Efficient Traveling Salesman Problem Solvers using the Ising Model with Simulated Bifurcation

Tingting Zhang
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Canada
ttzhang@ualberta.ca

Jie Han
Department of Electrical and Computer Engineering
University of Alberta
Edmonton, Canada
jhan8@ualberta.ca

***Abstract*—An Ising model-based solver has shown efficiency in obtaining suboptimal solutions for combinatorial optimization problems. As an NP-hard problem, the traveling salesman problem (TSP) plays an important role in various routing and scheduling applications. However, the execution speed and solution quality significantly deteriorate using a solver with simulated annealing (SA) due to the quadratically increasing number of spins and strong constraints placed on the spins. The ballistic simulated bifurcation (bSB) algorithm utilizes the signs of Kerr-nonlinear parametric oscillators' positions as the spins' states. It can update the states in parallel to alleviate the time explosion problem. In this paper, we propose an efficient method for solving TSPs by using the Ising model with bSB. Firstly, the TSP is mapped to an Ising model without external magnetic fields by introducing a redundant spin. Secondly, various evolution strategies for the introduced position and different dynamic configurations of the time step are considered to improve the efficiency in solving TSPs. The effectiveness is specifically discussed and evaluated by comparing the solution quality to SA. Experiments on benchmark datasets show that the proposed bSB-based TSP solvers offer superior performance in solution quality and achieve a significant speed up in runtime than recent SA-based ones.**

***Index Terms*—Traveling salesman problem, ballistic simulated bifurcation, parallel update, Ising model, simulated annealing**

I. INTRODUCTION

Combinatorial optimization is widely used in various social and industrial applications. As the problem size increases, the search space becomes very large when looking for the optimal solution, due to the large number of combinations of decision variable values. However, a globally optimal solution is generally unnecessary for many applications.

For these computationally complex problems, Ising model-based solvers can obtain a nearly optimal solution at a high speed and efficiency [1]. The Ising model mathematically describes the properties of ferromagnetism. It is constructed for a set of spins, each taking one of the two states $\{+1, -1\}$. By decreasing the energy of the Ising model, the combination of the spin states eventually provides a suboptimal solution of the problem.

Given the distances between cities, a traveling salesman problem (TSP) is a combinatorial optimization problem to find the shortest route that passes through all cites once and then returns to the origin city [2]. To solve an n-city TSP using an Ising model-based solver, n^2 spins in a lattice are required, with the column and the row representing the city index and the visiting order, respectively. Double constraints are placed on the spins that require only one spin with the state "+1" in the same column and the same row to avoid (1) (mistakenly) visiting multiple cities in a single step and (2) visiting a city in multiple steps. Thus, a large TSP is notoriously difficult to solve due to the increased spin counts and the strong constraints.

As a heuristic algorithm, simulated annealing (SA) emulates the thermal annealing in physics [3]. However, the states of neighbor spins cannot be simultaneously updated, thus increasing the search time for solutions [4]. A clustering approach works with SA to improve the efficiency by dividing the original TSP into several sub-problems [2]. However, it only achieves partially parallel spin-update and the use of the K-means clustering itself introduces overhead in time.

A quantum mechanics-inspired algorithm referred to as simulated bifurcation (SB) can realize massive parallelism in computation [5]. By simulating the quantum adiabatic optimization of Kerr-nonlinear parametric oscillator networks, SB searches for an approximate solution by solving a pair of differential equations [6]. Two branches of the bifurcation (indicated by the sign of an oscillator's position) are considered as the two states of a spin. To restrain the errors introduced due to the use of continuous variables (for the positions), the ballistic SB (bSB) introduces hard thresholds to limit the evolution of the oscillators' positions to quickly find suboptimal solutions [7]. However, bSB has not been considered for solving TSPs.

This paper presents a first study on an efficient bSB-based TSP solver with several improvement strategies by taking advantage of the adiabatic evolution in bSB. Initially, the TSP is converted to an Ising problem without external magnetic fields and adapted for bSB. A redundant spin is introduced as an oscillator position in bSB. Then, rather than using a fixed time step, dynamically configurable time steps are applied for solving the differential equations in bSB to accelerate the convergence of energy. Moreover, the redundant position evolves during the search process using different approaches to gradually increase the relative significance of an external field placed on a spin over the interactions between spins.

The remainder of this paper is organized as follows. Section

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada (Project Number: RES0048688). T. Zhang was supported by a Ph.D. scholarship from the China Scholarship Council (CSC).

978-3-9819263-6-1/DATE22/©2022 EDAA

II presents the basics. Section III discusses the TSP solvers using the bSB. Experiment results are presented in Section V. Section VI concludes the paper.

II. Preliminaries

A. Ising Model-based Solvers

The total energy (Hamiltonian, H) of an Ising model with the external magnetic fields is expressed as [1]:

$$H = -\sum_{i,j} J_{ij}\sigma_i\sigma_j - \sum_i h_i\sigma_i, \tag{1}$$

where σ_i (or σ_j) denotes the state of the ith (or jth) spin, J_{ij} is the coupling coefficient between the ith and jth spins, and h_i is the external magnetic field for the ith spin.

B. Traveling Salesman Problems (TSPs)

An n-city TSP can be formulated as an Ising problem with external magnetic fields using n^2 spins in a lattice [2], as

$$H_{tsp} = -\sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n} J_{ikjl}\sigma_{ik}\sigma_{jl} - \sum_{i=1}^{n}\sum_{k=1}^{n} h_{ik}\sigma_{ik}, \tag{2}$$

where

$$J_{ikjl} = \begin{cases} -\frac{A}{8}W_{kl} & j=i+1 \quad or \quad i=j+1 \quad or \\ & i=1, j=n \quad or \quad i=n, j=1 \\ -\frac{B}{4} & i=j, k \neq l \\ -\frac{C}{4} & k=l, i\neq j \\ -\frac{B}{4}-\frac{C}{4} & k=l, i=j \\ 0 & otherwise \end{cases}, \tag{3}$$

$$h_{ik} = -\frac{A}{2}\sum_{l\neq k} W_{kl} - \frac{(n-2)B}{2} - \frac{(n-2)C}{2}. \tag{4}$$

In these equations, σ_{ik} (or σ_{jl}) indicates whether the kth (or lth) city is visited ("$+1$") or not ("-1") at the ith (or jth) step, and W_{kl} ($= W_{lk}$) denotes the distance between the kth and lth cities. A, B and C are the parameters to balance the relative strength of the objective function (by A) and constraints (by B and C).

C. Ballistic Simulated Bifurcation (bSB)

To solve an Ising problem in (1) but without the external field (h_i), the classical Hamiltonian for the Ising model with bSB and the Hamiltonian equations of motion are given by [7]

$$H_{bSB} = \sum_i \frac{a_0}{2}{y_i}^2 + \frac{a_0 - a(t)}{2}\sum_i {x_i}^2 - c_0\sum_{i,k} J_{ik}x_i x_j, \tag{5}$$

$$\dot{x_i} = \frac{\partial H_{bSB}}{\partial y_i} = a_0 y_i, \tag{6}$$

$$\dot{y_i} = -\frac{\partial H_{bSB}}{\partial x_i} = -\{a_0 - a(t)\}x_i + 2c_0\sum_{j=1}^{N} J_{ij}x_j, \tag{7}$$

where x_i and y_i are the position and the momentum of the ith oscillator, $\dot{x_i}$ and $\dot{y_i}$ denote the derivatives with respect to time, a_0 and c_0 are manually tuned constants. $a(t)$ is a time-dependent variable to guarantee the adiabatic evolution. In bSB, x_i is replaced by its sign and $y_i = 0$ when $|x_i| > 1$.

An Ising model-based solver with bSB utilizes the semi-implicit Euler method as an integrator to solve the pair of differential equations, (6) and (7). At the end of the search, the sign of x_i indicates the state of the ith spin.

III. Solving TSPs using the Ising Model with bSB

A. TSP Solvers using the Ising Model without External Fields

1) Reformulation of the TSP: To formulate the TSP for bSB, a redundant spin with the state $\sigma_{(n+1)(n+1)}$ fixed to "$+1$" is first introduced to (2) as

$$H_{tsp} = -\sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n} J_{ikjl}\sigma_{ik}\sigma_{jl} - \sum_{i=1}^{n}\sum_{k=1}^{n} h_{ik}\sigma_{ik}\sigma_{(n+1)(n+1)} \tag{8}$$

Then, each h_{ik} is divided by 2 to convert the external magnetic fields to the coupling coefficients between n^2 spins and the redundant one. Different from the mapping in (2), therefore, an n-city TSP is reformulated as an Ising problem without external magnetic fields by expanding n^2 spins to $(n+1)^2$ spins in a lattice, as

$$\begin{aligned} H_{tsp} &= -\sum_{i=1}^{n}\sum_{k=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{n} J_{ikjl}\sigma_{ik}\sigma_{jl} - \sum_{i=1}^{n}\sum_{k=1}^{n}\frac{h_{ik}}{2}\sigma_{ik}\sigma_{(n+1)(n+1)} \\ &\quad -\sum_{j=1}^{n}\sum_{l=1}^{n}\frac{h_{jl}}{2}\sigma_{(n+1)(n+1)}\sigma_{jl} \\ &= -\sum_{i=1}^{n+1}\sum_{k=1}^{n+1}\sum_{j=1}^{n+1}\sum_{l=1}^{n+1} J'_{ikjl}\sigma_{ik}\sigma_{jl}, \end{aligned} \tag{9}$$

where

$$J'_{ikjl} = \begin{cases} J_{ikjl} & i,k,j,l \in \{1,2,...,n\} \\ \frac{h_{ik}}{2} & i,k \in \{1,2,...,n\} \ and \ j=l=n+1 \\ \frac{h_{jl}}{2} & j,l \in \{1,2,...,n\} \ and \ i=k=n+1 \\ 0 & otherwise \end{cases}. \tag{10}$$

To satisfy the constraint that there is only one spin with an up state ("$+1$") in the same row and the same column, $\sigma_{(n+1)(n+1)}$ is fixed to "$+1$" and the states of the other spins in the $(n+1)$th dimension are fixed to "-1".

2) Solving the TSP with bSB: Following (5)-(7), the classical Hamiltonian for the Ising model in (9) using bSB to solve TSPs ($H_{tsp_{bSB}}$) and the corresponding pair of differential equations are given by

$$H_{tsp_{bSB}} = \sum_{i,k}\frac{a_0}{2}{y_{ik}}^2 + \frac{a_0 - a(t)}{2}\sum_i {x_{ik}}^2 - c_0\sum_{i,j,k,l} J'_{ikjl}x_{ik}x_{jl}, \tag{11}$$

$$\dot{x_{ik}} = a_0 y_{ik}, \tag{12}$$

$$\begin{aligned} \dot{y_{ik}} &= -\{a_0 - a(t)\}x_{ik} + 2c_0\textstyle\sum_{j=1}^{n+1}\sum_{l=1}^{n+1} J'_{ikjl}x_{jl} \\ &= -\{a_0 - a(t)\}x_{ik} + 2c_0\textstyle\sum_{j=1}^{n}\sum_{l=1}^{n} J_{ikjl}x_{jl} \\ &\quad + c_0\textstyle\sum_{j=1}^{n}\sum_{l=1}^{n} h_{ik}x_{(n+1)(n+1)}, \end{aligned} \tag{13}$$

where x_{ik} and y_{ik} ($i,k \in \{1,2,...,n\}$) are the position and the momentum of the oscillator in the ith row and kth column in a lattice. $x_{(n+1)(n+1)}$ is expected to be 1 at the end of the search to ensure the spin state $\sigma_{(n+1)(n+1)}$ to be "$+1$".

With (12) and (13), a TSP can efficiently be solved by using bSB.

TABLE I
DIFFERENT DYNAMIC CONFIGURATIONS OF THE TIME STEP (DTS) IN THE ISING MODEL-BASED SOLVER WITH BSB

Dynamic Configurations		Formulation
Small-large	**DTS1: equally distributed**	$\triangle_t = \begin{cases} 0.5 & r < \frac{iter}{2} \\ 1 & r \geq \frac{iter}{2} \end{cases}$
	DTS2: large $\triangle_t$ preferred	$\triangle_t = \begin{cases} 0.5 & r < \frac{iter}{3} \\ 1 & r \geq \frac{iter}{3} \end{cases}$
	DTS3: small $\triangle_t$ preferred	$\triangle_t = \begin{cases} 0.5 & r < \frac{2 \times iter}{3} \\ 1 & r \geq \frac{2 \times iter}{3} \end{cases}$
Large-small-large	**DTS4**	$\triangle_t = \begin{cases} 0.5 & \frac{iter}{3} < r < \frac{2 \times iter}{3} \\ 1 & otherwise \end{cases}$

TABLE II
DIFFERENT EVOLUTION APPROACHES (EAS) FOR $x_{(n+1)(n+1)}$ IN THE ISING MODEL-BASED SOLVER WITH BSB

Evolution		Formulation
Linear	**EA1**	$x_{(n+1)(n+1)} = \frac{r}{2iter} + 0.5$
Constant-linear	**EA2**	$x_{(n+1)(n+1)} = \begin{cases} 0.5 & r < \frac{iter}{2} \\ \frac{r}{iter} & r \geq \frac{iter}{2} \end{cases}$
Linear-constant	**EA3**	$x_{(n+1)(n+1)} = \begin{cases} \frac{r}{iter} & r < \frac{iter}{2} \\ 1 & r \geq \frac{iter}{2} \end{cases}$
Constant-constant	**EA4**	$x_{(n+1)(n+1)} = \begin{cases} 0.5 & r < \frac{iter}{2} \\ 1 & otherwise \end{cases}$
Exponential	**EA5**	$x_{(n+1)(n+1)} = 0.5 + \frac{r^2}{2iter^2}$

B. Improvement Strategies

1) Dynamic Time Steps: To accelerate the convergence of Hamiltonian, the dynamic configuration of the time step (DTS) is considered to solve the pair of differential equations (12) and (13). In hardware, the multiplication with 0.5 can be implemented by using a shift operation and the multiplication with 1 does not need any specific processing. Therefore, for an efficient hardware implementation, the time step (denoted by $\triangle_t$) is selected to be 0.5 or 1 by using a piecewise function during a given iteration (denoted by $iter$) in the update of the spin states. Four different dynamic configurations of the time step are considered, as shown in Table I.

Since it is more challenging to skip the local minimum as time increases, a small time step $\triangle_t = 0.5$ is used at the beginning of a search to ensure the solution quality and a large time step $\triangle_t = 1$ is used near the end of the search to increase the probability of changing the state of a spin. Three configurations are developed by using different proportions of small and large time steps during the update of the spin states. As a basic configuration, DTS1 employs equally distributed time steps by taking the value of either 0.5 or 1. The large time step is preferred in the last two-thirds of iterations by using the configuration referred to as DTS2, whereas the configuration referred to as DTS3 uses small time steps during the first two-thirds of iterations.

The state of a spin (σ_{ik}) is determined by the sign of the related position (x_{ik}), which is difficult to change at the beginning before the bifurcation occurs. Therefore, the large time step is used at both the beginning and the end of search in the configuration referred to as DTS4.

2) Evolution of $x_{(n+1)(n+1)}$: If $x_{(n+1)(n+1)}$ is fixed at 1 during the entire simulation, the effect of the external field on the spin is more significant than the interaction between spins. Moreover, $\sigma_{(n+1)(n+1)}$ is "+1" as long as the sign of $x_{(n+1)(n+1)}$ is positive. Therefore, different evolution approaches (EAs) are considered for $x_{(n+1)(n+1)}$ to change gradually from a positive value to 1, as presented in Table II.

In EA1 and EA5, $x_{(n+1)(n+1)}$ increases linearly and exponentially, respectively, from 0.5 to 1 with time. Using EA2, $x_{(n+1)(n+1)}$ remains at 0.5 in the first half of the iteration and then linearly increases to 1 in the second half, whereas in EA3, $x_{(n+1)(n+1)}$ linearly increases in the first half of iterations and stays unchanged as 1 in the second half. $x_{(n+1)(n+1)}$ takes 0.5 at first and then takes 1 using EA4.

IV. EXPERIMENTAL RESULTS

A. Experiment Setup

Consider (12) and (13), where we set $a_0 = 1$ and $a(t)$ increases from 0 to 2 to ensure the adiabatic evolution. $\boldsymbol{x}$ and $\boldsymbol{y}$ are initialized to a zero matrix and a random matrix with entries within $[-0.1, +0.1]$, respectively. The parameters A, B, and C are, respectively, set to 1, $\max\{W\}$ and $\max\{W\}$, where $\max\{W\}$ represents the maximum value in W. Three datasets from the TSPLIB benchmark [8] are considered. The solution quality is evaluated by the average (Ave), the maximum (Max), the minimum (Min), and the standard deviation (Std) of the obtained route distances from 100 trials.

B. Using Different Dynamic Configurations of the Time Step

The quality of the TSP solvers based on the Ising model with bSB using different dynamic configurations of the time step are presented in Table III with the default value 1 for $x_{(n+1)(n+1)}$. Using $\triangle_t = 0.5$ leads to a higher solution quality than using $\triangle_t = 1$. An improvement of 9.4%, 10.9%, and 11.8% in Ave has been achieved, respectively, when solving the problems in burma14, ulysses16 and ulysses22.

Compared with using a constant time step, the use of dynamic configurations of the time step leads to a higher solution quality (with smaller Ave, Max, Min). As seen from the results obtained by using DTS1, DTS2 and DTS3 in Table III, the solution quality is improved with the increased use of the small time step ($\triangle_t = 0.5$). Compared with DTS1 and DTS2, DTS3 prefers small $\triangle_t$ in most iterations during the search, so it shows a higher solution quality. Moreover, DTS4 improves the efficiency and stability in the solutions by exploiting the inherent characteristics of the bSB algorithm. Although DTS4 employs the small time step ($\triangle_t = 0.5$) in only one third of all updates, it achieves a reduction up to 2.5% in Ave, 10.4% in Max, but with a slight increase up to 2.9% in Min than using the constant $\triangle_t = 0.5$; it also achieves a reduction up to 13.6% in Ave, 23.4% in Max, and 9.6% in Min than using the constant time step $\triangle_t = 1$.

TABLE III
SOLUTION QUALITY (IN THE STATISTICS OF TRAVELLED DISTANCES) OF USING DIFFERENT DTS FOR THE BSB-BASED TSP SOLVERS

Metrics	burma14						ulysses16						ulysses22					
	$\triangle_t = 0.5$	$\triangle_t = 1$	**DTS1**	**DTS2**	**DTS3**	**DTS4**	$\triangle_t = 0.5$	$\triangle_t = 1$	**DTS1**	**DTS2**	**DTS3**	**DTS4**	$\triangle_t = 0.5$	$\triangle_t = 1$	**DTS1**	**DTS2**	**DTS3**	**DTS4**
$\boldsymbol{Ave}$	3707	4091	4005	4011	3775	3679	7678	8619	8393	8389	7865	7479	8441	9577	9292	9547	8258	8267
$\boldsymbol{Max}$	4217	4647	4936	4776	4331	4150	8637	9853	9785	9853	9668	8496	10353	12109	11301	11701	9575	9273
$\boldsymbol{Min}$	3371	3672	3536	3511	3413	3417	6803	7180	7172	7439	7079	6863	7203	8208	7779	8210	7249	7419
$\boldsymbol{Std}$	211	207	263	297	255	230	433	643	633	559	572	459	584	945	830	803	597	489

TABLE IV
SOLUTION QUALITY (IN THE STATISTICS OF TRAVELLED DISTANCES) OF DIFFERENT EAS FOR $x_{(n+1)(n+1)}$ FOR THE BSB-BASED TSP SOLVERS

Metrics	burma14					ulysses16					ulysses22				
	EA1	**EA2**	**EA3**	**EA4**	**EA5**	**EA1**	**EA2**	**EA3**	**EA4**	**EA5**	**EA1**	**EA2**	**EA3**	**EA4**	**EA5**
$\boldsymbol{Ave}$	3780	3836	3939	3992	3792	7999	8015	8448	8529	7997	8646	8810	9410	9485	8499
$\boldsymbol{Max}$	4525	4751	4757	4841	4903	9224	10027	9914	9763	10718	10012	10397	11029	11191	10296
$\boldsymbol{Min}$	3323	3346	3511	3454	3323	7062	7027	7318	7332	6857	7330	7545	7993	8000	7451
$\boldsymbol{Std}$	269	464	274	260	400	539	702	608	575	674	608	642	642	721	672

C. Using Different Evolution Approaches for $x_{(n+1)(n+1)}$

Table IV presents the solution quality of using different evolution approaches. $\triangle_t$ is fixed to 1 to simply the hardware implementation. No matter which evolution approach is employed, the bSB-based TSP solver achieves an improvement in the solution quality. The EA1 and EA5, where $x_{(n+1)(n+1)}$ increases linearly and exponentially, respectively, lead to a significant improvement, while a slight improvement is obtained by using EA4. Using EA1 and EA5 obtains a similar improvement in Ave, at least by 7.3%, 7.1% and 9.7% for solving burma14, ulysses16, and ulysses22, respectively. Moreover, using the bSB-based TSP solver with EA1 can find a solution with higher stability (with a smaller Std).

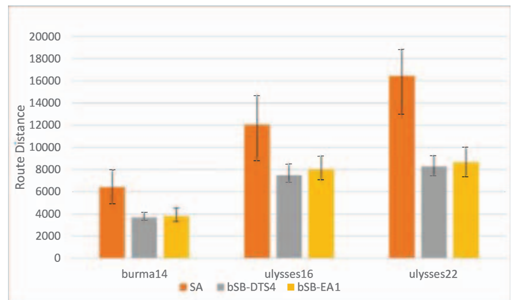

Fig. 1. Comparison of the TSP solvers using the improved SA-based [4] and the bSB-based Ising models.

D. Comparison

Fig. 1 shows a comparison of the performance of using the improved SA [4] and the proposed bSB. The bSB-based TSP solvers with the DTS4 and the EA1 are referred to as bSB-DTS4 and bSB-EA1, respectively. The number of iterations is $50k$ in the improved SA and $2k$ in the bSB.

The Ising model-based solver with bSB-DTS4 offers solutions of higher quality (with smaller Ave and Max) than that with bSB-EA1. Compared with the recent SA [4], due to the parallel spin update and faster convergence of Hamiltonian, bSB-DTS4 can significantly improve the Ave and the Std by 42% and 66% respectively, with about 7.34$\times$ shorter runtime in solving burma14. It further reduces the Ave and the Std by 37% and 62% respectively, with about 13.44$\times$ shorter runtime in solving ulysses16. Finally, it reduces the Ave and the Std by up to 47% and 67% respectively, with about 7.69$\times$ shorter runtime in solving ulysses22. A similar solution quality can be obtained by using the SA [4] if the iteration increases, but at the cost of more than 100$\times$ runtime.

V. CONCLUSION

In this paper, an efficient Ising model-based TSP solver using bSB with dynamic configurations of time steps and evolutions of a redundant oscillator position is proposed to realize the fully parallel update of the spin states. The external magnetic fields are first considered as the coefficients between spins by introducing a redundant spin as the position of an oscillator in bSB. In this way, the TSP is mapped to an Ising problem without external magnetic fields. Moreover, several improvement strategies are proposed for updating the spins' states by leveraging the features of bSB, including dynamically configuring the time step and evolving the redundant position. Due to the massive parallel processing capacity, the proposed method can solve the TSP with an improvement in the solution quality by at least 37% with 13.44$\times$ shorter runtime than using an SA method. These results provide an opportunity for devising parallel processing circuits to efficiently solve constrained combinatorial problems such as the TSP. This hardware design will be addressed in future work.

REFERENCES

[1] A. Lucas, "Ising formulations of many NP problems," Frontiers in Physics, vol. 2, no. 5, 2014.

[2] A. Dan, R. Shimizu, T. Nishikawa, S. Bian and T. Sato, "Clustering approach for solving traveling salesman problems via Ising model based solver," in Proc. ACM/IEEE Design Automation Conference, pp. 1-6, 2020.

[3] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," Science, vol. 220, no. 4598, pp.671-680, 1983.

[4] A. Minamisawa, R. Iimura, and T. Kawahara, "High-speed sparse Ising model on FPGA," in Proc. IEEE International Midwest Symposium on Circuits and Systems, pp. 670-673, 2019.

[5] H. Goto, T. Kosuke, and R. D. Alexander, "Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems," Science Advances, vol. 5, no. 4, 2019.

[6] H. Goto, "Bifurcation-based adiabatic quantum computation with a nonlinear oscillator network," Scientific Reports, vol. 6, no. 1, pp.1-8, 2016.

[7] H. Goto, E. Kotaro, S. Masaru, S. Yoshisato, K. Taro, H. Yohei, et al., "High-performance combinatorial optimization based on classical mechanics," Science Advances, vol. 7, no. 6, 2021.

[8] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/