Quantum-Enhanced Conservation: Optimizing Protected Area Networks with QAOA and Gate-Based Quantum Algorithms

OQI-UC002-DWave Project

November 2025

Abstract

This document presents a comprehensive quantum computing approach to solving the reserve design problem in conservation biology using gate-based quantum algorithms. We demonstrate how the biodiversity conservation planning challenge can be reformulated as a K-SAT problem and solved using QAOA (Quantum Approximate Optimization Algorithm) and Grover-based quantum search on universal quantum computers. Leveraging recent theoretical results showing quartic quantum speedups for planted inference problems, this work bridges sustainable development goals with cutting-edge quantum optimization, providing a pathway from classical constraint programming through SAT encoding to quantum-enhanced decision-making for protecting Earth's ecosystems.

Contents

1 SDG Alignment **2**

2 Short Summary **2**

3 Description of the Problem and Context **2**
3.1 The Global Biodiversity Crisis . . . 2
3.2 Conservation Planning Challenge . . . 3
3.3 The Reserve Design Problem . . . 3

4 Computational Challenge **3**
4.1 Mathematical Formulation . . . 3
4.2 Current Classical Approaches . . . 4
4.3 Computational Bottlenecks . . . 5

5 Potential Impact of Quantum Solution **5**
5.1 Quantum Approach: From Constraint Programming to K-SAT to Quantum Gate Algorithms 5
5.2 Quantum Algorithms and Methods . . . 6
5.3 Projected Benefits Over Classical Approaches . . . 7
5.4 Scalability and Resource Advantages . . . 8
5.5 Implementation Timeline . . . 8
5.6 Quantum Resource Requirements . . . 9
5.7 Available Datasets . . . 10
5.8 Proof of Concept Completion Plan . . . 11

6 Technical Implementation Details **14**
6.1 Current Implementation Status . . . 14
6.2 Encoding Complexity Analysis . . . 14

7 References **15**

A Code Repository Structure **15**

B Installation and Quick Start **16**

C Contact and Collaboration **16**

1 SDG Alignment

Primary SDG:

- **Goal 15: Life on Land** - Protect, restore and promote sustainable use of terrestrial ecosystems, sustainably manage forests, combat desertification, and halt and reverse land degradation and halt biodiversity loss.

Supporting SDGs:

- **Goal 13: Climate Action** - Forests and protected areas serve as critical carbon sinks and buffer zones against climate change impacts.
- **Goal 14: Life Below Water** - Coastal and marine-terrestrial interface protected areas support aquatic biodiversity.
- **Goal 11: Sustainable Cities** - Strategic reserve design around human settlements balances conservation with sustainable urban development.

2 Short Summary

This project develops a quantum computing framework for optimizing biodiversity conservation decisions. Specifically, we address the *reserve design problem*: selecting an optimal network of protected areas (planning units) that maximizes species representation while minimizing costs and ensuring spatial connectivity. We demonstrate a complete pipeline from mathematical formulation through K-SAT encoding to quantum annealing on D-Wave systems, with classical SAT solvers providing baseline performance benchmarks.

Key Objectives:

1. Formulate the reserve design problem as a constrained optimization challenge
2. Develop lossless conversion to Boolean satisfiability (K-SAT)
3. Implement classical SAT solver baseline (PySAT, Z3)
4. Design quantum annealing formulation for D-Wave platforms
5. Validate quantum approaches against real-world conservation scenarios
6. Demonstrate computational advantages for large-scale planning problems

3 Description of the Problem and Context

3.1 The Global Biodiversity Crisis

What is the societal challenge and why is it critical?

The world is facing an unprecedented biodiversity crisis. According to the United Nations Environment Programme:

- **1 million species** are threatened with extinction, many within decades - more than ever before in human history
- **100 million hectares** of forest lost between 2000-2020 (31% of Earth's land surface impacted)
- **3.2 billion people** affected by land degradation globally
- **90% of deforestation** driven by agricultural expansion
- **38% of tree species** face extinction due to habitat loss, climate change, and overexploitation

This rapid loss of biodiversity undermines ecosystem services worth over **half of global GDP**, threatens food security, accelerates climate change, and increases pandemic risk through wildlife-human interface disruption. Furthermore, a 2025 horizon scan of emerging biological conservation issues highlights novel challenges this project could help address, including [5]:

- **Accelerated Antarctic Changes:** Unanticipated, rapid changes such as the destabilization of sea ice and the accelerated melting of the Thwaites glacier, which have profound impacts on global biodiversity.
- **Novel Technological and Pollutant Impacts:** Including risks from near-surface ozone, the extraction of rare earth elements from macroalgae, and contamination from 'forever chemicals' (PFAS).
- **Compounded System Stresses:** The combined effects of deteriorating water quality and quantity on both human and natural systems.

3.2 Conservation Planning Challenge

Who and where is the affected population?

Conservation planners worldwide face the critical challenge of designing protected area networks with:

- **Geographic Scope:** From local (individual national parks) to continental (trans-boundary corridors)
- **Stakeholders:** Government agencies, NGOs (WWF, Conservation International), indigenous communities, private landowners
- **Resources:** Limited budgets (e.g., $10M-$1B for major conservation programs)
- **Biodiversity Hotspots:** Amazon rainforest, Congo Basin, Coral Triangle, Madagascar, etc.
- **Constraints:** Political boundaries, land ownership, economic development pressures, social equity

Current Impact: The Kunming-Montreal Global Biodiversity Framework (2022) commits nations to protecting 30% of Earth's land and oceans by 2030 ("30x30" target), requiring systematic planning for hundreds of thousands of new protected areas.

3.3 The Reserve Design Problem

The scientific formulation asks: *Given a landscape divided into planning units, each containing different species at different costs, which sites should be protected to ensure all target species are adequately represented while minimizing total cost and maintaining spatial connectivity?*

This is formally known as the **Minimum Set Cover Problem with Connectivity Constraints**, proven to be NP-hard.

4 Computational Challenge

4.1 Mathematical Formulation

How is the societal challenge connected to a computational problem? What is the mathematical description?

The reserve design problem can be formulated as a Mixed-Integer Linear Program (MILP):

Given:

- $S = \{1, 2, \ldots, n\}$: Set of n planning units (sites)
- $P = \{1, 2, \ldots, m\}$: Set of m species (features)
- c_i: Cost of protecting site i (land acquisition, management)
- $r_{ij} \in \{0, 1\}$: Presence of species j in site i
- t_j: Target representation for species j (e.g., ≥ 3 sites)

- B: Total budget constraint
- $G = (S, E)$: Adjacency graph where $(i, j) \in E$ if sites are neighbors

Decision Variables:

- $x_i \in \{0, 1\}$: Binary variable, 1 if site i is selected
- $y_j \in \{0, 1\}$: Binary variable, 1 if species j target is met
- $z_{ij} \in \{0, 1\}$: Connectivity variable for edge (i, j)

Objective:

$$\min \sum_{i=1}^{n} c_i x_i \tag{1}$$

Subject to:

$$\sum_{i=1}^{n} r_{ij} x_i \geq t_j \quad \forall j \in P \tag{Species representation}$$
$$\sum_{i=1}^{n} c_i x_i \leq B \tag{Budget constraint}$$
$$z_{ij} = x_i \wedge x_j \quad \forall (i, j) \in E \tag{Connectivity}$$
$$\text{Selected sites form connected subgraph} \tag{Compactness}$$

4.2 Current Classical Approaches

How is the challenge currently approached with existing technologies?

Classical Solution Methods:

1. **Greedy Heuristics (1980s-present):**
   - Tools: Marxan, Zonation, ConsNet
   - Method: Iteratively add sites with best cost-effectiveness ratio
   - Speed: Fast (seconds to minutes)
   - Quality: 10-30% suboptimal, no optimality guarantee
2. **Integer Linear Programming (2000s-present):**
   - Solvers: Gurobi, CPLEX, COIN-OR
   - Method: Branch-and-bound with cutting planes
   - Speed: Minutes to hours for $n < 1000$
   - Quality: Optimal (when solvable within time limits)
3. **Constraint Programming (2010s-present):**
   - Tools: Choco, Gecode, Google OR-Tools
   - Method: Propagation + search with global constraints
   - Speed: Comparable to ILP
   - Quality: Optimal (with complete search)
4. **Metaheuristics:**
   - Simulated annealing, genetic algorithms, tabu search
   - Speed: Fast but unpredictable
   - Quality: No optimality guarantee

4.3 Computational Bottlenecks

What is difficult to model and what are current limitations?

Scalability Crisis:

- **Problem Size:** Real-world instances have $n = 10^4$-10^6 planning units
- **Complexity:** $O(2^n)$ solution space grows exponentially
- **Time Limits:** ILP solvers timeout beyond $n \approx 5000$ sites with connectivity
- **Memory:** Large-scale problems exceed RAM (¿100GB for $n = 10^5$)

Specific Challenges:

1. **Connectivity Constraints:** Adding spatial connectivity increases complexity from Set Cover (NP-complete) to Steiner Tree (NP-hard with poor approximation ratios)
2. **Multiple Objectives:** Real planning requires balancing:
    - Species representation vs. cost
    - Compactness vs. connectivity
    - Present species vs. future climate projections
    - Conservation value vs. social/political feasibility
3. **Uncertainty:** Species occurrence data is incomplete, costs are estimates, climate change impacts are probabilistic
4. **Dynamic Planning:** Need to adapt plans as new data arrives or conditions change

Current Classical Limitations:

- Large instances ($n > 10,000$) require hours to days even with heuristics
- No guarantee of finding optimal solution within practical time
- Cannot efficiently handle very large problem instances for continental-scale planning
- Branch-and-bound explores exponentially growing search trees

5 Potential Impact of Quantum Solution

5.1 Quantum Approach: From Constraint Programming to K-SAT to Quantum Gate Algorithms

From the classical computing approach, which part could be tackled with quantum computing?

Our approach leverages gate-based quantum algorithms specifically designed for SAT solving:

Transformation Pipeline:

1. **Constraint Programming Formulation:** Express reserve design with global constraints (cardinality, connectivity, budget)
2. **K-SAT Encoding:** Convert constraints to Conjunctive Normal Form (CNF):
    - Species representation: AtLeast-K constraints using sequential counter encoding
    - Budget: Pseudo-Boolean constraints using binary adder circuits
    - Connectivity: AND/OR gates via Tseitin transformation
    - Objective: Binary search on cost bounds

3. **Quantum SAT Solving:** Apply quantum algorithms to the CNF formula:
   - QAOA (Quantum Approximate Optimization Algorithm) for k-SAT instances
   - Grover's algorithm with amplitude amplification for solution search
   - Quantum walk-based algorithms for structured SAT instances
   - Hybrid classical-quantum approaches

5.2 Quantum Algorithms and Methods

What type of quantum algorithms and methods would be used and why?

Primary Approach: QAOA for k-SAT and Quantum SAT Algorithms

1. **QAOA (Quantum Approximate Optimization Algorithm):**
   - Gate-based variational quantum algorithm
   - Phase separator: Apply problem Hamiltonian $U_P(\gamma) = e^{-i\gamma H_P}$ where H_P encodes SAT clauses
   - Mixer: Apply $U_M(\beta) = e^{-i\beta H_M}$ where $H_M = \sum_i X_i$
   - Parameterized circuit: $|\psi(\gamma, \beta)\rangle = U_M(\beta_p)U_P(\gamma_p) \cdots U_M(\beta_1)U_P(\gamma_1)|+\rangle^{\otimes n}$
   - Classical optimization of parameters (γ, β) to maximize solution probability
   - Depth p controls approximation quality vs. circuit complexity tradeoff
   - Recent work on applying QAOA to hard random k-SAT instances shows that it can be competitive with leading classical heuristics. For example, for random 8-SAT at the satisfiability threshold, QAOA with approximately 14 ansatz layers is estimated to match the performance of the highly optimized classical WalkSATlm solver, and is predicted to outperform it for a larger number of layers [3]. This provides theoretical and numerical evidence for the potential of a quantum advantage on this problem class.

2. **Grover-Based SAT Solving:**
   - Quantum search for satisfying assignments
   - Quadratic speedup: $O(\sqrt{2^n})$ vs. classical $O(2^n)$ for unstructured search
   - Oracle marks satisfying assignments: $O : |x\rangle \to (-1)^{f(x)}|x\rangle$
   - Amplitude amplification increases probability of finding solutions
   - Multiple solutions: Modified Grover for counting and sampling

3. **Quantum Speedups for Planted k-SAT (Recent Results):**
   - **Nearly quartic speedups demonstrated** for planted inference problems [2]
   - Planted k-SAT: instances with known satisfying assignments embedded
   - Conservation planning naturally has "planted" structure: an optimal or near-optimal conservation plan can be viewed as a "planted solution" hidden within the vast combinatorial search space of all possible land parcel combinations, making it an ideal candidate for these advanced quantum algorithms
   - Speedup mechanism: Quantum algorithms exploit problem structure via phase estimation
   - Applies to problems with hidden structure (biodiversity patterns, spatial clustering)

4. **Hybrid Quantum-Classical Approaches:**
   - **QAOA + classical optimizer:** Variational parameter optimization (COBYLA, BFGS)
   - **Recursive QAOA:** Solve smaller sub-problems, fix variables, iterate
   - **Quantum walks:** Explore solution space via continuous-time quantum walks on constraint graphs

- **VQE-style approaches:** Adapt Variational Quantum Eigensolver techniques for SAT

Why Gate-Based Quantum Algorithms for k-SAT?

- **Proven Speedups:** Grover's algorithm provides provable quadratic speedup for unstructured search
- **Structural Exploitation:** QAOA and quantum walks exploit problem structure (planted solutions, community structure)
- **Recent Advances:** Quartic speedups demonstrated for planted k-SAT instances [2]
- **Natural Encoding:** K-SAT maps directly to quantum circuits via Pauli operations
- **Universal Platforms:** Compatible with IBM Quantum, Google Sycamore, IonQ, etc.
- **Error Mitigation:** Active research on noise-resilient QAOA implementations
- **Ecological Structure:** Conservation problems have inherent structure (spatial autocorrelation, species co-occurrence) that quantum algorithms can exploit

5.3 Projected Benefits Over Classical Approaches

What are the projected expected benefits (speedup, accuracy, etc.)?

Potential Quantum Advantages:

1. **Computational Speedup:**
   - **Hypothesis:** Quadratic to polynomial speedup for structured problems
   - **Target:** Solve $n = 10,000$ site problems in minutes vs. hours
   - **Mechanism:** Quantum tunneling through energy barriers vs. thermal activation
   - **Evidence:** Recent results on planted k-SAT show quartic speedups (Schmidhuber et al., 2024)
2. **Solution Quality:**
   - Access to broader solution landscape through quantum superposition
   - Multiple near-optimal solutions (important for stakeholder decision-making)
   - Better escape from local minima compared to simulated annealing
3. **Scalability:**
   - Hybrid quantum-classical approaches handle $n > 100,000$ via decomposition
   - Parallel processing: 1000s of quantum samples in seconds
   - Reduced memory requirements vs. classical branch-and-bound
4. **Practical Conservation Impact:**
   - **Faster planning cycles:** Respond to changing conditions (fires, development)
   - **Larger scales:** Continental-level planning becomes feasible
   - **Scenario analysis:** Explore many "what-if" scenarios rapidly
   - **Adaptive management:** Real-time optimization as monitoring data arrives

Quantitative Performance Targets (Based on Recent Results):

- **Planted k-SAT instances:** Nearly quartic speedup demonstrated [2]
- **Grover-based search:** Quadratic ($\sqrt{2^n}$) speedup for unstructured instances

- **QAOA optimization:** Problem-dependent, typically polynomial to quadratic advantage for structured problems
- Small instances ($n = 50$-100 qubits): Demonstration on NISQ devices (IBM, IonQ)
- Medium instances ($n = 100$-500): Hybrid quantum-classical decomposition
- Large instances ($n > 500$): Future fault-tolerant quantum computers required

5.4 Scalability and Resource Advantages

Space Efficiency:

- **Reduced Memory Requirements:** The quantum algorithm for planted inference problems requires only $O(\log n)$ qubits, an exponential space saving compared to the classical Kikuchi method which requires storing a matrix of size polynomial in n^l [2]
- This memory advantage is particularly crucial for large-scale conservation planning where classical methods may become memory-bound before computation-bound

5.5 Implementation Timeline

What would be the timeline for implementation (NISQ vs. FTQC)?

Phase 1: NISQ Era (Current - 2027) - CURRENTLY IMPLEMENTED

1. **Proof of Concept (Completed):**
   - ✓ Classical K-SAT encoding implemented (700+ lines Python)
   - ✓ Integration with 6 SAT solvers (Glucose, MiniSat, Z3, etc.)
   - ✓ Mathematical correctness proofs (LaTeX document, 30+ pages)
   - ✓ Validation on random and grid instances (up to $n = 100$ sites)
   - ✓ Performance benchmarking vs. constraint programming
2. **QAOA Implementation (Next 6-12 months):**
   - Implement k-SAT to QAOA circuit compilation
   - Develop parameterized quantum circuits for SAT clauses
   - Test on IBM Quantum / Amazon Braket / Azure Quantum platforms
   - Implement classical parameter optimization (COBYLA, BFGS)
   - Compare QAOA vs. classical SAT solver performance
   - Explore hybrid quantum-classical decomposition strategies
3. **Real-World Validation (12-24 months):**
   - Partner with conservation organizations (e.g., The Nature Conservancy)
   - Test on actual reserve design datasets (Costa Rica, Madagascar, etc.)
   - Benchmark against Marxan, Zonation operational tools
   - Publish peer-reviewed results

Phase 2: Near-Term Quantum Advantage (2027-2030)

- 10,000+ qubit systems with improved coherence
- Error mitigation techniques mature
- Routine quantum advantage demonstrations on specific problem classes

- Commercial deployment for conservation agencies

Phase 3: Fault-Tolerant Era (2030+)

- Logical qubits with full error correction
- Gate-model quantum algorithms (QAOA, Grover) surpass annealing
- Quantum optimization as standard tool in computational ecology

5.6 Quantum Resource Requirements

What quantum resources would be required for a proof of concept?

Hardware Requirements:

1. **Minimum (NISQ Proof of Concept):**
    - **Platform:** IBM Quantum (127+ qubits, e.g., ibm_sherbrooke), IonQ (32+ qubits), or Rigetti (80+ qubits)
    - **Access:** Free tier (IBM Quantum, Amazon Braket free credits) or $0.30-3/shot commercial
    - **Problem Size:** $n = 20$-50 variables (sites), $m = 5$-10 species
    - **Qubits Required:** n qubits for SAT variables + ancillas for clause evaluation ($\sim$2-3n total)
    - **Circuit Depth:** $p = 1$-5 QAOA layers ($\sim$10-50 gates deep with current connectivity)
    - **Shots:** 1000-10,000 measurements per parameter setting
    - **Classical Optimization:** 50-200 iterations of parameter optimization
    - **Wall-Clock Time:** 10 minutes - 2 hours (including queue time)
2. **Intermediate (Near-Term Advantage Demonstration):**
    - **Problem Size:** $n = 50$-100 variables with 200-500 clauses
    - **Qubits:** 100-200 logical qubits (physical qubits with error correction)
    - **Approach:** Recursive QAOA with problem decomposition
    - **Circuit Depth:** $p = 5$-10 layers with hardware-efficient gates
    - **Error Mitigation:** Zero-noise extrapolation, probabilistic error cancellation
    - **Total Time:** 1-5 hours including multiple quantum-classical iterations
3. **Advanced (Fault-Tolerant Era):**
    - **Problem Size:** $n = 1000+$ variables (full conservation planning scales)
    - **Qubits:** 1000-10,000 logical qubits with T-gate count $\sim 10^6$-10^9
    - **Approach:** Grover's algorithm or amplitude amplification with deep circuits
    - **Circuit Depth:** $O(\sqrt{2^n})$ for Grover, polynomial for QAOA
    - **Total Time:** Minutes to hours depending on quantum advantage magnitude

Software Requirements:

- **Quantum Frameworks:** Qiskit (IBM), Cirq (Google), PennyLane, or Amazon Braket SDK
- **Classical Baseline:** PySAT, Z3 for validation
- **Encoding Tools:** Our custom K-SAT encoder (already implemented)
- **QAOA Implementation:** Parameterized quantum circuits for SAT
- **Classical Optimization:** SciPy (COBYLA, BFGS), NumPy for parameter tuning
- **Visualization:** NetworkX, Matplotlib for solution analysis
- **GIS Integration:** QGIS, GeoPandas for real-world data

5.7 Available Datasets

What datasets are available to this project?

Synthetic Datasets (Currently Used):

1. **Random Instances:**
   - Parameterized by n, m, budget fraction, species distribution
   - Costs: uniform or power-law distribution
   - Species presence: random with density parameter
   - Adjacency: random geometric graphs or lattices
2. **Grid Instances:**
   - 2D grid topology (e.g., 10×10, 20×20)
   - Clustered species distributions (mimics real biogeography)
   - Useful for visualization and algorithm development

Real-World Datasets (Available for Integration):

1. **Protected Area Databases:**
   - World Database on Protected Areas (WDPA)
   - National GAP Analysis datasets (US)
   - European Environment Agency conservation data
2. **Species Occurrence Data:**
   - GBIF (Global Biodiversity Information Facility): 2+ billion records
   - eBird: 1+ billion bird observations
   - IUCN Red List: 150,000+ assessed species
   - Regional biodiversity atlases
3. **Spatial Data:**
   - Land cover: Copernicus Global Land Service, MODIS
   - Elevation: SRTM, ASTER GDEM
   - Administrative boundaries: GADM
   - Land cost proxies: Property value maps, agricultural land prices
4. **Benchmark Instances:**
   - Conservation planning literature datasets (Marxan case studies)
   - Systematic conservation planning benchmarks
   - Published reserve design problems with known solutions

Data Preprocessing Pipeline:

1. Rasterize landscape to planning units (typical: 1-100 km^2 cells)
2. Overlay species range maps and occurrence points
3. Estimate protection costs (land value + management)
4. Build adjacency graph from spatial topology
5. Set representation targets (e.g., 10-30% of species range)
6. Calculate budget as fraction of total landscape cost

5.8 Proof of Concept Completion Plan

Describe how you envisage completing a proof of concept for this project.

Stage 1: Classical Baseline (COMPLETED)

✓ **Mathematical Formulation:**

- Complete MILP formulation with connectivity constraints
- LaTeX documentation with correctness proofs (30+ pages)
- Complexity analysis and encoding size bounds

✓ **K-SAT Encoding Implementation:**

- Python modules: ReserveDesignInstance, SATEncoder, SATSolver
- Multiple encoding strategies (sequential counter, binary adder, totalizer)
- Integration with 6 SAT solvers via PySAT
- Comprehensive test suite (6 unit tests)

✓ **Validation:**

- Synthetic instances: $n = 10$-100 sites, $m = 3$-20 species
- Grid instances: 4×4, 5×5 with visualization
- Solution verification: feasibility, optimality (via binary search)
- Performance benchmarking: encoding time, solving time, memory

Stage 2: QAOA Circuit Design (Next 3 months)

1. **SAT-to-QAOA Mapping:**
   - Implement clause-to-Hamiltonian conversion
   - Design phase separator circuits: $U_P(\gamma) = \prod_c e^{-i\gamma_c H_c}$ where H_c is clause Hamiltonian
   - Implement mixer operations: $U_M(\beta) = \prod_i e^{-i\beta X_i}$
   - Optimize gate count and circuit depth for NISQ devices
2. **Classical Parameter Optimization:**
   - Implement variational optimization loop
   - Test multiple optimizers: COBYLA, BFGS, ADAM, SPSA
   - Develop warm-start strategies from classical solutions
   - Implement parameter transfer across problem instances
3. **Classical QAOA Simulation:**
   - Simulate QAOA circuits classically (up to $n \approx 20$ qubits)
   - Verify correctness of quantum circuit compilation
   - Benchmark against classical SAT solvers
   - Identify optimal QAOA depth p for problem class

Stage 3: Quantum Hardware Execution (Months 4-6)

1. **Quantum Platform Setup:**
   - Install Qiskit / Cirq / PennyLane
   - Obtain cloud quantum access (IBM Quantum, Amazon Braket, Azure Quantum)

- Configure API connections and authentication

2. **QAOA Deployment:**
   - Transpile circuits to hardware-native gates
   - Implement qubit mapping and routing for device topology
   - Apply error mitigation techniques (measurement error mitigation, zero-noise extrapolation)
   - Handle device-specific constraints (gate fidelities, connectivity limits)
3. **Quantum SAT Experiments:**
   - Solve small instances ($n = 10$-30) directly on quantum hardware
   - Vary QAOA depth p: 1-10 layers
   - Collect 1000-10000 shots per parameter setting
   - Analyze solution quality distribution and success probabilities
   - Compare with Grover-based implementations
4. **Hybrid Algorithm Development:**
   - Recursive QAOA: solve sub-problems, fix variables, iterate
   - Quantum-classical feedback loops
   - Problem decomposition strategies
   - Warm-starting from classical heuristics

Stage 4: Performance Evaluation (Months 7-9)

1. **Comprehensive Benchmarking:**
   - Problem sizes: $n \in \{20, 50, 100, 200, 500\}$
   - Metrics: Solution quality, time-to-solution, success rate
   - Baselines: Classical SAT, ILP, greedy heuristics
   - Statistical analysis: 50-100 runs per configuration
2. **Quantum Advantage Analysis:**
   - Identify problem regimes where quantum excels
   - Quantify speedup factors
   - Characterize scaling behavior
   - Error analysis: embedding errors, thermal noise, control errors
3. **Conservation-Relevant Metrics:**
   - Species representation quality
   - Spatial compactness of solutions
   - Cost-effectiveness ratio
   - Diversity of near-optimal solutions

Stage 5: Real-World Validation (Months 10-12)

1. **Case Study Selection:**
   - Small scale: Local watershed ($n \approx 100$)
   - Medium scale: County/province ($n \approx 1000$)

- Data sources: GBIF + WDPA + regional land value data

2. **Stakeholder Engagement:**
   - Consult conservation practitioners on practical constraints
   - Incorporate domain knowledge (e.g., existing protected areas, political boundaries)
   - Validate solutions with ecological expertise

3. **Comparison with Operational Tools:**
   - Marxan (most widely used globally)
   - Zonation (European standard)
   - Custom ILP/CP solutions
   - Document advantages and limitations

Stage 6: Dissemination (Ongoing)

1. **Scientific Publication:**
   - Target journals: Nature/Science (if strong quantum advantage), or specialized (Quantum, Conservation Biology)
   - Paper structure: Introduction, Methods (K-SAT + QUBO), Results, Discussion
   - Supplementary materials: Code repository, datasets, full proofs

2. **Open-Source Release:**
   - GitHub repository with MIT/Apache license
   - Documentation: API reference, tutorials, examples
   - Docker containers for reproducibility
   - Jupyter notebooks for educational use

3. **Community Engagement:**
   - Workshops at conservation conferences (SCB, ICCB)
   - Quantum computing conferences (APS March Meeting, QIP)
   - Webinars for practitioners
   - Policy briefs for UN, IUCN

Success Criteria:

1. **Technical:**
   - QAOA solutions match or exceed classical quality on planted k-SAT instances
   - Demonstrated speedup consistent with theoretical predictions (quadratic for Grover, quartic for planted instances)
   - Successful execution on $n \geq 50$ qubits with NISQ hardware
   - Error-mitigated results show quantum advantage signal

2. **Scientific:**
   - Publication in peer-reviewed quantum computing or optimization journal
   - Experimental demonstration of quantum speedup for conservation-relevant problem class
   - Contribution to understanding of QAOA performance on real-world structured SAT
   - Presentation at quantum computing or conservation conferences

3. **Practical:**
   - Interest from at least one conservation organization for pilot studies
   - Open-source implementation used by quantum computing community
   - Integration pathway defined for future fault-tolerant systems

6 Technical Implementation Details

6.1 Current Implementation Status

Our proof-of-concept implementation includes:

1. **Problem Representation** (`reserve_design_instance.py`):
    - Object-oriented design for problem instances
    - Random and grid instance generators
    - Solution validation and feasibility checking
    - Connectivity verification algorithms
2. **SAT Encoding** (`sat_encoder.py`):
    - Site selection variables
    - Species representation (AtLeast-K cardinality constraints)
    - Budget constraints (pseudo-Boolean encoding)
    - Connectivity (AND gate encoding)
    - Multiple encoding strategies with complexity analysis
3. **Classical Solving** (`sat_solver.py`):
    - Unified interface to 6 SAT solvers
    - Feasibility checking
    - Optimization via binary search
    - Detailed performance statistics
4. **Examples and Tests** (`examples.py`, `test_ksat.py`):
    - 5 comprehensive usage examples
    - 6 unit test cases covering all functionality
    - Validation of encoding correctness

6.2 Encoding Complexity Analysis

CNF Size:

- **Variables:** $O(n + m + nm + nB)$ where terms represent site, species, presence, and budget encoding variables
- **Clauses:** $O(nm + nB + |E|)$ for sequential counter encoding
- **Alternative:** $O(nm \log B)$ with binary encoding for large budgets

QUBO Size (Projected):

- **Logical Qubits:** Similar to CNF variables, $O(n + nm)$ for most formulations
- **Couplings:** $O(nm + nB)$ quadratic terms
- **Physical Qubits:** $5 - 10\times$ logical qubits for D-Wave embedding

7 References

References

[1] Justiniano-Albarracin, X., Birnbaum, P., & Lorca, X. (2018). *Unifying reserve design strategies with graph theory and constraint programming*. International Conference on Principles and Practice of Constraint Programming.

[2] Schmidhuber, A., O'Donnell, R., Kothari, R., & Babbush, R. (2024). *Quartic quantum speedups for planted inference*. arXiv:2406.19378.

[3] Boulebnane, S., & Montanaro, A. (2022). *Solving boolean satisfiability problems with the quantum approximate optimization algorithm*. arXiv:2208.06909.

[4] Babbush, R., King, R., Boixo, S., Huggins, W., Khattar, T., Low, G. H., McClean, J. R., O'Brien, T., & Rubin, N. C. (2024). *The Grand Challenge of Quantum Applications*. arXiv:2511.09124.

[5] Sutherland, W. J., Brotherton, P. N. M., Butterworth, H. M., Clarke, S. J., Davies, T. E., Doar, N., ... & Thornton, A. (2025). *A horizon scan of biological conservation issues for 2025*. Trends in Ecology & Evolution, 40(1), 80-89.

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

[7] Grover, L. K. (1996). *A fast quantum mechanical algorithm for database search*. Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, 212-219.

[8] Qiskit Development Team (2024). *Qiskit: An Open-source Framework for Quantum Computing*. https://qiskit.org/

[9] IBM Quantum (2024). *IBM Quantum Computing Platform*. https://quantum.ibm.com/

[10] United Nations (2022). *Kunming-Montreal Global Biodiversity Framework*. Convention on Biological Diversity.

[11] United Nations Sustainable Development Goals. *Goal 15: Life on Land*. https://sdgs.un.org/goals/goal15

[12] Ignatiev, A., Morgado, A., & Marques-Silva, J. (2018). *PySAT: A Python Toolkit for Prototyping with SAT Oracles*. International Conference on Theory and Applications of Satisfiability Testing.

[13] Ball, I. R., Possingham, H. P., & Watts, M. (2009). *Marxan and relatives: Software for spatial conservation prioritisation*. Spatial conservation prioritisation: Quantitative methods and computational tools, 185-195.

[14] Biere, A., Heule, M., & van Maaren, H. (2009). *Handbook of Satisfiability*. IOS Press.

[15] Sinz, C. (2005). *Towards an optimal CNF encoding of boolean cardinality constraints*. International Conference on Principles and Practice of Constraint Programming.

A Code Repository Structure

```
KSAT/
|-- reserve_design_instance.py    # Problem representation (350 lines)
|-- sat_encoder.py                # CNF encoding (450 lines)
|-- sat_solver.py                 # SAT solving (350 lines)
|-- examples.py                   # Usage examples (350 lines)
|-- test_ksat.py                  # Unit tests (250 lines)
```

```
|-- visualization.py              # Plotting tools (250 lines)
|-- requirements.txt              # Python dependencies
+-- Docs/
    |-- README.md                 # User guide
    |-- QUICKSTART.md             # Quick start
    |-- SUMMARY.md                # Implementation overview
    |-- INDEX.md                  # File navigation
    +-- Latex/
        |-- reserve_design_ksat_conversion.tex    # Theory (30 pages)
        +-- quantum_reserve_design_proposal.tex   # This document
```

B Installation and Quick Start

```
# Install dependencies
pip install numpy matplotlib python-sat z3-solver

# Run tests
python test_ksat.py

# Run examples
python examples.py

# For quantum integration (future)
pip install dwave-ocean-sdk
```

C Contact and Collaboration

This project is open for collaboration with:

- Conservation biologists and practitioners
- Quantum computing researchers
- Applied optimization specialists
- GIS and spatial analysis experts
- Policy makers and NGOs

Project Repository: https://github.com/oss-esso/OQI-UC002-DWave
License: MIT (open source)