# Toward a mathematical programming solver based on neighborhood search

Frédéric Gardi

Innovation 24 & LocalSolver

13th December 2013
ORO Workshop, Nantes

# Who we are

Bouygues, one of the French largest corporation, €33 bn in revenues

Innovation 24, business analytics & optimization subsidiary of Bouygues

LocalSolver, mathematical optimization solver commercialized by Innovation 24

# Plan

1 – Local search: how to industrialize?

2 – Methodological feedbacks

3 – Local search for nonlinear 0-1 programming

4 – Toward a new kind of optimization solver

# Local search

How to industrialize?

# Local search

## An iterative improvement method

- Explore a neighborhood of the current solution
- Smaller or larger neighborhoods

⟶ Incomplete exploration of the solution space

## Essential in combinatorial optimization

- Hidden behind many textbook algorithms (ex: simplex, max flow)
- In the heart of all metaheuristic approaches
- Proved to be not efficient in the worst case
- Largely used because very effective in practice

# How to industrialize?

## 2000–2005: initiation

- OR engineering for Prologia: workforce scheduling
- ROADEF 2005 Challenge: car sequencing for Renault
  ($1^{st}$ Prize Junior & Senior with B. Estellon and K. Nouioua)

## Contributions

- Methodological feedbacks for combinatorial optimization
- Methodological feedbacks for mixed-variable optimization
- A solver for combinatorial optimization exploiting local search
- A hybrid, all-in-one solver based on neighborhood search
  for (large-scale) mixed-variable non-convex optimization

# Methodological feedbacks

Ten years of local search

# Why local search?

## When it is hopeless to enumerate

- Large-scale combinatorial problems
- When relaxation or inference brings nothing
  (ex: linear relaxation is very fractional)
- When computing relaxation or inference is costly

## Adapted to client needs

- Good-quality optima satisfy them
- Fast: each iteration runs in sublinear or even constant time

⟶ Solutions in short running times + ability to scale

# Methodological keys

## An appropriate search space

- To enlarge and densify the search space
- Goals (= objectives) instead of constraints
- Operational optimization model = good search space

## Local search: back to basics

- Don’t focus on “meta” aspects
- Focus on: enrich/enlarge moves, speed up move evaluation
- Let tests and client feedbacks guide you

⟶ Ultimately high-performance local search is a matter of expertise in algorithmics and of dexterity in computer programming

# Industrial applications

## Combinatorial optimization

- Car sequencing for Renault (2005)
- Technical intervention scheduling for France Telecom (2007)
- TV media planning for TF1 (2011)

## Mixed-variable optimization

- Inventory routing for Air Liquide (2008)
- Earthwork scheduling for DTP Terrassement (2009)
- Outage scheduling for EDF (2010)

# LocalSolver

Local search for nonlinear 0-1 programming

# Existing tools to automate local search

## Libraries and frameworks

- Complex to handle
- Limited to practitioners having programming skills
- Don't address key points (ex: moves)

## Solvers integrating "pure" local search

- Pioneering works in SAT community
- MIP and CP: a few attempts (Nonobe & Ibaraki 2001), not really conclusive
- MIP and CP: a lot of heuristic ingredients but no "pure" local search

# LocalSolver project

## 2007: launch of the project

- To define a generic modeling formalism (close to MIP) suited for a local search-based resolution (*model*)
- To develop an effective solver based on pure local search with first principle: “to do what an expert would do” (run)

## 2009-2011: release 1.x

- Large-scale combinatorial problems, especially assignment, partitioning, packing, covering problems, out of scope of classical solvers
- Use and integration in optimization solutions for Bouygues:
  TF1 Publicité, ETDE, Bouygues Telecom
- First uses outside Bouygues Group (ex: Eurodecision)

# P-median problem

## Select a subset P among N points minimizing the sum of distances from each point in N to the nearest point in P

```
function model() {
  x[1..N] <- bool() ; // decisions: point i belongs to P if x[i] = 1

  constraint sum[i in 1..N]( x[i] ) == P ; // constraint: P points selected among N

  minDist[i in 1..N] <- min[j in 1..N]
              ( x[j] ? Dist[i][j] : InfiniteDist ) ; // expressions: distance to the nearest point in P

  minimize sum[i in 1..N]( minDist[i] ) ; // objective: to minimize the sum of distances
}
```

Nothing more to write

# Available mathematical operators

| Arithmetical | | | Logical | Relational |
|---|---|---|---|---|
| sum | sub | prod | not | == |
| min | max | abs | and | != |
| div | mod | sqrt | or | <= |
| log | exp | pow | xor | >= |
| cos | sin | tan | if | < |
| floor | ceil | round | array + at | > |

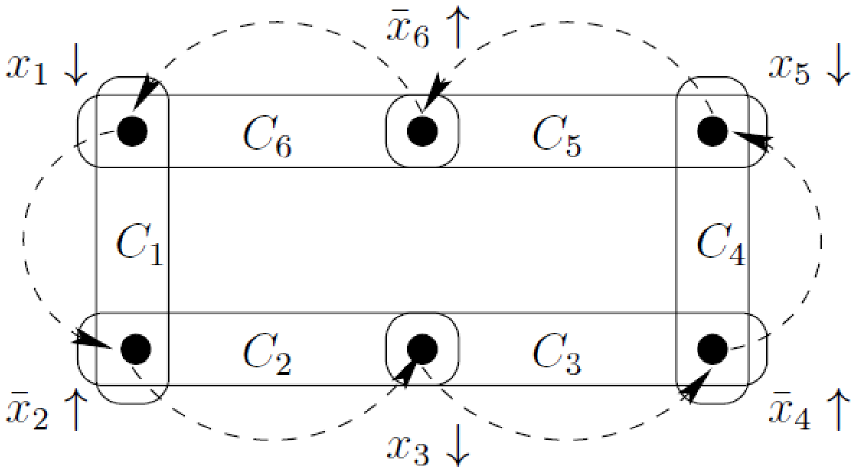
# Small, structured neighborhoods

## The classic in Boolean Programming: "k-flips"

- Lead to infeasible solutions for structured (= real-life) problems
- Feasibility is hard to recover: slow convergence

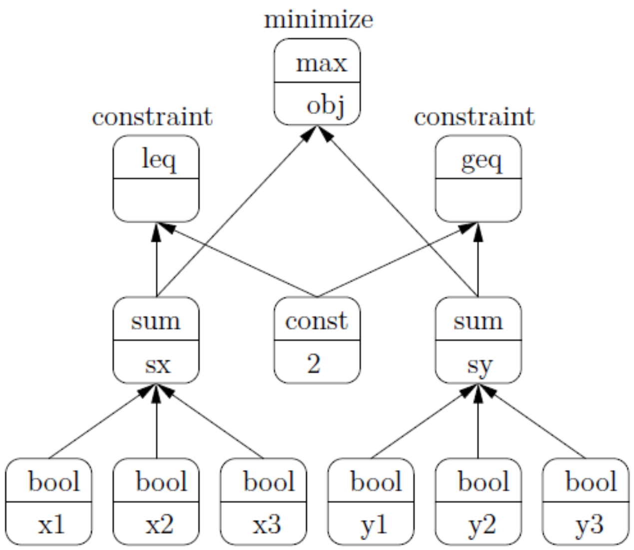
## LocalSolver moves tend to preserve feasibility

- Destroy & repair approach
- Ejection paths in the constraint hypergraph
- More or less specific to some combinatorial structures



# Fast exploration

```
x1 <- bool();
x2 <- bool();
x3 <- bool();
y1 <- bool();
y2 <- bool();
y3 <- bool();
sx <- sum(x1, x2, x3);
sy <- sum(y1, y2, y3);
constraint leq(sx, 2);
constraint geq(sy, 2);
obj <- max(sx, sy);
minimize obj;
```




## Incremental evaluation

- Lazy propagation of modifications induced by a move in the DAG
- Exploitation of invariants induced by math operators

⟶ Millions of moves evaluated per minute of running time

# LocalSolver 2.x et 3.x

## 2012: commercial launch of release 2.0

- To support financially the project over the long term
- To ensure to match the practical needs through user feedbacks
- To spread the software (and our ideas) out of Bouygues and out of France

## 2013: release 3

- 1500 visits per month on localsolver.com: thousands of downloads
- 400 registered users including 300 out of France
- 530 distributed licenses including 330 free academic licenses
- 15 commercial licenses (including support) sold out of Bouygues
  France: Air Liquide, Armée de Terre, Publicis, French universities
  International: Pasco, Fujitsu, Hitachi, NIES, Chinese universities

# Application panorama

Pasco FUJITSU — Supply chain optimization

MEDIATRANSPORTS — Design of outdoor advertising networks

TF1 PUBLICITE — TV media planning

AIR LIQUIDE — Vehicle routing, logistic clustering

ETDE — Street lighting maintenance scheduling

Bouygues Telecom — Mobile network design

SIEMENS — Energy optimization of tramway lines

EDF — Maintenance of nuclear assemblies (QAP)

SSI STRATEGIC SYSTEMS INTERNATIONAL — Hospital timetabling

ARMEE DE TERRE — Transport of military equipment

# Car sequencing for Renault

**2005 ROADEF Challenge:** http://challenge.roadef.org/2005/en

## Large-scale instances

- 1,300 vehicles to sequence: 400,000 binary decisions

## Instance 022_EP_ENP_RAF_S22_J1: 540 vehicles

- Small instance: 80,000 variables including 44,000 0-1 decisions
- State of the art: 3,109 by specific local search (winner of the Challenge)
- Lower bound: 3,103

**Minimization**

## Results

- Gurobi 5.5: 3.027e+06 in 10 min | 194,161 in 1 hour
- LocalSolver 3.1: 3,476 in 10 sec | 3,114 in 10 min

# Machine scheduling for Googlee

**2012 Challenge ROADEF/EURO:** http://challenge.roadef.org/2012/en

- Schedule processes on Google servers
- Running time limited to 5 minutes on a standard computer
- Ex: 10 M expressions, 300,000 constraints, 500,000 decisions

- 100-line model solved with LocalSolver (2.0)
- Ranked 25th over 82 teams (30 countries)
- Sole model-and-run solver to be qualified for final tour (30 teams)

# Routing problems

**TSP** http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95

- Asymmetric TSP
- LocalSolver 3.1 launched for 5 min on standard computer
- Average gap against state of the art: 3 %

**VRP** http://neo.lcc.uma.es/vrp/vrp-instances

- CVRPTW on Solomon instances
- LocalSolver 3.1 launched for 5 min on standard computer
- Average gap against state of the art: 14 %

# MIPLIB

## Some results on the largest & hardest instances

- 5 min for both LocalSolver and Gurobi
- MIP-oriented models: not suited for LocalSolver

**Minimization**

| Instances | Variables | LocalSolver 3.1 | Gurobi 5.1 |
|---|---|---|---|
| ds-big | 174,997 | 9,844 | 62,520 |
| ivu06-big | 2,277,736 | 479 | 9,416 |
| ivu52 | 157,591 | 4,907 | 16,880 |
| mining | 348,921 | - 65,720,600 | 902,969,000 |
| ns1853823 | 213,440 | 2,820,000 | 4,670,000 |
| rmine14 | 32,205 | - 3,470 | - 171 |
| rmine21 | 162,547 | - 3,658 | - 185 |
| rmine25 | 326,599 | - 3,052 | - 161 |

# LocalSolver

Toward a hybrid optimization solver based on neighborhood search

# A new kind of solver

## John N. Hooker (2007)

"Good and Bad Futures for Constraint Programming (and Operations Research)"
Constraint Programming Letters 1, pp. 21-32

"Since modeling is the master and computation the servant, no computational method should presume to have its own solver.

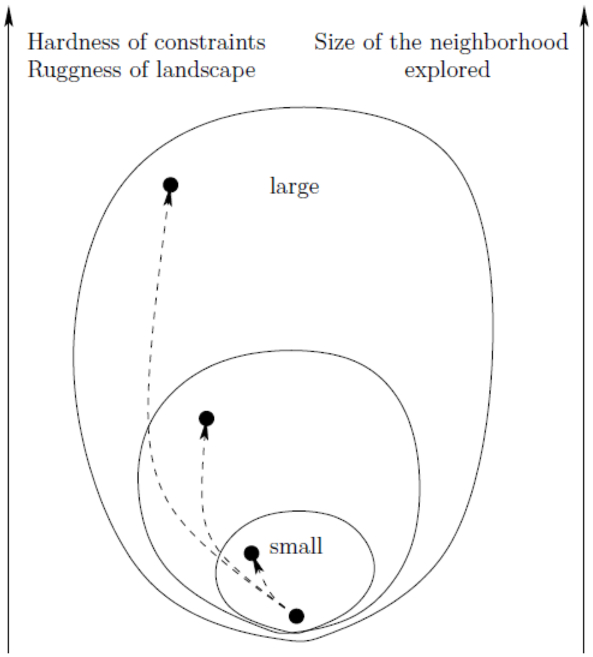
This means there should be no CP solvers, no MIP solvers, and no SAT solvers. All of these techniques should be available in a single system to solve the model at hand.

They should seamlessly combine to exploit problem structure. Exact methods should evolve gracefully into inexact and heuristic methods as the problem scales up."

# How to hybridize?

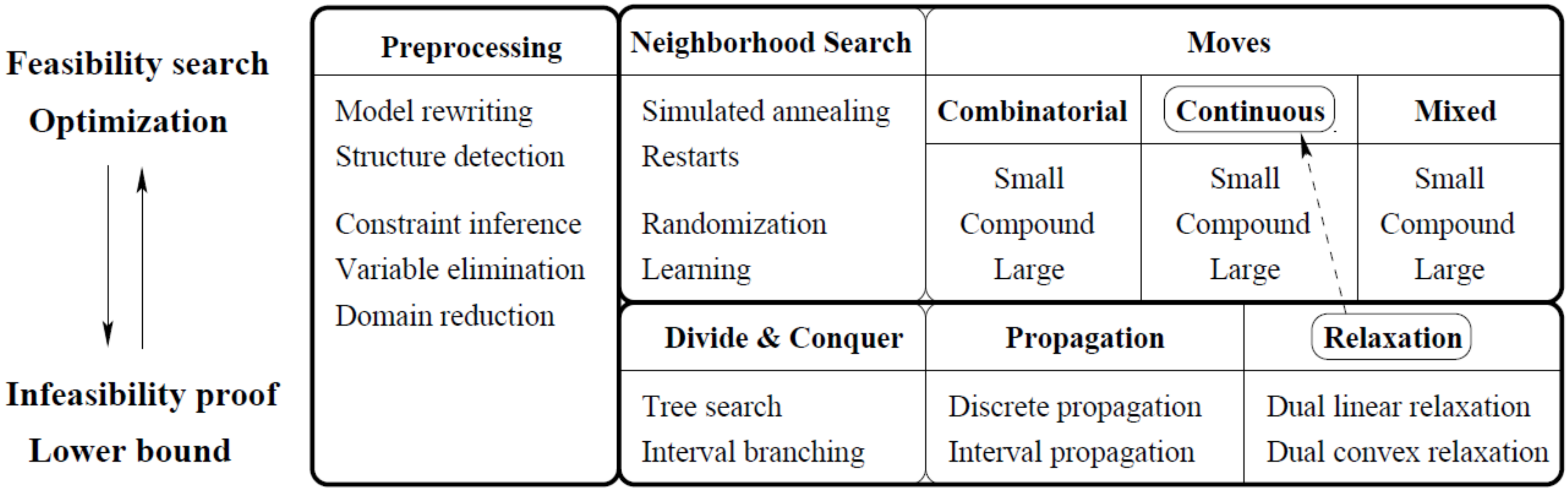
## Neighborhood search as global search strategy

- Speed up the search through fast exploration of small neighborhoods
- Adapt dynamically the explored neighborhood: shrink, enlarge, specialize
- Tree search (MIP, CP): a way to explore large (exponential) neighborhoods
- Complete neighborhood + exact exploration = optimal solution



# Target architecture

Integrating all appropriate optimization techniques
(LS, LP/MIP, CP/SAT, NLP, ...) into one solver
for large-scale mixed-variable non-convex optimization



# LocalSolver 4.0

## Release planned for Christmas

- Binary & continuous decisions
- Improved search for feasible solutions
- Improved preprocessing and inference → lower bounds
- Small & compound-neighborhood moves for continuous/mixed optimization
- First large-neighborhood moves explored through MIP techniques

Applications: supply chain optimization, unit commitment, portfolio optimization, numerical optimization arising in engineering (ex: mechanics)

**http://www.localsolver.com**

# Toward a mathematical programming solver based on neighborhood search

Frédéric Gardi

Innovation 24 & LocalSolver

13th December 2013
ORO Workshop, Nantes