Análisis comparativo de QAOA mediante la formulación Tensor QUDO para una variante del TSP aplicable a motores de búsqueda de viajes

Alejandro Mata Ali[1], Adriano Lusso[2], Elías F. Combarro[3], Javier Sedano[4], and Álvaro Herrero[5]

[1] Instituto Tecnológico de Castilla y León, Burgos, Spain alejandro.mata@itcl.es
[2] Universidad Nacional del Comahue, Neuquén, Argentina adriano.lusso@est.fi.uncoma.edu.ar
[3] Department of Computer Science, University of Oviedo, Gijón, Spain efernandezca@uniovi.es
[4] Instituto Tecnológico de Castilla y León, Burgos, Spain javier.sedano@itcl.es
[5] Grupo de Inteligencia Computacional Aplicada (GICAP), Departamento de Digitalización, Escuela Politécnica Superior, Universidad de Burgos, Av. Cantabria s/n., 09006 Burgos, Spain ahcosio@ubu.es

Abstract. El Problema del Viajante de Comercio (TSP por sus siglas en inglés) constituye un problema de optimización combinatoria fundamental con amplias aplicaciones en logística, diseño de transporte y planificación de la producción. Sin embargo, los mejores algoritmos clásicos exactos conocidos presentan complejidad temporal exponencial, lo que vuelve intratables en la práctica a las instancias de gran tamaño y motiva la exploración de paradigmas alternativos como la computación cuántica. En este trabajo, implementamos la formulación QUBO y su generalización, *Tensor Quadratic Unconstrained D-ary Optimisation (T-QUDO)*, que extiende la codificación en circuitos cuánticos de qubits a qudits, para una variante del TSP. Esta variante busca determinar una ruta de costo mínimo incorporando funciones de costo dependientes del tiempo y restricciones de precedencia, y permanece en gran medida inexplorada con T-QUDO a pesar de su relevancia en la planificación dinámica de viajes. Las formulaciones se codifican dentro del algoritmo QAOA y se analizan mediante simulaciones sin ruido utilizando las librerías Cirq y CUDA-Q, con el objetivo de evaluar las posibles ventajas de las representaciones basadas en qudits frente a las basadas en qubits, particularmente en términos de expresividad, eficiencia de recursos y calidad de solución.

Keywords: Problema del viajante de comercio, QAOA, QUDO, optimización

Benchmarking QAOA on a Tensor-Based QUDO Formulation of a TSP Variant for Travel Search Engines

Alejandro Mata Ali[1], Adriano Lusso[2], Elías F. Combarro[3], Javier Sedano[4], and Álvaro Herrero[5]

[1] Instituto Tecnológico de Castilla y León, Burgos, Spain alejandro.mata@itcl.es
[2] Universidad Nacional del Comahue, Neuquén, Argentina adriano.lusso@est.fi.uncoma.edu.ar
[3] Department of Computer Science, University of Oviedo, Gijón, Spain efernandezca@uniovi.es
[4] Instituto Tecnológico de Castilla y León, Burgos, Spain javier.sedano@itcl.es
[5] Grupo de Inteligencia Computacional Aplicada (GICAP), Departamento de Digitalización, Escuela Politécnica Superior, Universidad de Burgos, Av. Cantabria s/n., 09006 Burgos, Spain ahcosio@ubu.es

Abstract. The Travelling Salesman Problem (TSP) constitutes a fundamental Combinatorial Optimisation Problem (COP) with wide-ranging applications in logistics, transportation design and production scheduling. However, the best-known exact classical algorithms exhibit exponential time complexity, rendering large instances intractable in practice and motivating the exploration of alternative paradigms such as quantum computing.

In this work, we implement the Quadratic Unconstrained Binary Optimisation (QUBO) formulation and its generalisation, Tensor Quadratic Unconstrained D-ary Optimisation (T-QUDO), which extends quantum circuit encoding from qubits to qudits, for a TSP variant. This variant seeks to determine a minimum-cost route while incorporating a time-dependent cost function and precedence constraints, and remains largely unexplored with T-QUDO despite its relevance to dynamic travel planning.

The formulations are encoded within the Quantum Approximate Optimisation Algorithm (QAOA) framework and analysed under noiseless simulations with the Cirq and CUDA-Q libraries, aiming to assess the potential advantages of qudit-based representations over qubit-based ones, particularly in terms of expressivity, resource efficiency, and solution quality.

Keywords: Travelling Salesman Problem, QAOA, QUDO, Optimisation

1 Introduction

Combinatorial Optimisation Problems (COPs) involve finding optimal solutions within exponentially large discrete spaces (Ausiello et al., 1999; Du & Pardalos, 1998), with applications in logistics, transportation and job scheduling (Bhat et al., 2026; El-Sherbeny, 2010; Henderson & Nelson, 2006; Picariello et al., 2025), where small improvements can yield significant savings. Nonetheless, many of these problems are NP-hard (Ausiello et al., 1999), a property exemplified by the Travelling Salesman Problem (TSP), one of the most extensively studied problems in the field (Applegate et al., 2007; Gutin & Punnen, 2002), whose constrained variants are widely used in real-world applications and serve as standard benchmarks for optimisation methods. The computational intractability of COPs has driven interest in quantum computing as an alternative paradigm, in which gate-based variational algorithms, in particular Quantum Approximate Optimisation Algorithm (QAOA) (Farhi et al., 2014; Qian et al., 2023), have emerged as a promising approach (Picariello et al., 2025).

Accordingly, the performance of QAOA critically depends on how problems are encoded into the cost Hamiltonian. While the standard Quadratic Unconstrained Binary Optimisation (QUBO) formulation uses binary variables with penalty terms, requiring n^2 variables and a 2^{n^2} Hilbert space for problems like the TSP, with significant one-hot overhead, Quadratic Unconstrained D-ary Optimisation (QUDO) and Tensor Quadratic Unconstrained D-ary Optimisation (T-QUDO) use qudits to encode n integer variables, reducing the space to n^n and alleviating this overhead (Bhat et al., 2026; Bucher et al., 2024; Glover et al., 2022; Mata Ali, 2025; Shuvalov et al., 2025).

Despite the growing interest in quantum approaches to the TSP, practically constrained variants have largely been formulated using QUBO models or related encodings such as Higher-Order Binary Optimisation (HOBO) (Shuvalov et al., 2025). By contrast, QUDO and T-QUDO formulations, although promising in terms of resource efficiency and solution quality on benchmark problems, have not yet been widely applied to these constrained TSP variants. The variant considered in this study seeks a minimum-cost route with time-dependent costs and precedence constraints, and remains largely unexplored with T-QUDO and QAOA despite its relevance to dynamic travel planning.

This work makes three main contributions. First, it formulates a constrained TSP variant with a time-dependent cost function and precedence constraints, motivated by travel-itinerary optimisation, within both the QUBO and T-QUDO frameworks. Second, it implements QAOA for both formulations and evaluates them under noiseless simulation. Third, it systematically assesses the potential advantages of T-QUDO over QUBO, with particular emphasis on expressivity, quantum-resource efficiency, and solution quality.

2 Related Work

The application of quantum algorithms to the TSP and its variants has been investigated through diverse formulation strategies and algorithmic approaches.

This section surveys the relevant literature, organised into two strands, namely quantum algorithmic approaches to the TSP and alternative encodings beyond the standard QUBO formulation.

Regarding the former strand, the standard QUBO formulation of the TSP uses one-hot encoded binary variables and quadratic penalties to enforce the tour constraints (Lucas, 2014), requiring n^2 qubits for an instance with n cities. In contrast, Ruan et al. (2020) use an edge-based encoding and incorporates the closed-tour constraints directly into the mixing Hamiltonian, reducing the qubit requirement to $n(n-1)/2$ and restricting the evolution to feasible solutions only.

Alternatively, for TSP variants with real-world constraints, Salehi et al. (2022) analysed QUBO and HOBO formulations for the TSP with Time Windows, showing that integer linear programming (ILP)-inspired models improve solution quality, while edge-based encodings reduce the number of required qubits. Picariello et al. (2025) incorporated such constraints into a QUBO framework and proposed a Grover-inspired mixer with clustering QAOA (Cl-QAOA), achieving competitive approximation ratios with linear scalability up to 130 cities.

Regarding the second strand, alternative formulations such as HOBO aim to mitigate the quadratic qubit scaling of QUBO, reducing the number of variables to $n\lceil \log_2 n \rceil$ through integer encoding, at the cost of introducing higher-order interactions (Shuvalov et al., 2025). Similarly, Ramezani et al. (2024) proposed a compact binary encoding that lowers qubit requirements to $n \log_2(n)$ within QAOA.

Mata Ali (2025) introduced a unified combinatorial framework covering formulations QUDO, T-QUDO, and HOBO, and showed their formal equivalence. This work formulated the standard TSP as a T-QUDO problem, where each tour position is encoded by a single qudit of dimension n, removing the need for one-hot encoding. This approach contrasts with standard QUBO formulations, which require $\mathcal{O}(V^2)$ binary variables and a large number of couplings (Vargas-Calderón et al., 2021), leading to significant overhead on Noisy Intermediate-Scale Quantum (NISQ) hardware. Instead, T-QUDO employs integer-valued variables mapped to qudits, enabling a more compact representation and allowing certain constraints to be encoded directly within the cost structure.

Bhat et al. (2026) directly compared QUBO and T-QUDO formulations across several NP-hard problems, including the TSP. Using QAOA with the T-QUDO encoding, they achieved better approximation ratios and higher feasibility probabilities than QUBO-based approaches. This improvement is explained by the reduced Hilbert space, from 2^{n^2} to $n^n = 2^{n \log_2 n}$ (Bhat et al., 2026), which is especially beneficial at low circuit depths where QUBO often fails to find feasible solutions.

3 TSP for Travel Search Engines (TSE-TSP)

The classical TSP seeks the minimum-cost Hamiltonian cycle over a set of cities with pairwise symmetric distances. However, in practical travel planning costs

depend on the *time* at which each travel is undertaken, and additional expenses such as accommodation must be considered. The TSE-TSP generalises the TSP to capture these features. Consider an instance comprising n cities, of which one is the *depot*, indexed by $n-1$. The remaining $n_{\text{av}} = n-1$ cities must each be visited exactly once before the traveller returns to the depot. A route is represented by a permutation $\boldsymbol{\pi} = (\pi_0, \pi_1, \ldots, \pi_{n_{\text{av}}-1})$ of cities $\{0, 1, \ldots, n_{\text{av}}-1\}$, where π_p denotes the intermediate city visited at position p in the tour, and its total cost is decomposed into three components.

First, **travel costs** from city i to city j at timestep t, given by $W_{t,i,j}$ as an entry of a tensor $W \in \mathbb{R}^{n\times n\times n}$. The index $t \in \{0, \ldots, n-1\}$ represents the journey segment and the remaining indices $i, j \in \{0, \ldots, n-1\}$ denote origin and destination cities including the depot. The timestep dependence captures fare variability across journey segments. Self-loops carry zero cost, i.e. $W_{t,i,i} = 0$ for all t and i. Second, **accommodation costs** at each intermediate city, collected in a matrix $H \in \mathbb{R}^{n_{\text{av}}\times n_{\text{av}}}$, where $H_{p,i}$ is the accommodation cost at city i after it is visited at tour position p. Since accommodation costs are only associated with intermediate non-depot cities, the matrix H excludes the depot and therefore uses n_{av} timesteps rather than the n journey segments indexed in W. Consequently, $H_{p,i}$ corresponds to the accommodation cost incurred during journey segment $t+1$, as journey segment 0 represents the initial travel from the depot to the first intermediate city. Third, the **depot legs**, referring to the beginning and ending of the tour at the depot, require an outbound cost $W_{0,n-1,\pi_0}$ and a return cost $W_{n-1,\pi_{n_{\text{av}}-1},n-1}$. Consequently, the total cost function is defined by eq. (1).

$$C(\boldsymbol{\pi}) = \underbrace{W_{0,\,n-1,\,\pi_0}}_{\text{outbound}} + \sum_{t=0}^{n_{\text{av}}-2} \Big(H_{t,\,\pi_t} + W_{t+1,\,\pi_t,\,\pi_{t+1}} \Big) + \underbrace{H_{n_{\text{av}}-1,\,\pi_{n_{\text{av}}-1}} + W_{n-1,\,\pi_{n_{\text{av}}-1},\,n-1}}_{\text{return}} \tag{1}$$

The problem also incorporates **precedence constraints**. Specifically, it includes a set of directed pairs $\mathcal{P} \subset \{0, \ldots, n_{\text{av}}-1\}^2$ that imposes a partial order on the cities, requiring that for every $(a, b) \in \mathcal{P}$, city a be visited before city b. The edges of $\mathcal{P}$ form a directed acyclic graph, ensuring feasibility. Finally, given both the cost function and the set of precedence constraints, the TSE-TSP is defined by eq. (2), where $S_{n_{\text{av}}}$ denotes the symmetric group on n_{av} elements and $\pi^{-1}(a)$ is the timestep at which city a is visited.

$$\min_{\boldsymbol{\pi}\in S_{n_{\text{av}}}} C(\boldsymbol{\pi}) \quad \text{subject to} \quad \pi^{-1}(a) < \pi^{-1}(b) \;\; \forall\, (a, b) \in \mathcal{P} \tag{2}$$

3.1 QUBO Formulation

In eq. (3), the TSE-TSP is encoded as a QUBO problem suitable for both quantum annealing and QAOA (Glover et al., 2022). A one-hot encoding is adopted using a binary decision vector $\mathbf{x} \in \{0, 1\}^{n_{\text{av}}^2}$, where $x_{t,i} = 1$ if and only if city i is visited at timestep t. To represent this two-dimensional indexing within the vector $\mathbf{x}$, a composite index is defined as $\text{idx}(t, i) = t \cdot n_{\text{av}} + i$. The travel and

accommodation costs of eq. (1) translate into the first four terms of eq. (3).

$$
\begin{aligned}
C(\mathbf{x}) = & \sum_{t=0}^{n_{\mathrm{av}}-2} \sum_{i=0}^{n_{\mathrm{av}}-1} \sum_{\substack{j=0 \\ j\neq i}}^{n_{\mathrm{av}}-1} W_{t+1,i,j}\, x_{t,i} x_{t+1,j} + \sum_{i=0}^{n_{\mathrm{av}}-1} W_{0,n_{\mathrm{av}},i}\, x_{0,i} \\
& + \sum_{j=0}^{n_{\mathrm{av}}-1} W_{n_{\mathrm{av}},j,n_{\mathrm{av}}}\, x_{n_{\mathrm{av}}-1,j} + \sum_{t=0}^{n_{\mathrm{av}}-1} \sum_{i=0}^{n_{\mathrm{av}}-1} H_{t,i}\, x_{t,i} \\
& + \lambda_0 \sum_{t=0}^{n_{\mathrm{av}}-1} \sum_{i=0}^{n_{\mathrm{av}}-1} \sum_{\substack{j=0 \\ j\neq i}}^{n_{\mathrm{av}}-1} x_{t,i} x_{t,j} - \lambda_0 \sum_{t=0}^{n_{\mathrm{av}}-1} \sum_{i=0}^{n_{\mathrm{av}}-1} x_{t,i} \\
& + \lambda_1 \sum_{i=0}^{n_{\mathrm{av}}-1} \sum_{t=0}^{n_{\mathrm{av}}-1} \sum_{\substack{t'=0 \\ t'\neq t}}^{n_{\mathrm{av}}-1} x_{t,i} x_{t',i} - \lambda_1 \sum_{t=0}^{n_{\mathrm{av}}-1} \sum_{i=0}^{n_{\mathrm{av}}-1} x_{t,i} \\
& + \lambda_2 \sum_{t=0}^{n_{\mathrm{av}}-2} \sum_{t'=t+1}^{n_{\mathrm{av}}-1} \sum_{(i,j)\in\mathcal{P}} x_{t',i} x_{t,j}, \qquad x_{t,i} \in \{0,1\}.
\end{aligned} \tag{3}
$$

Additionally, since the one-hot encoding does not intrinsically enforce feasibility, three sets of Lagrangian penalties are incorporated into Q. The *one-city-per-timestep* and *one-timestep-per-city* constraints require, respectively, that exactly one city is selected at each timestep and that each city is visited at exactly one timestep. These are imposed through structurally analogous penalties defined by eq. (4) and correspond to the next four terms of eq. (3). Finally, the last term of eq. (3) refers to the precedence penalties, which discourage any assignment in which a city i that must precede city j is scheduled at a later timestep.

$$
P_0(\mathbf{x}) = \lambda_0 \sum_{t=0}^{n_{\mathrm{av}}-1} \left(\sum_{i=0}^{n_{\mathrm{av}}-1} x_{t,i} - 1 \right)^2, \quad P_1(\mathbf{x}) = \lambda_1 \sum_{i=0}^{n_{\mathrm{av}}-1} \left(\sum_{t=0}^{n_{\mathrm{av}}-1} x_{t,i} - 1 \right)^2 \tag{4}
$$

3.2 T-QUDO Formulation

In the T-QUDO formulation, defined in eq. (5), a candidate solution is a sequence $\mathbf{x} = (x_0, x_1, \ldots, x_{n_{\mathrm{av}}-1})$ of n_{av} variables, where each $x_t \in \{0, 1, \ldots, n_{\mathrm{av}} - 1\}$ denotes the intermediate city visited at tour position t.

$$
\begin{aligned}
C(\mathbf{x}) = & \sum_{t=0}^{n_{\mathrm{av}}-2} (W_{t+1,x_t,x_{t+1}}) + W_{0,n_{\mathrm{av}},x_0} + W_{n_{\mathrm{av}},x_{n_{\mathrm{av}}-1},n_{\mathrm{av}}} \\
& + \sum_{t=0}^{n_{\mathrm{av}}-2} (H_{t,x_t}) + H_{n_{\mathrm{av}}-1,x_{n_{\mathrm{av}}-1}} + \lambda_1 \sum_{t=0}^{n_{\mathrm{av}}-2} \sum_{t'=t+1}^{n_{\mathrm{av}}-1} \delta_{x_t,x_{t'}} \\
& + \lambda_2 \sum_{t=0}^{n_{\mathrm{av}}-2} \sum_{t'=t+1}^{n_{\mathrm{av}}-1} \sum_{(i,j)\in\mathcal{P}} \delta_{x_t,j} \delta_{x_{t'},i}, \qquad x_t \in \{0, 1, \ldots, n_{\mathrm{av}} - 1\}.
\end{aligned} \tag{5}
$$

The first three terms capture the travel cost, and the two following terms capture the accommodation cost at tour position t, both including the depot legs and final-stop accommodation.

The sixth term penalises repeated city assignments, whilst the seventh term penalises precedence violations. Notably, the one-city-per-timestep penalty λ_0 presented in section 3.1 is unnecessary here, since the integer encoding inherently guarantees that each variable assumes exactly one value from $\{0, \ldots, n_{\text{av}} - 1\}$.

4 Experiment Design and Implementation

The experimental evaluation uses a modular Python 3.11+ framework organised as a YAML-driven pipeline for instance generation, solver execution, metric computation, and visualisation. All solvers share a common interface that produces standardised outputs. The framework includes two QAOA simulators (Cirq and NVIDIA CUDA-Q) and a brute-force solver to obtain ground-truth optima for computing approximation ratios.

The Cirq backend supports QUBO, Virtual qudits QAOA (V-QAOA) and Native qudits QAOA (N-QAOA), whereas the CUDA-Q backend supports the QUBO and V-QAOA. For comparability, every QAOA backend uses sampling-based cost evaluation, Trotterised Quantum Annealing (TQA) initial parameters (Sack & Serbyn, 2021), and SciPy's Powell optimiser. Although the framework includes backend-agnostic noise simulation, the native-qudit noise model could not be executed reliably on the available hardware To maintain consistency across all backends and formulations, all experiments reported herein are conducted under ideal, noiseless simulation. More broadly, the use of real quantum hardware was deliberately left outside the scope of this work, owing to time and budget constraints. It is acknowledged that validating these formulations and algorithms under realistic hardware conditions constitutes an essential analysis that this work does not undertake. Nevertheless, the noiseless-simulation results presented here serve a necessary gate-keeping role because a formulation that fails to produce feasible or near-optimal solutions even under ideal conditions is unlikely to fare better under hardware noise, decoherence, and limited qubit connectivity. Establishing this ideal-case baseline therefore provides actionable evidence for deciding whether the investment of porting these formulations to current noisy, resource-constrained hardware is warranted, rather than discovering a formulation-level bottleneck only after incurring that cost.

The experimental configurations are as follows. Penalty coefficients λ_0, λ_1, and λ_2 are calibrated through a grid search over feasibility rate and mean cost relative to the brute-force optimum on small instances, thereby defining a baseline that is subsequently applied to larger problem sizes. It should be noted that this calibration is performed independently for each formulation, rather than under a joint search that controls for the differing number of penalty mechanisms across encodings. As mentioned in section 3.2, the QUBO formulation distributes its penalty budget across three terms with λ_0, λ_1, λ_2, whereas T-QUDO requires only terms with λ_1 and λ_2, since the integer encoding inherently enforces the constraint associated with λ_0. Consequently, this asymmetry in penalty-weight allocation could impact the relative behaviour of the algorithms; however, an analysis of this specific effect falls outside the scope of this work.

All settings are evaluated over 100 instances and $p \in \{1, 2, 3\}$. The CUDA-Q backend with QUBO is evaluated only at $n = 5$, because larger instances become computationally infeasible owing to the quadratic growth in the number of qubits and the corresponding optimisation cost. The CUDA-Q backend with V-QAOA is evaluated at $n \in \{5, 9\}$ because V-QAOA requires powers of two. The Cirq backend with N-QAOA is evaluated at $n \in \{5, 6, 7, 8, 9\}$. Reproducibility is ensured through deterministic seeding, and all source code and data are publicly available in https://github.com/DOKOS-TAYOS/Hotels-TSP-Tensor-QUDO and https://zenodo.org/records/19205809.

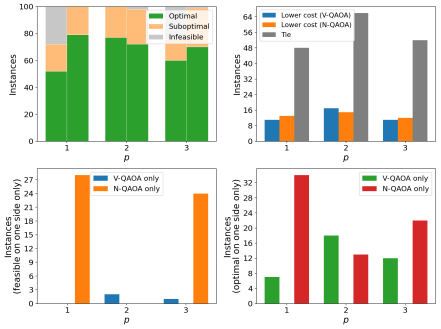
5 Results and Discussion

The results are organised around four complementary questions. The dashboards quantify reliability and pairwise wins: how often each method ends with an infeasible, feasible-suboptimal, or optimal most probable state, and what happens on the overlap of feasible instances. The relative-energy improvement plots measure how much the variational search lowers the objective with respect to its initial value. The $P(\text{opt})$ plots show how much probability mass is assigned to the optimal tour, discarding zero-mass runs. Finally, the energy-history curves reveal how the optimiser descends through the landscape and whether most of the progress happens early or late in the run. Taken together, these views allow us to separate implementation effects, formulation effects, and scaling effects. Moreover, only the statistics of the most probable state of the optimised circuit are taken into account to avoid biased comparison with unfair oversampling.

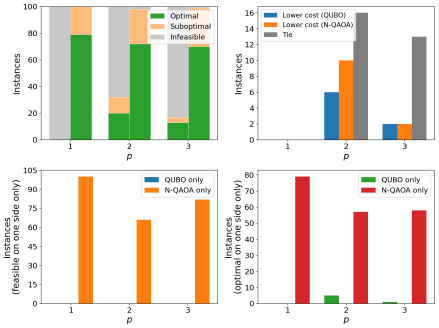
5.1 V-QAOA versus N-QAOA

Figure 1(a) shows that at $n = 5$ both tensor implementations are close in final solution quality, but not identical in reliability. N-QAOA has the clearest advantage at shallow depth: it achieves feasibility rates of 100%, 98%, and 99% for $p = 1, 2, 3$, whereas V-QAOA reaches 72%, 100%, and 76%. The same pattern appears in the optimal counts at $p = 1$ and $p = 3$: N-QAOA recovers 79 and 70 optimal instances, while V-QAOA recovers 52 and 60. At $p = 2$, however, the difference almost disappears, with V-QAOA reaching 77 optimal instances versus 72 for N-QAOA. The pairwise panels explain why the competition remains close: when both methods are feasible, ties dominate the real-cost comparison, accounting for 66.7%, 67.3%, and 69.3% of the shared feasible cohort for $p = 1, 2, 3$. The main gap therefore comes from asymmetric successes, especially the 28 and 24 instances that are feasible only for N-QAOA at $p = 1$ and $p = 3$.

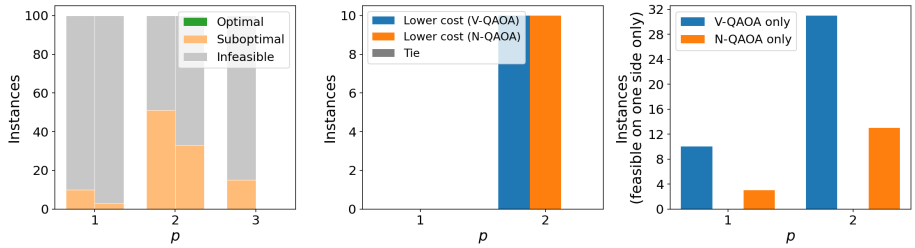
Figure 1(c) extends the same comparison to $n = 9$, where the task is already difficult for both tensor implementations. No optimal outcomes appear for either method at $p = 1$ or $p = 2$, but V-QAOA attains the larger feasible cohort: 10 feasible-suboptimal instances versus 3 for N-QAOA at $p = 1$, and 51 versus 33 at $p = 2$. At $p = 2$, the 20 instances that are feasible for both methods split evenly,



(a) $n = 5$ comparison: V-QAOA on left and N-QAOA on right.



(b) $n = 5$ comparison: QUBO on left and N-QAOA on right.



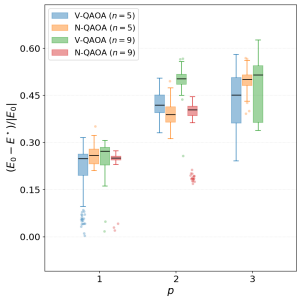
(c) $n = 9$ comparison: V-QAOA on left and N-QAOA on right.

Fig. 1: Dashboard views separate implementation effects from formulation effects. The first two panels compare V-QAOA against N-QAOA, and QUBO against N-QAOA for $n = 5$; the third compares V-QAOA against N-QAOA for $n = 9$.

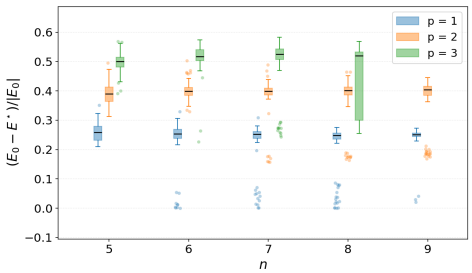
with 10 lower-cost wins for each side. At $p = 3$, no N-QAOA dashboard comparison is available because the native CPU runs became prohibitively expensive in wall-clock time. The dashboard therefore reports only the completed V-QAOA cohort in the stacked-feasibility panel, showing a decrease in performance due to non-convergence.

Figure 2(a) provides the complementary optimisation view for the implementation comparison. At $n = 5$, the mean relative energy improvement grows from about 0.211 to 0.421 and 0.431 for V-QAOA as p increases from 1 to 3, while N-QAOA moves from 0.260 to 0.391 and 0.497. At $n = 9$, V-QAOA still shows strong optimisation progress, with mean improvements of 0.252, 0.502, and 0.464 for $p = 1, 2, 3$, whereas the available N-QAOA runs reach 0.244 and 0.368 for $p = 1, 2$. Thus, deeper circuits help both implementations, but V-QAOA exhibits the broader spread, especially at $n = 9$.

The probability view at Figure 3(a) refines that picture. At $n = 5$, N-QAOA assigns more mass to the optimum than V-QAOA at $p = 1$ and $p = 3$, with mean $P(\text{opt})$ around 10^{-2}. At $p = 2$, the two are effectively tied around 10^{-2}. For $n = 9$, the visible tensor cohorts cluster around 10^{-5}.

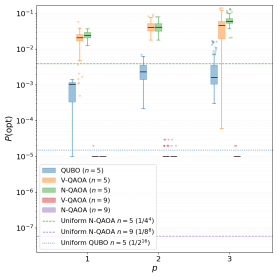


(a) Relative energy improvement for both implementations against depth.

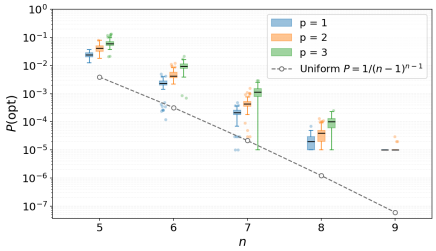


(b) Relative energy improvement for both implementations against n.

Fig. 2: Relative energy improvement comparison against depth and number of variables in different configurations.



(a) Probability of the optimum for both implementations against depth.



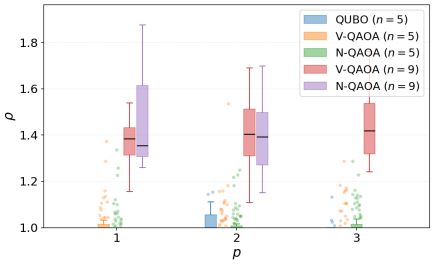
(b) Probability of the optimum for both implementations against n.

Fig. 3: Probability of the optimum after discarding zero-mass runs, for both formulations, and the probability of a random state in uniform superposition.

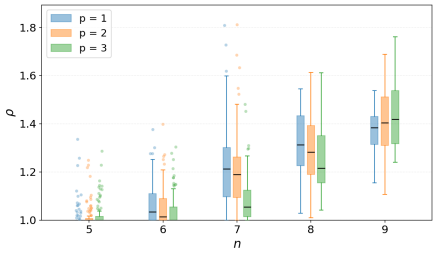
5.2 QUBO versus T-QUDO at $n = 5$

Figure 1(b) makes the main formulation effect explicit. The QUBO formulation reaches feasibility rates of 0%, 32%, and 17% for $p = 1, 2, 3$, whereas N-QAOA reaches 100%, 98%, and 99%. This carries directly into optimal recovery: QUBO finds 0, 20, and 13 optimal instances, while N-QAOA finds 79, 72, and 70. There are 100, 66, and 82 instances that are feasible only for the tensor formulation at $p = 1, 2, 3$, compared with none for QUBO. The dominant gain of T-QUDO is therefore not a mild conditional-cost advantage, but a much stronger ability to produce feasible and often optimal tours in the first place.

Figure 4(a) shows why the approximation ratio comparison must be read carefully. Conditional on feasibility, the $n = 5$ cohorts of all three methods remain fairly close to $\rho = 1$. N-QAOA gives the best mean approximation ratio at $p = 1$ and $p = 2$ (1.0167 and 1.0191), V-QAOA stays close behind (1.0253 and 1.0208), and QUBO reaches 1.0286 at $p = 2$ and even 1.0116 at $p = 3$. This



(a) Approximation ratio for both implementations against depth.



(b) Approximation ratio for both implementations against n.

Fig. 4: Approximation ratio of feasible runs for both formulations.

does not contradict the dashboard. It simply shows that once QUBO happens to return a feasible tour, its cost can still be competitive on that reduced cohort. The same left panel also includes the tensor $n = 9$ cohorts, whose much larger ratios, around 1.37 to 1.50, already indicate that the deterioration at larger sizes is a scaling effect rather than a formulation reversal.

Figure 3(a) reinforces the same interpretation with a much sharper signal. After discarding zero-mass runs, the mean $P(\text{opt})$ of QUBO is only 8.35×10^{-4}, 2.78×10^{-3}, and 3.43×10^{-3} for $p = 1, 2, 3$, whereas V-QAOA reaches 2.04×10^{-2}, 4.31×10^{-2}, and 4.55×10^{-2}, and N-QAOA reaches 2.46×10^{-2}, 4.23×10^{-2}, and 6.54×10^{-2} at $n = 5$. This is a one-to-two-order-of-magnitude advantage in probability concentration. The conclusion is therefore sharper than the approximation ratio comparison: T-QUDO does not win mainly because it lowers the conditional real cost more strongly when both methods are feasible; it wins because it turns many runs that are infeasible or weak under QUBO into feasible, and often optimal, tensor outcomes.

The uniform references make that formulation effect especially transparent. For $n = 5$, a tensor uniform state already places probability $1/4^4 \approx 3.91 \times 10^{-3}$ on the optimum, whereas uniform QUBO assigns only $1/2^{16} \approx 1.53 \times 10^{-5}$. In other words, even the unoptimised tensor superposition already places more mass on the optimum than the mean optimised QUBO run at every shown depth. The visible $n = 9$ tensor points in the same figure remain around 10^{-5} after filtering zeros, still far above the $n = 9$ tensor uniform baseline $1/8^8 \approx 5.96 \times 10^{-8}$.

5.3 Depth, scaling, and optimisation-budget effects within T-QUDO

Figures 4(a) and 4(b) show why the loss of quality at larger n should not be interpreted as a total optimisation collapse. The $n = 9$ tensor cohorts in Figure 4(a) already lie much farther from $\rho = 1$ than the $n = 5$ cohorts, and Figure 4(b) makes the same trend systematic. Yet the optimiser still reduces the energy substantially. Figure 2(b) shows that at $p = 2$ the mean relative improvement stays in a fairly narrow band around 0.4. At $p = 3$, the mean improvement is still

around $0.45 - 0.5$. The optimiser therefore keeps making meaningful energetic progress, but that progress is less and less converted into near-optimal tours as the search space expands.

Figure 3(b) makes that transition explicit. For N-QAOA at $p = 1$, the mean $P(\text{opt})$ falls from 2.46×10^{-2} at $n = 5$ to 1.00×10^{-5} at $n = 9$. At $p = 2$, it decreases from 4.23×10^{-2} at $n = 5$ to 1.18×10^{-5} at $n = 9$. Increasing the depth mitigates the decay but does not cancel it: at $p = 3$, the mean positive-mass $P(\text{opt})$ is 6.54×10^{-2} at $n = 5$, but it drops to 1.02×10^{-4} at $n = 8$.

Taken together, the results point to two distinct effects. The contrast between QUBO and T-QUDO is a formulation effect: for this TSE-TSP, the tensor encoding captures the constraint structure in a way that makes feasible and optimal outputs much more accessible under QAOA. By contrast, the regressions observed within the T-QUDO family across depth and scale are best understood as optimisation and scaling effects. This distinction supports the main conclusion of this work: T-QUDO is the more suitable formulation for this problem, and the remaining gap at larger n is more plausibly a matter of optimisation budget and computational cost than of representational inadequacy.

6 Conclusion

This work benchmarks QAOA on the TSE-TSP using two modelling approaches, a standard QUBO encoding and a tensor-based T-QUDO encoding, implemented with virtual qudits in CUDA-Q and native qudits in Cirq. The main finding is that formulation is the dominant factor; for this problem, T-QUDO more effectively concentrates probability on feasible and optimal tours than QUBO.

This advantage is not merely a marginal improvement in approximation ratio. The tensor formulation primarily converts runs that are infeasible or weak under QUBO into feasible, often optimal, solutions. At $n = 5$, this is reflected in higher optimal counts and in the probability-of-optimum analysis, where T-QUDO outperforms QUBO by one to two orders of magnitude. The difference is such that even a uniform tensor superposition assigns more probability mass to the optimum than the mean optimised QUBO run. These results are consistent with Shuvalov et al. (2025) and Bhat et al. (2026), which attribute these improvements to the reduction of the infeasible solution space, the effective Hilbert space, and native constraint satisfaction.

The comparison between virtual and native tensor QAOA is nuanced but consistent. N-QAOA is more reliable at $n = 5$, particularly at shallow depth and $p = 3$, while V-QAOA remains competitive in final cost. This indicates that the main advantage lies in the tensor encoding, whereas the choice of native or virtual qudits primarily impacts optimisation stability, runtime, and feasible instance–depth combinations.

The scaling experiments clarify current limitations. As n increases, the approximation ratio degrades and $P(\text{opt})$ decays rapidly, although the optimiser still achieves significant energy reductions. This indicates that the main bottleneck is not energy minimization, but concentrating probability on the optimal

tour within a larger search space under a fixed budget. The missing native-qudit point at $n = 9$, $p = 3$ reflects resource constraints, particularly high CPU cost, rather than a limitation of the tensor formulation.

Accordingly, the main claim of this paper is deliberately specific. For this TSE-TSP, and under the noiseless QAOA workflows evaluated, T-QUDO is the most suitable encoding among the tested alternatives. Generalising this conclusion to all QUBO/T-QUDO comparisons or combinatorial problems requires separate analysis. This work compares encodings rather than isolating the formulation choice from the distribution of penalty weights across constraint mechanisms, since QUBO and T-QUDO involve different numbers of penalty terms.

The analysis is confined to noiseless simulation, a controlled setting for assessing encoding-dependent QAOA behaviour before introducing hardware-specific effects. Hardware execution under realistic noise is the natural next validation stage. The present results do not replace that stage, but establish a principled basis for prioritising T-QUDO in more resource-intensive hardware validation.

For future work, the most relevant next steps are an analysis of the effect of the penalty-weight distribution, stronger optimisation strategies for large tensor instances, better initialisation and parameter-transfer schemes, and above all, benchmarking under matched hardware conditions and noisy execution models.

Acknowledgement

The research has been funded by ICECyL (Junta de Castilla y León) under project CCTT5/23/BU/0002 (QUANTUMCRIP).

References

Applegate, D. L., Bixby, R. E., Chvátal, V., & Cook, W. J. (2007). *The traveling salesman problem: A computational study*. Princeton University Press.

Ausiello, G., Crescenzi, P., Gambosi, G., Kann, V., Marchetti-Spaccamela, A., & Protasi, M. (1999). *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer.

Bhat, S. S., Wang, P., West, J., & Parampalli, U. (2026). Encoding matters: Benchmarking binary and D-ary representations for quantum combinatorial optimization.

Bucher, D., Kraus, N., Blenninger, J., Lachner, M., Stein, J., & Linnhoff-Popien, C. (2024). Towards robust benchmarking of quantum optimization algorithms. *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 159–170.

Bucher, D., Stein, J., Feld, S., & Linnhoff-Popien, C. (2025). Penalty-free approach to accelerating constrained quantum optimization. *Physical Review A*, *112*(6), 062605. https://doi.org/10.1103/PhysRevA.112.062605

Deller, Y., Schmitt, S., Lewenstein, M., Lenk, S., Federer, M., Jendrzejewski, F., Hauke, P., & Kasper, V. (2023). Quantum approximate optimization algorithm for qudit systems. *Physical Review A*, *107*(6), 062410.

Du, D.-Z., & Pardalos, P. M. (Eds.). (1998). *Handbook of combinatorial optimization*. Springer.

El-Sherbeny, N. A. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University - Science*, *22*(3), 123–131.

Farhi, E., Goldstone, J., & Gutmann, S. (2014). A quantum approximate optimization algorithm.

Glover, F., Kochenberger, G., Hennig, R., & Du, Y. (2022). Quantum bridge analytics i: A tutorial on formulating and using QUBO models. *Annals of Operations Research*, *314*(1), 141–183.

Gutin, G., & Punnen, A. P. (Eds.). (2002). *The traveling salesman problem and its variations* (Vol. 12). Springer. https://doi.org/10.1007/b101971

Henderson, S. G., & Nelson, B. L. (Eds.). (2006). *Simulation* (Vol. 13). Elsevier.

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

Mata Ali, A. (2025). Introduction to QUDO, tensor QUDO and HOBO formulations: Qudits, equivalences, knapsack problem, traveling salesman problem and combinatorial games.

Picariello, F., Turati, G., Antonelli, R., Bailo, I., Bonura, S., Ciarfaglia, G., Cipolla, S., Cremonesi, P., Dacrema, M. F., Gabusi, M., et al. (2025). Quantum approaches to urban logistics: From core QAOA to clustered scalability.

Qian, W., Basili, R. A. M., Eshaghian-Wilner, M. M., Khokhar, A., Luecke, G., & Vary, J. P. (2023). Comparative study of variations in quantum approximate optimization algorithms for the traveling salesman problem. *Entropy*, *25*(8), 1238.

Ramezani, M., Salami, S., Shokhmkar, M., Moradi, M., & Bahrampour, A. (2024). Reducing the number of qubits from n^2 to $n \log_2(n)$ to solve the traveling salesman problem with quantum computers: A proposal for demonstrating quantum supremacy in the NISQ era.

Ruan, Y., Marsh, S., Xue, X., Liu, Z., & Wang, J. (2020). The quantum approximate algorithm for solving traveling salesman problem. *Computers, Materials & Continua*, *63*(3), 1237–1247.

Sack, S. H., & Serbyn, M. (2021). Quantum annealing initialization of the quantum approximate optimization algorithm. *Quantum*, *5*, 491.

Salehi, Ö., Glos, A., & Miszczak, J. A. (2022). Unconstrained binary models of the travelling salesman problem variants for quantum optimization. *Quantum Information Processing*, *21*(2), 67.

Shuvalov, G. V., Krivtsova, E. N., Remnev, M. A., & Kapranov, A. V. (2025). Solving the traveling salesman problem using quantum devices: QUBO and HOBO formulations. *Russian Microelectronics*, *54*(8), 1553–1562.

Vargas-Calderón, V., Parra-A., N., Vinck-Posada, H., & González, F. A. (2021). Many-qudit representation for the travelling salesman problem optimisation. *Journal of the Physical Society of Japan*, *90*(11), 114002.

A T-QUDO and its Implementation in QAOA

To implement T-QUDO in QAOA, one may use either native qudits, i.e. quantum systems with d basis states, or virtual qudits, in which groups of $\lceil \log_2 d \rceil$ qubits emulate a single d-level system. In the former case, the number of required physical systems scales linearly with the number of variables, whereas in the latter it scales logarithmically in the local dimension d. Native qudits are especially convenient when d is not a power of two, because no extra basis states need to be removed. However, because qubit-based hardware is currently more mature and accessible than native-qudit hardware, the V-QAOA construction remains practically attractive.

The N-QAOA circuit employs three custom qudit gates. First, the **initial state preparation** is done by applying a d-dimensional Quantum Fourier Transform (QFT) gate to each qudit, yielding a uniform superposition given by eq. (6). This construction reduces to the standard Hadamard gate for $d = 2$, so in the V-QAOA it can be implemented by applying a Hadamard gate to each qubit.

$$QFT\,|0\rangle = |+_d\rangle = \frac{1}{\sqrt{d}} \sum_{k=0}^{d-1} |k\rangle \tag{6}$$

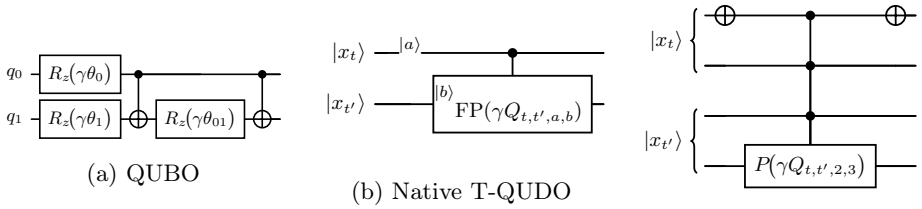
As in the QUBO case, the **cost layer** has two contributions. The first is a local interaction implemented by a diagonal gate $U_C^t(\gamma)$ of dimension d^2 acting on the pair of qudits $(t, f(t))$, where for example $f(t) = t+1$ when neighbouring qudits are coupled. The gate is

$$U_C^t(\gamma) = \sum_{x_t, x_{f(t)}=0}^{d-1} e^{-i\gamma\, Q_{t,\, x_t,\, x_{f(t)}}} \, |x_t, x_{f(t)}\rangle \, \langle x_t, x_{f(t)}| \,, \tag{7}$$

The second contribution is a two-qudit gate $U_C^{t,t'}(\gamma)$ applied to each relevant pair (t, t'):

$$U_C^{t,t'}(\gamma) = \sum_{x_t, x_{t'}=0}^{d-1} e^{-i\gamma\, Q_{t,t',\, x_t\, ,x_{t'}}} \, |x_t, x_{t'}\rangle \, \langle x_t, x_{t'}| \,, \tag{8}$$

There are n gates of type $U_C^t(\gamma)$ and $n(n-1)/2$ gates of type $U_C^{t,t'}(\gamma)$. For the resource analysis below, $U_C^t(\gamma)$ can be viewed as a special case of $U_C^{t,t'}(\gamma)$ with $t' = t+1$. Each gate can be decomposed into $\mathcal{O}(d^2)$ controlled qudit gates. Mata Ali (2025) defines a *focus phase gate* $FP(\theta, b)$, which applies a phase $e^{i\theta}$ only when the target qudit is in state $|b\rangle$ and acts as the identity otherwise. A tensor element $Q_{i,\,j,\,a,\,b}$ can then be encoded by applying a controlled $FP(\theta, b)$ to the j-th qudit with the i-th qudit acting as a control on state $|a\rangle$. This construction enables a direct implementation of T-QUDO cost functions in N-QAOA and can be translated to V-QAOA by using multi-controlled phase gates together with X masks that select the binary pattern encoding the pair (a, b), as illustrated in Fig. 5. However, this implementation requires higher connectivity than typical



(a) QUBO (b) Native T-QUDO (c) Virtual T-QUDO

Fig. 5: Cost-layer primitives for applying problem-dependent phases in QAOA under the three encodings used in this work. **(a)** QUBO: after mapping to an Ising Hamiltonian, diagonal terms appear as single-qubit R_z rotations and off-diagonal couplings as R_{ZZ} interactions, here drawn in the standard CNOT–R_z–CNOT decomposition. **(b)** N-QAOA on qudits: the focus-phase gate $\mathrm{FP}(\theta)$ acts as the identity unless the control qudit is in $|a\rangle$ and the target in $|b\rangle$, in which case it applies $e^{i\theta}$ on that component. **(c)** V-QAOA: each d-level decision is stored in $\lceil \log_2 d \rceil$ qubits (two qubits per register for $d = 4$); the same phase is realised with qubit gates, here schematically as a multiply controlled P with three control wires and the rotation on the fourth qubit. In the example, $a = 2, b = 3$.

QUBO QAOA, and requires each of the $FP(\theta, b)$ gates to be decomposed into $\mathcal{O}(d)$ CNOT gates without ancillas, or $\mathcal{O}(\log_2(d))$ CNOT gates with $\mathcal{O}(\log_2(d))$ ancillas.

This expressive power entails an increase in circuit complexity. The number of controlled $FP(\theta, b)$ per QAOA layer scales with the number of non-zero tensor elements, which for fully dense tensors reaches up to $O(n^2\, d^2)$ for n variables of dimension d (or $O(n^2\, d^2 \log d)$ with ancilla and $O(n^2\, d^3)$ without ancilla for V-QAOA), compared with at most $O(n^2)$ two-body ZZ interactions per layer in a standard QUBO Hamiltonian. The overall circuit depth therefore depends on the sparsity of the cost tensor for the specific problem instance under consideration. However, as we will see in the next section, the reduction of the number of variables partially compensates for this increase.

Third, the **mixer layer**, proposed by Deller et al. (2023), is defined in eq. (9) via the d-dimensional cyclic-shift operator X_d, namely the generalised Pauli-X gate. This generates transitions between all d basis states via rotations in the cyclic-shift eigenbasis. For $d = 2$ this reduces to $R_x(2\beta)$; for $d > 2$ it couples all levels through a single unitary rather than mixing individual bits independently. In the V-QAOA, this is directly performed via $R_x(2\beta)$ gates at each qubit, which is not equivalent.

$$U_M(\beta) = \exp\left(-i\,\beta\,\frac{X_d + X_d^\dagger}{2}\right), \tag{9}$$

B Quantum resources for the TSE-TSP T-QUDO

The formulation presented in eq. (5) can be formulated as using two tensor terms. The first one, the *adjacent-step tensor* $Q_{t,a,b}$, shown in eq. (10), captures the travel cost from city a at timestep t to city b at timestep $t{+}1$, as well as the accommodation at timestep t for $t = 0, \ldots, n_{\text{av}} - 2$ and $a, b \in \{0, \ldots, n_{\text{av}} - 1\}$.

$$Q_{t,a,b} = W_{t+1,\,a,\,b} + H_{t,\,a} + \begin{cases} W_{0,\,n_{\text{av}},\,a} & \text{if } t = 0, \\ W_{n_{\text{av}},\,b,\,n_{\text{av}}} + H_{n_{\text{av}}-1,\,b} & \text{if } t = n_{\text{av}} - 2, \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The second one, the *long-range tensor* $Q_{t,t',a,b}$, shown in eq. (11), encodes constraint penalties coupling non-adjacent timesteps. The first term penalises repeated city assignments, whilst the second penalises precedence violations in which a city α that must precede β appears at a later timestep.

$$Q_{t,t',a,b} = \lambda_1\, \delta_{a,b} + \lambda_2 \sum_{(\alpha,\beta)\in\mathcal{P}} \delta_{a,\beta}\, \delta_{b,\alpha}, \qquad 0 \le t < t' \le n_{\text{av}} - 1. \tag{11}$$

The reformulation of eq. (5) is given by eq. (12), in which both explained tensors are jointly normalised by their maximum absolute entry, and the normalisation factor is stored for cost recovery in original units.

$$C_{\text{T-QUDO}}(\mathbf{x}) = \sum_{t=0}^{n_{\text{av}}-2} Q_{t,\,x_t,\,x_{t+1}} + \sum_{t=0}^{n_{\text{av}}-2} \sum_{t'=t+1}^{n_{\text{av}}-1} Q_{t,\,t',\,x_t,\,x_{t'}}. \tag{12}$$

In its straightforward implementation, the latter formulation leads to $\mathcal{O}(n_{\text{av}}^4)$ phase operations. Each $(2\log_2 n_{\text{av}} - 1)$-controlled phase gate can be decomposed into $\mathcal{O}(\log_2 n_{\text{av}})$ CNOT gates with $\mathcal{O}(\log_2 n_{\text{av}})$ ancillas or $\mathcal{O}(n_{\text{av}})$ CNOT gates without ancillas, so V-QAOA requires $\mathcal{O}(n_{\text{av}}^5)$ CNOT gates without ancillas. This is asymptotically larger than in the QUBO formulation.

However, because $Q_{t,t',a,b}$ is used only for constraints, this cost can be reduced substantially. If the number of precedence constraints is sparse, e.g., $|\mathcal{P}| = O(n_{\text{av}})$, then the precedence contribution scales as $O(n_{\text{av}}^3)$. In the dense case, where $|\mathcal{P}| = O(n_{\text{av}}^2)$, this contribution scales as $O(n_{\text{av}}^4)$. In the sparse case, an additional qudit can be used as an indicator to implement the non-repetition penalty following an idea similar to the IF-QAOA (Bucher et al., 2025).

In the N-QAOA, the non-repetition constraint is performed for every (t, t') pair of qudits, comparing if they have the same state. We wish to implement a unitary that applies a phase only to those computational-basis components for which the two registers encode the same classical value. The target operation is given by eq. (13), where $\delta_{x_t,x_{t'}}$ is the Kronecker delta. This operation can be implemented coherently, without measurement, by means of a reversible comparator followed by a compute–phase–uncompute procedure.

$$U_{\text{eq}}(\phi)\,|x_t\rangle\,|x_{t'}\rangle = e^{i\phi\delta_{x_t,x_{t'}}}\,|x_t\rangle\,|x_{t'}\rangle \tag{13}$$

For two qudits of local dimension d, an ancillary qudit c of the same dimension, initialised in $|0\rangle_c$, is introduced to compute the modular difference between

Table 1: Quantum resource requirements for encoding the TSE-TSP with n cities, where $n_{\mathrm{av}} = n - 1$ visitable cities, assuming sparse precedence constraints.

Formulation	**Quantum resources**	Basic gate count
QUBO	n_{av}^2 qubits	$\mathcal{O}\left(n_{\mathrm{av}}^3\right)$
V-QAOA without ancilla	$n_{\mathrm{av}} \lceil \log_2 n_{\mathrm{av}} \rceil$ qubits	$\mathcal{O}\left(n_{\mathrm{av}}^5\right)$
V-QAOA with ancilla	$(n_{\mathrm{av}} + 1) \lceil \log_2 n_{\mathrm{av}} \rceil$ qubits	$\mathcal{O}\left(n_{\mathrm{av}}^4 \log n_{\mathrm{av}}\right)$
IF-V-QAOA	$(n_{\mathrm{av}} + 1) \lceil \log_2 n_{\mathrm{av}} \rceil$ qubits	$\mathcal{O}\left(n_{\mathrm{av}}^3 \log n_{\mathrm{av}}\right)$
N-QAOA	n_{av} qudits of dimension n_{av}	$\mathcal{O}\left(n_{\mathrm{av}}^4\right)$
IF-N-QAOA	$n_{\mathrm{av}} + 1$ qudits of dimension n_{av}	$\mathcal{O}\left(n_{\mathrm{av}}^3\right)$

the two basis labels into the ancilla, as given by eq. (14). Hence, the ancilla state coherently encodes equality of the two registers, as shown in eq. (15). Subsequently, the phase gate shown in eq. (16) can be applied to the ancilla and finally uncompute the modular difference.

$$|x_t\rangle |x_{t'}\rangle |0\rangle_c \mapsto |x_t\rangle |x_{t'}\rangle |x_t - x_{t'} \bmod d\rangle_c \tag{14}$$

$$x_t = x_{t'} \quad \Longleftrightarrow \quad x_t - x_{t'} \equiv 0 \pmod{d} \tag{15}$$

$$P_0(\phi) = e^{i\phi} |0\rangle \langle 0| + \sum_{k=1}^{d-1} |k\rangle \langle k| \tag{16}$$

For V-QAOA, the same idea can be implemented in binary form using an ancillary $\log_2(d)$-qubit register initialised in $|0\rangle^{\otimes \log_2(d)}$. As given by eq. (17), the bitwise XOR is computed into the ancilla by applying CNOT gates from the corresponding qubits of the two registers. Consequently, equality between the two registers is encoded in the ancilla being in the all-zero state, as shown in eq. (18).

$$|x_t\rangle |x_{t'}\rangle |0\rangle^{\otimes \log_2(d)} \mapsto |x_t\rangle |x_{t'}\rangle |x_t \oplus x_{t'}\rangle . \tag{17}$$

$$x_t = x_{t'} \quad \Longleftrightarrow \quad x_t \oplus x_{t'} = 0 \tag{18}$$

A phase can then be applied only when the ancilla is in $|0\rangle^{\otimes \log_2(d)}$ by using X gates to convert zero-controls into one-controls, applying a $(\log_2(d) - 1)$-controlled phase, and then undoing those X gates. Each comparison requires $\mathcal{O}(\log_2 d)$ CNOT gates for the XOR computation and $\mathcal{O}(d)$ additional gates for the controlled phase, because no extra ancillas are available.

Taking into account that there are $\mathcal{O}(n_{\mathrm{av}}^2)$ comparisons and $d = n_{\mathrm{av}}$, enforcing the non-repetition constraint requires $\mathcal{O}(n_{\mathrm{av}}^2)$ gates for N-QAOA and $\mathcal{O}(n_{\mathrm{av}}^3)$ CNOT gates for V-QAOA. The local-interaction term requires $\mathcal{O}(n_{\mathrm{av}}^3)$ gates for N-QAOA, the minimal possible taking into account the basic information of the problem, and $\mathcal{O}(n_{\mathrm{av}}^3 \log n_{\mathrm{av}})$ gates for V-QAOA, making use of the same ancillas to decompose the multicontrolled phase gates in $\mathcal{O}(\log n_{\mathrm{av}})$ for the $\mathcal{O}(n_{\mathrm{av}}^3)$ elements. Therefore, the local-interaction term remains the dominant contribution. The sparsity of the QUBO formulation for this case yields $\mathcal{O}(n_{\mathrm{av}}^3)$ interaction terms, so the same order of CNOT gates.

Table 1 summarises the resource requirements across all formulations. To keep the comparison fair and focused on vanilla QAOA, we therefore evaluate only the standard QAOA variants rather than IF-QAOA extensions.