

宅配個数の均等な割り振りを考慮した時間枠付き配送計画問題の発見的解法と量子アニーリングによる計算支援

メタデータ	言語: Japanese 出版者: 公開日: 2023-03-14 キーワード (Ja): キーワード (En): 作成者: 中塚, 怜花, 佐々木, 多希子, 友枝, 明保 メールアドレス: 所属:
URL	https://mu.repo.nii.ac.jp/records/1979

宅配個数の均等な割り振りを考慮した 時間枠付き配送計画問題の発見的解法と 量子アニーリングによる計算支援

Heuristic algorithms considering distribution work amount
equally with time-window constraints
and calculation support by quantum annealing

中 塚 怜 花¹

Reika Nakatsuka

佐 々 木 多 希 子²

Takiko Sasaki

友 枝 明 保³

Akiyasu Tomoeda

概要

物流・運送業界は過酷な労働条件による人員不足において課題を抱えており、従来のアナログ業務を自動化することが求められている。本研究では、最終物流拠点からエンドユーザーへの配達ルートを自動で作成するシステムを提案する。特に、宅配個数の均等な割り振りに着目する。複数のドライバーがいる場合、ドライバー毎に労働の偏りを無くすためには宅配個数は均等に割り振った方がよい。しかし、適当に宅配個数を分けると、配送距離にばらつきが出るため、労働の偏りを無くしたことになる。アナログ作業で配送距離などを考慮しながら、宅配個数を割り振るには経験値が必要となり、これが業務の属人化につながる。この作業を自動化することで、物流・運送業界の業務効率の向上にプラスの効果が期待できる。配送ルートを求める問題は NP 困難で計算機を使っても厳密解を求めることが難しいことが知られており、多くの場合で近似解法を用いて解を探索する。本稿では、「発見的解法」と呼ばれる近似解法を用いて、宅配件数の均等な割り振りを考慮した時間枠付き配送計画問題の近似解を求める。また、この計算時間の問題を解決する手法の一つとして、量子アニーリングが着目されている。本稿ではルート作成の一部に量子アニーリングを取り入れて時間枠付き配送計画問題を解いた結果についても述べる。

1 はじめに

現在、物流業界においてデジタル・トランスフォーメーション（以下、DX）化が推進されている。特に、コロナ禍に伴う巣ごもり需要でインターネット通販の利用者が増え、国内にお

¹ 武蔵野大学工学部数理工学科 B4

² 武蔵野大学数理工学センター員 / 武蔵野大学工学部数理工学科講師 / 東北大学大学院理学研究科数学専攻講師

³ 武蔵野大学数理工学センター客員研究員 / 関西大学総合情報学部教授

ける宅配便取扱個数は年々増加している．令和 2 年度の宅配便取扱個数は，48 億 3647 万個であった（うちトラック運送は，47 億 8494 万個，航空等利用運送は 5153 万個）．これを前年度と単純比較すると，5 億 1298 万個・対前年度比 11.9 % の増加となる [9]．また，令和 3 年度の宅配便取扱個数は，49 億 5323 万個（うちトラック運送は，48 億 8206 万個，航空等利用運送は 7117 万個）であり，これを前年度と単純比較すると，1 億 1676 万個・対前年度比 2.4 % の増加となる [10]．それにも関わらず，物流業界ではアナログ作業が多く，この現状が人手不足や新人教育のコストの増加，属人管理作業の慢性化，再配達問題などの課題につながっている．物流業界でのアナログ作業を DX 化することで，これらの課題を解決することが期待できる．

本稿では，デポとよばれる最終物流拠点から，いくつかのエンドユーザーへの配達経路を求める問題を考える．デポからエンドユーザーまでの配送経路の作成は，NP 困難で非常に難しい問題であるにも関わらず，細かい制約条件が多く DX 化が難しいため，ドライバーの采配に委ねられることが多かった．配送経路の作成がドライバーの経験に依存するため，宅配の経験がないドライバーの新規参入が難しく，人手不足となり労働環境の悪化につながる．この課題に対し，ドライバーの采配に委ねられていた配送経路の作成を自動化することで，ドライバーの負担削減を試みる取り組みがいくつかなされてきた．その一つに，配送経路の作成を 配送計画問題とよばれる最適化問題として定式化し，デポやエンドユーザーの位置情報や配達の時間指定，配達物やトラックの容量から，トラックの移動距離や移動時間が最小となる 配送経路を出力する手法を考察する研究がある．

配送計画問題は代表的な最適化問題の一つで，理論と実務の両方の側面で多くの研究がある．例えば，所要時間の不確実性を考慮した [22] や，災害時における配送量の優先度を考慮した救援物資配送モデル [18]，ルートの探索を繰り返すごとに得られる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 [23] などがある．配送計画問題のうち，制約条件として配達物の容量を考慮する問題を容量制約付き配送計画問題とよび，さらに時間指定を制約条件として課した問題を時間枠付き配送計画問題とよぶ．配送計画問題は NP 困難で計算量が非常に多く，通常のコンピュータでは厳密解を求めることが困難であることが多いため，近似解法を用いて解くことがほとんどである．特に，制約条件を満たすルートの一つを構成し，そのルートを少しずつ改善していく発見的解法は近似解法の一つであり，多くの研究がある．その一部は，例えば橋本-胡 [7] にまとめられている．

一方で，近年，量子力学を使って情報処理をする手法の一つである量子アニーリングを用いて，最適化問題を解く研究が注目されている．カナダのベンチャー企業である D-Wave Systems により，量子アニーリング理論に沿って作られた装置である量子アニーリングマシンが開発された．量子アニーリングマシンにより，いくつかの最適化問題が，シミュレーテッドアニーリングなどの既存手法より速く，また少ない消費電力で解けることが知られ，実用化へ向けた取り組みがなされている．例えば，回路の故障解析 [4]，機械学習 [5]，交通流最適化 [16]，工場における無人搬送車の効率化 [15]，材料開発 [8] 等が報告されている．

本稿では，ドライバーの負担の偏りを無くすために宅配個数を均等に割り振ることを制約条

件として課した時間枠付き配送計画問題の近似解法を求める．複数のドライバーとトラックが配送作業ができる場合，労働条件を均等にするためにはドライバー毎に宅配個数を均等に割り振る必要がある．一番単純なのは適当に宅配個数を等分することだが，このように分けてしまうと，配送距離にばらつきが出てしまい，この意味で労働を均等に割り振ったことにならない．また，効率的に配送地域を割り振ることができず，トラックの移動距離の和が大きくなり，燃料消費量が増える．また，労働時間を効率的に使えず，人件費の増加にもつながる．アナログ作業で配送距離や配送地域を考慮しながら宅配個数を割り振るには経験値が必要となるため，これが業務の属人化につながる．宅配個数を均等に割り振る作業を最適化問題として定式化し，それを解く操作を DX 化することで，業務効率の向上につながることを期待できる．

さらに，提案手法の一部を量子アニーリングで実装し，その結果について考察する．本稿の構成は以下の通りである．第 2 節で，時間枠付き配送計画問題と量子アニーリングの先行研究について紹介する．第 3 節では，提案手法について述べ，第 4 章でまとめを述べる．

2 時間枠付き配送計画問題と量子アニーリング

本節では，時間枠付き配送計画問題と量子アニーリングについて簡単に紹介する．

2.1 時間枠付き配送計画問題の定式化

本節では，時間枠付き配送計画問題の定式化について述べる．顧客の場所および経路を表すグラフ G を $G = (V, E)$ ，節点集合 V を $V = \{0, 1, \dots, n\}$ ，枝集合 E を $E = \{(i, j) \mid i, j \in V, i \neq j\}$ とする．車両集合を $M = \{1, 2, \dots, m\}$ とする．各車両 $k \in M$ の訪問客数を n_k ，総訪問客数を n とする．すなわち， $n = \sum_{k=1}^m n_k$ とする．車両 k の訪問順列を $\sigma_k = (\sigma_k(1), \sigma_k(2), \dots, \sigma_k(n_k))$ とし，全車両のルートを $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_m)$ と表す．時間枠付き配送計画問題の代表的な仮定は次の通りである：

仮定 1 節点 0 はデポといわれる点で，物流拠点を目指す．各車両はデポを出発し，デポに帰還しなければならない．すなわち， $\sigma_k(0) = \sigma_k(n_k + 1) = 0$ とする．

仮定 2 0 以外の節点はサービスを受ける顧客の場所を表す．各顧客は，1 台の車両に 1 度だけ荷物を受け取る．

仮定 3 (時間制約) 各顧客を表す番号 $i \in V$ に対し，また，顧客 i は $[e_i, l_i]$ の時間枠で訪問することを要求しているとする．ここで， e_i, l_i は非負の定数で， $e_i \leq l_i$ を満たすとする． s_0 をデポを出発する時間， s_i ($i \in V \setminus \{0\}$) を各顧客の訪問開始時刻， \bar{s}_k を車両 k がデポへ帰還する時刻とし， $s = (s_1, s_2, \dots, s_n, \bar{s}_1, \bar{s}_2, \dots, \bar{s}_m)$ とする．また， $t_{\sigma_k(h), \sigma_k(h+1)}$ を車両 k が顧客 h の場所から顧客 $h+1$ の場所へ移動するのにかかる時間とする．各顧客間の時間指定の

制約条件を「各顧客のサービス開始時間に、その顧客の位置から次の顧客の位置までの移動時間を足したものが、次の顧客のサービス開始時間より前である」と表す。

仮定 4 (容量制約) 各顧客 $i \in V \setminus \{0\}$ に届ける荷物量を a_i とし、各車両 $k \in M$ の積める荷物量は u_k とする。ここで、 a_i, u_k は正定数である。各車両 $k \in M$ に積む荷物量は、車両に積める荷物量 u_k を越さないとする。

車両 k の移動距離 $d_{\text{sum}}^k(\sigma_k)$ は以下のように表せる。

$$d_{\text{sum}}^k(\sigma_k) = \sum_{h=0}^{n_k} d_{\sigma_k(h), \sigma_k(h+1)}.$$

ここで、 $d_{i,j}$ は顧客 i と顧客 j の距離、すなわち

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

とする。ここで、 $(x_i, y_i), (x_j, y_j)$ はそれぞれ、顧客 i, j の位置を表す 2 次元座標である。また、 $z_{i,k} \in \{0, 1\}$ を、車両 k が顧客 i を訪問する場合 $z_{i,k} = 1$ 、訪問しない場合 $z_{i,k} = 0$ となるバイナリ変数とする。以上の条件のもとで、時間制約と容量制約を持つ配送計画問題は以下のように定式化される：

$$\text{最小化} \quad \sum_{k=1}^m d_{\text{sum}}^k(\sigma_k) \quad (2)$$

制約条件

$$\sum_{k \in M} z_{i,k} = 1 \quad (i \in V \setminus \{0\}), \quad (3)$$

$$\sum_{h=1}^{n_k} a_{\sigma_k(h)} \leq u_k \quad (k \in M), \quad (4)$$

$$s_0 = 0, \quad s_i \in [e_i, l_i] \quad (i \in V \setminus \{0\}), \quad \bar{s}_k \in [e_0, l_0] \quad (k \in M), \quad (5)$$

$$s_{\sigma_k(h)} + t_{\sigma_k(h), \sigma_k(h+1)} \leq s_{\sigma_k(h+1)}, \quad (k \in M, h = 0, 1, \dots, n_k - 1), \quad (6)$$

$$s_{\sigma_k(n_k)} + t_{\sigma_k(n_k), \sigma_k(n_k+1)} \leq \bar{s}_k, \quad (k \in M). \quad (7)$$

(2) は各車両の走行距離の合計を最小化する目的関数を表す。(2) は走行距離のみに着目していることに注意する。(3) から (7) は制約条件を表す。(3) は仮定 2 を、(4) は仮定 4 を、(5)–(7) は仮定 3 を表している。この他にも、状況に合わせて制約条件が追加されることもあるが、(2)–(7) を時間枠付き配送計画問題とよぶ。次節では、構築型解法と一般化割当法、罰金法により時間枠付き配送計画問題の近似解を求める手法を紹介する。

2.2 時間枠付き配送計画問題に対する発見的解法

本節では時間枠付き配送計画問題に対する近似解法である構築型解法と Fisher-Jaikumar の一般化割当法、罰金法について概説する。

2.2.1 構築型解法

構築型解法とは実行可能解を生成し、そこから徐々に巡回路を拡大し、解を構成する手法の総称である。ここでは、挿入法 [20] や Clark-Wright セービング法 [2] に基づく構築型解法の手順を紹介する。

1. 何かしらの基準で訪問する顧客 i を選ぶ。例えば、デポから一番遠い顧客を選ぶ。
2. 顧客 i とのセービング値 $S_{i,j}$ が一番小さくなる顧客 j を選ぶ。ここで、セービング値 $S_{i,j}$ は

$$S_{i,j} = (d_{i,0} + d_{j,0}) \times 2 - (d_{0,i} + d_{i,j} + d_{j,0}) = d_{i,0} + d_{j,0} - d_{i,j}.$$

である。ここで、(1) で定義される距離 $d_{i,j}$ は、 $d_{i,j} = d_{j,i}$ が成り立つことに注意する。セービング値が正であれば、デポから顧客 i と顧客 j をそれぞれ往復するより、デポから顧客 i へ行き、顧客 i から顧客 j へ行き、顧客 j からデポへ戻る、もしくはデポから顧客 j へ行き、顧客 j から顧客 i へ行き、顧客 i からデポへ戻る方が走行距離が短くなることを意味している。また、値が大きいほど短縮される距離が長いことを意味している。 $S_{i,j}$ が最大となる顧客 j を選び、ルートをデポから顧客 i (もしくは顧客 j)、顧客 i から顧客 j (もしくは顧客 j から顧客 j)、顧客 j (もしくは顧客 i) からデポとする。ただし、新たに作成したルートが時間制約や容量制約を満たしていることは別途確認する。

3. 顧客 p と顧客 $p+1$ が既にこのルートに入っているとし、まだルートに入れていない顧客 i を顧客 p と顧客 $p+1$ の間に入れてルートを作成する方がいいかを考察する。セービング値 $S_i^{(p)}$ を以下で定義する。

$$\begin{aligned} S_i^{(p)} &= d_{0,p} + d_{p,p+1} + d_{p+1,0} + 2d_{i,0} - (d_{0,p} + d_{p,i} + d_{i,p+1} + d_{p+1,0}) \\ &= d_{p,p+1} + 2d_{i,0} - (d_{p,i} + d_{i,p+1}). \end{aligned}$$

$S_i^{(p)}$ が正の値を取ると、デポから顧客 i に個別で訪問するより、顧客 p と顧客 $p+1$ の間に顧客 i を回った方が走行距離が短くなり、値が大きくなればなるほど、短縮される距離が短くなることを意味している。また、 $p=0$ もしくは $p+1=n_{k+1}$ である場合は、顧客をデポとみなす。 $S_i^{(p)}$ が一番大きくなる p と i を選び、もとのルートの顧客 p と顧客 $p+1$ の間に顧客 i を挿入する。ただし、時間制約や容量制約を満たしていることは別途確認する。

4. 時間制約や容量制約を満たさない顧客以外がルートに入るまで、3. を繰り返す。
5. 時間制約や容量制約を満たさなかった顧客で新たにグループを作成し、1. から 4. までの手順で配送ルートを求める。
6. 1. から 5. までを繰り返し、全ての顧客 $i \in V \setminus \{0\}$ がいずれかの車両 $k \in M$ の配送ルートに含まれるまで繰り返す。ただし、時間制約や容量制約によっては、どの車両 $k \in M$

の配送ルートにも含めることができない場合もあり, そのときは後で紹介する罰金法などを利用して対応する.

2.2.2 Fisher-Jaikumar 一般化割当法

2.2.1 節で述べた近似解法では, 各車両の走行距離の合計のみに着目し, 車両ごとに宅配件数を分割して配送ルートを作成することはできない. Fisher-Jaikumar 法ではまず, 最初に顧客を指定台数 m 個に分割し, その後に配送ルートを求める. 以下に手順を述べる.

1. 各車両 k ごとに, 何かしらの基準を持って顧客 r_k を選ぶ. 例えばデポから遠い客を, 車両台数 m 個分選び, 各車両のグループに割り当てる.
2. 顧客 i を車両 k のルートに加えるコスト $c_{i,k}$ を計算する. $c_{i,k}$ は以下で与えられる.

$$c_{i,k} = d_{0,i} + d_{i,r_k} + d_{r_k,0} - (d_{0,r_k} + d_{r_k,0}) = d_{0,i} + d_{i,r_k} - d_{0,r_k}.$$

$c_{i,k}$ の値は, デポから顧客 i (もしくは顧客 r_k) まで行き, 顧客 i (もしくは顧客 r_k) から顧客 r_k (もしくは顧客 i) まで行き, 顧客 r_k からデポまで行く場合の距離からデポと顧客 r_k の往復の距離を引いたものである. この意味で, $c_{i,k}$ は顧客 i を車両 k のルートに加えるコストである.

3. 次の最小化問題を解くことで, 容量制約を満たしつつ, 各車両へ全ての客を割り当てる.

$$\text{最小化} \quad \sum_{k \in M} \sum_{i \in V \setminus \{0\}} c_{i,k} z_{i,k} \quad (8)$$

制約条件

$$\sum_{k \in M} z_{i,k} = 1 \quad (i \in V \setminus \{0\}), \quad (9)$$

$$\sum_{i \in V \setminus \{0\}} a_i z_{i,k} \leq u_k, \quad (k \in M). \quad (10)$$

2.2.3 罰金法

2.2.1 節, 手順 6. で述べた通り, 全ての時間指定を満たす配送ルートを探索できなかった場合, 指定した車両台数で配送ルートを作成することはできない. そこで, 時間制約を制約条件として課すのではなく, 時間制約の違反度を表す罰金関数を目的関数に入れてルートを作成する. 容量制約も同様に目的関数に入れることは可能であるが, ここでは時間制約のみに着目し, 容量制約は制約条件として課す. 手順を以下に述べる.

1. 2.2.2 節で述べた手法などを用いて車両台数 m 個分のグループを作成する.
2. 各グループ k に対し, 何かしらの基準で顧客 r_k を選ぶ. 例えばデポから遠い顧客を, 車両台数 m 個分選び, 各車両のグループに割り当てる.
3. 2.2.1 節, 手順 2. で述べたセービング値 S_{i_k, j_k} に次で定義する罰金関数値 $p_{1, j_k}(s_{j_k})$ ま

たは $p_{2,j_k}(s_{j_k})$ を減じた値 c_{n,j_k}^* ($n = 1, 2$), すなわち

$$c_{n,j_k}^* = S_{i_k,j_k} - p_{n,j_k}(s_{j_k}) \quad (n = 1, 2)$$

が大きくなるように顧客 j_k を選ぶ. ここで, s_{j_k} は顧客 j_k のサービス開始時刻である. s_{j_k} は, 仮定 3 に述べたように, その前の顧客のサービス終了時刻やその顧客の位置から顧客 j の位置までの移動時間を考慮して求める. 定数 M を十分大きくとり, 罰金関数 $p_{n,j}(t)$ ($n = 1, 2$) を以下の形で定義する.

$$p_{1,j}(t) = \begin{cases} 0 & (t \in [e_j, l_j]) \\ M & (t \notin [e_j, l_j]) \end{cases}, \quad p_{2,j}(t) = \begin{cases} 0 & (t \in [e_j, l_j]) \\ M|t - l_j| & (t \notin [e_j, l_j]) \end{cases}.$$

ただし, 容量制約を満たしていることは別で確認する. 罰金法は制約条件を満たさないことがあるため, 複数の罰金関数を用意して経路を複数求め, その中から一番よい経路を求めるのが望ましい. ここでは二つの罰金関数を考える.

4. 各グループ k に対し, 顧客 p と顧客 $p+1$ が既にこのルートに入っていると, グループ k に属しているが, まだルートに入れていない顧客 i_k を顧客 p と顧客 $p+1$ の間にに入れてルートを作成する方がいいかどうかを考察する. 2.2.1 節 手順 3. で述べたセービング値 $S_{i_k}^{(p)}$ に手順 3. で述べた罰金関数値 $p_{1,j_k}(s_{j_k})$ または $p_{2,j_k}(s_{j_k})$ を足した値 c'_{n,j_k} ($n = 1, 2$), すなわち

$$c'_{n,j_k} = S_{i_k}^{(p)} - p_{n,j_k}(s_{j_k}) \quad (n = 1, 2)$$

が大きくなるように, 顧客 i_k と場所 p を選ぶ. ただし, 容量制約を満たしていることは別で確認する.

5. 各グループ k の顧客が全てルートに入りきるまで 4. を繰り返す.
6. 全てのグループ k に対し, 1. から 4. の手順を行う.

2.3 量子アニーリング

量子アニーリングは次の形で表現される無制約 2 次計画問題とよばれる問題を解く技術の一つである.

$$\text{最小化} \quad E(\mathbf{q}) = \mathbf{q}^\top \mathbf{Q} \mathbf{q} \quad (11)$$

ここで, \mathbf{q} はバイナリ変数のベクトルで, \mathbf{Q} は最適化問題の特徴付ける行列である.

最適化問題を解くための手法として焼きなまし (アニーリング) という手法がある. (金属の) 焼きなましとは結晶欠陥の少ない綺麗な金属材料を得るために, 金属材料を高温に熱した後徐冷する作業のことを指す. この焼きなましをコンピュータ上で擬似的に再現したシミュレーテッドアニーリングから着想を得て, 量子アニーリングが考案された. 初期状態として全ての解を等確率で同時に探索する重ね合わせ状態を用意した後, パラメータを制御して徐々に

エネルギーの低い解 ((11) の解) に確率を収束させていく. 量子アニーリングは明確に異なる 2 状態を同時に重ね合わせて実現できる量子力学の原理にしたがって設計されている.

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (12)$$

の 2 状態の規格化された線形結合

$$\frac{1}{\sqrt{|a|^2 + |b|^2}} \left(a \begin{pmatrix} 1 \\ 0 \end{pmatrix} + b \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right)$$

は (12) が同時に存在する量子力学的重ね合わせを表現していると解釈できる. すなわち, 二つのベクトルを一つのベクトルで表現でき, 計算回数を節約していると解釈することができる. 最適な組み合わせを探す試行の回数を, 量子重ね合わせの原理で増やせるので, 最適化問題の中でも特に, 組み合わせ最適化問題に適しているといわれている. 既に述べたとおり, 量子アニーリングの理論をハードウェアとして実装する装置である量子アニーリングマシンが既に開発され, 市販に至っている.

量子アニーリングは, 理論的には効率的に最適解を探索することが知られているが, [11, 12, 21], 現時点のシステムでは理論通りに動くのは難しく, 量子アニーリングのパフォーマンスは明らかにされていない [15, 17].

D-Wave Systems の提供する量子アニーリングマシンの量子ビット数は増加し続けており, 将来的には大規模な問題を効率的に解くことが期待されている. また, 量子アニーリングマシンは最適解を探索するのに絶対零度で動作することが好ましいが, 現在よりずっと大きいシステムになっても, 極低温に冷却する部分はごく小さいままで, 消費電力はほとんど増えないことが現在のコンピュータとの大きな違いで, 省エネルギー性能も期待されている.

また, 現在の量子アニーリングマシンでは, デバイスの制限と環境の影響により, 必ずしも最適解を出力するとは限らないが [1], その性質を利用して, 出力される解全てを使って機械学習などに応用する研究もなされている [14].

このように, 量子アニーリングによる最適化問題の解の探索は注目されており, 配送計画問題にも応用されているが, 著者らが知る限り, 容量制約付き配送計画問題には応用されているものの, 時間枠付き配送計画問題には, まだ応用されていない. 以下では, 時間枠付き配送計画問題の解の探索の一部に, 量子アニーリングを取り入れる手法を提案し, その結果について述べる.

3 提案手法

デポからいくつかのトラックを用いて客へ配送物を届ける. ここでは, 各トラックに宅配件数を均等に振り分ける制約を入れてグループ分けをする最適化問題を定式化し, それを時間枠付き配送計画問題の発見的解法に活用する.

配送指定時間の組み合わせによっては, 全員の指定時間を守って配送ルートを作成することが困難なことがある. そこで, 罰金法を用いて時間制約の違反が少ないルートの提案を行う.

また、発見的解法で探せる解は局所最適解であるため、初期値の取り方が重要である。既存の手法では、デポから一番遠い客を初期値に取ることが多いが、全ての顧客を初期値としてルートを作成することで、よりよい配送ルートが探せる例についても述べる。グループに属する顧客数が多いと、計算量が多くなり、全ての顧客を初期値としてルートを作成することは困難であるが、今回の数値例で用いる顧客数ではこの試行が可能である。

最後に、今回提案するグループ分けの最適化問題を量子アニーリングで解き、その結果を時間枠付き配送計画問題の発見的解法に活用した結果について述べる。なお、3.1 節から 3.5 節では [6] を参考に近似解の作成を行なっている。

3.1 宅配件数を均等に割り振るグループ分けをする最適化問題

本節では、宅配件数を均等に割り振る一般化割当法について述べる。定式化は以下の通りである。各車両 k が配送する客を決める。2.2.2 節で述べた最小化問題 (8)–(10) に宅配件数を均等に振り分ける制約式 (16) を追加することで、宅配件数を均等に割り振るグループ分けをする最適化問題を次の通りに定式化する。

$$\text{最小化} \quad \sum_{k \in M} \sum_{i \in V \setminus \{0\}} c_{i,k} z_{i,k} \quad (13)$$

制約条件

$$\sum_{k \in M} z_{i,k} = 1 \quad (i \in V \setminus \{0\}), \quad (14)$$

$$\sum_{i \in V \setminus \{0\}} a_i z_{i,k} \leq u_k \quad (k \in M), \quad (15)$$

$$\sum_{i \in V \setminus \{0\}} x_{i,k} \geq C \quad (k \in M). \quad (16)$$

ここで、定数 C は $\frac{\text{客数}}{\text{車両数}}$ の小数点以下を切り下げた整数である。

ある地図上に配置された 50 件の顧客に対して一般化割当法を適用した結果が以下の通りである。(16) の制約条件を課した方が、目的関数値が大きくなるが、顧客数は均等に割り振れていることが分かる。2.2.2 節に述べた最小化問題 (8)–(10) に宅配件数を均等に割り振る制約式 (16) を追加する。

表 1 顧客数を均等に割り振っていない場合の分割 (目的関数値:420.2)

車両番号	顧客数	顧客番号
1	8	7, 10, 11, 19, 27, 31, 32, 49
2	18	2, 5, 6, 13, 14, 15, 16, 17, 21, 22, 23, 37, 38, 40, 41, 42, 43, 44
3	7	8, 18, 36, 45, 46, 47, 48
4	17	1, 3, 4, 9, 12, 20, 24, 25, 26, 28, 29, 30, 33, 34, 35, 39, 50

表 2 顧客数を均等に割り振った場合の分割 (目的関数値:504.9)

車両番号	顧客数	顧客番号
1	12	1, 7, 10, 11, 19, 26, 27, 28, 30, 31, 32, 49
2	13	2, 14, 15, 16, 21, 22, 23, 37, 38, 41, 42, 43, 44
3	12	5, 6, 8, 13, 17, 18, 36, 40, 45, 46, 47, 48
4	13	3, 4, 9, 12, 20, 24, 25, 29, 33, 34, 35, 39, 50

図 1 の手順にしたがって配送ルートを求めた結果が 図 2, 図 3 である. 2.2.1 節 手順 1. の初期値は各グループの顧客番号で一番小さいものとする. 車両 4 台でルートを作成しているが, 実際には 5 台の車両でルートを作成している. 5 台目のルートは紫色に表示されているものである. これは, 割り振られたグループに対し, 2.2.1 節で述べた手法を使ってルートを作成した際に, 時間制約を満たさないものが出てしまうためである.

しかし, 現実には, 用意された車両台数のみで運用する. この問題に対する解決策として, 一般化割当問題を解く際に, 時間制約を考慮してグループ分けをすることが考えられるが, 現時点で具体的な解決策は知られていない. そこで次節では, 罰金法を用いて指定台数を守りながら, 時間制約を考慮する配送ルートについて考察する.

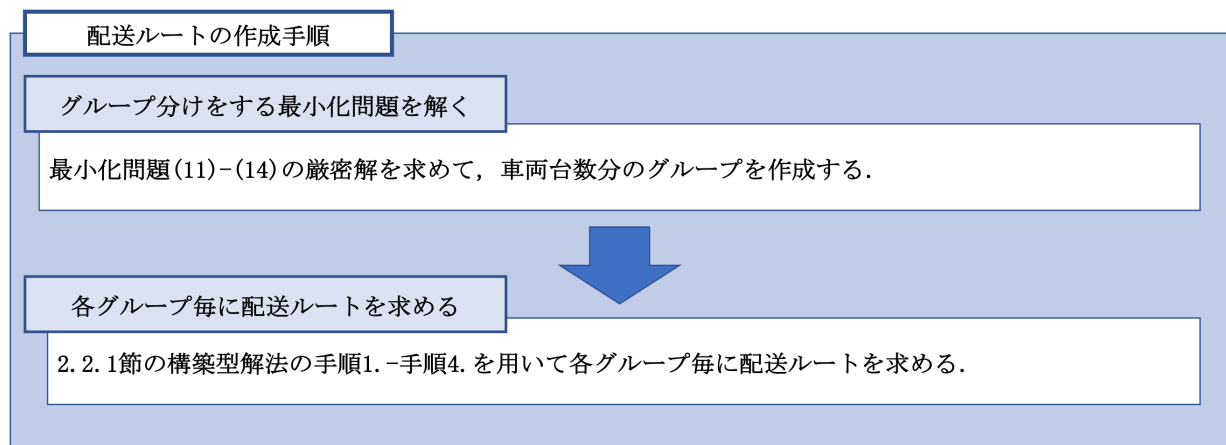


図 1 配送ルート作成手順

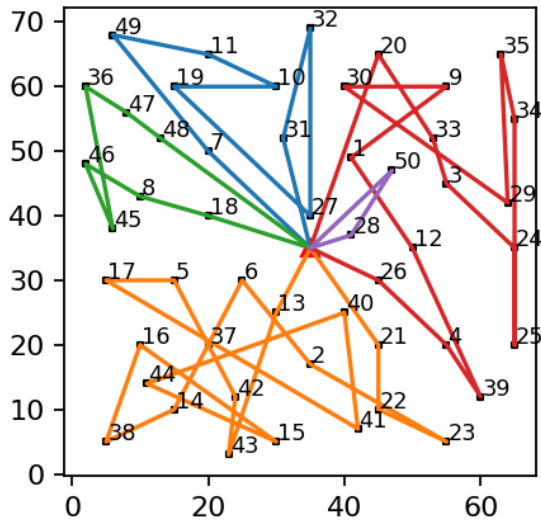


図 2 顧客数を均等に割り振っていない分割を用いた配送ルート

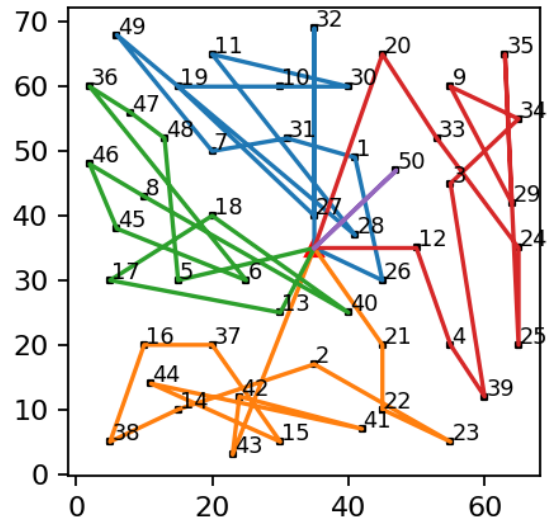


図 3 顧客数を均等に割り振った分割を用いた配送ルート

3.2 罰金法の適用

3.2 節で述べた罰金法を配送ルート作成に適用する．配送ルートの作成手順は図 4 の通りである．数値実験結果は図 5, 6 のとおりである．グループ分けは表 2 と同じものを用いている．2.2.1 節 手順 1. の初期値は各グループの顧客番号で一番小さいものとする．また，罰金関数に用いる定数 M は $M = 10000$ としている．この結果，罰金関数 $p_{1,j}(t)$ を用いた場合， $p_{2,j}(t)$ を用いた場合の両方で全く同じルートが作成され，指定した 4 台の車両のみで配送ルートを作成することができた．また，車両 4 のみ，39 番の顧客の指定した時間を守ることができなかった．

表 3 時間指定を守れなかった客数

車両番号	時間制約を守れなかった客数	時間制約を守れなかった客
1	0	—
2	0	—
3	0	—
4	1	39

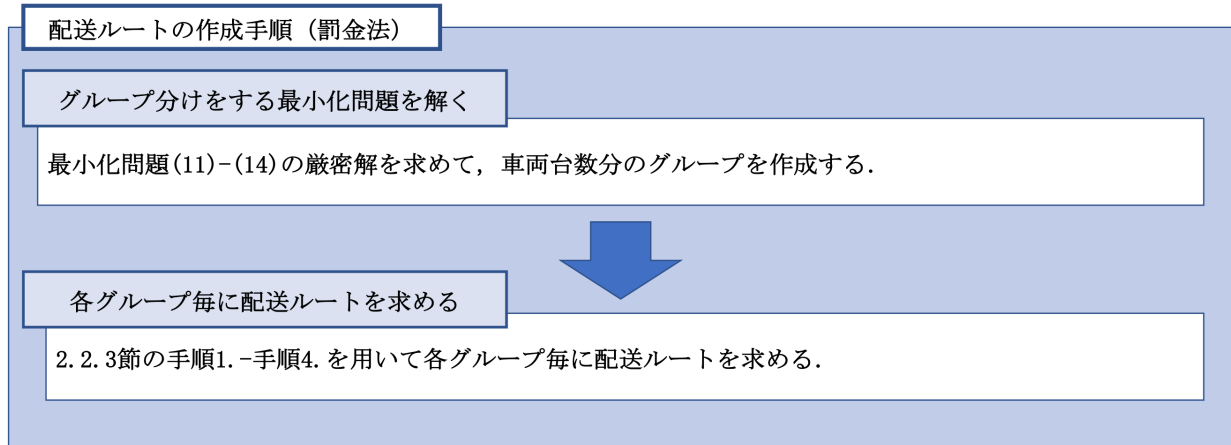


図 4 配送ルート作成手順 (罰金法)

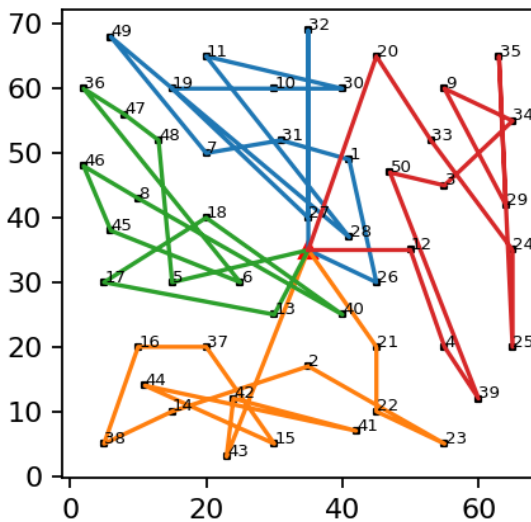


図 5 罰金関数 $p_{1,j}(t)$ を用いて作成した配送ルート

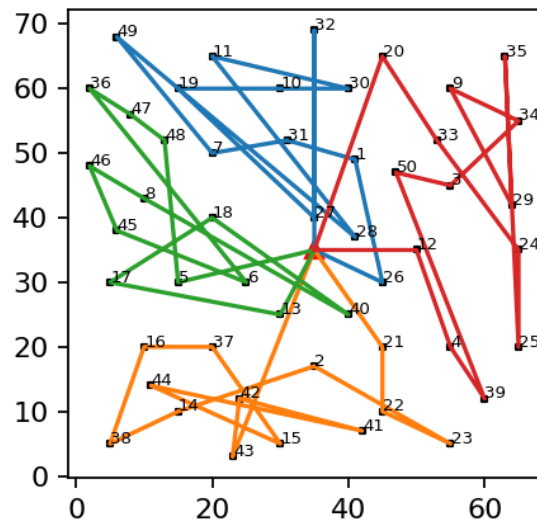


図 6 罰金関数 $p_{2,j}(t)$ を用いて作成した配送ルート

3.3 初期値の取り方の工夫

構築型解法の手順 1. (2.2.1 節 手順 1.) で、配送ルートを構築する際に、初期値としてどの顧客を選ぶべきか、適切な方法は知られていない。既存手法では、グループの中で一番デポから遠い顧客を選ぶことが多いが、必ずしもそれが最善でないことを本節で数値的に示す。

グループ内の客全てを初期値として選んで、配送ルートを作成し、一番目的関数値が小さくなるルートを採用する。グループ分けは 3.1 節と同じものを、ルートの作成法は 3.2 節と同じものを採用する。配送作成手順は図 7 の通りである。この結果、罰金関数 $p_{1,j}(t)$ を用いた場合、 $p_{2,j}(t)$ を用いた場合の両方で、初期値を最適化した場合と一番遠い客を初期値とした場合で違う配送ルートが作成された。なお、図 8 と図 10、図 9 と図 11 は全く同じものとなった。

図 8 と図 10 で目的関数値は 1014, 図 9 と図 11 で目的関数値は 1033 である. また, どの場合も時間指定を違反するものはなかった.

このことから, 初期値の取り方の工夫に改善の余地があることが分かった.

表 4 最適な初期値

車両番号	目的関数が一番小さくなる顧客	デポから一番遠い顧客
1	28	49
2	14	38
3	6	36
4	9	35

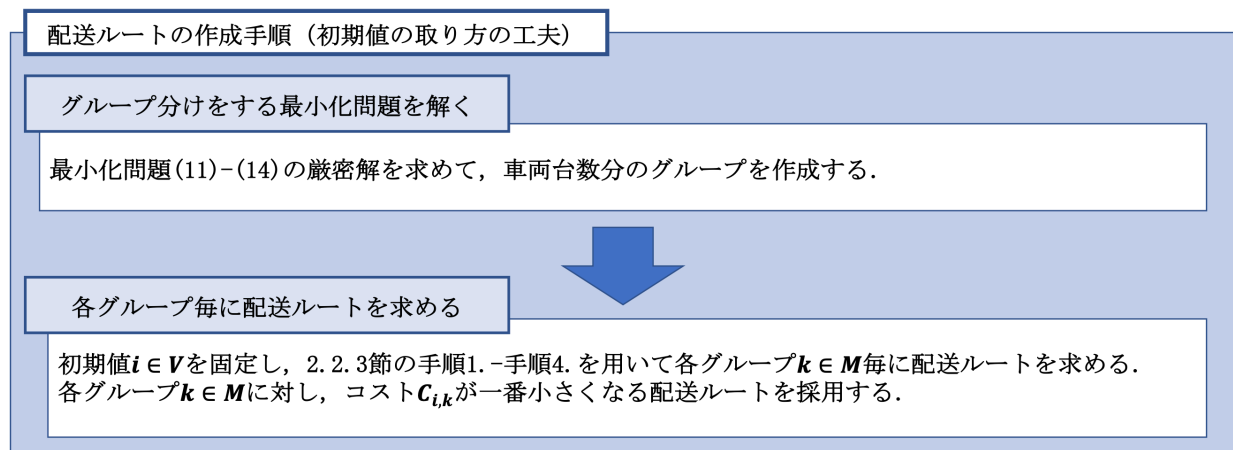


図 7 配送ルート作成手順 (初期値の取り方の工夫)

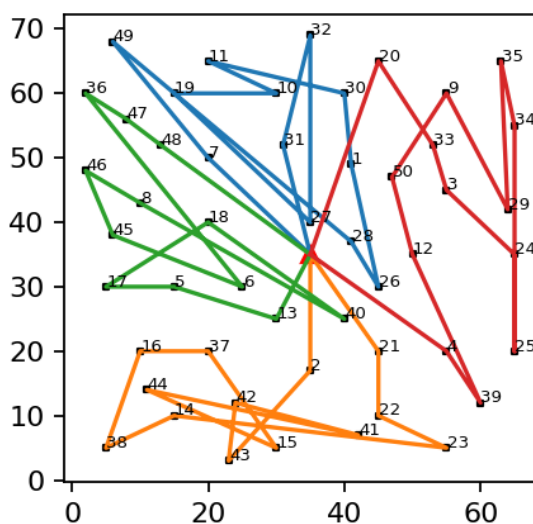


図 8 罰金関数 $p_{1,j}(t)$ を用い, 最適な初期値を取って作成したルート

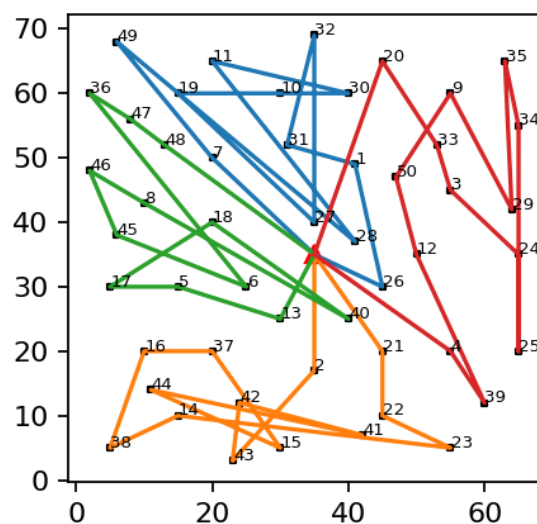


図 9 罰金関数 $p_{1,j}(t)$ を用い, 一番遠い顧客を初期値として作成したルート

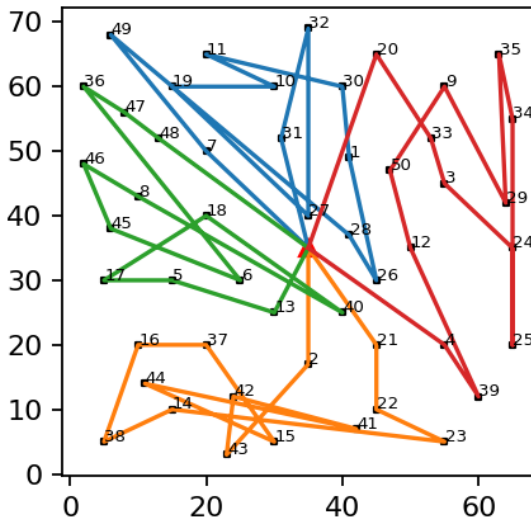


図 10 罰金関数 $p_{2,j}(t)$ を用い、最適な初期値を取って作成したルート

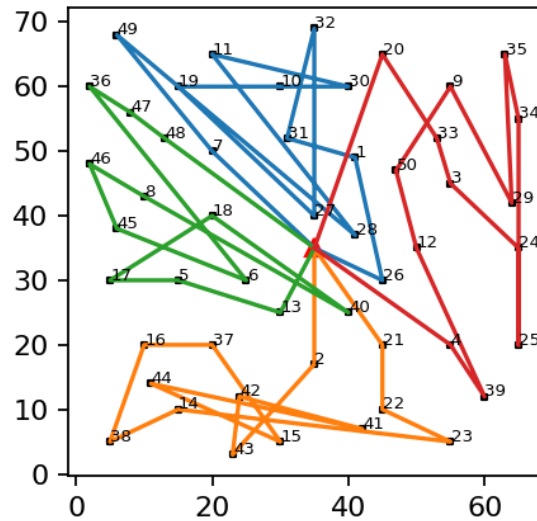


図 11 罰金関数 $p_{2,j}(t)$ を用い、一番遠い顧客を初期値として作成したルート

3.4 実際の地図での配送計画

本節では、3.3 節の手法を実際の地理空間データに適用した結果について述べる。3.1–3.3 節では客間の距離は直線距離 (1) で計算を行っているが、本節では地理空間データをグラフとみなし、ダイクストラ法で最短経路を求め、その最短経路の長さを客と客の間の距離とみなしている。

東京都江東区の地図データを取得してグラフ化する。デポとして 1 個のノードを、顧客として 20 個のノードを取得する。配送車両は 2 台あるとして、配送先の 20 個のノードを二つに分割し、3.3 節で用いた手法を用いて、車両二つ分の配送経路を作成した。地図データの取得や最短経路の求め方、地図情報を可視化するプログラムは [13] を参考に作成した。配送ルート作成手順は図 12 の通りである。図 14 は車両 1 の配送経路、図 15 は車両 2 の配送経路を表している。このことから、最短経路が存在すれば、配送経路を実際の地図データで作成できることが分かった。しかし、時間制約は 3.3 節の数値実験と同じものを用いているがデポへの帰還時間を含め、全ての顧客の時間指定に違反しており、よりよい配送経路の作成法や車両数の増加が必要である。

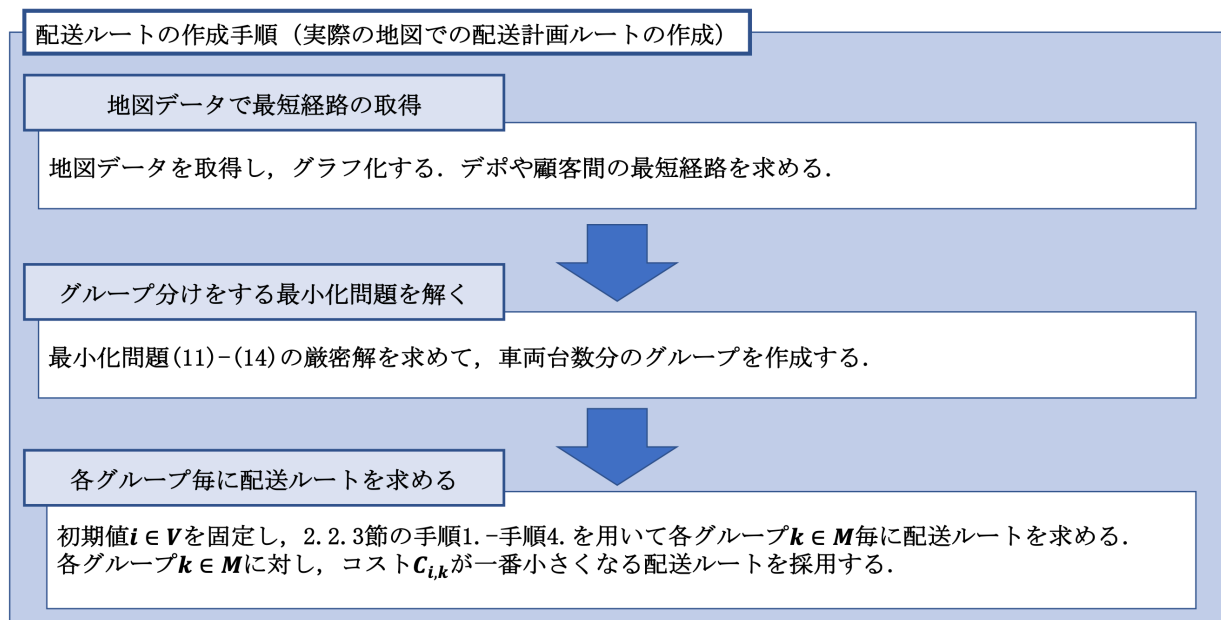


図 12 配送ルート作成手順（実際の地図での配送計画ルートの作成）



図 13 デポと顧客のノード



図 14 車両 1 の配送ルート



図 15 車両 2 の配送ルート

3.5 量子アニーリングによる試み

本節では 3.1 節で述べた最適化問題 (13)–(16) を量子アニーリングで解き，その結果を 3.4 で用いた手法に適用し，得た配送経路について述べる．

2.3 節で述べた通り，量子アニーリングはバイナリ変数の最適化問題の解を探索する手法であるため，連続変数を含む時間枠付き配送計画問題に適しているとは言い難い．そこで，整数変数だけで表現できる (13)–(16) の部分のみを量子アニーリングで解き，その結果を時間枠付き配送計画問題に利用する．整数変数は後述する Log encoding とよばれる方法でバイナリ変数に書き直すことができる．

本節で提案する手法の結果が，前節までの数値結果と比べて優れた結果になるわけではないが，将来的に，量子アニーリングマシンが古典的なコンピュータで解くことができないような

大規模な問題を解くことが可能となったとき, 時間枠付き配送計画問題を本稿の手法で効率的に解けるようになることが期待される.

量子アニーリングで最適化問題の解を探索するためには, (13)–(16) を (11) の形に書き直す必要がある. そのためにまず, (13)–(16) における不等式制約を等式制約に書き直す. $c_{i,k}, a_i, u_k \in \mathbb{N}$ ($i \in V \setminus \{0\}, k \in M$) は定数とし, $z_{i,k}$ ($i \in V \setminus \{0\}, k \in M$) はバイナリ変数で, $s_k, t_k \geq 0$ ($k \in M$) は整数値のスラック変数とする. スラック変数については, 後述する注意 1 を参照せよ.

$$\text{最小化} \quad \sum_{k \in M} \sum_{i \in V \setminus \{0\}} c_{i,k} z_{i,k} \quad (17)$$

制約条件

$$\sum_{k \in M} z_{i,k} = 1 \quad (i \in V \setminus \{0\}), \quad (18)$$

$$\sum_{i \in V \setminus \{0\}} a_i z_{i,k} + s_k = u_k, \quad (k \in M) \quad (19)$$

$$\sum_{i \in V \setminus \{0\}} z_{i,k} - C = t_k \quad (k \in M) \quad (20)$$

ここで $C = \frac{\text{客数}}{\text{車両数}}$ の小数点以下を切り下げた整数である. (17)–(20) は以下のように書き直すことができる.

$$\begin{aligned} \text{最小化} \quad & \sum_{k \in M} \sum_{i \in V \setminus \{0\}} c_{i,k} z_{i,k} + \lambda_1 \sum_{i \in V \setminus \{0\}} \left(\sum_{k \in M} z_{i,k} - 1 \right)^2 \\ & + \lambda_2 \sum_{k \in M} \left(\sum_{i \in V \setminus \{0\}} a_i z_{i,k} + s_k - u_k \right)^2 + \lambda_3 \sum_{k \in M} \left(\sum_{i \in V \setminus \{0\}} z_{i,k} - C - t_k \right)^2 \end{aligned} \quad (21)$$

また, (11) で表すためには, バイナリ変数のみで (17)–(20) を表現する必要がある. Log encoding とよばれる方法で, 2 進数のように $0 \leq y \leq M - 1$ を満たす整数を表現することができる [19].

$$y = \sum_{i=0}^{\lfloor \log_2(M-1) \rfloor} 2^i y_i, \quad y_i \in \{0, 1\}, \quad (i \in \{0, 1, 2, \dots, \lfloor \log_2(M-1) \rfloor\}).$$

ここで, $x \in \mathbb{R}$ に対して, $\lfloor x \rfloor$ は x を超えない最大整数である. 例えば,

$$10 = 1 \times y_0 + 2^1 \times y_1 + 2^2 \times y_2 + 2^3 \times y_3$$

と表すことができる. このとき,

$$y_0 = 0, \quad y_1 = 1, \quad y_2 = 0, \quad y_3 = 1$$

である.

この書き直しにより，対象とする問題を (11) の形で書き表すことができた．配送ルート作成手順は図 16 の通りである．

また，表 5 は (21) を量子アニーリングによって得た結果の 1 例である．100 回分アニーリングを試行し，その中で目的関数値が一番小さかった値を採用した結果である．この結果から分かる通り，宅配件数に多少ばらつきが出るが，概ね均等に割り振れていることが分かった．完全には均等に割り振れていないため，定式化の更なる工夫が必要であることも分かる．

図 18，図 19 は表 5 によりグループ分けしたグループに対して作成した配送ルートである．デポや顧客の位置は図 13 と同じであるが，グループ分けが異なるため，作成された配送ルートは図 14，図 15 と異なることが分かる．

注意 1 スラック変数とは不等式制約を等式制約に書き直すための変数である．例えば，

$$a + b \geq 1, \quad a \geq 0, \quad b \geq 0$$

はスラック変数 c を追加して，

$$a + b - 1 = c, \quad a \geq 0, \quad b \geq 0, \quad c \geq 0$$

と書き直すことができる．

表 5 量子アニーリングによるグループ分け

車両番号	顧客数	顧客番号
1	9	1, 4, 5, 11, 12, 13, 15, 17, 20
2	11	2, 3, 6, 7, 8, 9, 10, 14, 16, 18, 19

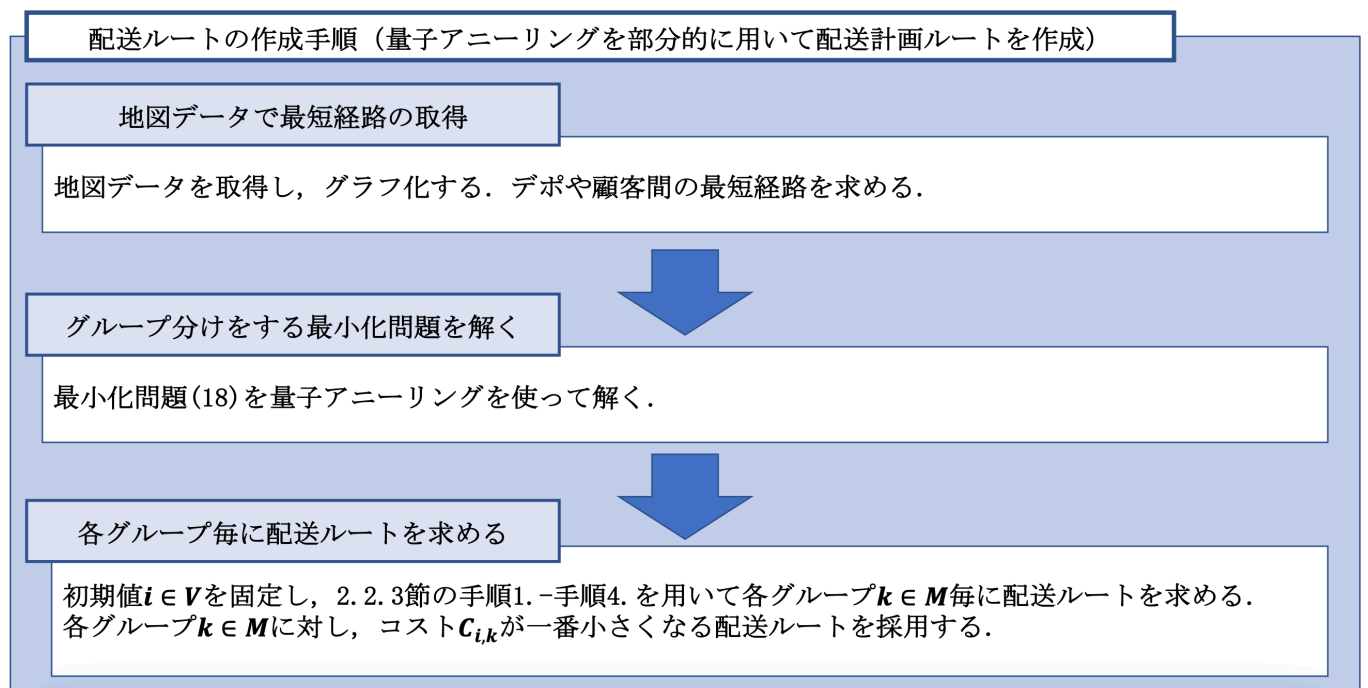


図 16 配送ルート作成手順（量子アニーリングを部分的に用いて配送ルートを作成）



図 17 デポと顧客のノード

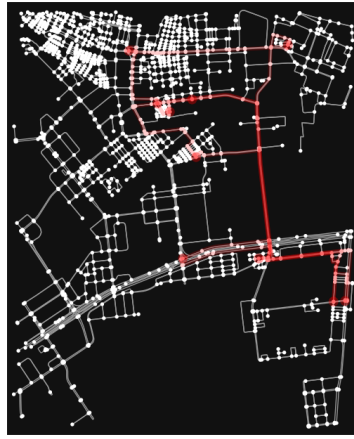


図 18 車両 1 の配送ルート



図 19 車両 2 の配送ルート

4 まとめ

本稿では、各トラックに宅配件数を均等に振り分ける制約を入れてグループ分けを最適化問題を定式化し、それを時間枠付き配送計画問題の発見的解法に活用した。また、罰金法や初期値の取り方を工夫し、指定台数や時間指定を守る解の探索を試みた。この試行を通して、時間制約や容量制約を守りながら宅配件数を均等に割り振る配送ルートを作成することができた。

さらに、量子アニーリングを用いても宅配件数の均等な割り振りを考慮して時間枠付き配送計画問題の発見的解法で配送ルートを作成することができた。よりよい問題設定、定式化、古典コンピュータとのハイブリッドを検討することで、将来的に量子アニーリングを利用してより大規模な配送計画問題を省エネルギーに実装できる可能性を見出すことができた。

ただし、課題も多く残る。発見的解法は局所解のみを探す手法であり、大域解である保証はない。また、現実の問題はより複雑である。大域解の探索手法やより現実の問題に近づけた最適化問題の定式化、近似解法の提案が求められる。

謝辞

本研究の一部は武蔵野大学研究費、武蔵野大学しあわせ研究費の助成を受けたものです。

参考文献

- [1] M. H. Amin, Searching for quantum speedup in quasistatic quantum annealers, *Phys. Rev. A* Vol.92, (2015) 052323.
- [2] G. Clark and J. W. Wright, Scheduling of vehicles from a central depot to a number of delivery points, *Operation Research*, Vol.12, No.4, (1964), pp.568–581.
- [3] F. Barahona, On the computational complexity of Ising spin glass models, *Physica*

A Vol. 388,(2009),pp.2821–2838.

- [4] Z. Bian, F. Chudak, R. B. Israel, B. Lackey, W. G. Macready and A. Roy, Mapping Constrained Optimization Problems to Quantum Annealing with Application to Fault Diagnosis, *Frontiers in ICT* Vol.3, No.15 (2016).
- [5] E. Boyda, S. Basu, S. Ganguly, A. Michaelis, S. Mukhopadhyay, and R. R. Nemani, Deploying a quantum annealing processor to detect tree cover in aerial imagery of California, *PLOS ONE* Vol.12, No.2, e0172505 (2017).
- [6] Dung Tran, Modelling and Analysis of a Vehicle Routing Problem with Time Windows in Freight Delivery, <https://github.com/dungtran209/Modelling-and-Analysis-of-a-Vehicle-Routing-Problem-with-Time-Windows-in-Freight-Delivery>
- [7] 橋本英樹, 胡艶楠, 配送計画問題に対する発見的解法, システム/制御/情報, Vol.64, No.6, (2020), pp.218–223.
- [8] K. Kitai, J. Guo, S. Ju, S. Tanaka, K. Tsuda, J. Shiomi, and R. Tamura, Expanding the horizon of automated metamaterials discovery via quantum annealing, *arXiv:1902.06573* (2019).
- [9] 国土交通省, 令和 2 年度 宅配便取扱実績について, 2021 年
- [10] 国土交通省, 令和 3 年度 宅配便取扱実績について, 2022 年.
- [11] S. Morita and H. Nishimori, Mathematical foundation of quantum annealing, *J. Math. Phys.* Vol.49, 125210 (2008).
- [12] M. Ohzeki and H. Nishimori, Quantum annealing: an introduction and new developments, *J. Comput. Theor. Nanosci.* Vol. 8, (2011), 963–971.
- [13] 大関 真之, Quantum Annealing for You, <https://altema.is.tohoku.ac.jp/QA4U/>
- [14] M. Ohzeki, S.Okada, M. Terabe and S. Taguchi, Optimization of neural networks via finite-value quantum fluctuations., *Sci. Rep.*, Vol. 8 (2018) 9950.
- [15] M. Ohzeki, A. Miki, M. J. Miyama, and M. Terabe, Control of automated guided vehicles without collision by quantum annealer and digital devices, *Front. Comput. Sci.*, Vol.1, No.9, (2019).
- [16] F. Neukart, D. V. Dollen, G. Compostella, C. Seidel, D. V. Dollen, S. Yarkoni, and B. Parney, Traffic Flow Optimization Using a Quantum Annealer, *Frontiers in ICT* Vol.4, No.29 (2017).
- [17] 西森 秀稔, 大関 真之, 量子アニーリングの基礎, 共立出版, (2018).
- [18] 岡林 楠博, 中村 有克, 安東 直紀, 山田 忠史, 谷口 栄一, 災害時における配送量の優先度を考慮した救援物資配送モデルの構築, 土木学会論文集 D3 (土木計算学), Vol.67, No.5 (土木計画学研究・論文集第 28 号巻), (2018), I.887–I.897.
- [19] OpenJij チュートリアル, <https://tutorial.openjij.org/build/html/ja/index.html>

- [20] M. M. Solomon, Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operation research*, Vol.35, No.2, (1987), pp.254–265.
- [21] S.Suzuki and M. Okada, Residual energies after slow quantum annealing, *J. Phys. Soc. Jpn.* Vol.74, (2005), pp.1649–1652.
- [22] 谷口 栄一, 山田 忠史, 柿本 恭志, 所要時間の不確実性を考慮した都市内集配トラックの確率論的配車配送計画, 土木学会論文集, No.674/IV-51,(2001), 49–61.
- [23] 棚橋 優, 今堀 慎治, 配送計画問題に対するデータベース付きメタ戦略, 数理解析研究所講究録, 第 1879 巻, (2014), pp.164–pp.179.

(原稿提出: 2023 年 1 月 8 日; 修正稿提出: 2023 年 1 月 23 日)