LocalSolver

mathematical programming by local search

www.localsolver.com

LocalSolver

Why?

Our goals with LocalSolver

A solver aligned with enterprise needs

- Provide high-quality solutions quickly (minutes or seconds)
- Scalable: able to tackle problems with millions of decisions
- Refine optimality gap in a best effort mode

- Easy and light installation, licensing, deployment
- Fully portable: Windows, Linux, Mac OS (x86, x64)
- Full exploitation of many-core architectures (multithreading for free)

For this, we need to change the technology

- Integrating “pure & direct” local search to speed/scale the search
- Computing solutions separately from lower bounds

Our goals with LocalSolver

A solver aligned with practitioner needs

- « Model & Run »
  - Simple mathematical modeling formalism
  - Direct resolution: no need of complex tuning
- Coupled with an innovative modeling/scripting language (LSP)
- Easy integration through object-oriented C++, Java, .NET APIs

- Competitive prices: lower than leading MIP solvers
- Dedicated support by a reactive and expert team, even for modeling issues
- Free for academics

LocalSolver

Quick tour

Classical knapsack

8 items to pack in a bag: maximize the total value of items while not exceeding a total weight of 102 kg

```
function model() {
  // 0–1 decisions
  x_0 <- bool(); x_1 <- bool(); x_2 <- bool(); x_3 <- bool();
  x_4 <- bool(); x_5 <- bool(); x_6 <- bool(); x_7 <- bool();

  // weight constraint
  bagWeight <- 10*x_0+ 60*x_1+ 30*x_2+ 40*x_3+ 30*x_4+ 20*x_5+ 20*x_6+ 2*x_7;
  constraint bagWeight <= 102;

  // maximize value
  bagValue <- 1*x_0+ 10*x_1+ 15*x_2+ 40*x_3+ 60*x_4+ 90*x_5+ 100*x_6+ 15*x_7;
  maximize bagValue;
}
```

Binary decisions

Integer or continuous expressions

You write the model: nothing else to do!

declarative approach = model & run

LocalSolver

Multiobjective nonlinear knapsack

```
function model() {
  // 0-1 decisions
  x[1..nbItems] <- bool();

 // weight constraint
  bagWeight <- sum[i in 1..nbItems]( weights[i] * x[i] );
  constraint bagWeight <= 102;

  // maximize value
  bagValue <- sum[i in 1..nbItems]( values[i] * x[i] );
  maximize bagValue;

  // secondary objective: minimize the product of minimum and maximum values
  bagMinValue <- min[i in 1..nbItems]( x[i] ? values[i]  : 1000 );
  bagMaxValue <- max[i in 1..nbItems]( x[i] ? values[i]  : 0 );
  minimize bagMinValue * bagMaxValue;
}
```

Nonlinear operators: prod, min, max, and, or, if-then-else, …

Lexicographic objectives

Mathematical operators

Arithmetic			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	>

LocalSolver

Let’s go inside

Our idea

Using local search as global search strategy

- Local search means “neighborhood search”
- To speed up the search with fast small-neighborhood explorations
- To scale by adapting the kind and size of neighborhoods explored
- Instead of embedding LS into TS, we view TS as a way to explore exponential-size neighborhoods

Seems to be a small change, but…

Future Architects & Shikishima Baking Co:

“When do you think that a MIP solver would be able to tackle problems with 20 million variables including 3 million binaries? LocalSolver tackles it today!”

Small-neighborhood moves

Volume 8 · Number 1 · 2010

4OR

A Quarterly Journal of Operations Research

Springer

T. Benoist, B. Estellon, F. Gardi, R. Megel, K. Nouioua (2011). LocalSolver 1.x: a black-box local-search solver for 0-1 programming. *4OR, A Quarterly Journal of Operations Research* 9(3), pp. 299-316.

http://www.localsolver.com/technology.html

LocalSolver

Applications & Benchmarks

Panorama

FUJITSU	Supply Chain Optimization
MSI Management Solutions Integration	Workforce planning
TF1 PUBLICITE	TV Media Planning
AIR LIQUIDE	Logistic clustering
ETDE	Street lighting maintenance planning
Bouygues Telecom	Network deployment planning
SIEMENS	Energy optimization for tramway lines
EDF	Placement of nuclear fuel assemblies in pools
整数計画 0-1 Integer Inc.	Painting shop scheduling
ARMEE DE TERRE	Transportation of equipment

Car sequencing in Renault's plants

Some instances are public. This problem was submitted as ROADEF Challenge in 2005: http://challenge.roadef.org/2005/en

Large instances

- 1300 cars to sequence → 400 000 binary decisions

Instance 022_EP_ENP_RAF_S22_J1

- Small instance: 80,000 variables, including 44,000 binary decisions
- State of the art: 3,109 obtained by a specific local search algorithm
- Best lower bound: 3,103

Minimization

Results

- Gurobi 5.5: 3.116647e+07 in 10 min | 25,197 in 1 hour
- LocalSolver 3.1: 3,478 in 10 sec | 3,118 in 10 min

LocalSolver

Google machine scheduling

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

Running time limited to 5 minutes on a standard computer (4 GB RAM).

Using a 100-line model, LocalSolver 2.0 was the sole general-purpose solver to be qualified for the final tour of the Challenge, ranked 25th over 82 teams from all around the world.

LocalSolver tackles models with 10 M variables.

Totally out of scope of MIP, CP, SAT solvers.

MIPLIB

Some results obtained on the hardest MIPLIB instances

- Lower objective is better
- 5 minutes for both LocalSolver and MIP
- MIP-oriented models: not suitable for LocalSolver

Minimization

Problem	Variables	LS 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

Roadmap

A new-generation 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."

LocalSolver 4.0

Planned for the end of 2013

- Binary + **continuous** decisions
- Stronger lower bounds through constraint propagation and linear relaxation

→ Our first step toward **large-scale mixed-variable non-convex programming**

But do not wait, try **LocalSolver 3.1**. We are ready to support you!

Meet us on our OR 2013 booth for more info

http://www.localsolver.com

LocalSolver

mathematical programming by local search

www.localsolver.com