The Traveling Salesman Problem (TSP)

qoptmodeler package

nacedob

1 Introduction

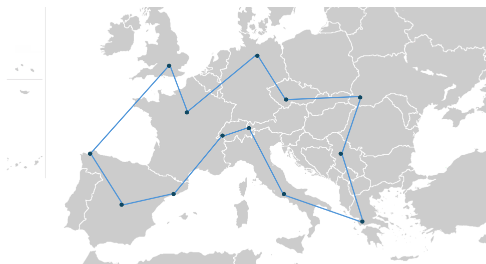

Figure 1: Example of TSP problem in Europe

The Traveling Salesman Problem (TSP) is a classic problem in the field of combinatorial optimization. It is concerned with finding the shortest possible route that visits each city exactly once and returns to the origin city. The problem is formally defined as follows:

Given a set of cities and the distances between every pair of cities, find the shortest possible tour that visits each city exactly once and returns to the starting city.

2 Mathematical Formulation

While there exist more complex formulations that can be more optimally implemented, for the sake of simplicity and to illustrate a simple example of the package usage, we will formulate the problem as follows:

Let:

- n: Number of cities.
- $T = \{0, \ldots, n-1\}$: Time steps, corresponding to the order in which cities are visited.
- $D \in \mathbb{R}^{n\times n}$: Cost matrix, where D_{ij} represents the distance between city i and city j.
- $x \in \{0,1\}^{n\times n}$: Binary decision matrix, where $x_{it} = 1$ if the traveling salesman is located at city i at time t, and 0 otherwise.

The problem can be formulated as:

$$\min \sum_{t=0}^{n-2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} D_{ij} x_{it} x_{j,t+1} + \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} D_{ij} x_{i,n-1} x_{j,0} \tag{1}$$

subject to:

$$\sum_{i=0}^{n-1} x_{it} = 1, \quad \forall t \in T \quad \text{(Each time step has one city)} \tag{2}$$

$$\sum_{t=0}^{n-1} x_{it} = 1, \quad \forall i \in \{0, \ldots, n-1\} \quad \text{(Each city is visited once)} \tag{3}$$

This can be rewritten in matrix form as:

$$\begin{aligned}
\text{Minimize} \quad & \boldsymbol{x}^t \boldsymbol{C} \boldsymbol{x} \\
\text{s.t.} \quad & A_{\text{time}} \boldsymbol{x} = \boldsymbol{b} \\
& A_{\text{space}} \boldsymbol{x} = \boldsymbol{b} \\
\text{where:} \quad & \boldsymbol{C} := \underbrace{\begin{pmatrix} 0 & 1 & \ldots & 1 \\ 1 & 0 & \ldots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \ldots & 0 \end{pmatrix}}_{\text{ones with diag} = 0} \otimes \left[\underbrace{\begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \\ 1 & 0 & 0 & \ldots & 0 \end{pmatrix}}_{\text{id with one row transposed}} \odot D \right] \in \mathbb{R}^{n \times n}
\end{aligned}$$

$$A_{\text{time}} = \left(\begin{array}{cccc|cccc|c|cccc} 1 & 0 & \ldots & 0 & 1 & 0 & \ldots & 0 & \ldots & 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & 0 & 1 & \ldots & 0 & \ldots & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 & 0 & 0 & \ldots & 1 & \ldots & 0 & 0 & \ldots & 1 \end{array} \right) = (\vec{1}_n^{\,t} \otimes I_n) \in \mathbb{R}^{nn \times n}$$

$$A_{\text{space}} = (I_n \otimes \vec{1}_n^{\,t}) \in \mathbb{R}^{nn \times n}$$

$$\boldsymbol{b} = \vec{1}_n \in \mathbb{R}^n$$

$$\vec{1}_n = (1, ..., 1)^T \in \mathbb{R}^n$$

$$I_n = \begin{pmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{pmatrix} \in \mathbb{R}^{n \times n}$$

$\odot$ is the Hadamard/element-wise product (link to Wikipedia)

$\otimes$ is the kronecker product (link to Wikipedia)

3 Example

Consider the following example with 4 cities (A, B, C, D) and their respective distances:

	A	B	C	D
A	0	10	15	20
B	10	0	35	25
C	15	35	0	30
D	20	25	30	0

4 Solutions

The TSP is NP-hard, meaning there is no known polynomial-time algorithm that solves it optimally for all cases. However, several heuristic and exact methods are used to find good solutions:

- Exact algorithms: Dynamic programming, branch and bound.
- Heuristic algorithms: Nearest neighbor, genetic algorithms, simulated annealing.
- Quantum algorithms: Quantum Approximate Optimization Algorithm (QAOA).

QAOA is a quantum-classical hybrid algorithm designed to approximate solutions to combinatorial optimization problems. To apply QAOA to the TSP, the problem must be formulated in terms of spin variables rather than binary variables. This transformation is necessary because quantum algorithms typically operate on Hamiltonians expressed in terms of Pauli matrices.

In this example, we will use the package `qoptmodeler` and its class `QuantumTranslator` to automatically perform the conversion, enabling us to seamlessly map the Traveling Salesman Problem (TSP) instance onto a quantum optimization framework. Next, we will utilize the `QAOASolver` class to obtain a solution to the problem.

5 Conclusion

The Traveling Salesman Problem is a fundamental problem in optimization and computer science. Despite its computational complexity, it has applications in logistics, circuit design, and many other areas where optimal routes or tours are required.