Temporal Linear Relaxation in IBM ILOG CP Optimizer

Philippe Laborie, Jerome Rogerie
Optimization CP Team
IBM ILOG France Lab

laborie@fr.ibm.com , rogerie@fr.ibm.com

Rationale

- We present a loose coupling between a linear relaxation and a CP engine Large Neighborhood Iteration.
- The relaxation exploits the dominant structures of the problem to provide well informed guidelines for the completion strategy.
- We apply this idea for having a robust automatic search for scheduling problems involving irregular costs
  - tardiness cost, non convex temporal costs, non execution cost, execution mode cost, or length and workload costs

IBM ILOG CP Optimizer

- A constraint programming library
  - integer variable and numerical expressions
  - Algebraic language and global logical constraints
  - Extension interface by encoding your own constraints and search traversal solution generation

+ Automatic search and optimization
+ Specific algebraic modeling language for temporal problems

Temporal Model in CPO

- Based on the notion of interval as a temporal segment

  $[s, e)$ $s \leq e$ integers, $l = e - s$

- Interval variables
  - Decisions
    - Is the interval present in the solution
    - Start and end points of a present interval
  - Domain

    $dom(a) \subseteq \{\perp\} \cup \{ [s,e) \mid s,e \in \mathbb{Z}, s \leq e \}$
  - Ranges

    start, end, and length integer ranges (notation s, e, l), presence Boolean range (notation x)
  - Presence Constraint

    presenceOf(a) states $\perp$ not in dom(a)

Precedence and Implication graphs

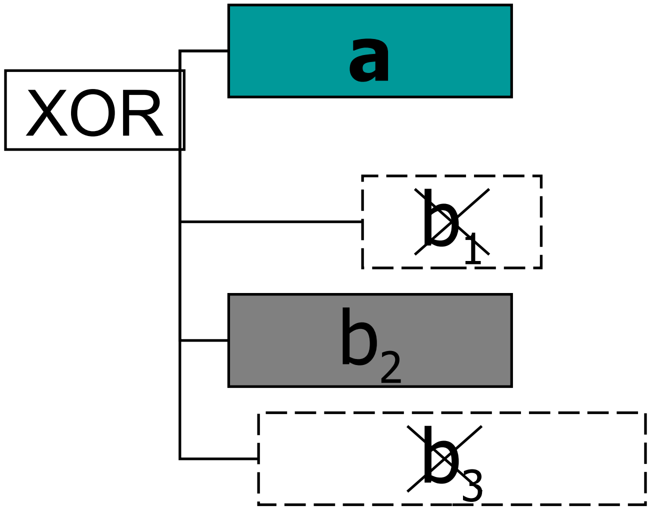
- All precedences between interval variables are collected in a precedence graph
  - a ends before start of b means:

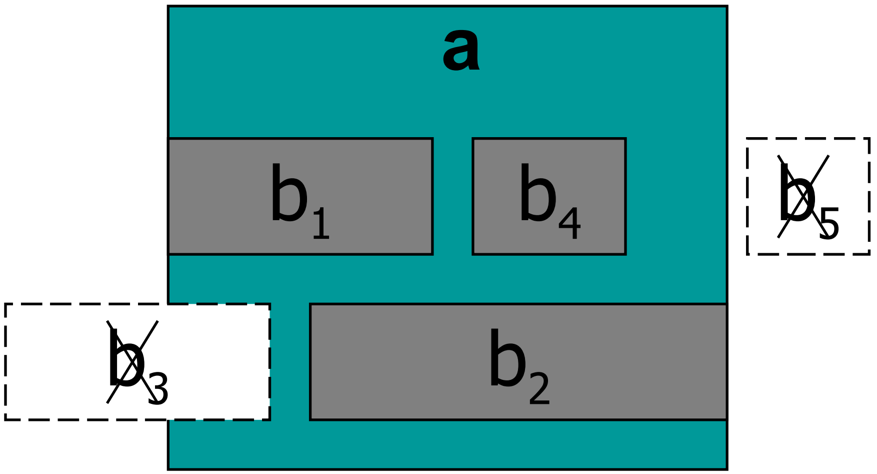
    if a and b are present then $s(a) \leq e(b)$

- All implications between interval variables are collected in an implication graph
  - The constraint $presenceOf(a) \leq presenceOf(b)$ adds the arc (a implies b)

Breakdown structure

- The breakdown structure of a task is stated thanks to alternative and span constraints on interval variables.





Resources and objective terms

- Resources are structures defined everywhere in time (between origin and horizon)
  - Lines (segmented integer functions)

    $\{([s_i, e_i), v_i)$ integers, $s_i \leq e_i$, $0 \leq v_i$, $e_i \leq s_{i+1}\}$
    - Calendar, cumulative, state resources, oven, …
  - Sequence of intervals

    permutation of a set of intervals linked by precedences
    - Disjunctive resources, transportation systems, batches, campaign, …

- Numerical expressions are functions of ranges of the intervals.
  - Example startOf(a, v) is the value of the start point of a if present, v if a is absent
  - Start, end, length, height of cumul, next or previous values on sequences, …

LNS in CPO temporal problems

- Issue is having a robust iterative procedure for scheduling problems including presence (alternative, optional tasks) costs, early tardy costs, and duration (or workload) costs,
  - Main hypothesis is the cost is dominated by the time placement.
  - The incumbent is relaxed in a Preferred Ordered Set (POS) given by a acyclic precedence network.
  - The LNS moves consist in selecting a fragment of the POS

Incumbent

Interval variable assignments

Objective value

Model elements

Lines, sequences, span, alternative, expressions

POS

Acyclic Precedence network of present intervals

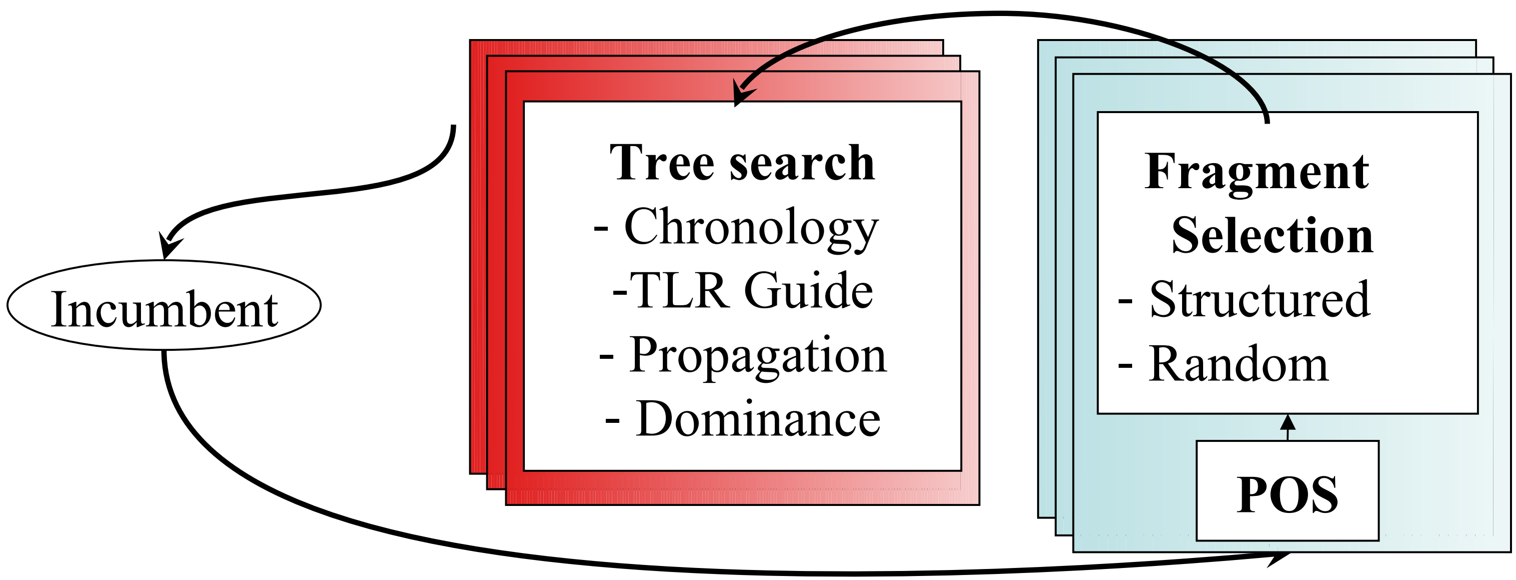
Absent Intervals

Move

Model + POS Fragment + better objective constraint

TLR Guideline

- The tree search completion procedure generating solutions is essentially a chronological decisions procedure on the interval variables presence and end-points
- Eventually a Temporal Linear Relaxation (TLR) is computed at the fix point of the initial propagation of the move and solved by Linear Programming.
  - The TLR is used as a guide for date decision selections and presence values.



Notations for the TLR formulation

- Notations in the Linear model
  - Interval ranges define the range variables s, e, l in the LP
  - Interval range presence: x boolean

- A CP numerical range or expression defines a range variable v whose bound are set from the current domain in the propagation engine.

$$v \text{ in } [V_{min}, V_{max}]$$

Principle of the TLR formulations

- The implication graph is stated by binary constraints:

$$x(a) \leq x(b)$$

- The precedence graph is stated thanks to big-M constraints reified by known implications (y in start or end point) (including e = s + l for an interval variable)

$$y_1 - y_2 + delay - M_1(1 - x_1) - M_2(1 - x_2) \leq 0$$

x_1 implies not x_2

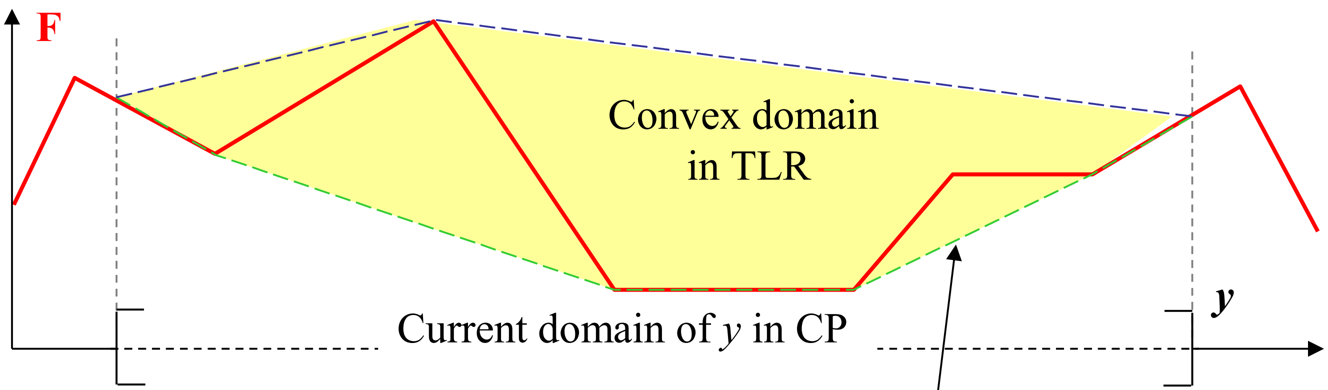
x_1 and x_2 present

x_1 implies x_2

- From the set of precedences for each y in s, e and l of an interval variable, we state Y_o in $[Y_{min}, Y_{max}]$ as the value of y when the interval is absent (x = 0) and such that Y_o tightens the big-M constraints.

- The TLR formulation is then calculated from objective terms introspection.

Unary constraints or expressions



- The expression is a numerical function of the range y (in s, e, l) of the interval. For example a piecewise linear function.
- The value of the expression is v_a when the interval is absent. Be v_p the linear relaxation of V when the interval is present

$$v = x*v_p + (1-x)*v_a$$

- The expression defines a range variable v in the TLR that is initialized from the current bounds of y in the propagation engine $[V_{min}, V_{max}]$
- The idea is to used the absence values in the precedence graph relaxation to state the big-M constraint.
- Be V_o the value of F for Y_o, the formulation is: $v = v_p + x(v_a - V_o)$

Binary constraints and expressions

- A full examples is shown in the paper for the overlap length of two intervals.
- The general formulation is $v = x_1 * x_2 * v_p + (1 - x_1 * x_2) * v_a$
- From the implication graph (when possible e.g. x1 is equivalent with x2) we used a similar formulation as for unary expressions:

$$v = v_p(a_1, a_2) + x*(v_a - V_o(a1, a2))$$

- v_p formulation:
  - Binary expressions usually expresses some temporal disjunction between the interval variables
  - The formulation of v_p is improved by cuts computed from
    - The current domain of a_1 and a_2 in the propagation engine.
    - Eventual precedence graph arcs involving a_1 and a_2.

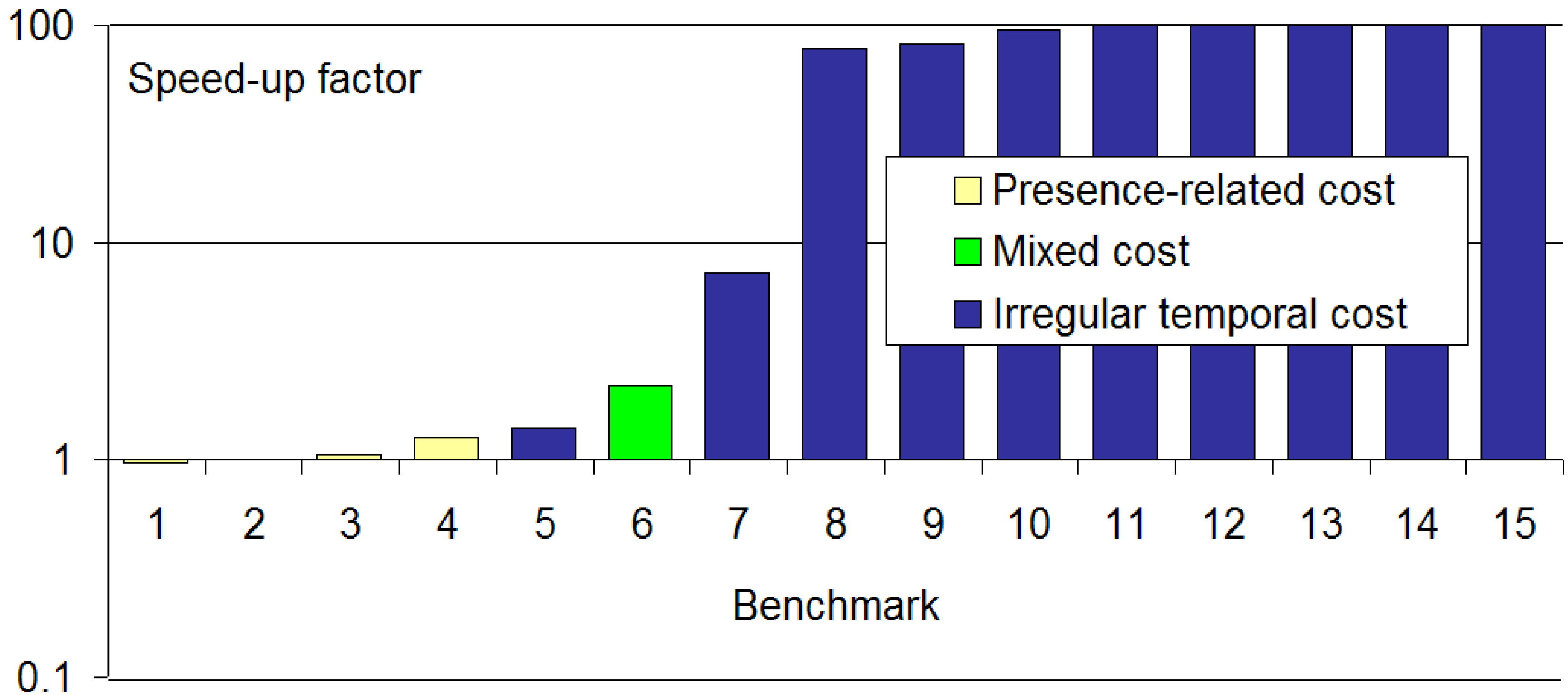
Other elements of the model

- Breakdown structures are composition of temporal constraints and logical constraints
  - an alternative is a SOS constraints on presence variables + synchronization of end-points of intervals

- Resource constraints are still captured by frozen precedence arcs of the POS
  - TLR formulations use rough energetic relaxations

- Numerical expression formulations use similar convex relaxations as the piecewise linear function formulations

Experimental Results



- 15 scheduling benchmarks comparing the iterative procedure with against without the TLR guide. The LP is ILOG IBM Cplex.
- Speed up is relative to the time at which the compared algorithms reached the same objective value under a global time limit.
- A caped 100 factor are problems such that the without TLR procedure never reaches even the first incumbent of the with TLR procedure.

Conclusion

- The TLR formulation as a guide is a very effective method for optimizing non regular scheduling costs
  - The TLR tends to be very informative as most of the resource constraints are still captured by frozen precedence arcs of the POS.

- Future works will concern both the control and the diversification of the temporal relaxation by:
  - Controlling the size and the content of the TLR depending on the characteristics of the problem and the current impact on the search,
  - Exploring stronger relaxations in the TLR, for instance by solving a MILP enforcing the Boolean presence status of interval variables,
  - Considering concurrent relaxations that exploits other dominant structures of the problem like capacity planning or sequencing sub-problems.