC サイト運営企業でその注文を溜め、その後の動的な割り当ての動向やほかに溜まっている注文との兼ね合いなどを鑑みながら静的な割り当てを行うことで、さらに配送の効率性を上げるアルゴリズムを構築する必要があると考えられる。

参考文献

- [1] 総務省, 「平成 29 年版情報通信白書」, <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h29/pdf/29honpen.pdf> (2020 年 12 月 18 日閲覧)
- [2] 経済産業省商務情報政策局情報経済課, 「平成 30 年度我が国におけるデータ駆動型社会に係る基盤整備 (電子商取引に関する市場調査) 報告書」, <https://www.meti.go.jp/press/2019/05/20190516002/20190516002-1.pdf> (2020 年 12 月 18 日閲覧)
- [3] 丹下博文, “日本の共同配送と米国の混載システムに関する研究,” 経営管理研究所紀要, **22**, pp. 41–57, 2015.
- [4] 小野山隆, 前川拓也, 久保田仙, 鶴田節夫, 薦田憲久, “マルチステージ GA による共同物流網における配送計画作成手法,” 電気学会論文誌 C (電子・情報・システム部門誌), **127**, pp. 1460–1467, 2007.
- [5] 内田哲也, 鈴木卓, 吉本一穂, “異なるサイズのトラックを有する企業における共同配送計画—SA, TS の Hybrid Heuristic の適用 (機械要素, 潤滑, 工作, 生産管理など)—,” 日本機械学会論文集 C 編, **68**, pp. 2825–2832, 2002.
- [6] 杉原厚吉, 『なわばりの数理モデル—ボロノイ図からの数理工学入門—』, 共立出版, 2009.
- [7] K. Hormann and A. Agathos, “The point in polygon problem for arbitrary polygons” *Computational Geometry*, **20**, pp. 131–144, 2001.
- [8] Google Maps Platform, 「位置情報 API」, <https://cloud.google.com/maps-platform/> (2020 年 12 月 18 日閲覧)
- [9] SciPy.org, [scipy.spatial.Voronoi](https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.Voronoi.html), <https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.Voronoi.html> (2020 年 12 月 18 日閲覧)
- [10] GeoPy, Calculating Distance, <https://geopy.readthedocs.io/en/stable/#module-geopy.distance> (2020 年 12 月 18 日閲覧)

付録

ダミー店舗について

母点集合 S に対して、 S に属するすべての点を含む最小の凸集合を S の凸包 (Convex Hull) といい、 $C(S)$ で表すとき、ボロノイ分割を行った際に、 $C(S)$ の境界上の母点は、ボロノイ領域が無制限遠方まで延びるという性質がある [6].

東京都の数値実験では全国の加盟店でボロノイ分割を行うが、東京都内およびその周辺の加盟店は Convex Hull 境界上の点が存在しないため、この性質を意識することなく提案アルゴリズムを用いることができる。一方、北海道の数値実験では、道内店舗のみでボロノイ分割を行うため、必ず Convex Hull 境界上に存在する

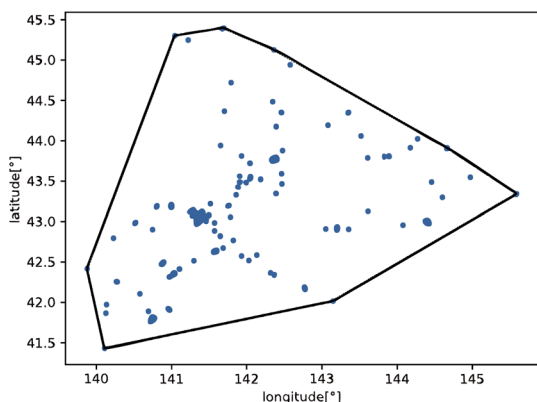


図 8 道内店舗における凸包 (Convex Hull)

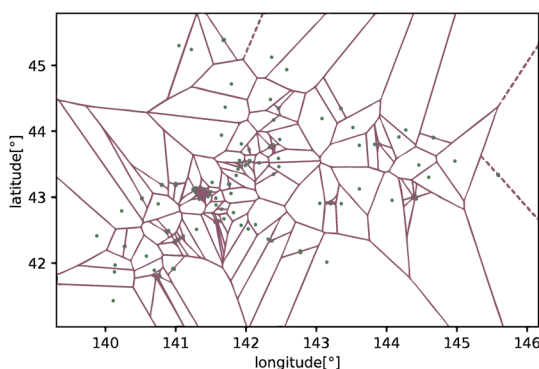


図 9 道内店舗のみでのボロノイ図

加盟店が発生してしまう。道内店舗における Convex Hull を図 8 として示す。

道内店舗における Convex Hull 境界上の加盟店は全部で 9 店舗あり、これらの店舗はボロノイ分割を行った際にボロノイ領域が適切に生成されない。実際に道内店舗のみでボロノイ分割を行ったボロノイ図を図 9 として示す。

図 8 で Convex Hull 境界上に存在する母点のボロノイ領域は、図 9 では、図の外側へと開いた状態で延びるボロノイ辺が無限遠方へ延びていき、ボロノイ領域が閉じないことがわかる。本研究の提案アルゴリズムでは、多角形の内外判定アルゴリズムを適用する際に、正しい判定結果を返すためには多角形であることが前提であるが、閉じていない領域は多角形とはいえないため、望ましくない結果が返されてしまう。

この問題を解消するため、図 4 のように実際には存在しない店舗をダミー店舗として Convex Hull のさらに外側に追加する。この操作を加えることで、追加前に凸包境界上に存在していたボロノイ領域でも、無限遠方へ延びていたボロノイ辺がダミー店舗のボロノイ

表 9 有用性検証における各評価指標と実行時間の比較

	距離 [km]	利益のばらつき [円]	実行時間 [秒]
提案	6,702.25	132,275.01	404.00
全探索のみ	6,656.17	133,276.63	6100.66

イ辺と交わり、領域を閉じることができる。ボロノイ領域が閉じていれば多角形と認識されるため、多角形の内外判定アルゴリズムにおいて望ましい結果が返される。

なお、ダミー店舗を追加すると、追加されたダミー店舗が新たな Convex Hull 境界上の点となる。よって、ダミー店舗のボロノイ領域は多角形の内外判定アルゴリズムを適用した際に、正しい結果を返さないということに注意されたい。ゆえに、注文がダミー店舗のボロノイ領域に入らないよう、地域ごとの配送先となる範囲をあらかじめ考えながらダミー店舗を追加する必要がある。

提案手法の有用性検証

一般的に知られているアルゴリズムよりも優れた結果を示すかどうかを調べるため、代表的なアルゴリズムとして「全探索アルゴリズム」を構築し、比較することで、その有用性を検証する。全探索アルゴリズムでは、距離優先アルゴリズムと同様、距離の指標を重視するため、「制約をすべて満たす加盟店のうち、最も直線距離が短い加盟店へ注文を割り当てる」という手続きに従って、アルゴリズムの構築を行った。実験は 4.1.1 節とすべて同様の条件とする。なお、距離優先アルゴリズムでは 3 件の未配送注文が発生するが、両アルゴリズムを比較する際には、同様の配送件数で比較を行うことが望ましい。したがって、6 節の距離優先 + 全探索アルゴリズムと、全探索のみを利用したアルゴリズムとで比較を行う。距離優先 + 全探索アルゴリズムと全探索のみのアルゴリズムの両評価指標および実行時間の比較を表 9 に示す。

各評価指標は、距離優先 + 全探索と全探索のみでおおむね同様の結果を示した。やや結果が異なっている原因は、隣接するボロノイ領域の加盟店よりも隣接していないボロノイ領域の加盟店の方が、直線距離上近くに存在するケースが稀にあるためと考えられる。そのほかにも、ボロノイ分割を行う際の、ボロノイ辺が必ずボロノイ領域として閉じるように調整する際に生じる数値誤差が考えられる。

表 9 より、ボロノイ分割を用いた提案アルゴリズムの方が全探索のみのアルゴリズムと比較して 15 倍程

度速い実行時間であることがわかる。ボロノイ分割を用いた提案手法では、多角形の内外判定アルゴリズムで、対象の点がいずれかのボロノイ領域の内側にあると判定された時点で次の処理へ進む。一方、全探索アルゴリズムでは各加盟店と配送先との直線距離を一度

すべて計算しなければ、最も近い加盟店を特定できないため、次の処理へ進むまでに時間を要する。この違いが、二つのアルゴリズムで実行時間に 15 倍程度の差を生んだと考えられる。