Constrained parallel tempering in traveling-salesman problems with circular neighborhoods

Andrea Grimaldi,[1,2] Davi R. Rodrigues,[1] Eleonora Raimondo,[3] Vito Puliafito,[1] Vincenza Crupi,[4] Vitoantonio Bevilacqua,[1] Mario Carpentieri,[1] Francesca Garesci,[2] and Giovanni Finocchio[3,4,*]

[1] *Department of Electrical and Information Engineering, Politecnico di Bari, Bari, Italy*

[2] *Department of Engineering, University of Messina, Messina, Italy*

[3] *Istituto Nazionale di Geofisica e Vulcanologia, Rome, Italy*

[4] *Department of Mathematical and Computer Sciences, Physical Sciences and Earth Sciences, University of Messina, Messina, Italy*

(Received 24 June 2025; revised 15 November 2025; accepted 6 January 2026; published 6 February 2026)

This paper is a contribution to the Physical Review Applied collection titled Physics-Inspired Computing.

Combinatorial optimization problems such as the traveling-salesman problem (TSP) are central to many applications ranging from physics to mechanical engineering, such as toolpath planning, robotic navigation, and drone routing. In this work, we propose the concept of constrained parallel tempering (CPT), an energy-minimization algorithm implemented on probabilistic Ising machines (PIMs) with probabilistic bits (p-bits) tailored for constrained problems. CPT separates the constraint and target components of the encoding into distinct energy terms with independent sampling temperatures, improving convergence and reducing computational overhead compared with traditional parallel tempering. We demonstrate the effectiveness of CPT on a benchmark of a TSP instance and extend the method to a more general formulation, the TSP with circular neighborhoods (TSPCNs), where each node is defined as a region, not a point. This model reflects real-world flexibility in robotic and drone-based applications. A hybrid deterministic–Monte Carlo refinement is then used to handle the continuous solution space of TSPCNs. Additionally, we present a hardware-friendly approach to generate Gaussian-distributed random numbers using a PIM with p-bits. Our results suggest that CPT-PIM frameworks offer an approach with better scalability and potentially low-power implementation for embedding intelligent decision-making into edge computing applications.

DOI: 10.1103/plr8-xys1

I. INTRODUCTION

The solution of combinatorial optimization problems (COPs) is one of the most pervasive problems that industries have to face in the modern era. COPs arise across a wide range of fields and are essential to applications such as airline route planning [1] and network routing [2]. One important use case is optimization of tool paths for computer numerical control machines to minimize travel time and maximize energy efficiency [3]. Similar strategies are applied to route autonomous robots for assembly and material handling tasks, as well as drones for data collection.

These optimization tasks can be formulated as COPs [4,5] that can be solved by well-established algorithms. Among the most prominent formulations is the traveling-salesman problem (TSP) [6], which seeks the shortest route that visits each node in a graph exactly once and returns to the starting point. Because of its versatility, the TSP serves as a key model for a wide range of applications, from large-scale transportation logistics to small-scale robotic path planning. However, as an NP-hard problem [5], solving the TSP becomes computationally intensive for large instances, resulting in significant energy demands.

In embedded and resource-constrained systems, such as autonomous drones or mobile robots, this computational load is often off-loaded to cloud infrastructure. The device transmits the problem, waits for remote processing, and receives the optimized solution. While effective, this approach introduces latency, increases energy consumption, and relies on consistent network connectivity, all of which limits the autonomy and real-time responsiveness of such systems in practical scenarios.

*Contact author: gfinocchio@unime.it

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

Edge computing has emerged as a promising alternative to cloud-based computing for systems with resource constraints [7,8]. Recent advances in unconventional computing paradigms have significantly amplified its potential by enabling local processing with increased efficiency [9,10]. In particular, Ising machines (IMs) [11,12] have attracted considerable attention due to their ability to solve COPs directly in hardware. Leveraging analog architectures and inherently energy-efficient implementations, IMs offer a compelling solution for edge-based optimization tasks [13,14].

Encoding problems such as the TSP into Ising models remains challenging due to the need to incorporate hard constraints into the energy landscape alongside the optimization objective. Both are typically scaled with use of the same parameter in the Hamiltonian, often interpreted as a temperature or annealing coefficient, making it difficult to balance constraint enforcement with cost minimization. This can lead to convergence issues or invalid solutions if tuning is not done carefully.

We introduced the concept of constrained parallel tempering (CPT), a modified parallel tempering (PT) algorithm tailored for Ising problems with constraint-heavy formulations. This idea is supported by other constraint-aware PT implementations in the literature. Delacour *et al*. [15] showed that a two-dimensional PT can help improve performance on problems with copy nodes, which usually suffer from chain-break issues for improper values of the coupling strength. We evaluated CPT using a standard benchmark problem, the TSP, and demonstrated its superiority compared with traditional PT and other energy-minimization algorithms. Additionally, we integrate CPT with a Monte Carlo (MC) sampling approach to solve a variant of the TSP in which destinations are defined as circular neighborhoods rather than discrete nodes. This abstraction is relevant to real-world robotic and drone navigation. To support this, we developed a probabilistic Ising machine (PIM) that can generate real-valued samples from a discretized Gaussian distribution using auxiliary probabilistic bits (p-bits). Unlike standard p-bits, which typically produce uniformly distributed outputs, our design enables tunable, nonuniform sampling. These contributions are particularly relevant for autonomous systems, offering scalable, hardware-efficient optimization solutions under uncertainty [16,17].

II. ISING MODEL ENCODING OF THE TSP

Formally, IMs are characterized by the following Hamiltonian [18]:

$$\mathcal{H}(\boldsymbol{\sigma}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} J_{ij}\sigma_i\sigma_j - \sum_{i=1}^{N} h_i\sigma_i, \tag{1}$$

where $\boldsymbol{\sigma}$ is the state of the system, consisting of N spins, each of them with a binary value (-1 or $+1$), J_{ij} is the coupling between the ith spin and the j th spin, and h_i is the bias of the ith spin.

When one is encoding the TSP into an Ising model, the first thing to consider is that any valid solution of a TSP is a Hamiltonian cycle of its graph. As a Hamiltonian cycle requires an ordered sequence of nodes, more than a single bit of information is necessary to completely characterize a cycle. One way is to encode an N-node graph into an N^2-spin system, as illustrated in Fig. 1(a). To each node of a graph, shown at the top left, correspond N spins, their position in the Hamiltonian cycle represented as one-hot vectors, as shown by the matrix at the top right. Each row of the matrix must have a single value that is equal to $+1$, as the route cannot visit more than one city at once. Each column also requires there to be exactly one value equal to $+1$, as each node must be visited only once.

To embed the shortest route requirement into the energy landscape of the Ising model [19], an established way is to divide the Hamiltonian into two components: the constraint Hamiltonian $\mathcal{H}_C$, and the target Hamiltonian $\mathcal{H}_T$. The former is the one that ensures that only Hamiltonian cycles are ground states of the system, and can be written as follows:

$$\mathcal{H}_C = \sum_{v=1}^{N}\left(1-\sum_{j=1}^{N} x_{v,j}\right)^2 + \sum_{j=1}^{N}\left(1-\sum_{v=1}^{N} x_{v,j}\right)^2 + \sum_{(uv)\notin E}\sum_{j=1}^{N} x_{u,j}x_{v,j+1}, \tag{2}$$

where $x_{v,j}$ is the binarized variable that encodes whether the vth node is visited at the j th of the cycle, and E is the set of the edges of the graph. The correspondence between binarized variables and spins is affine, as $x_{v,j} = (\sigma_{Nv+j}+1)/2$, meaning that expressing the Hamiltonian in terms of spins does not increase the order of interactions, which are still at most quadratic (still an Ising Hamiltonian). The first term is minimal when each node is visited only once, and the second term is minimal when only one node is visited at a given position of the cycle. The last term is zero only if the cycle does not contain movements along edges that are not present in the graph. For the context of this work, as all problems considered have all-to-all connections, this term is absent.

For an all-to-all problem, $\mathcal{H}_T$ can be defined as

$$\mathcal{H}_T = \sum_{u=1}^{N}\sum_{v=u}^{N} W_{uv}\sum_{j=1}^{N} x_{u,j}x_{v,j+1}, \tag{3}$$

where W_{uv} is the weight (distance) between the uth node and the vth node. The reciprocal weights between all the

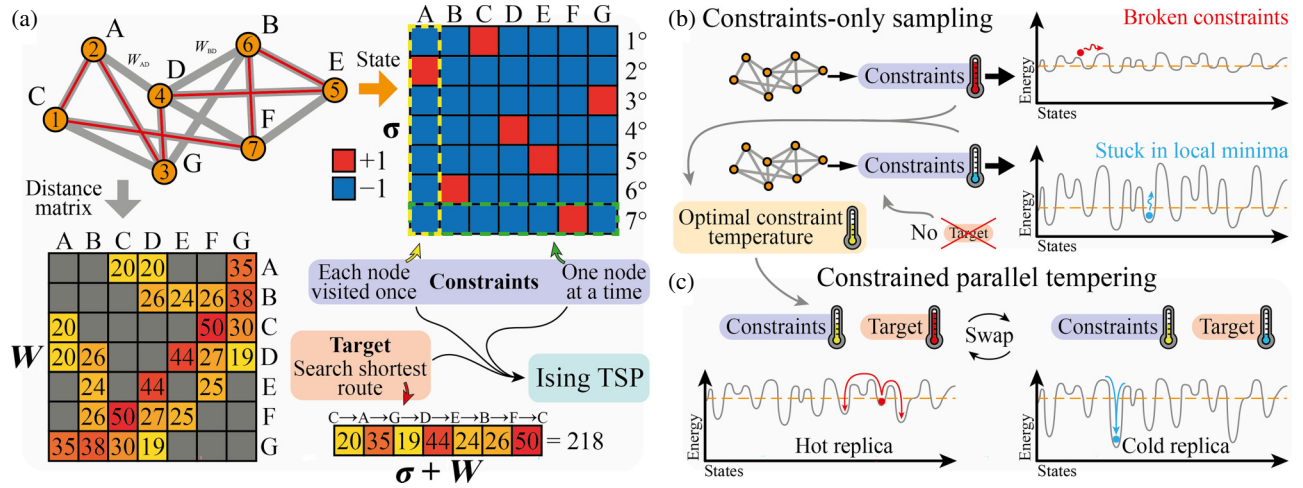


FIG. 1. (a) TSP encoding into an Ising model. The TSP instance (top left) is characterized by a $\boldsymbol{W}$ matrix (bottom left) that contains the edge information of the graph (which nodes are connected and their weight). The state, i.e., the visiting order, is encoded in binary with N^2 spins, each node represented by N spins, indicating when that node is visited. (b) By sampling from the constraints-only Hamiltonian at fixed temperature, we can find an optimal constraint temperature. (c) In CPT, it is used along with a set of target temperatures to enhance the search for the ground state.

nodes are stored in a weight matrix $\boldsymbol{W}$. This matrix typically contains the Euclidean distances between the nodes, although other distance-based metrics, such as the L1 norm or geodesic, or time-based or cost-based metrics are also possible [20]. Generally, this means that $\boldsymbol{W}$ is symmetrical and its main diagonal is zero. An example of a weight matrix is illustrated in Fig. 1(a), at the bottom left.

The full Hamiltonian of the TSP instance can thus be defined as

$$\mathcal{H}_{\mathrm{TSP}} = \mathcal{H}_C + \rho \mathcal{H}_T, \tag{4}$$

where ρ is a scaling factor that balances energy minimization and constraint satisfaction.

III. PROBABILISTIC ISING MACHINES

IMs can be obtained with the use of different strategies, spin encodings, and hardware devices [11,21,22]. In this work we focus on PIMs [12], a paradigm that uses the p-bit as the Ising spins [23], particularly compatible with spintronics implementations that use magnetic tunnel junctions to physically represent p-bits [13,14]. To minimize the energy landscape defined by the Hamiltonian (1), PIMs use the following equation to update the state of each p-bit:

$$\sigma_i = \mathrm{sgn}(\mathrm{rand}(-1,+1) + \tanh(\beta I_{i,\mathcal{H}}(\sigma))), \tag{5}$$

where $\mathrm{sgn}(x)$ is the sign function, $\mathrm{rand}(-1,+1)$ is a random number between -1 and $+1$, β is a pseudotemperature parameter, and $I_{i,\mathcal{H}}(\boldsymbol{\sigma})$ is the input signal received by the ith p-bit according to the Hamiltonian $\mathcal{H}$ and with state $\boldsymbol{\sigma}$; it is expressed as

$$I_{i,\mathcal{H}}(\boldsymbol{\sigma}) = \sum_{j=i}^{N} J_{ij}\sigma_j + h_i, \tag{6}$$

which effectively represents the discrete derivative of $\mathcal{H}$ with respect to σ_i with a negative sign. The inverse-temperature parameter β plays a critical role in energy minimization. High β induces near-deterministic behavior that favors energy minimization but increases the risk of becoming trapped in local minima. In contrast, low β promotes stochastic exploration of the solution space, facilitating escape from local minima but often at the expense of convergence. To balance these effects, algorithms such as simulated annealing (SA) [24] and PT [25] are widely used in problems such as the maximum-satisfiability problem, the maximum-cut problem, and integer factorization [13], where both exploration and exploitation are essential.

When a Hamiltonian defined as in Eq. (4) is used, it is possible to define a second temperature parameter as $\gamma = \beta\rho$, which can be considered the target temperature. In constrained problems such as the TSP, tuning β and γ is particularly challenging. Low values of β lead to frequent sampling of invalid solutions due to weak constraint enforcement, while high values hinder the system's ability to escape local minima by overly restricting the search space. Different strategies have been proposed to address this challenge, such as the use of a fixed, optimally tuned β [26], or application of simulated annealing, where β is gradually increased from zero [27].

ALGORITHM 1. Modified swapping algorithm.

if $\boldsymbol{\sigma}_H$ is valid**:**
  if $\boldsymbol{\sigma}_C$ is valid**:**
    Perform standard PT swap on $\mathcal{H}_T$
  else:
    Reject swap
else:
  Reject swap

The effect of β is illustrated in Fig. 1(b). At $\gamma = 0$, low β results in the PIM frequently visiting constraint-violating states (upper part), whereas high β prevents the system from escaping any local minimum that satisfies the constraints (lower part), since reaching a new minimum requires temporarily violating constraints, an event highly improbable at large β. The optimal pseudotemperature β_C is the value that maximizes the probability of finding the global optimum by balancing constraint enforcement with sufficient dynamic flexibility.

IV. CONSTRAINED PARALLEL TEMPERING

In CPT, rather than using the update algorithm outlined in Eq. (2), we use the following equation:

$$\sigma_i = \mathrm{sgn}(\mathrm{rand}(-1,+1) + \tanh(\beta_C I_{i,\mathcal{H}_C}(\boldsymbol{\sigma}) + \gamma_r I_{i,\,\mathcal{H}_T}(\boldsymbol{\sigma}))), \tag{7}$$

where γ_r is the target pseudotemperature of the rth replica. Rather than our considering $\mathcal{H}_{\mathrm{TSP}}$ as a single Ising model, it is treated as two separate models, as shown in Fig. 1(c): one is kept at the optimal sampling pseudotemperature β_C, and the other has variable pseudotemperature, depending on the replica. The hotter replicas sample valid states (satisfying all constraints) with no regard to path length, and the colder replicas settle on optimal valid states. The modified swapping algorithm is given in Algorithm 1.

The standard PT swap consists in calculating $\exp((\mathcal{H}_T(\boldsymbol{\sigma}_H) - \mathcal{H}_T(\boldsymbol{\sigma}_C))(\beta_H - \beta_C))$ and comparing it with a uniform random number between 0 and 1. If the random number is smaller than the swapping probability coefficient, the swap is performed.

A simple way to determine an optimal β_C value is to estimate it by running several simulated annealing schedules with β varying from zero to some large value that freezes the sampler. β with a high variance of valid states on average is designated as β_C. Although β_C is computed with a heuristic approach with a trade-off between preprocessing computational time and sampling efficiency, this approach makes the CPT easily adaptive to scenario changes, a characteristic that is useful for edge applications. However, a comprehensive study taking into consideration the effect of both temperature parameters is a useful tool to gain insights into the system's dynamics.

V. CPT PARAMETER OPTIMIZATION

To study the effect of the choice of the parameters β and γ on the system's dynamics, we performed a systematic study on the benchmark instance considered, burma14, by choosing a set of 25×25 (β, γ) pairs, running 5000 tests of 1000 sweeps for each pair, and studying the trend of some performance metrics. The results are shown in Fig. 2.

The first metric considered is the probability of sampling a valid route, i.e., a route that does not break any of the TSP constraints, shown in Fig. 2(a). The trend shows that increasing β generally leads to the system visiting almost exclusively valid routes, as the constraints become more stringent on the dynamics. Increasing γ has the opposite effect, as its increase leads to the target Hamiltonian being prioritized over the constraint Hamiltonian, which naturally leads to less valid routes. Interestingly, however, a low value of γ, about 1.5×10^{-3} for the problem considered here, leads to the visiting of valid routes with even lower β values compared with having $\gamma = 0$.

A metric complementary to the number of valid routes is the number of unique routes, shown in Fig. 2(b). While too large of an increase in γ still leads to no valid routes (and thus no unique routes) being visited, the behavior is more complex when we factor in the parameter β. More unique routes are visited when β and γ balance each other's effects on the dynamics; however, when β is too large, the system is stuck in a valid route and unable to explore the solution space further.

Figures 2(c) and 2(d) show the probability of sampling a solution within 70% and 40% approximation of the ground state, respectively. The approximation ratio of a route $\boldsymbol{\sigma}$ is defined as $(\mathcal{H}_T(\boldsymbol{\sigma}) - \mathcal{H}_{\mathrm{GS}})/\,\mathcal{H}_{\mathrm{GS}}$, with $\mathcal{H}_{\mathrm{GS}}$ being the optimal route length. The behavior is somewhat similar to that of the number of unique routes, although larger (β, γ) pairs lead to better sampled routes.

One interesting observation is that all metrics show a linear boundary between the region in which the parameters are inadequate and the region in which the PIM behaves as intended. However, this linear boundary does not follow a line passing through the origin (both β and γ equal to 0). This gives a natural insight into why annealing and tempering techniques using the traditional Hamiltonian of the TSP, with a fixed scaling parameter ρ between the constraint Hamiltonian and the target Hamiltonian, do not perform very well. Any simulated annealing schedule or replicas' temperature range can be represented as a line in the β vs γ space. All such lines that pass through (β, γ) pairs with constant ρ also pass through the origin. However, the highest probability lines for all metrics do not intersect one of the axes in the first quarter at the origin, but rather at some $(\beta,\ 0)$ point.

We identify four working points, shown in Fig. 2 as white crosses, as particular (β, γ) combinations that strike

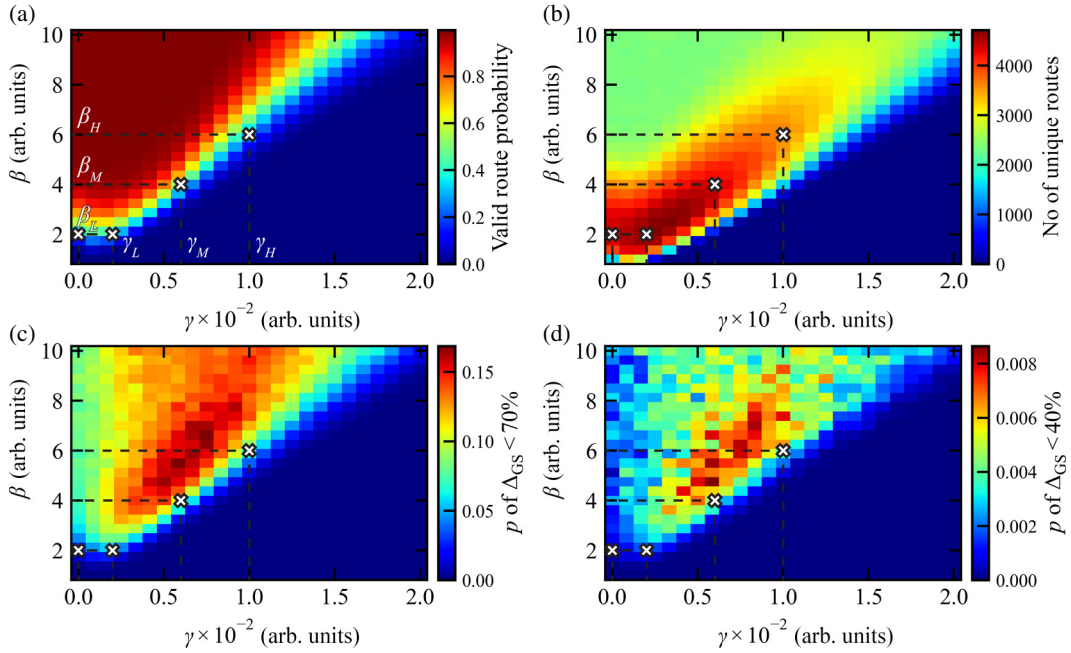


FIG. 2. Parameter study on the instance burma14 on different metrics as a function of β and γ. (a) Probability of sampling a valid route (all constraints satisfied). (b) Number of unique valid routes found in the 5000 tests of 1000 sweeps. (c),(d) Probability of sampling a valid route that is within 70% (40%) approximation of the ground state. In all plots, four working points are identified for subsequent testing, chosen in the regions that strike a balance between the four metrics. The three β values are labeled as β_L, β_M, and β_H, and the three γ values are labeled as γ_L, γ_M, and γ_H.

a balance between the considered metrics as a base for a comprehensive comparison, as illustrated in the next section. The three β values (β_L, β_M, β_H)and the three γ values (γ_L, γ_M, γ_H) selected are used to identify the four working points: $(\beta_L, 0)$, (β_L, γ_L), (β_M, γ_M), and (β_H, γ_H).

VI. ENERGY-MINIMIZATION COMPARISON

To assess the quality of the constrained formulation of the problem, we use the four working points identified in the previous section for the burma14 instance to set up a comprehensive comparison between energy-minimization algorithms with several choices of the two temperature parameters. The algorithms examined are standard Gibbs sampling at fixed temperatures (S), linear SA, and PT. In the last case, since PT performance is nontrivially affected by the number of replicas used, three tests per chosen parameter ranges with 5, 10, and 20 replicas were performed. For S and SA, as each replica is independent of the others, solving the probability for R replicas $p_{S,R}$ can be trivially achieved from the single replica p_S with $p_{S,R} = 1 - (1 - p_S)^R$. In the case of S, as both β and γ are fixed for the entire schedule, ρ is also necessarily constant. In SA, since both β and γ change linearly from one value to another, it is possible to have ρ change during the schedule. In PT, since replicas have β and γ values linearly distributed over a range, it is possible to have ρ change between replicas. We refer to schedules in which ρ is not constant (either in time or between replicas) as "constrained" (thus "CSA" and "CPT"), the others being "standard."

In Fig. 3, the energy-minimization schedules considered for the comparison are summarized in β vs γ graphs. The S schedules are represented by a single marker with unique face and edge color in Fig. 3(a), the SA and PT schedules are represented by solid lines of different colors in Fig. 3(b), and the CSA and CPT schedules are represented by lines with unique style and color in Fig. 3(c). The schedules are also listed in terms of their β and γ ranges in the legend at the bottom. The standard schedules (with constant ρ) are framed in orange, and the constrained schedules (with varying ρ) are framed in blue.

The results of the comparison are shown in Fig. 4. All tests consist of 100 runs with 10^5 sweeps each. Several metrics are considered: in Figs. 4(a)–4(c) the probability of finding at least one state that approximates the ground state within 20% ($\Delta_{GS} = (E - E_{GS})/E_{GS}$); in Figs. 4(d)–4(f) the probability of finding a 10% approximation; in Figs. 4(g)–4(i) the average best energy found in a run. Figures 4(a), 4(d), and 4(g), show the results for the S schedules, Figs. 4(b), 4(e), and 4(h) show the results for the SA schedules, and Figs. 4(c), 4(f), and 4(i) show the results for the PT schedules. The standard schedule bars are colored in shades of orange, while constrained ones are in shades of blue. The schedule with the best average energy is circled in red.

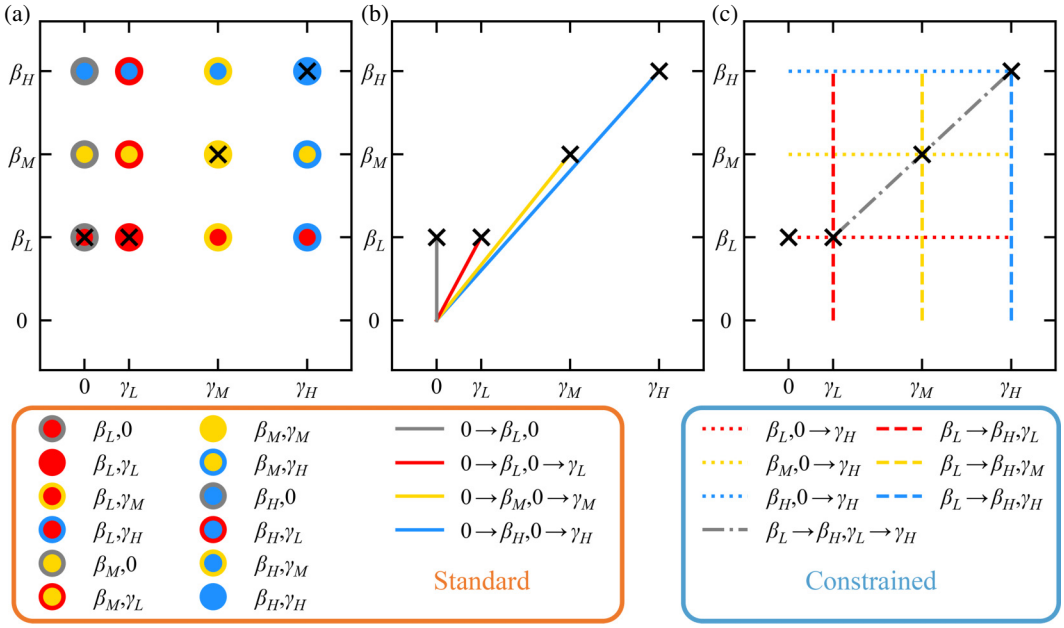


FIG. 3. Schedules chosen for the comparison between energy-minimization algorithms on the TSP instance burma14 shown in a γ-β space. β_L, β_M, β_H, γ_L, γ_M, and γ_H are obtained through the parameter optimization study reported in Fig. 2; the four working points are marked by black crosses. (a) Sampling schedules, each identified by a unique marker. (b) Linear temperature ranges for SA and PT, each represented by a full line with different colors. (c) Linear temperature ranges for CSA and CPT schedules, each identified by a line with different color or style. Standard (constrained) schedules are framed in orange (blue).

These results show that PT schedules perform generally better, even when we account for the number of replicas, with the single best schedule being the one that keeps β fixed at an optimal value for all replicas and has γ vary from 0 to a high value.

To further consolidate these results, we performed a comprehensive study with different topologies and problem sizes. We generated random instances with increasing size, from five nodes to 12 nodes, ten instances per size, and found the optimal route length using a well-established complete solver, CONCORDE.

The best schedules for each paradigm are selected from those shown in Fig. 4, for a total of ten schedules. To compare them, rather than using their average performance on a given metric, we used a score system to account for the inconsistent difficulty of instances with different topologies. All ten schedules are used on each instance for 100 runs of 10^5 sweeps each. S and SA schedules are run with a single replica, and the estimated probabilities are used to determine the probabilities for 5, 10, and 20 replicas. PT schedules are run with the three replica sets. For all compared metrics, the results for each schedule are ranked from best to worst and are assigned points on the basis of their rank for each instance. The first ranked is assigned 10 points, the second 9 points, and so on (ties are considered to have the same rank). The results are shown in Fig. 5. Figures 5(a)–(c) show the ranking on the probability of finding the absolute ground state with 5, 10, and 20 replicas, respectively. Figure 5(d) shows the probability of finding a 5% approximation of the ground state with 20 replicas.

The $\beta_M, 0 \leftrightarrow \gamma_H$ CPT schedule is consistently the best performing one, with the trend that PT schedules perform generally better than S and SA schedules, and that constrained schedules perform better than standard ones. The full results, showing the rankings with all three replica sets and with four approximation metrics, as well as the same tests on larger instances, are provided in Supplemental Material [28].

VII. PIM IMPLEMENTATION OF THE TSP WITH CIRCULAR NEIGHBORHOODS

An instance of the TSP with circular neighborhoods (TSPCNs) is characterized by a set of nodes and a set of radii. Each node is thus identified as a circle with its corresponding radius, as illustrated in an example in Fig. 6(a). While the goal is still to find the shortest path that visits all nodes and returns to the initial position, any point of a node's circle is valid. It is noteworthy that in the limit in which the average distance d is much greater than the average radius r, a TSPCN instance effectively reduces to a standard TSP instance.

TSPCNs extends the classical TSP by allowing continuous-valued positions along the contour of each

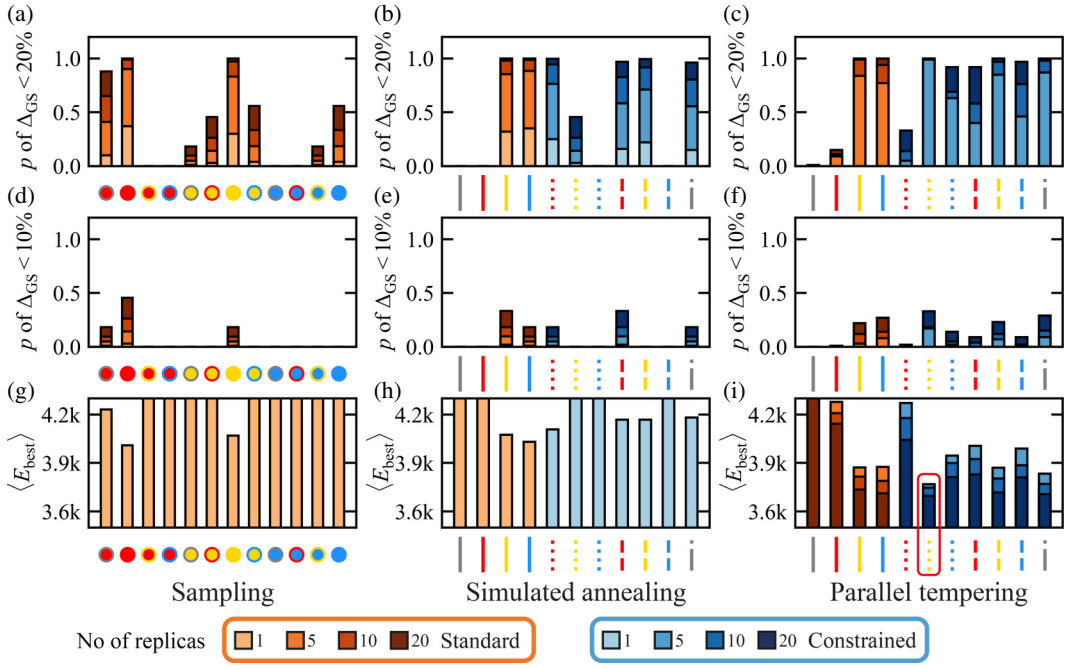


FIG. 4. Comparison between energy-minimization algorithms on the TSP instance burma14. Each schedule's metrics are represented by a bar with a tick referring to the legend in Fig. 3. The rows show the performance in terms of the probability of finding a route within (a)–(c) 20% and (d)–(f) 10% from the ground state, and (g)–(i) the average best energy. (a),(d),(g) Sampling, (b),(e),(h), simulated annealing, and (c),(f),(i) parallel tempering. Standard (constrained) schedule results are shown in orange (blue). The number of replicas used is reflected by the shade of the color.

node, requiring a generalization beyond the standard discrete IM formulation of the TSP. To address this, we propose a pipeline that maps a standard TSP instance to an approximate TSPCN solution, which we refer to as the "TSP → TSPCN algorithm." After a desirable route has been achieved by the solving of the TSPCN instance as a standard TSP, for each node, the intersection of the TSP path with the corresponding circular neighborhood is computed. The nearest midpoint on the circumference is then selected, and these points, ordered according to the TSP solution, form the TSPCN trajectory. Although this method does not guarantee an optimal TSPCN solution, since the optimal visit order for the TSP may differ from that of the TSPCN, it offers a practical compromise. Since the TSPCN application prioritizes fast, energy-efficient solution generation over exact optimality, this approach is well-suited for real-time and resource-constrained scenarios. As illustrated in Fig. 6(b), the process begins with a conventional TSP instance, lacking radius information, that is solved by a PIM to obtain the TSP ground state, followed by the TSP → TSPCN algorithm.

To fine-tune the TSPCN state, we use a MC-like algorithm, MC-TSPCN. The TSPCN state from the previous step is mapped from the coordinates of the points on the circumferences to the angles they form when their radii intersect with the x axis. For each of these angles, a small, randomly generated variation is proposed. If the variation leads to a decrease in the route length, it is accepted; otherwise it is rejected, and the algorithm moves on to the next node. To achieve convergence faster, we use a simple linear annealing schedule controlling the range of the sampled angle. If we use uniform sampling, we decrease the sampling range linearly from an initial $[-0.2, +0.2]$ rad down to 0. With this schedule, convergence is reached in fewer than 100 sweeps of the nodes. If the small angle is instead sampled from a Gaussian distribution, with standard deviation starting from 0.2 and linearly decreasing to 0, convergence is faster (only 40 sweeps are required to minimize consistently).

In Fig. 6(c), the solution of a 20-node graph taken from Ref. [29] and encoded with 400 p-bits is shown in its TSP solution (in red), its TSP → TSPCN conversion (in blue), and its optimized MC-TSPCN state (in green). The route length is improved at each step, reaching a near optimal value [29].

VIII. GAUSSIAN-DISTRIBUTED RANDOM NUMBERS WITH PIMS

The generation of a Gaussian distribution is essential for optimization of the convergence time of the MC-TSPCN algorithm, which relies on the generation of small angle perturbations with zero mean. However, single p-bits can generate only uniform distributions. To sample from a

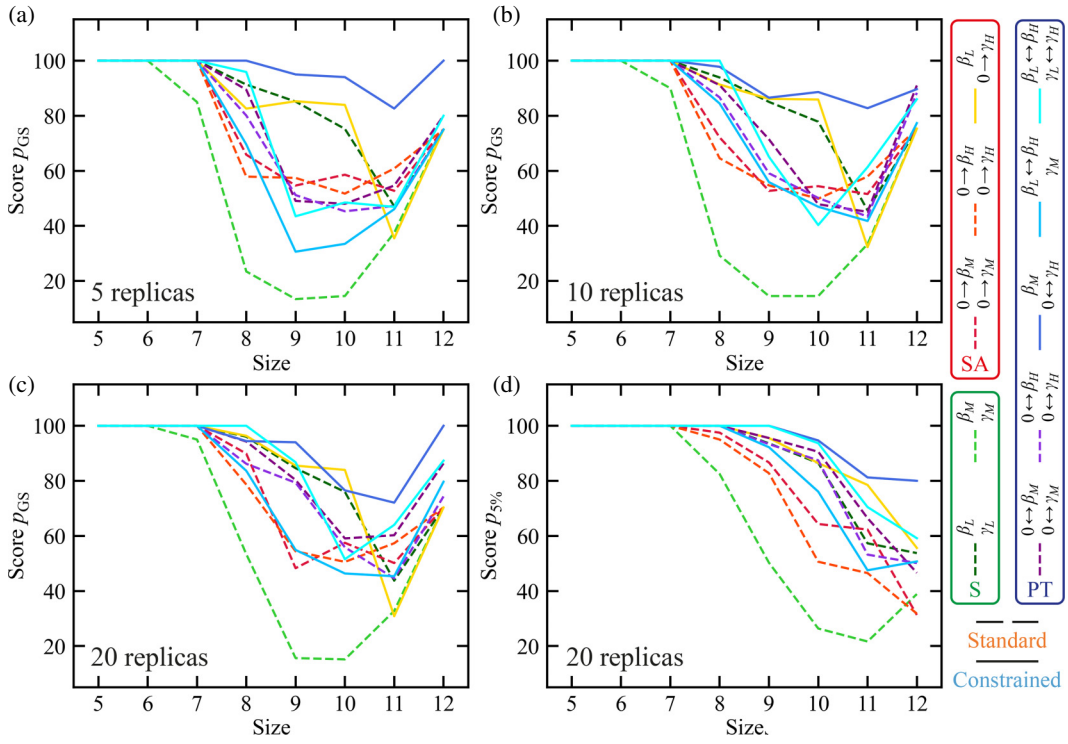


FIG. 5. Comparison between energy-minimization algorithms on randomly generated TSP instances with differing numbers of nodes in terms of score. The score is assigned by ranking the algorithms on the basis of the compared metric on each instance, assigning 10 points to the first ranked, 9 points to the second, and so on. Ties are given the same rank and score. (a)–(c) Performance of the best schedules of each of the three paradigms in Fig. 4 in terms of score for the probability of finding the optimal route with (a) five replicas, (b) ten replicas, and (c) 20 replicas. (d) Performance in terms of score for the probability of finding a state within 5% of the ground state with 20 replicas. Standard (constrained) schedules are represented by dashed (full) curves. S schedules are in shades of green, SA schedules are in shades of red, and PT schedules are in shades of blue.

normal distribution using p-bits, we need to couple them in such a way that the collective behavior enforces constraints on the standard deviation of the distribution. The target Gaussian distribution is given by

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-0.5x^2} . \tag{8}$$

This distribution can be encoded into an IM by our defining the following Hamiltonian:

$$\mathcal{H}_G = -\frac{1}{2}\left(\left(\sum_{i=1}^{N} 2^{i-1}x_i - 2^{-1}(2^N-1)\right) + \sum_{i=1}^{M}\Delta_i(2x_i-1)\right)^2 , \tag{9}$$

with $x_i \in \{0,1\}$. The first term inside the square encodes a Gaussian distribution with mean $\mu = 0$ and standard deviation $s = 1$, sampled on 2^N possible values. These values are half-integers, and they can be obtained by our considering their binary representation using the states of the binary spins, i.e., $v \in B_N$ with $B_N = \left\{\left(\sum_{i=1}^{N} 2^{i-1}x_i - 2^{-1}(2^N-1)\right)\ i \in \{1,\cdots,N\}\right\}$ and $x_i = \{0,1\}$. This term is critical because it ensures proper normalization of the Boltzmann distribution. Specifically, we have $\lim_{N\to\infty}\sum_{n\in B_N} e^{-0.5n^2} = \sqrt{2\pi}$. Given a choice of N, the PIM samples each possible value with probability $\int_{v-\frac{1}{2}}^{v+\frac{1}{2}} f(x)dx$. Figure 7(a) shows an example of the sampling process with a PIM with $N = 3$ (the blue bars) with 1×10^5 sweeps. In this case, eight possible values are sampled from the Gaussian distribution. As each integration interval is exactly 1 in this case, the probability of visiting each of the eight values numerically matches the values of $f(x)$ at those values [the dashed line in Fig. 7(a)]. The J_0 matrix used for the simulation, derived from Eq. (9), is shown in the inset on the right.

The second term inside the square in Eq. (9) introduces additional values, beyond the set of half-integers B_N by adding M auxiliary p-bits to the first N. As $\sigma_i = (2x_i - 1)$ is either -1 or $+1$, each Δ_i can be added or subtracted, shifting the Gaussian distribution by $\pm\Delta_i$. This expands the range of values generated by a factor of 2^M. To avoid degeneracy and have the sampled values spaced with a

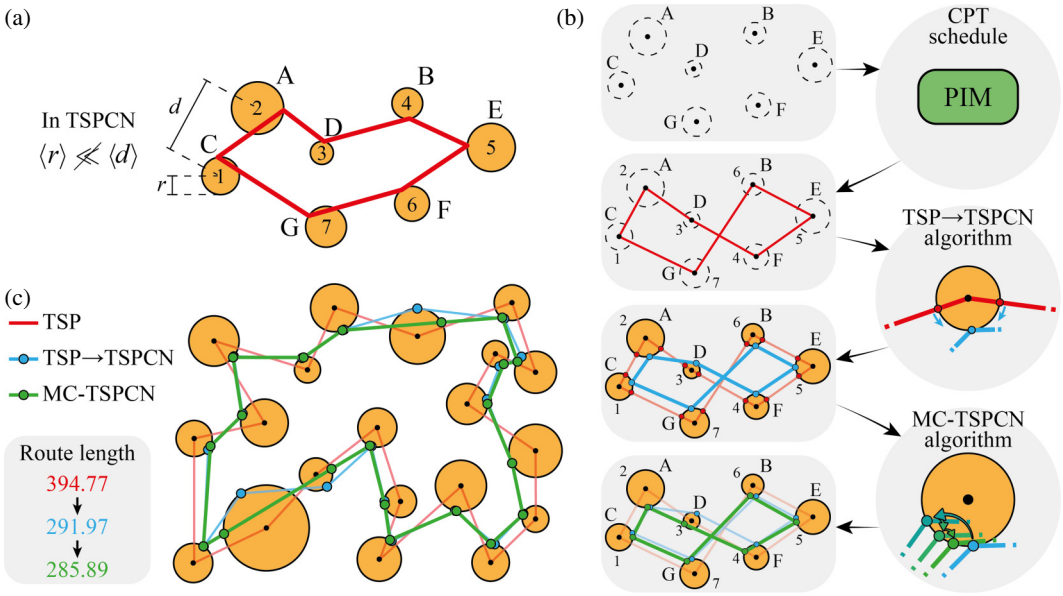


FIG. 6. (a) Example graph of a TSPCN instance. (b) PIM approach to solve a TSPCN instance. The instance is initially solved as a standard TSP with the PIM using CPT; then a deterministic algorithm converts the TSP state to a TSPCN state; finally, a MC algorithm nudges the points into their optimal configuration. (c) Example of a 20-node TSPCN instance from Ref. [29] solved by the method outlined in (b). The red line represents the TSP path, the blue line represents the TSP → TSPCN path, and the green line represents the MC-TSPCN path.

fixed interval, it is necessary to choose $\Delta_i = \sum_{j=1}^{M} 2^{-j}$. In Fig. 7(a), the orange bars show the sampling results for a PIM with $N = 3$ and $M = 1$; J_1 is shown in the inset on the right. The number of possible values is $2^{N+M} = 16$, and $\Delta = 1/4$ to ensure the sampled values are at a constant distance.

Figure 7(b) illustrates the resulting distribution generated with this Ising model formulation with $N = 3$ and $M = 3$ with a total of $2^6 = 64$ possible values. The dashed black line is the Gaussian probability obtained by integration over each possible value v according to $\int_{v-2^{-1-M}}^{v+2^{-1-M}} f(x)dx$.

Overall, this method provides a promising versatile and compact approach for energy-efficient generation of true random numbers sampled from a Gaussian distribution with tunable precision using IMs.

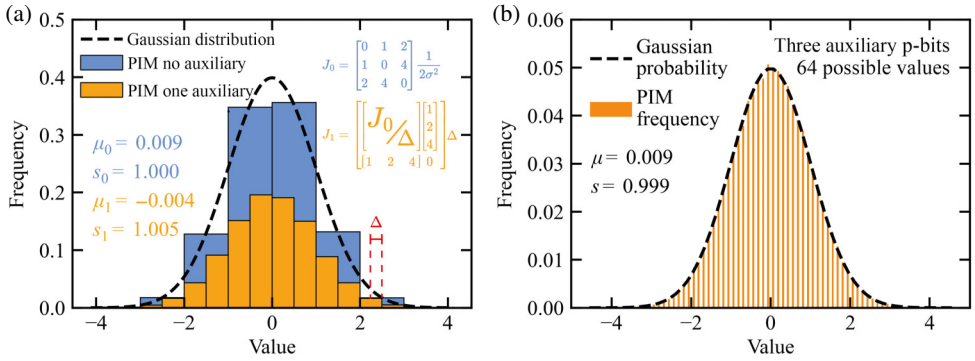


FIG. 7. PIM-based normally distributed random number generation. (a) Sampling of a PIM with $N = 3$ half-integer p-bits is shown with no auxiliary p-bits, $M = 0$, (in blue) and one auxiliary p-bit, $M = 1$, (in orange). With $M = 0$ (the J_0 matrix is shown in the inset on the right), the PIM, samples eight possible values from a standard Gaussian distribution with $\mu = 0$ and $s = 1$ (the probability density is shown as a dashed black line). With $M = 1$ (J_1 is shown in the inset on the right), the PIM samples 16 possible values, each at $\pm\Delta$ from the sampled values at $M = 0$. (b) Sampling results for the PIM (orange bars) with $N = 3$ and $M = 3$. The dashed line is the probability obtained by integration of the Gaussian distribution over the sampled values.

IX. SUMMARY AND CONCLUSIONS

In this work we have introduced the concept of CPT for IMs dealing with COPs with hard constraints, such as the TSP. We show the advantage that CPT provides over standard PT sampling with fixed scaling parameters between the constraint Hamiltonian and the target Hamiltonian and evaluate the efficacy of the algorithm for a medium-sized instance of a TSPCN variant of the TSP. For this, we also developed a heuristic algorithm to adapt the TSP solution provided by our PIM to the TSPCN instance.

One of the most impactful areas for IMs is edge computing, in which the reduced energy consumption and high operation speed of hardware implementations of IMs can be leveraged to perform *in situ* computation on hardware embedded in systems such as drones. In particular, TSPCN is a particularly useful COP to evaluate in this context, as it can be used for networks of interacting moving objects.

An important element to refine is the method in which the optimal array of γ values for CPT are chosen, as the way in which sampling is affected by the varying total energy landscape is different from that in standard PT and requires further attention.

In addition, we propose a method for generating true random numbers with a Gaussian distribution using PIMs. The generated samples exhibit zero mean and unit standard deviation, with tunable precision governed by the number of p-bits used. This allows dynamic control over both the density and the range of the output values, enabling adaptable accuracy tailored to specific application needs. The ability to generate Gaussian-distributed noise natively in hardware is particularly relevant for machine-learning applications, where such distributions are essential, for instance, in diffusion-based generative models, Bayesian neural networks, and regularization schemes that encode uncertainty. By embedding this functionality directly into the PIM architecture, our approach offers a hardware-efficient solution for probabilistic computing tasks, eliminating the need for external random number generators or costly software-based sampling.

ACKNOWLEDGMENTS

This work was supported by the projects PRIN 2020LWPKH7 "The Italian factory of micromagnetic modeling and spintronics" (IT-SPIN), PRIN 20225YF2S4 "Magneto-mechanical accelerometers, gyroscopes and computing based on nanoscale magnetic tunnel junctions" (MMAGYC), PRIN P2022C338Z.PE6 "Development of inclusive quantitative photoacoustic imaging solutions enabled by ethical artificial intelligence" (ImPACT-AI), and PON Capitale Umano (Project No. CIR_00030), all funded by the Italian Ministry for Universities and Research.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available. The data are available from the authors upon reasonable request.

[1] G.-F. Deng and W.-T. Lin, Ant colony optimization-based algorithm for airline crew scheduling problem, Expert Syst. Appl. **38**, 5787 (2011).

[2] Bo Yuan, M. Orlowska, and S. Sadiq, On the optimal robot routing problem in wireless sensor networks, IEEE Trans. Knowl. Data Eng. **19**, 1252 (2007).

[3] K. Castelino, R. D'Souza, and P. K. Wright, Toolpath optimization for minimizing airtime during machining, J. Manuf. Syst. **22**, 173 (2003).

[4] A. Schrijver, in *Discrete Optimization*, edited by K. Aardal, G. L. Nemhauser, and R. Weismantel (Elsevier, Amsterdam, 2005), Vol. 12, pp.1–68.

[5] M. R. Garey and D. S. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness* (W. H. Freeman and Company, San Francisco, 1979).

[6] A. P. Punnen, in *The Traveling Salesman Problem and Its Variations*, edited by G. Gutin, and A. P. Punnen (Springer, Boston, MA, 2007), Vol. 12, pp. 1–28.

[7] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos, in *2016 IEEE International Conference on Smart Cloud (SmartCloud)* (IEEE, 2016), pp. 20–26.

[8] K. Cao, Y. Liu, G. Meng, and Q. Sun, An overview on edge computing research, IEEE Access **8**, 85714 (2020).

[9] G. Finocchio, J. A. C. Incorvia, J. S. Friedman, Q. Yang, A. Giordano, J. Grollier, H. Yang, F. Ciubotaru, A. V. Chumak, A. J. Naeemi, *et al.*, Roadmap for unconventional computing with nanotechnology, Nano Futur. **8**, 012001 (2024).

[10] D. Rodrigues, E. Raimondo, R. Tomasello, M. Carpentieri, and G. Finocchio, A design of magnetic tunnel junctions for the deployment of neuromorphic hardware for edge computing, Appl. Phys. Lett. **126**, 092402 (2025).

[11] N. Mohseni, P. L. McMahon, and T. Byrnes, Ising machines as hardware solvers of combinatorial optimization problems, Nat. Rev. Phys. **4**, 363 (2022).

[12] S. Chowdhury, A. Grimaldi, N. A. Aadit, S. Niazi, M. Mohseni, S. Kanai, H. Ohno, S. Fukami, L. Theogarajan, G. Finocchio, *et al.*, A full-stack view of probabilistic computing with p-bits: Devices, architectures, and algorithms, IEEE J. Explor. Solid-State Comput. Devices Circuits **9**, 1 (2023).

[13] A. Grimaldi, L. Sánchez-Tejerina, N. Anjum Aadit, S. Chiappini, M. Carpentieri, K. Camsari, and G. Finocchio, Spintronics-compatible approach to solving maximum-satisfiability problems with probabilistic computing, invertible logic, and parallel tempering, Phys. Rev. Appl. **17**, 024052 (2022).

[14] Y. Shao, C. Duffee, E. Raimondo, N. Davila, V. Lopez-Dominguez, J. A. Katine, G. Finocchio, and P. Khalili Amiri, Probabilistic computing with voltage-controlled dynamics in magnetic tunnel junctions, Nanotechnology **34**, 495203 (2023).

[15] C. Delacour, M. M. H. Sajeeb, J. P. Hespanha, and K. Y. Camsari, Two-dimensional parallel tempering for constrained optimization, Phys. Rev. E **112**, L023301 (2025).
[16] R. Roberti and M. Ruthmair, Exact methods for the traveling salesman problem with drone, Transp. Sci. **55**, 315 (2021).
[17] P. A. Tu, N. T. Dat, and P. Q. Dung, in *Proceedings of the Ninth International Symposium on Information and Communication Technology - SoICT 2018* (ACM Press, New York, New York, USA, 2018), pp. 46–53.
[18] E. Ising, Beitrag Zur Theorie des Ferromagnetismus, Zeitschrift Für Phys. **31**, 253 (1925).
[19] A. Lucas, Ising formulations of many NP problems, Front. Phys. **2**, 1 (2014).
[20] R. E. Burkard, V. G. Deineko, R. van Dal, J. A. A. van der Veen, and G. J. Woeginger, Well-solvable special cases of the traveling salesman problem: A survey, SIAM Rev. **40**, 496 (1998).
[21] Y. Bao, S. Yang, Z. Yao, and H. Yang, Computing with magnetic tunnel junction based sigmoidal activation functions, Appl. Phys. Lett. **124**, 242403 (2024).
[22] Y. Liu and J. Xiao, Application of probabilistic bits in measurement and sensing, Appl. Phys. Lett. **126**, 122403 (2025).
[23] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, Stochastic p-bits for invertible logic, Phys. Rev. X **7**, 031014 (2017).
[24] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Optimization by simulated annealing, Science. **220**, 671 (1983).
[25] R. H. Swendsen and J.-S. Wang, Replica Monte Carlo simulation of spin-glasses, Phys. Rev. Lett. **57**, 2607 (1986).
[26] J. Si, S. Yang, Y. Cen, J. Chen, Y. Huang, Z. Yao, D.-J. Kim, K. Cai, J. Yoo, X. Fong, and H. Yang, Energy-efficient superparamagnetic Ising machine and its application to traveling salesman problems, Nat, Commun. **15**, 3457 (2024).
[27] E. Raimondo, E. Garzón, Y. Shao, A. Grimaldi, S. Chiappini, R. Tomasello, N. Davila-Melendez, J. A. Katine, M. Carpentieri, M. Chiappini, *et al.*, High-performance and reliable probabilistic Ising machine based on simulated quantum annealing, Phys. Rev. X **15**, 041001 (2025).
[28] See Supplemental Material http://link.aps.org/supplemental/10.1103/plr8-xys1 for full results comparing the algorithms in terms of score.
[29] A. Nedjatia and B. Vizvárib, Robot path planning by traveling salesman problem with circle neighborhood: modeling, algorithm, and applications, arXiv preprint arXiv:2003.06712 (2020).