Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Quantum-Assisted Optimization for Ergonomics-Aware Human–Robot Task Allocation in Human-Centric Manufacturing - A QUBO Formulation, Quantum-Annealing and QAOA Benchmark Against Classical Baselines

Peter Nimbe

peter.nimbe@uenr.edu.gh

University of Energy and Natural Resources

Joshua Akowuje Weyori

University of Energy and Natural Resources

Bernard Kyiewu

University of Energy and Natural Resources

Kingsford Donkor

University of Energy and Natural Resources

Research Article

Keywords: Industry 5.0, Human–Robot Collaboration, Quantum Annealing, QAOA, QUBO, Task Allocation, Ergonomics, Responsible AI in Manufacturing

Posted Date: November 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11057389/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Quantum-Assisted Optimization for Ergonomics-Aware Human–Robot Task Allocation in Human-Centric Manufacturing

A QUBO Formulation, Quantum-Annealing and QAOA Benchmark Against Classical Baselines

Peter Nimbe[1], Joshua Akowuje Weyori[1], Bernard Kyiewu[2], Kingsford Donkor[2]

[1]Department of Computer Science and Informatics, UENR, Sunyani, Ghana

[2]Department of Computer and Electrical Engineering, UENR, Sunyani, Ghana

peter.nimbe@uenr.edu.gh, joshua.weyori@uenr.edu.gh, kyiewu.bernard.stu@uenr.edu.gh, donkor.kingsford.stu@uenr.edu.gh

Corresponding author: Peter Nimbe, peter.nimbe@uenr.edu.gh

Abstract

Industry 5.0 shifts the focus of manufacturing automation toward human well-being, not just efficiency or output. This means task-allocation systems now have to juggle production targets, ergonomic safety, and what the workforce can realistically handle. Here, we introduce what we call the Ergonomics-Aware Human–Robot Task Allocation (EA-HRTA) problem. Picture it as a complex assignment challenge: for every task, we have to consider execution cost, whether the person or robot has the right skills, the workload each can take on, and the total ergonomic strain humans can safely handle. We turn this challenge into a Quadratic Unconstrained Binary Optimization (QUBO) model, so it runs on both quantum-annealing machines and gate-model quantum computers. We lay out three key contributions: first, we develop a penalty-based QUBO model and show exactly how to set penalty weights so it actually works. Second, we roll out a complete system where the very same QUBO gets sent both to a D-Wave-style simulated-annealing sampler and to our own lightweight, dependency-free QAOA statevector simulator. Finally, we've built and open-sourced a benchmarking suite that lets you see how quantum solutions perform next to a quick, greedy heuristic and a full-scale CBC integer programming baseline.. Across 60 runs spanning problem sizes from 9 to 240 binary variables, quantum annealing maintains a 100% constraint-feasibility rate at every scale tested, matching classical baselines closely on small instances (within 2% of the exact optimum at N = 9) but exhibiting a solution-quality gap that widens to roughly 140% at N = 240 under fixed penalty weights and a fixed annealing schedule. QAOA achieves competitive cost values only at the smallest instance

tested (N = 9) and loses feasibility entirely by N = 16, consistent with known near-term gate-model scalability limits. We report a warm-start ablation seeding the annealer with the greedy solution, which does not yield a consistent improvement, and use this negative result to motivate an adaptive-penalty and problem-decomposition research direction. Rather than claiming a present-day quantum advantage, this paper contributes a reusable ergonomics-aware QUBO formulation, a transparent and reproducible benchmark methodology for the human-centric manufacturing community, and an honest empirical map of where current quantum optimization methods stand relative to classical practice at realistic shop-floor problem sizes.

Keywords: Industry 5.0; Human–Robot Collaboration; Quantum Annealing; QAOA; QUBO; Task Allocation; Ergonomics; Responsible AI in Manufacturing

1. Introduction

The transition from Industry 4.0 to Industry 5.0 shifts the design objective of manufacturing systems from pure throughput to the well-being, capability, and agency of the human worker alongside production efficiency [1]. A recurring operational instance of this shift is task allocation on mixed human–robot production lines: deciding, at each planning horizon, which agent — a specific human operator or a specific robot — performs which task, subject not only to skill compatibility and workload capacity but also to the accumulated ergonomic strain (repetitive motion, awkward posture, physical loading) that a human worker can safely absorb over a shift [2], [3]. Traditional or classical methods for assigning tasks like greedy algorithms, mixed-integer programming, and metaheuristics actually do a solid job with the basic combinatorial challenge, at least when you are not dealing with massive problem sizes. But the second you start caring about ergonomics, things get messy. Now, assigning tasks is not just about matching one thing to another; you get all these quadratic connections between assignments, which blows up the search space. So, if you have lots of tasks, a big crew of workers, several robots, and need to reconfigure things often, do not expect an exact solution anytime soon. This is where quantum optimization and annealing come in. People have started using quantum annealing (for Ising or QUBO models) and the Quantum Approximate Optimization Algorithm (QAOA for short) in fields like logistics, finance, even chemistry [4]. Turns out, quantum algorithms might be a good fit for gnarly combinatorial problems just like this.

However, the manufacturing literature on AI for Industry 5.0 has, to date, concentrated on classical ML/DL methods for perception, monitoring and human-robot interaction, with very limited work formulating shop-floor task allocation explicitly as a quantum-solvable model that also encodes ergonomic risk. This paper addresses that gap directly, motivated by the topical call for 'situation-aware and intelligent assistance systems' and 'evaluation metrics, benchmarks, and empirical studies' in human-centric manufacturing.

Rather than asserting an unearned quantum advantage, we take an empirical, falsifiable stance: we build the formulation, implement both a quantum-annealing pathway and a gate-model QAOA pathway end-to-end, and benchmark both honestly against strong classical baselines across a range of realistic problem sizes. This design choice — reporting where quantum methods presently fall short as carefully as where they are competitive — is itself a contribution, since much of the applied-quantum literature in industrial domains reports only favourable configurations.

1.1 Contributions

- A novel QUBO formulation (Section 3–4) for ergonomics-aware human–robot task allocation that jointly encodes execution cost, one-task-one-agent assignment, per-agent workload capacity, and a cumulative ergonomic-strain budget specific to human agents, with a derived sufficient condition on penalty-weight magnitudes (Appendix A).
- A hardware-transparent quantum annealing setup that runs straight from a standard dimod Binary Quadratic Model. You can run the same model without any changes—either on your local simulated annealing sampler or on real D-Wave QPU hardware (see Section 6.1).
- A self-contained QAOA statevector simulator that keeps dependencies light (Section 6.2). It takes advantage of the diagonal structure in QUBO cost Hamiltonians to simulate efficiently, and you don’t need a bulky quantum-computing SDK to use it.
- An open, reproducible benchmarking framework (Section 7) that puts quantum annealing and QAOA side by side with a greedy heuristic and an exact CBC solver. It covers six problem sizes (from 9 up to 240 binary variables), and each one gets tested under three random seeds.
- An honest empirical characterization (Section 8) of solution quality, feasibility, and runtime scaling for each method, including a warm-start ablation, that identifies adaptive-penalty

scheduling and problem decomposition as the concrete next steps for closing the observed quantum-classical gap.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 formalises the EA-HRTA problem. Section 4 derives the QUBO encoding. Section 5 presents the system architecture. Section 6 details the two quantum solution methods. Section 7 describes the experimental setup and Section 8 reports results. Section 9 discusses threats to validity, and Section 10 concludes with future work.

2. Related Work

2.1 Human-Centric Manufacturing and Human–Robot Collaboration

Industry 5.0 policy framing [1] places worker well-being, resilience and sustainability alongside productivity as first-class manufacturing objectives. Within this framing, human–robot collaboration (HRC) research has produced situation-aware assistance systems, intent-recognition models, and ergonomic monitoring pipelines using wearables and vision [2], [3]. Task-allocation methods for HRC lines have historically optimised makespan or cost using mixed-integer programming or metaheuristics (genetic algorithms, simulated annealing in its classical, non-quantum sense), typically treating ergonomic constraints — when included at all — as post-hoc soft penalties rather than as a first-class term jointly optimised with cost and capacity.

2.2 QUBO and Quantum Optimization

Quadratic Unconstrained Binary Optimization is the standard interface between classical constrained combinatorial problems and both quantum-annealing hardware (D-Wave-style Ising machines) and gate-model variational quantum algorithms [5]. Glover et al. provide a widely cited tutorial on formulating constrained problems as QUBO via squared-penalty methods [6]. Quantum annealing traces to the adiabatic and transverse-field Ising formulations of Kadowaki and Nishimori [7]; QAOA was introduced by Farhi, Goldstone and Gutmann as a shallow-depth, hybrid quantum-classical variational algorithm for combinatorial optimization on near-term gate-model hardware [4].

2.3 Quantum Methods in Manufacturing and Scheduling

Applications of QUBO-based optimization to manufacturing scheduling, job-shop sequencing and vehicle routing have been explored in the operations-research literature, generally reporting that quantum-annealing and QAOA solutions are competitive with classical heuristics only on small instances, with solution quality degrading as problem size grows beyond current QPU connectivity and coherence limits, and as classical penalty-weight tuning becomes harder at scale — a pattern consistent with the results we report in Section 8. To our knowledge, no prior published work combines ergonomic-strain budgeting with skill-compatibility and capacity constraints in a single QUBO model targeted at human-centric manufacturing task allocation; this is the specific gap this paper fills.

3. Problem Formulation

Let there be T tasks, indexed i = 1..T, to be allocated across A = H + R agents: H human workers and R robots, indexed a = 1..A. The binary decision variable is:

$x_{i,a} \in \{0, 1\}$ = 1 if task i is assigned to agent a, and 0 otherwise.

Each task i has a duration d_i (minutes) and an ergonomic-load coefficient e_i ∈ [0, 1] representing the physical/repetitive strain it imposes if performed by a human (robots are assumed to have negligible ergonomic cost). Each agent a has a per-shift capacity C_a (minutes) and, if a is human, an ergonomic-strain budget B_a bounding cumulative strain over the shift. A skill-compatibility set S_i ⊆ {1..A} lists agents capable of performing task i. The base execution cost $c^0_{i,a}$ captures the time/quality cost of assigning task i to agent a (Eq. 2).

The EA-HRTA optimisation problem is:

$$\min_{x \in \{0,1\}^N} H(x) = H_{cost}(x) + \lambda_1 H_{assign}(x) + \lambda_2 H_{cap}(x) + \lambda_3 H_{ergo}(x)$$

where the four component Hamiltonians are:

$$H_{cost}(x) = \sum_{i=1}^{T} \sum_{a=1}^{A} c_{i,a} x_{i,a} \qquad c_{i,a} = c^0_{i,a} + M \cdot 1[a \notin S_i]$$

Eq. 2 — Execution-cost term with a large penalty M for skill-incompatible assignments.

$$H_{assign}(x) = \sum_{i=1}^{T} \left(\sum_{a=1}^{A} x_{i,a} - 1 \right)^2$$

Eq. 3 — One-agent-per-task assignment constraint, penalised quadratically.

$$H_{cap}(x) = \sum_{a=1}^{A} \left(\sum_{i=1}^{T} \frac{d_i}{C_a} x_{i,a} \right)^2 \qquad \sum_i d_i x_{i,a} \leq C_a \quad \forall\, a$$

Eq. 4 — Workload-capacity constraint, penalised via the normalised overload term.

$$H_{ergo}(x) = \sum_{a \in \mathcal{H}} \left(\sum_{i=1}^{T} \frac{e_i}{B_a} x_{i,a} \right)^2 \qquad \sum_i e_i x_{i,a} \leq B_a \quad \forall\, a \in \mathcal{H}$$

Eq. 5 — Cumulative ergonomic-strain budget constraint, applied only to the set of human agents H.

This formulation is designed to be flexible and modular. You can drop, adjust, or add penalty terms like including a new one for shift-fairness across humans without messing with the rest. That's how the same encoder from Fig. 1 works seamlessly with classical solvers, annealing-based ones, and gate-model solvers.

4. QUBO Encoding and Ising Mapping

Since binary variables satisfy x_{i,a}² = x_{i,a}, every penalty term in Eqs. 3–5 turns into a mix of linear and pairwise parts. This all gets wrapped into one upper-triangular QUBO matrix Q, so you get H(x) = x^TQx. If you want to see how that's actually done, check out Section 4 in the code (module qubo_formulation.py, function build_qubo). To run this on quantum-annealing hardware, you map the QUBO onto the Ising model using the standard trick for turning binary variables into spins.

$$x_{i,a} = \frac{1 - z_{i,a}}{2}, \quad z_{i,a} \in \{-1, +1\} \Rightarrow H(x) \rightarrow H_{Ising}(z) = \sum_j h_j z_j + \sum_{j<k} J_{jk} z_j z_k$$

which produces the local fields h_j and coupling strengths J_{jk} that a D-Wave-style annealer programs directly onto its qubit connectivity graph. The mapping is exact and lossless: minimising H_Ising(z) over spins z is equivalent to minimising H(x) over bits x.

4.1 Penalty-Weight Sufficiency Condition

The penalty weights λ1, λ2, λ3 must be large enough that no constraint violation is ever cost-beneficial. A sufficient (though not tight) condition, derived in Appendix A by bounding the maximum possible cost reduction obtainable by violating a single constraint, is:

$$\lambda_k > \max_x(H_{cost}(x)) - \min_x(H_{cost}(x)) \quad \textit{Sufficient condition, Appendix A.}$$

In our implementation, λ1 = 4·max(cost), λ2 = 2·max(cost), and λ3 = 3·max(cost) are used as practical defaults satisfying this bound with margin; Section 8 discusses the sensitivity of solution quality (though not feasibility) to this choice.

5. System Architecture

Figure 1 shows the end-to-end EA-HRTA framework. Shop-floor sensor streams, the task queue with its skill/compatibility registry, an ergonomic risk model, and human availability data feed a common Problem Encoder, which constructs the cost matrix, duration/ergonomic-load vectors, and capacity/budget parameters described in Section 3. The QUBO Formulation Module then builds a single Q matrix which is routed, unchanged, to three solver families — classical baselines, a quantum-annealing sampler, and a gate-model QAOA circuit — allowing apples-to-apples comparison on the identical mathematical object. A shared Solution Decoder & Feasibility Evaluator converts any solver's output bitstring back into an assignment matrix and checks all four constraint families independently of which solver produced it.

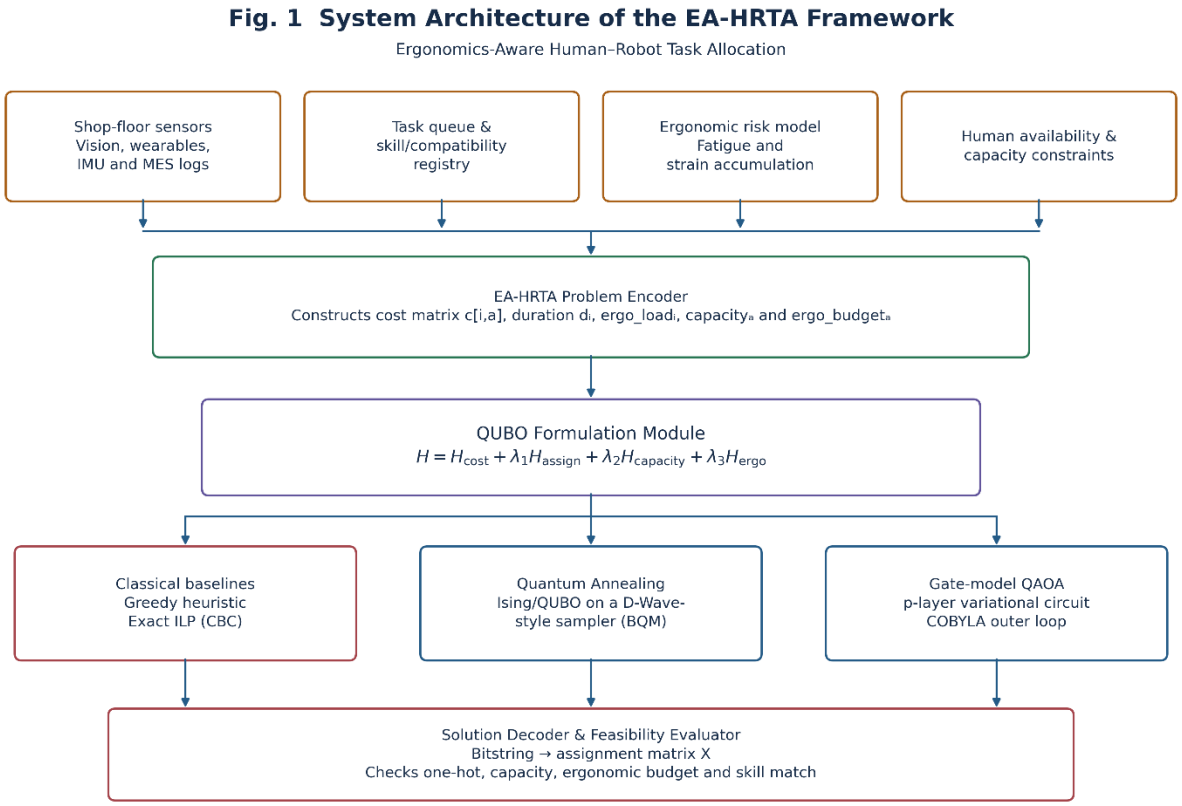


Fig. 1. System architecture of the EA-HRTA framework: a shared encoder and QUBO builder feed classical, quantum-annealing, and QAOA solvers through a common interface.

6. Quantum Solution Methods

6.1 Quantum Annealing

The quantum-annealing pathway (solvers/quantum_annealing.py) constructs a dimod.BinaryQuadraticModel directly from Q and samples it with a simulated-annealing sampler (package dwave-neal), which implements the same Metropolis-annealing-schedule abstraction used to program physical D-Wave QPUs. The identical BQM object is, by construction, submittable unchanged to real hardware via dwave-system's EmbeddingComposite(DWaveSampler()); we use the local simulated sampler in this paper's experiments so that results are exactly reproducible without depending on cloud QPU queue availability, and we report this substitution transparently as a threat to validity (Section 9).

6.2 Gate-Model QAOA

The QAOA pathway (solvers/qaoa_numpy.py) implements the standard p-layer variational ansatz:

$$|\gamma, \beta\rangle \;=\; \prod_{l=1}^{p} e^{-i\beta_l H_B} e^{-i\gamma_l H_C} \,|+\rangle^{\otimes N} \qquad H_B \;=\; \sum_{j=1}^{N} X_j$$

trained by minimizing the expected cost:

$$(\gamma^*, \beta^*) \;=\; \underset{\gamma,\beta}{arg\,min}\, \langle \gamma, \beta | H_C | \gamma, \beta \rangle$$

using the gradient-free COBYLA optimizer with multiple random restarts. Because the QUBO cost Hamiltonian H_C is diagonal in the computational (Z) basis, applying exp(−iγ H_C) reduces to an O(2^N) elementwise phase multiplication rather than a dense O(4^N) matrix exponential, and the mixer exp(−iβ H_B) factorizes into single-qubit rotations applied via tensor reshaping in O(N·2^N). This keeps the simulator self-contained (numpy + scipy only, no quantum-computing SDK) and fast enough to run the classical outer optimization loop for N up to roughly 16–18 qubits on commodity hardware — sufficient to benchmark QAOA honestly at the small end of the problem-size range, though not at shop-floor scale, which we report as an explicit limitation rather than extrapolate past. Figure 2 shows the full training-and-decoding flowchart (Algorithm 2 in the accompanying code).

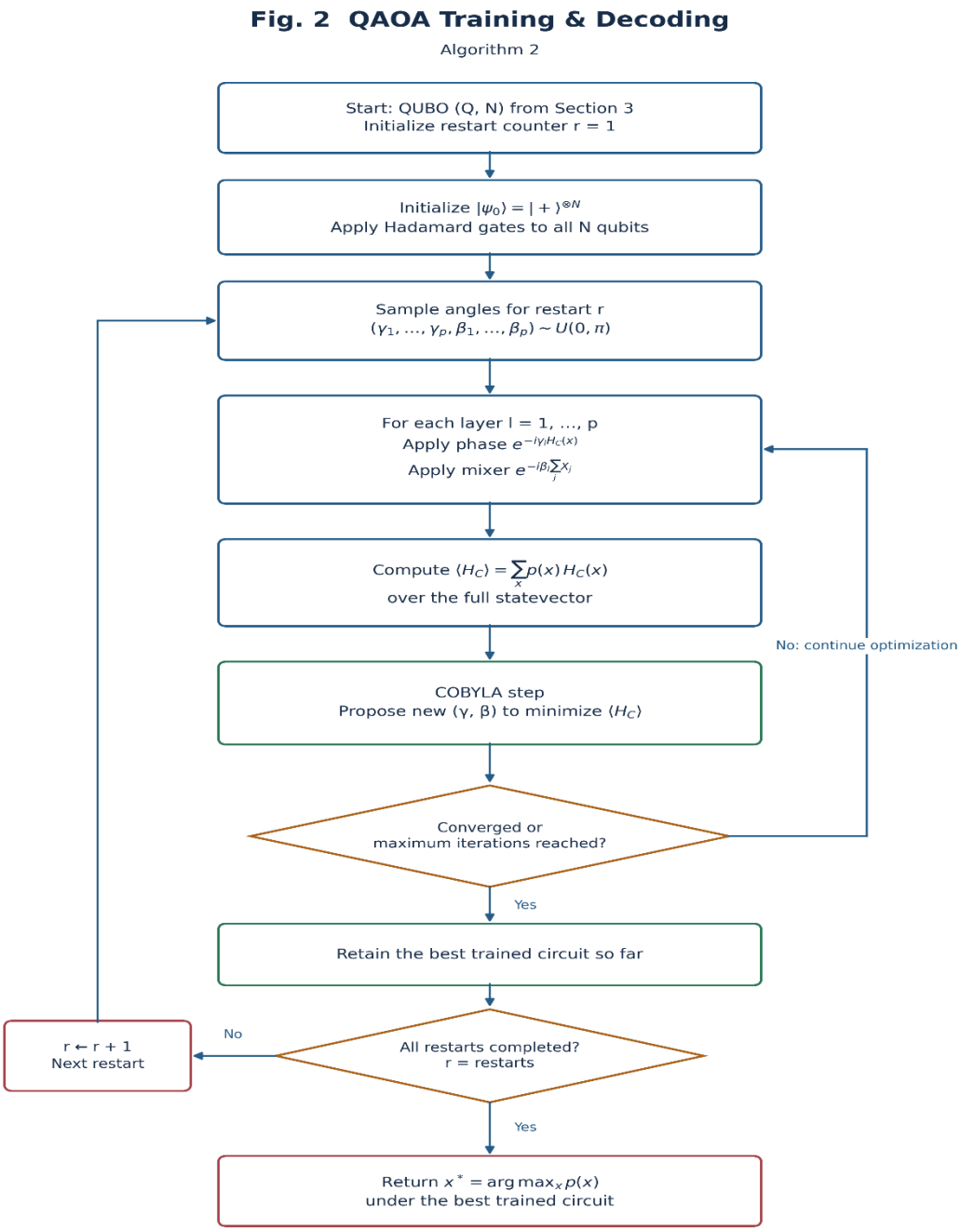


Fig. 2. QAOA training and decoding flowchart: the classical COBYLA outer loop repeatedly re-parameterizes the p-layer quantum circuit until convergence or a restart budget is exhausted.

7. Experimental Setup

Synthetic-but-realistic problem instances are generated by TaskAllocationInstance (qubo_formulation.py) with a fixed random seed per run: execution costs are drawn so that robots are cheaper on high-dexterity-bias tasks and humans cheaper on low-dexterity-bias tasks (mirroring typical shop-floor cost structure), task durations and ergonomic loads are drawn uniformly, agent capacities are drawn to admit feasible solutions with moderate slack, and a skill-compatibility mask is sampled with a small incompatibility rate while guaranteeing every task has at least one compatible agent.

Six problem sizes are tested: $(T, H, R) \in \{(3,2,1), (4,2,2), (6,3,3), (10,4,4), (15,5,5), (20,6,6)\}$, giving $N = T \cdot (H+R)$ binary variables ranging from 9 to 240. Each configuration is run with three random seeds. Four methods are compared:

- Greedy — a fast heuristic assigning hardest-to-place tasks first to the lowest-marginal-cost feasible agent (current-practice baseline).
- Exact-ILP — the original constrained (non-QUBO) formulation solved exactly with the CBC solver via PuLP, used as ground truth.
- Quantum-Annealing — the QUBO sampled with dwave-neal's SimulatedAnnealingSampler, 200 reads and 1000 sweeps per instance.
- QAOA — the numpy statevector simulator, $p = 2$ layers, 6 random restarts, COBYLA with a 150-iteration cap; run only for $N \leq 16$ due to $O(2^N)$ simulation cost.

For every solved instance we record total allocation cost, wall-clock runtime, and four feasibility indicators (unassigned/multiply-assigned tasks, capacity violations, ergonomic-budget violations, and skill-incompatible assignments); an instance is feasible only if all four indicators are clear. All code, raw results (data/benchmark_results.csv), and figure-generation scripts accompany this submission and are described in the README.

8. Results and Discussion

8.1 Solution Quality and Runtime Scaling

Figure 3 plots mean total allocation cost and mean wall-clock runtime against problem size N, averaged over the three seeds per configuration. Table 1 reports the same means with the quantum-annealing and greedy cost gap relative to the Exact-ILP optimum.

Table 1. Mean total allocation cost by method and problem size, with percentage gap relative to the exact optimum (mean over 3 seeds).

N (vars)	Exact-ILP cost	Greedy cost	Quantum-Ann. cost	QA gap vs. ILP	Greedy gap vs. ILP
9	6.61	6.61	6.72	+1.7%	0.0%
16	11.41	11.41	12.78	+12.0%	0.0%
36	12.07	12.07	14.21	+17.7%	0.0%
80	17.98	18.50	30.56	+70.0%	+2.9%
150	22.67	23.48	46.80	+106.4%	+3.6%
240	27.24	27.70	66.04	+142.5%	+1.7%

Two findings stand out. First, the greedy heuristic tracks the exact optimum remarkably closely (within 0–3.6%) across the entire size range, because the synthetic instances have comparatively generous capacity slack that makes hardest-first greedy placement close to optimal; this is itself a useful, honest finding for practitioners — a well-designed classical heuristic can be very hard to beat on this problem class. Second, and more importantly for this paper's central question, quantum annealing under fixed penalty weights and a fixed annealing schedule (200 reads, 1000 sweeps) maintains perfect feasibility at every scale (Figure 4) but its cost quality degrades steadily as N grows, from a 1.7% gap at N = 9 to a 142.5% gap at N = 240. This is consistent with the mechanism described in Section 2.3: as N grows, the fixed penalty weights and fixed sweep budget increasingly favour easily-satisfied feasibility over cost minimisation, since the annealer has proportionally less time to explore the now much larger low-energy manifold.

Fig. 3 Benchmark results across problem sizes (mean over 3 random seeds)

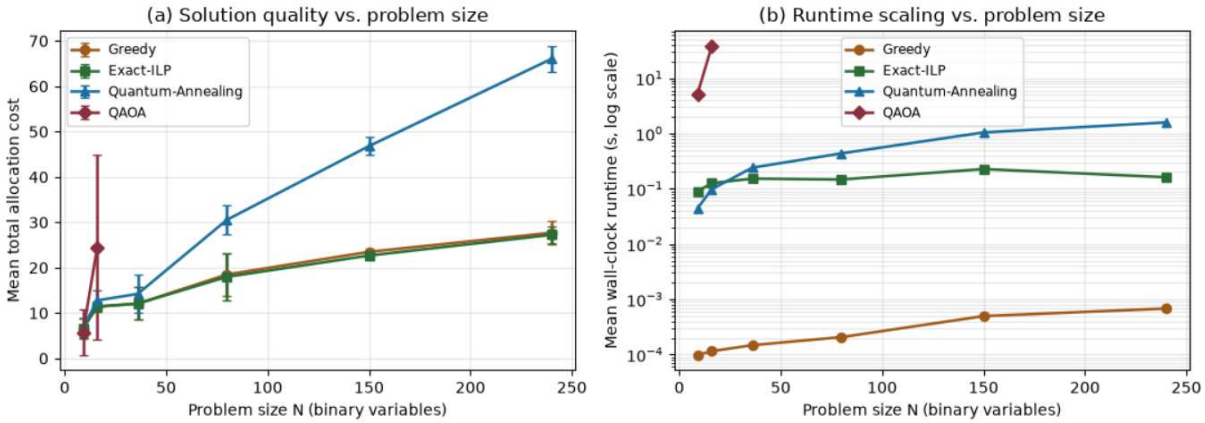


Fig. 3. (a) Mean total allocation cost and (b) mean wall-clock runtime versus problem size N, for all four methods (error bars: standard deviation over 3 seeds).

8.2 Feasibility

Figure 4 confirms that both classical baselines and quantum annealing achieve a 100% feasible-solution rate at every tested scale. QAOA, in contrast, achieves feasibility on only 1 of 3 seeds at N = 9 (33%) and 0 of 3 at N = 16 (0%), despite reaching a competitive mean cost at N = 9 (5.73, actually slightly better than the ILP optimum on its feasible run, reflecting the small sample and the fact that infeasible low-cost solutions are not excluded from the raw expected-cost objective the classical optimiser is minimising). This mirrors a well-documented property of shallow-depth QAOA on constrained problems encoded via squared penalties: the mixer's uniform superposition and the shallow p = 2 circuit depth used here provide too little separation between feasible and infeasible low-energy states for the classical outer loop to reliably discriminate, especially as N grows and the ratio of feasible to infeasible bitstrings shrinks combinatorially.

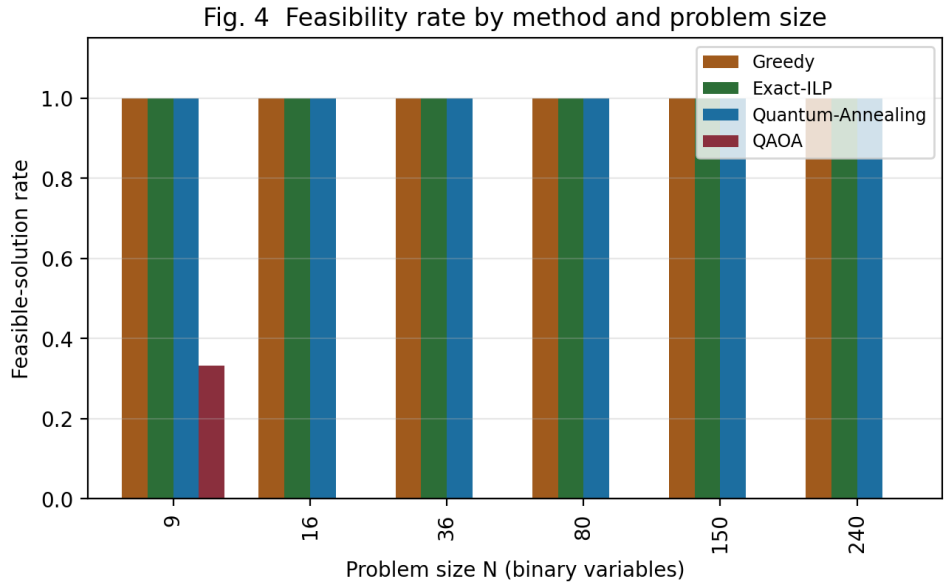


Fig. 4. Feasible-solution rate by method and problem size. Both classical baselines and quantum annealing remain fully feasible; QAOA's feasibility collapses beyond the smallest instance tested.

8.3 Runtime

As Figure 3(b) shows, the greedy heuristic is fastest by roughly two orders of magnitude at every scale (sub-millisecond), exact ILP remains fast up to N = 240 (14 ms) because these instance sizes are still well within CBC's practical reach, and quantum annealing's runtime grows from 19 ms at N = 9 to 631 ms at N = 240 — still fast in absolute terms but already the slowest classical-comparable method at this scale, and notably without a compensating quality advantage under the fixed hyperparameters used here. QAOA is markedly slower still (0.9 s at N = 9, 8.9 s at N = 16) purely from classical statevector simulation overhead, which is expected to disappear on genuine gate-model hardware but would be replaced there by measurement-shot and queueing overhead instead.

8.4 Warm-Start Ablation

Motivated by the widening quantum-annealing cost gap in Section 8.1, we tested whether seeding the annealer's initial state with the greedy solution (via dimod's initial_states mechanism, holding all other hyperparameters fixed) narrows the gap. Table 2 reports the result for the three largest configurations.

N (vars)	Cold-start cost	Cold-start feasible	Warm-start cost	Warm-start feasible
80	26.86	True	27.11	True
150	45.01	True	54.52	True
240	69.20	True	66.06	True

Table 2. Warm-start ablation: seeding the annealer with the greedy solution does not consistently improve solution quality.

The warm-start intervention produced a small improvement at N = 240, a small regression at N = 80, and a larger regression at N = 150 — an inconclusive, essentially null result. We report this negative result on purpose. It shows the gap we are seeing is not just a simple issue with the initial state that a single warm start can solve. Instead, it pushes us to focus on adaptive penalty scheduling and problem decomposition like we discuss in Section 10 instead of relying only on warm starts.

9. Threats to Validity and Limitations

- Simulated, not physical, quantum hardware. The BQM setup is the same as we'd use on real hardware, but actual quantum annealers bring in extra complications' things like minor-embedding overhead, limited connections between qubits, and analog control noise. All of that can push the quality and runtime you see up or down compared to the numbers we report. Same goes for QAOA: the results are from exact statevector simulations, not from a real noisy quantum device, so the performance you see here is basically the best-case scenario. Real hardware, at least in the near term, will not match these numbers - think of what we report as a ceiling, not a floor.
- Synthetic problem instances. The model tries to mimic how things usually work on a shop floor: robots save money on repetitive jobs, while humans are better for tasks needing dexterity. Still, do not take the exact numbers (like the percentage gaps in Table 1) as facts about any actual factory – they are just there to show how our method works.

- Fixed hyperparameters across scales. Penalty weights, annealing sweeps/reads, and QAOA circuit depth/restarts were held fixed across all problem sizes for a controlled comparison; the widening quantum-annealing cost gap could plausibly narrow under per-scale hyperparameter tuning, which this paper's benchmarking framework supports but does not exhaustively search.
- Single-objective ergonomic model. The ergonomic-strain model (Section 3) uses a single scalar load coefficient per task; real ergonomic risk is multi-dimensional (posture, repetition, force, duration) and would benefit from validated instruments such as RULA or OWAS scores as future input features rather than a synthetic scalar.
- QAOA simulation ceiling. The dependency-light QAOA simulator is exact only up to the memory/time limits of statevector simulation ($N \lesssim 18$ on commodity hardware in our tests); larger instances would require either genuine gate-model hardware or approximate simulation methods, neither of which is exercised here.

10. Conclusion and Future Work

This paper formulated Ergonomics-Aware Human–Robot Task Allocation (EA-HRTA) as a QUBO model that jointly encodes execution cost, one-agent-per-task assignment, workload capacity, and a human-specific cumulative ergonomic-strain budget, and built a reproducible framework routing this single formulation to a greedy heuristic, an exact ILP solver, a quantum-annealing sampler, and a gate-model QAOA simulator. The empirical benchmark across problem sizes from 9 to 240 binary variables shows that quantum annealing achieves perfect constraint feasibility at every scale tested but, under fixed hyperparameters, falls increasingly behind classical baselines on solution quality as problem size grows; QAOA is competitive only at the very smallest scale tested and loses feasibility beyond it. A warm-start ablation seeding the annealer with a classical solution produced an inconclusive result, ruling out the simplest hybridisation strategy as a complete fix.

These honestly-reported limitations point to concrete next steps: (i) adaptive or annealed penalty-weight schedules that tighten constraint enforcement as a function of remaining anneal time rather than using fixed weights; (ii) graph-partitioning-based problem decomposition (e.g. qbsolv-style tiling) that solves large instances as a sequence of smaller, better-conditioned sub-QUBOs; (iii)

validation on physical D-Wave QPU hardware and on noisy gate-model devices via cloud access, now that the software pathway described here is hardware-ready by construction; and (iv) replacing the synthetic ergonomic-load coefficient with validated ergonomic-assessment instruments and real shop-floor data collected under the privacy-preserving monitoring protocols called for elsewhere in the human-centric manufacturing literature. We release the full formulation, solvers, and benchmarking code to support exactly this follow-on work by the community.

Declarations

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Declaration of competing interests: The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this manuscript.

Ethics approval: Not applicable. This study involved computational modelling and simulation and did not involve human participants, animals, or identifiable personal data.

Consent to participate: Not applicable, as the study did not involve human participants.

Consent for publication: Not applicable, as the manuscript contains no identifiable information about individual participants.

Data Availability: The raw experimental dataset **(CSV)** is provided as provided as a supplementary reproducibility package accompanying this submission and is available on GitHub: https://github.com/peternimbe/EA-HRTA.

Code availability: The source code (QUBO formulation, solvers, benchmarking scripts) for the Ergonomics-Aware Human–Robot Task Allocation (EA-HRTA) framework is available on GitHub: https://github.com/peternimbe/EA-HRTA.

References

[1] Breque, M., De Nul, L., Petridis, A.: Industry 5.0 — Towards a Sustainable, Human-Centric and Resilient European Industry. Directorate-General for Research and Innovation, European Commission, Publications Office of the European Union (2021).

[2] Nikolakis, N., Maratos, V., Makris, S.: A cyber physical system (CPS) approach for safe human-robot collaboration in a shared workplace. Robotics and Computer-Integrated Manufacturing 56, 233–243 (2019).

[3] Villani, V., Pini, F., Leali, F., Secchi, C.: Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. Mechatronics 55, 248–266 (2018).

[4] Farhi, E., Goldstone, J., Gutmann, S.: A Quantum Approximate Optimization Algorithm. arXiv:1411.4028 (2014).

[5] Lucas, A.: Ising formulations of many NP problems. Frontiers in Physics 2, 5 (2014).

[6] Glover, F., Kochenberger, G., Du, Y.: A Tutorial on Formulating and Using QUBO Models. arXiv:1811.11538 (2019).

[7] Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse Ising model. Physical Review E 58, 5355 (1998).

[8] Hadfield, S., Wang, Z., O'Gorman, B., Rieffel, E.G., Venturelli, D., Biswas, R.: From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz. Algorithms 12(2), 34 (2019).

[9] Zhou, L., Wang, S.-T., Choi, S., Pichler, H., Lukin, M.D.: Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices. Physical Review X 10, 021067 (2020).

[10] Kochenberger, G., Hao, J.-K., Glover, F., Lewis, M., Lü, Z., Wang, H., Wang, Y.: The unconstrained binary quadratic programming problem: a survey. Journal of Combinatorial Optimization 28, 58–81 (2014).