Final Project Proposal

EE 5283 Optimization in Engineering and Data Science
Fall 2023

Tien Van Nguyen
wdp076

Abstract—In this pursuit, we grapple with the intricacies of the Multiple Traveling Salesmen Problem (MTSP), an NP-hard challenge with far-reaching implications in transportation, shipping, and logistics. As the scale of the problem swells, stretching the capacities of classical computing devices like CPUs and GPUs, the quantum realm beckons as a potential solution. We begin by elegantly formulating the MTSP as a Mixed-Integer Linear Program, then turn to established solvers from MathWorks and Google OR-Tools to establish a foundation for comparison. Yet, the real breakthrough lies in our venture into quantum-based solving, building on the insights from [2], and subjecting it to rigorous statistical scrutiny. This sets the stage for a captivating face-off, pitting the computational efficiency of quantum-powered methods against their classical counterparts, promising not just practical applications, but an intellectual odyssey at the nexus of quantum computing and optimization.

I. Engineering context

The Multiple Traveling Salesman Problem (MTSP) is a general form of the well-known Traveling Salesman Problem (TSP), where multiple salesmen start from the initial position to visit a given number of cities exactly once and return to the initial position with the minimum traveling cost [1]. The solution is applied for various fields such as Transportation and delivery, WSN data collection and network connectivity, Search and rescue, etc. For instance, in smart city logistics, determining the optimal routes for a fleet of vehicles such as drones, and also minimize total travel distance is a critical task.

A. Definition

MTSP is widely studied and was originally defined as which, given a set of cities, one depot, m salesmen and a cost function (e.g. time or distance), MTSP aims to determine a set of routes for m salesmen minimizing the total cost of the m routes, such that each route starts and ends at the depot and each city is visited exactly once by one salesman. Figure 1 shows an illustration of MTSP, in which two workers start at depot v_1 to travel to all cities $v_2, \ldots, v_9$.

The MTSP minimizing the sum of all salesmen's tour costs can be stated in the general form [1] as below.

$$\begin{aligned}
\text{minimize}_{\text{Tour}_{R_i} \in \text{TOURS}} \quad & (\sum_{i=1}^{M} C(Tour_{R_i})) \\
\text{subject to} \quad & \text{Tour}_{R_i} \cap \text{Tour}_{R_j} = \emptyset \\
& \forall i \neq j, 1 \leq i, j \leq m \\
& \cup_{i=1}^{i=m} \text{Tour}_{R_i} = \{T_j, 1 \leq j \leq n\}
\end{aligned} \tag{1}$$

, where $C(\text{Tour}_{R_i})$ is the tour cost of agent R_i and TOURS is the set of all possible Tours, and $\{T_1, \ldots, T_n\}$ is a set of n places need to visit. Moreover, the two conditions in (1) guarantee that all targets are visited, and that each target is visited by only one agent. If $m = 1$, the problem becomes the standard (single) Traveling Salesman Problem (TSP).

B. Variants

The variants of MTSP result from considering the different characteristics of the salesmen, the depot, the city, and the problem constraints and objectives [1]. For example, the problem may consider a single depot, or multiple depots, or the salesman's path is closed, or open in some applications, the salesman does not need to return to the depot and it can stay in the last visited city; single or multiple objectives; etc.

C. Scope of this project

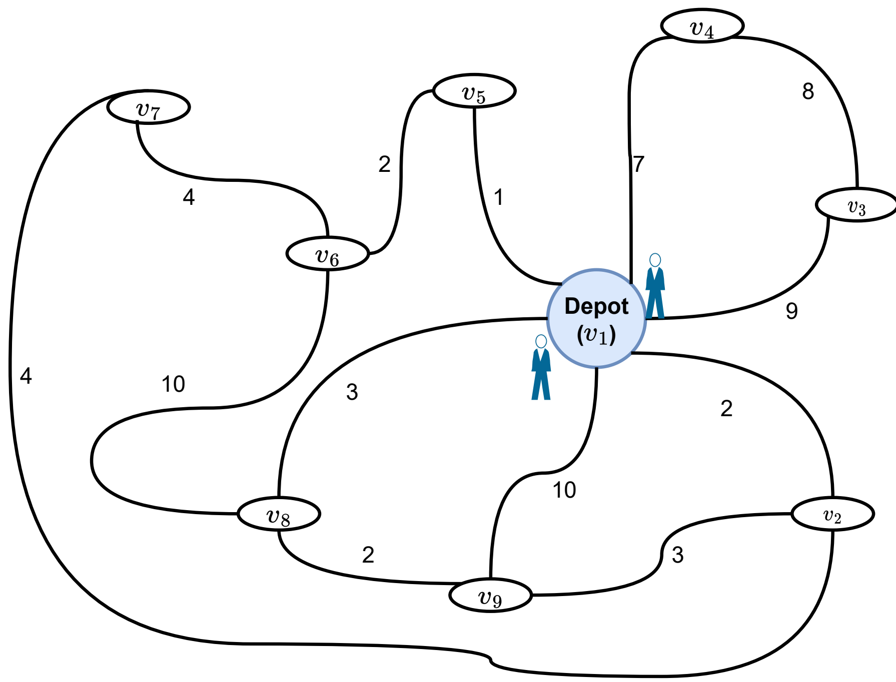
This project tries to solve the MTSP with the settings below. In summary, we formalize the problem as mixed-integer linear programming under the Miller-Tucker-Zemlin formulation [6], which is stated in (11).

- Salesmen do not share the same vehicle.
- One single static depot.
- All salesmen must start and return to the original starting point.
- Single cost function as the total distances of all salesmen traveled.

II. Problem formulation

A. Notation

We use graph representation to formalize MTSP. Let denote $G = (V, E)$ is the undirected graph established by V, E, in which $V = \{v_1, \ldots, v_n\}, E = \{(i, j)\}$ are the set of vertices, and set of edges respectively. In that, $n = |V|$ is the number of vertices, and (i, j) denotes the path between two vertices v_i, v_j. If G is complete (existing path between



	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
v_1		2	9	7	1			3	10
v_2	2						4		3
v_3	9			8					
v_4	7		8						
v_5	1					2			
v_6					2		4	10	
v_7		4				4			
v_8	3					10			2
v_9	10	3						2	

Fig. 1. Example for the multiple salesman traveling problem. The graph (on the left) shows that there are two salesmen, the v_1 is used as a depot where all salesmen start and also end their travel. There are 9 vertices (cities), some pairs of cities have the direct path moving between each other (e.g., $v_2, v9$), but some others do not (e.g., v_2, v_3). The table on the right side is the distance matrix which is constructed from the cost, especially the distance between cities. If there is a direct path between v_i, and v_j, the value corresponding to (v_i, v_j) is filled by the distance shown in the graph, otherwise, they are displayed by gray cells.

any two vertices), then $|E| = n(n-1)/2$. Each city is assigned with the ith vertex. The distance between two vertices v_i and v_j is $d_{ij} > 0$, $\forall i \neq j$, which is stored in a matrix $D \in {\mathbb{R}_+}^{n \times n}$. Due to G is undirected, D is symmetric, or $D^T = D$; and (k,l), (l,k) are the same for any pair $(k,l) \in E$. Simply, we only consider the set E containing pairs (k,l) with $k < l$.

We use the first vertex v_1 as the beginning location (called depot) of m salesmen. The goal is to find the formulation of tours for all m salesmen, so that:

- all vertices are visited exactly once, and
- selecting edges such that minimize the total traveling distance of all m salesmen.

Also, denote $X \in \{0,1\}^{n \times n}$, in which x_{ij} indicate whether we choose the pair (v_i, v_j) in the tour of solution or not. If (i,j) is chosen, then $x_{ij} = 1$, otherwise it would be 0.

$$x_{ij} = \begin{cases} 1 & ,(i,j) \text{ is included.} \\ 0 & ,o.w \end{cases} \tag{2}$$

B. Establish objective and constraints

The goal is to minimize the total distance of all m salesmen which is stated as the objective function below.

$$f(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \quad ,\forall\ (i,j) \in E, \text{and } i \neq j \tag{3}$$

with variables are x_{ij}, $1 \leq i,j \leq n$. We explicitly set that if $(i,j) \notin E$, $d_{ij} = \infty$; and $\forall i \in [1,n]$, then $d_{ii} = \infty$, to eliminate $(\forall\ (i,j) \in E, \text{and } i \neq j)$ in (3). Therefore, the objective function becomes:

$$f(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \tag{4}$$

To explain setting these values above to infinity to we want to minimize the objective function, and we also are not able to select edges are not exist in the graph, therefore extremely large value is justifiable, and it also helps the optimization algorithm not select this path to the routine solution ($x_{ij} = 0$). The same explanation for setting d_{ii} is the same. Although the distance between v_i and itself is literally 0, we do not want the agents to jump to the same position (not moving). Instead, setting d_{ii} would push the salesmen to validate other reachable vertices and help the algorithm work.

Besides minimizing the cost, we also design X such that satisfies these constraints:

- Every vertex v_i is included exactly once and every salesman, only enters and exits a vertex exactly one.

$$\sum_{j=1}^{n} x_{ij} = 1 \quad ,\forall\ v_i \in V \setminus \{v_1\} \tag{5}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad ,\forall\ v_j \in V \setminus \{v_1\} \tag{6}$$

- All m salesmen must start at the depot.

$$\sum_{j=2}^{n} x_{1j} = m \tag{7}$$

- All m salesmen must end their journey at the depot.

$$\sum_{i=2}^{n} x_{i1} = m \tag{8}$$

- In addition, to guarantee sub-tour elimination, the found solutions must satisfy:

$$\begin{aligned} &u_i - u_j + n \times x_{ij} \le n-1 \\ &\forall (i,j) \in E,\ 2 \le i \ne j \le n \end{aligned} \tag{9}$$

, with $u_i, \forall\ 1 \le i \le n$, be an integer decision variable for each vertex i, that represents visiting order in the tour (e.g., $u_3 = 4$ indicate before visiting the vertex v_3, there are already 3 vertices were cross through before). Thanks to setting $d_{ii} = \infty$, and $d_i j = 0, \forall\ (i,j) \notin E$, we can eliminate checking condition from ($\forall\ (i,j) \in E,\ 2 \le i \ne j \le n$) to ($\forall\ 1 \le i,j \le n$). Hence, (9) is written equivalently as below.

$$u_i - u_j + n \times x_{ij} \le n-1 \quad ,\forall\ 1 \le i,j \le n \tag{10}$$

C. Optimization problem

Based on these formulas above, we end up with the optimization problem which is written explicitly below. Given $V = \{v_1, \dots, v_n\}$, $E = \{(i,j)\}_{i,j=1,i\neq j}^{n}$ and $D \in {\mathbb{R}_+}^{n\times n}$, containing entries d_{ij}, we want to find the solution $X \in \{0,1\}^{n\times n}$ including x_{ij} such that:

$$\begin{aligned} \min_{X\in\{0,1\}^{n\times n}} \quad & f_0(X) = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} \\ \text{subject to} \quad & \sum_{j=1}^{n} x_{ij} = 1 \quad ,\forall\ v_i \in V \setminus \{v_1\} \\ & \sum_{i=1}^{n} x_{ij} = 1 \quad ,\forall\ v_j \in V \setminus \{v_1\} \\ & \sum_{j=2}^{n} x_{1j} = m \\ & \sum_{i=2}^{n} x_{i1} = m \\ & u_i - u_j + n \times x_{ij} \le n-1 \\ & \quad \forall\ 1 \le i,j \le n \end{aligned} \tag{11}$$

, which is a mixed-integer linear programming problem with variables $X \in \{0,1\}^{n\times n}$, and $u_i \in \mathbb{Z}_+$.

III. Preliminary analysis

We form the problem as a compact form (19), then introduce some characteristics of problems corresponding approaches seeking solutions from the research literature.

A. Compact form of original problem

Consider the product between 2 matrix $X \in \{0,1\}^{n\times n}$ containing column vectors $\{x_1, \dots, x_n\}$ and $D \in {\mathbb{R}_+}^{n\times n}$, which is symmetric matrix, containing column vectors $\{d_1, \dots, d_n\}$, we have:

$$\begin{aligned} (DX)_{ii} = d_i^T x_i &= \sum_{j=1}^{n} d_{ij}x_{ji} \\ &= \sum_{j=1}^{n} d_{ji}x_{ji} \end{aligned} \tag{12}$$

Consider the trace of DX, we have:

$$\begin{aligned} tr(DX) = \sum_{i=1}^{n}(DX)_{ii} &= \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ji}x_{ji} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}x_{ij} = f_0(X) \end{aligned} \tag{13}$$

Noticeably, $XD \neq DX$ in general, however $Tr(XD) = Tr(DX)$ holds, for any X, D satisfy the setting above. Therefore, the objective function 4 can be written equivalently as below.

$$f_0(X) = tr(DX) \tag{14}$$

The constraints (5) and (7) can be written equivalently as below.

$$X\mathbf{1} = \tilde{1} \tag{15}$$

, with $\mathbf{1}$ is an n-dimensional column vector with all entries equal 1, and $\tilde{1} = [m, 1, \dots, 1]^T$ is the n-dimensional column vector modified from $\mathbf{1}$ by replacing the 1st element by m.

Similarly, constraints (6) and (8) together are equivalent with:

$$X^T\mathbf{1} = \tilde{1} \tag{16}$$

Denote $u = [u_1, \dots, u_n] \in \mathbb{Z}_+^n$, The constraints (10) can be written as below.

$$u - \mathbf{1}u_j + nx_j \le \mathbf{1}(n-1) \quad ,1 \le j \le n \tag{17}$$

By defining $\mathbf{k}(j) : \mathbb{Z}_+ \mapsto \mathbb{Z}_+^n$ containing only one value 1 at j-position and other entries are 0, we can rewrite the equation above as below.

$$u - \mathbf{1}\mathbf{k}^T(j)u + nX\mathbf{k}(j) \preceq \mathbf{1}(n-1) \quad ,1 \le j \le n \tag{18}$$

Therefore, the problem in (11) can be written as a compact form below. Given $D \in \mathbb{R}_{+}^{n\times n}$, and $m \in \mathbb{Z}_{+}$, we seek to find binary matrix X, vector u such that:

$$\begin{aligned} \min_{X\in\{0,1\}^{n\times n}} \quad & f_0(X) = tr(DX) \\ \text{subject to} \quad & X1 = \tilde{1} \\ & X^T 1 = \tilde{1} \\ u - \mathbf{1}\mathbf{k}^T(j)u + nX\mathbf{k}(j) & \preceq \mathbf{1}(n-1) \\ 1 \leq j & \leq n \end{aligned} \tag{19}$$

with variables $X \in \{0,1\}^{n\times n}$, $u \in \mathbb{Z}_{+}^{n}$

B. Convexity

The problem (19) have $2n$ equalities ($X1 = \tilde{1}$, and $X^T 1 = \tilde{1}$), and n^2 inequalities (n ones for each j corresponding $u - \mathbf{1}\mathbf{k}^T(j)u + nX\mathbf{k}(j) \preceq \mathbf{1}(n-1)$).

Although the objective function is linear, all constraints are linear with respective X, it is non-convex (in general, the mixed integer linear program is non-convex) due to the domain of the problem, $X \in \{0,1\}^{n\times n}$, a discrete set, which is non-convex. Or we can show that it does not qualify the convexity condition, $\forall\ x, y \in \{0,1\}$,

$$\theta x + (1-\theta)y \in \{0,1\} \tag{20}$$

do not holds $\forall\theta \in [0,1]$.

C. Mixed-Integer Linear Programming Solvers

There are some Mixed-Integer Liner Programming Solvers which are available in well-known mathematical optimization frameworks such as Google OR-Tools[1], MathWorks[2].

D. Meta-heuristic based approaches

Such meta-heuristic approaches, namely Genetic Algorithm (GA) [3], and Practical Swarm Optimization (PSO) [4] also were investigated for solving MTSP.

E. Quantum-based solution

In [2], the authors investigated the Quantum Approximate Optimisation Algorithm (QAOA) to solve the m-TSP, leveraging the inherent parallelism and quantum computational power to potentially obtain high-quality solutions. By mapping the graph edges E to qubits and encoding the constraints inside the cost Hamiltonian.

F. Proposal works

In this project, we try to solve MTSP which is stated in (11). Due to the hardness of the NP-hard characteristic, which is intractable when scaling up the problem size (number of vertices, edges, salesmen), the quantum-based approach has the potential to deal with which needs to be more examined.

[1]https://developers.google.com/optimization/mip/mip_example
[2]https://www.mathworks.com/help/optim/ug/mixed-integer-linear-programming-algorithms.html

- Formalize the Multiple Salesmen Traveling Problem as the mixed-integer linear program.
- Use available solvers in MathWorks, and Google OR-Tools to find solutions.
- Implement the quantum-based solver following the design of [2]. The authors do not include the efficiency explicitly when showing the results, therefore a statistical result should be present.
- Finally, a comparison between experimented solvers should be analyzed on typical benchmark [5] to make a strongly qualitative conclusion on the performance of the current solver on MTSP.

References

[1] Cheikhrouhou, Omar, and Ines Khoufi. "A comprehensive survey on the Multiple Traveling Salesman Problem: Applications, approaches and taxonomy." Computer Science Review 40 (2021): 100369.

[2] Spyridis, Yannis, Athanasios Gkelias, and Vasileios Argyriou. "Variational Quantum Approach for the Multiple Traveling Salesman Problem Optimisation." 2023 19 th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT). IEEE, 2023.

[3] Yuan, Shuai, et al. "A new crossover approach for solving the multiple travelling salesmen problem using genetic algorithms." European journal of operational research 228.1 (2013): 72-82.

[4] Wei, Changyun, Ze Ji, and Boliang Cai. "Particle swarm optimization for cooperative multi-robot task allocation: a multi-objective approach." IEEE Robotics and Automation Letters 5.2 (2020): 2530-2537.

[5] Necula, Raluca, Mihaela Breaban, and Madalina Raschip. "Performance evaluation of ant colony systems for the single-depot multiple traveling salesman problem." Hybrid Artificial Intelligent Systems: 10th International Conference, HAIS 2015, Bilbao, Spain, June 22-24, 2015, Proceedings 10. Springer International Publishing, 2015.

[6] Miller, Clair E., Albert W. Tucker, and Richard A. Zemlin. "Integer programming formulation of traveling salesman problems." Journal of the ACM (JACM) 7.4 (1960): 326-329.