



OPEN

Quantum annealing for inverse kinematics in robotics

Hadi Salloum^{1,2,3✉}, Sergei Savin², Yaroslav Kholodov^{1,3}, Gleb Ryzhakov⁴, Mirko Farina^{5,6✉} & Ivan Oseledets⁷

We study a proof-of-concept workflow that reformulates planar inverse kinematics (IK) for robotic manipulators as a Quadratic Unconstrained Binary Optimization (QUBO) using a linear binary discretization of joint angles and one-hot (big-M) constraints, and then solves the QUBO with quantum annealing on D-Wave hardware. We (i) define and report time-to-solution (TTS) rigorously, (ii) evaluate solution accuracy back in the original IK space via end-effector error and feasibility of one-hot constraints, and (iii) analyze embedding choices on Pegasus/Zephyr. On the tested two-link planar IK instances, Global Embedding on Zephyr uses fewer physical qubits and yields shorter access times than alternatives (see Tables 1,2), and a hybrid quantum-classical solver achieves lower TTS than classical QUBO baselines at larger Q sizes (see Figure 6). These results do not claim superiority over state-of-the-art continuous IK solvers; rather, they demonstrate that an IK-to-QUBO mapping can be executed on current quantum annealing systems with quantified accuracy and runtime, clarifying where such an approach may be useful as hardware improves.

Inverse Kinematics (IK) maps task-space targets to joint-space configurations for robotic manipulators. While closed-form solutions exist for some geometries, general IK is typically non-convex and solved numerically. With good initial guesses, local methods—e.g., gradient-based solvers, SQP, or related techniques—perform well^{1–3}; without such guesses, global search can be desirable. Mixed-integer convex programming (MICP) offers a principled way to address local minima and to integrate IK with other constraints⁴, but worst-case runtimes can be prohibitive as the number of integer variables grows, limiting practicality for larger, time-sensitive problems.

Many robotics problems exhibit NP-hard characteristics, motivating approximation and heuristic approaches when exact solutions are computationally prohibitive^{5–8}. In parallel, quantum computing has emerged as an alternative computational paradigm with potential advantages for certain classes of optimization problems^{9–11}. Quantum annealing (QA), implemented in devices such as D-Wave’s quantum annealers¹⁷, provides hardware-assisted sampling of low-energy configurations in QUBO/Ising models.

This paper studies a proof-of-concept workflow that reformulates planar IK as a binary Quadratic Unconstrained Binary Optimization (QUBO): we discretize joint angles via a linear binary approximation and enforce one-hot selections with quadratic penalties. The resulting QUBO is solved using QA, and decoded joint angles are evaluated back in the original IK space. Our goals are to (i) characterize runtime with a *formal time-to-solution (TTS)* metric, (ii) quantify *solution quality in IK space* (end-effector error and one-hot feasibility), and (iii) study embeddings on Pegasus/Zephyr hardware. The pipeline follows the stages illustrated in Fig. 1.

Scope and claims. We position this work as a *hardware-validated, reproducible baseline* for $\text{IK} \rightarrow \text{QUBO}$ on current annealers. Throughout this paper, all time-to-solution (TTS) comparisons are performed within the QUBO-solver family (QA, SA, PROTES, hybrid). We intentionally do not compare against continuous IK solvers (e.g., TRAC-IK, SQP-based methods), and we therefore do not claim any runtime advantage over state-of-the-art continuous IK algorithms. Our goal is instead to quantify the behavior of different QUBO solvers on the same $\text{IK} \rightarrow \text{QUBO}$ formulation.

Motivation

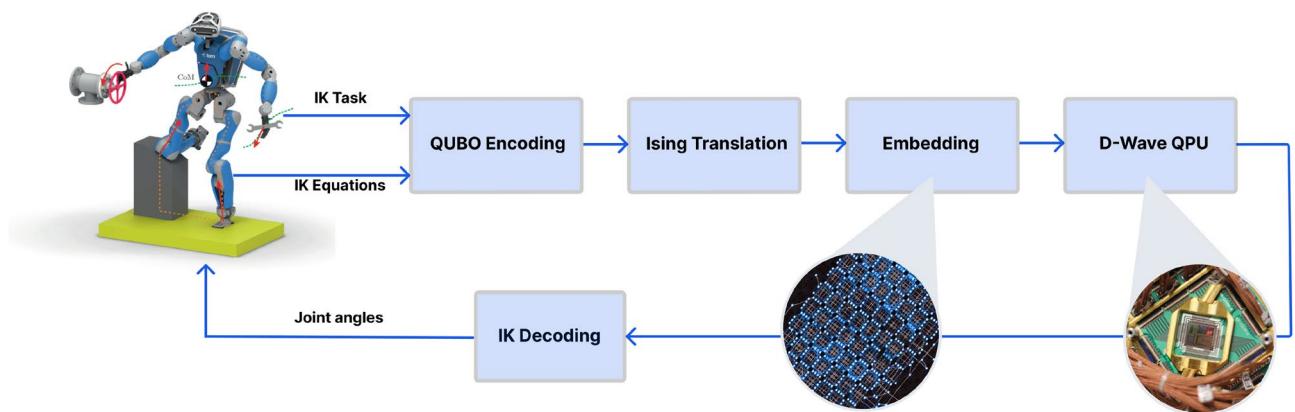
Casting IK into a *binary* optimization model provides access to mature algorithms and hardware ecosystems tailored to QUBO/Ising forms, including QA. Theoretically, QA can exhibit advantages over classical simulated annealing (SA) for certain energy landscapes¹³, and empirical reports show instance-dependent benefits

¹Phystech School of Applied Mathematics and Computer Science, Moscow Institute of Physics and Technology, Institutsky lane 9, Dolgoprudny, Moscow region 141700, Russia. ²Research Center for Artificial Intelligence, Innopolis University, Innopolis 420500, Russia. ³Q Deep, Innopolis 420500, Russia. ⁴Central University, Gasheka street 7, Moscow 123056, Russia. ⁵School of Philosophy and Social Development, Huaqiao University, 668 Jimei Avenue, Jimei District, Xiamen, Fujian Province, PR China. ⁶Ano Neimark, Kirova Prospekt 12, premise 14, Nizhny Novgorod, Nizhegorodskaya Region 603138, Russia. ⁷Artificial Intelligence Research Institute (AIRI), Presnenskaia Naberejnaia 6, building 2, Moscow 123112, Russia. ✉email: h.salloum@innopolis.ru; m.farina@hqu.edu.cn

| Logical Qubits | N_{GP} | N_{CP} | N_{GZ} | N_{CZ} |
|----------------|-----------|----------|-----------|----------|
| 10 | 15 | 20 | 12 | 30 |
| 12 | 20 | 36 | 20 | 36 |
| 14 | 27 | 42 | 24 | 42 |
| 16 | 35 | 48 | 31 | 48 |
| 18 | 41 | 54 | 35 | 54 |
| 20 | 50 | 60 | 42 | 60 |

Table 1. Average physical qubits for different embedding techniques in Pegasus and Zephyr.

| Logical Qubits | t_{GP} (ms) | t_{CP} (ms) | t_{GZ} (ms) | t_{CZ} (ms) |
|----------------|---------------|---------------|---------------|---------------|
| 10 | 584.4 | 648.76 | 413.3 | 534.5 |
| 12 | 602.2 | 878.16 | 415.2 | 542.1 |
| 14 | 611.4 | 878.16 | 490.1 | 552.5 |
| 16 | 635.76 | 889.56 | 566.5 | 552.5 |
| 18 | 618.3 | 1013.31 | 492.9 | 560.1 |
| 20 | 540.0 | 1016.7 | 413.8 | 567.7 |

Table 2. Average QPU access time for different embedding techniques in Pegasus and Zephyr (ms).**Fig. 1.** Workflow for solving inverse kinematics using quantum annealing: Translating joint angle equations into QUBO/Ising models, embedding them on a D-Wave quantum processor¹⁷, and decoding the quantum solution into actionable results.

on practical problems^{14,20}. Devices like D-Wave's annealers¹⁷ additionally enable hybrid quantum-classical workflows that interleave QPU sampling with classical improvement steps.

However, binary reformulation introduces design choices that directly affect fidelity to the original IK objective: (i) *discretization* (choice of angle grids) and (ii) *penalty design* (enforcing one-hot feasibility). Our study makes these trade-offs explicit and measurable: we give a safe penalty-setting recipe (Section 3.5), evaluate decoded solutions in IK space (Section "IK-Space solution quality"), and use a consistent TTS definition for all solvers (Section "Time-to-Solution (TTS) Metric").

Novelty

The novelty is an *end-to-end, accuracy-aware, hardware-validated* IK → QUBO pipeline. Concretely, we:

- implement a linear binary discretization with one-hot penalties for planar IK and run it on real QA hardware¹⁷;
- adopt a penalty design that preserves one-hot feasibility while limiting objective distortion (Section 3.5);
- evaluate with *formal* TTS and *IK-space accuracy* rather than energy alone (Sections "IK-Space solution quality"–"Time-to-Solution (TTS) Metric");
- study embeddings/topologies (Pegasus, Zephyr) and discuss solver behavior, including non-monotonic SA TTS on these instances (Section 4.6).

This reframing responds to reviewer requests for rigor and scope, positioning the work as a reproducible baseline rather than a claim of end-to-end dominance over continuous IK.

Contributions

1. **IK → QUBO with rigorous metrics.** A complete pipeline for planar IK with *formal TTS* and *IK-space accuracy* (end-effector error, one-hot feasibility), executed on D-Wave hardware¹⁷.
2. **Penalty-design guideline.** A practical, norm-based recipe for setting the one-hot penalty to ensure feasibility without excessive objective warping (Section 3.5).
3. **Embedding study.** An empirical comparison of embeddings/topologies (Global vs. Clique; Pegasus vs. Zephyr) showing when Global/Zephyr is most qubit-efficient for our dense (but non-clique) IK QUBOs.
4. **Runtime behavior analysis.** A discussion of when hybrid solvers reduce *QUBO-level TTS* and why SA can show *non-monotonic* (“zig-zag”) TTS with problem size on these instances (Section 4.6), while noting ICE effects when relevant.

Paper structure

The paper is organized as follows: Section “Quantum Annealing” provides an overview of Quantum Annealing in D-Wave quantum processing unit (QPU), discussing the principles and mechanisms underlying the quantum annealing process. Section “Inverse kinematics as a QUBO” introduces QUBO, formulating the IK problem within the framework of Quantum Unconstrained Binary Optimization. Section “Empirical results and analysis for solving IK using quantum annealing” presents Empirical Results and Analysis for Solving IK Using Quantum Annealing. Section “Limitations and shortcomings” discusses Limitations and Shortcomings. Finally, Section “Conclusions” concludes the paper with a summary of key contributions and future research directions.

Quantum annealing

Let us consider a quadratic unconstrained binary optimization (QUBO) problem:

$$\underset{x}{\text{minimize}} \sum_i a_i x_i + \sum_{i>j} b_{ij} x_i x_j, \quad (1)$$

where $x_i \in \{0, 1\}$ are binary optimization variables, a_i are biases and b_{ij} are coupling weights of the optimization problem. This type of problem can be efficiently solved by quantum annealing (QA).

Quantum annealing is a heuristic method for solving combinatorial optimization problems, particularly binary optimization, by controlling a quantum system. It evolved from classical simulated annealing (SA)¹⁶ but replaces thermal fluctuations with quantum fluctuations to drive state transitions as shown in Fig. 2. Kadowaki and Nishimori pioneered this approach¹³, implementing it using a transverse Ising model where a time-varying transverse field acts analogously to temperature changes in SA, aiming for faster convergence to optimal states. This requires special hardware that can support quantum annealing, such as D-Wave’s Advantage quantum processing unit (QPU). It also limits the optimization problems to the form that can be embedded in the given QPU; for example, D-Wave’s QPU are limited to the problems that can be cast as a QUBO¹⁸.

One can see quantum annealing as a process with two stages. The first stage involves embedding the problem and initializing the QPU by mapping logical binary variables to physical qubits (potentially requiring multiple linked qubits to represent one variable) and configuring couplings to match the QUBO structure. The system is initialized in a superposition state with all possible configurations equally likely through a transverse magnetic field. The values of the qubits of the QPU are linked with the values of the binary variables in the original problem; the coupling between qubits is linked with the coupling weights of the original problem, and the external flux bias is linked with the weights of the original problem. On the second stage, the system undergoes controlled quantum evolution: the transverse field is gradually reduced while the problem Hamiltonian (encoding the QUBO) is increased, allowing the system to explore low-energy states quantum-mechanically and settle into an optimal or near-optimal configuration. Lastly, we read the state of the QPU and recover the solution to the

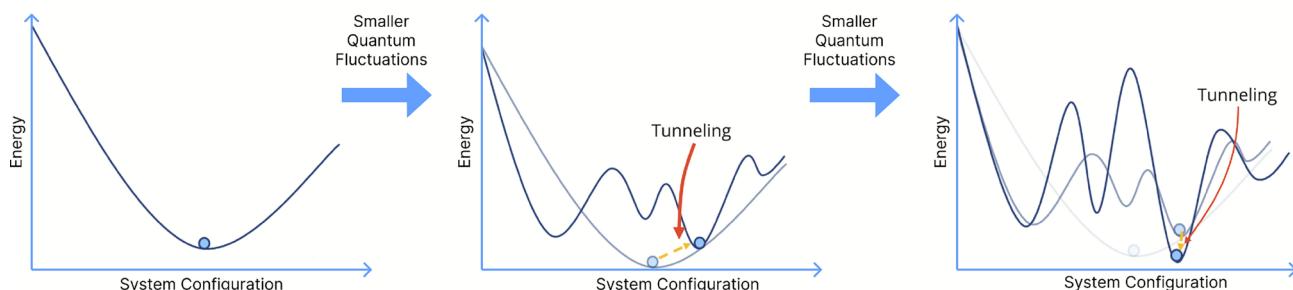


Fig. 2. Schematic representation of the adiabatic quantum annealing process, illustrating how a system transitions from an initial, relatively simple potential landscape to a more complex one while quantum fluctuations are gradually reduced. The red arrows denote quantum tunneling events, through which the system can escape local minima and ultimately converge on the global minimum energy configuration.

original problem. The rest of the section describes these stages in more detail. Fig. 3 gives an illustration of the described process.

Embedding QUBO

The QUBO problem must be mapped to the physical hardware graph of the quantum processor. Since D-Wave's Pegasus architecture has limited connectivity (each qubit connects to 15 others at most), complex QUBO interactions require *minor embedding*. This involves: (1) Identifying the problem's logical graph structure, (2) Finding subgraphs (chains) in the hardware graph that can represent single logical variables, and (3) Setting strong ferromagnetic couplings ($J_{\text{chain}} < 0$) between chained qubits to ensure they remain correlated.

Annealing

Ising Hamiltonian that represents the embedded optimization problem during quantum annealing can be written as¹⁹:

$$H(s) = -\frac{A(s)}{2} \sum_i \sigma_x^{(i)} + \frac{B(s)}{2} \left(\sum_i h_i \sigma_z^{(i)} + \sum_{i>j} J_{ij} \sigma_z^{(i)} \sigma_z^{(j)} \right)$$

where:

- J_{ij} is the coupling strength between qubits i and j (related to b_{ij} in the QUBO formulation);
- h_i is the external flux bias for qubit i (related to a_i in the QUBO formulation);
- $\sigma_x^{(i)}, \sigma_z^{(i)}$ are Pauli matrices operating on the qubit i in the x and z directions, respectively;
- $s \in [0, 1]$ is the annealing parameter representing the evolution of the system from the initial state $s = 0$ to the final state at $s = 1$.

At $s = 0$, the system is in the quantum ground state dominated by H_{initial} ($A(0) \gg B(0)$). As s progresses, $A(s)$ decreases while $B(s)$ increases, transitioning the system into a classical regime. At $s = 1$, the final Hamiltonian describes a classical Ising spin model.

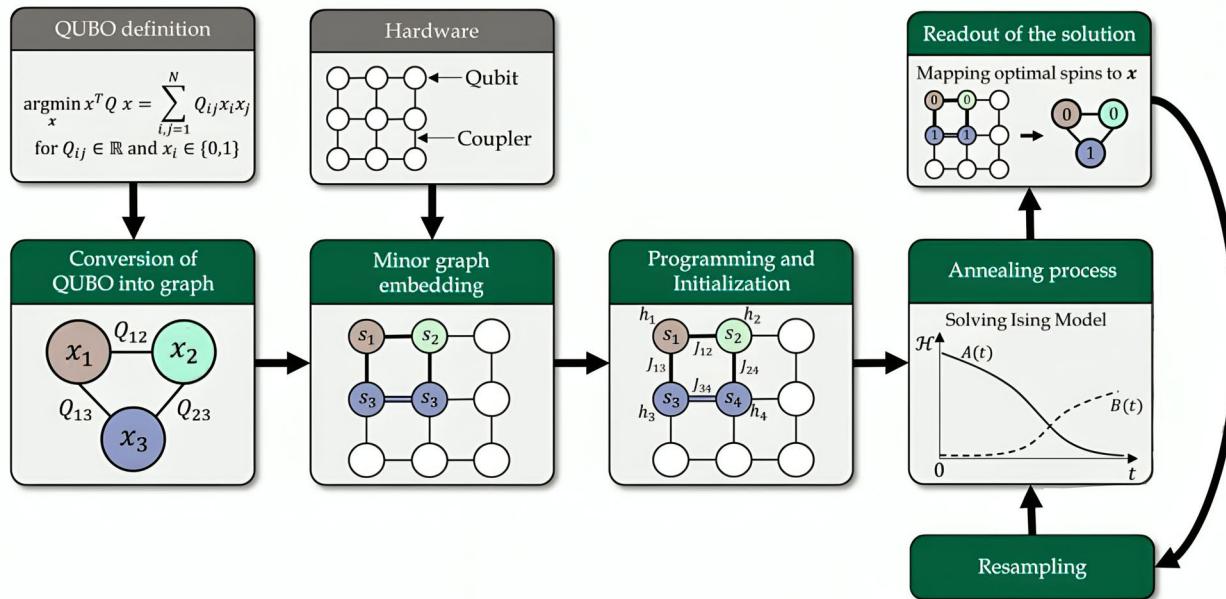


Fig. 3. Overview of the quantum annealing workflow for solving Quadratic Unconstrained Binary Optimization (QUBO) problems: (1) QUBO formulation $\arg\min_x x^T Q x$ with binary variables $x_i \in \{0, 1\}$ and conversion QUBO into graph (Ising model) (2) Hardware components (qubits, couplers) and minor graph embedding, (3) Programming and initialization of annealing parameters ($A(t)$: transverse field, $B(t)$: longitudinal field, $J(t)$: coupling strength), (4) Quantum annealing dynamics evolving spins from quantum superposition to classical states, (5) Readout and mapping of optimal spin configurations to binary solution x . This overview illustrates how QUBO problems are solved via quantum annealing in Fig. 1, demonstrating the inverse Ising mapping process from combinatorial optimization to physical quantum implementation. Figure adapted from¹⁵ and edited by us.

For Solving the IK, it must first be converted into a QUBO formulation. This is then transformed into an Ising model. Afterward, by applying minor embedding, we embed the IK problem onto the Pegasus and Zephyr topologies. The Quantum Annealing process in the D-Wave QPU aims to find the ground state of H_p , corresponding to the optimal solution of the optimization problem. By starting with a known ground state of H_{initial} and evolving the system adiabatically through $H(s)$, the process converges to the ground state of H_p , thereby solving the problem efficiently. The entire pipeline, as depicted in the Fig. 1, is followed to decode the results into the required joint angles.

Inverse kinematics as a QUBO

Linear binary approximation

We introduce a binary selector vector $q \in \mathbb{B}^m$, for which one and only one of its elements $q_i \in \{0, 1\}$ is equal to 1:

$$\sum_{i=1}^m q_i = 1. \quad (2)$$

Given a sequence of angles $0 \leq \varphi_1 < \varphi_2 < \dots < \varphi_m \leq 2\pi$ we can find associated values of the trigonometric functions $\cos(\varphi)$ and $\sin(\varphi)$ and arrange them as elements of a vector:

$$t_1 = [c_1 \dots c_m]^T, \quad c_i = \cos(\varphi_i), \quad (3)$$

$$t_2 = [s_1 \dots s_m]^T, \quad s_i = \sin(\varphi_i); \quad (4)$$

we call t_1, t_2 value vectors. With that, we can introduce a linear binary approximation (LBA) of the cosine and sine functions:

$$\cos(\varphi) \approx t_1^T q, \quad \sin(\varphi) \approx t_2^T q. \quad (5)$$

LBA is exact for the angles φ_i that were used to compute elements of the value vectors t_1, t_2 . For any selector vector q the fundamental identity holds:

$$(t_1^T q)^2 + (t_2^T q)^2 = 1. \quad (6)$$

For the two-link case, the full selector vector consists of two disjoint one-hot blocks of size m , each enforcing selection of a single candidate angle per joint.

Inverse kinematics of a planar serial linkage

Consider a planar kinematic chain with n links connected via rotary joints. Let the length of the j -th link be l_j and its orientation in the global frame be defined by the angle φ_j . Then the Cartesian coordinates (r_1, r_2) of the end effector placed at the end of the n -th link are given by the following expression:

$$r_1 = \sum_{j=1}^n l_j \cos(\varphi_j), \quad r_2 = \sum_{j=1}^n l_j \sin(\varphi_j). \quad (7)$$

To find angles φ_j^* corresponding to the desired position r_1^*, r_2^* of the end-effector we solve the following optimization problem:

$$\varphi_j = \underset{\varphi_j}{\operatorname{argmin}} (r_1^* - \sum_{j=1}^n l_j \cos(\varphi_j))^2 + (r_2^* - \sum_{j=1}^n l_j \sin(\varphi_j))^2 \quad (8)$$

This represents an inverse kinematics (IK) problem cast as a nonconvex optimization with continuous variables. In the next subsection, we approximate it as a quadratic binary optimization problem.

Inverse kinematics as a QUBO

To approximate the IK problem described in the last subsection, we use n linear binary approximations $\cos(\varphi_j) \approx t_1^T q_j$ and $\sin(\varphi_j) \approx t_2^T q_j$, each associated with its selector vector $q_j \in \mathbb{B}^m$. Concatenation of all selector vectors is denoted as $\bar{q} = [q_1^T, \dots, q_n^T]^T$.

For convenience, we define vector $d = [l_1, \dots, l_n]^T$. The resulting optimization problem takes the form:

$$\underset{\bar{q}}{\operatorname{minimize}} \sum_{k=1}^2 (\bar{q}^T ((dd^T) \otimes (t_k t_k^T)) \bar{q} - 2r_k (d \otimes t_k)^T \bar{q}) \quad (9)$$

where \otimes is a Kronecker product.

To ensure that every vector q_j abides by the selector constraints (2) we introduce a big-M penalty function $p(q_j)$:

$$p(\mathbf{q}_j) = M(1_m^T \mathbf{q}_j - 1)^2 \quad (10)$$

where $1_m \in \mathbb{R}^m$ is a vector of ones and $M \gg 0$ is a sufficiently big number. With big-M penalty, the optimization takes the form:

$$\begin{aligned} \underset{\bar{\mathbf{q}}}{\text{minimize}} \sum_{k=1}^2 & \left(\bar{\mathbf{q}}^T \left((\mathbf{d}\mathbf{d}^T) \otimes (\mathbf{t}_k \mathbf{t}_k^T) \right) \bar{\mathbf{q}} - 2r_k (\mathbf{d} \otimes \mathbf{t}_k)^T \bar{\mathbf{q}} \right) \\ & + M \left(\bar{\mathbf{q}}^T \left(\mathbf{I} \otimes (1_m 1_m^T) \right) \bar{\mathbf{q}} - 2(1_m \otimes 1_m)^T \bar{\mathbf{q}} \right) \end{aligned} \quad (11)$$

Note that the constant components of the cost function are omitted, as they do not influence the optimal choice of the decision variable, denoted as $\bar{\mathbf{q}}$. The resulting problem is formulated as a Quadratic Unconstrained Binary Optimization (QUBO), which is compatible with quantum annealer hardware, such as the D-Wave Advantage. This problem will be embedded into a specific topology for an Ising model, where the variables in the QUBO are mapped (i.e. minor embedded³¹) to physical qubits, as illustrated in Fig. 4.

In our IK formulation, the big- M penalty is used exclusively to enforce two independent one-hot constraints: selecting exactly one angle from the first n candidate angles (joint 1) and exactly one angle from the second n candidate angles (joint 2). Since all angle candidates lie within fixed joint limits, the maximum possible difference between any candidates is bounded by 2π . Therefore, the required M is a small constant that does not grow with problem size. This ensures that the penalty does not distort the objective landscape and preserves the fidelity of the IK \rightarrow QUBO mapping, unlike general big- M constructions that may require exponentially large constants.

Higher-order inverse kinematics problems

Only a subset of inverse kinematics problems resolves to a QUBO form under the proposed linear binary approximation. Instead, the resulting problem will take a form of a higher-order unconstrained optimization (HUBO), unless a different approximation is used. HUBO problems cannot be solved by the quantum annealer hardware directly, but they can be approximated by a larger (in terms of the number of variables) QUBO problems sharing the same global minimum^{25–28,42}.

Empirical results and analysis for solving IK using quantum annealing

All experiments are based on the same family of planar Two-Linked IK instances and the same linear binary encoding of joint angles. Each link has unit length ($L_1 = L_2 = 1$), and the desired end-effector position (g_x, g_y) is chosen inside the reachable annulus of the manipulator.

For a given problem size $m \in \{5, \dots, 10\}$, each joint angle is discretized into m equidistant samples $\varphi_i = 2\pi(i-1)/m$, $i = 1, \dots, m$, and represented by a one-hot binary vector $x^{(k)} \in \{0, 1\}^m$ for joint

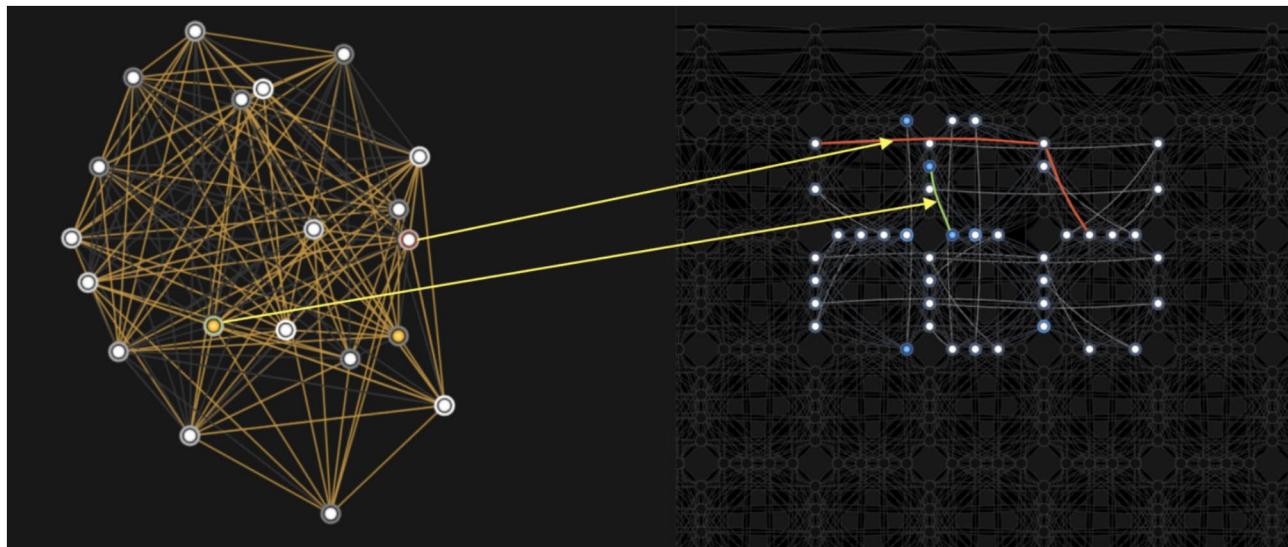


Fig. 4. Mapping a Two-Linked IK problem graph onto a quantum annealer’s hardware architecture—highlighting the embedding of a densely connected problem graph (left) onto the physical qubit connectivity of a D-Wave quantum processing unit (right). The yellow arrows illustrate the correspondence between logical nodes and their mapped qubits, while the red and green edges represent couplings between qubits in the hardware graph. The white nodes on the left figure and the right figure represent zeroes in the QUBO and Ising formulations, respectively, while the yellow and blue nodes represent ones in the QUBO and Ising formulations, respectively.

$k \in \{1, 2\}$. The corresponding unit direction vectors are precomputed as $h_x = (\sin \varphi_i)_i$ and $h_y = (\cos \varphi_i)_i$, and used to construct the quadratic cost in the workspace coordinates.

The continuous objective $\|p(\theta_1, \theta_2) - g\|_2^2$ is then approximated by a quadratic form in the binary variables, $E(x) = x^\top Qx + \text{const}$, where the block-structured matrix $Q \in \mathbb{R}^{2m \times 2m}$ is obtained from $H = h_x h_x^\top + h_y h_y^\top$ together with additional diagonal and off-diagonal penalties that (i) enforce the one-hot constraint for each joint and (ii) encode the linear terms induced by the target coordinates (g_x, g_y) . This construction yields a dense, but not fully connected, QUBO of logical size $2m$, matching the implementation used to generate the matrices submitted to the D-Wave solvers.

To evaluate the performance of solving IK problem reformulated into a QUBO model using quantum annealing, two embedding strategies—**Global Embedding** and **Clique Embedding**³²—were systematically analyzed. These methods were evaluated for their effectiveness in mapping the problem onto quantum annealing hardware. Specifically, the analysis focused on physical qubit utilization and quantum processing unit access times, leveraging two state-of-the-art hardware topologies: “Pegasus”²⁹ and “Zephyr”³⁰. The IK problem addressed was a Two-Linked Inverse Kinematics problem. Problem sizes ranged from $m = 5$ to $m = 10$, corresponding to 10 to 20 logical qubits. Each instance of the QUBO problem required a matrix size of $2m \times 2m$, necessitating $2m$ logical qubits to accurately represent the problem.

- **Global Embedding:** A general-purpose embedding tool which models all constraints collectively and maps the aggregate onto the QPU graph. This approach typically uses fewer qubits and shorter chains, making it more efficient for smaller problems.
- **Clique Embedding:** A specialized embedding technique optimized for fully connected QUBOs.

The IK problem is not fully connected but exhibits dense connectivity, with the number of connections approximated as $\frac{3}{2}m$.

Two quantum annealing hardware topologies were considered:

- **Pegasus Topology** (Advantage System 4.1): Capable of embedding fully connected QUBOs with up to 177 logical qubits.
- **Zephyr Topology** (Advantage Prototype 2.6): Supports embedding fully connected QUBOs with up to 80 logical qubits, featuring higher local connectivity compared to Pegasus.

Qubit usage analysis

The analysis began with the number of physical qubits required to represent logical variables under different embedding techniques and topologies. Table 1 presents the average physical qubit usage for various problem sizes and embedding strategies. The columns represent:

- N_{GP} : Global Embedding on Pegasus
- N_{CP} : Clique Embedding on Pegasus
- N_{GZ} : Global Embedding on Zephyr
- N_{CZ} : Clique Embedding on Zephyr

We observe that N_{GZ} is the most efficient in terms of physical qubit usage. The results are the average of multiple tests conducted for each embedding technique and topology.

QPU access time analysis

In Table 2, we present the average QPU access times for each embedding strategy. The columns represent:

- t_{GP} : Global Embedding on Pegasus
- t_{CP} : Clique Embedding on Pegasus
- t_{GZ} : Global Embedding on Zephyr
- t_{CZ} : Clique Embedding on Zephyr

The results show that t_{GZ} was the best-performing configuration, aligning with the findings from the qubit usage analysis. Global Embedding is better suited for this problem because the QUBO is not fully connected. Global Embedding required fewer physical qubits compared to Clique Embedding, particularly for larger problem sizes. For smaller problem sizes, the dense local connectivity of the Zephyr topology provided a slight advantage in embedding efficiency. Clique Embedding exhibited longer QPU access times due to its general-purpose nature, whereas Global Embedding achieved faster programming but required manual optimization.

Large-scale QUBOs

For large-scale problem instances that exceed the limitations of purely quantum solvers, we utilize the D-Wave Hybrid Solver Service (HSS). This hybrid solver integrates quantum annealing with classical optimization techniques to efficiently process QUBO problems that are too large to be embedded directly on the QPU. The overall workflow is illustrated in Fig. 5.

Mathematical Structure. Let $Q \in \mathbb{R}^{n \times n}$ denote the QUBO matrix. The hybrid solver extends the principles of the open-source `qbsolv` algorithm by decomposing Q into smaller subproblems that fit on the QPU³⁷. For a subset of variable indices $S \subseteq \{1, \dots, n\}$, let Q_S denote the restriction of Q to rows and columns in S , and let x_S be the corresponding partial assignment.

Given a current full assignment $x \in \{0, 1\}^n$, the subproblem optimized at iteration t is the *conditional sub-QUBO*

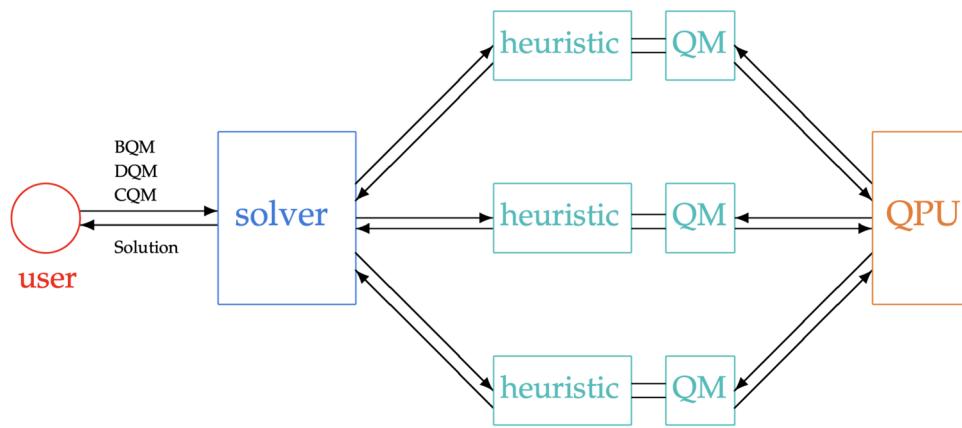


Fig. 5. Structure of a hybrid solver in D-Wave’s hybrid solver service. Adapted from³⁸.

$$E_{S_t}(x_{S_t}; x_{\bar{S}_t}) = x_{S_t}^\top Q_{S_t} x_{S_t} + 2x_{S_t}^\top Q_{S_t, \bar{S}_t} x_{\bar{S}_t}, \quad (12)$$

where \bar{S}_t is the complement of S_t . This formulation fixes variables outside S_t , enabling the subproblem to remain small enough to be embedded onto the QPU.

Hybrid Optimization Loop. The hybrid solver iteratively refines the global solution using a combination of classical neighborhood selection, quantum subproblem optimization, and classical postprocessing:

1. **Neighborhood selection.** A subset S_t is chosen using heuristics such as energy-based gradients, constraint violations, or tabu memories. This selects a promising region of variables to optimize jointly.
2. **Quantum refinement.** The restricted QUBO Q_{S_t} is embedded onto the QPU, where quantum annealing generates K low-energy samples $\{x_{S_t}^{(k)}\}_{k=1}^K$.
3. **Classical postprocessing.** Each quantum sample is merged with the fixed variables:

$$x^{(k)} = (x_{S_t}^{(k)}, x_{\bar{S}_t}).$$

Classical local search (hill climbing, steepest descent, and tabu search) is then applied to refine $x^{(k)}$ and reduce the global energy $E(x^{(k)})$.

4. **Update step.** The next iterate is chosen as

$$x_{t+1} = \arg \min_k E(x^{(k)}),$$

and neighborhood-selection heuristics are updated accordingly.

This scheme can be seen as a large-neighborhood search where quantum annealing acts as a specialized optimizer for selected subregions, while classical heuristics manage global coordination, decomposition, and refinement.

```

1:  $x \leftarrow \text{InitializeSolution}()$ 
2: while not converged do
3:    $S \leftarrow \text{SelectNeighborhood}(x)$ 
4:    $Q_S \leftarrow \text{ExtractSubQUBO}(Q, S)$ 
5:    $\{x_S^{(k)}\} \leftarrow \text{QuantumAnneal}(Q_S)$ 
6:   for each  $x_S^{(k)}$  do
7:      $x^{(k)} \leftarrow \text{Merge}(x_S^{(k)}, x_{\bar{S}})$ 
8:      $x^{(k)} \leftarrow \text{LocalSearch}(x^{(k)})$ 
9:   end for
10:   $x \leftarrow \arg \min_k E(x^{(k)})$ 
11:   $\text{UpdateHeuristics}(x)$ 
12: end while
13:
14: return  $x$ 

```

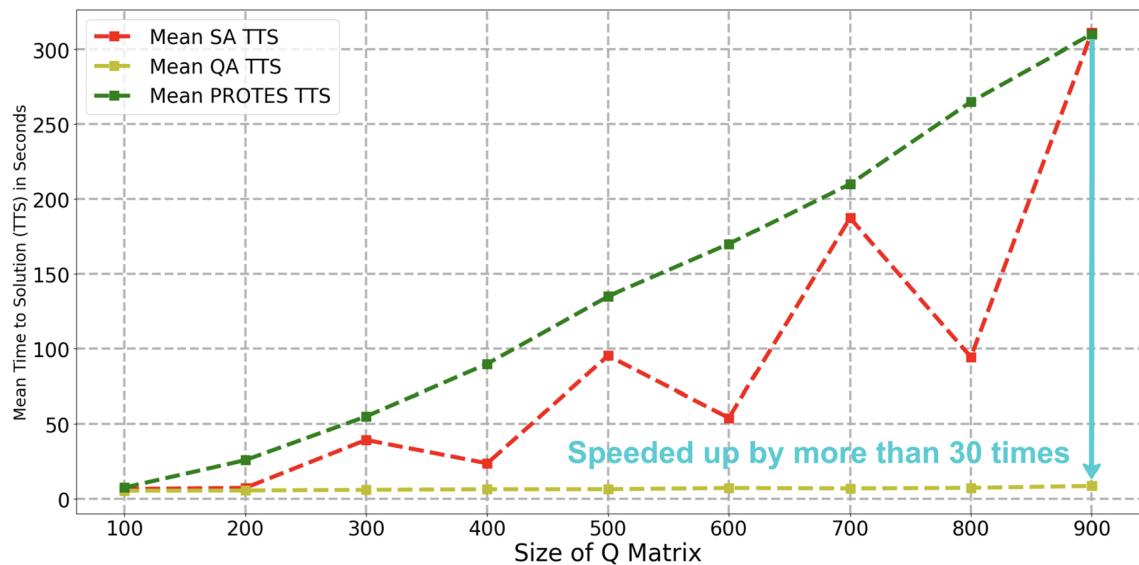


Fig. 6. Mean time-to-solution (TTS) for Quantum Annealing (QA), Simulated Annealing (SA), and PROTES/TT on large-scale two-link IK QUBO instances. A result is counted as successful only if it is IK-correct (satisfies both one-hot constraints and yields end-effector accuracy within the discretization tolerance). SA always finds the exact optimal QUBO energy. QA, when applied to the 500×500 QUBO, typically finds the second-best energy but still yields IK-correct angles. PROTES/TT fails to produce an IK-correct solution for the 300×300 QUBO.

Algorithm 1. Hybrid QA–classical optimization loop

This hybrid strategy enables the solver to handle QUBO instances with thousands of variables—well beyond the direct embedding capacity of the QPU—while still leveraging the quantum annealer’s ability to efficiently explore low-energy regions of complex energy landscapes.

In Fig. 6, the difference in time-to-solution (TTS) among Quantum Annealing (QA), Simulated Annealing (SA)^{39,40}, and the Probabilistic Tensor Train Sampler (PROTES), which employs probabilistic sampling from a probability density function represented in a low-parametric tensor train format⁴¹, is illustrated (In these experiments, number of executed samples in SA and QA were 5000, while the PROTES number of allowed requests to the objective function—was set to 2000). For a QUBO instance of size 900×900 (corresponding to 2^{900} possible solutions), QA demonstrates a speedup of more than 30 times compared to classical methods. SA attains the exact optimal QUBO energy for all tested sizes and always produces IK-correct solutions. QA applied to the 500×500 QUBO typically achieves the second-lowest energy, yet still returns IK-correct joint configurations. In contrast, PROTES/TT applied to the 300×300 QUBO does not produce any IK-correct solution: its lowest-energy configurations violate one-hot consistency or result in significantly larger workspace errors. Therefore, all TTS values in Fig. 6 reflect the time required to obtain an IK-correct solution.

The results in Fig. 6 emphasize the significant advantage of the Hybrid D-Wave Solver in reducing time-to-solution for practical, large-scale optimization problems. The hybrid approach effectively combines the strengths of quantum and classical techniques, making it a robust solution for tackling computationally intensive QUBO instances. However, its performance is limited by the overhead and variability introduced by integrating these two components, which means that for certain problems it may not achieve the optimal solution. Reliance on classical processing can restrict the overall speedup and may prevent consistently achieving the global optimum.

IK-space solution quality

While Fig. 6 compares solver performance in terms of time-to-solution (TTS), it is crucial to verify that the returned binary vectors correspond to *valid joint configurations* in the original IK space. For each solver, we therefore evaluate the following:

1. **One-hot feasibility.** Each joint angle is represented by a one-hot block. We validate that each block contains exactly one active element. Any violation indicates that the big- M penalty failed to enforce angle selection.
2. **End-effector accuracy.** After decoding the joint angles (φ_1, φ_2), we evaluate the forward-kinematics position $p(\varphi_1, \varphi_2)$ and compute the workspace error

$$\|p(\varphi_1, \varphi_2) - g\|_2.$$

A solution is classified as *IK-correct* if both one-hot constraints are satisfied and the error is within the discretization tolerance implied by the angular grid.

Across all QUBO sizes tested in Fig. 6 (from 100×100 up to 900×900), SA consistently attains the optimal QUBO energy (as verified using an exact classical solver), and its decoded joint angles are always IK-correct.

QA produces IK-correct solutions across the entire range of QUBO sizes; however, beginning at QUBO sizes of approximately 500×500 and above, QA typically returns the *second-best* QUBO energy due to the Integrated Control Errors (ICE) effect. Specifically, the dynamic range of h and J values may be limited by integrated control errors (ICE). Instead of finding low-energy states for an optimization problem defined by h and J as in equation 13, the quantum processing unit solves a slightly altered problem that can be modeled as:

$$E_{\delta\text{ising}}(s) = \sum_{i=1}^N (h_i + \delta h_i) s_i + \sum_{i=1}^N \sum_{j=i+1}^N (J_{i,j} + \delta J_{i,j}) s_i s_j, \quad (13)$$

where δh_i and $\delta J_{i,j}$ characterize the errors in the parameters h_i and $J_{i,j}$, respectively³³. Because δh and δJ are summed over N , fidelity limitations tend to have a greater effect on performance for full-QPU-sized problems, for a given dynamic range and distribution of h and J . This can result in slightly different solutions compared to the ideal case. Despite this slight energy deviation, the decoded angles remain fully IK-correct and yield the same end-effector position as SA within discretization accuracy. In contrast, PROTEŠ begins to lose IK correctness once the QUBO size exceeds approximately 300×300 . For QUBO dimensions larger than this threshold, TT's lowest-energy configurations either violate one-hot feasibility or produce noticeably larger workspace errors. Consequently, TT does not yield IK-correct solutions for the larger QUBO sizes reported in Fig. 6.

Time-to-Solution (TTS) metric

To compare stochastic solvers on a common runtime scale, we use the standard *time-to-solution* (TTS) metric. For each solver, a single run has a measured wall-clock time t_{run} (including all overheads such as QPU programming, annealing, and readout for QA, and CPU overhead for classical solvers). A run is deemed *successful* if it returns an IK-correct solution (Sec. 4.4) whose QUBO energy is at most a reference value E_{ref} , chosen as the SA optimum, verified where possible using an exact classical solver. Let p_{succ} denote the empirical success probability.

Assuming independence between runs, the expected number of repetitions needed to achieve at least one success with confidence $1 - \delta$ is

$$N_{\text{TTS}}(\delta) = \frac{\log \delta}{\log(1 - p_{\text{succ}})}. \quad (14)$$

We fix $\delta = 0.01$ (99% confidence), and define

$$\text{TTS} = N_{\text{TTS}}(0.01) t_{\text{run}}. \quad (15)$$

The curves in Fig. 6 report the mean TTS across all two-link IK instances of a given QUBO size. This metric accounts for both the wall-clock cost and the probability of obtaining an IK-correct solution. The TTS curve for simulated annealing (SA) is not strictly increasing; in particular, the TTS at QUBO size 400×400 is slightly lower than at 300×300 . From our definition for TTS in eq 15 with t_{run} essentially constant across these sizes, this simply means that the empirical success probability p_{succ} of SA is slightly *higher* at 400 than at 300. In other words, for our IK-derived instances SA finds the ground state more frequently at size 400, so the expected time to solution is lower, even though the problem is nominally larger. This is not contradictory: the difficulty of SA is governed by the detailed energy landscape (barrier structure, number and width of attraction basins), which changes non-monotonically with the discretization parameter that defines each QUBO, and small local reversals of TTS between neighboring sizes are therefore expected and not interpreted as a fundamental complexity effect^{44–46}.

Limitations and shortcomings

Unlike general mixed-integer IK reformulations, our use of big-M for one-hot angle selection does not require large constants and does not contribute to objective warping. Additionally, while quantum annealing offers compelling advantages for IK optimization, this work faces several constraints inherent to current quantum hardware and methodological choices:

1. Hardware Limitations:

- *Qubit scarcity*: The D-Wave QPU's limited qubit count ($\sim 5,000$ qubits in Advantage systems) and sparse connectivity restrict problem size. For $m = 20$ (requiring 60 physical qubits), scaling to $m > 150$ becomes infeasible due to embedding overhead.
- *Noise and errors*: Integrated Control Errors (ICE) alter the effective Ising Hamiltonian (Eq. 12), causing deviations from optimal solutions.

2. **Approximation Trade-offs**: The linear binary approximation (LBA) introduces discretization errors for non-sampled angles $\varphi \notin \{\varphi_i\}$. High-resolution approximations increase QUBO size quadratically.

3. **Scalability of Embedding**: Global Embedding exhibits quadratic growth in physical qubits ($N_{GZ} \propto N^2$). For $N = 60$, N_{GZ} exceeds 150 qubits—straining current QPUs.

4. **Problem Applicability:** The QUBO reformulation applies only to **planar serial linkages**. Complex kinematics yield higher-order optimizations (HUBO) requiring non-trivial reductions.
5. **Hybrid Solver Overhead:** The $30\times$ speedup relies on classical heuristics, introducing latency without global optimality guarantees.

Conclusions

The application of quantum annealing to IK problem, reformulated as a QUBO, demonstrates the potential of quantum computing for complex optimization in robotics. Among embedding strategies, Global Embedding on Zephyr proved most efficient, offering lower physical qubit usage and faster QPU access times. Hybrid quantum-classical approaches achieved over 30-fold speedups. As discussed in Section "Limitations and shortcomings", future work will address hardware-aware embedding optimizations, error mitigation for ICE, and efficient HUBO-to-QUBO conversions.

Data availability

Researchers may request access to the data from the corresponding author, Hadi Salloum (H.S.), and such requests will be reviewed individually to ensure appropriate and secure data usage.

Received: 11 July 2025; Accepted: 28 December 2025

Published online: 31 December 2025

References

1. Tevatic, G. & Schaal, S. Inverse kinematics for humanoid robots, in Proceedings 2000 ICRA. Millennium Conference. *IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, **1**, 294–299 (IEEE, 2000).
2. Beeson, P. & Ames, B. TRAC-IK: An open-source library for improved solving of generic inverse kinematics, In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 928–935 (IEEE, 2015).
3. Yenamandra, T., Bernard, F., Wang, J., Mueller, F. & Theobalt, C. Convex optimisation for inverse kinematics, In *2019 International Conference on 3D Vision (3DV)*, 318–327 (IEEE, 2019).
4. Dai, H., Izatt, G. & Tedrake, R. Global inverse kinematics via mixed-integer convex optimization, *The International Journal of Robotics Research*, **38**, no. 12–13, 1420–1441 (SAGE Publications, 2019).
5. Erickson, L. & LaValle, S. A simple, but NP-hard, motion planning problem. *Proceedings of the AAAI Conference on Artificial Intelligence*, **27**(1), 1388–1393 (2013).
6. Solovey, K. *Complexity of planning*. arXiv preprint [arXiv:2003.03632](https://arxiv.org/abs/2003.03632), (2020).
7. Canny, J. & Reif, J. New lower bound techniques for robot motion planning problems. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, 49–60, (IEEE, 1987).
8. Almaghout, K., Cherubini, A. & Klimchik, A. Robotic co-manipulation of deformable linear objects for large deformation tasks. *Robotics and Autonomous Systems*, 104652 (Elsevier, 2024).
9. Brooks, M. Quantum computers: what are they good for?. *Nature Spotlight*, (2023). <https://www.nature.com/articles/d41586-023-01692-9>
10. Montanaro, A. Quantum algorithms: an overview. *npj Quantum Information*, **2**(1), 1–8 (Nature Publishing Group, 2016).
11. Möller, M. & Vuik, C. On the impact of quantum computing technology on future developments in high-performance scientific computing. *Ethics and information technology*, **19**, 253–269 (Springer, 2017).
12. Votroubek, T. & Kroupa, T. Globally Optimal Inverse Kinematics as a Non-Convex Quadratically Constrained Quadratic Program. *IEEE Robotics and Automation Letters*, (IEEE, 2024).
13. Kadouaki, T. & Nishimori, H. Quantum annealing in the transverse Ising model. *Physical Review E*, **58**(5), 5355 (APS, 1998).
14. Wang, C., Chen, H. & Jonckheere, E. Quantum versus simulated annealing in wireless interference network optimization. *Scientific reports*, **6**(1), 25797 (Nature Publishing Group UK London, 2016).
15. Goodrich, T. D., Sullivan, B. D. & Humble, T. S. Optimizing adiabatic quantum program compilation using a graph-theoretic framework. *Quantum Information Processing*, **17**, 1–26 (2018).
16. Ayanzadeh, R., Haleem, M. & Finin, T. Reinforcement quantum annealing: A hybrid quantum learning automata. *Scientific Reports*, **10**(1), 7952 (2020).
17. King, A. D. et al. Quantum critical dynamics in a 5,000-qubit programmable spin glass. *Nature*, **617**(7959), 61–66 (2023).
18. Carugno, C., Ferrari Dacrema, M. & Cremonesi, P. Evaluating the job shop scheduling problem on a D-Wave quantum annealer. *Scientific Reports*, **12**(1), 6539 (2022).
19. Salloum, H., Zhanalin, S., Badr, A. A. & Kholodov, Y. Mini-scale traffic flow optimization: an iterative QUBOs approach converting from hybrid solver to pure quantum processing unit. *Scientific Reports*, **15**(1), 22904 (2025).
20. King, A. D., et al. Computational supremacy in quantum simulation. arXiv preprint arxiv.org/abs/2403.00910 (2024).
21. Harris, R., et al. Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor. *Physical Review B—Condensed Matter and Materials Physics*, **82**(2), 024511 (APS, 2010).
22. Johnson, M. W., et al. A scalable control system for a superconducting adiabatic quantum optimization processor. *Superconductor Science and Technology*, **23**(6), 065004 (IOP Publishing, 2010).
23. Bunyk, P. I., et al. Architectural considerations in the design of a superconducting quantum annealing processor. *IEEE Transactions on Applied Superconductivity*, **24**(4), 1–10 (IEEE, 2014).
24. D-Wave Systems. *D-Wave Quantum Processing Unit Documentation*. (2025). https://docs.dwavesys.com/docs/latest/doc_qpu.html (Accessed: 2025-01-08)
25. Jun, K. & Lee, H. HUBO and QUBO models for prime factorization. *Scientific Reports*, **13**(1), 10080 (Nature Publishing Group UK London, 2023).
26. Djidjev, H. N. Quantum annealing with inequality constraints: the set cover problem. *Advanced Quantum Technologies*, **6**(11), 2300104 (Wiley Online Library, 2023).
27. Salloum, H., Lukin, R. & Mazzara, M. Quantum computing in drug discovery: A review of quantum annealing and gate-based approaches. In *International Conference on Computational Optimization* (2024).
28. Bishwas, A. K., Pitchai, A. & Som, A. Molecular unfolding formulation with enhanced quantum annealing approach. arXiv preprint [arXiv:2403.00507](https://arxiv.org/abs/2403.00507), (2024).
29. Dattani, N., Szalay, S., & Chancellor, N. Pegasus: The second connectivity graph for large-scale quantum annealing hardware. arXiv preprint [arXiv:1901.07636](https://arxiv.org/abs/1901.07636), (2019).
30. D-Wave Systems, Inc. Zephyr Topology of D-Wave Quantum Processors. https://www.dwavesys.com/media/2uznec4s/14-1056a-a_zephyr_topology_of_d-wave_quantum_processors.pdf (Accessed: Jan. 9, 2025)

31. D-Wave Systems. *Minor-Embedding*. (2025). <https://docs.ocean.dwavesys.com/en/stable/concepts/embedding.html> (Accessed: 2025-01-09)
32. D-Wave Systems. *Embedding*. (2025). https://docs.dwavesys.com/docs/latest/handbook_embedding.html (Accessed: 2025-01-09)
33. D-Wave Systems Inc. *Quantum Processing Unit: Introduction to the QPU*. (2025). https://docs.dwavesys.com/docs/latest/c_qpu_ic_e.html#id1
34. Jünger, M., et al. Performance of a quantum annealer for ising ground state computations on chimera graphs. arXiv preprint [arXiv:1904.11965](https://arxiv.org/abs/1904.11965), (2019).
35. McGeoch, C. & Farré, P. Advantage processor overview. *D-Wave systems*, 14–105 (2022).
36. Boothby, K., Bunyk, P., Raymond, J. & Roy, A. Next-generation topology of d-wave quantum processors. arXiv preprint [arXiv:2003.00133](https://arxiv.org/abs/2003.00133), (2020).
37. Booth, M., Reinhardt, S. P. & Roy, A. Partitioning optimization problems for hybrid classical. *Technical Report*, 01–09 (2017).
38. D-Wave Systems. Hybrid Solvers for Quadratic Optimization. *Technical report*, (D-Wave Systems Inc., 2020). <https://www.dwavesys.com/media/soxph512/hybrid-solvers-for-quadratic-optimization.pdf>
39. D-Wave Systems. dwave-neal. <https://github.com/dwavesystems/dwave-neal> (Accessed: March 1, 2025)
40. Kirkpatrick, S., Gelatt Jr, C. D., Vecchi, M. P. Optimization by simulated annealing. *Science*, **220**(4598), 671–680 (American association for the advancement of science, 1983).
41. Batsheva, Anastasiia, Chertkov, Andrei, Ryzhakov, Gleb & Oseledets, Ivan. PROTES: probabilistic optimization with tensor sampling. *Advances in Neural Information Processing Systems***36**, 808–823 (2023).
42. Salloum, H., et al. Quantum Annealing for the Mechanics of Robot Inspectors on Electrical Transmission Lines' Conductors. In *2024 International Conference on Control, Robotics and Informatics (ICCR)*, 49–52 (2024).
43. Salloum, H., et al. Performance of Quantum Annealing Machine Learning Classification Models on ADMET Datasets. *IEEE Access* (IEEE, 2025).
44. Huang, T. et al. Benchmarking quantum (-inspired) annealing hardware on practical use cases. *IEEE Transactions on Computers***72**(6), 1692–1705 (2022).
45. Oshiyama, H. & Ohzeki, M. Benchmark of quantum-inspired heuristic solvers for quadratic unconstrained binary optimization. *Scientific Reports***12**(1), 2146 (2022).
46. Zeng, Q.-G. et al. Performance of quantum annealing inspired algorithms for combinatorial optimization problems. *Communications Physics***7**(1), 249 (2024).

Author contributions

H.S., S.S. wrote the original manuscript, while H.S., S.S., G.R. and M.F. were responsible for editing. The methodology was developed by S.S and H.S., G.R., and I.O., with S.S. contributing more to the robotic side and H.S. focusing on the Quantum aspects. Project supervision was carried out by H.S. and S.S. All authors reviewed the manuscript.

Funding

The research has been supported by the Ministry of Economic Development of the Russian Federation (agreement No. 139-10-2025-034 dd. 19.06.2025, IGK 000000C313925P4D0002).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to H.S. or M.F.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025