Digital Annealer for High-Speed Solving of Combinatorial Optimization Problems and Its Applications

Satoshi Matsubara, Motomu Takatsu, Toshiyuki Miyazawa, Takayuki Shibasaki,
Yasuhiro Watanabe, Kazuya Takemoto and Hirotaka Tamura
Technology Development Project, Digital Annealer Unit
Fujitsu Laboratories LTD.
Kawasaki, Kanagawa 211-8588
Tel : +81-44-754-2049
Fax : +81-44-754-2723
e-mail : s-matsubara@fujitsu.com

Abstract - A Digital Annealer (DA) is a dedicated architecture for high-speed solving of combinatorial optimization problems mapped to an Ising model. With fully coupled bit connectivity and high coupling resolution as a major feature, it can be used to express a wide variety of combinatorial optimization problems. The DA uses Markov Chain Monte Carlo as a basic search mechanism, accelerated by the hardware implementation of multiple speed-enhancement techniques such as parallel search, escape from a local solution, and replica exchange. It is currently being offered as a cloud service using a second-generation chip operating on a scale of 8,192 bits. This paper presents an overview of the DA, its performance against benchmarks, and application examples.

I. Introduction

As Moore's Law, which has held for over 50 years, is approaching its end, the performance improvement of general-purpose processors is slowing down [1]. Thus, domain specific computing that improves the performance by applying dedicated hardware and algorithms to specific application domains is gaining attention. If a dedicated system is tailored to a particular application, however, even the slightest change in the application could lead to a system needing to be redesigned. Thus, a market must be found that satisfies contradictory demands: having a compelling range of applications while allowing specialization to achieve significant performance gain. An application domain having such a possibility is combinatorial optimization. The combinatorial optimization problem is a decision-making problem to find the best possible combination under various constraints. Many problems come down to combinatorial optimization problems, ranging from routine decisions such as schedules and the shortest route for automotive navigation systems to social issues such as disaster recovery plans and elimination of urban congestion. However, many combinatorial optimization problems belong to the complexity class of NP-hardness, and the calculation time increases exponentially as the problem size increases. Therefore, many heuristic algorithms have been investigated to obtain suboptimal solutions efficiently considering the characteristics of each problem, and promising solutions have been obtained for some problems.

Various optimization methods using the Ising model have recently emerged as meta-heuristics that can universally adapt to combinatorial optimization problems and are used as a core of the system to solve those problems. The optimization methods using the Ising model are categorized into annealing and nonlinear oscillation methods. The annealing methods are categorized into quantum or classical computation, and the nonlinear oscillation methods are categorized as having physical devices or not.

The quantum annealing is guaranteed to provide a solution in ideal situations, but its operation requires sensitive hardware control. D-Wave Systems [2] commercialized hardware based on quantum annealing for the first time. However, there are structural restrictions on the number of couplings between qubits [3]. CMOS annealing [4] performs simulated-annealing (SA) [5] like operation with controlled injection of numerical noise to escape from local minima. Coherent-Ising Machine [6] and Simulated Bifurcation [7] are based on nonlinear oscillation and have fully coupled connectivity, making problems to be solved easily mapped to the Ising model.

From an application-driven viewpoint, the Digital Annealer (DA) [8] was developed as a completely new computer architecture that can be implemented in digital circuit. Since the coupling resolution between all coupled bits can be set in 16 bits (65,536 gradations), the DA can handle the actual combination optimization problem formulated in the Ising model. A cloud service using the first generation of the DA supporting 1,024 binary variables (1,024 bit scale) has been launched since 2018, and the bit scale has been expanded up to 8,192 in the second generation.

In the following, we describe the operating principle of the DA in chapter II, evaluation result of the second-generation DA in chapter III, and application examples in chapter IV.

II. Operating Principle

A. Ising energy function search method

The DA searches the lowest value of the energy function expressed in binary quadratic form on the basis of the Markov Chain Monte Carlo (MCMC) method. The energy function is expressed as an Ising model as follows:

$$E(X) = -\sum_{i,j} W_{ij} x_i x_j - \sum_i b_i x_i \quad (1)$$

$$x_i \in \{0,1\}\ (i = 1,2,\dots,N), W_{ii} = 0, W_{ij} = W_{ji}$$

where $X = (x_1, x_2, \dots, x_N)$ is a state which is a set of binary bits, W_{ij} is the coupling coefficient between bit i and bit j, and b_i is the bias term for bit i. With fully coupled connectivity and high resolution of coupling coefficient, a wide variety of combinatorial optimization problems expressed in

978-1-7281-4123-7/20/$31.00 ©2020 IEEE.

Ising models can be easily mapped to the DA.

B. *Speed-up by parallel search*

The operation process of the DA is divided into two phases: a trial phase in which the bit flip satisfying the acceptance criterion is selected, and an update phase in which the selected bit is flipped.

In the trial phase, one of bits is chosen from the current state X, and the energy change for flipping the bit is calculated. The energy change ΔE when the current bit value x_i is flipped to $x_i'(= 1 - x_i)$ can be written as follows.

$$\Delta E_i = E(X)|_{x_i \to x_i'} - E(X) = (x_i - x_i')\left(\sum_j W_{ij}\, x_j + b_i\right) \quad (2)$$

$$A(\Delta E_i) = \min[1, \exp(-\beta \cdot \Delta E_i)] \quad (3)$$

This energy change is used to determine whether to accept the bit flip in accordance with Metropolis-Hastings criterion. Here, $A(\Delta E_i)$ represents the flip acceptance probability for the energy change ΔE_i, and β represents the inverse temperature. A bit flip candidate is extracted by comparing the value of ΔE_i with a random number for each state change.

In the update phase, the flip-bit selector selects one bit to be flipped and updates the value of x_i. If there is no candidate to update, the update selector outputs a flag value of "0" and returns to the trial phase. The above processes repeat by the set number of iterations, and the series of processing ends.

Figure 1 shows a schematic diagram of the parallel search technique. In the DA, the trial phase of the above process is performed in parallel, so that the probability of transition to a new state increases at most N times as compared with a serial search.

C. *Escape technique from local minimum*

If a state falls into a local minimum state of an Ising energy function, the probability of escaping from it and transitioning to another state can be much lower even with parallel search. In this case, the state stays at the same local minimum state for many cycles, and the time to reach the global minimum state increases.

The DA is equipped with a technique to reduce the time spent in a local minimum state. If no bit flip candidate is found, the escape from a local minimum state is facilitated by adding the positive offset E_{off} to the energy. This is approximately equivalent to multiplying the common factor $\exp(\beta * E_{off})$ by the acceptance probability of a bit flip. In this technique, the value of the offset E_{off} is dynamically controlled so that the probability of finding the next bit flip becomes "1," and the time staying at local minimum states is shortened (Fig. 2).

The number of useless stays in a state is reduced by the parallel search technique and the escape technique from local minimum states, and the speed of the global minimum state search is improved in comparison with conventional SA [5] (Fig. 3).

D. *Replica exchange method*

In stochastic search, there are various speedup methods using multiple replicas defined as MCMC search processes specified by state variables and temperature. One of the simplest methods is a simple parallel annealing method that gives the same problem to multiple replicas and performs statistically independent searches. Here, an annealing method is a MCMC method with a temperature schedule. When the lowest energy state obtained by multiple replicas is used as a solution, the target accuracy rate of each search may be small. Therefore, the time required to obtain the lowest energy state can be expected to be shortened.

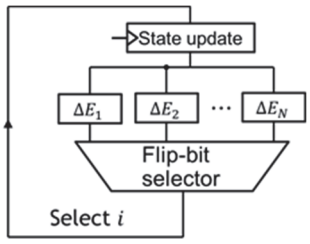


Fig. 1. A diagram of the parallel search technique

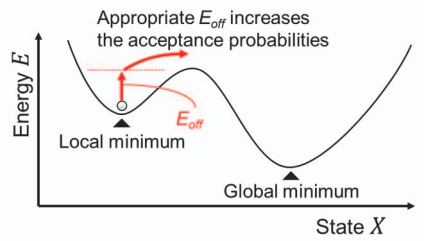


Fig. 2. A diagram of the escape technique

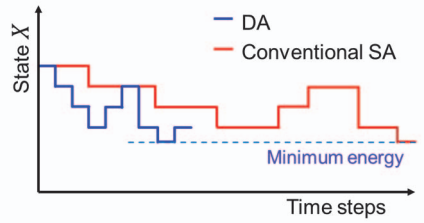


Fig. 3. Conceptual diagram of speed-up achieved by DA

On the other hand, the replica exchange method is an algorithm for promoting thermal equilibrium by performing solution search by the MCMC method using multiple replicas having different temperatures and further exchanging the replicas [9] (Fig. 4). In the replica exchange method, K replicas having different temperatures ($T_1 > T_2 > \ldots > T_K$) are prepared, and stochastic search is performed for each. States are exchanged between replicas at adjacent temperatures under the Metropolis rule. Exchange transition probability between replica index n and replica index $n+1$ can be calculated with temperature difference and energy difference in Eq. (4) :

$$P_{n,n+1} = \min(1, \exp(\beta_n - \beta_{n+1})\,(E(X_n) - E(X_{n+1})) \quad (4)$$

where n is replica index, β_n is inverse temperature of replica n, X_n is a set of states of replica n, and $E(X_n)$ is energy of replica n. By exchanging states, a path can be established between the lower temperature replica and the higher temperature one, thereby improving convergence degradation due to staying at a local minimum. The DA implements both simple parallel annealing method and replica exchange methods.

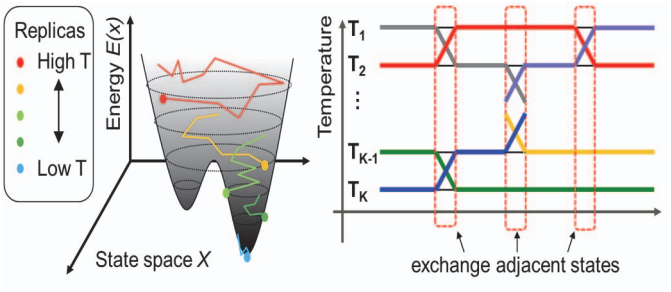


Fig. 4. Replica Exchange technique

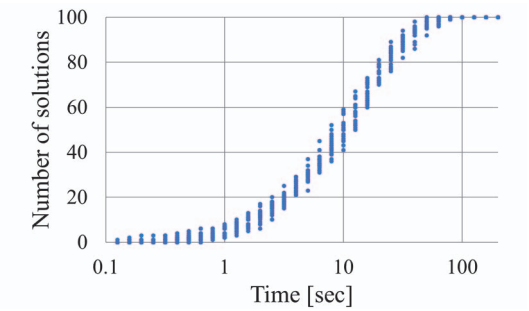


Fig. 5. The number of replicas reaching the best-known solution, when the DA solved MaxCut problem G53.

E. *Behavior of solution search by DA*

This section provides an example of how the DA equipped with the above techniques behaves in the properties of solution search. Figure 5 shows the experimental results of the number of replicas that reached the best-known solution over time, when the DA in the replica exchange mode solved a problem 20 times with different seeds of the random number. The problem instance was G53 from Gset [10] which was a standard maximum cut problem (MaxCut) library. The number of replicas was 100. In the experimental results, the number of replicas that reached the best-known solution increased over time, with all of them eventually reaching it. This result shows that the behavior of the solution search by DA is almost independent of the seed, and the probability of reaching the optimal solution also increases over time.

III. Evaluation Results

The second-generation Digital Annealer, the Digital Annealing Unit (DAU) implements techniques described in Chapter II on a dedicated chip (ASIC). Compared with the first generation, the DAU has been extended from 1,024 to 8,192 in bit scale and from 16 bits to a maximum of 64 bits in coupling resolution.

Here, we present the DAU evaluation results for three examples of combinatorial optimization problems: MaxCut, minimum cut problem (MinCut), and quadratic assignment problem (QAP). We compared the DAU with a general-purpose solver CPLEX and a state-of-the-art dedicated solver for each problem.

Since the performance of an exact solver like CPLEX and that of heuristic like the DAU cannot be compared exactly, we took a compromise commonly used in annealing hardware: defining "Time to Solution" as a computation time to achieve the correct or the best-known solution with a probability of 99%. "Time to Solution" is originally defined when multiple correct solutions are obtained, but to obtain a comparable time, we calculate it as the time to reach the minimum energy obtained even if no correct solution is obtained.

A. *Maximum cut problem*

MaxCut is a common combinatorial optimization problem, that divides the vertices of an undirected graph into two subsets and maximizes the sum of the weights of the edges between the vertices belonging to different subsets. MaxCut can be formulated as Eq. (5) [11] as follows:

$$E(X) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} e(i,j)\ \ (x_i + x_j - x_i x_j) \tag{5}$$

where x_i is a state variable for vertex i that becomes "1" when the vertex i belongs to the subset 0 and "0" when it does not, $e(i,j)$ is interaction weight between bit i and bit j.

The problems used in this evaluation are 65 instances of the standard problem Gset [10] G1 (800-bit scale) through G65 (8000-bit scale). For benchmarking, we compared the evaluation results of the DAU with those of CPLEX and the multiple operator heuristic (MOH) in previous papers. MOH is a MaxCut dedicated algorithm developed by Ma and Hao [12] that has the highest performance we know of in terms of solution quality and speed. The quality of the solution was compared by using the number of cuts of the solution, and the speed was compared by using the time to reach the solution in the instances in which the solution agreed. For the MOH, we substituted the time when the solution is obtained with a probability of 50% as described by Ma and Hao.

The results of 18 instances from 2K to 8K bit scale are shown in TABLE I. The red numbers mean that the solution reaches the best-known solution. DAU reached the best-known solution in 62 of 65 instances, which is more than CPLEX or MOH. Besides, the DAU updated the best-known solutions in three instances: G58, G61 and G63.

DAU performs better than CPLEX [13] except for two instances: G62 and G65. The error of the CPLEX solution to the best-known solutions varies greatly depending on the problem and was more than 25% in some problems. On the other hand, the quality of solutions obtained by DAU is always very stable with less than 0.1% error.

When comparing the speeds, the DAU was faster in 52 of 65 instances even under unfavorable conditions for the DAU, where the DAU uses the time to obtain a solution with 99% probability and MOH uses that with 50% probability.

B. *Minimum cut problem*

MinCut is a combinatorial optimization problem commonly used in the very large scale integration (VLSI) design field that divides the vertices of an undirected graph into two equally sized subsets and minimizes the sum of the weights of the edges between the vertices belonging to different subsets.

With the state variable $x_i \in \{0,1\}$, the MinCut can be formulated as Eq. (6) [11] as follows:

TABLE I
MaxCut benchmark results

Problem		Quality of Solution			Time to Solution (sec)		
Instance	|V|	DAU	CPLEX	MOH	DAU	CPLEX	MOH
G22	2000	**13359**	12843	**13359**	0.6	20003	352
G27	2000	**3341**	2446	**3341**	0.4	20000	42
G32	2000	**1410**	**1410**	**1410**	9	59	66
G33	2000	**1382**	**1382**	**1382**	1577	38	504
G34	2000	**1384**	**1384**	**1384**	4	234	84
G35	2000	**7687**	7389	**7687**	142	20000	797
G39	2000	**2408**	2045	**2408**	5	10001	788
G55	5000	**10299**	9780	**10299**	4700	20001	1230
G56	5000	**4017**	3661	4016	1381	20004	990
G57	5000	**3494**	**3494**	**3494**	49127	2118	1528
G58	5000	**19294**	18753	19288	13526	20000	1522
G59	5000	6086	5150	**6087**	31336	20000	2499
G60	7000	**14190**	13589	**14190**	9831	20000	2945
G61	7000	**5799**	5237	5798	893	20000	6603
G62	7000	4868	**4872**	4868	34462	5475	5569
G63	7000	**27052**	26246	27033	12195	40001	6492
G64	7000	**8751**	7340	8747	94891	1E+05	4011
G65	8000	5556	**5562**	5560	24099	12288	4710

Note) Instances in this table are the one that exist in the CPLEX benchmark [13] whose bit scale is 2K or larger.

$$E(X) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} e(i,j)\left(x_i + x_j - x_i x_j\right) + p(n_0 - \sum_{i=1}^{n} x_i) \quad (6)$$

where x_i is a state variable that becomes "1" when the vertex i belongs to the subset 0 and "0" when it does not. p is a penalty coefficient for constraint violation, and n_0 is a vertex count constraint belonging to the subset 0.

We used the test graphs in the Graph Partitioning Archive [14] which is dedicated to documenting the best quality solutions found to date for a set of benchmark graphs, and tree graphs that we created by emulating the circuits graph for the performance evaluation.

TABLE II shows the evaluation results using the Graph Partitioning Archive, including a comparison with a previous study [15]. The execution time is limited to 60 seconds for the DAU in this experiment. The DAU obtained high quality results, in all cases having a cut size equal to or smaller than the solutions from the other solvers: METIS [16], KaHIP [17] and QBSolv [18] [19]. The DAU solutions matched the best-known solution for the last two graphs. The experimental results show that the DAU performs as effectively as a high-precision MinCut solver.

Next, we evaluated the performance in tree graphs assuming the problem of element placement in Electronic Design Automation (EDA) as a more practical application of MinCut. After generating tree graphs with a fixed fanout, the density is adjusted by randomly adding branches between vertices.

We compared the performances of the DAU and METIS [16] with five instances of 8,192-bit scale tree graphs. METIS is a set of highly evaluated serial programs for partitioning graphs. METIS using the multilevel recursive bisection paradigm ran on a single-core Xeon processor clocked at 3.6 GHz.

TABLE II
MinCut benchmark results

Problem			Quality of Solution			
Instance	|V|	Best-known	DAU	METIS [16]	KaHIP [17]	QBSolv [18] [19]
add20	2395	596	603	723	760	647
data	2851	189	**189**	225	221	191
3elt	4720	90	**90**	91	92	**90**

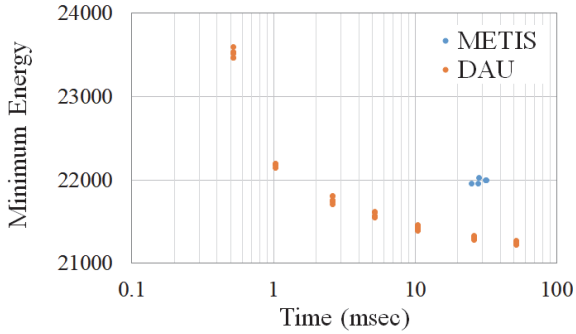


Fig. 6. Behavior of minimum energy when 8K-bit scale tree graph MinCut problems are solved by DAU and METIS.

Multiple runs with the same seed of the random number were performed for the same problem with different execution times in order to evaluate the behavior of the quality of the solution to execution time. As shown in Fig.6, the DAU reached the solution equivalent to that of METIS 18 times faster than METIS and finally reached a solution 4.0% better than that of METIS. By adjusting the execution time, the DAU can obtain a METIS-quality solution at a higher speed or a higher-quality solution at a longer time.

C. *Quadratic assignment problem*

QAP is the problem of minimizing the sum of the product of the amount of flow and distance for n factories and n locations, given the amount of flow and the distance between factories. QAP can be formulated as Eq. (7) [20] as follows:

$$E(X) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\sum_{l=1}^{n} f_{ij} d_{kl} x_{ik} x_{jl} + p_1 \sum_{i=1}^{n}(1 - \sum_{k=1}^{n} x_{ik})^2 + p_2 \sum_{k=1}^{n}(1 - \sum_{i=1}^{n} x_{ik})^2 \quad (7)$$

where f_{ij} is the flow of supplies moving from factory i to factory j, d_{kl} is the distance between location k and location l, and p_1 and p_2 are penalty coefficients for constraint violation. x_{ik} is a state variable that becomes "1" if factory i is placed at location k and "0" otherwise.

We compared the DAU results against published benchmark results with a general-purpose solver CPLEX [21] and with

TABLE III
QAP Benchmark result

Problem			Time to Solution (sec)		
Instance	|V|	Density	DAU	CPLEX [21]	EO-QAP [22]
had12	144	100.0%	3.392	24444.5	120.000
rou12	144	98.5%	0.448	11295.8	0.013
tai12a	144	97.0%	0.142	3876.7	0.011
nug12	144	68.2%	0.024	3955.3	0.012
scr12	144	42.4%	0.008	20.2	0.006
chr12a	144	15.4%	0.014	1.2	0.011

state-of-the-art dedicated heuristic solver EO-QAP [22] using the standard instance sets of QAP in QAPLIB [23].

The quality of these solutions is the same because the three solvers including DAU, have reached the best-known solutions for all problem instances. Regarding speed, the DAU is about four orders of magnitude faster than CPLEX [21] and on par with EO-QAP for the instances with 144 nodes (TABLE III). Benchmark results show that the DAU performs as well as the state-of-the-art dedicated solver in terms of speed and solution quality.

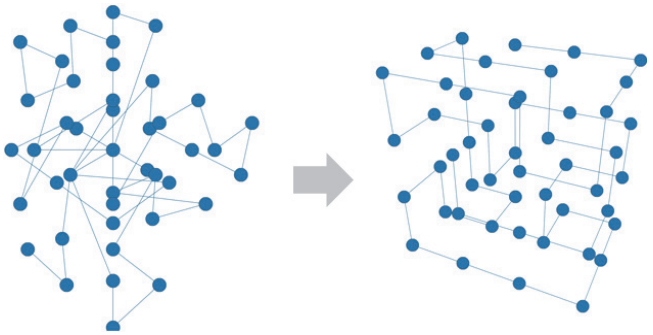
IV. Application Examples

In this chapter, we introduce two examples of the application of the DA to real world problems: search for drug candidates in the field of middle-molecule drug discovery, and optimization of collection and delivery routes at post offices.

A. Application to middle-molecule drug discovery

In the field of drug discovery, middle-molecular drugs with a molecular weight of about 500 ~ 2000 have recently attracted attention. Middle-molecule drugs have a structure in which a few dozen amino acids are linked in a chain and have high target selectivity and fewer side effects than small-molecule drugs with a molecular weight of 500 or less, so they are expected to be next-generation drugs. However, a simulation technique for determining the stable structure of a middle-molecule drug candidate has not yet been established. Here, we show an application example of the DA to a stable structure search for the middle-molecule drug candidate having 48-amino acids.

In this simulation, the linear molecular structure of amino acids is simplified by replacing one amino acid with one pseudo-atom, and the stable structure search based on the Lattice Protein Model [24] is performed. When this model is formulated in the Ising model, W_{ij} and b_i are determined on the basis of the interaction strength between each amino acid in the amino acid sequence. In the formulation, an about 31,000-bit scale is required for a 48-residue amino acid. Since the bit scale of this problem is larger than 8,192 in the DAU, we applied the software dividing technique. The optimization results are shown in Fig. 7. The processing time including the division processing is about 1 minute, which corresponds to a speedup of about 100 times compared with the case where a similar simulation is carried out by a conventional computer [25]. In the actual simulation, docking simulation to examine the binding strength between the searched amino acid structure and the target protein is repeated about 1000 times. This result shows that the search time for the middle-molecule drug candidate, which takes about 6 months by the conventional method, can be shortened to several days.



(a) Early stage of optimization (b) End stage of optimization

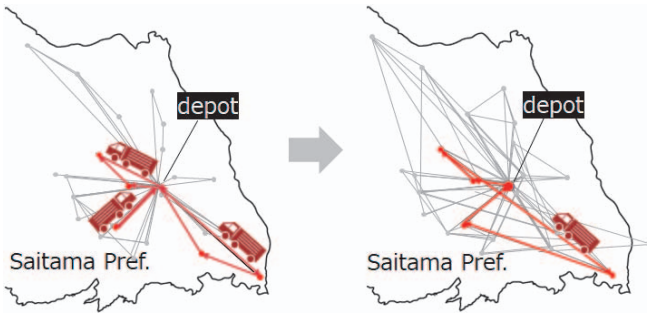
Fig. 7. DA-optimized structures of amino acid linear molecules.

B. Application to delivery planning

The delivery planning problem is very difficult to solve because it involves various problems settings and constraints such as the amount of cargo at each customer site, the load of trucks, delivery time and order restrictions, and total travel distance and time. Here, we introduce an example of the transportation network optimization in a post office as an application of the DA to delivery planning. This case study was conducted by A*QUANTUM (A STAR QUANTUM) in Japan Post's Open Innovation Program "POST LOGITECH INNOVATION PROGRAM 2018."

The problem addressed in this program is a vehicle routing problem (VRP) that minimizes delivery costs between the customer site and the pickup/delivery destination in the eastern Saitama area, which has around 30 post offices. The mandatory requirement for this problem is that trucks departing from the customer site load/unload at the pickup/delivery destination within a specified time range, and that the delivery demand of all the pickup/delivery destinations is satisfied. Here, factors such as the type and number of trucks used in the plan and the operation time determine the delivery cost.

The routes used in the distribution plan according to the conventional solution and the solution obtained in the DA are illustrated in Fig. 8. Compared with the conventional solution, the solution obtained by the DA improves the loading ratio of trucks by 12% and can reduce the number of vehicles required



(a) Conventional Route (b) DA-optimized Route

Fig. 8. Optimized routes of the delivery planning problem.

to satisfy all the loads from 52 to 48. Moreover, the solution obtained by the DA can be expected to reduce delivery cost by 7% compared with the conventional solution by reducing the number of trucks.

V. Conclusions

In this paper, we introduced the operating principle of the Digital Annealer (DA), the evaluation results of the second-generation DA in some benchmark problems, and application examples.

The DA accelerates Markov Chain Monte Carlo by the hardware implementation of multiple speed-enhancement techniques such as parallel search, escape from a local solution, and replica exchange. Since the second-generation DA can handle problems up to 8,192-bit scale with fully coupled connectivity and high coupling resolution, the DA can be applied to a wide variety of combinatorial optimization problems.

Benchmark results revealed that the DA performed comparably or superiorly to the general-purpose solver CPLEX and the state-of-the-art dedicated solvers in terms of speed and solution quality. Although the DA is a generally applicable architecture for solving combinatorial optimization problems mapped to an Ising model, it performs comparably to dedicated heuristic solvers in each application.

Since combinatorial optimization problems in the real world involve complicated constraints, they are difficult to solve with a dedicated solver. Therefore, the versatility and high speed of the DA should be effective in solving various real-world problems. The DA is currently being offered as a cloud service and has already been applied to problems in the real world such as drug discovery and delivery planning. Although we need to develop more technologies to expand the application fields of the DA, it will lead to more solutions to real-world problems in the future.

References

[1] R. Colwell, "The Chip Design Game at the End of Moore's Law," *Hot Chips* 27, 2015.

[2] https://www.dwavesys.com/press-releases/d-wave-previews-next-generation-quantum-computing-platform

[3] Paul. I. Bunyk, Emile M. Hoskinson, Mark W. Johnson, E. Tolkacheva, F. Altomare, Andrew. J. Berkley, R. Harris, Jelemy. P. Hilton, T. Lanting, Anthony J. Przybysz, and J. Whittaker, "Architectural Considerations in the Design of a Superconducting Quantum Annealing Processor," *IEEE Transactions on Applied Superconductivity*, Vol. 24, p. 1, August, 2014.

[4] T. Takemoto, M. Hayashi, C. Yoshimura, and M. Yamaoka, "A 2×30k-Spin Multichip Scalable Annealing Processor Based on a Processing-In-Memory Approach for Solving Large-Scale Combinatorial Optimization Problems," *ISSCC*, pp. 52–54, 2019.

[5] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, New Series, Vol. 220, pp. 671-680, May, 1983.

[6] T. Inagaki, Y. Haribara, K. Igarashi, T. Sonobe, S. Tamate, T. Honjo, A. Marandi, Peter. L. McMahon, T. Umeki, K. Enbutsu, O. Tadanaga, H. Takenouchi, K. Aihara, K. Kawarabayashi, K. Inoue, S. Utsunomiya, and H. Takesue, "A coherent Ising machine for 2000-node optimization problems," *Science* 04 Nov 2016: Vol. 354, pp. 603-606, 2016.

[7] H. Goto, K. Tatsumura and A. R. Dixon, "Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems," *Science Advances* Vol. 5, eaav2372., 2019.

[8] https://www.fujitsu.com/digitalannealer/

[9] K. Hukushima and K. Nemoto, "Exchange Monte Carlo method and application to spin glass simulations," *Journal of the Physical Society of Japan*, Vol. 65, pp. 1604-1608, June, 1996.

[10] http://www.stanford.edu/~yyye/yyye/Gset

[11] A. Lucas, "Ising formulations of many np problems," *Frontiers in Physics*, Vol. 2, February, 2014.

[12] Fuda Ma and Jin-Kao Hao., "A multiple search operator heuristic for the Max-k-cut problem," CoRR abs/1510.09156.

[13] T. Ikuta, H, Imai, and Y. Yano, *Research Institute for Mathematical Sciences*, Vol. 1941, pp. 49–56, 2015. (in Japanese).

[14] http://chriswalshaw.co.uk/partition

[15] H. Ushijima-Mwesigwa, Christian F. A. Negre, and Susan M. Mniszewski, "Graph Partitioning using Quantum Annealing on the D-Wave System," arXiv:1705.03082, 2017.

[16] http://glaros.dtc.umn.edu/gkhome/metis/metis/overiew

[17] P. Sanders and C. Schulz, "Think locally, act globally, Highly balanced graph partitioning," *Proceedings of the 12th International Symposium on Experimental Algorithms*, pp. 164-175, October, 2012.

[18] Y. Wang, Z. Lu, F. Glover, and J. Hao, "Path Relinking for Unconstrained Binary Quadratic Programming," *European Journal of Operational Research*, Vol. 223, pp.595-604, December, 2012.

[19] S. Feld, C. Roch, T. Gabor, C. Seidel, F. Neukart, I. Galter, W. Mauerer, and Claudia L. Popien, "A Hybrid Solution Method for the Capacitated Vehicle Routing Problem Using a Quantum Annealer," *Frontiers in ICT*, Vol. 6, June, 2019.

[20] Eranda Cela, "*The Quadratic Assignment Problem Theory and Algorithm*", Springer, NewYork,1998.

[21] Edward, K., "Performance Tuning for Cplex's Spatial Branch-and-Bound Solver for Global Nonconvex (Mixed Integer) Quadratic Programs," *Informs optimization Society Meeting*, 2017.

[22] Munera, D., Diaz, D., and Abreu, S., "Solving the Quadratic Assignment Problem with Cooperative Parallel Extremal Optimization," *the 16th European Conference on Evolutionary Computation in Combinatorial Optimization*. Porto, 2016.

[23] http://anjos.mgi.polymtl.ca/qaplib/

[24] K. F. Lau and Ken A. Dill, "A lattice statistical mechanics model of the conformational and sequence spaces of proteins," *Macromolecules*, Vol.22, No.10, pp.3986-3997, 1989.

[25] B. Maher, A. A. Albrecht, M. Loomes, X.-S. Yang and Kathleen Steinhöfel, "A firefly-inspired method for protein structure prediction in lattice models," *Biomolecules*, Vol.4, pp.56-75, 2014.