

Bonnin Camille^{*,0}, Thomas Dissaux[†], Malapert Arnaud⁰,
Nicolas Nisse[†], Espinouse Marie-Laure^{*}

^{*}Univ. Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, 38000 Grenoble, France. {camille.bonnin, marie-laure.espinouse}@grenoble-inp.fr
⁰Université Côte d'Azur, CNRS, I3S, France. arnaud.malapert@univ-cotedazur.fr
[†]Université Côte d'Azur, Inria, CNRS, I3S, Sophia Antipolis, France. {thomas.dissaux, nicolas.nisse}@inria.fr

Context and main objective of the thesis

Scheduling: allocating time and limited **resources** to **tasks** respecting their **constraints**. Often, we try to optimize one or more **objectives**.

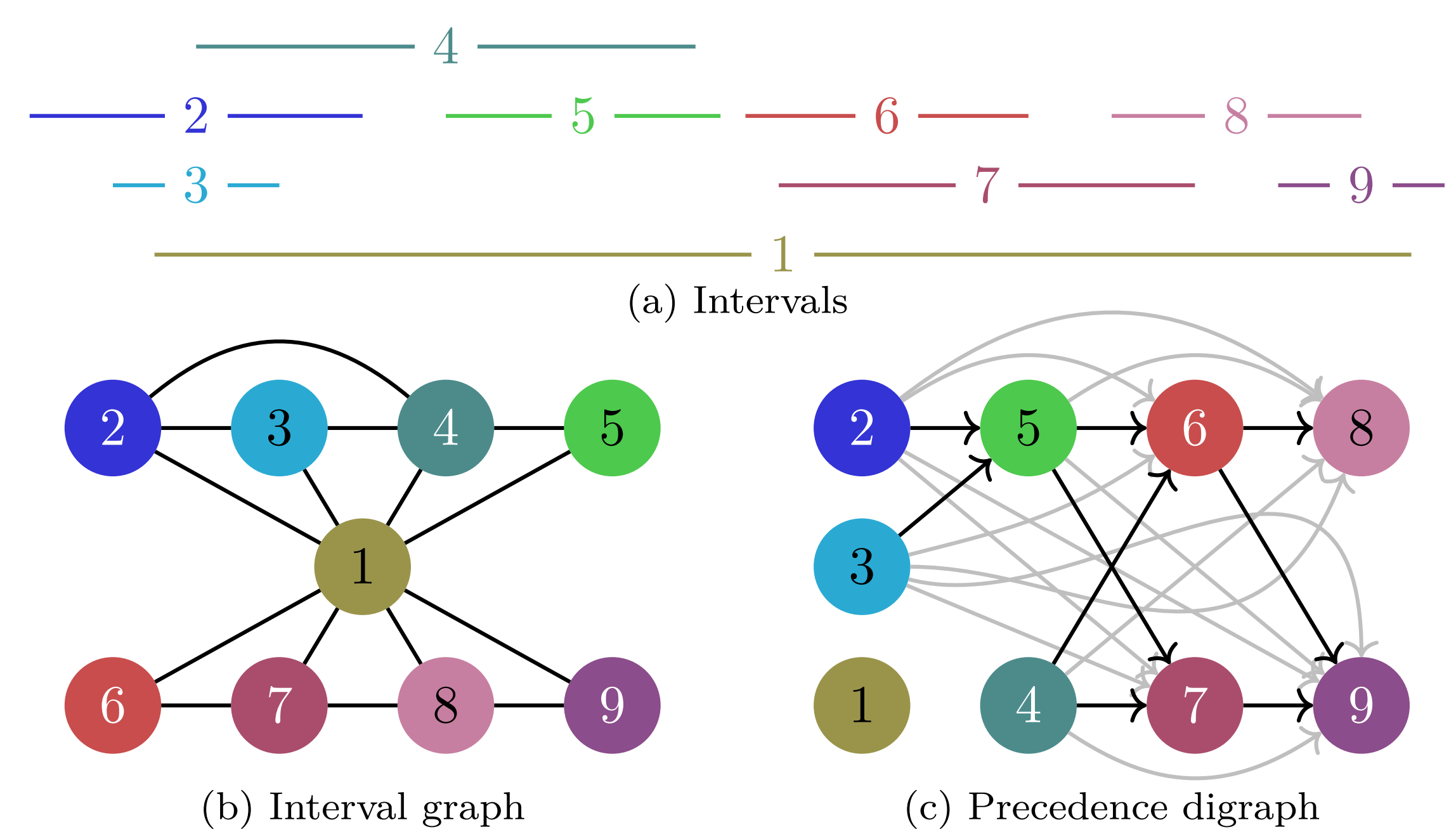
CP and scheduling: CP focuses on **satisfaction** problems. It is possible to transform optimisation problems with "max" objective into a succession of satisfaction problems, but it is more difficult with "Σ" objectives.

Thesis main objective: to improve the consideration of "Σ" objectives in CP, in particular the **flowtime** objective (*i.e.*, the sum of the completion times):
 $\sum C_j$ and its **weighted** version: $\sum w_j C_j$.

The COMPLETION constraint

- Defined by Kovács and Beck in 2011¹
- COMPLETION($[S_1, \dots, S_n], [p_1, \dots, p_n], [w_1, \dots, w_n], C$)
 $\stackrel{\text{def}}{\iff} ((S_i + p_i \leq S_j \vee S_j + p_j \leq S_i) \wedge \sum_i w_i (S_i + p_i) = C)$, where $C = \sum_i w_i C_i$
- The filtering algorithm uses a relaxation: $1|r_j;pmtn|\sum w_j M_j$
- Ongoing work:** defining another filtering algorithm for COMPLETION using another relaxation: $1|sp-graph|\sum w_j C_j$. This relaxation can be solved in polynomial time by an algorithm given by Lawler² that uses the tree decomposition of an SP-digraph.

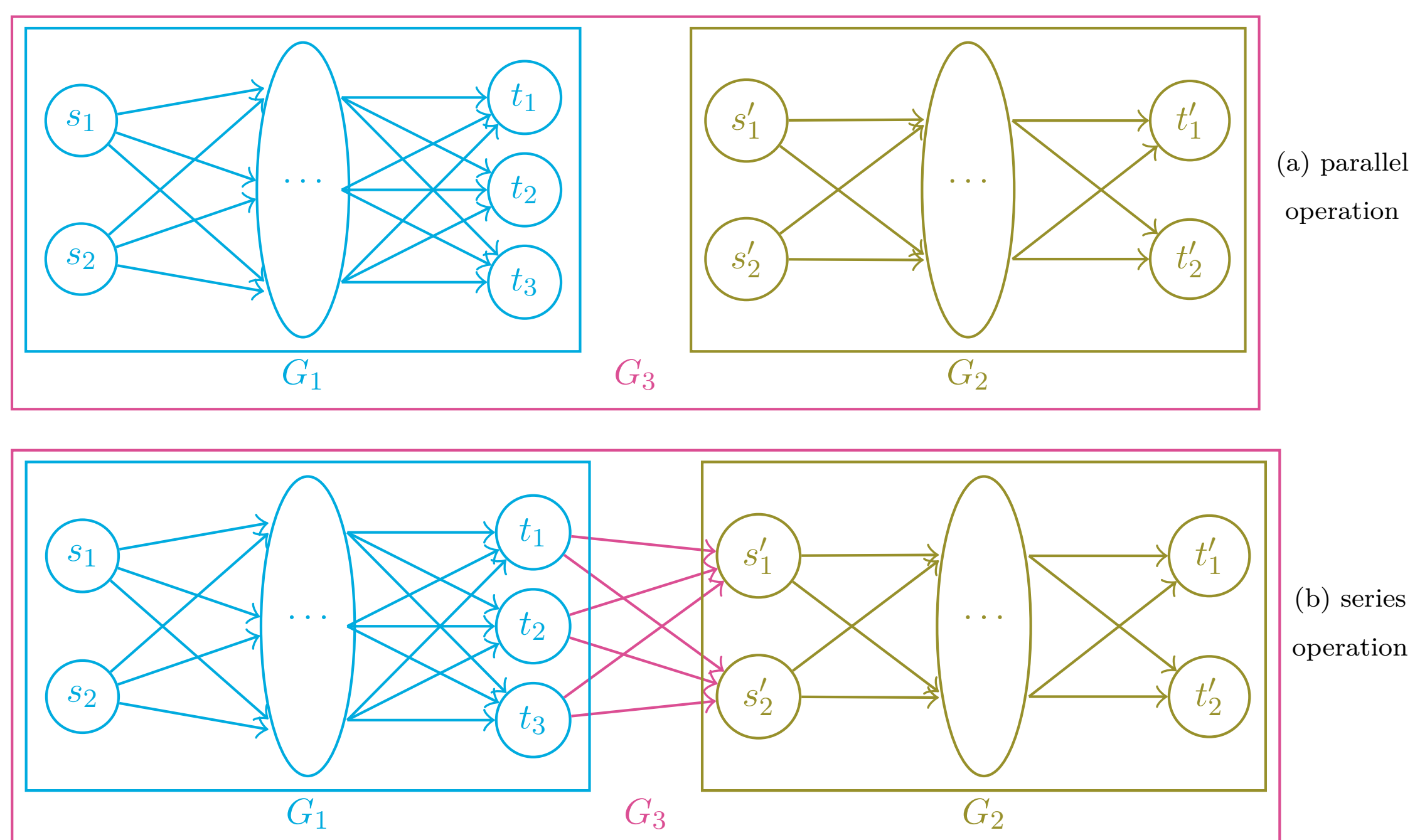
Interval and precedence (di)graphs



$1|interval\ order|\sum w_j C_j$ is \mathcal{NP} -hard, so we cannot directly use the precedence digraph to solve the problem. The goal is to extract a SP-digraph from the precedence digraph.

(Vertex) Series Parallel digraph

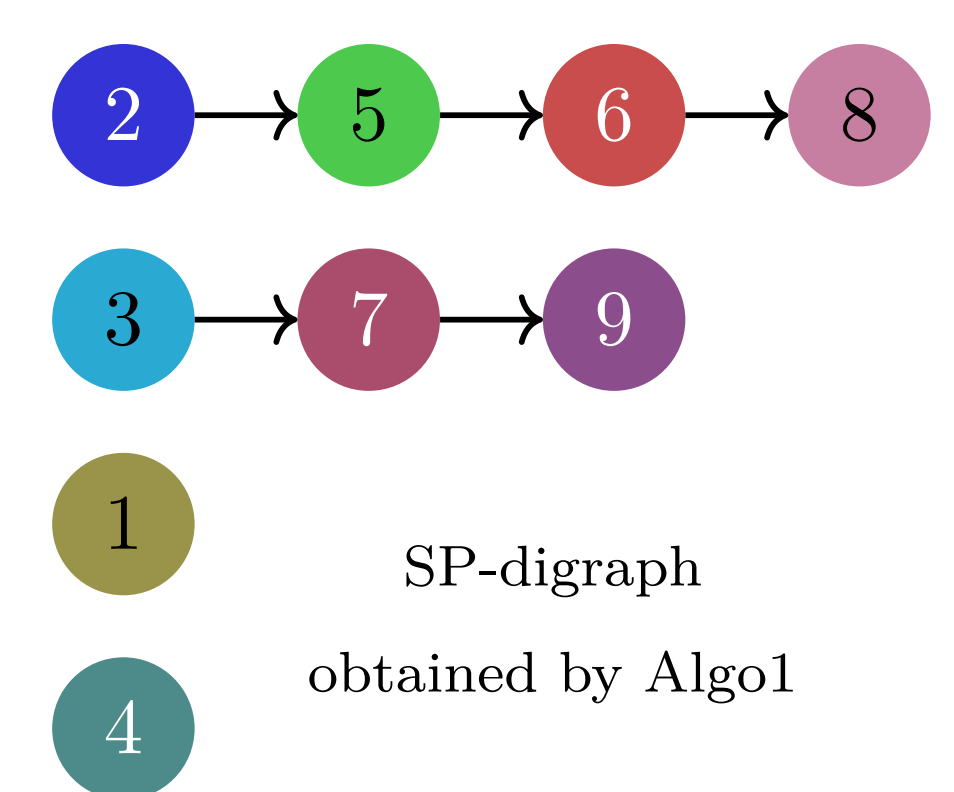
A graph that can be built exclusively by using two operations:



The base case is a single vertex

Algo1: Coloring

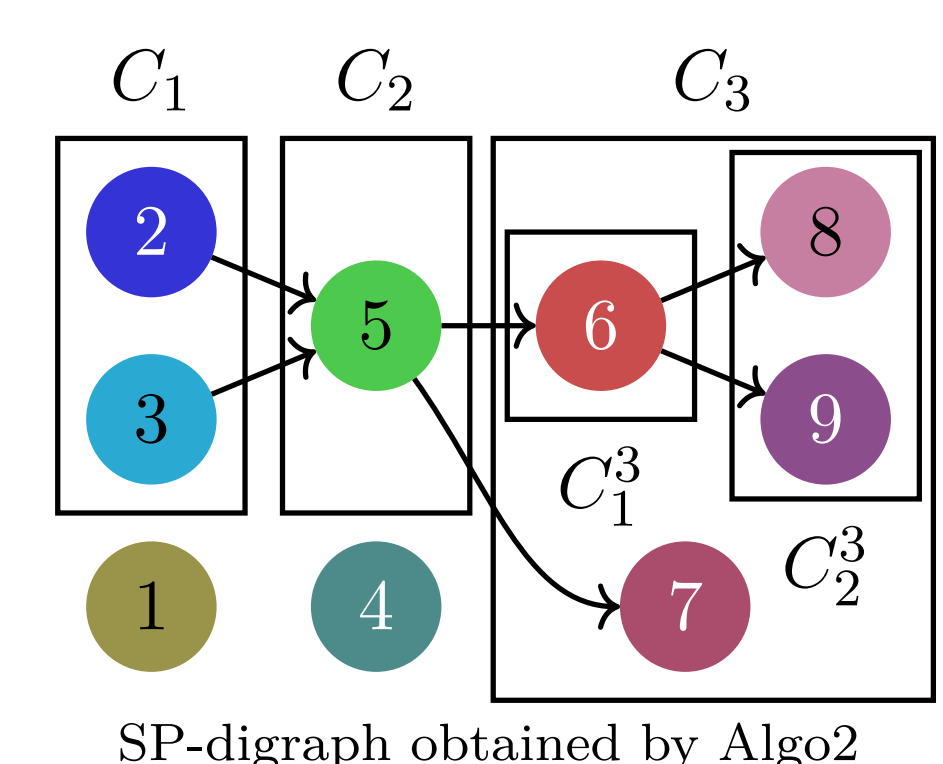
Algo1 (A): Uses the interval graph I . Considers all tasks by increasing order of their release date. Each task is given the smallest label possible. Two adjacent tasks in I cannot have the same label. Each time a task is given a label, it is put in series with other tasks of the same label.



Algo2: Minimal separators

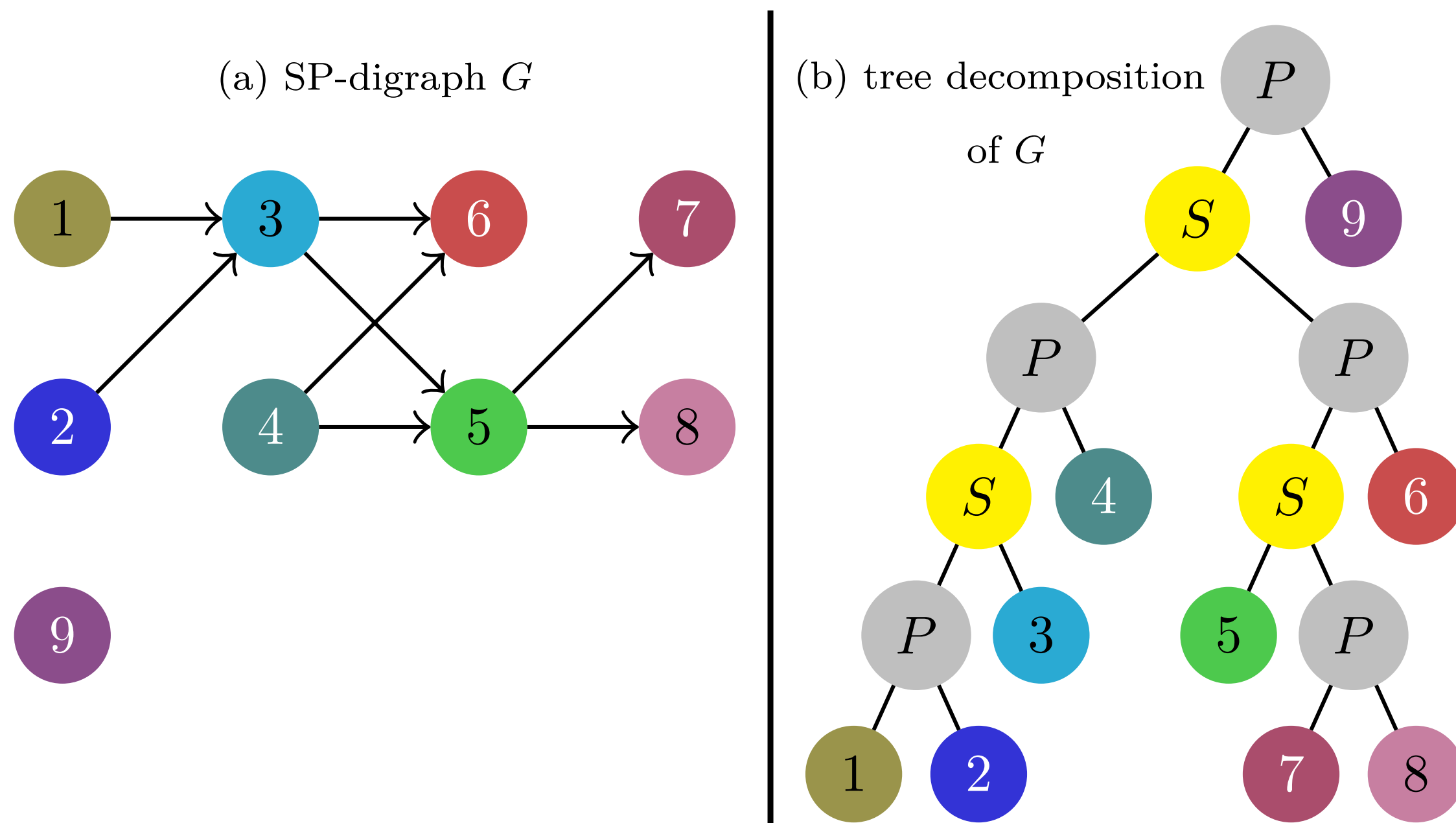
Minimal separator K : A minimal (by inclusion) set of vertex that if deleted from the graph, separates at least two other vertex. Its deletion creates connected components C .

Algo2 (A): Uses the interval graph I . It is recursive. If I is complete, all the vertex are in parallel. Else, a minimal separator is selected and put in parallel with all its connected components C_K . All the $A(C_K)$ are put in series according to their release dates.



Minimal separators used:
 $K_1 = \{1, 4\}$ and $K_2 = \{7\}$

SP-digraph and tree decomposition



References

- [1] A. Kovács and J. C. Beck. A global constraint for total weighted completion time for unary resources. *Constraints* 16, 100–123 (2011).
[2] E.L. Lawler. Sequencing Jobs to Minimize Total Weighted Completion Time Subject to Precedence Constraints. *Algorithmic Aspects of Combinatorics 2*, Annals of Discrete Mathematics, Elsevier, 75-90 (1978).