High-performance local search for task scheduling with human resource allocation

Frédéric GARDI

Bouygues e-lab, Paris
fgardi@bouygues.com

Bertrand ESTELLON Karim NOUIOUA

Laboratoire d'Informatique Fondamentale,
Faculté des Sciences de Luminy, Marseille

{ estellon, nouioua }@lif.univ-mrs.fr

03/09/2009, Brussels

Local search

Paradigm: to improve iteratively a solution by exploring a neighborhood of this solution.

Neighborhoods are induced by (local) transformations applied to the solution, that is, by modifying some decision variables in the solution: local search = neighborhood exploration

Today, we observe a tendency to confuse local search and metaheuristics, leading to neglect two major ingredients in the design of local search algorithms:

- the definition of moves
- the algorithmic machinery (for evaluating moves)

High performance

In combinatorial optimization, the definition (or goal) of high performance could be : providing solutions of better quality (for larger instances) with shorter running times (and more generally, using less resources).

Why ?

Because this is the main demand of people having some needs in optimization (engineers, analysts, operational teams, etc.).

A challenge is to meet this growing demand for performance facing to physical constraints (hardware), economical constraints (budget), ecological constraints (green IT), etc.

High performance

Another confusion : high performance is not synonym of parallel computing. A reference on this subject:

B.M.E. Moret, D.A. Bader, T. Warnow (2002). High-performance algorithm engineering for computational phylogenetics. *Journal of Supercomputing* 22(1), pp. 99-111.

Before parallelization issues, the performance must be sequential. Before hardware issues, the performance must be algorithmic.

A mean (our credo!): experimental algorithmics (or algorithm engineering) mixing foundations of computer science (complexity theory) and practical aspects of implementation (software engineering).

Methodology

A methodology was derived from our experiences for designing and engineering high-performance local-search algorithms. We do not claim that the recipe is new.

The methodology (and the resulting software) is composed of three layers:

a) search strategy & (meta)heuristics

b) moves & neighborhoods

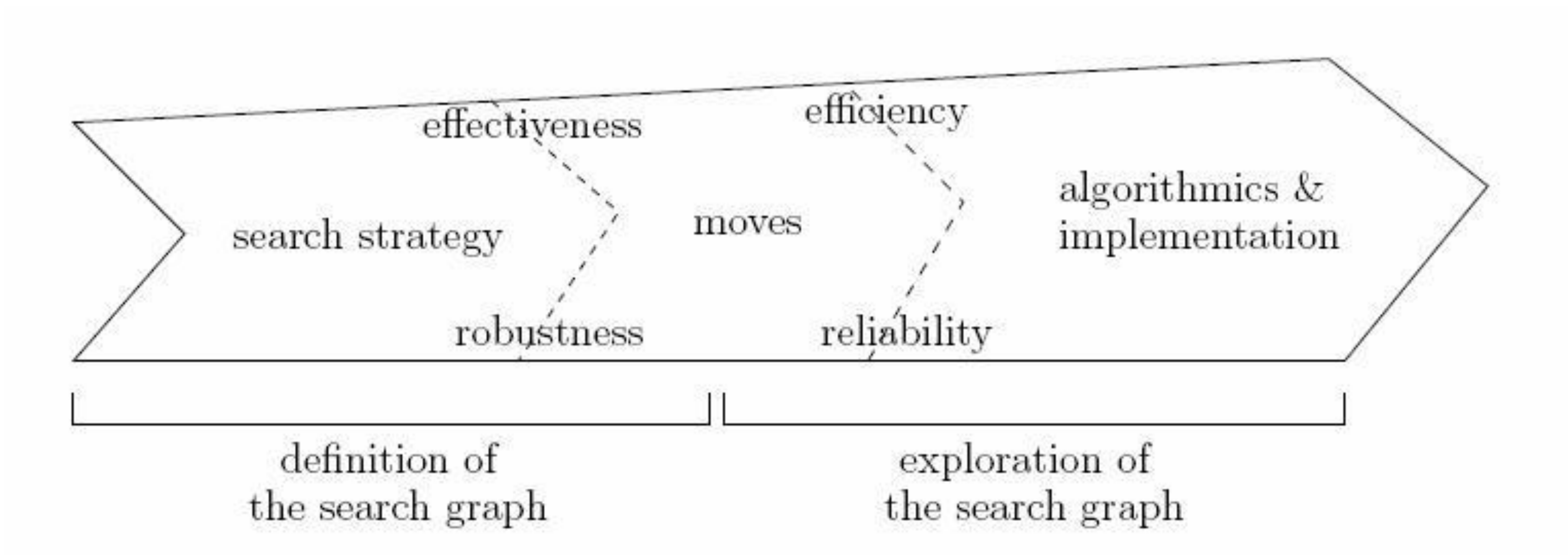
c) algorithms & implementation

We claim that the performance of a local-search heuristic depends equally on the careful treatment of each layer. But we observe that the working time spent to treat each layer follows the rule:

a : 10 %   b : 30 %   c : 60 %

Methodology

These 3 layers cover the two fundamental aspects of local search :
- definition of the search space (density + connectivity)
- exploration of the search space



Application

ROADEF 2007 Challenge: task scheduling with human resource allocation (real-life problem from France Telecom)

n interventions, a set of available technicians each day, d skill domains, l skill levels in each domain.

Each intervention requires a number of technicians with level at least l in each domain d. Each technician has a level l in each domain d.

Interventions can be assigned to a set of technicians one day if:

- the number of technicians in each level l and domain d is greater than the one required by each intervention (sequential execution)

- the sum of durations of interventions is lower than H

Application

Extensions :

- precedence between interventions (sparse)
- budget *B* allowing to subcontract interventions

Objective : minimizing the makespan of the schedule

Scale : 800 interventions, 150 technicians, 40 domains and 7 levels of skill, resulting schedules with 60 days

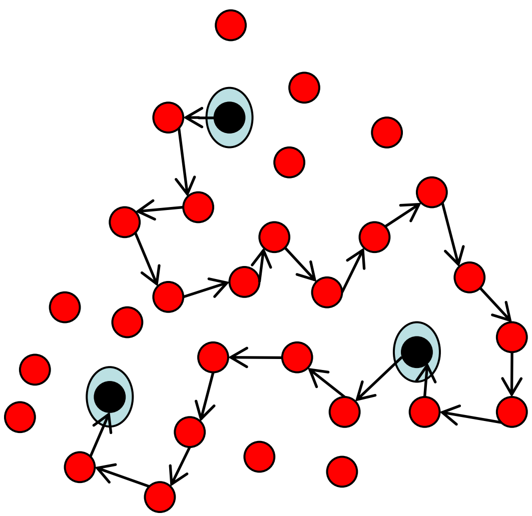
Resources : running time limited to 20 minutes per instance on a standard computer (AMD Athlon64 1.8 GHz, 1 Mo L2, 1 Go RAM)

Search strategy

1) Define the search space (surrogate solution space)

→ increase the density of the search space

Relaxing business constraints and/or using surrogate cost function allows to increase the search space's density (and connectivity)



Non admissible solutions (red points)

= bridging points for local search :

(+) reachable by local moves

(+) increase diversification

(–) must converge toward admissible solutions (black points)

Search strategy

Example on France Telecom's problem:

skill constraints on technician teams are relaxed

For each intervention, one violation is counted if:

- the team of technicians to which it is assigned does not have enough skills to perform it
- its ending time is greater than CURRENT_DEADLINE

Objective : minimizing the number of violations (by local search)

When no violation remains, an improving admissible solution is found. The process is iterated by setting

$$\text{CURRENT\_DEADLINE} \leftarrow \text{CURRENT\_DEADLINE} - 1$$

Search strategy & Moves

2) Define how local search walks into the search space

→ increase the connectivity of the search space

If 1) treated carefully, prefer simplicity (at least initially):

- first-improvement descent
- stochastic selection of moves (stochastic ≠ uniform)

The pool of moves ensure the search space's connectivity:

more moves → greater connectivity → larger diversification

Generic moves → diversification

Specific moves → intensification (acceleration of convergence)

Reminder: density + connectivity → convergence

Moves

On France Telecom's problem

Pool of 31 moves derived from 8 basic transformations:

- move technician to another team in a day
- swap technicians of two different teams in a day
- move intervention in another day ("inter days")
- move intervention in another team of the day ("intra day")
- move intervention in the team schedule ("intra team")
- swap two interventions "inter days"
- swap two interventions "intra day"
- swap two interventions "intra team"

The stochastic selection of moves follows a specific distribution determined experimentally (by hand).

Moves

Main derivations: for choosing technicians and interventions to which a transformation is applied

- randomly
- randomly among days with (interventions inducing) violations
- randomly among teams with violations

Specific derivations: for dealing with extensions of the problem

- precedence: swap interventions A and B if start(A) $\leq$ start(B) and B has more successors than A in the DAG
- subcontracting: swap a scheduled intervention causing violations with a subcontracted intervention

Algorithms

Local search is an incomplete search technique: its performance depends strongly on the number of solutions explored within the time limit.

algorithms = engine of local search

3 crucial routines for each move : evaluate, commit, rollback

1) incremental algorithms relying on special data structures, exploiting invariants of moves → (high-level) efficiency

2) careful implementation (cache-aware programming, CPU & RAM profiling) → (low-level) efficiency

3) programming with assertions, data structures checked at each iteration in debug mode (checkers) → correctness & reliability

Algorithms

France Telecom: evaluating skills provided by technicians versus skills required by interventions assigned to a team.

Skill matrix (i,j) with positive entries, non increasing in columns.
Problem: decide if $T(i,j) \geq I(i,j)$ for all (i,j). Worst-case: $O(dl)$ time.

$T(i,j)$

$l = 3$	1	0	0	1
	3	1	0	2
	3	2	0	2
$l = 0$	3	3	0	2

$I(i,j)$

0	0	0	1
3	1	0	2
3	2	0	3
3	3	0	3

if $T(i,j) < I(i,j)$, then $T(i,j') \leq T(i,j) < I(i,j) = I(i,j')$ with $j' < j$

Algorithms

1	0	0	1
3	1	0	2
3	2	0	2
3	3	0	2

23	10	6	0	7

0	0	0	1
3	1	0	2
3	2	0	3
3	3	0	3

9	6	0	9	24

Stop evaluation earlier: 3 tests in cascade $O(1) \longrightarrow O(d) \longrightarrow O(dl)$

experimental algorithmics = practical efficiency and not only theoretical worst case

Challenge results

- ≈ 120 man-days
- 12000 lines of ISO C99 code

- runs with less than 10 Mo of RAM
- 1.5 million moves/sec, 2 billion moves over 20 min
- acceptance rate of moves between 5 % and 50 %

- average gain of 30 % compared to FT solutions
- best solutions of the challenge for 13 instances over 30
- far from 7.3 % of the best solution on average

- 2nd Senior over 35 participating teams from 10 countries

  1st : Hurkens (Netherlands)

  3rd : Cordeau, Laporte, Pasin, Ropke (Canada)