Local search for mixed-integer nonlinear optimization: methodology and applications

Frédéric Gardi

Bouygues e-lab, Paris
fgardi@bouygues.com

Joint work with T. Benoist, B. Estellon, A. Jeanjean, K. Nouioua
Bouygues e-lab & LIF UMR 6166 - Université Aix-Marseille 2

Theory says “no, no, no”

EDF’s unit commitment problem is a very large-scale mixed-integer nonlinear (MINL) problem.

Example of very large-scale MINLP: scheduling outages of EDF nuclear power plants (ROADEF Challenge 2010) involves 10^9 decision variables, including 10^7 boolean variables.

MINLP $\leftarrow$ MILP $\leftarrow$ IP $\leftarrow$ 0-1 IP

Thus, large-scale MINLP are extremely hard to solve:

- Theoretically: NP-complete, non-approximable, …

- Practically: proving optimum (= finding feasible solution) in reasonable running times (less than one century :-) is impossible.

But no matter

What are the needs in business and industry?

1) Clients have optimization problems, and rarely satisfaction problems.

“No solution found” is rarely an acceptable answer for users. Thus, once the model is well stated, finding a feasible solution should be easy.

→ Goal programming (soft constraints, etc.)

2) Optimal solution is not what clients really want.

- *Proof* of optimality is much less what they want
- They want a nice software providing good solutions quickly
- Better and faster than before having your software
- Then, they could be interested in optimality gap...

→ Don’t be focused on optimality

So… Tree Search?

Mixed-integer programming techniques (B&B, B&C, BCP) are:

- Designed for *proving optimality*

- Not designed to find *feasible solutions*

MIP techniques are powerful for tackling small instances (1,000 binaries). When relaxation is good, medium instances (10,000 to 100,000 binaries) can be tractable.

<u>Our conviction</u>: pure tree-search techniques will remain powerless for solving very large-scale combinatorial problems.

So… Tree Search?

Why?

1) Relaxation is often useless but costs a lot in efficiency. So why losing running time to enumerate *partial* solutions?

2) Why an *incomplete* tree search should be better than a local search? Moreover, tree search is not really suited for exploring randomly (without bias) a search space.

Facts:

State-of-the-art solvers integrate more and more local-search ingredients in B&B (Local Branching, Relaxation Induced Neighborhood Search).

TSP records:
- B&C [Applegate, Bixby, Cook, Chvátal, etc.]: 85,900 cities
- LS [Helsgaun]: 1,904,711 cities (World TSP), and until 10,000,000 cities

So… Local Search!

LS paradigm: iterative improvements by applying (local) transformations on the current solution.

Performance (efficiency and effectiveness) not well understood today. Rare theoretical results: LS is very bad… in the worst case!

But a renowned practical solution for solving hard practical problems:

good-quality solutions with short running times (minutes)

Then, the common vision of what is LS can be summarized as:

LS = metaheuristics = cooking

Local Search is not cooking

Our vision:

LS = incomplete & non deterministic search

Consequently, LS must be:

1) Pure & direct : no decomposition, no hybridization.

2) Highly randomized : any decision taken is randomized.

3) **Aggressive** : millions of feasible solutions explored.

Methodology

LS = randomized moves + incremental computation

Therefore, our work is concentrated on:

- Designing moves enabling an effective exploration of search space.
- Speeding up the evaluation of moves (algorithm engineering).

"Incremental computation", what's that?

Given a solution S to an optimization problem and a transformation Δ: $S \rightarrow S'$. Denote by $|\Delta|$ the length of "changes" between S and S'.

Question: design an $O(|\Delta|)$-time algorithm to compute the cost of S'.

Methodology

Methodology developed during the last 10 years while solving several combinatorial optimization problems with high economic stakes:

- Car sequencing (Renault*, ROADEF 2005 Challenge)
- Workforce and task scheduling (France Telecom, 2007 Challenge)
- Media planning (TF1*, 2011)

Extended to mixed-variable optimization:

- Inventory routing (Air Liquide*, 2008): MILP
- Resource scheduling for mass transportation (By Cons*, 2009) : MILP
- Nuclear plant maintenance planning (EDF, 2010 Challenge): MINLP

Methodology for MINLP

Local Search is rarely used in the context of MINL optimization.

Main principle: combinatorial and continuous parts are treated together

→ Combinatorial and continuous decisions are simultaneously modified by a move during the search

Main difficulty: solving efficiently the continuous subproblem

Practical solution: an incremental randomized combinatorial algorithm for solving approximately but very efficiently the continuous subproblem:

- From 1,000 to 10,000 times faster than using LP approximations
- Near-optimal solutions found in practice

Methodology

Work surrounded by an important effort in software engineering for ensuring reliability of this critical evaluation machinery:

- programming with assertions

- checkers for incremental structures

- continuous refactoring

- CPU & memory profiling

→ Quest of high performance

Note: we have relaxed this effort the last week of EDF Challenge in order to concentrate our work on some improving technical features, and we have crashed…

Results on EDF Challenge

Ranked 1st over 44 teams on benchmark A (qualification)
Ranked 1st over 16 teams on benchmark B (final)
We fall to the 8th rank due to a bug on hidden benchmark X :-(

Table 1 Official ranking of solution approaches on instances B.

instance	team		technique	average gap
1	S22	Gardi-Nouioua	LS	0.23 %
2	S24	Kuipers-Peekstok	LS	0.24 %
3	S23	Wolfler Calvo et al.	MIP	1.46 %
4	J06	Kjeldsen et al.	LS	2.13 %
5	S21	Jost et al.	MIP	4.71 %
6	J08	Steiner et al.	ACO/LS	11.99 %
7	S04	Dell'Amico	LNS/MIP	13.02 %
8	S14	Weber et al.	LS	14.52 %
9	S08	Hurkens	MIP	29.17 %
10	S17	Soumis et al.	MIP	35.10 %
11	S16	Cambazard et al.	LNS/CP	55.57 %
12	J05	Ahlroth et al.	LNS	106.36 %
13	S10	Petersen et al.	MIP	1726.61 %
14	J16	Heinz	CP/MIP	1850.71 %
15	S11	Nattero et al.	MIP/LS	2332.98 %
16	S25	Gavranovic et al.	CP	3458.06 %

More results

Our recent works on local search for mixed-integer optimization:

T. Benoist, B. Estellon, F. Gardi, A. Jeanjean (2011). Randomized local search for real-life inventory routing. *Transportation Science* 45(3), pp. 381-398.

F. Gardi, K. Nouioua (2011). Local search for mixed-integer nonlinear optimization: a methodology and an application. In *Proceedings of EvoCOP 2011*, *LNCS* 6622, pp. 167-178. Springer.

A. Jeanjean (2011). Local search for mixed-variable optimization: methodology and industrial applications. PhD Thesis (Bouygues e-lab & LIX).

Web : http://pageperso.lif.univ-mrs.fr/~frederic.gardi

Conclusion

LS = practical solution for practical problems
LS = good-quality solutions within short running times

But LS is not cooking. Our vision:

LS = incomplete & non deterministic search
LS = randomized moves + incremental computation (= run fast)

→ Less “Maths” (analytical), more “Computer Science” (algorithmic)
→ A lot of software and algorithm engineering

So why not ?

LocalSolver

Based on these methodology and experiences, we start developing in 2007 a black-box solver entirely based on local search for combinatorial optimization.

LocalSolver is able to tackle large-scale real-life 0-1 programs (with nonlinear constraints and objectives): 10 millions of binary variables.

www.localsolver.com

Exploited in Bouygues Group (TF1, ETDE, Colas), but also outside (Eurodecision). Commercial version (2.0) prepared for early 2012.

Bouygues e-lab: T. Benoist, *J. Darlay*, F. Gardi, *R. Megel*
LIF Aix-Marseille: B. Estellon, K. Nouioua