# The resource-constrained activity insertion problem with minimum and maximum time lags

Christian Artigues and Cyril Briand

April 25, 2008

# Insertion problems in scheduling: generalities

**An insertion problem is a scheduling problem in which the set of activities $A$ is partitioned into two sets**

- The set of already scheduled activities $A \setminus I$ (forming the baseline schedule).
- The set of unscheduled activities $I$ (to be inserted in the baseline schedule).

**Solving the insertion problem amounts to schedule all the (scheduled and unscheduled) activities**

- Subject to scheduling constraints (precedence constraints, time windows, resource-constraints).
- Optimizing a scheduling objective function (makespan,...).
- Involving constraints or objective of **stability**: the obtained schedule must respect in some way the baseline schedule.

# Interest of solving insertion problems

## On-line scheduling

- Insertion of unexpected activities without performing a global rescheduling.

## Neighborhood search

- Neighborhood defined in terms of reinsertion of (critical) activities.

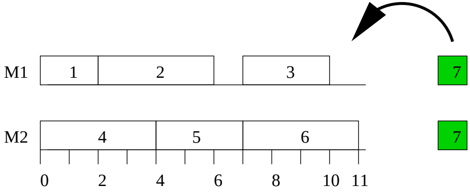
## Issues for designing an insertion problem

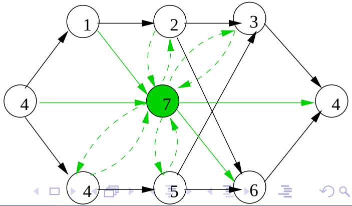
- The insertion algorithm should be fast, preferably polynomial.
- The search space should contain a significant number of solutions.

# The disjunctive-graph model

## Disjunctive graph $G = (V, U, E)$

- $n$ nodes $V$.
- Precedence constraints (original and set by the baseline schedule) modeled by directed arcs $U$.
- Each unscheduled activity is linked to any other activity requiring the same machine through a pair of opposite (disjunctive) arcs. $E$ is the set of all such pairs.
- Each arc is valuated by the duration of the origin activity.
- Dummy nodes 0 ($n+1$) is a predecessor (successor) of all activities $A$.



# Relevant work for disjunctive insertion problems


- Vaessens, R.J.M., Generalized Job Shop Scheduling: Complexity and Local Search, Ph.D. thesis, Department of Mathematics and Computing Science, Eindhoven University of Technology, 1995.
- F. Werner and A. Winkler, Insertion techniques for the heuristic solution of the job shop problem, Discrete Applied Mathematics 58 (1995)
- P. Brucker and J. Neyer, Tabu-search for the multi-mode job-shop problem, OR Spektrum 20 (1998)
- Y.N. Sotskov, T. Tautenhahn and F. Werner, On the application of insertion techniques for job shop problems with setup times, RAIRO Operations Research. 33 (1999)
- C. Artigues and F. Roubellat, An efficient algorithm for operation insertion in a multi-resource job-shop schedule with sequence-dependent setup times, Production Planning and Control 2 (2002)
- T. Kis and A. Hertz, A lower bound for the job insertion problem, Discrete Applied Mathematics 128 (2003)
- H. Gröflin and A. Klinkert, Feasible insertions in job shop scheduling, short cycles and stable sets, 177 (2007)

# The (single-)resource-constrained project scheduling problem with minimum and maximum time lags (RCPSP/max)

## Definition

- a resource of availability $B$
- $n$ activities.
- Each activity must be processed without interruption on $b_i$ **unspecified** units of the resource.
- The availability of the resource cannot be exceeded at any time period.
- Generalized Precedence constraints between activities $S_j - S_i \geq l_{ij}$, a positive or negative time lag.
- Objective: makespan minimization.

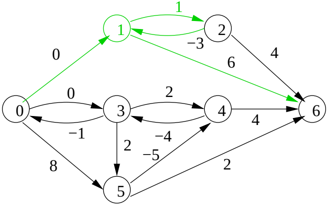
# Insertion problem example

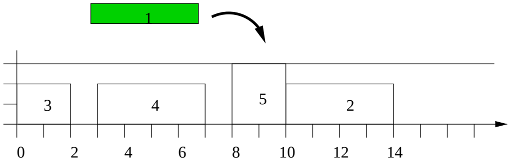
## Data

- $m = 3$
- $n = 5$, $I = \{1\}$

| $i$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| $p_i$ | 0 | 6 | 4 | 2 | 4 | 2 | 0 |
| $b_i$ | 0 | 1 | 2 | 2 | 2 | 3 | 0 |

- precedence constraints and baseline schedule



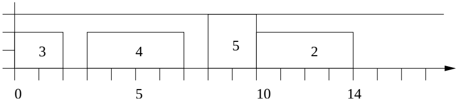




From Neumann *el al* (2003)

# Resource flow network

## Definition

- a flow $f_{ij} \geq 0$ from activity $i$ to activity $j$ represents the number of machines on $j$ is a direct successor of activity $i$.
- Each activity has a capacity of $b_i$ units.
- Activity 0 is a source of $m$ units while Activity $n+1$ is a sink.
- A flow is valid for an activity set $V$ if it satisfies the conservation properties and the activity capacity on each resource for each activity of $V$




The following resource flow is valid for $A \setminus \{1\}$: $f_{03} = 2$, $f_{34} = 2$, $f_{45} = 2$, $f_{05} = 2$, $f_{52} = 2$, $f_{56} = 1$, $f_{26} = 1$.

Fortemps and Hapke (1997), Artigues and Roubellat (2000), Neumann *el al* (2003)

# Flow-based formulation of the insertion problem (RCAIP/max)

## Baseline resource flow

A Baseline resource flow is any flow valid for $A \setminus I$ such that

- There is no flow involving $I$
- $S_j < S_i + p_i$ in the baseline schedule implies $f_{ijk} = 0$

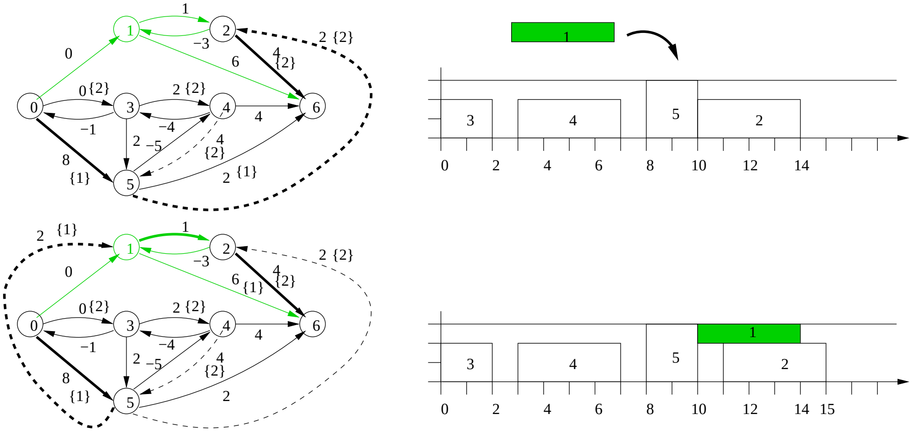
## Graph induced by a flow

The graph induced by a flow is the graph of precedence constraints augmented by an arc $(i, j)$ valuated by $p_i$ for each flow $f_{ij} > 0$.

## RCAIP

Given a baseline flow $f$, find a flow $f'$ valid for $A$ such that its induced graph has a minimal longest path length between 0 and $n + 1$ **without increasing the flow between two scheduled activities**.

# Insertion example




## Remark

An insertion position can be represented by a subcut $(\alpha, \beta)$ of the graph induced by the baseline flow such that $\sum_{i \in \alpha, j \in \beta} f_{ij} \geq b_i$

# Characterization of feasible insertions for the decision variant of the RCAIP/max (1/2)

## Search variant

To obtain the search variant of an insertion such that $C_{\max} \leq v$, we set the value of arc $(n+1, 0)$ to $\max(l_{(n+1)0}, -v)$.

## Distance matrix

- $\delta^v_{ij}$ denotes the length of the longest path from $i$ to $j$ in the graph induced by the baseline flow for a makespan of at most $v$.
- $\delta^v_{ij}$ can be computed in $O(n^3)$ by the Floyd-Warshall algorithm.

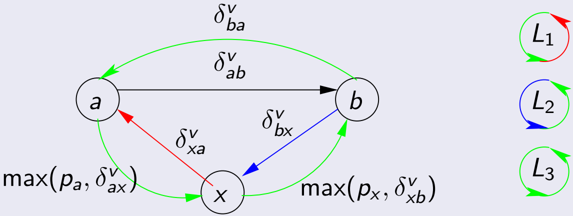
## Necessary condition for feasibility

There is no solution with a makespan of at most $v$ is the problem of computing Matrix $\delta^v$ admits no solution.

# Characterization of feasible insertions for the search variant of the RCAIP/max (2/2)

Inserting $x$ into $(\alpha, \beta)$ may generated the following cycles ($a \in \alpha$, $b \in \beta$).

## Insertion-induced elementary cycles




## Theorem

*The insertion of $x$ in an insertion position $(\alpha, \beta)$ is feasible if and only if* $\max(L_1, L_2, L_3) \leq 0$ *with* $L_1 = \max_{a\in\alpha}(p_a + \delta^v_{xa})$, $L_2 = \max_{b\in\beta}(p_x + \delta^v_{bx})$, $L_3 = \max_{(a,b)\in\alpha\times\beta}(p_a + p_x + \delta^v_{ba})$.

# RCAIP/max complexity

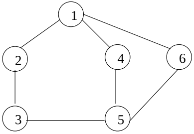
**Theorem**

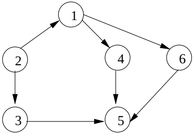
*The RCAIP/max is NP hard.*

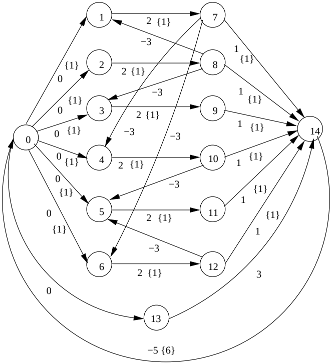
**Proof**

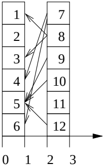
- We prove NP-completeness of the decision variant by reduction from the independent set problem.

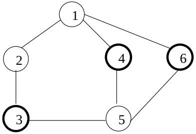
# Proof (illustration)

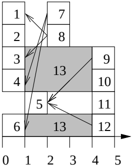


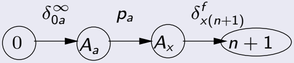



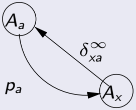
# RCAIP with minimum time lags only (optimization variant)

## insertion-induced elementary cycles and $0, n+1$ longest paths

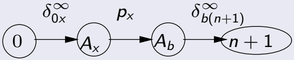
Consider distance matrix $\delta_{ij}^{\infty}$, $a \in \alpha$, $b \in \beta$



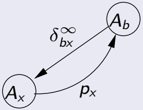

$0, n+1$ path type 1



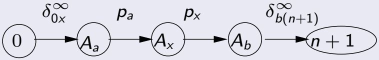

cycle type 1



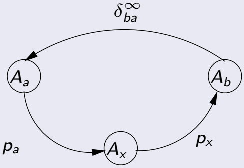

$0, n+1$ path type 2




cycle type 2




$0, n+1$ path type 3




cycle type 3

# Dominant insertions and polynomial algorithm

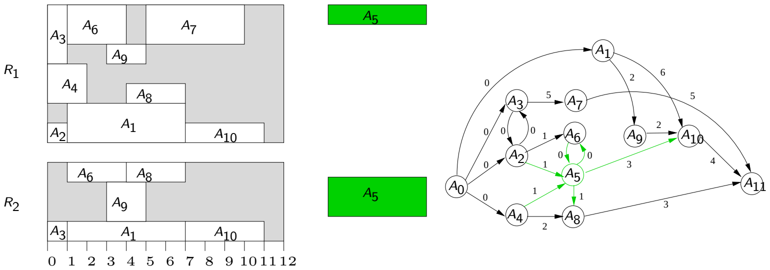
## Definitions

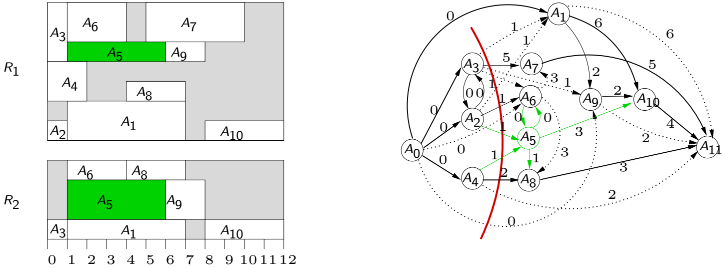
- $\gamma_0 = \{i \in A | \delta^\infty_{ix} = 0 \text{ and } \delta^\infty_{xi} = 0\}$ is the set of activities synchronized with $x$.
- $\alpha_0 = \{i \in V \setminus \gamma_0 | \delta^\infty_{ix} \geq 0\}$
- $\beta_0 = V \setminus (\gamma_0 \cup \alpha_0)$
- $\mu(\alpha, \beta) = \{i \in \alpha | \delta^\infty_{0i} + p_i = \max_{A_a \in \alpha}(\delta^\infty_{0a} + p_a)\}$
- $\nu(\alpha, \beta) = \{i \in \beta | \delta^\infty_{i(n+1)} = \max_{b \in \beta}(\delta^\infty_{b(n+1)})\}$
- $\nu'(\alpha, \beta) = \{i \in \nu | \delta^\infty_{xi} = -\infty\}$

## Algorithm $O(n^2 m)$

- Set initial cut $(\alpha, \beta)$ to $(\alpha_0, \beta_0)$
- While there remains enough capacity in $(\alpha, \beta)$ perform evaluate recursively all subcuts obtained by removing $\mu$ from $\alpha$ and set next cut $(\alpha, \beta)$ to $(\alpha \cup \nu', \beta \setminus \nu)$.

# RCPSP with minimum time lags only Example



# Concluding remarks and further work

- Even inserting a single activity is hard when maximum time lags are considered with the resource-flow model.
- Design a polynomial insertion framework for the RCPSP/max.
- Develop an insertion heuristic for the RCAIP/max limiting the insertions to the ones of the time lag min-only case.
- Develop an insertion-based heuristic for the RCPSP/max.

*C. Artigues and F. Roubellat, A polynomial activity insertion algorithm in a multi-resource schedule with cumulative constraints and multiple modes, European Journal of Operational Research 127 (2000)*

*C. Artigues, P. Michelon and S. Reusser, Insertion techniques for static and dynamic resource-constrained project scheduling, European Journal of Operational Research 149 (2003)*

*C. Artigues, C. Briand, Complexity of insertion problems for resource-constrained project scheduling with minimum and maximum time lags, LAAS report 07678, LAAS-CNRS, Toulouse, 2007*