

CDMO - report paper

Luca Babboni, Gabriele Nanni, Luca Tedeschini

November 26, 2024

1 Introduction

The Multiple Traveling Salesperson Problem (mTSP) is a well-known generalization of the Traveling Salesperson Problem (TSP), in which more than one salesperson is allowed. The mTSP can be formalized as follows: given m couriers that must distribute $n \geq m$ items to different distribution points j , the goal is to minimize the maximum distance covered by the couriers. Additionally, each courier i has a maximum load capacity l_i , and each item j has a designated distribution point j and a size s_j . Every client must be served, and each courier must be able to transport its assigned load in a single trip.

The input data for each instance includes these parameters along with a distance matrix $D_{j \times j}$.

To tackle the problem, we chose to represent it as a 3D Boolean tensor: each vertical slice $j \times j$ of the structure corresponds to a single courier. This Boolean matrix can take values of 0 or 1. A value of 1 at position (x, y) indicates that the courier is traveling from x to y ; a 0 indicates otherwise. Stacking multiple slices on top of each other simplifies the formulation of many constraints. This representation is also commonly used in the literature (cite the reference).

This representation proved particularly useful in the SMT and MIP formulations, as it allowed many constraints to be expressed concisely. Another common strategy employed across all our formulations was the computation of upper and lower bounds (cite the reference).

This approach enabled us to achieve optimality in instances that were previously suboptimal, significantly boosting performance.

We all worked together most of the time, but each one of us specialized in one or more field: Gabriele Nanni worked mainly on the CP and SAT encoding, Luca Babboni worked mainly on the CP and MIP encoding and Luca Tedeschini worked mainly on the SMT and MIP. The software structure was mainly maintained by Luca Babboni, but everyone worked on it.

2 CP Model

2.1 Decision variables

2.2 Constraints

2.3 Validation

3 SMT model

The SMT model is totally based on the $3D$ tensor representation. We decided to implement it using the python library Z3py, based on the Z3 solver. We got mixed results because the model struggle to solve instance 7 while it is quite efficient on the other mandatory instances. It is not able to solve any instance above the tenth.

3.1 Decision variables

These are our decision variables:

- n the number of vehicles
- m the number of clients
- ub the upperbound of the objective function
- lb the lowerbound of the objective function
- $paths_{i,j,k}$ the tensor representation where the matrix $(i \times j)$ represents the trip of one courier, and the added dimensions $k = n$ are stacked to form the complete data structure
- $numvisit_i$ MZN subtour
- $maxdist$ the maximum distance travelled by the couriers

3.2 Objective function

The objective function is defined as

$$\sum_{i=1}^m \sum_{j=1}^m paths_{i,j,k} \cdot distances_{i,j} \leq maxdist \quad \text{for all } k \in 1 \dots n$$
$$\min(maxdist)$$

3.3 Constraints

Here's the list of constraints we have used to build our model

$$0 \leq numvisit_i \leq m - 1 \quad \forall i \in 1 \dots n \quad (1)$$

$$\sum_{j=0}^{m+1} paths_{i,j,k} = 1 \implies \sum_{j=0}^{m+1} paths_{i,k,j} = 1 \quad \forall k \in 1 \dots m, \forall i \in \dots n \quad (2)$$

$$\sum_{i=0}^n \sum_{j=0}^{m+1} paths_{i,j,k} = 1 \wedge \sum_{i=0}^n \sum_{j=0}^{m+1} paths_{i,k,j} = 1 \quad \forall k \in 1 \dots m \quad (3)$$

$$paths_{i,j,k} \implies numvisit_j < numvisit_k \quad \forall k, j \in 1 \dots m, \forall i \in 1 \dots n \quad (4)$$

$$\sum_{k=0}^m \sum_{j=0}^{m+1} paths_{i,j,k} * package_k \leq vehiclescap_i \quad \forall i \in 1 \dots n \quad (5)$$

$$\sum_{j=0}^{m+1} \sum_{i=0}^n paths_{i,j,j} = 0 \quad (6)$$

$$\sum_{k=0}^m paths_{i,m,k} = 1 \wedge \sum_{j=0}^m paths_{i,j,m} = 1 \quad \forall i \in 1 \dots n \quad (7)$$

$$lb \leq maxdist \leq ub \quad (8)$$

$$\forall i_1, i_2 \in \{1 \dots n\}, \quad i_1 < i_2 \wedge vehiclescap_{i_1} = vehiclescap_{i_2} \implies (\forall j, k \in \{1 \dots m\}, paths_{i_1,m,j} \wedge paths_{i_2,m,k} \wedge j < k) \quad (9)$$

- (1): Questo vincolo assicura che il numero di visite $numvisit_i$ di ciascun nodo i sia compreso tra 0 e $m - 1$, rispettando il dominio valido per le visite.
- (2): Questo vincolo garantisce che ogni nodo visitato abbia una connessione valida in ingresso e in uscita, rappresentando un circuito chiuso per ciascun veicolo.
- (3): Questo vincolo stabilisce che ciascun nodo intermedio venga visitato esattamente una volta in ingresso e una volta in uscita da tutti i veicoli combinati.
- (4): Questo vincolo impone che, se un arco collega j e k , il nodo j deve essere visitato prima del nodo k .

- (5): Questo vincolo limita il carico trasportato da ciascun veicolo in modo che non superi la capacità $vehiclescap_i$.
- (6): Questo vincolo elimina i cicli autoconnessi, impedendo che un nodo abbia un arco verso sé stesso.
- (7): Questo vincolo stabilisce che ogni veicolo inizi e termini il proprio percorso nel deposito m .
- (8): Questo vincolo impone che la distanza massima percorsa dai veicoli sia compresa tra i limiti inferiori (lb) e superiori (ub).
- (9): Questo vincolo di rottura della simmetria impedisce ridondanze tra veicoli con capacità uguali, imponendo un ordinamento tra le visite iniziali di tali veicoli.

3.4 Validation

Mettere i risultati in formato tabellare, ma prima controllare che gli indici abbiano senso perché nel modello su python sembra ci sia roba a caso che se cambi però funziona, sospetto. Una volta poi capiti si può completare la sezione sopra

4 MIP Model

This model is quite similar to the previous one, but thanks to the better performance of the gurobi solver we were able to find optimality in instances above the tenth.

4.1 Decision variables

These were our decision variables:

- ub the upperbound of the objective function
- lb the lowebound of the objective function
- $x_{i,j,k}$ the boolean tensor representation of the problem
- $y_{i,k}$ a boolean matrix representing if courier k is delivering at node i
- $d_{i,k}$ a continuous matrix used for the subtour elimination with the MZN formulation
- $maxdist$ the variable to minimize
- $NODES$ the number of nodes (considering the depot)
- $CLIENT$ the number of nodes (not considering the depot)
- $VEHICLES$ the number of couriers

4.2 Objective function

The objective function is the following

$$\sum_{i=0}^{NODES} \sum_{j=0}^{NODES} distances_{i-1,j-1} * x_{i,j,k} \leq maxdist \quad \forall k \in 1 \dots VEHICLES$$

$$\min(maxdist)$$

4.3 Constraints

As said before, the constraints are very similar to the SMT formulation:

$$lb \leq maxdist \leq ub \quad (10)$$

$$\sum_{k=1}^{VEHICLES} y_{0,k} = |VEHICLES| \quad (11)$$

$$\sum_{k=1}^{VEHICLES} y_{i,k} = 1 \quad \forall i \in CLIENTS \quad (12)$$

$$\sum_{j=0}^{NODES} \sum_{k=1}^{VEHICLES} = 1 \quad \forall i \in CLIENTS \quad (13)$$

$$\sum_{i=0}^{NODES} \sum_{k=1}^{VEHICLES} x_{i,j,k} = 1 \quad \forall j \in CLIENTS \quad (14)$$

$$\sum_{i=1}^{CLIENTS} y_{i,k} * packagesize_{i-1} \leq vehiclesscap_{k-1} \quad \forall k \in VEHICLES \quad (15)$$

$$\sum_{j=0}^{NODES} x_{j,j,k} \quad \forall k \in VEHICLES \quad (16)$$

$$\sum_{i=0}^{NODES} x_{i,0,k} \quad \forall k \in VEHICLES \quad (17)$$

$$\sum_{j=0'}^{NODES} x_{i,j,k} = \sum_{j=0}^{NODES} x_{j,i,k} \quad \forall i \in CLIENTS, \forall k \in VEHICLES \quad (18)$$

$$\sum_{j=0}^{NODES} x_{j,i,k} = y_{i,k} \quad \forall k \in VEHICLES, \forall i \in CLIENTS \quad (19)$$

$$\forall k \in VEHICLES, \forall i, j \in CLIENTS, i \neq j, \quad d_{i,k} - d_{j,k} + |CLIENTS| * x_{i,j,k} \leq |CLIENTS| - 1 \quad (20)$$

Spiegazione come sopra

4.4 Validation

Mettere risultati come sopra

5 Conclusion