The Princeton Journal of Interdisciplinary Research 2025 (Vol. 1, No. 2) - ISSN 3069-8200

QUBO-Based Optimization of Hybrid Renewable Energy Portfolios: Benchmarking Classical and Quantum-Inspired Solvers

Devavrat J. Gokal

Hill Spring International School, Mumbai

Abstract

This paper considers the problem of optimal site selection for hybrid solar-wind energy systems by formulating it as a Quadratic Unconstrained Binary Optimization (QUBO) model, which seeks to maximize annual energy yield under budgetary and site-exclusivity constraints. The performance of a classical metaheuristic solver (MQLib) is benchmarked against a quantum-inspired solver (D-Wave's Simulated Annealing) using empirical solar and wind resource data across up to 60 candidate sites in Gujarat and Maharashtra, India. Results show that D-Wave's simulated annealing consistently yields higher objective values and exhibits more favorable computational scaling as the problem size increases. In addition, a related location-allocation problem for energy system planning, previously optimized by Ajagekar and You in 2019 using Gurobi and the D-Wave 2000Q is revisited. The results demonstrate that MQLib outperforms both solvers across all benchmark instances in terms of solution quality and computational runtimes. The relative strengths and limitations of classical and quantum-inspired solvers are examined, with emphasis on trade-offs in scalability, solution quality, and real-world applicability. While D-Wave's current simulated annealing shows better performance in this setting, this advantage may diminish with the adoption of newer hardware such as the D-Wave Advantage2 system, which is expected to offer higher qubit connectivity and improved noise performance.

Keywords: QUBO, Combinatorial Optimization, Renewable Energy, Metaheuristic Algorithms, Quantum Computing

Introduction

With rising global energy demand and increasing pressure to decarbonize, the deployment and effective optimization of hybrid renewable energy systems have become essential to modern infrastructure planning (Jackson et al., 2018). As the world transitions to cleaner energy, renewable technologies like solar photovoltaics and wind turbines are increasingly being integrated into energy portfolios to enhance efficiency, lower greenhouse gas emissions, reduce dependence on fossil fuels, and strengthen grid reliability. When deployed across geographically dispersed sites, hybrid renewable energy systems can significantly mitigate generation intermittency, enhance energy availability, and increase operational flexibility (Maradin, 2021).

The increasing adoption of such hybrid systems, however, introduces a range of challenges due to the uneven distribution and fluctuating nature of renewable resources. This creates a complex infrastructure planning problem of identifying the most effective configuration and geographic allocation of technologies across a set of candidate sites to maximize total energy yield under a fixed budget. Addressing this optimization problem is an essential step toward the efficient and scalable deployment of hybrid renewable energy systems that can meet both environmental and economic objectives.

Achieving these outcomes requires a careful evaluation of trade-offs between energy yield, cost, land use, and site-specific performance. Only through the strategic optimization of technology placement can these systems deliver robust, scalable benefits within real-world financial and spatial constraints.

In this context, optimization algorithms play a vital role by enabling the systematic exploration of complex, constrained decision spaces and supporting the identification of solutions that balance competing objectives (Kochnderfer & Wheeler, 2019). The hybrid site-selection problem falls into the class of NP-hard combinatorial optimization problems, where the solution space grows exponentially with the number of candidate sites (Cheeseman et al., 1991). This makes the choice of solver and optimization framework crucial to achieving high-quality results within reasonable computational limits.

This paper explores the application of novel computational tools to tackle the hybrid renewable energy site selection problem using a QUBO-based formulation. Specifically, the effectiveness of MQLib (Abdel-Basset et al., 2018) and D-Wave's Simulated Annealing (McClean et al., 2016) is compared in terms of solution quality and computational runtime for this complex combinatorial optimization (Bernhard Korte et al., 2006) task. MQLib, a powerful metaheuristic solver that has demonstrated strong performance in domains such as scheduling and graph partitioning, remains underexplored in the context of energy systems planning, motivating its evaluation in this work. Similarly, recent advancements in D-Wave's algorithmic capabilities and hardware have positioned its Simulated Annealing sampler as a competitive tool for optimization problems. This paper starts by reviewing quantum computing fundamentals and QUBO solvers, emphasizing their relevance to large-scale, real-world optimization problems. It then examines a facility location–allocation problem in the context of energy systems optimization, revealing that an optimization algorithm not previously considered (MQLib) significantly outperforms the results reported by Ajagekar and You (2019), who used a Gurobi solver and the D-Wave 2000Q system. Next, a novel QUBO formulation for hybrid renewable energy site selection is introduced, incorporating budget constraints and empirical solar–wind resource data. The performance of classical (MQLib) and quantum-inspired (D-Wave) solvers is benchmarked, with an analysis of trade-offs in runtime and solution quality across problem sizes. Finally, implementation challenges and limitations are discussed, along with proposals to extend the model with more realistic constraints and to evaluate additional hybrid and quantum solvers.

Preliminaries

Quantum Computing Overview

Quantum computation explores how quantum mechanical principles (P A M Dirac, 2013) can be harnessed to perform information processing tasks beyond the capabilities of classical systems. Just as a bit is the fundamental unit of classical computing, the qubit serves as the fundamental unit of quantum computing. Unlike a classical bit, which can exist only in one of two states, 0 or 1, a qubit can exist in a superposition (Friedman et al., 2000) of both states simultaneously. This allows it to encode a combination of 0 and 1 at the same time, enabling quantum computers to process information in parallel. Another key property that distinguishes qubits from classical bits is entanglement (Horodecki et al., 2009), a phenomenon in which two or more qubits share a joint quantum state. Changes to one qubit instantaneously affect the other, regardless of distance. These properties of superposition and entanglement underpin the potential of quantum computers to solve certain complex problems more efficiently than classical systems, a capability referred to as quantum advantage. Upon measurement, a qubit's superposed state collapses to one of the basis states, yielding an outcome with a probability determined by its quantum amplitudes, allowing quantum algorithms to explore vast solution spaces in parallel. By leveraging these quantum features, algorithms such as Shor's (Shor, 1994), which factors large integers in polynomial time, and Grover's (Grover, 1996), which achieves a quadratic speedup for unstructured search problems, demonstrate clear advantages over their classical counterparts. These capabilities are harnessed through different models like the quantum circuit model and the quantum annealing model, each suited to distinct classes of problems.

Quantum Circuit Model

The quantum circuit model is the most widely studied framework for quantum computation. It is typically implemented using architectures such as superconducting circuits, trapped ions, or other technologies that allow precise control over two-level quantum systems. Computation is performed by applying a sequence of quantum gates, which are unitary operations acting on one or two qubits and serve a role similar to that of logic gates in classical computing. After the gate operations are complete, qubits are measured, collapsing their quantum states into classical outcomes.

In practice, the complexity of quantum circuits is currently limited by decoherence, gate errors, and noise, which constrain the size of problems that can be reliably solved. Despite these challenges, various quantum algorithms such as the Quantum Approximate Optimization Algorithm (QAOA) (Farhi et al., 2014) and the Variational Quantum Eigensolver (VQE) (Tilly et al., 2022) have been developed within this model, with platforms like IBM's Qiskit (Javadi-

Abhari et al., 2024) providing open-source tools for designing, simulating, and running quantum circuits on real quantum hardware.

Quantum Annealing Model

The quantum annealing model offers an alternative paradigm tailored to combinatorial optimization. A quantum annealer is a specialized quantum computer that uses quantum mechanics to efficiently search for optimal solutions in complex, high-dimensional spaces. The process begins by initializing qubits in the ground state of a simple transverse-field Hamiltonian, which is then gradually transformed to encode a problem-specific cost function, typically in the form of an Ising Hamiltonian or Quadratic Unconstrained Binary Optimization (QUBO) formulation (Hauke et al., 2020).

According to the adiabatic theorem (Kato, 1950), if the evolution is slow enough, the system remains in its ground state throughout, yielding a low-energy solution upon measurement. This process allows the system to naturally converge to the binary decision vector that minimizes the QUBO objective, regardless of the size or complexity of the problem space. As a result, quantum annealing offers a powerful approach for solving a broad class of optimization problems.

QUBO Solvers

QUBO is a standard mathematical framework used to model a wide variety of combinatorial optimization problems (Glover et al., 2019). In a QUBO problem, the goal is to find a binary vector $x \in \{0, 1\}^n$ that minimizes a quadratic objective function of the form:

$$\min_{x \in \{0,1\}^n} x^t Qx$$

where $Q \in R^{n \times n}$ is a symmetric square matrix of real-valued coefficients. The diagonal entries Q_{ii} represent the linear costs associated with choosing variable $x_i = 1$, while the off-diagonal entries Q_{ij} represent the interaction cost between variables x_i and x_j.

The strength of the QUBO framework lies in its flexibility to represent a wide range of NP-hard problems such as graph coloring, scheduling, and facility location-allocation within a single mathematical form. This makes it compatible with both classical heuristics and quantum-inspired solvers. The remainder of this section outlines two such solution approaches: classical metaheuristics using MQLib and quantum annealing using D-Wave.

MQLib

MQLib is an open-source C++ library that provides a comprehensive suite of high-performance metaheuristic algorithms (Abdel-Basset et al., 2018) for solving QUBO problems. It serves as a benchmarking platform, enabling the evaluation of diverse classical heuristic types such as tabu search (Glover & Laguna, 2002), simulated annealing (P.J. van Laarhoven & Aarts, 2013), and genetic algorithms (Sivanandam et al., 2008) across a wide range of combinatorial optimization tasks. With each algorithm finely tuned to deliver competitive performance, and the library's customizable parameters, it supports rapid experimentation and comparative analysis across varying problem structures.

D-Wave

D-Wave Systems has developed quantum annealers that natively solve QUBO problems using a specialized quantum processing unit (QPU) (Britt & Humble, 2017) comprising over 5,000 superconducting niobium qubits, which operate at cryogenic temperatures to maintain quantum coherence. These qubits are arranged in sparse, programmable topologies such as the

Chimera and Pegasus graphs, with couplers connecting qubits within and across unit cells to support flexible problem embedding. Despite challenges such as limited qubit connectivity and analog noise, D-Wave's devices can handle QUBO problems with thousands of variables, making them well-suited for large-scale optimization tasks across a wide range of areas (McClean et al., 2016). For small-scale problems, D-Wave's Simulated Annealing closely mirrors the behavior of its quantum hardware, offering a practical and accessible proxy for benchmarking quantum-inspired optimization strategies.

Benchmarking Classical Solvers for the Facility Location-Allocation Problem

Overview

Ajagekar and You (2019) explored combinatorial optimization in energy systems by framing a Quadratic Assignment Problem (QAP) (Lawler, 1963), a well-known NP-hard problem, as a QUBO formulation. Their specific QAP instance tackled the facility location-allocation problem within energy systems, optimizing the placement of power generation plants to minimize the total energy transportation costs. To rigorously benchmark their approach, they borrowed problem instances from the quadratic assignment problem library (QAPLIB), which provides data on the best-known solutions. Although Ajagekar and You (2019) demonstrated that classical solvers such as Gurobi could effectively address QUBO formulations, solution quality and computational runtime remained constrained by solver architecture. To improve upon these results, the problem was re-implemented using MQLib, a metaheuristic-based QUBO solver, resulting in substantial enhancements in performance.

Problem Description

The problem aims to place n power plants to n regions in such a way that the cost of interplant transportation is minimized. Two $n \times n$ matrices $C = |C_{ij}|$ and $T = |T_{pq}|$ are given, where C_{ij} is the cost of transporting one unit of energy from location i to location j , and T_{pq} is the number of units of energy to be transported from plant p to plant q. The binary variable $x_{pi} \in \{0,1\}$ equals 1 if plant p is assigned to location i, and 0 otherwise. The scalar A is a penalty weight applied to constraint terms in the QUBO; it is chosen such that $A \gg \max\{C_{ij}T_{pq}\}$ to ensure that feasibility is strongly prioritized over minimizing transportation cost. This formulation yields the final QUBO as shown in Equation 1.

$$C(x) = 2nA + \sum_i \sum_j \sum_p \sum_q C_{ij} T_{pq} x_{pi} x_{qj} + 2A \sum_i \sum_p \sum_{q>p} x_{pi} x_{qi} + 2A \sum_p \sum_i \sum_{j>i} x_{pi} x_{pj} - 2A \sum_p \sum_i x_{pi} \tag{1}$$

Beating existing solvers with MQLib

This section presents a summary of the results obtained using MQLib and those reported by Ajagekar and You (2019).

Table 1.

Comparison of computational results from different solvers on QAP instances.

No. of facilities	Best known solution	Gurobi Solver[1]		Quantum Solver (D-Wave 2000Q)		MQLib Solver[2]	
		time (s)	obj. fun.	time (s)	obj. fun.	time (s)	obj. fun.
3	24	1.33	24	0.024	24	**0.000017**	**24**

[1] Intel(R) Core (TM) i7-6700 3.40 GHz CPU.
[2] Apple M4 4.40 GHz CPU.

4	32	1.48	32	0.062	32	**0.000017**	**32**
5	58	1.50	58	0.066	58	**0.000022**	**58**
6	94	1.35	94	0.043	94	**0.000021**	**94**
8	214	1.96	214	0.127	214	**0.000146**	**214**
9	264	2.01	264	445.23	264	**0.000595**	**264**
12	578	325.68	578	1946.12	578	**0.117367**	**578**
14	1014	42010.42	1014	1008.70	1026	**0.586288**	**1014**
15	1150	Timeout	1160	986.19	1160	**0.748820**	**1152**
17	1732	Timeout	1750	921.71	1786	**13.652437**	**1732**
20	2570	Timeout	2674	744.76	2640	**17.857517**	**2586**

Across all tested instances, ranging from 3 to 20 facilities, MQLib consistently outperformed both the Gurobi solver and the D-Wave 2000Q, with better objective function values and significantly faster computational times. These results underscore MQLib's effectiveness in solving complex QUBO problems and establish it as a viable method for large-scale combinatorial optimization. Validation of MQLib against established QUBO benchmark instances substantiates its reliability and computational efficiency, thereby enhancing the credibility of its application in subsequent research.

Novel QUBO Formulation for Hybrid Energy Portfolio

Problem Description

Here, the problem of optimally allocating hybrid renewable energy technologies under cost and exclusivity constraints is considered. More precisely, the analysis focuses on 60 candidate sites across inland and coastal regions of Gujarat and Maharashtra (states in India), selected for their diverse climates, favorable topography, and established reputations as renewable energy hubs. To ensure rigor and real-world relevance, renewable resource performance data were sourced from the National Renewable Energy Laboratory (NREL) for each site. Hourly wind speed data at 80 m hub height were obtained from the NREL Wind Toolkit and converted to annual energy yields using a standard reference turbine power curve. Similarly, solar resource data comprising hourly irradiance and meteorological variables were drawn from the NREL NSRDB and processed using the PVWatts v6 model to estimate annual solar output in MWh per MW installed. Region-specific capital expenditures and operation expenditures for both technologies were derived from the NREL 2023 Annual Technology Baseline (ATB). All inputs were integrated into a Python-based SAM pipeline to produce the site-level yield and cost parameters used in the QUBO.

The objective is to identify the optimal configuration of solar photovoltaic units and wind turbines across a set of n candidate sites, indexed by $i \in \{1, \dots, n\}$, to maximize total annual energy yield while ensuring that the total installation cost does not exceed a fixed budget B, and that no site hosts more than one technology. For each site i, two binary decision variables are defined: $x_i^{(s)} \in \{0,1\}$, which equals 1 if a 1 MW solar photovoltaic unit is placed at site i, and $x_i^{(w)} \in \{0,1\}$, which equals 1 if a 1 MW wind turbine installation is placed at the same site. The energy yield contributed by each technology is site-specific and denoted by $\eta_{s,i}$ for solar and $\eta_{w,i}$ for wind, both expressed in MWh/year per MW of installed capacity.

The total energy yield from a given configuration x is the sum of the yields from each site, forming the objective function to be maximized, as shown in Equation 2. This is subject to a budget constraint, which ensures the total installation cost does not exceed a fixed budget B (in thousands of USD), where c_s and c_w denote the per-MW costs for solar photovoltaic units and wind turbines (including capital and 20-year operating expenditures); and a exclusivity constraint, which ensures that at most one technology is selected per site.

$$\max_{\{x_i^{(s)}, x_i^{(w)}\}} \sum_{i=1}^{n} \left(\eta_{s,i} \cdot x_i^{(s)} + \eta_{w,i} \cdot x_i^{(w)} \right) \quad (2)$$

s.t

$$\sum_{i=1}^{n} \left(c_s \cdot x_i^{(s)} + c_w \cdot x_i^{(w)} \right) \leq B$$

$$x_i^{(s)} + x_i^{(w)} \leq 1, \quad \forall i = 1, \dots, n$$

This constrained maximization problem can be reformulated as a QUBO problem, making it compatible with both classical metaheuristics, such as those implemented in MQLib, and quantum-inspired solvers like D-Wave's simulated annealing. In a QUBO formulation, both the objective and the constraints are embedded into a single quadratic cost function over binary variables.

The fixed budget B is calculated as shown in Equation 3, assuming a constant $\alpha = 0.6$:

$$B = \alpha \times n \times \frac{c_s + c_w}{2} \quad (3)$$

To incorporate the constraints, the budget and exclusivity conditions are expressed as quadratic penalty terms as shown in Equations 4 and 5:

$$A\left(\sum_{i=1}^{n}\left(c_s x_i^{(s)} + c_w x_i^{(w)}\right) - B\right)^2 \tag{4}$$

$$C\sum_{i=1}^{n} x_i^{(s)} x_i^{(w)} \tag{5}$$

The penalty weights A and C are selected such that $A, C \gg \max\{\eta_{s,i}, \eta_{w,i}\}$, ensuring that constraint violations are heavily penalized relative to the energy yield. This encourages the solver to prioritize feasible configurations during optimization. The constant term AB^2 is added to preserve the equivalence of the penalized cost function to the original constrained formulation.

This yields the final QUBO cost function as shown in Equation 6:

$$C(x) = AB^2 - \sum_{i=1}^{n}\left(\eta_{s,i} x_i^{(s)} + \eta_{w,i} x_i^{(w)}\right) + A\sum_{i=1}^{2n}\sum_{j=1}^{2n} c_i c_j x_i x_j - 2AB\sum_{i=1}^{2n} c_i x_i + C\sum_{i=1}^{n} x_i^{(s)} x_i^{(w)} \tag{6}$$

Results

To assess the performance of classical and quantum-inspired solvers on the proposed QUBO formulation for hybrid renewable energy site selection, problem instances of varying sizes ranging from 10 to 60 candidate sites were systematically analyzed. This evaluation considers two solvers: MQLib, which implements high-performance classical metaheuristics, and D-Wave's Simulated Annealing sampler, a quantum-inspired optimizer accessed via the Ocean SDK. For each problem size, the QUBO was constructed as shown, and both solvers were applied to identical instances to ensure methodological consistency.

The MQLib runs were executed on an Apple MacBook Pro equipped with an Apple M4 chip (4.40 GHz, 16-core CPU, 20-core GPU, 24 GB RAM), while the D-Wave results were obtained through its cloud-based SDK using the native SimulatedAnnealingSampler interface. The primary performance metrics recorded were runtime (in seconds) and the final objective function value, adjusted for any constant offset after being returned by each solver.

Table 2.

Comparison of MQLib and D-Wave Simulated Annealing on site allocation QUBO instances.

No. of sites	**MQLib Solver**		**D-Wave Simulated Annealing**	
	time (s)	**obj. fun.**	**time (s)**	**obj. fun.**
10	1.112338	13 055.41	**0.0189**	**12 575.18**
20	1.345481	24 794.81	**0.0698**	**23 485.21**
30	3.035778	35 154.13	**0.6660**	**33 311.80**
40	11.985475	**48 654.44**	**2.2532**	48 845.15
50	9.985112	290 830 347 474.30	**4.6461**	**59 015.80**
60	**5.823809**	418 803 279 244.00	7.0113	**17 161 971.02**

The results presented in Table 2 highlight a consistent performance gap between the two solvers as the value of n increases. D-Wave’s Simulated Annealer returns lower objective function values across most instances, indicating superior solution quality relative to MQLib. Its computational runtimes remain extremely fast for small- and medium-sized instances and increase gradually with problem size.

MQLib, in contrast, consistently yields higher objective values, especially for larger instances such as $\mathrm{n} = 50$ and $n = 60$, indicating a decline in solution quality and scalability at higher problem sizes. It also exhibits greater fluctuation in runtimes as n increases, suggesting sensitivity to problem structure and potential inefficiencies in large-scale search.

Due to the probabilistic nature of both solvers, the results vary across runs and problem sizes. At $n = 60$, both MQLib and D-Wave's Simulated Annealer produce high objective values, reflecting the increased difficulty of navigating the exponentially expanding solution space and its impact on solution quality.

Discussion

The decline in solver performance as n increases is directly related to how the size and complexity of the QUBO model scale with problem size. Each instance adds $2n$ binary variables (Cox, 2018), which causes the number of possible solutions to grow exponentially. This rapid growth makes the optimization landscape significantly more complex, leading to increased difficulty in finding high-quality solutions as the problem size grows.

In the case of MQLib, the addition of hard constraints such as budget limits and one-site-per-location exclusivity are enforced through penalty terms within the QUBO. These penalties were set to $\mathrm{A} = \mathrm{C} = 2 \times \max(\eta_{s,i}, \eta_{w,i})$ to ensure constraint satisfaction. This scaling can lower the effectiveness of classical metaheuristics by increasing the difficulty of escaping local minima, especially as the problem size increases.

For D-Wave's Simulated Annealer, performance remains sensitive to annealing schedule parameters. To improve convergence, hyperparameters (Bischl et al., 2023) such as num_reads, num_sweeps, and beta_range were manually tuned for each instance. While this tuning supports strong performance across all tested values of n, the algorithm faces growing difficulty navigating high-dimensional energy landscapes, which can be attributed to the annealer's difficulty in identifying effective cooling schedules, increasing the likelihood of convergence to suboptimal local minima. This trade-off between solution quality and runtime must be managed carefully, as searching a larger problem space for a longer duration may lead to better-quality solutions, but at the cost of increased computational time. Although D-Wave's Simulated Annealer closely replicates the behavior of quantum annealers on small QUBO instances, this advantage diminishes with scale, suggesting the need for hybrid or quantum methods to address real-world problem complexity (Koshka & Novotny, 2020).

Conclusion

This study applied a QUBO-based approach to the hybrid renewable energy site selection problem and benchmarked the performance of MQLib and D-Wave's simulated annealing solver. Key implementation details and computational performance across varying problem sizes were discussed, along with challenges related to scalability and solution quality for both solvers.

The current formulation excludes several practical aspects such as transmission losses, spatial constraints, regulatory restrictions, and fluctuating energy demand. Future work can aim to incorporate these real-world factors into the QUBO framework and evaluate solver performance under more realistic conditions.

Performance tends to improve when solvers are explicitly tuned to the structure and constraints of a specific problem. Certain solvers are likely to be more effective for different classes of real-world problems, especially at larger scales, depending on their ability to balance solution quality, computational time, and resource efficiency. As computational capabilities advance and the perceived quantum advantage improves, it will be increasingly important to identify which types of optimization problems benefit most from classical, quantum, or hybrid solvers (Nannicini, 2019).

References

Abdel-Basset, M., Abdel-Fatah, L., & Sangaiah, A. K. (2018). Metaheuristic Algorithms: A Comprehensive Review. *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*, 185–231. https://doi.org/10.1016/b978-0-12-813314-9.00010-4

Ajagekar, A., & You, F. (2019). Quantum computing for energy systems optimization: Challenges and opportunities. *Energy*, *179*, 76–89. https://doi.org/10.1016/j.energy.2019.04.186

Bernhard Korte, Jens Vygen, & Springerlink (Online Service. (2006). *Combinatorial Optimization : Theory and Algorithms*. Springer Berlin Heidelberg.

Bischl, B., Binder, M., Lang, M., Pielok, T., Richter, J., Coors, S., Thomas, J., Ullmann, T., Becker, M., Boulesteix, A., Deng, D., & Lindauer, M. (2023). Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery*, *13*(2). https://doi.org/10.1002/widm.1484

Britt, K. A., & Humble, T. S. (2017). High-Performance Computing with Quantum Processing Units. *ACM Journal on Emerging Technologies in Computing Systems*, *13*(3), 1–13. https://doi.org/10.1145/3007651

Cheeseman, P., Kanefsky, B., & Riacs, W. (1991). *Where the Really Hard Problems Are*. https://www.dcs.gla.ac.uk/~pat/cpM/papers/cheeseman91where.pdf

Cox, D. R. (2018). Analysis of Binary Data, Second Edition. In *Routledge eBooks*. Informa. https://doi.org/10.1201/9781315137391

Farhi, E., Goldstone, J., & Gutmann, S. (2014, November 14). *A Quantum Approximate Optimization Algorithm*. ArXiv.org. https://doi.org/10.48550/arXiv.1411.4028

Friedman, J. R., Patel, V., Chen, W., Tolpygo, S. K., & Lukens, J. E. (2000). Quantum superposition of distinct macroscopic states. *Nature*, *406*(6791), 43–46. https://doi.org/10.1038/35017505

Glover, F., Kochenberger, G., & Du, Y. (2019, November 4). *A Tutorial on Formulating and Using QUBO Models*. ArXiv.org. https://doi.org/10.48550/arXiv.1811.11538

Glover, F., & Laguna, M. (2002). *Tabu search*. Kluwer Academic.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing - STOC '96*. https://doi.org/10.1145/237814.237866

Hauke, P., Katzgraber, H. G., Lechner, W., Nishimori, H., & Oliver, W. D. (2020). Perspectives of quantum annealing: methods and implementations. *Reports on Progress in Physics*, *83*(5), 054401. https://doi.org/10.1088/1361-6633/ab85b8

Horodecki, R., Horodecki, P., Horodecki, M., & Horodecki, K. (2009). Quantum entanglement. *Reviews of Modern Physics*, *81*(2), 865–942. https://doi.org/10.1103/revmodphys.81.865

Jackson, R. B., Le Quéré, C., Andrew, R. M., Canadell, J. G., Korsbakken, J. I., Liu, Z., Peters, G. P., & Zheng, B. (2018). Global energy growth is outpacing decarbonization. *Environmental Research Letters*, *13*(12), 120401. https://doi.org/10.1088/1748-9326/aaf303

Javadi-Abhari, A., Treinish, M., Krsulich, K., Wood, C. J., Lishman, J., Gacon, J., Martiel, S., Nation, P. D., Bishop, L. S., Cross, A. W., Johnson, B. R., & Gambetta, J. M. (2024). Quantum computing with Qiskit. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2405.08810

Kato, T. (1950). On the Adiabatic Theorem of Quantum Mechanics. *Journal of the Physical Society of Japan*, *5*(6), 435–439. https://doi.org/10.1143/jpsj.5.435

Kochnderfer, M. J., & Wheeler, T. A. (2019). *Algorithms for optimization*. The Mit Press.

Koshka, Y., & Novotny, M. A. (2020). Comparison of D-Wave Quantum Annealing and Classical Simulated Annealing for Local Minima Determination. *IEEE Journal on Selected Areas in Information Theory*, *1*(2), 515–525. https://doi.org/10.1109/jsait.2020.3014192

Lawler, E. L. (1963). The Quadratic Assignment Problem. *Management Science*, *9*(4), 586–599. https://doi.org/10.1287/mnsc.9.4.586

Maradin, D. (2021). ADVANTAGES AND DISADVANTAGES OF RENEWABLE ENERGY SOURCES UTILIZATION. *International Journal of Energy Economics and Policy*, *11*(3), 176–183. https://doi.org/10.32479/ijeep.11027

McClean, J. R., Romero, J., Babbush, R., & Aspuru-Guzik, A. (2016). The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, *18*(2), 023023. https://doi.org/10.1088/1367-2630/18/2/023023

Nannicini, G. (2019). Performance of hybrid quantum-classical variational heuristics for combinatorial optimization. *Physical Review E*, *99*(1). https://doi.org/10.1103/physreve.99.013304

P A M Dirac. (2013). *The principles of quantum mechanics*. Snowball Publishing.

P.J. van Laarhoven, & Aarts, E. H. (2013). *Simulated Annealing: Theory and Applications*. Springer Science & Business Media.

Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 124–134. https://doi.org/10.1109/sfcs.1994.365700

Sivanandam, S. N., Deepa, S. N., & Springerlink (Online Service. (2008). *Introduction to Genetic Algorithms*. Springer Berlin Heidelberg.

Tilly, J., Chen, H., Cao, S., Picozzi, D., Setia, K., Li, Y., Grant, E., Wossnig, L., Rungger, I., Booth, G. H., & Tennyson, J. (2022). The Variational Quantum Eigensolver: A review of methods and best practices. *Physics Reports*, *986*, 1–128. https://doi.org/10.1016/j.physrep.2022.08.003