Improved methods for the Travelling Salesperson with Hotel Selection

Marco Castro
marco.castro@ua.ac.be
ANT/OR

November 23rd, 2011

Contents

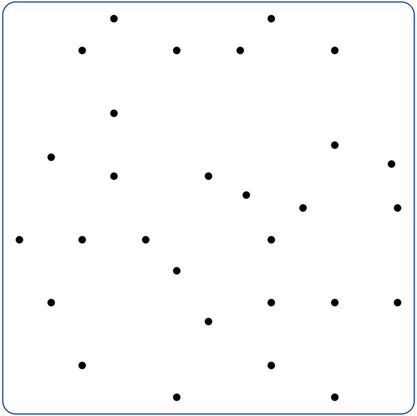
Contents

The classical Travelling Salesperson Problem

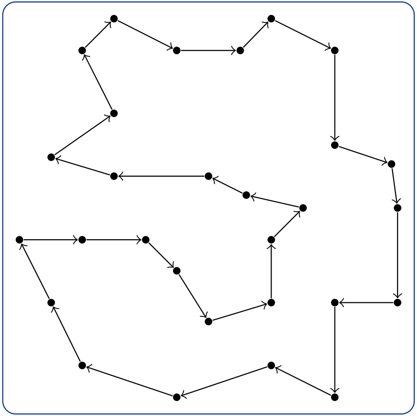
Definition

Given a collection of cities and the cost of travel between each pair of them, the **Travelling salesperson problem (TSP)** is to find the cheapest way of visiting all of the cities and returning to your starting point.

An instance with 30 cities



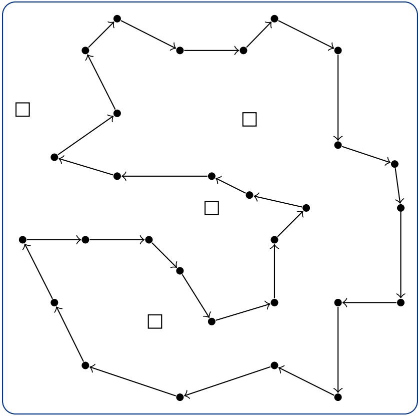
A TSP tour



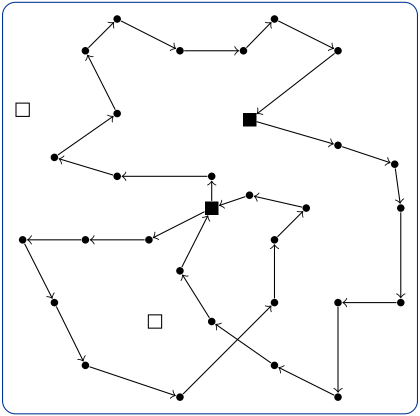
The Travelling Salesperson Problem with Hotel Selection

- A new variant of the TSP called Travelling Salesperson Problem with Hotel Selection (TSP-HS)
- A TSP tour is cut into trips no longer than a day, each of them staring and ending at a hotel
- A tour is a set of connected trips that combined visit all customers
- The objective is to minimize the number of connected trips and then minimize the total length of the tour

A TSP tour



A TSP-HS tour containing three trips



Motivation

The TSP-HS has a number of interesting applications:

- A travelling salesperson
- Truck drivers
- Planning touristic multi-day tours
- Google Street View

Mathematical formulation

- Given a set of $s+1$ hotels ($i = 0, \ldots, s$) and n customers ($i = s+1, \ldots, s+n$), let $c_{i,j}$ be the time needed to travel from location i to location j
- Let $x_{i,j,d}$ be a binary variable that takes the value of 1 if salesperson visit point j immediately after point i on trip d, or 0 if not
- Let y_d be a binary variable that denotes if the trip d is needed or not
- Each customer is assigned a service time T_i
- The available time that each trip takes is limited to a given time budget C

Mathematical formulation

- The set of hotels and/or customers visited on trip d is represented by N_d
- Let D denote the maximum number of trips and let M be a big number
- Each trips should start and end in one of the available hotels
- The ending hotel on day i has to be equal to the starting hotel on day $i+1$
- The starting and ending hotel of the tour are assumed to be identical and given ($i = 0$)

Mathematical formulation

minimize $\quad M\sum_{d=1}^{D} y_d + \sum_{i=0}^{s+n}\sum_{j=0}^{s+n}\sum_{d=1}^{D} c_{i,j}x_{i,j,d}$

subject to $\quad \sum_{d=1}^{D}\sum_{i=0}^{s+n} x_{i,j,d} = 1$, for $j = s+1,\dots,s+n$

$$\sum_{i=0}^{s+n} x_{i,j,d} = \sum_{i=0}^{s+n} x_{j,i,d}, \text{ for } j = s+1,\dots,s+n,\ d = 1,\dots,D$$

$$\sum_{i=0}^{s+n}\sum_{j=0}^{s+n} (c_{i,j} + T_j)\, x_{i,j,d} \le C, \text{ for } d = 1,\dots,D$$

Mathematical formulation

$$\sum_{h=0}^{s}\sum_{j=0}^{s+n} x_{h,j,d} = \sum_{h=0}^{s}\sum_{i=0}^{s+n} x_{i,h,d} = y_d, \text{ for } d = 1, \dots, D$$

$$\sum_{j=1}^{s+n} x_{0,j,1} = 1$$

$$\sum_{i=1}^{s+n} x_{i,0,d} \geq y_d - y_{d+1}, \text{ for } d = 1, \dots, D-1$$

$$x_{,i,jd} \leq y_d, \text{ for } i,j = 0, \dots, s+n, d = 1, \dots, D$$

$$y_d \geq y_{d+1}, \text{ for } d = 1, \dots, D-1$$

Mathematical formulation

$$\sum_{i=0}^{s+n} x_{i,h,d} + y_d \geq \sum_{j=0}^{s+n} x_{h,j,d+1} + y_{d+1}, \text{ for } d = 1, \ldots, D-1, h = 1, ..., s$$

$$\sum_{i=0}^{s+n} x_{i,h,d} - \sum_{j=0}^{s+n} x_{h,j,d+1} \leq 1 - y_{d+1}, \text{ for } h = 0, \ldots, s, d = 1, \ldots, D-1$$

$$\sum_{j \in S} \sum_{j \in S} x_{i,j,d} \leq |S| - 1 \text{ for } S \subset N_d,\ 2 \leq |S| \leq n-1$$

$$x_{i,j,d} \in \{0,1\},\ y_d \in \{0,1\}$$

Contents

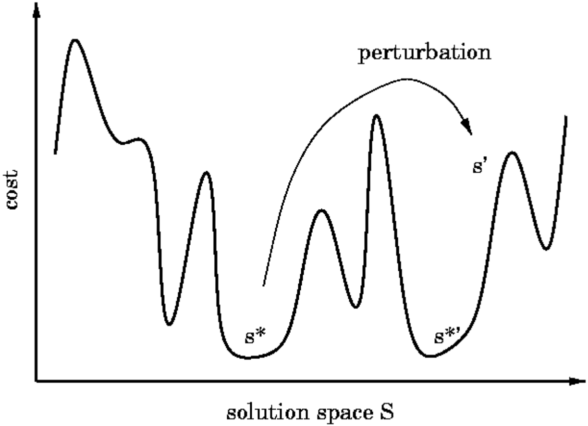
How to solve it?

- Exhaustive enumeration (brute force)
  - Impractical!
- Exact algorithms (eg. B&B, B&C, ...)
  - Able to solve small instances (≤ 40)
  - Still impractical!
- Metaheuristics
  - Good solutions in very limited computing times

Metaheuristics

- Allow to tackle large-size problem instances by delivering satisfactory solutions in a reasonable time
- There is no guarantee to find global optimal solutions
- Heuristics
  - Improve a solution by performing small changes to a solution
  - Trial and error
- The “Meta-” prefix
  - Ideas, strategies which allow to escape from local optima

Metaheuristics



Solution technique

- A multi-start procedure
  - Multi-Neighbourhood Local Search (MNLS)
  - Variable Neighbourhood Descend (VND)
- The algorithm implements well-known heuristics for solving optimization problems like TSP and VRP
  - involving a single trip, e.g. 2-opt
  - involving two trips, e.g. Relocation

Pseudocode

The main procedure performs as follows:

```
repeat
    Generate an initial solution
    Improve initial solution
until Maximum number of restarts is reached
return Best solution found
```

Pseudocode for the generation of initial solutions

repeat
    Generate a complete TSP tour in a greedy/randomized way
    Improve the tour using heuristics to improve a single trip
    Split the TSP tour into trips no longer than the maximum allowed length (C)
until Maximum number of restarts is reached
return Best initial solution found

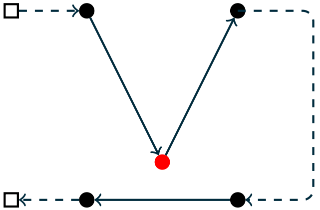
Heuristics for improving a trip

There are three available heuristics:

- Or-opt
- 2-opt
- 3-opt

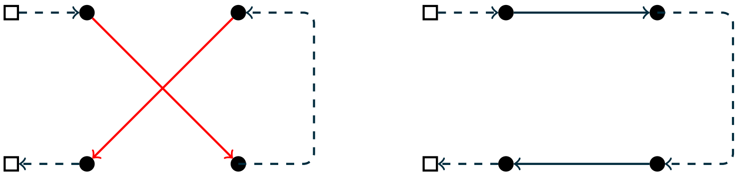
Or-opt

- Or-opt relocates a chain of consecutive customers inside a trip



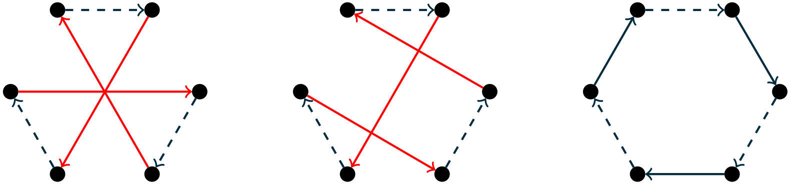
2-opt

- 2-opt removes two edges from the trip to replace them with two new edges not previously included in the trip



3-opt

- 3-opt removes three edges from the trip to replace them with three new edges not previously included in the trip



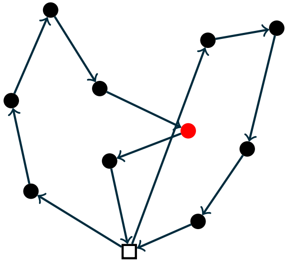
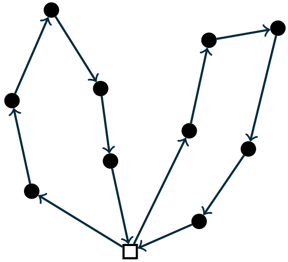
Improving a trip

Require: Set of neighbourhood structures $N_k, k = 1, \ldots, k_{\max}$

repeat
    for $k = 1 \rightarrow k_{\max}$ **do**
        Find the best neighbor x' of x $(x' \in N_k(x))$
        if x' is the best neighbor found so far **then**
            $x'_{best} \leftarrow x'$
        end if
    end for
    if x'_{best} is better than x **then**
        $x \leftarrow x'_{best}$
    end if
until No improvement is possible

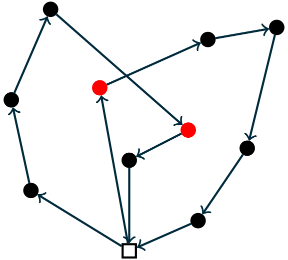
Relocate

- A string of at most k customers is moved from one trip to another

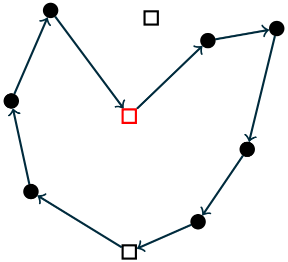
Exchange

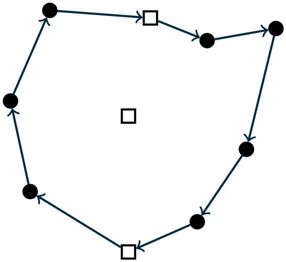
- Two strings of at most k customers are exchanged between two trips



Optimize hotels

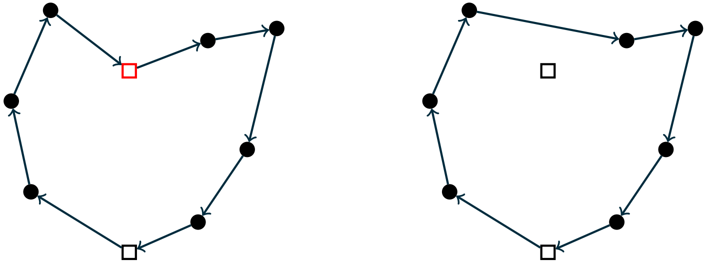
- Look at every pair of consecutive trips and it tries to change the intermediate hotel





Reduce trips

- Attempts to reduce the number of trips by joining two trips sharing the same hotel



Improving solutions

Once a “good” initial solution is generated, then it is improved:

```
repeat
    Relocate
    Exchange
    Optimize hotels
    Reduce trips
    Improve trips
until No improvement is possible
Update best solution
```

Contents

Computational experiments

- Set 1: made up of benchmarks for CVRP and MDVRP
  - Between 48 and 288 customers
  - Optimal solutions not known
- Set 2
  - Smaller instances containing 10, 15, 30 and 40 customers
  - Optimal solutions known for most of them
- Set 3
  - Instances generated from TSP benchmarks
  - Optimal solutions known for all of them
- Set 4
  - Same instances as in Set 3, with arbitrary budget
  - Optimal solutions not known

Results for SET 1

Instance	N	C	MNLS+VND Trips	MNLS+VND Length	Best known Trips	Best known Length
c101	100	1236	8	9669.9	9	9685.6
r101	100	230	8	1719.9	9	1801.3
rc101	100	240	8	1691.9	8	1724.1
c201	100	3390	3	9562.0	3	9600.0
r201	100	100	2	1646.2	2	1678.0
rc201	100	960	2	1645.7	2	1670.0
pr01	48	1000	2	1412.2	2	1446.0
pr02	96	1000	3	2552.3	3	2569.3

Results for SET 1

			MNLS+VND		Best known	
Instance	N	C	Trips	Length	Trips	Length
pr03	144	1000	4	3448.0	4	3584.1
pr04	192	1000	5	4266.5	5	4366.3
pr05	240	1000	6	5015.8	6	5122.1
pr06	288	1000	7	6064.6	7	6137.3
pr07	72	1000	3	2072.2	3	2090.9
pr08	144	1000	4	3391.3	4	3504.7
pr09	216	1000	5	4457.8	5	4617.6
pr10	288	1000	7	6012.5	7	6097.5

Results for SET 2 / 10 customers

		Gurobi		MNLS+VND		
Name	C	Trips	Length	Trips	Length	Gap (%)
c101	1236	1	955.1	1	955.1	0.00
r101	230	2	272.8	2	272.8	0.00
rc101	240	1	237.5	1	237.5	0.00
pr01	1000	1	426.6	1	426.6	0.00
pr02	1000	1	661.9	1	661.9	0.00
pr03	1000	1	553.3	1	553.3	0.00
pr04	1000	1	476.4	1	476.4	0.00

Results for SET 2 / 10 customers

		Gurobi		MNLS+VND		
Name	C	Trips	Length	Trips	Length	Gap (%)
pr05	1000	1	528.9	1	528.9	0.00
pr06	1000	1	597.4	1	597.4	0.00
pr07	1000	1	670.2	1	670.2	0.00
pr08	1000	1	573.4	1	573.4	0.00
pr09	1000	1	645.5	1	645.5	0.00
pr10	1000	1	461.5	1	461.5	0.00

Results for SET 2 / 15 customers

		Gurobi		MNLS+VND		
Name	C	Trips	Length	Trips	Length	Gap (%)
c101	1236	2	1452.2	2	1452.2	0.00
r101	230	2	379.8	2	379.8	0.00
rc101	240	2	303.2	2	303.2	0.00
pr01	1000	1	590.4	1	590.4	0.00
pr02	1000	1	745.6	1	745.6	0.00
pr03	1000	1	632.9	1	632.9	0.00
pr04	1000	1	683.4	1	683.4	0.00

Results for SET 2 / 15 customers

		Gurobi		MNLS+VND		
Name	C	Trips	Length	Trips	Length	Gap (%)
pr05	1000	1	621.2	1	621.2	0.00
pr06	1000	1	685.2	1	685.2	0.00
pr07	1000	1	795.3	1	795.3	0.00
pr08	1000	1	707.2	1	707.2	0.00
pr09	1000	1	771.7	1	771.7	0.00
pr10	1000	1	611.9	1	611.9	0.00

Results for SET 2 / 30 customers

		Gurobi		MNLS+VND		
Name	C	Trips	Length	Trips	Length	Gap (%)
c101*	1236	3	2829.4	3	2863.6	1.21
r101	230	3	655.2	3	656.1	0.15
rc101*	240	3	610.0	4	709.7	32.99
pr01	1000	1	964.8	1	964.8	0.00
pr02	1000	2	1078.3	2	1082.9	0.43
pr03	1000	1	952.5	1	952.5	0.00
pr04	1000	2	1091.6	2	1091.6	0.00

Results for SET 2 / 30 customers

		Gurobi		MNLS+VND		
Name	C	Trips	Length	Trips	Length	Gap (%)
pr05	1000	1	924.7	1	924.7	0.00
pr06	1000	2	1063.2	2	1069.3	0.57
pr07	1000	2	1130.4	2	1130.4	0.00
pr08	1000	2	1006.2	2	1006.2	0.00
pr09	1000	2	1091.4	2	1123.3	2.92
pr10	1000	1	918.9	1	918.9	0.00

Results for SET 2 / 40 customers

		Gurobi		MNLS+VND		
Name	C	Trips	Length	Trips	Length	Gap (%)
c101*	1236	4	3817.5	4	3867.3	0.11
r101*	230	4	842.9	4	862.8	0.41
rc101*	240	3	652.1	4	850.3	33.27
pr01	1000	2	1160.5	2	1170.9	0.90
pr02	1000	2	1336.9	2	1336.9	0.00
pr03	1000	2	1303.4	2	1316.1	0.97
pr04	1000	2	1259.5	2	1259.5	0.00

Results for SET 2 / 40 customers

		Gurobi		MNLS+VND		
Name	C	Trips	Length	Trips	Length	Gap (%)
pr05	1000	2	1200.7	2	1200.7	0.00
pr06	1000	2	1242.9	2	1251.1	0.66
pr07	1000	2	1407	2	1418.9	0.85
pr08	1000	2	1222.2	2	1222.2	0.00
pr09	1000	2	1284.2	2	1286.3	0.16
pr10	1000	2	1200.4	2	1200.4	0.00

Results for SET 3 / 3 extra hotels

Instance	N	TSP	C	Trips	Distance	Gap (%)
eil51	51	426	79	4	426	0.00
berlin52	52	7542	2041	4	7542	0.00
st70	70	675	181	4	675	0.00
eil76	76	538	138	5	558	3.71
pr76	76	108159	30734	4	108159	0.00
kroA100	100	21282	5639	4	21282	0.00
kroC100	100	20749	5269	4	20749	0.00

Results for SET 3 / 3 extra hotels

Instance	N	TSP	C	Trips	Distance	Gap (%)
kroD100	100	21294	5629	4	21330	0.17
rd100	100	7910	2048	5	8051	1.78
eil101	101	629	164	5	653	3.81
lin105	105	14379	3661	4	14379	0.00
ch150	150	6528	1656	5	6703	2.68
tsp225	225	3916	993	5	4050	3.42
a280	280	2579	436	5	2745	6.43
pcb442	442	50778	12819	5	52772	3.92
pr1002	1002	259045	65186	5	272453	5.17

Results for SET 3 / 5 extra hotels

Instance	N	TSP	C	Trips	Distance	Gap (%)
eil51	51	426	79	6	438	2.82
berlin52	52	7542	1478	6	7542	0.00
st70	70	675	132	6	675	0.00
eil76	76	538	101	6	555	3.16
pr76	76	108159	20331	6	108159	0.00
kroA100	100	21282	3819	6	21282	0.00
kroC100	100	20749	3751	6	20769	0.10

Results for SET 3 / 5 extra hotels

Instance	N	TSP	C	Trips	Distance	Gap (%)
kroD100	100	21294	3814	6	21330	0.17
rd100	100	7910	1424	7	8048	1.74
eil101	101	629	109	7	659	4.77
lin105	105	14379	2585	6	14379	0.00
ch150	150	6528	1131	7	6736	3.19
tsp225	225	3916	671	7	4098	4.65
a280	280	2579	436	7	2705	4.89
pcb442	442	50778	8678	7	53293	4.95
pr1002	1002	259045	43448	7	271599	4.85

Results for SET 3 / 10 extra hotels

Instance	N	TSP	C	Trips	Distance	Gap (%)
eil51	51	426	46	10	426	0.00
berlin52	52	7542	1148	8	7865	4.28
st70	70	675	72	10	675	0.00
eil76	76	538	55	12	569	5.76
pr76	76	108159	11268	11	108159	0.00
kroA100	100	21282	2202	11	21282	0.00
kroC100	100	20749	2215	11	20749	0.00

Results for SET 3 / 10 extra hotels

Instance	N	TSP	C	Trips	Distance	Gap (%)
kroD100	100	21294	2184	11	22962	7.83
rd100	100	7910	921	10	7910	0.00
eil101	101	629	65	12	655	4.13
lin105	105	14379	1827	10	14416	0.26
ch150	150	6528	678	11	6749	3.39
tsp225	225	3916	396	12	4152	6.03
a280	280	2579	248	13	2772	7.48
pcb442	442	50778	4830	13	54079	6.50
pr1002	1002	259045	23799	13	274110	5.82

Results for SET 4

Instance	C	Trips	Length	Trips	Length
eil51	85.2	6	430	6	479
berlin52	1508.4	7	8680	7	8823
st70	135.0	7	724	7	745
eil76	107.6	6	549	6	595
pr76	21631.8	7	117828	7	129789
rd100	1582.0	6	8255	6	8769
kroA100	4256.4	6	22127	7	22828
kroC100	4149.8	6	21342	7	23744

Results for SET 4

Instance	C	Trips	Length	Trips	Length
kroD100	4258.8	6	21691	6	24904
eil101	125.8	6	647	6	693
ch150	1305.6	6	6673	7	7679
tsp225	783.2	7	4692	7	4819
a280	515.8	6	2801	7	3123
pcb442	10155.6	7	56406	7	63822
pr1002	51809.0	7	303101	7	330282

Contents

Conclusions

- We provide a new number of optimal solutions for Set 2, however, the use of the mathematical formulation is still impractical
- The heuristic obtained good results when compared with Gurobi's optimal solutions and with known-optimal solutions (SET 3)
- Our MNLS+VND approach has been able to consistently produce better results than the previous approach
- In order to reduce the number of trips, we could relax the budget constraint and allow certain infeasibility