Vol.2022-QS-5 No.14
2022/3/24

相乗りを考慮したデマンド交通サービスの配車計画問題への量子アニーリングの適用

篠原　秀人[1,a)]　山城　悠[1,b)]　齋藤　和広[2,c)]　梅木　智光[2]

概要：近年，デマンド交通サービスにおいて，同一方向に移動する利用者のマッチングを行い，同乗することで効率的に移動する相乗りサービスが注目されている．相乗りを考慮したデマンド交通サービスの配車計画問題は，一般的な配送計画問題（VRP: Vehicle Routing Problem）をより現実的な条件に拡張した問題となる．本稿では，相乗りを考慮したデマンド交通サービスの配車計画問題を最適化問題としてモデル化を行い, 量子アニーリングを用いて本問題の求解を行った. 制約付き最適化問題を量子アニーリングが扱える形である QUBO（Quadratic Unconstrained Binary Optimization）へ変換するには一般的にはペナルティー法が用いられるが, ペナルティー法を元にした QUBO では効率的に解を得られることができなかったため, 拡張ラグランジュ法を用いて変換を行い, 量子アニーリングを用いて求解し, 評価を行った.

キーワード：配送計画問題, VRP, 量子アニーリング

1. はじめに

近年，デマンド交通サービスにおいて，同一方向に移動する利用者のマッチングを行い，同乗することで効率的に移動する相乗りサービスが注目されている．相乗りを考慮したデマンド交通サービスの配車計画問題は，一般的な配送計画問題（VRP: Vehicle Routing Problem）をより現実的な条件に拡張した問題となる．そこで, 効率的に良い配車計画を求める必要があるが, VRP は NP 困難と呼ばれるクラスに属する問題であり, 多項式時間で厳密解法が存在しないと言われている．このような問題に対して注目されているのが, 量子アニーリング法と呼ばれる, 量子力学的な現象を利用したアルゴリズムである．量子アニーリング法では, 制約なし 0-1 整数計画問題を解くアルゴリズムであり, アニーリング法と同様に最適解に収束することが示されている [1].

量子アニーリングを用いた配送計画問題はこれまでにもいくつかの研究が行われている [2] [3]. 一般的な配送計画問題では, どの車両が, どの順番で顧客を訪問するか, というのを決定する最適化問題となる．一方で今回の相乗りを考慮したデマンド交通サービスの配車計画問題は, 容量制約・時間枠制約があり, 顧客へ訪問する時間や訪問する際の車の容量も決定する必要がある．そのため, 今回の問題は, 従来の量子アニーリングを用いた VRP に関する研究よりも現実に近い問題となり, より実用的なものとなる.

本稿では, 今回の相乗りを考慮したデマンド交通サービスの配車計画問題を最適化問題としてモデル化を行い, 量子アニーリングを用いたアルゴリズムの提案を行った．さらに, モデルとアルゴリズムの評価のために, 具体的なデータを用いて数値実験を行った.

2. モデル化

2.1 問題設定

今回対象とする相乗りを考慮したデマンド交通サービスの配車計画問題についての具体的な説明を行う．本問題は, 容量・時間枠付き・Pickup-Dropoff の配送計画問題の拡張となる．相乗りが可能か不可能かのフラグを持った予約と使用可能なタクシーの情報が事前に与えられるとする．ここで, 予約はその予約を処理する人数, 予約した人々を乗せる地点, 降ろす地点, 乗せる際の時間枠, 降ろす際の時間枠を持っているとする．タクシーには営業開始時間, 営業終了時間, 休憩開始時間, 休憩終了時間, タクシーの乗せることのできる人数の容量を持っているとする．また, 予約を乗せたり降ろしたりする地点には, 待機可能な時間の上限

[1] 株式会社 Jij
Jij Inc., SB Buildgin, 1-4-6, Nezu, Bunkyo, Tokyo 113–0031, Japan
[2] KDDI 株式会社
KDDI CORPORATION, Garden Air Tower, 3-10-10 Iidabashi, Chiyoda-ku, Tokyo, 102-8460 JAPAN
[a)] h.shinohara@j-ij.com
[b)] y.yamashiro@j-ij.com
[c)] ku-saitou@kddi.com

が与えられているとする．これは, 駅前などでタクシーが早く付きすぎてその場で可能な待機時間の超過を禁止することを意味する．

ここで, 全ての予約を処理する配車計画の中で, 全てのタクシーの空き時間を最大化するようなものを求めることを目的とする．ここで空き時間とは, 後述する Depot に向かう時間と Depot で待機する時間の総和である．またこのとき, 予約の時間枠, 待機時間, 相乗りの可不可を満たさなくてはいけない．すなわち, 予約には時間枠に収まる範囲でタクシーがサービスを提供し, 相乗り不可の予約に対してはその予約を乗せてから降ろすまでの間, 他の予約の顧客を乗せている状態であることを許さない．

現実の配車計画問題では，予約は逐次的に与えられ，その都度に配車の可否を判定する．今回の問題では，事前に逐次的に与えられた予約に対して，配車が成立した予約を対象に最適化を行う．したがって，最適化対象となる全ての予約は，必ず条件を満足してタクシーに割り当てることが可能な予約であることを前提とする．この最適化によってタクシーの空き時間を最大化することで，その後に逐次的に与えられる予約に対して最大限対処できるように今回の目的関数を設計した．

2.2 用語の定義

2.1 で定義した問題を最適化問題へ定式化を行うために, 用語の定義を行う．問題を定式化する際には, タクシーは休憩開始前と休憩終了後で独立した 2 つのタクシーに分けるとする．こうすることで全てのタクシーに関しては営業開始時間と営業終了時間のみを考えれば良くなる．

相乗りが可能な予約はその予約した顧客を乗せる（Pick-Up）ための予約 r^+ とその顧客を降ろす（Drop-Off）ための予約 r^- に分ける．このとき, 実際の顧客の人数を k として, 乗せる予約の容量を $p_{r^+} = k$, 降ろす予約の容量を $p_{r^-} = -k$ と定義する．また, 乗せる予約 r^+ とそれに対する下ろす予約 r^- の組は対応づけられているとする．相乗りが不可能な予約に対しては, このような分離を行わない. なぜならば相乗りが不可の予約を処理する場合は, その予約を乗せる地点から下ろす地点へ直接移動する以外方法が存在しないからである．相乗り不可の予約 s の容量 $p_s = 0$ と定義する．

今回の問題では, 1 つの Depot と呼ばれる特別な拠点が存在する．全てのタクシーはこの拠点から営業をスタートし, かならず営業終了時刻までにこの拠点に帰ってくる必要がある．タクシーが長時間待機する場合には Depot に戻って待機するとする．定式化では, この出発する際の Depot, 到着する際の Depot, 待機する際の Depot を分けて, これらをダミーの予約として扱う．

予約間の移動時間は移動時間行列 D として事前に計算されているとする．このとき, 移動時間行列の各要素はある単一時刻に関して規格化されており, 要素が全て整数値となる行列として与えられているとする．

これらの用語をまとめると以下のようになる．

- $k \in K$：タクシーの集合
- C_k：タクシー k の乗車可能人数（容量）
- $[e^k, l^k]$：タクシー k の営業時間
- N：予約数
- p_r：予約 r の容量
- $[e_r, l_r]$：予約 r の時間枠
- W_r：予約 r における待機可能時間
- D：予約間の移動時間の行列
- D^+：出発の Depot
- D^-：到着の Depot
- D^0：待機専用の Depot
- $r^+ \in R^+$：Pick-Up 予約の集合
- $r^- \in R^-$：Drop-Off 予約の集合
- $(r^+, r^-) \in R^{+,-}$：Pick-Up 予約と Drop-Off 予約の組の集合
- $s \in S$：相乗り不可予約の集合
- $r \in R = R^+ \cup R^- \cup S$：全ての予約の集合
- $t \in T$：時刻の集合

2.3 整数計画問題への定式化

今回の問題を整数計画問題へ定式化する．まず以下の 4 つの変数を定義する．

$$x^k_{r,r'} := \begin{cases} 1 : \text{タクシー } k \text{ が予約 } r \text{ から予約 } r' \text{に直接移動するとき} \\ 0 : \text{otherwise} \end{cases}$$

$$t^k_r := \text{タクシー } k \text{ が予約 } r \text{ に到着する時間}$$

$$w^k_r := \text{タクシー } k \text{ が予約 } r \text{ を処理する際に待機する時間}$$

$$c^k_r := \text{予約 } r \text{ を処理する直前のタクシー } k \text{ の容量}$$

移動時間行列 D や予約の容量 p_r が整数である限り, 変数 t, w, c は整数の範囲で最適解を持つことがわかるため, 量子アニーリングでこの問題を扱う際は変数 t, w, c を整数値の範囲の変数として扱う．

以上を用いて今回の問題を整数計画問題として定式化すると以下のようになる．ここで M は十分大きな定数である．

$$\begin{aligned} \min \sum_{\forall k \in K} (w^k_{D^+} + w^k_{D^-}) + \sum_{\forall k \in K} w^k_{D^0} \sum_{\forall r \in R} x^k_{r,D^0} + \\ \sum_{k, r \in R \cup \{D^0\}} \sum_{d \in \{D^-, D^0\}} D_{r,d} x^k_{r,d} \end{aligned} \tag{1}$$

subject to

$$\sum_{\forall k \in K} \sum_{\forall r \in R \cup \{D^0, D^+\}} x^k_{r,r'} = 1, \quad \forall r' \in R \cup \{D^-\} \quad (2)$$

$$\sum_{\forall r \in R \cup \{D^0, D^+\}} x^k_{r,r'} = \sum_{\forall r \in R \cup \{D^-, D^0\}} x^k_{r',r}, \quad \forall r' \in R \cup \{D^0\}, \forall k \in K \quad (3)$$

$$\sum_{\forall r \in R \cup \{D^0, D^-\}} x^k_{D^+,r} = 1, \quad \forall k \in K \quad (4)$$

$$t^k_{D^+} = e^k, \quad \forall k \in K \quad (5)$$

$$c^k_{D^+} = 0, \quad \forall k \in K \quad (6)$$

$$\sum_{\forall r \in R \cup \{D^0, D^+\}} x^k_{r,D^-} = 1, \quad \forall k \in K \quad (7)$$

$$t^k_{D^-} + w^k_{D^-} = l^k, \quad \forall k \in K \quad (8)$$

$$c^k_{D^-} = 0, \quad \forall k \in K \quad (9)$$

$$\sum_{\forall r \in \{D^+\}} x^k_{r,D^0} \leq 1, \quad \forall k \in K \quad (10)$$

$$c^k_{D^0} = 0, \quad \forall k \in K \quad (11)$$

$$\sum_{\forall r \in R \cup \{D^+, D^0\}} x^k_{r,r^+} = \sum_{\forall r' \in R \cup \{D^+, D^0\}} x^k_{r',r^-}, \quad \forall k \in K, \forall (r^+, r^-) \in R^{+,-} \quad (12)$$

$$e_r \leq t^k_r + w^k_r \leq l_r, \quad \forall r \in R, \forall k \in K \quad (13)$$

$$0 \leq w^k_r \leq W_r, \quad \forall r \in R, \forall k \in K \quad (14)$$

$$t^k_r + w^k_r + D_{r,r'} - M(1 - x^k_{r,r'}) \leq t^k_{r'}, \quad \forall r, r' \in R, \forall k \in K \quad (15)$$

$$t^k_{r'} \leq t^k_r + w^k_r + D_{r,r'} + M(1 - x^k_{r,r'}) \quad \forall r, r' \in R, \forall k \in K \quad (16)$$

$$0 \leq c^k_r \leq C^k, \quad \forall r \in R, \forall k \in K \quad (17)$$

$$c^k_r + p_r - M(1 - x^k_{r,r'}) \leq c^k_{r'}, \quad \forall r, r' \in R, \forall k \in K \quad (18)$$

$$c^k_r = 0, \quad \forall s \in S \cup \{D^+, D^-, D^0\}, \forall k \in K \quad (19)$$

目的関数 (1) の第一項は出発前と到着後の Depot での待機時間を, 第二項は営業中の Depot で待機時間を, 第三項は Depot へ戻る際の移動時間を表す. 制約 (2) は Depot を除く全ての予約をちょうど 1 回処理しなければいけないことを意味する. 制約 (3) はタクシーが巡回する経路がちょうど 1 つの閉路を成している必要があることを意味する. 制約 (4) ～ (6) はタクシーが Depot から出発する際に満たすべき条件である. 制約 (7) ～ (9) はタクシーが Depot から到着する際に満たすべき条件である. 制約 (10), (11) はタクシーが Depot で待機する際に満たすべき条件である. 制約 (12) は Pick-Up の予約と Drop-Off の予約を同一のタクシーが処理する必要があることを意味する. 制約 (13) は予約にサービスを開始する際の時間枠を満たす必要があることを意味する. 制約 (14) はタクシーが予約を処理する地点において待機できる時間の上限を表している. 制約 (15), (16) はタクシーの予約間の移動が矛盾なく表されていることを意味する. 制約 (17) はタクシーに関する容量制約である. 制約 (18) はタクシーの容量の変化に関する制約である. 制約 (19) は相乗り不可の予約に関する制約である.

3. アルゴリズム

3.1 0-1 整数計画問題への変換

2.3 で定式化したモデルは, 変数に整数（場合により連続変数と解釈できる）を含めた整数計画問題と呼ばれる問題である. 一方で, 一般に量子アニーリングで扱える問題は 0 か 1 どちらか一方のみの値を取る変数から成る 0-1 整数計画問題である. よって, 2.3 の問題を 0-1 整数計画問題に変換する必要がある. 今定義されている変数は x, t, w, c であり, この中で 3 つ以上の値を取りうる整数変数は t, w, c である. 具体的に t の 0-1 整数への変換方法を示すが他の変数も同様である.

一般的に整数変数は, 2 進数表記を用いて変数の個数を増やすことにより 0-1 整数の変数への変換が可能である. t には制約 (13) により取りうる値の上限・下限が定められている. 今変数 t に以下のような上限・下限が定められているとする.

$$l \leq t \leq r$$

このとき, t の取りうる値の種類数は $m = r - l + 1$ であり, このとき 2 進数表記を用いて,

$$t = l + \sum_{k=0}^{\lfloor \log_2 m \rfloor - 1} 2^k * t'_k + R t'_{\lfloor \log_2 m \rfloor}$$

と表すことができる. ここで t'_k は t を 2 進数表記したときの k 桁目が 0 か 1 かを表す変数であり, $R = r - l - (2^{\lfloor \log_2 m \rfloor} - 1)$ である. このような変換を行い一般的な整数変数を 0-1 整数計画問題へ変換する [4].

3.2 前処理

今回の問題を愚直に解こうとすると変数の数や制約の個数が非常に大きくなる. より大きな問題を解くために, 変数や制約の削減をするための前処理を行う.

3.2.1 変数の削減

今回定義される変数の個数は $O(|K|N^2)$ と非常に大きくなることが予想される. しかしながら予約の時間枠とタクシーの時間枠が存在するため, 実際に使用する変数はそこまで多くない. そこで, 実際には使用しない変数を 0 と予め値を割り振ることで変数の削減を行う. 具体的には,

- 処理不可能なタクシーと予約の組に対しての変数の削減
- 移動不可能な予約の組に対する変数の削減

に対して行う.

タクシーには営業開始時間と営業終了時間が存在し, 各予約にもその予約にサービスを提供できる時間枠が与えられている. そのため, それぞれのタクシーに対して予約を提供できない組合せが存在する. そのような組合せに対して変数の削減を行う. またタクシーには容量が存在し乗せられる顧客の上限が定められている. よって, 容量的に乗せることのできないタクシーと予約の組が存在する. そのような組合せに対して変数の削減を行う.

前述の通り予約にはサービスを提供できる時間枠が与えられているため, 直接移動すると時間枠制約を満たさない予約の組が存在する. よってそのような予約の組に対して変数の削減を行う. また予約の容量を考慮した際に, 乗せる予約間を直接移動した際には容量的に処理できないタクシーが存在する. 例えば, 5 人を乗せる予約から 4 人を乗せる予約に直接移動した場合, 容量が 8 以下のタクシーはこれを処理することができない. このようなタクシーと予約の組に対して変数の削減を行う.

3.2.2 制約の削減

3.2.1 の変数の削減を行った結果, 全て定数から成り，変数を含まない制約式が存在する場合がある. そのような制約式は変数の削減に併せて削除する. また, 制約式の中には 1 変数から成る制約が存在する. そのような制約式が等式制約の場合は，予めその等式制約の値をその変数に割り当てることで対処し, 不等式の場合には上限・下限から 2 進数変換の際の処理によって対処する. 具体的には, 制約 (5), (6), (9), (11), (19) は変数をあらかじめ固定することによって制約を削減する. また, 制約 (13), (14), (17) などの不等式制約は, 2 進数変換の際の処理によって制約を削減する.

3.2.3 QUBO への変換

2.3 で定義された問題は整数への変換処理を行えば, 制約あり 0-1 整数計画問題と呼ばれる問題の形式となる. 量子アニーリングが扱える問題の形式は制約無しの最適化問題であるため, 制約式を何らかの形で変換する必要がある. 量子アニーリングにおける制約を変換する手法はペナルティー法と呼ばれる手法が主流として扱われている [5]. しかしながら, 今回行った定式化には Big M を用いた不等式制約を採用しており, これらの制約をペナルティー法として変換するためには, 決定変数の他に制約の個数分のスラック変数が必要となる. 量子アニーリングが扱える変数の個数は多くないため, 実際の問題を扱う場合にこれは大きな困難となる.

この困難を回避する方法として, ペナルティー法ではなく拡張ラグランジュ法を使用することを提案する [6][7][8]. 拡張ラグランジュ法では, 以下のような制約付き最適化問題

$$
\begin{aligned}
&\min && f(x) \\
&\text{subject to} && \\
& && g_k(x) = 0, \quad \forall k \\
& && h_l(x) \le 0,, \quad \forall l
\end{aligned}
$$

に対して, 拡張ラグランジュ関数

$$
L(x,\lambda,\mu,\lambda',\mu') = f(x) + \sum_{\forall k} \lambda_k g_k(x) + \sum_{\forall k} \frac{\mu_k}{2} |g_k(x)|^2 + \sum_{\forall l} \lambda'_l h_l(x) + \sum_{\forall l} \frac{\mu'_l}{2} |\max(\mu'_l h_l(x), 0)|^2
$$

をパラメータ $\lambda, \mu, \lambda', \mu'$ の更新を行いながら解く方法である. パラメータの更新及び, 具体的な拡張ラグランジュ法の手順は以下のようになる.

Step1： パラメータ $\lambda^0, \mu^0, \lambda'^0, \mu'^0$ を 適当な値で初期化し, $k = 0$ とする

Step2： $L(x, \lambda^k, \mu^k, \lambda'^k, \mu'^k)$ を解き, 解 x^k を得る.

Step3： 各パラメータを以下のように更新する

- $\lambda^{k+1} = \lambda^k + \mu^k * g(x^k)$, $\mu^{k+1} = \alpha\mu^k, \alpha > 1$,
- $\lambda'^{k+1} = \lambda'^k + \max(\mu'^k h(x^k), 0)$, $\mu'^{k+1} = \beta\mu'^k, \beta > 1$

Step4： $k = k + 1$ として Step2 へ

以上の反復を予め決められた回数行う.

3.3 2 段階アニーリング

今回の問題では, 移動時間をちょうど満たすようなタクシーの配車計画を求める必要がある. そのため, 時間に関する変数 t, w が取りうる変数の値の領域に対して, 実行可能な領域が相対的に非常に小さくなる. 以上のことが原因で, 実行可能解を見つけることが困難になると予想される. そこで, 2 段階で最適化を行うことを提案する. 2 段階の最適化では, 最初に元々の問題である 2.3 をそのまま解き, 得られた変数の値の中で変数 x のみを固定した最適化問題を考える. その後, 2 段階目で x を固定した最適化問題をもう一度解く. これは, 最初にタクシーの配車のルートのみを最初に決定して, 時間や容量の詳細は 2 段階目で決定することを意味する. この 2 段階の手法の手順を以下にまとめる.

Step1： 2.3 のモデルに対してアニーリングアルゴリズムを実行する

Step2： Step1 で得られた解のうち, 変数 x を得られた解の値で固定した最適化問題に対して再度アニーリングアルゴリズムを実行する

4. 実験と結果

提案したモデルとアルゴリズムの評価のために, いくつかのデータセットに対していくつかの量子アニーリングアルゴリズムを適用し実験を行った.

4.1 評価環境

使用した各種ソルバーは以下の通りである

- D-Wave Advantage 4.1 [9]
- Jij SASampler [10]

D-Wave Advantage 4.1 は, 量子アニーリングを実行する物理的なマシンである. Jij SASampler は, 古典的なアニーリング法である. どちらも Python 用 API が提供されており, それを利用した.

4.2 評価データ

モデルとアルゴリズムの正当性の確認のための小規模な人工的に作成したデータと, 実在する状況から作成した中規模のデータセットを用いて実験を行った. 以下の表 1 が使用したデータセットの問題の詳細である. Simple Data Not Shared では, 全ての予約が相乗り不可のデータであるのに対し, その他のデータセットでは相乗り不可の予約と可能な予約の両方が存在するデータセットである. v2-0900-1500 は, 実在のデータとして公開されているニューヨーク市のデマンド交通サービスの利用データ [11] を利用し, Brooklyn エリア内で完結する Yellow Taxi と Green Taxi のタクシー利用履歴から, 本エリア内の中心地点を Depot として 2019 年 01 月 09 日の 9:00 から 15:00 にタクシー 2 台相乗りなしで成立する予約を抽出した.

表 1 使用したデータセット

Data Set	$\|K\|$	N	相乗り	説明
Simple Data	2	3	可	人工データ
Simple Data Not Shared	2	2	不可	人工データ
v2-0900-1500	2	28	可	オープンデータ

4.3 提案モデルの評価

2.3 で定式化した提案モデルに対して 3.2 で説明した前処理を行い各ソルバーで実験を行った. D-Wave Advantage4.1 を用いて, Simple Data Not Shared に対して chain strength と実行時間を変化させて各 500 回サンプルを行った. その実験を行った結果が図 1, 2 である. 最適解を得ること以前に実行可能解を得ることが困難な問題であったため, 実行可能解にたどり着くまでの時間として Time To Solution を計算した. すなわち,

$$\text{Time To Solution} = \tau \frac{\ln(1-0.99)}{\ln(1-p_s(\tau))}$$

τ : 一回あたりのアニーリング時間

$p_s(\tau)$: 実行可能解を得る確率（実験から得たサンプルで推定）

として計算を行った.

Jij SASampler では前処理のみの実行では, Simple Data Not Shared を含め他のデータセットで実行可能解を得ることができなかった. また, D-Wave Advantage4.1 を用いた実験でも, その他のインスタンスでは実行可能解を得ることができなかった.

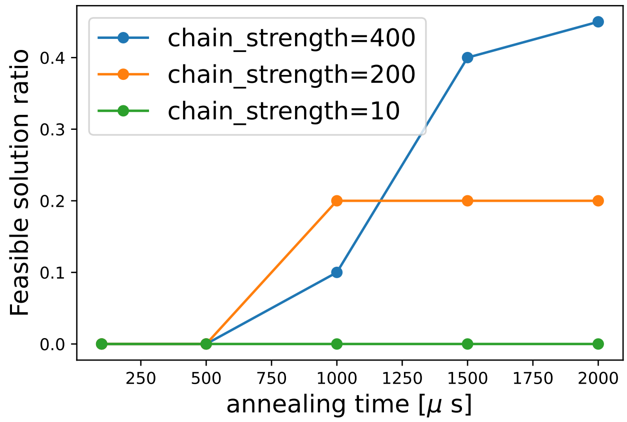

図 1 Simple Data Not Shared に対して各 chain strength 別のアニーリング時間に対する得られた実行可能解の割合

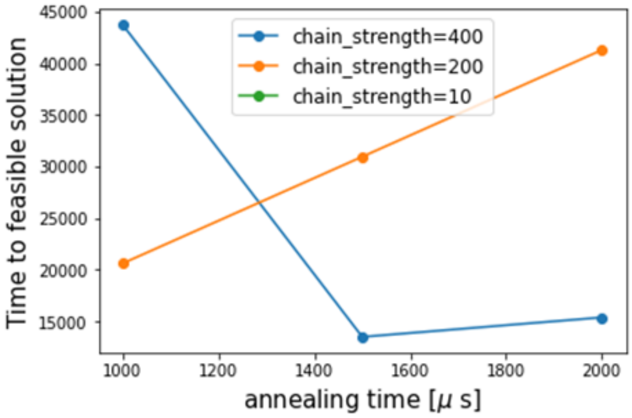

図 2 Simple Data Not Shared に対して各 chain strength 別のアニーリング時間に対する TTS

4.4 2 段階アニーリング

3.3 で提案した 2 段階アニーリングを用いて実験を行った. Simple Data Not Shared に対して 500 回サンプルを取った実験結果が以下の図 3, 4 である. なお実験では 1 段階目・2 段階目ともに Jij SASampler を使用した.

Simple Data に対して 500 回サンプルを取った実験結果が以下の図 5, 6 である. なお実験では 1 段階目・2 段階目ともに Jij SASampler を使用した.

この実験結果では, Feasible Solution Ratio が非常に低くなっているが, 2 段階目の最適化においてアニーリングではなく通常の最適化ソルバーを用いると実行可能解の割合は 8 割程度まで上昇した. これらの実験より, 2 段階アニーリングの手法が有効であることがわかる.

5. 考察

今回の問題は,

- どのタクシーがどの予約を処理するかという割当
- 各タクシーに対して, 割当られた予約を処理する順番

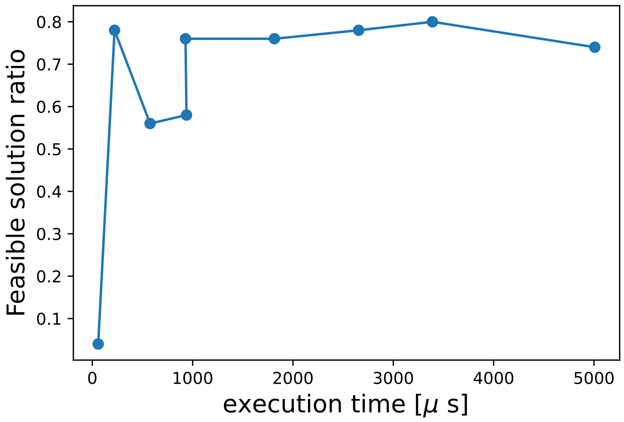

図 3 Simple Data Not Shared における実行時間に対する実行可能解の割合

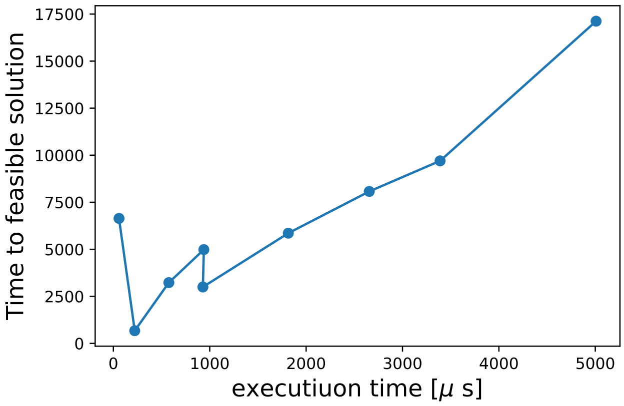

図 4 Simple Data Not Shared における実行時間に対する TTS

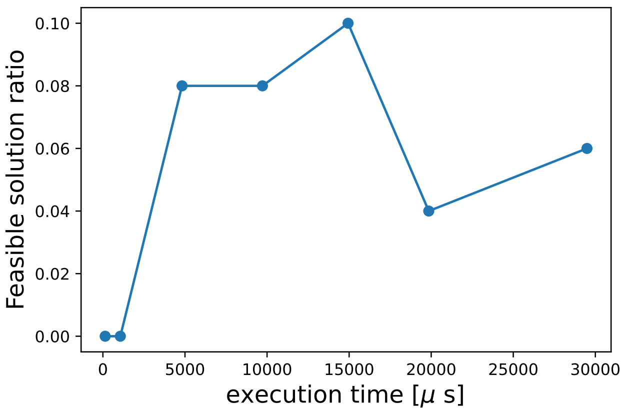

図 5 Simple Data における実行時間に対する実行可能解の割合

- 予約を処理する際の, 時間・容量の決定

を求める必要があり, 最適化問題の中でも非常に難しい問題のクラスである. 量子アニーリングが解ける形は QUBO (Quadratic Unconstrained Binary Optimization：二次制約なし二値最適化) であり, 今回のような制約があり二値以上の変数を用いる問題では, 制約のペナルティーへの変換・整数の二値変数への変換を行う必要がある. 更に, 不等式制約などは変数を新たに追加して等式制約へ変換した後でペナルティーに変換する必要がある. 今回の実験を通

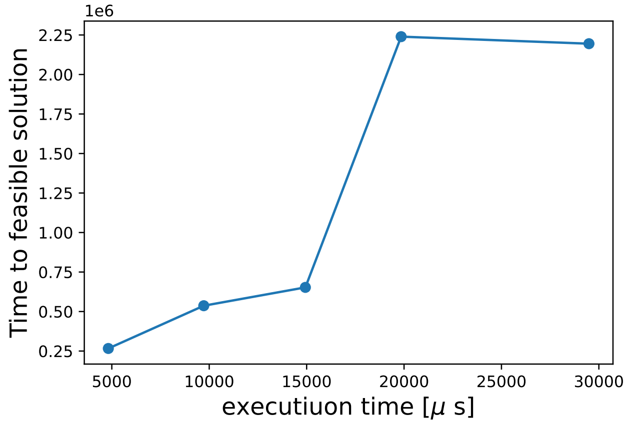

図 6 Simple Data における実行時間に対する TTS

して, 本来整数であった変数に関する制約の違反が多い傾向にあることがわかった. これは整数に関する解領域が広いことに比べて実行可能な領域が非常に狭いことが原因であると推測できる. 実際に 2 段階アニーリングを用いた実験では, 本来二値整数である x は 1 段階目で実行可能な割当を得られたが, 整数変数 t, w, c に関しては 2 段階目のアニーリングを行わなければ実行可能な解を得られなかったことからも推測することができる. また, 不等式制約についても制約違反が多い傾向にあった. 特に Big M を用いた不等式制約は, 制約が満たされない場合のペナルティーの差が大きくペナルティーの重み調整が非常に難しいことがわかった.

6. おわりに

相乗りを考慮したタクシーの配車計画問題をモデル化し, 量子アニーリング技術を用いて解いた. 非常に小さいインスタンスであれば D-Wave でそのまま解くことができた. 一方で問題規模が予約 10 程度になると D-Wave や Jij SASampler では現実的な時間で実行可能解を得ることができなかった. また, 2 段階でアニーリングの手法を適用することで実行可能解を得る確率が上昇し, 本手法が本問題で有効であることがわかった. 現実における最適化問題では, 今回の問題のような

- 0-1 の二値変数ではなく, 複数の値を取りうる整数変数
- 不等式制約や Big M を用いた不等式制約

などが出てくる場合がほとんどである. このような問題に対してそのままアニーリングを実行すると, 現段階のアニーリングの精度では実行可能解を得られる確率は非常に低い. 今回用いた 2 段階アニーリングが有効であったためこれらのアルゴリズムを発展させることにより, 現実的な問題もアニーリング技術を用いて解くことができると期待される.

Vol.2022-QS-5 No.14
2022/3/24

参考文献

[1] Tadashi Kadowaki and Hidetoshi Nishimori: Quantum annealing in the transverse Ising model, Phys. Rev. E - Stat. Physics, Plasmas, Fluids, Relat. Interdiscip Top., vol. 58, no. 5, pp. 5355–5363, 1998.

[2] Hirotaka Irie, Goragot Wongpaisarnsin, Masayoshi Terabe, Akira Miki and Shinichirou Taguchik : Quantum Annealing of Vehicle Routing Problem with Time, State and Capacity (2019).

[3] 齋藤 和広, 大山 重樹, 梅木 智光, 黒川 茂莉, 小野 智弘: 配送計画問題における量子アニーリングの評価, 情報処理学会論文誌データベース, vol. 14, no. 1, pp. 8–17 (2021).

[4] jijmodeling docs, https://www.jijmodeling.jijzept.com/

[5] A. Lucas: Ising formulations of many NP problems, Front. Phys., vol. 2, no. February, pp. 1–14, 2014.

[6] Bertsekas, Dimitri P: Constrained optimization and Lagrange multiplier methods, Athena Scientific (1996) .

[7] Xiaoxi Jia, Christian Kanzow, Patrick Mehlitz and Gerd Wachsmuth: An Augmented Lagrangian Method for Optimization Problems with Structured Geometric Constraints (2021).

[8] Kouki Yonaga, Masamichi J. Miyama, and Masayuki Ohzeki: Solving Inequality-Constrained Binary Optimization Problems on Quantum Annealer (2020) .

[9] Catherine McGeoch and Pau Farré: The Advantage System: Performance Update (2021), https://www.dwavesys.com/media/kjtlcemb/14-1054a-a_advantage_system_performance_update.pdf.

[10] JijZept Documentation, https://www.documentation.jijzept.com/

[11] NYC TLC Trip Record Data, https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[12] D-Wave Ocean Software Documentation, https://docs.ocean.dwavesys.com/en/stable/index.html