LocalSolver 4.0: novelties and benchmarks

Thierry Benoist Julien Darlay Bertrand Estellon
Frédéric Gardi Romain Megel

www.localsolver.com

LocalSolver 3.1

Solver for combinatorial optimization

- Simple mathematical modeling formalism
- Allows to tackle large-scale combinatorial problems
- Provides good-quality solutions in short running times

Solver based in local search

- Moves based in decisions/constraints hypergraph
- Incremental evaluation: millions of moves per minute
- Adaptive, randomized, parallelized simulated annealing with restarts

Free academic licenses

Commercial licenses from 990 €

P-median

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

```
function model() {
  x[1..N] <- bool() ; // decision : point i is in P iff x[i] = 1

  constraint sum[i in 1..N](x[i]) == P ;

  minDist[i in 1..N] <- min[j in 1..N](x[j] ? Dist[i][j] : InfiniteDist);

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

Nothing else to write: “model & run” approach

- Straightforward mathematical model
- Direct resolution: no tuning

LocalSolver 4.0

LocalSolver 4.0

Mathematical programming solver

- For combinatorial optimization
- **For numerical optimization**
- **For mixed-variable optimization**

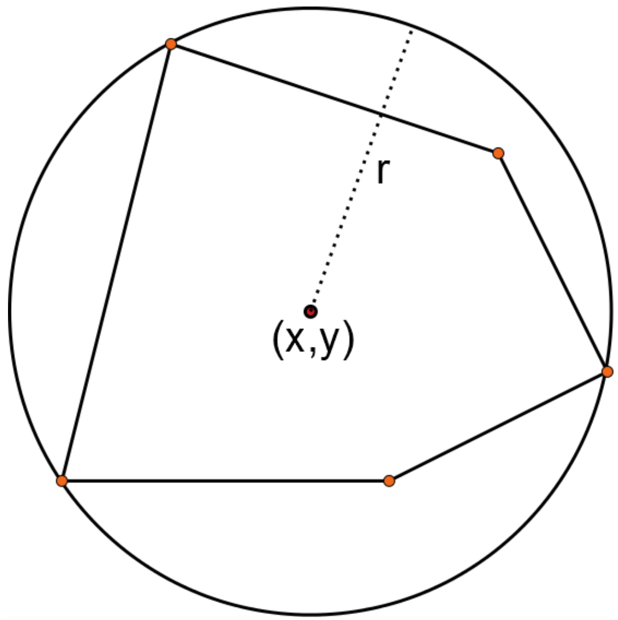
- Provides solutions (upper bounds)
- **Provides lower bounds**
- **Infeasibility gap/proof, optimality gap/proof**

Suited for large-scale non-convex optimization

- Millions of combinatorial and/or continuous variables
- Non-convex constraints and/or objectives
- Short resolution times

Smallest Circle

- Find the circle of minimum radius including a set of points
- Two continuous decisions: ***x*** and ***y***
- The radius ***r***: expression deduced from decisions
- Straightforward quadratic model



Continuous decision

Quadratic expression

```
x <- float(minX, maxX);
y <- float(minY, maxY);

r2 <- max[i in 1..n] (pow(x-coordX[i],2) + pow(y-coordY[i],2));

minimize sqrt(r2);
```

Technical novelties

Hybrid solver

- Mixed-variable local search → unique technology on the market
- Constraint propagation and inference
- Mixed-integer and linear programming

Reinforced preprocessing

- Detection and reformulation of “MIP-oriented” models
- Lower bounds through inference
- Improved infeasibility detection

Large neighborhood search

- Composition of small neighborhoods (combinatorial, continuous, mixed)
- Exploitation of linear relaxation to search in combinatorial spaces
- Exploitation of linear approximation to search in non-convex spaces

Local search

Main idea for combinatorial optimization

- Sequential modification of a small number of decisions
- Maintaining the feasibility of current solution
- Incremental evaluation, generally in $O(1)$ time

→ Small improvement probability but small time and space complexity

In continuous optimization?

- Known under another name: *direct = derivative-free = zeroth-order search*
- Don't use gradients (1^{st} order) nor Hessian (2^{nd} order)
- Ex: Nelder-Mead simplex algorithm
- Mainly used in unconstrained non-convex optimization

Mixed-variable local search

Unification for combinatorial and continuous optimization

- Local search for combinatorial, continuous or mixed-variable optimization
- The same idea than for combinatorial optimization
- The main difficulty: to maintain feasibility

Contradicts the mainstream in optimization

- Guaranty to converge toward an optimum
- Search for the best improvement
- Sophisticated and costly improvement techniques

→ Iteration running in quadratic or even cubic time

LocalSolver

Benchmarks

Combinatorial optimization

Car Sequencing: scheduling cars along an assembly line (minimization)

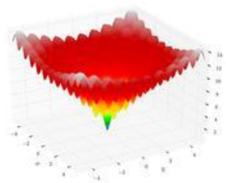
10 sec	100 cars	200 cars	300 cars	400 cars	500 cars
Gurobi 5.5	140	274	X	429	513
LocalSolver 3.1	6	8	9	11	24
LocalSolver 4.0	8	5	8	10	19
60 sec	**100 cars**	**200 cars**	**300 cars**	**400 cars**	**500 cars**
Gurobi 5.5	3	66	1	356	513
LocalSolver 3.1	6	3	3	7	10
LocalSolver 4.0	6	4	3	5	6
600 sec	**100 cars**	**200 cars**	**300 cars**	**400 cars**	**500 cars**
Gurobi 5.5	3	2	*0	1	20
LocalSolver 3.1	6	2	1	2	4
LocalSolver 4.0	4	*0	*0	2	*0

LocalSolver

Unconstrained non-convex optimization

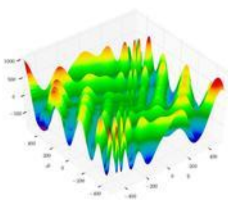
Quasi optimal solutions in seconds on fifty artificial landscapes

Oldenhuis (2009). Test functions for global optimization algorithms. Matlab



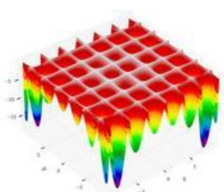
$$f(x,y) = -20\exp\left(-0.2\sqrt{0.5\left(x^2+y^2\right)}\right) - \exp\left(0.5\left(\cos\left(2\pi x\right)+\cos\left(2\pi y\right)\right)\right) + 20 + e.$$

gap (%) < 10^{-6}



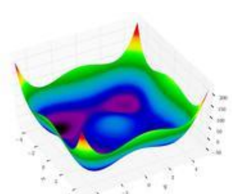
$$f(x,y) = -\left(y+47\right)\sin\left(\sqrt{\left|y+\frac{x}{2}+47\right|}\right) - x\sin\left(\sqrt{\left|x-\left(y+47\right)\right|}\right).$$

gap (%) < 10^{-4}



$$f(x,y) = -\left|\sin\left(x\right)\cos\left(y\right)\exp\left(\left|1-\frac{\sqrt{x^2+y^2}}{\pi}\right|\right)\right|.$$

gap (%) < 10^{-4}



$$f(\boldsymbol{x}) = \frac{\sum_{i=1}^{n} x_i^4 - 16x_i^2 + 5x_i}{2}.$$

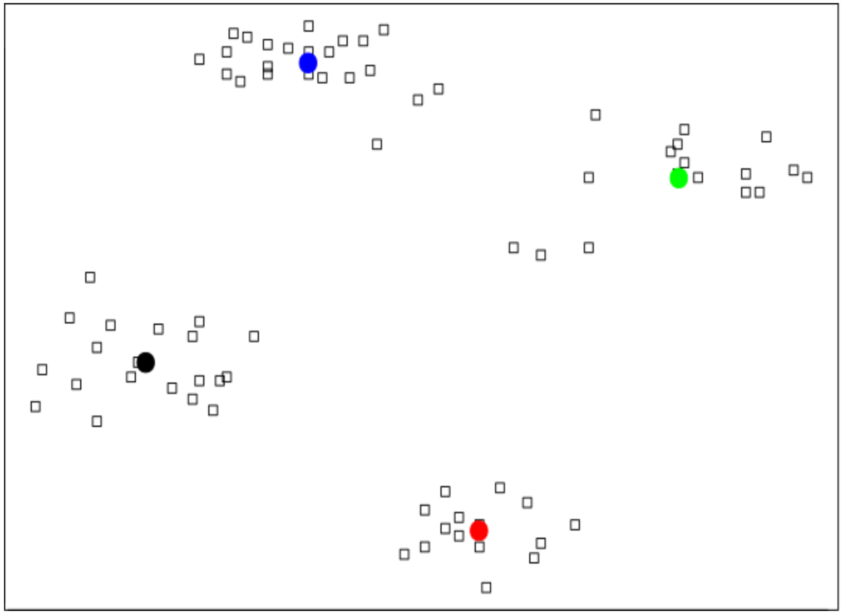
n = 10 → 10000

gap (%) < 10^{-6} → 10^{-1}

Constrained non-convex optimisation

K-means

- Machine learning problem
- Quadratic, NP-hard
- Solutions in 300 sec



Instance	k	OPT*	LS 4.0	GAP
ruspini	2	89337	89337,9	0,00%
	3	51063,4	51063,5	0,00%
	4	12881	12881,1	0,00%
	5	10126,7	10126,8	0,00%
	6	8575,41	8670,86	1,11%
	7	7126,2	7159,13	0,46%
	8	6149,64	6158,26	0,14%
	9	5181,64	5277,11	1,84%
	10	4446,28	4856,98	9,24%
iris	2	152,348	152,369	0,01%
	3	78,8514	78,9412	0,11%
	4	57,2285	57,3556	0,22%
	5	46,4462	46,5363	0,19%
	6	39,04	41,7964	7,06%
	7	34,2982	34,6489	1,02%
	8	29,9889	30,3029	1,05%
	9	27,7861	28,0667	1,01%
	10	25,834	26,0521	0,84%
glass	20	114,646	120,048	4,71%
	30	63,2478	74,1251	17,20%
	40	39,4983	58,3912	47,83%
	50	26,7675	52,4679	96,01%

*[Aloise et al. 2012]

Mixed-variable optimization

Hydro valleys optimization (KEPCO, Japon)

- 4 hydroelectric barrages, from 2 to 6 pumps per barrage
- 30 thermal production units
- Water inputs and electricity prices variable over time
- Long-term horizon: 8000 time steps

→ Large-scale, strongly nonlinear, mixed-variable dynamic system, with hard coupling constraints (1 million of decisions)

Cplex, Xpress, Gurobi: no solution after 3 hours

LocalSolver: quality solution in one minute

Toward an "all-in-one" solver

One solver to tackle all kinds of problems

- Discrete, numerical, or mixed-variable optimization
- From small-scale to large-scale problems
- Best effort to prove infeasibility or optimality
- Able to scale heuristically faced with large problems

One solver offering the best of all optimization techniques

- Local and direct search
- Constraint propagation and inference
- Linear and mixed-integer programming
- Nonlinear programming (convex and non-convex)
- Dynamic programming
- Specific algorithms: paths, trees, flows, matchings, etc.