LIGHT ROBUSTNESS

Matteo Fischetti, Michele Monaci

DEI, University of Padova

1st ARRIVAL Annual Workshop and Review Meeting, Utrecht, April 19, 2007

Work supported by the Future and Emerging Technologies unit of the EC (IST priority),
under contract no. FP6-021235-2 (project ARRIVAL)

Motivation

- Basic assumption in combinatorial optimization:

The exact value of all input data is known in advance

- This assumption is often violated in practical (real-world) applications

- $\Rightarrow$ The optimal solution found using nominal values of the parameters can be **suboptimal** or even **infeasible**

- $\Rightarrow$ Small uncertainty in the data can make the usual optimal solutions completely meaningless from a practical point of view

Dealing with Uncertainty

Consider the **nominal LP** (or MIP) $\min \{cx : Ax \leq b,\ x \geq 0\}$

Stochastic Programming

Find a solution that is optimal by considering possible **recourse** variables y^k implementing corrective actions after a random **scenario** $k \in K$ has taken place

$$
\begin{aligned}
\min\ cx + \sum_{k \in K} p_k (d^k y^k) \\
Ax \leq b, \quad x \geq 0 \\
T^k x + W^k y^k = h^k, \quad y^k \geq 0, \qquad k \in K
\end{aligned}
$$

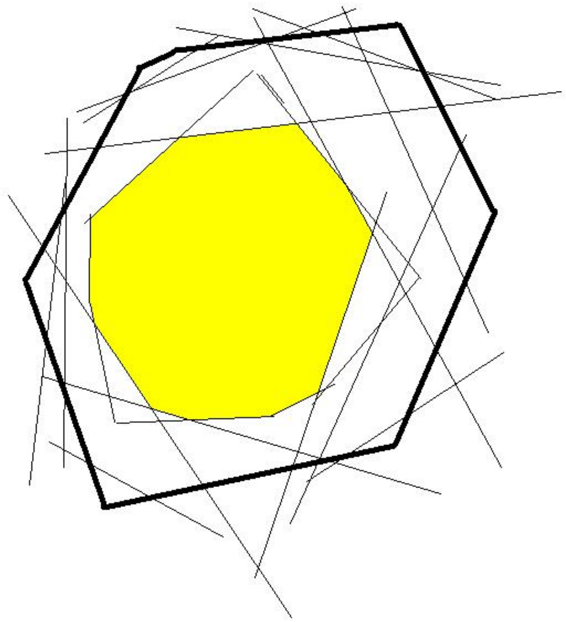
⇒ (+) Does not restrict the original solution space, just penalizes the corrective actions needed to face a certain scenario

⇒ (-) Requires the knowledge of the probability/main features of the various scenarios

⇒ (-) Huge LP's to be solved (through clever decomposition techniques)

Robust Optimization

- Uncertainty is associated with **hard constraints** restricting the solution space



- Find a solution that is still feasible for worst-case parameters chosen in a certain **uncertainty domain**
- $\Rightarrow$ (+) Easy way to model uncertainty
- $\Rightarrow$ (-) The solution can be **overconservative**, hence be quite bad in terms of efficiency (actually, a feasible solution may not exist)

Literature on Robust Optimization

- Soyster (1973)
  - First attempt to handle data uncertainty through mathematical models
  - Definition of the robust counterpart of an uncertain linear program
  - Uncertain linear program of the form

$$\min \ \{cx | \sum_{j=1}^{n} A_j \ x_j \leq b, \ \forall A_j \in K_j, j = 1, \cdots, n\}$$

    where K_j are convex sets associated with "column-wise" uncertainty
  - Very conservative model
- Ben-Tal and Nemirovski (1998-2000)
  - Less conservative models by considering ellipsoidal uncertainty
  - Nonlinear (convex) models $\rightarrow$ computationally hard problems
  - Similar results by El-Ghaoui et al. (1998-2000)

Literature on Robust Optimization

Bertsimas and Sim (2001-2004)

- W.l.o.g., assume the RHS b and cost vector c are not affected by uncertainty, and recall that $x \geq 0$ is required
- Each entry of matrix A takes values in the interval $[a_{ij}, a_{ij} + \hat{a}_{ij}]$
- **Assumption**: for each constraint i, at most Γ_i coeff.s can change
- Each constraint $\sum_{j=1}^{n} a_{ij}\ x_j \leq b_i$ in the nominal LP is replaced by

$$\sum_{j=1}^{n} a_{ij}\ x_j + \beta(x, \Gamma_i) \leq b_i \tag{1}$$

where

$$\beta(x, \Gamma_i) = \max_{S \subseteq \{1,\cdots,n\}:|S| \leq \Gamma_i} \sum_{j \in S} \hat{a}_{ij} x_j \tag{2}$$

- The role of the **protection-level** parameter Γ_i (here assumed to be integer to simplify notation) is to adjust the robustness of the solution:
- $\Gamma_i = 0 \rightarrow$ constraint i equivalent to the nominal one
- $\Gamma_i = n \rightarrow$ conservative method by Soyster

- Using LP duality, a **compact formulation** for the robust LP is

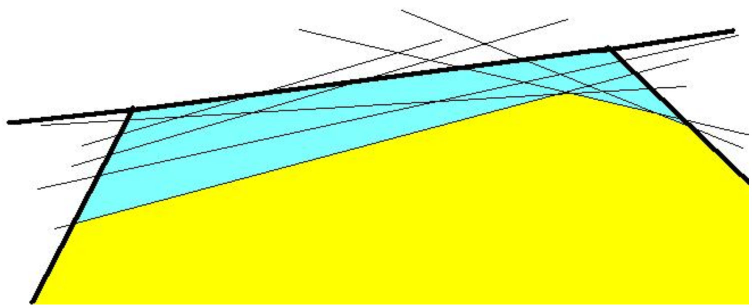
$$
\text{(BS)} \qquad \min \sum_{j=1}^{n} c_j \, x_j
$$

$$
\begin{aligned}
\sum_{j=1}^{n} a_{ij}\, x_j + \Gamma_i z_i + \sum_{j=1}^{n} p_{ij} \le b_i, &\quad i = 1, \cdots, m, \\
z_i + p_{ij} \ge \hat{a}_{ij}, &\quad i = 1, \cdots, m, j = 1, \cdots, n, \\
z_i \ge 0, &\quad i = 1, \cdots, m, \\
p_{ij} \ge 0, &\quad i = 1, \cdots, m, j = 1, \cdots, n, \\
x_j \ge 0, &\quad j = 1, \cdots, n
\end{aligned}
$$

- The robust counterpart of an uncertain problem has the same complexity/approximability of the nominal problem
- The compact formulation of Bertsimas and Sim extends easily to (M)ILP's—in fact, a MIP is just a sequence of LP's

Robustness through cutting planes

- Alternative approach: **cutting plane method** based on cut separation



- **Separation problem**: given x^*, find $S \subseteq \{1, \cdots, n\}$ (if any) such that $|S| \leq \Gamma_i$ and

$$\sum_{j=1}^{n} a_{ij} x_j^* + \sum_{j \in S} \hat{a}_{ij} x_j^* > b_i$$

- Separation problem solvable in $O(n)$ time
- **Important extensions**: more detailed (and realistic) descriptions of the uncertainty domain can be handled within the cut separation procedure (possibly involving integrality of the worst-case parameters...)

Computational experiments: knapsack problem

- Nominal Problem: Knapsack problems as in the BS paper, with 10% uncertainty on the nominal LHS coefficients
- $n \in \{200, 400, 600, 800, 1000\}$
- weights w_j randomly generated in $[16, 77]$
- profits p_j randomly generated in $[20, 29]$
- $c = 4000$
- Times expressed in CPU seconds of a PC AMD Athlon 4200+

Instance	Nominal KP		$\Gamma = 5$			$\Gamma = 20$			$\Gamma = 50$		
n	z	T	z	BS	B&C	z	BS	B&C	z	BS	B&C
200	8463	0.00	8456	0.02	0.02	8442	0.02	0.00	8409	0.03	0.02
400	10417	0.05	10406	1.05	0.85	10377	0.05	0.04	10314	0.72	8.13
600	11384	0.02	11371	0.65	1.15	11336	0.06	1.42	11262	0.15	10.91
800	11982	0.02	11971	0.04	2.03	11932	0.12	7.53	11853	20.04	43.61
1000	12361	0.02	12348	0.40	0.14	12307	7.70	64.83	12225	14.14	44.75

- For KP problems, the BS compact formulation is faster than Branch-and-Cut
- BUT ...

Computational experiments: set covering problem

- Set covering instances from the literature (scp41, scp42, and scp43) with $n = 1,000$ and $m = 200$
- Uncertainty: $\Gamma_i = 1$ for each row (meaning that at most one entry can go from 1 to 0 in each row of A).

Instance	$\Gamma = 0$		$\Gamma = 1$		
	z^*	T	z	BS	B&C
scp41	429	0.02	1148	3,278.24	1,463.88
scp42	512	0.04	1205	23,251.23	5,393.41
scp43	516	0.03	1213	211,842.50	12,794.06

For set covering instances:

- Branch-and-Cut is faster than the BS compact formulation
- Even more effective (ad hoc) models exist...

Robustness for problems with a strong structure

Many Combinatorial Optimization Problems have a **strong combinatorial structure**, e.g.

Set Covering Problem

- covering constraint for row $i = 1, \cdots, m$:

$$\sum_{j=1}^{n} a_{ij}\, x_j \geq 1 \qquad\qquad \text{with } a_{ij} \in \{0, 1\}$$

- Robust counterpart:
  - What does it mean that a coefficient a_{ij} can change?
  - Do values like 0.9 or 1.1 make sense?
  - It makes sense to assume that (at most) Γ_i entries in row i go from 1 to 0
  - $\Rightarrow$ Row i has to be covered by (at least) $\Gamma_i + 1$ different columns

- Conclusions:
  - the classical scheme is either **meaningless** or **too conservative**
  - $\Rightarrow$ robust solutions can be useless or very inefficient

Train Timetabling Problem

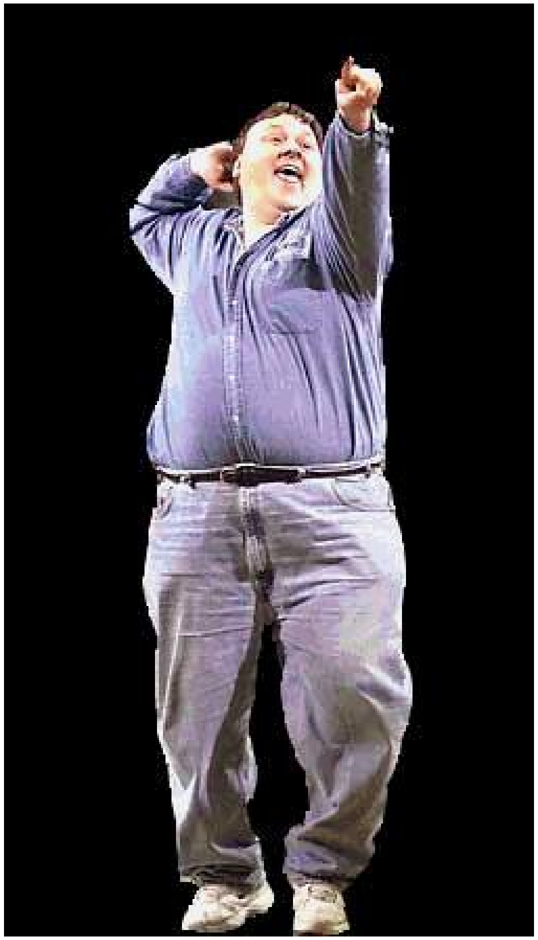
- MIP models where a minimum distance in time between two events i and j (departures/arrivals of trains) is imposed:

$$t_i - t_j \geq \Delta_{ij}$$

- Robust counterpart:
  - The only nonstructural coefficient in each row is the required minimum time-distance Δ_{ij}
  - Increasing Δ_{ij} to $\Delta_{ij} + \hat{\Delta}_{ij}$ is a way to cope with train delays ...
  - ... but the BS worst-case parameter setting occurs when **all** coefficients Δ_{ij} go to their upper bound $\Delta_{ij} + \hat{\Delta}_{ij}$

- Conclusions:
  - a robust solution is just obtained from the nominal problem by replacing all Δ_{ij} with $\Delta_{ij} + \hat{\Delta}_{ij}$
  - $\Rightarrow$ **highly inefficient** solutions (and often impossible to find due to train capacity constraints)

From Robustness to Light Robustness



Robustness (left) vs. Light Robustness (right)

Light Robustness

Idea: robustness is about putting enough slacks on the constraints $\Rightarrow$ introduce explicit *unsatisfied slack* var.s γ_i (kind of **second-order recourse** var.s) to be minimized, and fix an upper bound on the acceptable solution cost (so as to limit inefficiency)

$$(\mathtt{LR}) \qquad z_{LR} = \min \sum_{i=1}^{m} \gamma_i^2$$

$$Ax \leq b, \qquad x \geq 0, \qquad cx \leq z^*(1+\delta)$$

$$\sum_{j=1}^{n} a_{ij}\, x_j + \beta(x, \Gamma_i) - \gamma_i \leq b_i, \qquad i = 1, \cdots, m,$$

- The role of parameter δ is to balance the quality (optimality) and the feasibility (robustness) of the solution
- $\delta = 0 \longrightarrow$ nominal problem (of value z^*)
- $\delta = \infty \longrightarrow$ robustness in the spirit of Bertsimas and Sim
- **Just a thin idea, but ... maybe it works?!**

Computational experiments: light robustness for set covering problems

- Set covering instances from the literature (scp41, scp42, and scp43)
- $n = 1000$ and $m = 200$
- Uncertainty: $\Gamma = 1$ for each row
- Loss of optimality bound: $\delta \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$

Instance	Nominal Problem		Light Robustness										BS
			$\delta = 0.1$		$\delta = 0.2$		$\delta = 0.3$		$\delta = 0.4$		$\delta = 0.5$		
	z^*	T	z	z_{LR}	z	z_{LR}	z	z_{LR}	z	z_{LR}	z	z_{LR}	z
scp41	429	0.02	471	94	514	79	557	67	600	57	640	49	1,148
scp42	512	0.04	563	87	613	68	664	56	716	46	768	37	1,205
scp43	516	0.03	567	86	619	68	670	55	721	47	759	40	1,213

Computational experiments: light robustness for set covering problems

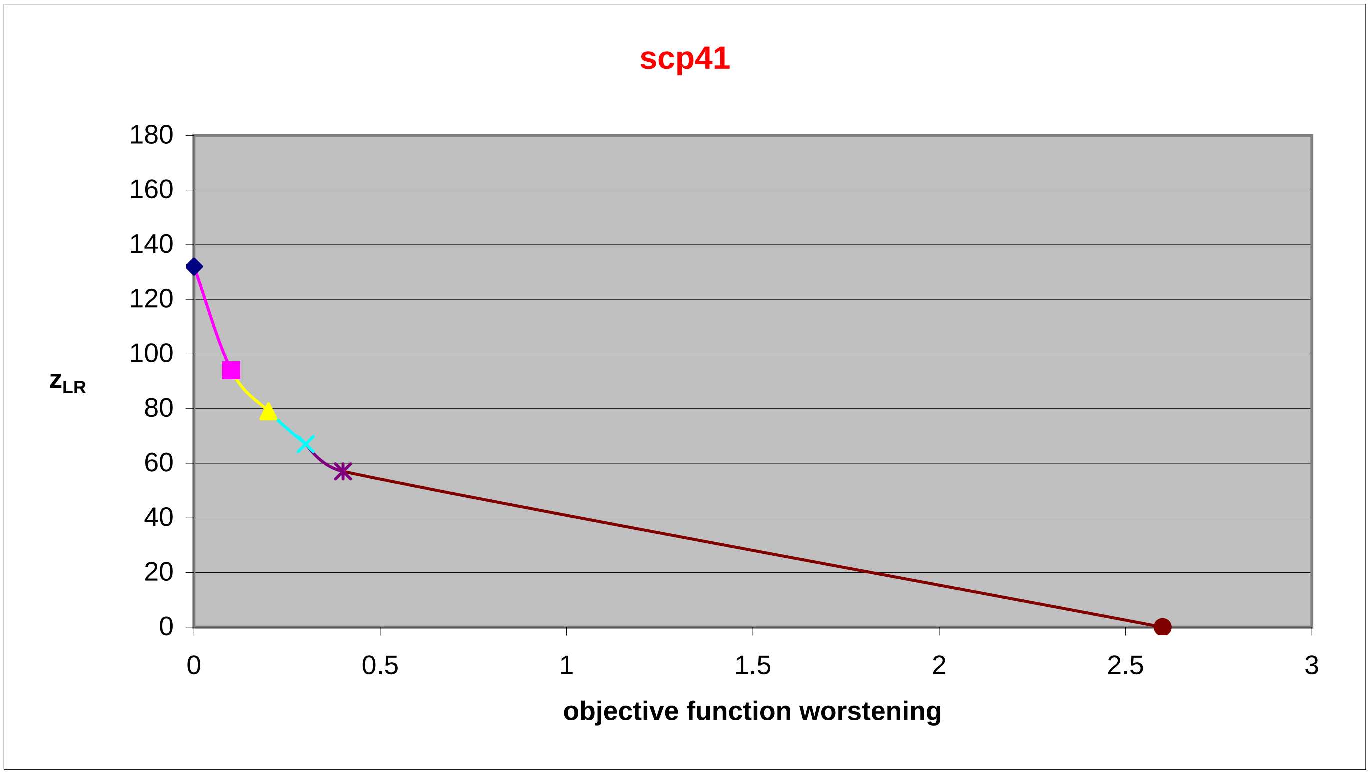


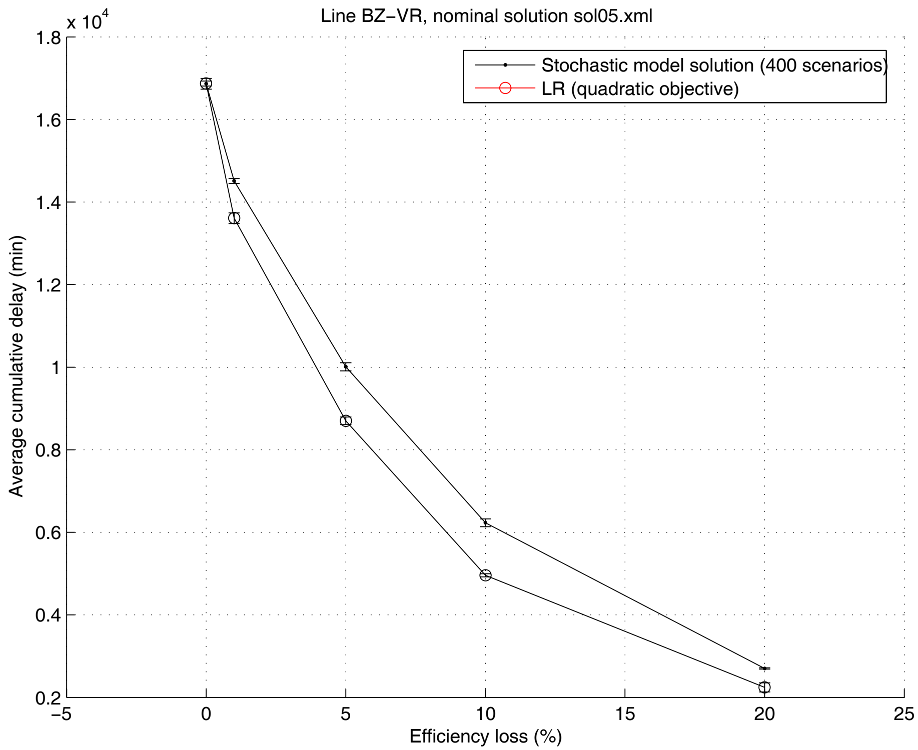
Figure 1: $z_{LR} =$ n. of rows covered only once

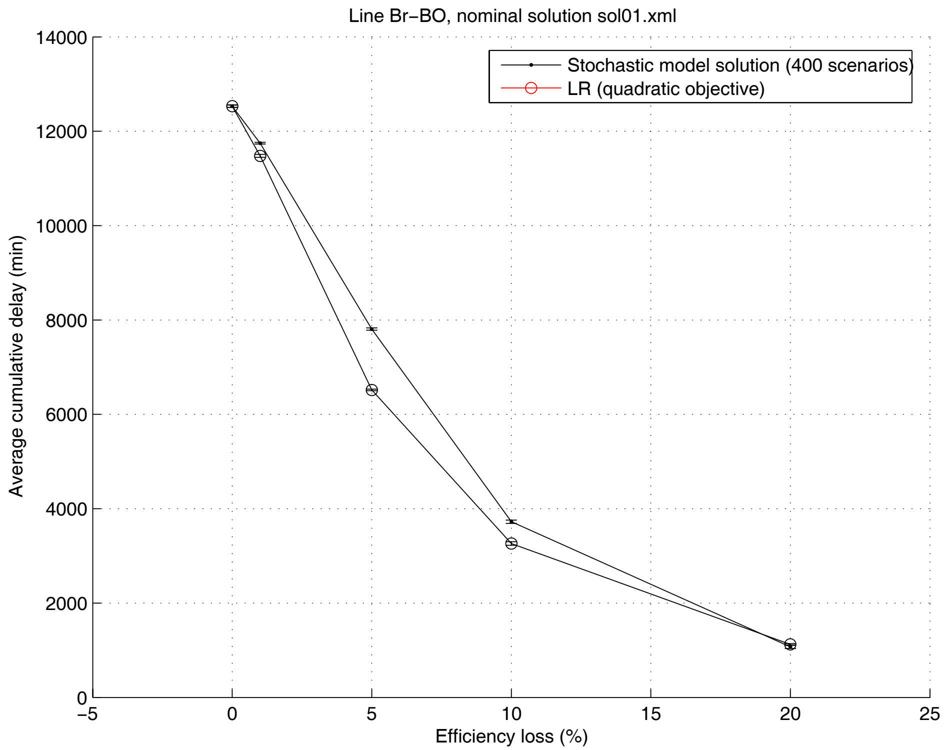
Computational experiments: train timetabling

- Aperiodic train timetabling instances from Rete Ferroviaria Italiana (RFI), the Italian railway infrastructure management company (from the ARRIVAL project)

- Different lines, for each line different schedules–each corresponding to a timetable that is feasible with respect to nominal data
- BS robustness: each travel and stop time to be increased by 5% (just infeasible)
- Comparison w.r.t. a Stochastic Programming (SP) model (400 scenarios)
- Robustness evaluated *a posteriori* through a **simulation** model

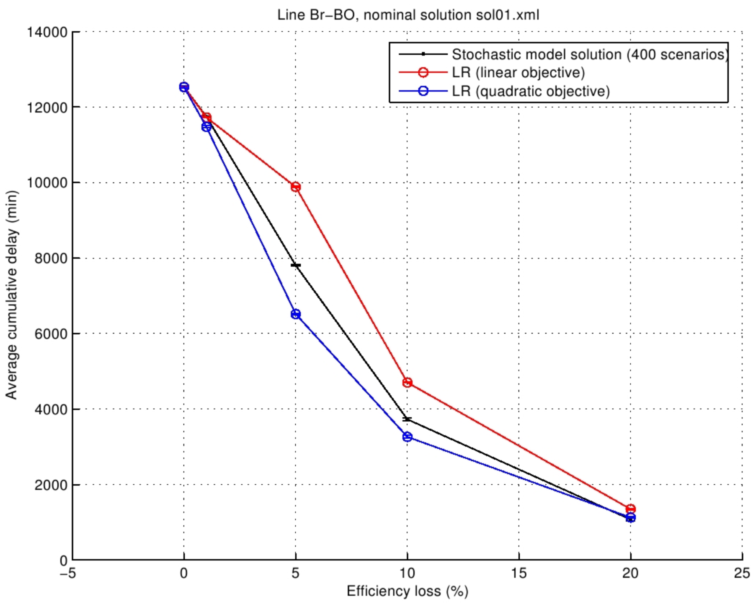
Computational experiments: train timetabling

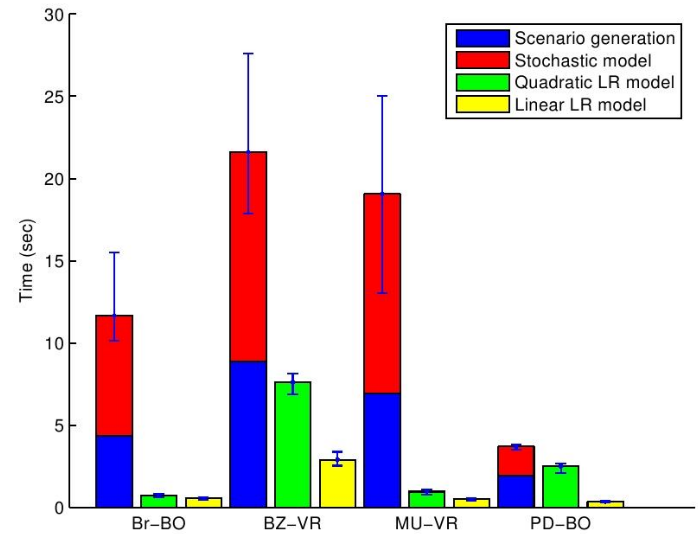




- Computing times: LR is **one order of magnitude faster** than SP
- Quality: the a-posteriori robustness of the LR solution is **comparable or better** than that obtained through SP

Role of the quadratic LR function (train timetabling)





- LR with linear objective function: shorter computing times but considerably less robustness

- Results to be validated on other combinatorial problems (work in progress)
- Extension obtained by **sampling a set of scenarios** so as to get average slacks and covariance matrices to be used in the LR model (work in progress)