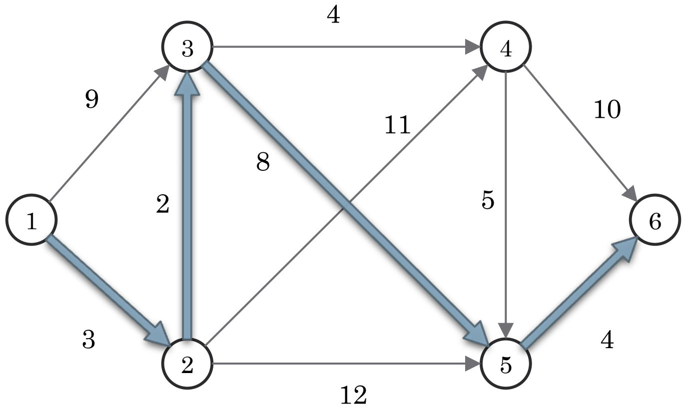
Arc Interdiction Vehicle Routing Problem using Quantum Annealing

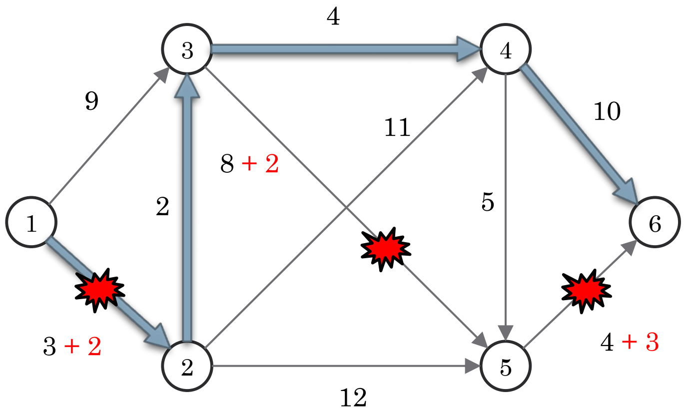
Dheeraj Peddireddy, Gurcan Comert, Mashrur Chowdhury, Vaneet Aggarwal

Introduction

- Vehicle Routing Problem: NP-Hard optimization problem for finding efficient routes for a fleet of vehicles
- Arc Interdiction introduces an adversary to the problem which tries to disrupt routes to maximize the minimum cost of VRP



Before Interdiction



After Interdiction

Introduction

- Traditional VRP assumes a single decision maker optimizing the route
- In reality, adversaries may disrupt critical shipments, making the classical VRP model inadequate
- Many high security applications like military logistics, law enforcement routes etc. may require a bi-level optimization approach to model defender and attacker

Problem Definition

- x_{ij} : $Binary\ variable\ indicating\ if\ arc\ (i,j) is\ interdicted$
- y_{ijk} : $Binary\ variable\ indicating\ if\ arc\ (i,j) is\ traversed\ by\ vehicle\ k \in K$
- c_{ij} : $Cost\ of\ traversing\ arc\ (i,j)$
- d_{ij}: $Penalty\ imposed\ on\ interdicted\ arc\ (i,j)$
- B : $Total\ interdiction\ budget$

Defender

Defender seeks to minimize the routing cost on the disrupted graph:

$$V(x) = \min_{y \in Y} \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} (c_{ij} + d_{ij} x_{ij}) y_{ijk}$$

subject to

$$\begin{aligned}
\sum_{k \in K} \sum_{i \in V} y_{ijk} &= 1 \quad \forall j \in V \setminus \{0\} \\
\sum_{i \in V} y_{ijk} &= \sum_{i \in V} y_{jik} \quad \forall j \in V \setminus \{0\}, k \in K \\
\sum_{j \in V} y_{0jk} &= 1 \quad \forall k \in K \\
\sum_{i \in V} y_{i0k} &= 1 \quad \forall k \in K \\
x_{ij} &\in \{0, 1\} \quad \forall i, j \in V \\
y_{ij} &\in \{0, 1\} \quad \forall i, j \in V
\end{aligned}$$

- x_{ij} : *Binary variable indicating if arc* (i, j) *is interdicted*
- y_{ijk} : *Binary variable indicating if arc* (i, j) *is traversed by vehicle* $k \in K$
- c_{ij} : *Cost of traversing arc* (i, j)
- d_{ij}: *Penalty imposed on interdicted arc* (i, j)
- B : *Total interdiction budget*

Attacker

Interdictor selects arcs to increase cost under a budget constraint:

$$\max_{x \in X} V(x)$$

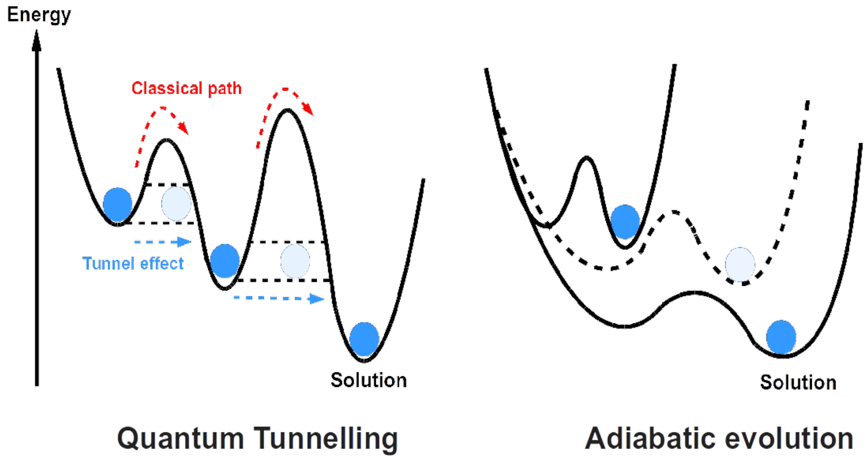
subject to

$$\sum_{(i,j) \in A} x_{ij} \leq B$$

- x_{ij} : $Binary\ variable\ indicating\ if\ arc\ (i,j) is\ interdicted$
- y_{ijk} : $Binary\ variable\ indicating\ if\ arc\ (i,j) is\ traversed\ by\ vehicle\ k \in K$
- c_{ij} : $Cost\ of\ traversing\ arc\ (i,j)$
- d_{ij}: $Penalty\ imposed\ on\ interdicted\ arc\ (i,j)$
- B : $Total\ interdiction\ budget$

Quantum Annealing

- Optimization technique that finds the lowest energy state (optimal solution) of a problem
- Encodes combinatorial optimization problem into an energy landscape known as the Hamiltonian
- Uses quantum tunnelling to explore solutions, avoiding local minima
- Gradually reduces fluctuations to settle into the lowest energy state which corresponds to the optimal solution



Quantum Annealing

- Classical solvers struggle with large scale NP-hard problems
- Quantum Annealing explores solution space in parallel and finds good solutions faster for certain problem instances
- QA can natively solve problems in Ising Model or Quadratic Unconstrained binary optimization (QUBO) format:

$$QUBO : \min \sum_{i,j} Q_{ij} x_i x_j + \sum_i Q_i x_i$$

where $x_i \in \{0, 1\}$

Arc Interdiction – QUBO

- We reformulate the binary optimization problem into a QUBO to solve the problem using Quantum Annealers
- First, we transform bi-level optimization to single level IPP using the dual of the lower-level problem

$$\max_{x,\pi,\lambda,\mu,\nu} \sum_{j \in V \setminus \{0\}} \pi_j + \sum_{k \in K} (\mu_k + \nu_k) \tag{10}$$

subject to

$$\sum_{(i,j) \in A} x_{ij} \leq B \tag{11}$$

$$\pi_j + \lambda_{jk} - \lambda_{ik} + \delta_{i0}\mu_k + \delta_{j0}\nu_k \leq c_{ij} + d_{ij}x_{ij} \quad \forall i, j \in V; k \in K \tag{12}$$

$$\delta_{i0} = \begin{cases} 1, & \text{if } i = 0 \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

$$y_{ij} \in \{0, 1\} \quad \forall i, j \in V \tag{14}$$

Arc Interdiction - QUBO

- Remove constraints by introducing penalty terms.

$$
\begin{aligned}
\max_{x,\pi,\lambda,\mu,\nu,r,s} \sum_{j \in V \setminus \{0\}} \pi_j + \sum_{k \in K} (\mu_k + \nu_k) \\
+P_1 \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij} + d_{ij} x_{ij} \\
-(r_{ijk} + \pi_j + \lambda_{jk} - \lambda_{ik} + \delta_{i0}\mu_k + \delta j 0 \nu_k))^2 \\
+P_2 \sum_{(i,j) \in A} (B - (s_{ij} + x_{ij}))^2
\end{aligned} \tag{15}
$$

subject to

$$\pi_j, \lambda_{jk}, \mu_k, \nu_k, r_{ijk} \in [0, max_{(i,j) \in A}\{c_{ij} + d_{ij}\}]; \tag{16}$$

$$s_{ij} \in [0, B] \tag{17}$$

- Additionally, we use binary encoding to transform all integer variables into binary variables

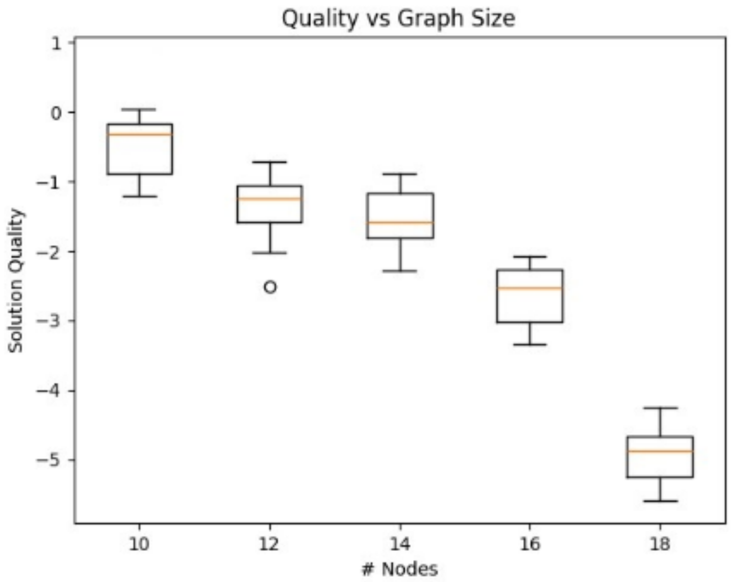
Experiments

- Randomly generated problem instances with varying number of nodes and costs
- Solution Quality (Closer to zero indicates optimal):

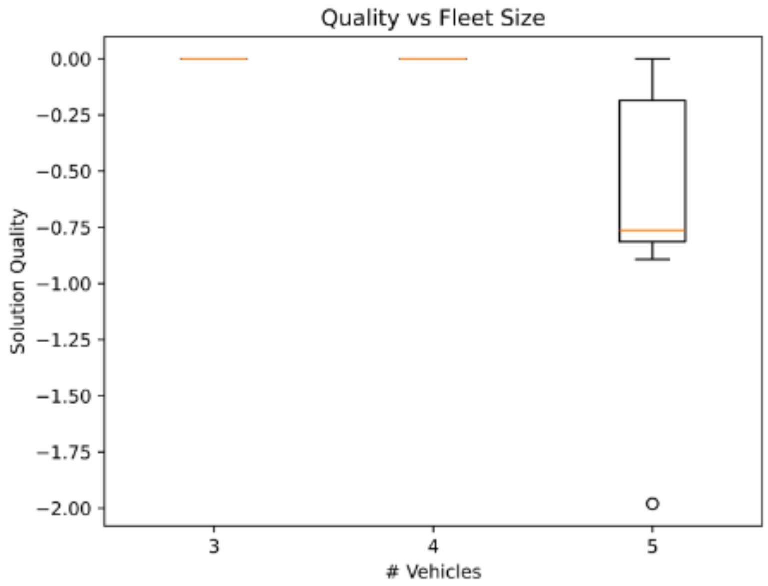
$$Q = 100\times\frac{f_{qubo} - f_{ub}}{f_{ub}}$$

- The upper bound was computed by solving the bi-level optimization problem using Gurobi's native Branch and Bound solver
- f_{QUBO} was evaluated by solving the proposed QUBO on D-Wave's Leap Hybrid Solver

Results



(a) Solution quality against number of nodes



(b) Solution quality against number of vehicles

Figure 1: Solution quality plotted against (a) graph size, ranging from 10 to 18 nodes, with a fixed fleet size of 5 vehicles across all simulations, and (b) fleet size, ranging from 3 to 5 vehicles, with a fixed graph size of 10 nodes for all experiments. Each box plot represents the results of 10 different experiments, each run with distinct problem instances generated using the same parameters.

Thank you