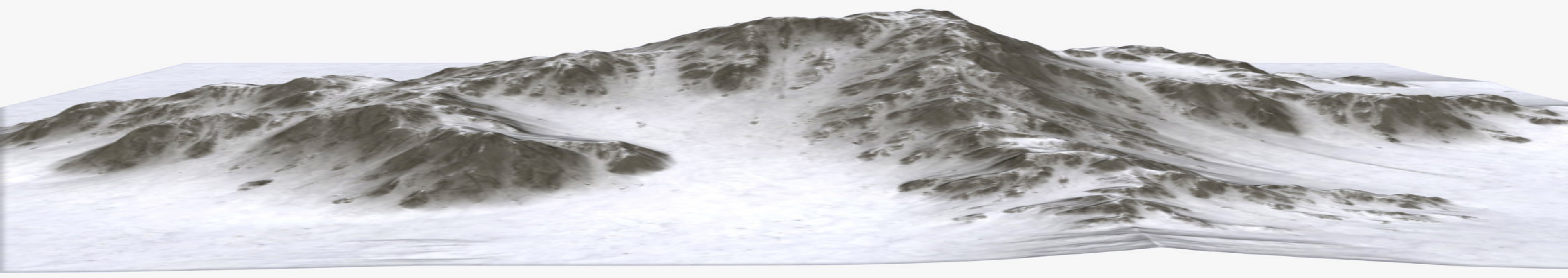
Objective Landscapes for **Constraint Programming**

Philippe Laborie, IBM
laborie@fr.ibm.com



 |

Goal

- Improve performance of a **Black-Box CP Solver** (in our case IBM ILOG CP Optimizer) by exploiting some information on the relation between decision variables and objective value
- This idea is not new !
  - Impact-Based Search [Refalo-2014]
  - Activity-Based Search [Michel&Van Hentenryck-2012]
  - Linear Relaxations [Beck&Refalo-2003, Laborie&Rogerie-2016]
  - ..., ..., ...
- But we had specific requirements:
  - It should work for variables with **huge domains**, typically with continuous semantics (start/end times of activities)
  - It should scale to **large problems** with several 1000s of variables

Goal

- Typical examples of objective functions in the real-world:
  - Scheduling in semiconductor manufacturing:

$$\boxed{\sum_i C_i \cdot x_i} + \boxed{\sum_j W_j \cdot F_j(e_j)} + \boxed{\sum_k U_k \cdot \max(0, d_k - D_k)^2}$$

    - x_i: Boolean variables (allocations of tasks to machines)
    - e_j: End time of task j, F_j: Earliness/Tardiness cost function
    - d_k: Delay variable between some particular tasks

  - Resource Constrained Project Scheduling with Net Present Value:

$$\boxed{\sum_p C_p \cdot \exp(-R \cdot e_p)} - \boxed{\sum_c C_c \cdot \exp(-R \cdot e_c)} + \boxed{W \cdot \max_i e_i}$$

    - e_p: End times of activities producing cash
    - e_c: End times of activities consuming cash

The Philosophy of Objective Landscapes

"Lazy Programmers for Lazy Machines."

- The new technique should (almost) not add any computational cost to the existing search algorithms
- The new technique should reuse existing algorithms as much as possible

→ Use existing algorithms in a different way !

Objective Landscape Definition

Context:

- A constrained-optimization problem:

  minimize $f(x_1,\ldots,x_n)$
  subject to $c(x_1,\ldots,x_n,y_1,\ldots,y_m)$

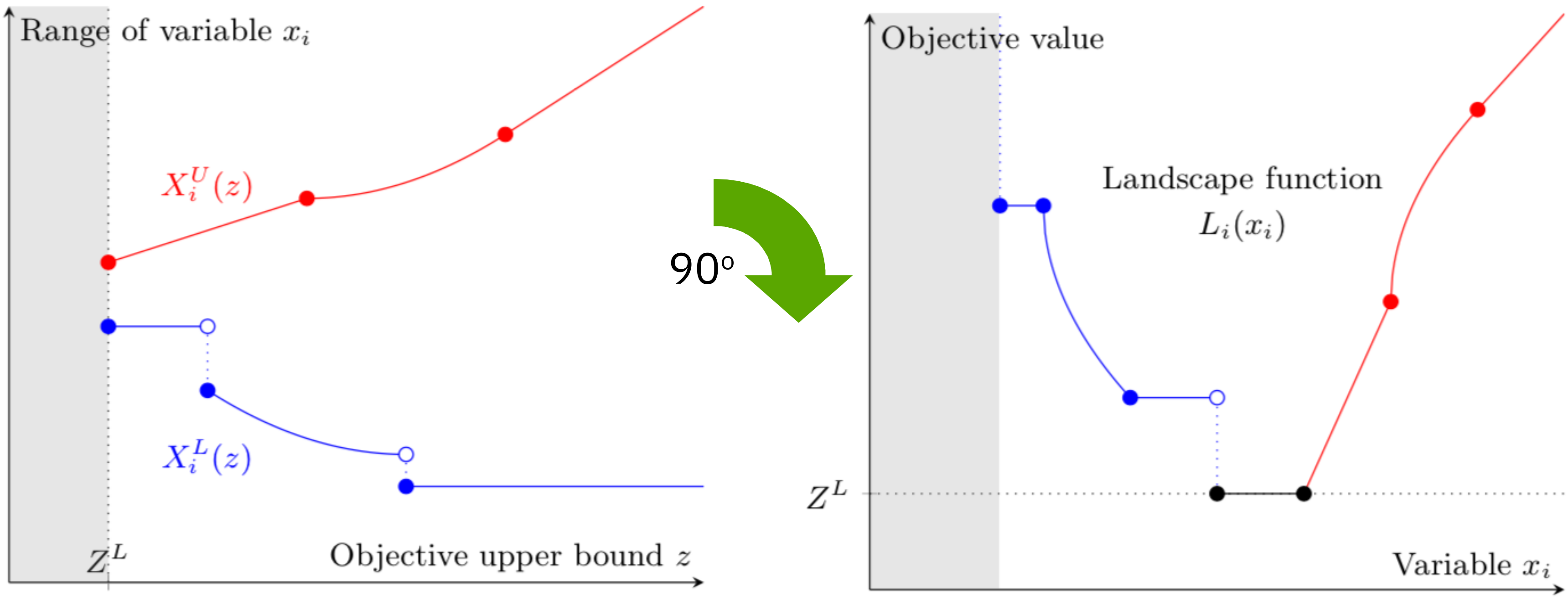
  $x_1,\ldots, x_n, y_1,\ldots, y_m$ are decision variables

- A constraint propagation algorithm $\mathcal{P}$ that propagates the constraint $f(x_1,\ldots,x_n) \leq z$ for a given value z

- Let $X_i^L(z)$ (resp. $X_i^U(z)$) denote the min (resp. max) value in the domain of x_i after propagation of $f(x_1,\ldots,x_n) \leq z$ by $\mathcal{P}$

Objective Landscape Definition

- Landscape function L_i of a variable x_i

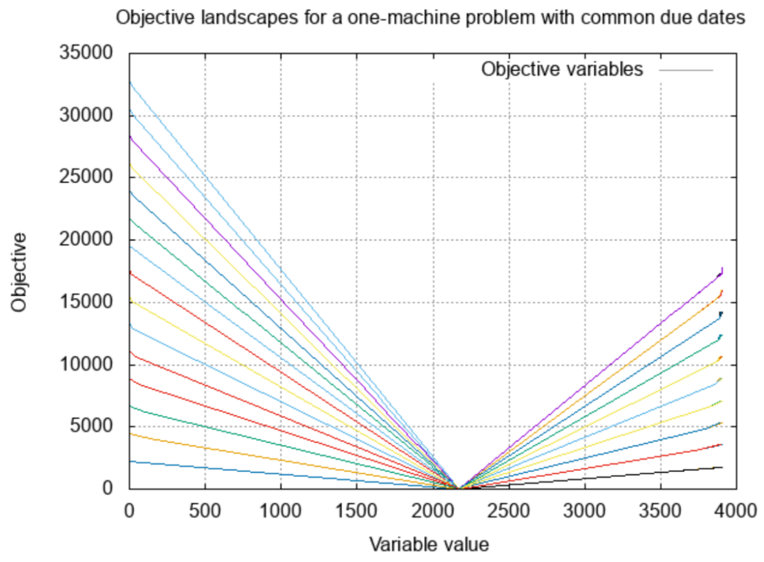


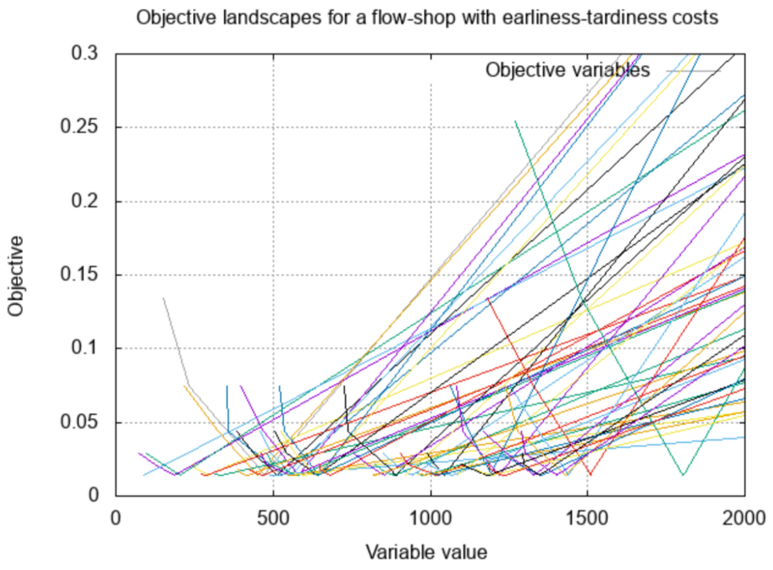
- $L_i(v)$ can be interpreted as an optimistic estimation of the objective value when $x_i=v$ (more formalization in the paper)

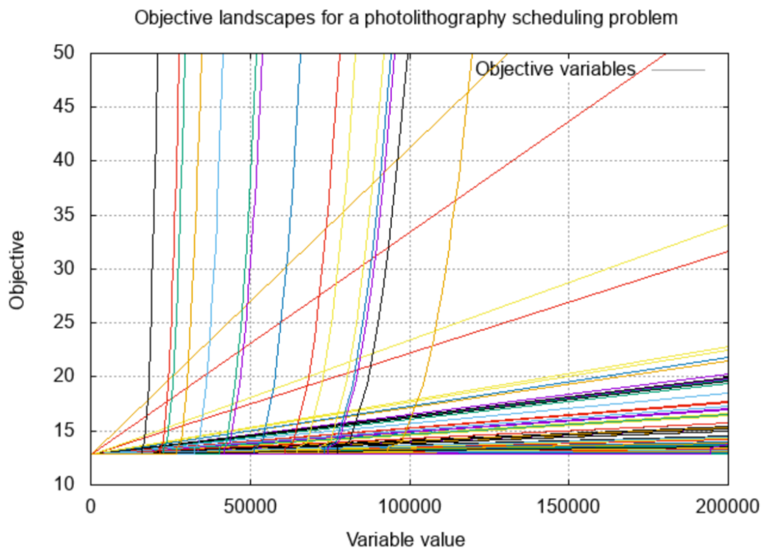
Objective Landscape Computation

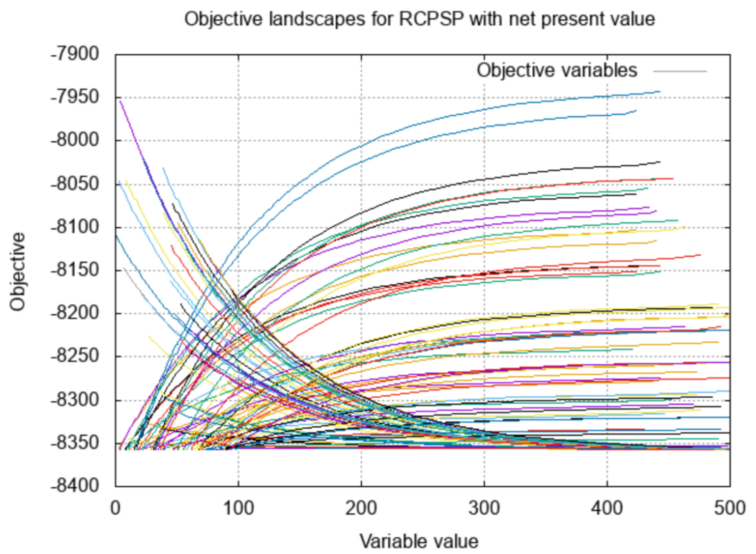
- Objective Landscape functions can be **quickly** computed **once and for all** before entering the search by sampling a discrete set of objective values z and running propagation algorithm $\mathcal{P}$ and recording the bounds for all variables x_i
- They are represented using a **light-weight** structure (array of pairs variable value/objective value)
- Computing the landscape value $L_i(v)$ for a value v in the domain of x_i is **fast**: it can be done in O(log(N)) if N is the number of samples (generally only a few tens) by simple binary search

Examples of Landscape Functions







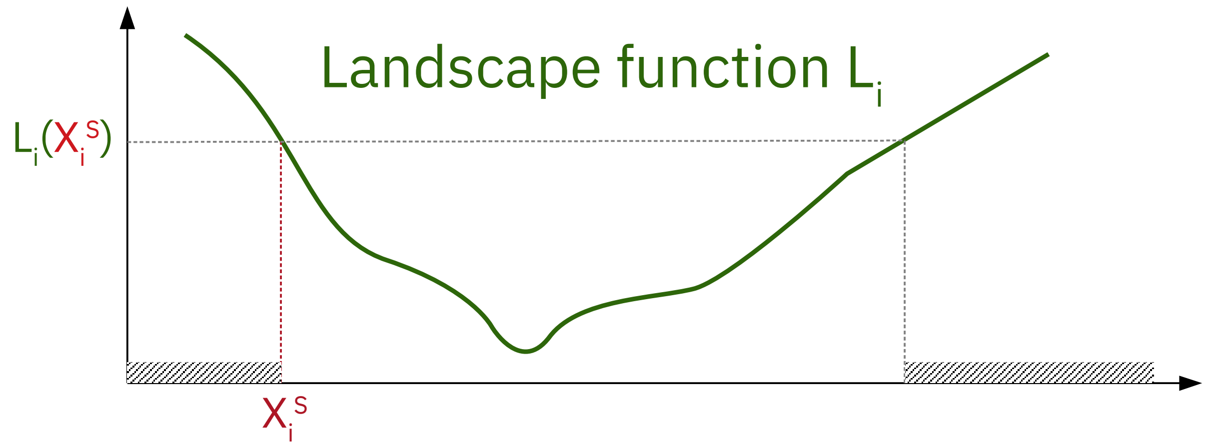


Objective Landscape Exploitation

- In CP, landscape functions can be used in many ways during the search:
  - Selection of variables
  - Selection of values
  - In a Large Neighborhood Search (LNS) context:
    - Selection of LNS fragment
    - <u>Prevent cost degradation for some variables in LNS moves</u>
    - Selection of variables and values in LNS completion
- We still did not fully explore all these ideas

Objective Landscape Exploitation

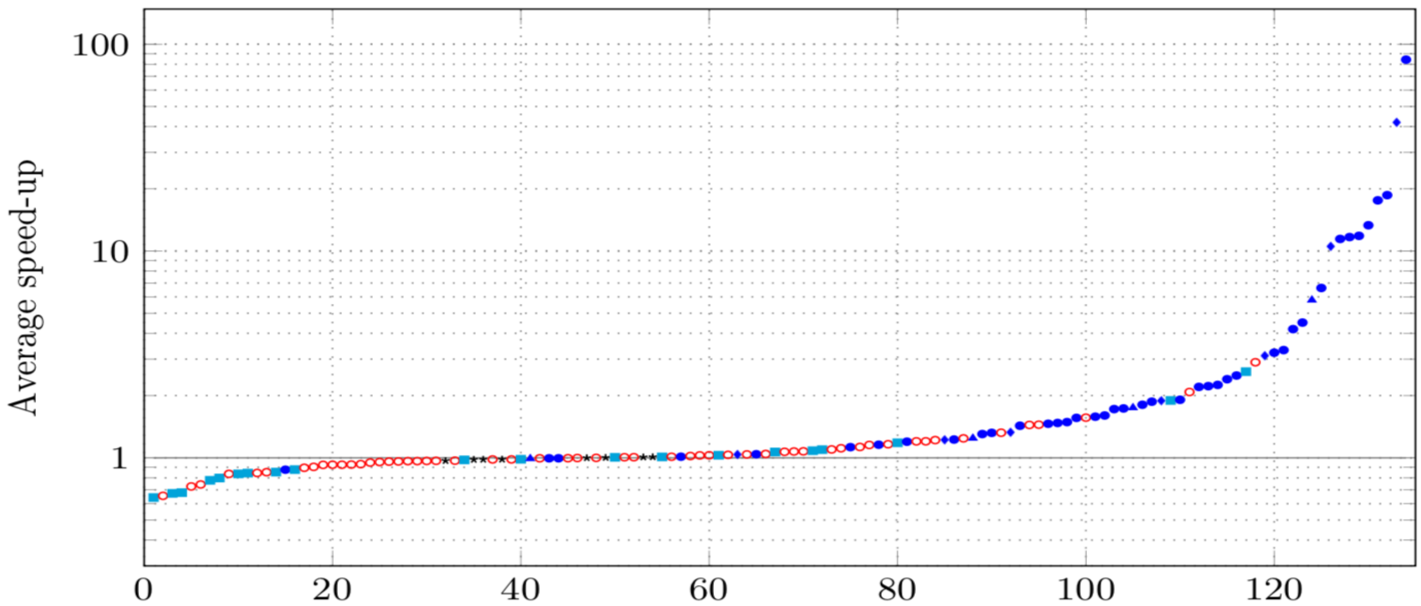
- Preventing cost degradation for some variables in LNS moves
  - At the top level of a move, we have the current solution S and a relaxed model around S that we want to re-optimize
  - For a given variable x_i, let X_i^S denote the value of x_i in solution S



  - For a certain number of variables x_i (randomly selected) we just consider the additional constraint $L_i(x_i) \le L_i(X_i^S)$

Experimental Results

- Average speed-up when using objective landscapes on 135 different families of scheduling problems



Aggregation type	Main variables type	Number of families	Average speed-up	Mark on Fig. 10
Max	Start/End	55	1.05	○
Sum	Start/End	40	2.62	●
Sum	Presence	29	0.98	■
Sum	Length	7	3.31	♦
Sum	Height	4	1.88	▲

Conclusion and Future Work

- Implemented in automated search of IBM ILOG CP Optimizer V12.7.1/12.8
- Average speed-up x1.5 between 12.7 and 12.8
- Up to two orders of magnitude speed-up for some problems with objective expressed as a weighted sum of variables with continuous semantics (very common in real problems)
- Future work:
  - Investigate other forms of landscape exploitation in the search
  - Objective landscapes can be generalized to store holes in the domains
  - Constraints of the problem could be exploited too