

事例・実践論文

配送計画問題における量子アニーリングの評価

齋藤 和広^{1,a)} 大山 重樹¹ 梅木 智光² 黒川 茂莉¹ 小野 智弘¹

受付日 2020年6月9日, 採録日 2020年9月30日

概要: 配送計画問題 (VRP: Vehicle Routing Problem) は, 複数配送車の配送順序を最適化する問題で, 物流における配達順序やデマンド交通サービスにおける配車計画など, 様々な現実問題に応用可能な問題である. VRP は代表的な組合せ最適問題の 1 つとして NP-Hard と呼ばれる問題であり, 従来のコンピュータでは多項式時間で厳密解を求めることが困難であることが知られている. 量子アニーリングは, このような組合せ最適化問題を現実時間で解くことが期待されている. 本論文では, デマンド交通サービスを想定した規模の VRP における量子アニーリングの効果を明らかにするために, 基礎的な VRP で量子アニーリングを評価した. D-Wave 2000Q を用いて量子アニーリングの性能評価を行った結果, 配送車 2 台 4 拠点の VRP において約 40ms で最適解を導出することを確認した. しかし, 現在のハードウェアでは量子ビット数に限界があり, デマンド交通サービスの実利用を想定した規模にスケールした場合の性能を評価することができない. そこで, アニーリングシミュレータとして, ハードウェア実装の富士通デジタルアニーラとソフトウェア実装の OpenJij を用いることで量子アニーリングのスケール評価を行った. 加えて VRP の従来手法として厳密解手法および近似解手法のソルバーと比較した. 評価の結果, 厳密解手法で実行できない規模において, 目的関数の観点でアニーリングシミュレータが既存の近似解手法より最大 10% 精度向上することを確認した. 一方, 実行時間の観点ではアニーリングシミュレータが既存の近似解手法より 2 桁以上の差で劣後することが分かった. これらの評価結果をふまえ, 実利用を想定した VRP において量子アニーリングを活用するための課題とマシン活用の条件を議論する.

キーワード: 量子アニーリング, アニーリングシミュレータ, 組合せ最適化問題, 配送計画問題

Evaluation of Quantum Annealing for Vehicle Routing Problem

KAZUHIRO SAITO^{1,a)} SHIGEKI OYAMA¹ TOMOAKI UMEKI² MORI KUROKAWA¹ CHIHIRO ONO¹

Received: June 9, 2020, Accepted: September 30, 2020

Abstract: Vehicle Routing Problem (VRP) is an NP-Hard combinatorial optimization problem which can be applied for logistics, transportation system, etc. by adding service constraints like that should be visited exactly once. It is widely known that a current computer is hard to calculate exact solutions of the VRP in polynomial time. Quantum annealing is expected to solve such an NP-Hard problem in polynomial time as one of a quantum computer. In this paper, we evaluate a simple VPR using quantum annealing machine to reveal the availability of quantum annealing for a scale of demand-based ride sharing system as a real world application of the VRP. As a result of the evaluation using D-Wave 2000Q, we found that it takes about 40ms to derive an exact solution of the VRP setting on 2 vehicles and 4 locations. However, the current quantum annealing machine cannot scale up to the size assuming the real world application due to a limitation of the hardware implementation. Therefore, to evaluate scalability of the quantum annealing, we use two annealing simulators, Fujitsu Digital Annealer as a hardware and OpenJij as a software, and compare them to two existed solvers, an exact method and an approximate method. The evaluation of the objective function indicated that the annealing simulators reduce up to 10% of the total costs compared to the approximate method in scale where the exact method cannot execute. On the other hand, regarding the execution time, the existed approximate method outperformed the annealing simulators over 100x. Based on these evaluation results, we discuss future problems and conditions of quantum annealing utilization for VRP intended for real world applications.

Keywords: quantum annealing, annealing simulator, combinatorial optimization problem, VRP

1. はじめに

配送計画問題 (VRP: Vehicle Routing Problem) は、複数配送車の配送順序を最適化する組合せ最適化問題であり、巡回セールスマン問題を一般化した問題である [1]. 応用例として、物流における物品配送や工場内の部品移動などがある. VRP は NP-Hard と呼ばれる問題として知られており、従来のコンピュータでは多項式時間で厳密解を求めることが困難な問題である. 本研究では、VRP の応用として、乗客の要求に応じて配車するデマンド交通サービスの配車計画問題を対象とする. この配車計画問題は、VRP として非常に複雑な問題である. 制約として、出発・到着時間、地点到着の前後関係、車載可能人数の制約が必要であり、加えてサービス形態に応じた様々な制約 (中継地点対応、許容遅延時間など) が発生する.

近年、従来のコンピュータでは計算が困難な組合せ最適化問題に、量子アニーリングの活用が期待されている. 量子アニーリングは、量子効果を利用して NP-Hard な組合せ最適化問題をヒューリスティックに解く手法である [2]. 2011 年に D-Wave Systems 社から発表された量子アニーリングを実装したマシン [3] の登場を境に、量子アニーリングを実際のビジネス課題に適用できる可能性が出てきた.

量子アニーリングは、2つの量子ビット σ_i, σ_j の相互作用係数 J_{ij} 、および量子ビット σ_i に影響する外部磁場エネルギー係数 h_i から、ハミルトニアン \mathcal{H} を最小化する量子ビット σ_i のスピン方向 (+1/-1) を決定する. この仕組みはイジングモデルとして以下のように表現される.

$$\mathcal{H} = \sum_{i,j} J_{ij} \sigma_i \sigma_j + \sum_i h_i \sigma_i \quad (1)$$

組合せ最適化問題を最小化問題としてこのイジングモデルに定式化することで、量子アニーリングによって解くことができる. 定式化を直感的に行うため、以下のように量子ビット σ_i をバイナリ変数 $x_i \in \{0, 1\}$ に変換する.

$$x_i = \frac{\sigma_i + 1}{2} \quad (2)$$

量子アニーリング利用時は、これを式 (1) のイジングモデルに適用して以下のような QUBO (Quadratic Unconstrained Binary Optimization) 問題とする [4], [5], [7].

$$\mathcal{H} = \sum_{i,j} Q_{ij} x_i x_j \quad (3)$$

したがって、VRP の目的関数および制約を定式化し、式 (3) の QUBO 式を構築することで、VRP を量子アニー

リングに適用できる. しかし、現在の量子アニーリングの実装は、量子ビット間の相互作用関係が限定的で、利用できる量子ビットの数に限界がある. たとえば D-Wave Systems 社のマシンはキメラグラフと呼ばれるスパースなグラフ構造で、量子ビットどうしが複雑な相互作用関係になる問題の場合にグラフ変換が必要となり、利用する量子ビット数が大幅に増加する [6]. デマンド交通サービスの配車計画問題は、複雑な制約が必要となる VRP であり、量子ビットの相互作用関係が複雑となるため、量子ビット数の限界からサービス適用が困難な可能性がある.

そこで本論文では、デマンド交通サービスへの量子アニーリング適用を検証するために、制約を限定した VRP で量子アニーリングの性能を評価した. その結果、配送車 2 台 4 拠点の VRP において約 40 ms で最適解を導出することを確認した. しかし、現在のハードウェアでは量子ビット数に限界があり、デマンド交通サービスを想定した規模にスケールした場合の性能を評価することができない. そこで量子アニーリングのスケール性を評価するために、2種類のアニーリングシミュレータを用いて評価を行った. 本評価では、デマンド交通サービスを想定した 3 種の規模のデータを用意し、VRP の従来手法として全探索の厳密解手法およびヒューリスティックな近似解手法のソルバーと比較した. この比較評価から、厳密解手法で実行できない規模において、目的関数の観点でアニーリングシミュレータが近似解手法より最大 10% 精度向上することを確認した. 一方、実行時間の観点ではアニーリングシミュレータが近似解手法より 2 桁以上の差で劣後することが分かった. これらの評価結果をふまえ、実利用を想定した VRP において量子アニーリングを活用するための課題を議論し、マシン活用に関する条件について整理した.

本論文における貢献は、基礎的な VRP の QUBO 式を構築し、実サービス利用を想定したデータで量子アニーリングを評価した点と、評価結果をふまえて実利用を想定した VRP における量子アニーリング活用の課題とマシン活用条件を整理した点である.

本論文の構成は以下のとおりである. 2 章にて VRP の問題設定と量子アニーリングで実行するための定式化および QUBO について述べ、3 章で量子アニーリングで VRP を評価し、4 章で評価結果をふまえて VRP における量子アニーリング活用について議論し、5 章で関連研究を、6 章で結論を述べる.

2. VRP の量子アニーリング適用

2.1 問題設定

本研究で対象とする問題は、基礎的な VRP として、複数の配送車で対象となるすべての拠点を巡回する場合に、総移動コストが最小となる巡回ルートを探索する組合せ最適化問題である. 加えて、すべての配送車は Depot と呼ば

¹ 株式会社 KDDI 総合研究所

KDDI Research, Inc., Fujimino, Saitama 356-8502, Japan

² KDDI 株式会社

KDDI CORPORATION, Chiyoda, Tokyo 102-8460, Japan

^{a)} ku-saitou@kddi-research.jp

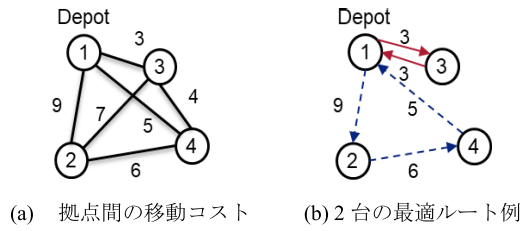


図 1 2台4拠点における VRP
Fig. 1 VRP with two vehicles and four locations.

れる拠点を出発点とし、必ず最後に Depot に帰還する問題を想定する。移動コストは拠点間の距離を利用する。なお、VRP を解くうえで必要な拠点間の移動コストは事前に計算済みとする。

問題例として4拠点における2台の配送車の巡回ルートを考える。図 1(a) で示すように、すべての拠点間の移動コストを事前に計算する。ここで、拠点1は Depot であり、配送車の出発点かつ到着点となる。最適解の1つとして図 1(b) が算出できる。赤線の配送車はルート 1→3→1 で移動コスト 6、青点線の配送車がルート 1→2→4→1 で移動コスト 20 となり、合計の移動コストが 26 となる。

2.2 定式化

量子アニーリング適用に向け、バイナリ変数をとる式 (3) の QUBO 式を構築するために、VRP の目的変数を以下で定義する。

$$x_{ps}^{(v)} = \begin{cases} 0 & \text{配送車 } v \text{ が拠点 } p \text{ に } s \text{ 番目に滞在していない} \\ 1 & \text{配送車 } v \text{ が拠点 } p \text{ に } s \text{ 番目に滞在している} \end{cases} \quad (4)$$

配送車の数を V 、拠点数を P 、訪問順序の最大値を S とすると、配送車 v 、拠点 p 、訪問順序 s は、以下の値をとる。

$$1 \leq v \leq V, \quad 1 \leq p \leq P, \quad 1 \leq s \leq S \quad (5)$$

拠点 i から拠点 j を移動する場合の移動コストを d_{ij} とすると、VRP の目的関数は式 (4) の目的変数 $x_{ps}^{(v)}$ の集合 X に対する最小化問題として以下で表現される。

$$\operatorname{argmin}_X \sum_{v=1}^V \sum_{i=1}^P \sum_{j=1}^P \sum_{s=1}^{S-1} d_{ij} x_{is}^{(v)} x_{j(s+1)}^{(v)} \quad (6)$$

配送車 v が訪問順序 s から $s+1$ に変化したときに滞在している拠点が変更することで移動と判断できる。式 (6) は、訪問順序 s で拠点 i に滞在し、かつ次の訪問順序である $s+1$ で拠点 j に滞在していた場合に移動コスト d_{ij} が加算される。なお、 $i=j$ では拠点が変更されていないので $d_{ij}=0$ となる。図 2 は、図 1(b) で示した配送車2台4拠点の最適ルート例における目的変数の集合 X の値である。たとえば $v=1, s=1, i=1, j=2$ において移動が判定

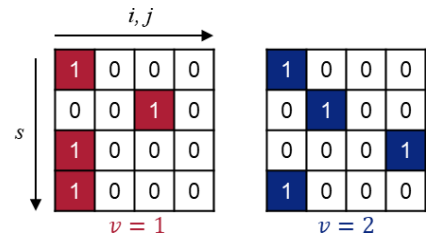


図 2 図 1(b) における目的変数の集合 X の値
Fig. 2 Binary values in a set of objective variables X in Fig. 1(b).

され、 $d_{12}=3$ が移動コストとして式 (3) で加算される。

制約条件として、拠点1を Depot とした場合に、それぞれの配送車における訪問順番1と S では必ず Depot に滞在する必要がある。したがって、以下の制約が発生する。

$$\forall v, \quad x_{11}^{(v)} = 1 \quad (7)$$

$$\forall v, \quad x_{1S}^{(v)} = 1 \quad (8)$$

VRP はいずれかの配送車ですべての拠点を訪問する必要がある。したがって、いずれかの配送車、およびいずれかの訪問順番で、すべての拠点が必ず1回ずつ訪問される制約が必要である。ただし、式 (7), (8) の制約によって Depot は最低2度訪問されるため、Depot を除いた制約とする。これらをふまえて以下の制約を加える。

$$2 \leq p \leq P, \quad \sum_{v=1}^V \sum_{s=1}^S x_{ps}^{(v)} = 1 \quad (9)$$

また、式 (4) の目的変数 $x_{ps}^{(v)}$ は、ある訪問順番にて配送車が消える、あるいは同時に複数の拠点到存在する可能性がある。これらの発生を排除し、すべての配送車、およびすべての訪問順番で、必ず1つの拠点到訪問するために、以下の制約を加える。

$$\forall v, \forall s, \quad \sum_{p=1}^P x_{ps}^{(v)} = 1 \quad (10)$$

以上の4つの制約項 (式 (7), (8), (9), (10)) をすべて満たしたうえで、目的関数である式 (6) から最小コストとなる各配送車の訪問ルートを算出する。

2.3 QUBO

量子アニーリングで実行するために、QUBO 式を構築する。2.2 節で定義した目的関数および制約からハミルトニアン \mathcal{H} を以下のように定義する。

$$\mathcal{H} = \mathcal{H}_A + \lambda_1 \mathcal{H}_B + \lambda_2 \mathcal{H}_C + \lambda_3 \mathcal{H}_D + \lambda_4 \mathcal{H}_E \quad (11)$$

$$\mathcal{H}_A = \sum_{v=1}^V \sum_{i=1}^P \sum_{j=1}^P \sum_{s=1}^{S-1} d_{ij} x_{is}^{(v)} x_{j(s+1)}^{(v)} \quad (12)$$

$$\mathcal{H}_B = \sum_{v=1}^V \left(1 - x_{11}^{(v)}\right)^2 \quad (13)$$

$$\mathcal{H}_C = \sum_{v=1}^V \left(1 - x_{1S}^{(v)}\right)^2 \quad (14)$$

$$\mathcal{H}_D = \sum_{p=2}^P \left(1 - \sum_{v=1}^V \sum_{s=1}^S x_{ps}^{(v)}\right)^2 \quad (15)$$

$$\mathcal{H}_E = \sum_{v=1}^V \sum_{s=1}^S \left(1 - \sum_{p=1}^P x_{ps}^{(v)}\right)^2 \quad (16)$$

\mathcal{H}_A は目的関数である式 (6) と一致し、拠点の移動に応じて移動コストが加算される。したがって、量子アニーリングによって、移動コストを最小化するように、量子ビットに対応する目的変数の値が変化する。 \mathcal{H}_B , \mathcal{H}_C , \mathcal{H}_D , \mathcal{H}_E はそれぞれ制約項である式 (7), (8), (9), (10) と一致する。それぞれの数式は制約違反時に括弧の二乗によって正の整数となり \mathcal{H} が増加するため、量子アニーリングが違反を回避するように目的変数の値を変化させる。

式 (11) における制約違反は、どの制約項であっても VRP としての問題設定に反するため、誤った解となる。そこで式 (11) におけるペナルティ係数 λ_1 , λ_2 , λ_3 , λ_4 の値を、 \mathcal{H}_A がとりうる値より十分に大きくすることで、制約違反の発生確率を下げるができる。一方、ペナルティ係数が極端に大きいことで、 \mathcal{H}_A の収束が不安定となり、総移動コストが低い解を得るまでに時間を要する。そのため、ペナルティ係数は制約違反が起きない程度に小さくするチューニングが必要となる。なお、解く問題によってペナルティ係数の基準となる数値を計算可能であり [7]、本研究の VRP においては、それぞれのペナルティ係数の初期値を $\max(d_{ij})$ とし、実行可能解 (制約違反していない解) を得る確率が目標値になること、またはその確率の増加が止まることを条件にペナルティ係数を増加させるチューニングを行う。このとき、それぞれの制約条件において違反回数を計測し、特に違反の多い制約条件のペナルティ係数を優先して増加させる。初期値で条件を満たしていた場合は、違反回数が最小の制約条件のペナルティ係数を減少させる。

3. 評価

3.1 評価内容

本評価では、制約を限定した基礎的な VRP として、2.1 節で述べた問題設定の VRP に対する量子アニーリングの適用を評価する。

式 (11) の QUBO 式を利用して、量子アニーリングマシンで実行することでその性能を評価する。しかし現在の量子アニーリングマシンは適用できる規模に著しい限界があるため、実サービスへの適用を評価するために、比較的大きな規模を実行可能で、かつ量子アニーリングと同様にイジングモデルを処理可能なアニーリングシミュレータを利用し、目的関数である総移動コストの値と、その解を導出

するまでの実行時間を計測する。この結果を VRP の既存手法と比較することで、スケーリングの観点で量子アニーリング適用を考察する。さらに、実際のデマンド交通サービスにおける規模を想定したデータ規模を複数利用した実行結果を同様に評価する。

3.2 評価環境

量子アニーリングの性能評価では、量子アニーリングをハードウェアで実装したマシンである D-Wave 2000Q [8] を利用した。本マシンをクラウド上で利用するために、D-Wave Ocean SDK [9] の API を利用した。D-Wave 2000Q の量子ビットはキメラグラフと呼ばれるスパースなグラフ構造で接続されているため、D-Wave Ocean SDK で提供されている Minor embedding というグラフ埋め込み手法 [5] を利用して QUBO を変換した。D-Wave 2000Q では 2,048 量子ビットを利用可能であるが、量子ビット間の相互作用が全結合になることで、このグラフ埋め込みによって利用可能な量子ビット数が 64 量子ビットとなる。本評価ではソルバーとして ‘DW_2000Q_VFYC.5’ を利用し、マシンパラメータを自動チューニング可能な ‘optimization’ オプションを設定した。D-Wave Ocean SDK はバージョン 1.6.0 を使用した。

スケーリング評価では、量子アニーリングと同様にイジングモデルを処理可能なアニーリングシミュレータとして、富士通デジタルアニーラ (DA) [10] と、OpenJij [11] を利用した。DA は、イジングモデルの処理アルゴリズムをデジタル回路で実装した専用マシン [12] で、Web API を介して利用する。本評価では、ソルバーとして一部のマシンパラメータを自動チューニングする ‘FujitsuDAPTSolver’ の Web API を利用した。利用可能なビット数は 1,024 bit であった。OpenJij は、オープンソースソフトウェアとして、量子アニーリングをシミュレートする Simulated Quantum Annealing (SQA) の Python 用 API を提供している。本評価ではバージョン 0.0.8 の SQA を利用した。DA および OpenJij はともに量子ビット間の相互作用を全結合で利用可能であり、QUBO をそのまま実行可能である。

スケーリング評価において VRP の既存手法と比較するために、全探索による厳密解手法と、ヒューリスティックな近似解手法を用いる。厳密解手法は Python でスクラッチ実装し、深さ優先探索によって各配送車が巡回する拠点の組合せを決め、その組合せから各配送車が巡回する順序を順列で洗い出す。近似解手法には Google OR-Tools [13] の VRP solver を Python で利用した。この VRP solver では動的計画法とツリー探索を組合せた局所探索法を採用し、同一の実行条件ではつねに同じ解を出力する。

OpenJij の SQA、厳密解手法、および近似解手法を利用した評価では、それらのプログラムを汎用のサーバで実行した。利用したサーバは Dell PowerEdge R410 であり、ハー

表 1 評価データにおける問題の規模 (V: 配送車数, P: 拠点数, S: 訪問順序の最大値)

Table 1 Scale of VRP problem in evaluations (V: vehicle numbers, P: location numbers, S: max values of visiting order).

	V	P	S	ビット
*	2	4	4	32
小規模	2	10	8	140
		11	9	176
		12	10	216
		13	10	260
中規模	4	19	10	760
		20	10	800
		21	10	840
		22	11	968
大規模	6	40	15	3,600
	10	160	40	64,000
	15	400	71	426,000

* 量子アニーリング評価用データ

ドウェア仕様は CPU: Xeon X5675 (3.06 GHz, 6 Cores) × 2, Memory: 96 GB, SAS-HDD: 1 TB × 4 (RAID 1), NIC: 1000Base-T で, ソフトウェア仕様は OS: CentOS 6.6, Python: 3.6.0 である.

3.3 評価データ

問題の規模として, 実際のデマンド交通サービスを想定し, 配送車数 V および拠点数 P から 3 通りの規模を用意した. 訪問順番の最大値 S は, 1 台ですべての拠点を巡回することを考慮する場合, Depot を 2 度訪問するため, $S = P + 1$ となる. しかし現実問題を考慮すると, 複数台で巡回することで公平性を保つ必要がある. また, DA はビット数に上限があり, S を減らして目的変数の数を減らすことで, 実行できる問題の規模が大きくなる. そこで, 1 台あたりの平均拠点数が 3 以上の場合に, 訪問順番の最大値 S を配送車数 V と拠点数 P を用いて以下のように設定した.

$$S = \begin{cases} \left\lfloor \frac{3P}{V+2} + 0.5 \right\rfloor, & \text{if } \frac{P}{V} \geq 3.0 \\ P, & \text{otherwise} \end{cases} \quad (17)$$

これをふまえ, 3 通りの問題の規模の評価データとして, 表 1 で示す V, P, S の値に設定した 12 種類のデータを用意した. あわせてアニーリングシミュレータで必要となる目的変数のビット数も記載する. なお, $V = 2, P = 4$ に関しては, D-Wave 2000Q による量子アニーリングの性能評価にのみ利用した.

VRP を解くためには, データとして拠点間の距離が必要である. 本評価では, 前述の各規模における拠点数に応じ

表 2 各アニーリングマシン実行時のペナルティ係数

Table 2 Penalty parameters of each annealing machine.

	λ_1	λ_2	λ_3	λ_4
D-Wave 2000Q	55	42	42	41
DA-小規模	750	750	875	750
DA-中規模	1000	1000	1250	1000
OpenJij	1500	1300	1230	1250

て, 拠点間の距離を生成した. まず, 各拠点の座標 $[X, Y]$ をランダムに生成する. 座標の X および Y の範囲は 0-1000 の整数とした. また, Depot の座標を $[500, 500]$ とした. このランダム生成した座標に対して, すべての 2 座標間のユークリッド距離を計算し, 拠点間の距離として利用した. なお, 量子アニーリングの評価で利用する D-Wave 2000Q は量子ビット間の相互作用の解像度が 5 bit に制限されているため, 量子アニーリング評価用データに限り, 拠点間の距離を 1/50 倍に再計算した.

これらの評価データをもとに, 2.3 節で述べたペナルティ係数のチューニングを行い, 各アニーリングマシン実行時のペナルティ係数を表 2 で示す値とした. 実行可能解を得る確率の目標値を 80% とし, 小規模の問題におけるチューニングは $V = 2, P = 10$ で, 中規模の問題は $V = 4, P = 19$ でを行い, 同規模の問題内では共通のペナルティ係数を利用した. D-Wave 2000Q は目標値に達せず実行可能解を得る確率の増加が止まり, DA と OpenJij は目標値に達している. OpenJij に関しては小規模と中規模の問題で同じペナルティ係数となった.

3.4 量子アニーリングの性能評価

D-Wave 2000Q を利用して, 量子アニーリングにおける VRP の性能を評価する. グラフ埋め込みにより最大 64 量子ビットまでの利用となるため, 本評価では配送車 2 台 4 拠点のデータを利用した. 実行時のパラメータとして, 実際に解を探索するアニーリング処理の時間を指定でき, 設定可能な最小値である 1μ 秒と, 10μ 秒で評価した. また, D-Wave 2000Q は 1 回の実行で複数回のアニーリング処理を実行できる. 本評価ではアニーリング処理回数を 100 回に設定した. これは 1 回の D-Wave 2000Q の実行で 100 個の解が得られることを意味する.

D-Wave 2000Q にて前述の環境で実行した結果, アニーリング処理の時間である 1μ 秒と 10μ 秒ともに厳密解を導出できることが確認できた. それぞれのアニーリング処理の時間における実行可能解を得る確率は 1μ 秒で 41%, 10μ 秒で 52% であった. したがって, アニーリング処理の時間を増やすことで, 解の安定性が増していることが分かる.

2 つのアニーリング処理の時間における実行時間とその内訳を図 3 に示す. 各実行時間における内訳の定義は以下

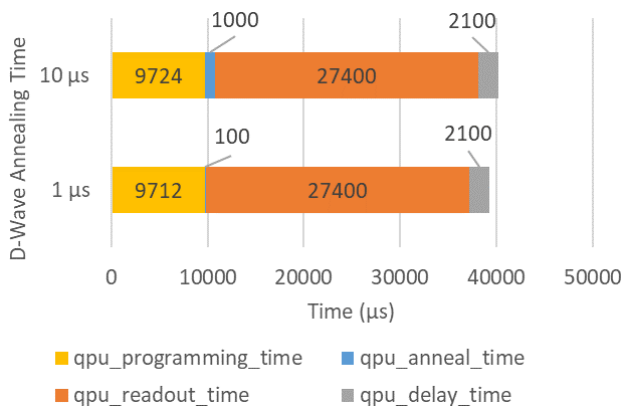


図 3 D-Wave 2000Q における実行時間

Fig. 3 Execution time on D-Wave 2000Q.

のとおり.

- qpu_programming_time: QUBO における量子ビットの外部磁場係数および量子ビット間の相互作用をデバイスに設定する時間
- qpu_anneal_time: アニーリング処理の合計時間
- qpu_readout_time: アニーリング処理結果読み取りの合計時間
- qpu_delay_time: アニーリング処理のためにデバイス環境の再設定にかかる合計時間

qpu_programming_time で計測される時間は D-Wave 2000Q 実行で 1 回のみ行われるが, qpu_anneal_time, qpu_anneal_time, qpu_delay_time はアニーリング処理ごとに計測された時間の 100 回の合計値を示している. qpu_anneal_time は実行時に指定したアニーリング処理 100 回分の時間となっており, その他の処理時間と比較して非常に短く, アニーリング処理以外が支配的である. 現在のマシンは, アニーリング処理の時間に対して, アニーリング以外の処理オーバーヘッドが相対的に大きいことが分かる. また, 1 μ 秒と 10 μ 秒の実行時間を比較すると, アニーリング処理の時間の比のとおり qpu_anneal_time で 10 倍の差があるが, qpu_anneal_time 以外の処理時間は, アニーリング処理の時間に依存せずほぼ固定の値となっている. したがって, 解の安定性を高めるためにアニーリング処理の時間を長くしても, マシン実行の全体の時間では大きく遅延しないため, スケーリングの観点で有利であるといえる.

3.5 スケーリング評価

量子アニーリングのマシンである D-Wave 2000Q は利用可能な量子ビットの限界からデマンド交通サービスを想定した規模のデータで評価することができない. 一方でアニーリングシミュレータであればより規模の大きいデータで実行できる. そこで本節では, アニーリングシミュレータである DA と OpenJij を用いて, VRP の従来手法と目的関数および実行時間を比較することで, より大きい規模

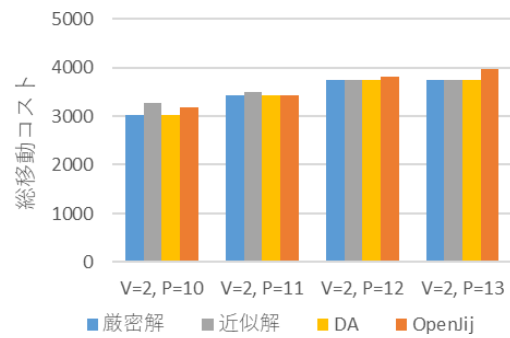


図 4 小規模な問題における総移動コスト

Fig. 4 Total travel cost in small scale.

の VRP における量子アニーリングの実用性を評価する.

本評価におけるアニーリングシミュレータの実行は, 量子アニーリングの評価と同様にアニーリング処理回数を 100 回に設定した. DA および OpenJij とともに, ペナルティ係数のチューニングにより実行可能解を 80 個程度得られる. VRP のアプリケーションではこのように得られる複数個の解のうち最小のものを解として利用することを想定し, 得られた実行可能解のうち目的関数が最小値となった解で評価を行う. 加えて, アニーリングシミュレータでは, 1 回のアニーリング処理の時間をステップ数として指定できる. ステップ数を増やすことで良い解が得られる確率を高めることができるが, 実行時間はステップ数と比例して長くなる. 本評価では, ペナルティ係数のチューニングで利用したステップ数を初期値に, データごとにステップ数を増加させてアニーリング処理 100 回を複数回実行し, 目的関数の最小値が変わらなくなった時点での最小のステップ数の結果を採用する. その実行における目的関数の最小値とアニーリング 100 回の実行時間を評価する. なお VRP の従来手法は得られる解が固定なので, 1 回の実行結果を利用する.

3.5.1 目的関数

VRP における目的関数の総移動コストを評価する. 図 4 が厳密解手法, 近似解手法, DA, OpenJij で小規模な問題を実行した結果である. 近似解および OpenJij の一部で厳密解より総移動コストが大きくなった. 一方, DA は厳密解と同じ結果となった. 図 5 は中規模な問題を実行した結果である. ここでは, 厳密解手法の実行が困難なため計測を行わず, 近似解手法, DA, および OpenJij で計測した. DA は近似解手法と比較して約 10%, 総移動コストが低い. OpenJij に関しては近似解手法と比較して同程度か, 約 5% 低下する結果であった. なお大規模な問題は, DA および OpenJij で実行ができなかったため, 省略する. この原因について, DA に関しては, ビット数が不足したため実行できず, OpenJij に関しては, 実行時間の観点で実行を断念した.

DA は目的関数を低い値に安定して収束した. OpenJij

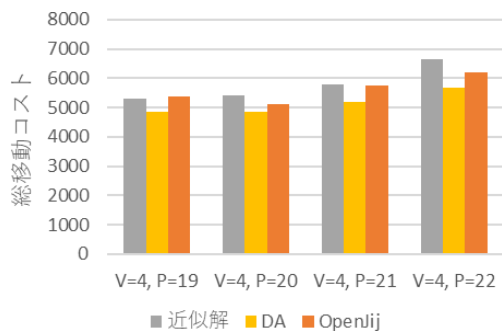


図 5 中規模な問題における総移動コスト
Fig. 5 Total travel cost in middle scale.

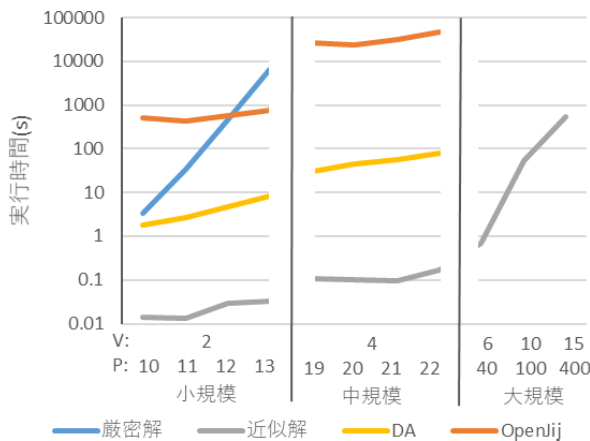


図 6 各マシンの実行時間 (対数表記)
Fig. 6 Execution time on each annealing machine.

は近似解と比較して目的関数の低い値を出すことが可能ではあるが、他の結果と比較して収束の安定性が低い。DA および OpenJij とともに、問題の規模の拡大とともに近似解に対する優位が大きくなっている。これらはシミュレータではあるが、イジングモデルに従って最良解に収束していく動作は量子アニーリングと同様であり、目的関数の観点では、量子アニーリングが近似解手法より優位となる可能性があるといえる。一方、現状ではシミュレータにおいても実行できる問題の規模に限界があることが分かる。

3.5.2 実行時間

VRP における実行時間を評価する。ここで利用した実行時間は、3.5.1 項における総移動コストを算出する時間である。なお、計測した時間は、各マシンにおけるイジングモデルの処理時間であり、DA の Web API における通信時間やキューの待ち時間は含まれていない。図 6 は、3 通りの問題の規模におけるすべてのデータで実行時間を計測した結果である。厳密解手法は小規模な問題のみ実行でき、DA と OpenJij は小規模と中規模の問題の両方を実行できた。近似解手法のみ、大規模な問題を実行できた。厳密解手法は、小規模で拠点数が少ない問題で OpenJij より高速だったが、規模に対する傾きが大きく、拠点数の増加で OpenJij の実行時間を超え、数時間単位の実行時間を要する結果となった。近似解手法、DA、および OpenJij は、

同一規模内における拠点数の増加において、同様の傾きとなっているが、OpenJij は小規模から中規模で大きく実行時間が増加した。DA は傾きに関して近似解手法と同様であるが、絶対値として 2 桁の差がある結果となった。

厳密解手法の結果と比較すると DA および OpenJij は規模の拡大に応じて大きく優位な結果となった。しかし、近似解手法と比較すると大きく劣後している。特に OpenJij は最大 5 桁の差がある。OpenJij はソフトウェアであり、現実装では逐次実行のため、ビット数の増加にともない指数的に実行時間が増加する。DA は専用のデジタル回路実装による並列実行の効果が大きく、OpenJij と比較して高速であるが、それでも近似解手法に比べて低速である。

4. 量子アニーリング活用に関する議論

3.4 節の量子アニーリングの性能評価の結果から、現在のマシンでは量子ビットの制約や実行時のオーバーヘッドの制約が強いことが分かった。しかしアニーリング処理自体は μ 秒単位と高速であり、マシン実装の進化によって適用可能な量子ビット数が大きくなることで、より効率的に VRP を解けることが期待できる。また、より大きい規模の問題を解くための方法として、VRP のビット数を減らすための定式化方法の考案と、キメラグラフへのグラフ埋め込み方法の考案が考えられる。前者の定式化方法に関しては、たとえば Depot に関する制約である式 (7), (8) を定数として設定することで、変数の削減が可能となる。この副次的な効果として、これらの制約式である式 (13), (14) を QUBO 式の式 (11) から削除でき、ペナルティ係数のチューニングの簡略化が期待できる。後者に関して、今回の評価でのグラフ埋め込みは、汎用的な方法としてシンプルにキメラグラフを全結合グラフへ変換している。問題設定として VRP の目的変数のグラフ構造を工夫することで、全結合への単純な変換をなくし、グラフ埋め込みの効率性を向上させ、利用可能な問題の規模を増やすことが可能となる。たとえば、物理的に距離が遠い拠点間のエッジの排除などにより、部分グラフを作成するなどが考えられる。このグラフ構造の工夫は、利用する物理的な量子ビット数が変わらないため、処理時間は変わらない。ただし問題の規模は大きくなるので、目的関数の精度のためにアニーリング処理の時間や回数を増やす必要がある。これに対しては、本評価からアニーリング処理の時間以外のオーバーヘッドは変わらないことが分かったため、今回の評価における全体の実行時間 40 ms に対して μ 秒単位での遅延で実行できると考えられ、グラフ構造の工夫の効果が期待できる。

スケーリングの観点では、3.5.1 項のアニーリングシミュレータによる目的関数の評価結果から、量子アニーリングの物理的な量子ビット数が増えた場合にも、精度の求められる VRP の実応用において量子アニーリングが優位となる可能性が示された。一方で、実行時間の観点では、ア

ニーリングシミュレータによる評価の中規模な問題において、DA では 10 秒オーダー、OpenJij では 10,000 秒オーダーとなり、近似解を求める従来手法の 0.1 秒オーダーと比較して非常に遅い。現段階では、実装上の限界によりこの規模での量子アニーリングの実行時間を定量的に評価することができないが、DA および OpenJij は目的関数の観点で優位性が見られたことから、量子アニーリングのマシンの進化によってこの実行時間の差が近づくことで、VRP における量子アニーリングの実用性は高まると考えられる。特に近似解手法は大規模な問題において大幅な遅延が見られたことから、シミュレータも含めて、量子アニーリングの大規模対応や、前述した VRP で利用する量子ビット数を減らす手法の適用により優位を示す可能性がある。

ここまでの評価結果をふまえ、VRP における量子アニーリングに関連するマシン活用の条件を整理する。量子アニーリングが実行可能な規模である場合には、目的関数の精度と実行時間の観点で量子アニーリングを活用すべきである。この実行可能な規模の範囲を広げるために、定式化やグラフ構造にさらなる工夫が必要である。それ以上の規模で実行する場合、利用すべきマシンは求められる性能と精度に依存する。Web サービスのように最適化要求に対してリアルタイムに解を求める場合は、現段階では従来手法のソルバーが適切である。目的関数の精度が求められ、バッチ的に最適化処理することが可能な場合は、DA のような比較的高速なアニーリングシミュレータが適切である。なお、OpenJij のようなソフトウェア実装のアニーリングシミュレータは汎用サーバで実行でき、比較的気軽に利用できるため、今回の評価で利用した VRP に対して様々な制約を追加する場合など、試験的にアニーリングを利用したい場合に活用の機会があると考えられる。

5. 関連研究

VRP の既存手法には、近似解手法として局所探索法、シミュレーテッドアニーリング、遺伝的アルゴリズムなどが、厳密解手法では数理計画法、分枝限定法などがあげられる。具体的には、VRP に容量制約を加えた Capacitated VRP (CVRP) と呼ばれる問題に関する研究 [14], [15] や、時間制約を加えた VRP with Time Window (VRPTW) と呼ばれる問題に関して研究されている [16], [17]。これらは、それぞれの問題に特化した手法であり、デマンド交通サービスなどの実应用到に即した制約を適用するにあたり、さらなる工夫が必要となる。この観点で、量子アニーリングで利用するイジングモデルは汎用的に組合せ最適化問題に適用でき、制約の追加を QUBO 式の制約項の追加などにより比較的容易に対応可能な点は大きな利点であるといえる。

量子アニーリングを活用して NP-Hard な組合せ最適化問題を解く研究が近年増加している [4], [18], [19]。特に VRP に量子アニーリングを適用する研究として、バスなどの車

両によってお客様を配送する最適化問題として CVRP に適用する研究 [20] や、物流をユースケースとして CVRP に時間と車の状態の制約を追加した問題に適用する研究 [21] がある。これらの研究では、VRP の基礎的な条件下における評価や、デマンド交通サービスを想定してスケールアップさせたデータで評価していない。

6. おわりに

本論文では、制約を限定した基礎的な VRP の QUBO を構築し、量子アニーリングのマシンである D-Wave 2000Q で性能評価を、2 つのアニーリングシミュレータ：DA, OpenJij で量子アニーリングのスケールアップ評価を行った。量子アニーリングの性能評価により、40 ms で最適解を導出できることを確認した。アニーリング処理自体は μ 秒単位で解を導出できたが、現在のマシンでは実行可能な規模が小さく、またアニーリング以外の処理におけるオーバーヘッドが相対的に大きい。スケールアップ評価では、デマンド交通サービスを想定した規模で、DA および OpenJij と、VRP の従来手法として厳密解手法および近似解手法と比較し、目的関数の観点で近似解手法より最大 10% 精度向上することを示した。一方で実行時間の観点では、近似解手法に 2 桁以上の差で劣後していた。これらの評価結果をふまえ、実利用を想定した VRP において量子アニーリングを活用するための課題とマシン活用の条件を整理した。

現段階でも中規模程度の問題であればアニーリングシミュレータにおいてデマンド交通サービス適用に優位な可能性があり、実装技術の進化とともにその可能性はより高まると考えられる。この可能性を高めるため、活用側の観点で、量子ビット数を減らすための定式化手法や、グラフ構造の変換手法を検討したい。また、デマンド交通サービスへの適用をふまえて制約条件を追加した場合、近似解手法の Google OR-Tools のように限定的な制約の VRP に特化している手法をそのまま利用できなくなるため、汎用的に利用できる量子アニーリングが優位になりえると考えられる。そのため、現実的な問題設定として制約を追加した一般問題である CVRP や VRPTW への QUBO 適用、およびそのほかのサービス実現条件を組合せた問題の QUBO 式の構築および評価を行っていきたい。

参考文献

- [1] Dantzig, G.B. and Ramser, J.H.: The Truck Dispatching Problem, *Management Science*, Vol.6, No.1, pp.80–91 (1959).
- [2] Kadowaki, T. and Nishimori, H.: 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).
- [3] Johnson, M.W. et al.: Quantum annealing with manufactured spins, *Nature*, Vol.473, No.7346, pp.194–198 (2011).

- [4] Rieffel, E.G., Venturelli, D., O’Gorman, B., Do, M.B., Prystay, E.M. and Smelyanskiy, V.N.: A case study in programming a quantum annealer for hard operational planning problems, *Quantum Inf. Process.*, Vol.14, No.1 (2014).
- [5] Choi, V.: Minor-embedding in adiabatic quantum computation: I. the parameter setting problem, *Quantum Inf. Process.*, Vol.7, No.5, pp.193–209 (2008).
- [6] Bian, Z., Chudak, F., Israel, R., Lackey, B., Macready, W.G. and Roy, A.: Discrete optimization using quantum annealing on sparse Ising models, *Front. Phys.*, Vol.2, No.56, pp.1–10 (2014).
- [7] Lucas, A.: Ising formulations of many NP problems, *Front. Phys.*, Vol.2, No.5, pp.1–14 (2014).
- [8] D-Wave 2000Q, available from (<https://www.dwavesys.com/d-wave-two-system>) (accessed 2020-06-20).
- [9] D-Wave Ocean SDK, available from (<https://github.com/dwavesystems/dwave-ocean-sdk>) (accessed 2020-06-20).
- [10] 富士通デジタルアニーラ, 入手先 (<https://www.fujitsu.com/jp/digitalannealer/>) (参照 2020-06-20).
- [11] OpenJij, available from (<https://github.com/OpenJij/OpenJij>).
- [12] Tsukamoto, S., Takatsu, M., Matsubara, S. and Tamura, H.: An accelerator architecture for combinatorial optimization problems, *Fujitsu Sci. Tech. J.*, Vol.53, No.5, pp.8–13 (2017).
- [13] Google OR-Tools, available from (<https://developers.google.com/optimization>) (accessed 2020-06-20).
- [14] Pecin, D., Pessoa, A., Poggi, M. and Uchoa, E.: Improved branch-cut-and-price for capacitated vehicle routing, *Mathematical Programming Computation*, Vol.9, No.1, pp.61–100 (2017).
- [15] Mohammed, M.A. and Ghanil, M.K.A.: Solving Vehicle Routing Problem by Using Improved K-Nearest Neighbor Algorithm for Best Solution, *J. Comput. Sci.*, Vol.21, pp.232–240 (2017).
- [16] Pecin, D., Contardo, C., Desaulniers, G. and Uchoa, E.: New enhancements for the exact solution of the vehicle routing problem with time windows, *INFORMS Journal on Computing*, Vol.29, No.3, pp.489–502 (2017).
- [17] Ropke, S. and Pisinger, D.: An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows, *Transportation Science*, Vol.40, No.4, pp.455–472 (2006).
- [18] Tran, T.T. et al.: A hybrid quantum-classical approach to solving scheduling problems, *Proc. 9th Annual Symposium on Combinatorial Search*, pp.98–106 (2016).
- [19] Smelyanskiy, V.N. et al.: A Near-Term Quantum Computing Approach for Hard Computational Problems in Space Exploration (2012), available from (<http://arxiv.org/abs/1204.2821>).
- [20] Feld, S. et al.: A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer, *Front. ICT*, Vol.6, No.13, pp.1–13 (2019).
- [21] Irie, H., Wongpaisarnsin, G., Terabe, M., Miki, A. and Taguchi, S.: Quantum Annealing of Vehicle Routing Problem with Time, State and Capacity, *International Workshop on Quantum Technology and Optimization Problems*, pp.145–156 (2019).



齋藤 和広 (正会員)

2010年成蹊大学大学院工学研究科博士課程前期修了。同年 KDDI 株式会社入社。2017~2018年カーネギーメロン大学客員研究員。現在、株式会社 KDDI 総合研究所統合機械学習グループ研究主査。この間、分散システム、データベースシステム、データ統合、量子アニーリング応用等の研究開発に従事。日本データベース学会会員。



大山 重樹

1994年東北大学大学院工学研究科修士課程修了。1996年第二電電株式会社(現 KDDI 株式会社)入社。現在、株式会社 KDDI 総合研究所人間協調 AI グループ研究員として位置情報 AI 研究開発に従事。



梅木 智光

2010年新潟大学大学院自然科学研究科数理・情報電子工学専攻博士前期修了。同年 KDDI 株式会社入社。現在、KDDI 株式会社技術統括本部技術企画本部技術戦略部渉外グループ課長補佐。電子情報通信学会会員。



黒川 茂莉 (正会員)

2007年慶應義塾大学大学院理工学研究科修了後、KDDI (株) 入社。以降、(株) KDDI 総合研究所にて機械学習、転移学習、データマイニング、位置情報分析等の研究に従事。現在、同社統合機械学習グループ研究マネージャー。電子情報通信学会会員。



小野 智弘 (正会員)

1994年慶應義塾大学大学院理工学研究科修了後、国際電信電話(株)入社。1999～2000年スタンフォード大学客員研究員。2004～2008年産業技術総合研究所協力研究員。現在、(株)

KDDI総合研究所データインテリジェンス部門長。この間、利用者嗜好抽出、ソーシャルメディア解析、顧客行動分析、行動変容等の研究に従事。電子情報通信学会(シニア)会員、博士(工学)。

(担当編集委員 田島 敬史)