LocalSolver

www.localsolver.com

Inventory routing optimization with LocalSolver

Thierry Benoist
tbenoist@localsolver.com

LocalSolver Optimizer

EASY MODELING

FAST SOLUTIONS

HIGHLY SCALABLE

A global Optimization Solver

- Fast solutions and optimality proofs

A simple and powerful modeling formalism

- nonlinear operators
- Set-based modeling

Designed for large scale problems

- Supply chain optimization
- Vehicle routing
- Production scheduling
- Media planning and pricing
- …

Combining exact and heuritic algorithms

- Simplex, Interior-Point, Augmented Lagrangian methods
- Black-box, derivative-free, surrogate modeling methods
- Branch-and-bound, cutting planes, interval methods
- Inference, propagation, clause learning
- Primal heuristics, large neighborhood search
- Metaheuristics, matheuristics
- Multiobjective optimization
- Statistical learning techniques for autotuning

LocalSolver

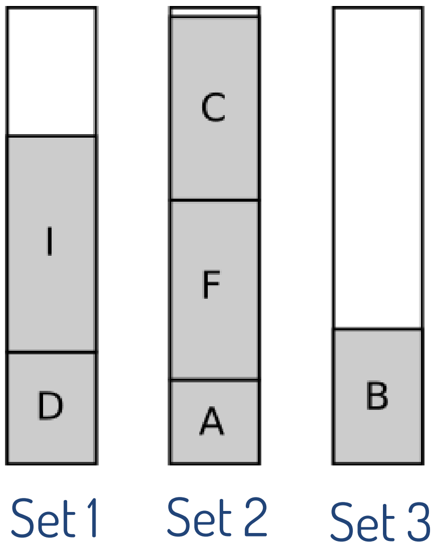
Set-based modeling

Set

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

x : a subset of {0, 1, …, n-1}

- Uniqueness of items
- Variable size

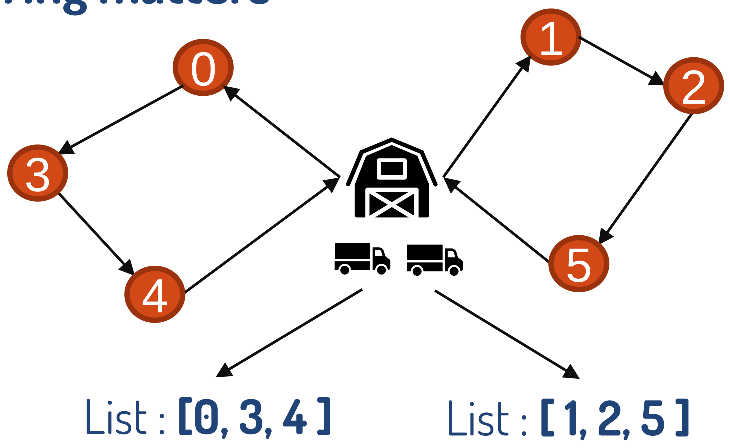


List

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

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

- Uniqueness of items
- Variable size
- **Ordering matters**



Set based modeling : 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]]);

    minimize obj;
}
```

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

Windows PowerShell

```
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
```

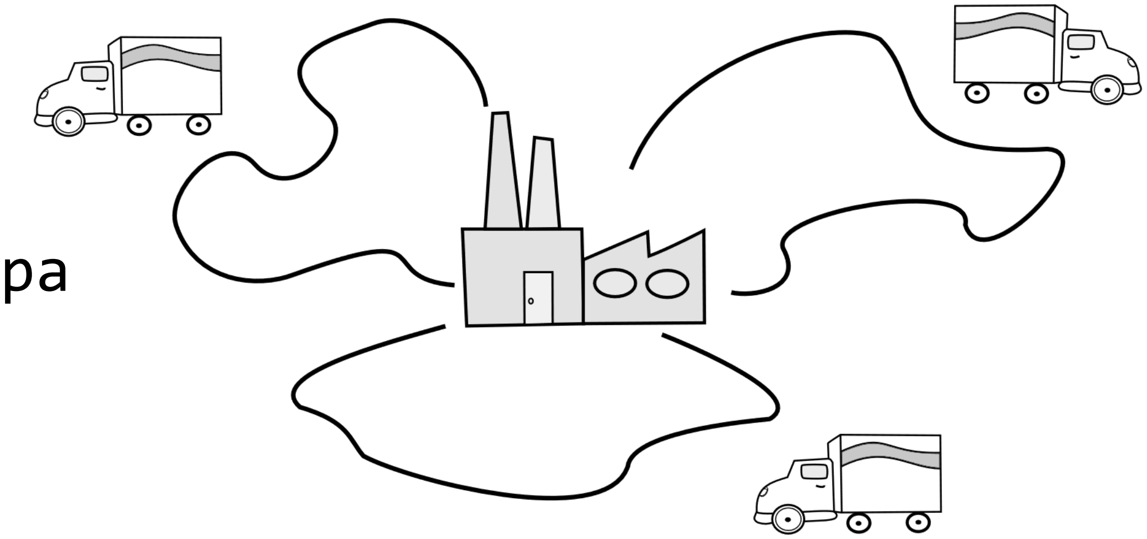
Set based modeling : vehicle routing example

We use several vehicles instead of one

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

We have a capacity limit for each tour k

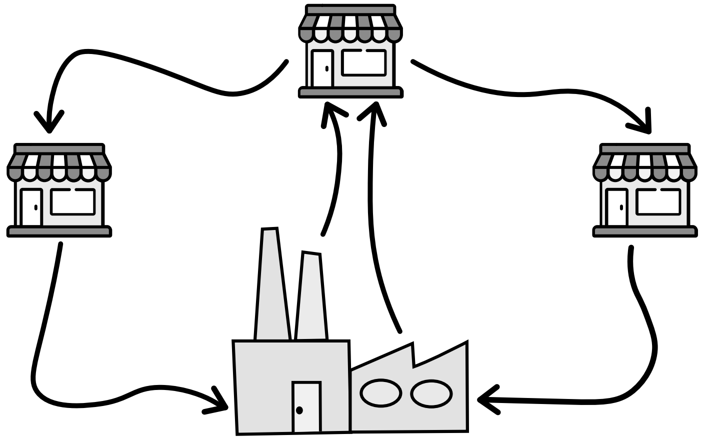
```
sum(tours[k], c => quantity[c]) <= capa
```



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

The number of terms in this sum is variable

Split delivery model



A client can be served by several vehicles, provided that the total delivered quantity is OK

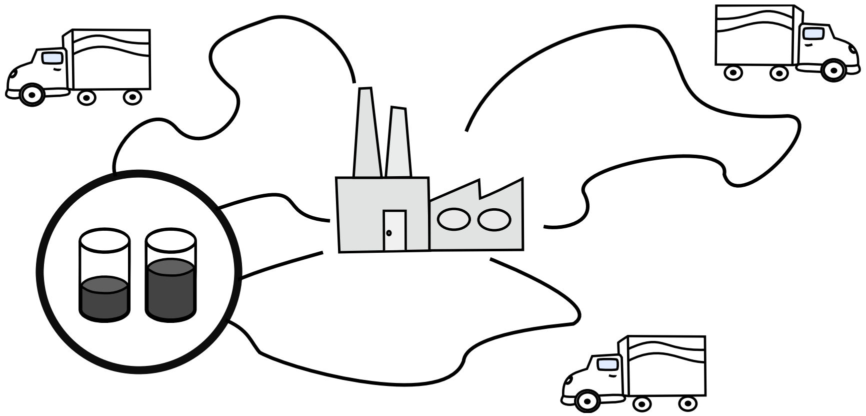
Modeling:

`cover(tours)` instead of `partition(tours)`

`quantity[client]` becomes `quantity[client][truck]`

→ it is now a decision variable, with an associated `sum` constraint

Inventory Routing model



Vendor Managed Inventory:
Compute one delivery tour per day and associated quantities such that no client reaches dry out level

Modeling:

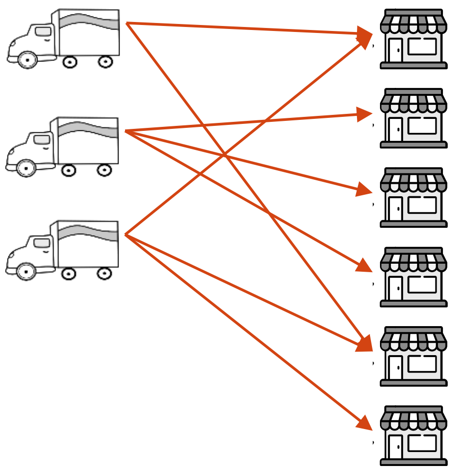
Neither `cover` nor `partition`: a client can be visited between 0 and nbDays times

`quantity[client]` becomes `quantity[client][day]`

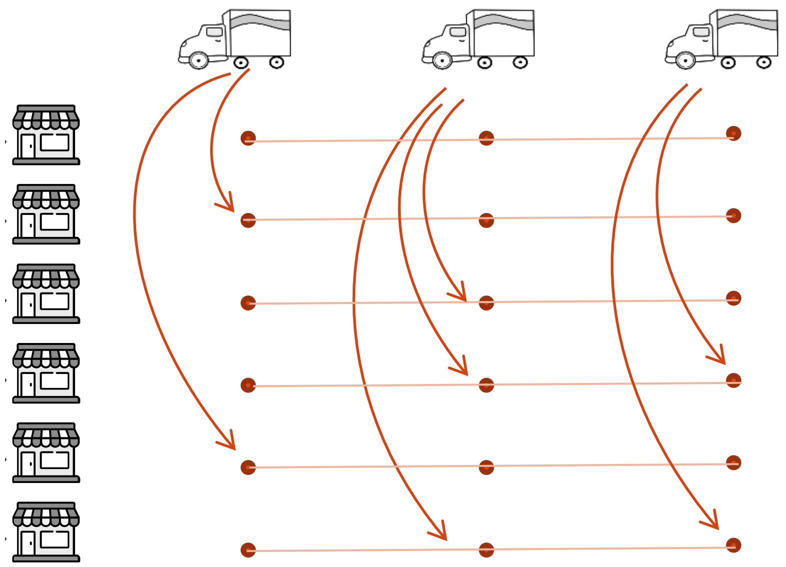
→ it is a decision variable, with associated linear constraints computing the inventory level for each client at the end of each day

Under the hood (1/2)

In both Split Delivery and Inventory routing models, once list variables are fixed, the residual problem (on quantity variables) is linear



Split Delivery



Inventory routing

Under the hood (2/2)

Inside LocalSolver, the local search component cooperates with the LP component

Each time a transformation is applied to combinatorial variables (lists), numeric variables are repaired to their linear optimum

This mecanism is not specific to routing

$$\min f_1(x) + f_2(y)$$
$$s.c.$$
$$g(x) = 0$$
$$h(y) = 0$$
$$l(y) \leq x \leq u(y)$$
$$y \in \boldsymbol{Z}^q$$

If f1 and g are linear,
the residual problem once
red terms are fixed is linear

Applications

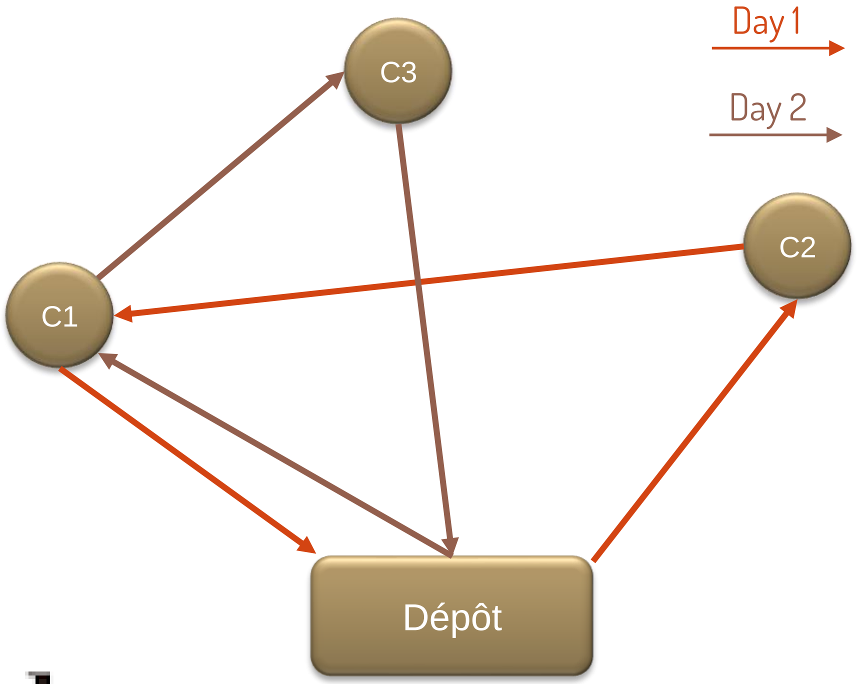
Split delivery for a Japanese client

Factories operate from Monday to Saturday morning.
➔ deliveries must occur on Saturdays afternoon and Sundays

100 clients, 100 trucks, 15mn solving time

For each client:

- Demand (can exceed the capacity of the largest truck)
- Max number of deliveries
- Time-windows
- Allowed truck sizes (2 tons or 4 tons or both)



C1 receives two deliveries

BrainPad

Results

The goal is to minimize a linear combination of **the number of trucks**, the **traveled distance**, and the **working time**.

	Gurobi 9.1	LocalSolver 11.0
36 clients and 36 trucks	13,747	13,760
50 clients and 60 trucks	25,350	23,221
100 clients and 115 trucks	No solution	51,585

Results after 15 minutes (minimization)

Inventory Routing – Air Liquide (Canada,Germany, USA)

Highly complex route optimization

- Up to 500 clients to serve
- Order & inventory management
- Heterogeneous resources (drivers, tractors, trailers) with time windows
- Sourcing and delivery time windows
- Multiple depots, multiple products
- Complex driver rules and regulations
- High quality solutions in minutes

Step by step

2018 : Performance benchmark (Proof-of-Concept) organized by Air Liquide

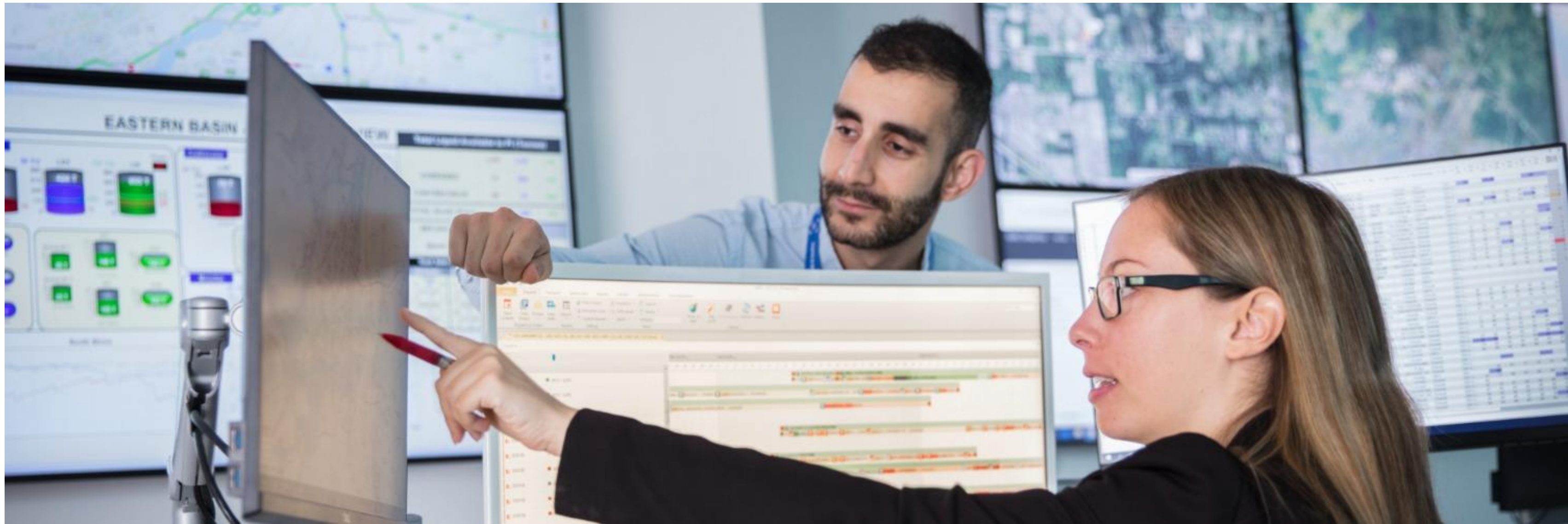
2019: Minimum Viable Product deployed in Canada →

2020- : Deployment country by country

2%-7% savings on $/ton

$250,000 annually on a small region

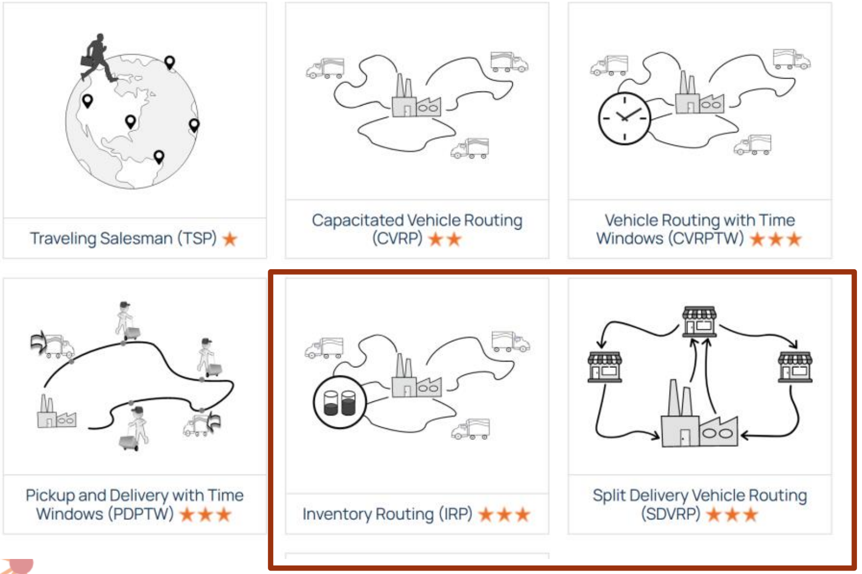
ALIZENT
ASSET INTERACTIVE



Conclusion

An easy modeling of inventory routing problems

- See our example tour for detailed models on www.localsolver.com



SDVRP and IRP optimization for industries

- Large-scale problems
- Many practical constraints to be taken into account
- Significant savings

BrainPad

Air Liquide