LocalSolver

Solving the Capacitated Arc Routing Problem with LocalSolver in an industrial context

Bienvenu Bambi

September 2023 - OR Hamburg

www.localsolver.com

LocalSolver

Optimization & Decision-Making Tool

- A generic, powerful solver
- 200 customers, 10,000 users, 25 countries
- Linear, non-linear and collection modeling
- Exact and heuristic techniques
- Quality solutions in seconds

Business problem

Public Asset Maintenance

Business problem : public asset maintenance

- Client's Goal: Efficient public asset maintenance.
    - Traverse entire road network within a specified area.
    - Service each road segment once.
- Constraints & Objectives:
    - Minimize total travel time and operational costs.
    - Navigate through various factors:
        - Traffic direction.
        - Speed limits.
- Desired Outcome:
    - Optimized route selection.
    - Boosted productivity and reduced overheads.

Arc routing: the underpinning concept

- Defining Arc Routing:
  - Routes are designed on arcs (or edges) rather than vertices (or nodes).
  - Ideal for problems where service is needed on roads/lanes, not at specific points.
- Ubiquity in the Industry:
  - Chinese Postman Problem (Classic example).
  - Other applications include snow removal, street cleaning, and mail delivery.
- Why It Matters:
  - CARP (Capacitated Arc Routing Problem) is NP-hard.
  - Solutions are computationally intensive and time-consuming.
  - The state-of-the-art usually involves heuristics like tabu search.
- Our Challenge:
  - Mapping the business problem to CARP.
  - Seeking optimal routes that fulfill constraints and minimize costs.

Our approach

Problem resolution

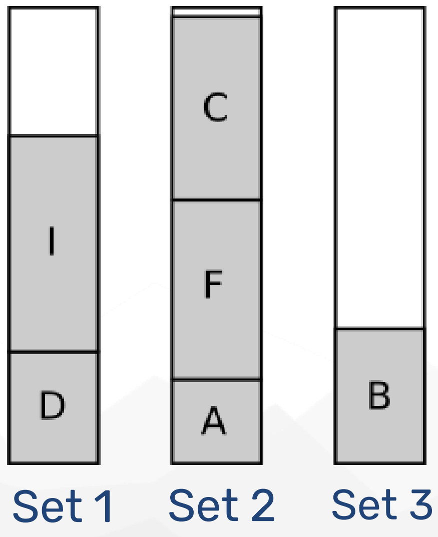
LocalSolver set modelling

Set

```
x <- set(n);
```

x : subset of {0, 1, …, n-1}

- unicity
- Variable size

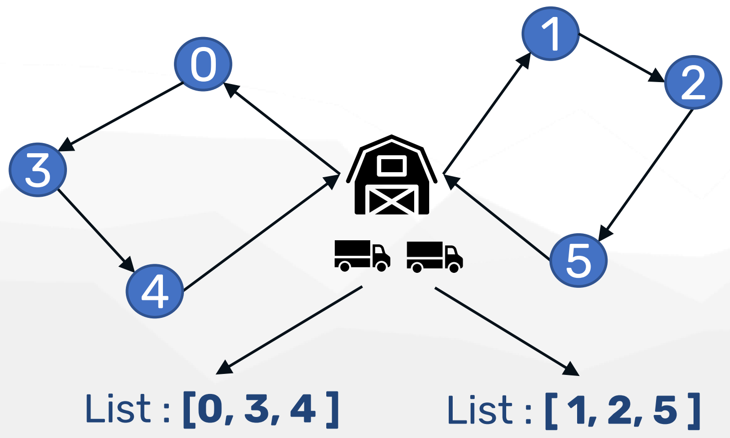


List

```
y <- list(n);
```

y : permutation of a subset of {0, 1, …, n-1}

- unicity
- Variable size
- **order**



LocalSolver set modelling: TSP example

```
function model() {
    // A list variable
    cities <- list(nbCities);

    // All cities must be visited
    constraint count(cities) == nbCities;

    // Minimize the total distance
    obj <- sum(1..nbCities-1, i =>
                distance[cities[i-1]][cities[i]])
              + distance[cities[nbCities-1]][cities[0]];

    minimize obj;
}
```

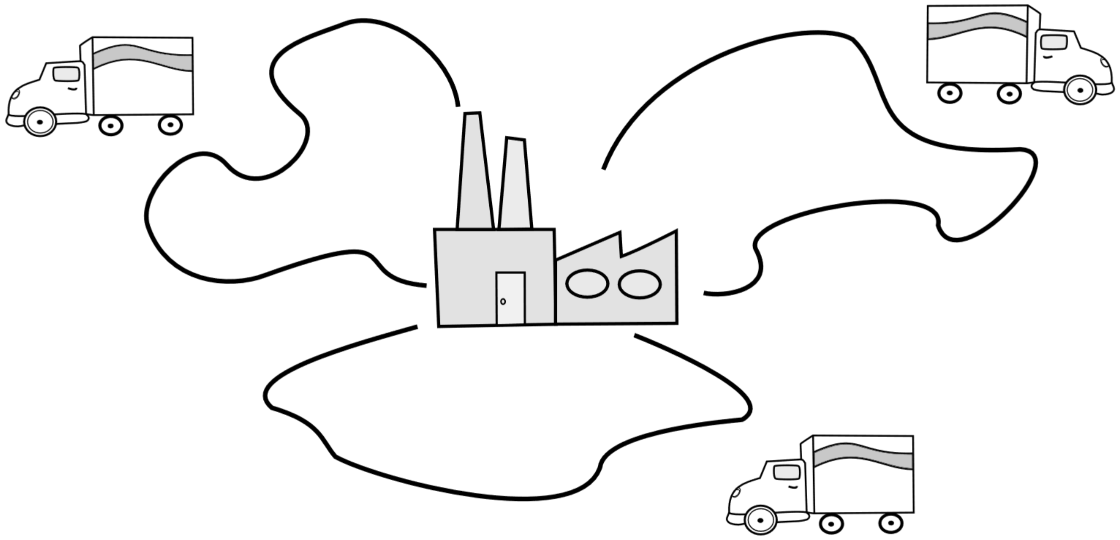
```
PS C:\localsolver_11_0\examples\tsp> localsolver .\tsp.lsp inFileName=
LocalSolver 11.0.20220214-Win64. All rights reserved.
Load .\tsp.lsp...
Run input...
Run model...
Run param...
Run solver...

Model:  expressions = 495, decisions = 1, constraints = 1, objectives
Param:  time limit = 60 sec, no iteration limit

[objective direction ]:     minimize

[  0 sec,       0 itr]: No feasible solution found (infeas = 2)
[  1 sec,   40639 itr]:         3036
[  2 sec,  105606 itr]:         2835
[  3 sec,  161355 itr]:         2787
[  4 sec,  223548 itr]:         2751
[  5 sec,  223548 itr]:         2751
[  6 sec,  342635 itr]:         2725
[  7 sec,  342635 itr]:         2725
[  7 sec,  404510 itr]:         2720
[ optimality gap     ]:           0%

404510 iterations performed in 7 seconds

Optimal solution:
  obj    =         2720
  gap    =           0%
  bounds =         2720
```

$$\sum_{i=1}^{n-1} distance[C_{i-1}][C_i]$$

LocalSolver set modelling: VRP example

Multiple trucks available:

```
tours[k] <- list(nbClients);
constraint partition(tours);
```



Capacity constraint for each tour k

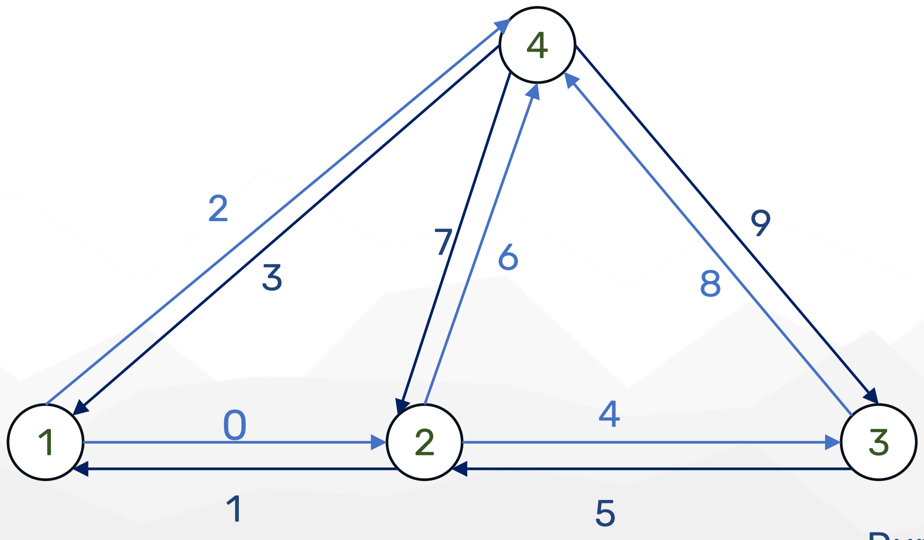
```
sum(tours[k], c => quantity[c]) <= capa;
```

$$\sum_{c \in tours[k]} quantity[c] \leq capa$$

Variadic number of elements in the sum

LocalSolver set modelling: CARP

- Duplicate edges of the graph and associate a unique number
  - Only one direction of the edge can be serviced
- List variable to model vehicles



edges = [(1, 2), (2, 1), (1, 4), (4, 1), (2, 3), (3, 2), (2, 4), (4, 2), (3, 4), (4, 3)]

edges = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Duplicate values are **alternatives**

LSP model overview

```
// Sequence of edges visited and "serviced" by each truck
// A sequence can contain an edge in one direction or its reverse
edgesSequences[k in 0..nbTrucks-1] <- list(2 * nbRequiredEdges);

// An edge must be serviced by at most one truck
constraint disjoint(edgesSequences);

// An edge can be travelled in both directions, but its demand must be satisfied only once
for [i in 0..nbRequiredEdges-1] {
 constraint contains(edgesSequences, 2 * i) + contains(edgesSequences, 2 * i + 1) == 1;
}
```

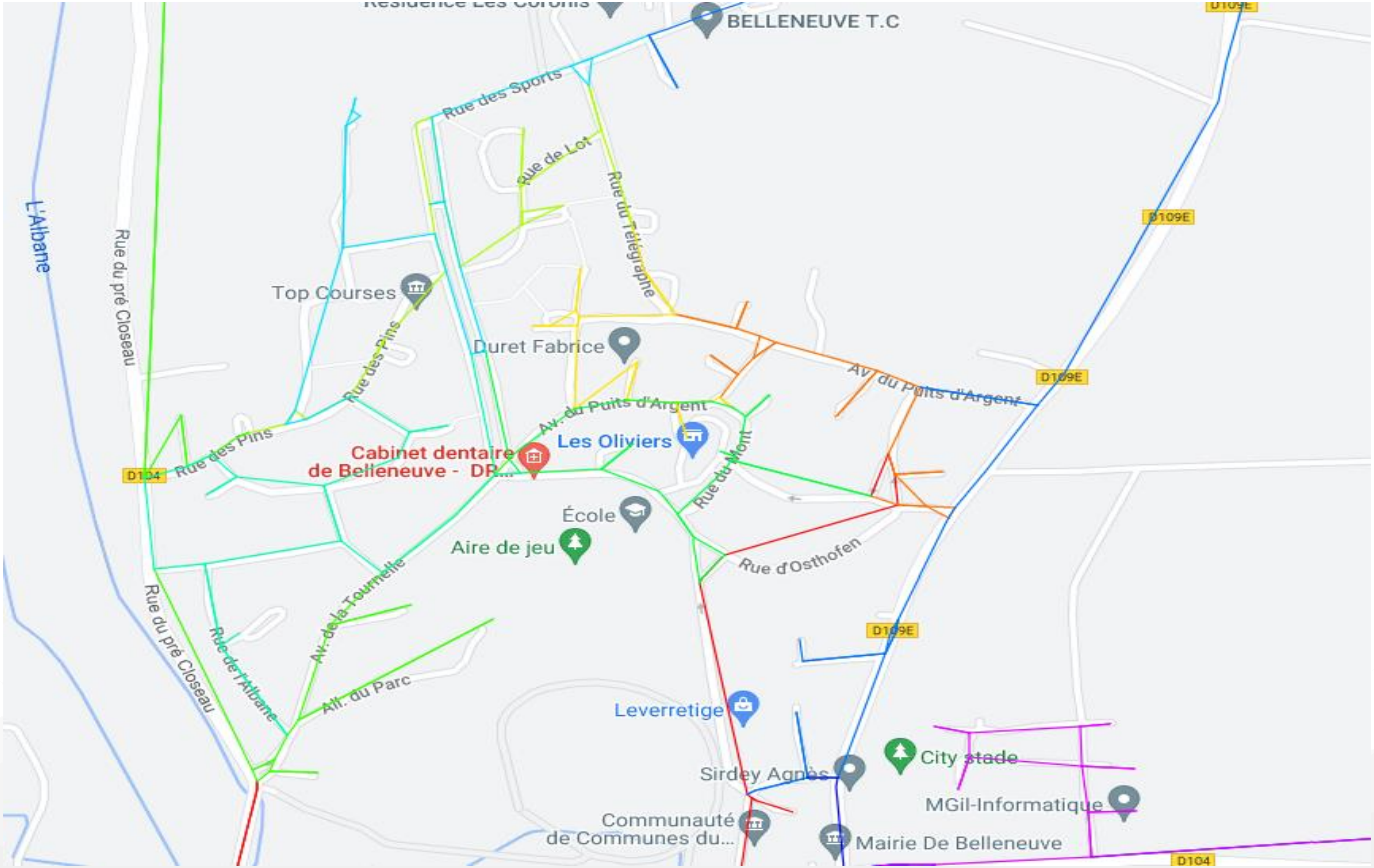
LSP model overview

```
for [k in 0..nbTrucks-1] {
    local sequence <- edgesSequences[k];
    local c <- count(sequence);

    // Quantity in each truck
    routeQuantity <- sum(0..c-1, i => requiredEdgesDemands[sequence[i]]);

    // Capacity constraint : a truck must not exceed its capacity
    constraint routeQuantity <= truckCapacity;

     // Distance travelled by each truck
    routeDistance[k] <- sum(1..c-1, i => requiredEdgesCosts[sequence[i]]
            + distanceBetweenEdges[sequence[i - 1]][sequence[i]])
            + (c > 0 ? requiredEdgesCosts[sequence[0]] + distanceFromDepot[sequence[0]]
            + distanceToDepot[sequence[c - 1]] : 0);
}
```

LSP model overview

```
// Total distance travelled
totalDistance <- sum[k in 0..nbTrucks-1](routeDistance[k]);

// Objective: minimize the distance travelled
minimize totalDistance;
```

Results obtained


BELLENEUVE T.C
Rue des Sports
Rue de Lot
Rue du Télégraphe
L'Albane
Rue du pré Closeau
Top Courses
Duret Fabrice
Rue des Pins
Av. du Puits d'Argent
Les Oliviers
Cabinet dentaire
de Belleneuve - DR...
D104
D109E
École
Aire de jeu
Rue du Mont
Rue d'Osthofen
Av. de la Tournelle
Rue de l'Albane
All. du Parc
Leverretige
City stade
Sirdey Agnès
MGil-Informatique
Communauté
de Communes du...
Mairie De Belleneuve

Numeric results

- J. Brandão, et R. Eglese. A Deterministic Tabu Search Algorithm for the Capacitated Arc Routing Problem (CARP), Computers & Operations Research., 2008
- 34 instances of 3 categories
  - 98 edges
  - 190 edges
  - 375 edges

Results

Instance	Gap to BK in 60 sec	Gap to BK in 600 sec
98	**0.27%**	**-0.19%**
egl-e1-A	0.00%	0.00%
egl-e1-B	0.60%	0.00%
egl-e1-C	0.32%	0.00%
egl-e2-A	0.00%	0.00%
egl-e2-B	0.11%	-0.36%
egl-e2-C	0.55%	-0.71%
egl-e3-A	0.00%	0.00%
egl-e3-B	0.18%	-0.22%
egl-e3-C	-0.33%	-0.58%
egl-e4-A	0.05%	0.05%
egl-e4-B	0.81%	0.18%
egl-e4-C	0.93%	-0.65%

Results

Instance	Moyenne de Gap to BK en 60 sec	Moyenne de Gap to BK en 600 sec
98	**0.27%**	**-0.19%**
190	**1.25%**	**0.03%**
egl-s1-A	0.00%	0.00%
egl-s1-B	-0.73%	-0.73%
egl-s1-C	0.76%	0.00%
egl-s2-A	2.29%	0.07%
egl-s2-B	4.86%	2.85%
egl-s2-C	0.18%	-1.09%
egl-s3-A	0.83%	0.09%
egl-s3-B	-0.52%	-1.43%
egl-s3-C	1.76%	0.31%
egl-s4-A	1.01%	-0.44%
egl-s4-B	-0.33%	-0.93%
egl-s4-C	4.89%	1.68%

Results

Instance	Moyenne de Gap to BK en 60 sec	Moyenne de Gap to BK en 600 sec
98	**0.27%**	**-0.19%**
190	**1.25%**	**0.03%**
375	**0.32%**	**-0.45%**
egl-g1-A	-0.30%	-0.81%
egl-g1-B	1.84%	0.84%
egl-g1-C	-0.01%	-0.38%
egl-g1-D	0.94%	0.28%
egl-g1-E	2.33%	0.94%
egl-g2-A	0.54%	-0.54%
egl-g2-B	-0.46%	-1.01%
egl-g2-C	-2.32%	-2.63%
egl-g2-D	0.18%	-0.84%
egl-g2-E	0.50%	-0.34%

LocalSolver

Capacitated Arc Routing

Bienvenu Bambi

bbambi@localsolver.com

September 2023 - OR Hamburg