

A Series-Parallel digraph based relaxation for the COMPLETION constraint

Bonnin Camille^{*, \circ} Thomas Dissaux[†] Malapert Arnaud ^{\circ}
Nicolas Nisse[†] Espinouse Marie-Laure^{*}

^{*}Univ. Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, 38000 Grenoble, France
{camille.bonnin, marie-laure.espinouse}@grenoble-inp.fr

^{\circ} Université 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

Doctoral Program of CP 2023 (26th - 31th August), Toronto, ON, Canada

Thesis main objective

- Improve CP integration of **scheduling** objectives that are not studied a lot in CP, in particular " \sum " objectives
- $\sum C_j$ (flowtime) and $\sum w_j C_j$ (weighted flowtime)

Chosen approach and goal

- **Cost-based domain filtering**: approach that uses filtering algorithms for global constraints which filter combinations of values that cannot lead to a better-cost solution
- A way to do that is to use a lower bound which can be found by using a relaxation for scheduling problems
- Goal: find new such filtering algorithms using new polynomial relaxations

Thesis main objective

- Improve CP integration of **scheduling** objectives that are not studied a lot in CP, in particular " \sum " objectives
- $\sum C_j$ (**flowtime**) and $\sum w_j C_j$ (**weighted flowtime**)

Chosen approach and goal

- **Cost-based domain filtering**: approach that uses filtering algorithms for global constraints which filter combinations of values that cannot lead to a better-cost solution
- A way to do that is to use a lower bound which can be found by using a relaxation for scheduling problems
- Goal: find new such filtering algorithms using new polynomial relaxations

Thesis main objective

- Improve CP integration of **scheduling** objectives that are not studied a lot in CP, in particular " \sum " objectives
- $\sum C_j$ (**flowtime**) and $\sum w_j C_j$ (**weighted flowtime**)

Chosen approach and goal

- **Cost-based domain filtering**: approach that uses filtering algorithms for **global constraints** which filter combinations of values that cannot lead to a better-cost solution
- A way to do that is to use a **lower bound** which can be found by using a **relaxation** for scheduling problems
- Goal: find **new** such **filtering algorithms** using **new polynomial relaxations**

Thesis main objective

- Improve CP integration of **scheduling** objectives that are not studied a lot in CP, in particular " \sum " objectives
- $\sum C_j$ (**flowtime**) and $\sum w_j C_j$ (**weighted flowtime**)

Chosen approach and goal

- **Cost-based domain filtering**: approach that uses filtering algorithms for **global constraints** which filter combinations of values that cannot lead to a better-cost solution
- A way to do that is to use a **lower bound** which can be found by using a **relaxation** for scheduling problems
- Goal: find **new** such **filtering algorithms** using **new polynomial relaxations**

Thesis main objective

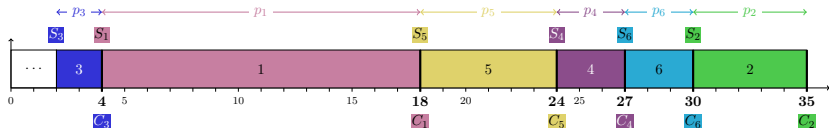
- Improve CP integration of **scheduling** objectives that are not studied a lot in CP, in particular " \sum " objectives
- $\sum C_j$ (**flowtime**) and $\sum w_j C_j$ (**weighted flowtime**)

Chosen approach and goal

- **Cost-based domain filtering**: approach that uses filtering algorithms for **global constraints** which filter combinations of values that cannot lead to a better-cost solution
- A way to do that is to use a **lower bound** which can be found by using a **relaxation** for scheduling problems
- Goal: find **new** such **filtering algorithms** using **new polynomial relaxations**

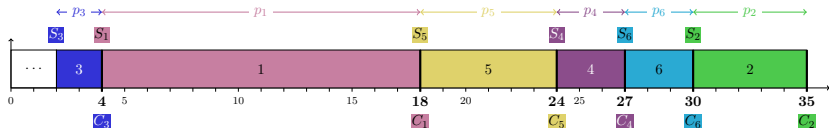
The COMPLETION constraint

- Defined by Kovács and Beck in 2011
- $\text{COMPLETION}([S_1, \dots, S_n], [p_1, \dots, p_n], [w_1, \dots, w_n], C)$
 $\stackrel{\text{def}}{\iff} ((S_i + p_i \leq S_j \vee S_j + p_j \leq S_i) \wedge \sum_i w_i (S_i + p_i) = C)$,
 where $C = \sum_i w_i C_i$
- The filtering algorithm uses a relaxation: $1 |r_j; pmtn| \sum w_i M_i$



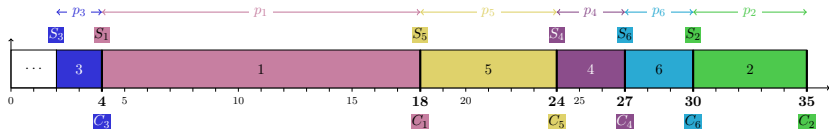
The COMPLETION constraint

- Defined by Kovács and Beck in 2011
- $\text{COMPLETION}([S_1, \dots, S_n], [p_1, \dots, p_n], [w_1, \dots, w_n], C)$
 $\stackrel{\text{def}}{\iff} ((S_i + p_i \leq S_j \vee S_j + p_j \leq S_i) \wedge \sum_i w_i(S_i + p_i) = C),$
 where $C = \sum_i w_i C_i$
- The filtering algorithm uses a relaxation: $1|r_j; pmtn| \sum w_i M_i$



The COMPLETION constraint

- Defined by Kovács and Beck in 2011
- $\text{COMPLETION}([S_1, \dots, S_n], [p_1, \dots, p_n], [w_1, \dots, w_n], C)$
 $\stackrel{\text{def}}{\iff} ((S_i + p_i \leq S_j \vee S_j + p_j \leq S_i) \wedge \sum_i w_i(S_i + p_i) = C),$
 where $C = \sum_i w_i C_i$
- The filtering algorithm uses a relaxation: $1|r_j; pmtn| \sum w_i M_i$



The COMPLETION constraint

- Defined by Kovács and Beck in 2011
- $\text{COMPLETION}([S_1, \dots, S_n], [p_1, \dots, p_n], [w_1, \dots, w_n], C)$
 $\stackrel{\text{def}}{\iff} ((S_i + p_i \leq S_j \vee S_j + p_j \leq S_i) \wedge \sum_i w_i(S_i + p_i) = C),$
 where $C = \sum_i w_i C_i$
- The filtering algorithm uses a relaxation: $1|r_j; pmtn| \sum w_i M_i$

Current work

- Definition of a **global constraint** based on COMPLETION for the non-weighted case (not presented here): FLOWTIME
- A second filtering algorithm for COMPLETION using another relaxation: $1|sp-graph| \sum w_i C_i$

The COMPLETION constraint

- Defined by Kovács and Beck in 2011
- $\text{COMPLETION}([S_1, \dots, S_n], [p_1, \dots, p_n], [w_1, \dots, w_n], C)$
 $\stackrel{\text{def}}{\iff} ((S_i + p_i \leq S_j \vee S_j + p_j \leq S_i) \wedge \sum_i w_i(S_i + p_i) = C),$
 where $C = \sum_i w_i C_i$
- The filtering algorithm uses a relaxation: $1|r_j; pmtn| \sum w_i M_i$

Current work

- Definition of a **global constraint** based on COMPLETION for the non-weighted case (not presented here): FLOWTIME
- A **second filtering algorithm** for COMPLETION using another relaxation: $1|sp-graph| \sum w_i C_i$

(Vertex) Series Parallel (di)graph

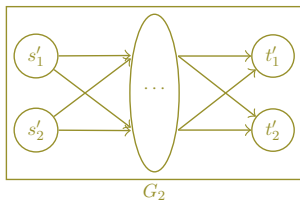
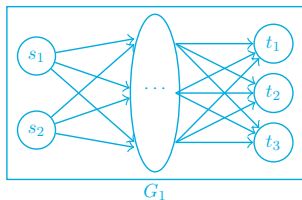
A digraph that can be built exclusively by using two operations starting from single vertex:

(a) Parallel operation

(b) Series operation

(Vertex) Series Parallel (di)graph

A digraph that can be built exclusively by using two operations starting from single vertex:

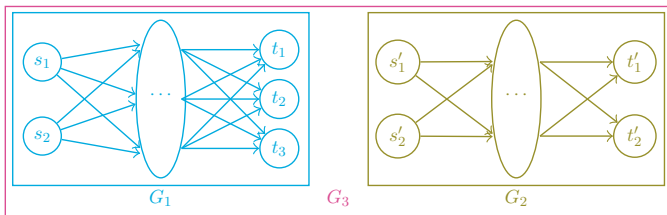


(a) Parallel operation

(b) Series operation

(Vertex) Series Parallel (di)graph

A digraph that can be built exclusively by using two operations starting from single vertex:

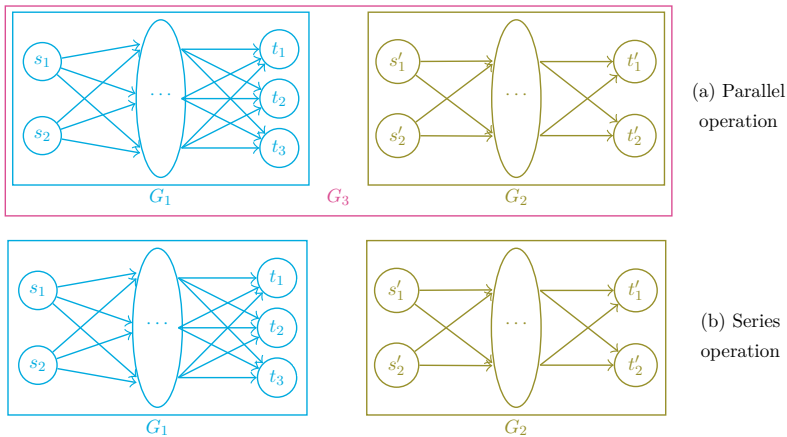


(a) Parallel operation

(b) Series operation

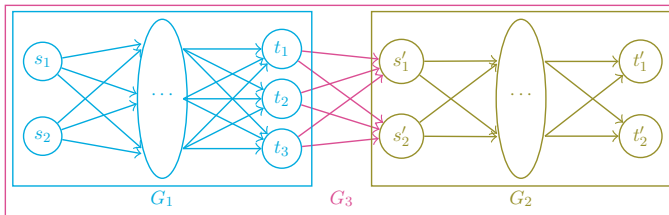
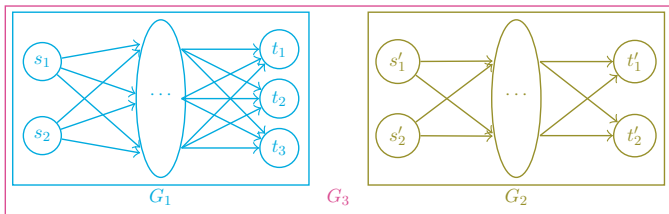
(Vertex) Series Parallel (di)graph

A digraph that can be built exclusively by using two operations starting from single vertex:



(Vertex) Series Parallel (di)graph

A digraph that can be built exclusively by using two operations starting from single vertex:



SP-digraph and tree decomposition

(a) SP-digraph G

(b) Tree decomposition of G

(a) SP-digraph G

1

(b) Tree decomposition of G

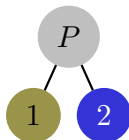
1

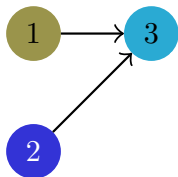
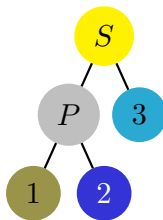
SP-digraph and tree decomposition

(a) SP-digraph G

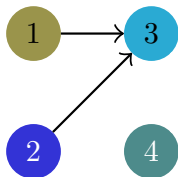


(b) Tree decomposition of G

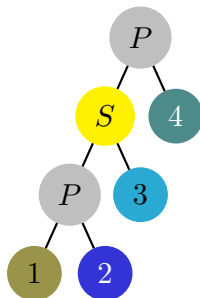


(a) SP-digraph G (b) Tree decomposition of G 

(a) SP-digraph G

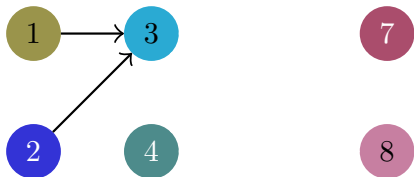


(b) Tree decomposition of G

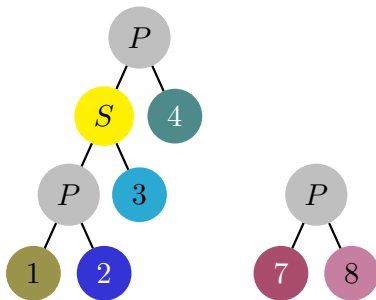


SP-digraph and tree decomposition

(a) SP-digraph G

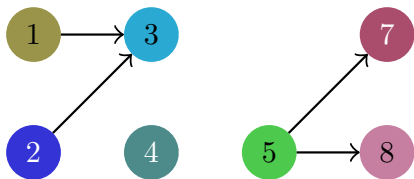


(b) Tree decomposition of G

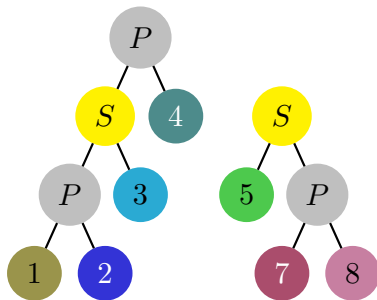


SP-digraph and tree decomposition

(a) SP-digraph G

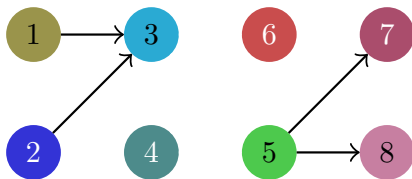


(b) Tree decomposition of G

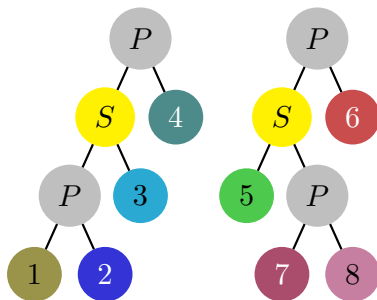


SP-digraph and tree decomposition

(a) SP-digraph G

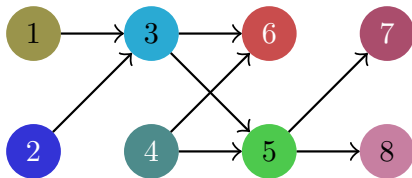


(b) Tree decomposition of G

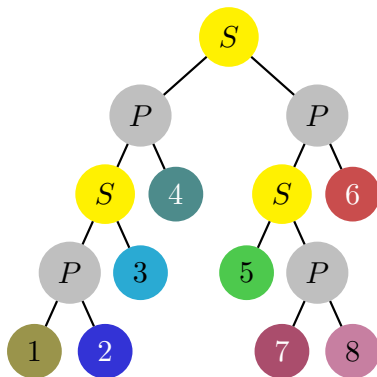


SP-digraph and tree decomposition

(a) SP-digraph G



(b) Tree decomposition of G



Polynomial relaxation

- General problem $1|r_j; d_j| \sum w_j C_j$ is \mathcal{NP} -hard
- Polynomial relaxation of COMPLETION: $1|r_j; pmtn| \sum w_j M_j$
- $1|sp-graph| \sum w_j C_j$ can be solved in polynomial time ($\mathcal{O}(n \log n)$) by Lawler's algorithm using the tree decomposition of SP-digraphs

Cost-based domain filtering approach

- First step: Extract an instance of $1|sp-graph| \sum w_j C_j$ from an instance of $1|r_j; d_j| \sum w_j C_j$ (currently)
- Second step: Use it to compute a lower bound for a new filtering algorithm for COMPLETION (prospect)

Polynomial relaxation

- General problem $1|r_j; d_j| \sum w_j C_j$ is \mathcal{NP} -hard
- Polynomial relaxation of COMPLETION: $1|r_j; pmt_n| \sum w_j M_j$
- $1|sp-graph| \sum w_j C_j$ can be solved in polynomial time ($\mathcal{O}(n \log n)$) by Lawler's algorithm using the tree decomposition of SP-digraphs

Cost-based domain filtering approach

- First step: Extract an instance of $1|sp-graph| \sum w_j C_j$ from an instance of $1|r_j; d_j| \sum w_j C_j$ (currently)
- Second step: Use it to compute a lower bound for a new cost-based domain filtering algorithm for COMPLETION (prospect)

Polynomial relaxation

- General problem $1|r_j; d_j| \sum w_j C_j$ is \mathcal{NP} -hard
- Polynomial relaxation of COMPLETION: $1|r_j; pmtn| \sum w_j M_j$
- $1|sp-graph| \sum w_j C_j$ can be solved in polynomial time ($\mathcal{O}(n \log n)$) by Lawler's algorithm using the tree decomposition of SP-digraphs

Cost-based domain filtering approach

- First step: Extract an instance of $1|sp-graph| \sum w_j C_j$ from an instance of $1|r_j; d_j| \sum w_j C_j$ (currently)
- Second step: Use it to compute a lower bound for a new cost-based filtering algorithm for COMPLETION (prospect)

Polynomial relaxation

- General problem $1|r_j; d_j| \sum w_j C_j$ is \mathcal{NP} -hard
- Polynomial relaxation of COMPLETION: $1|r_j; pmt_n| \sum w_j M_j$
- $1|sp-graph| \sum w_j C_j$ can be solved in polynomial time ($\mathcal{O}(n \log n)$) by Lawler's algorithm using the tree decomposition of SP-digraphs

Cost-based domain filtering approach

- **First step:** Extract an instance of $1|sp-graph| \sum w_j C_j$ from an instance of $1|r_j; d_j| \sum w_j C_j$ (currently)
- **Second step:** Use it to compute a lower bound for a new filtering algorithm for COMPLETION (prospect)

Polynomial relaxation

- General problem $1|r_j; d_j| \sum w_j C_j$ is \mathcal{NP} -hard
- Polynomial relaxation of COMPLETION: $1|r_j; pmt_n| \sum w_j M_j$
- $1|sp-graph| \sum w_j C_j$ can be solved in polynomial time ($\mathcal{O}(n \log n)$) by Lawler's algorithm using the tree decomposition of SP-digraphs

Cost-based domain filtering approach

- **First step:** Extract an instance of $1|sp-graph| \sum w_j C_j$ from an instance of $1|r_j; d_j| \sum w_j C_j$ (currently)
- **Second step:** Use it to compute a lower bound for a new filtering algorithm for COMPLETION (prospect)

Interval and precedence graphs

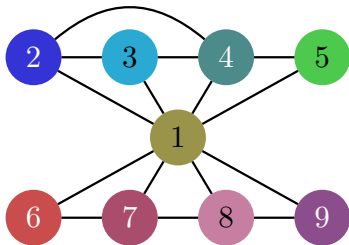


(a) Intervals representing the time windows of the tasks

Interval and precedence graphs



(a) Intervals representing the time windows of the tasks

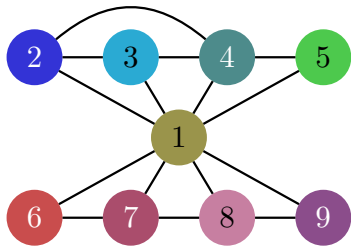


(b) Interval graph

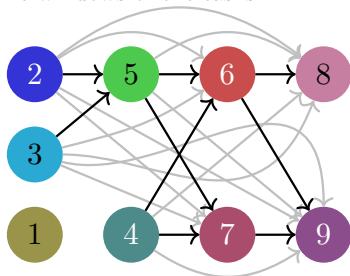
Interval and precedence graphs



(a) Intervals representing the time windows of the tasks



(b) Interval graph



(c) Precedences digraph

Non-preemptive relaxations of $1|r_j; d_j| \sum w_j C_j$

- General problem $1|r_j; d_j| \sum w_j C_j$ is \mathcal{NP} -hard
- Relaxation $1|interval\ order| \sum w_j C_j$ is also \mathcal{NP} -hard
- Relaxation $1|sp-graph| \sum w_j C_j$ is polynomial

Consequence

We need to extract a SP-digraph of precedences from the time windows of the tasks \Rightarrow 2 algorithms

Non-preemptive relaxations of $1|r_j; d_j| \sum w_j C_j$

- General problem $1|r_j; d_j| \sum w_j C_j$ is \mathcal{NP} -hard
- Relaxation $1|interval\ order| \sum w_j C_j$ is also \mathcal{NP} -hard
- Relaxation $1|sp-graph| \sum w_j C_j$ is polynomial

Consequence

We need to extract a SP-digraph of precedences from the time windows of the tasks \Rightarrow 2 algorithms

Non-preemptive relaxations of $1|r_j; d_j| \sum w_j C_j$

- General problem $1|r_j; d_j| \sum w_j C_j$ is \mathcal{NP} -hard
- Relaxation $1|interval\ order| \sum w_j C_j$ is also \mathcal{NP} -hard
- Relaxation $1|sp-graph| \sum w_j C_j$ is polynomial

Consequence

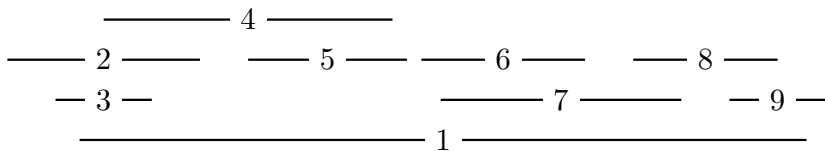
We need to extract a SP-digraph of precedences from the time windows of the tasks \Rightarrow 2 algorithms

Non-preemptive relaxations of $1|r_j; d_j| \sum w_j C_j$

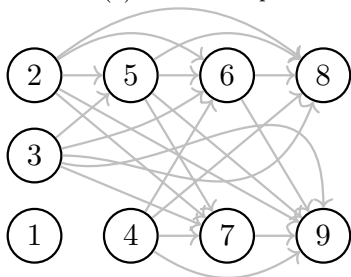
- General problem $1|r_j; d_j| \sum w_j C_j$ is \mathcal{NP} -hard
- Relaxation $1|interval\ order| \sum w_j C_j$ is also \mathcal{NP} -hard
- Relaxation $1|sp-graph| \sum w_j C_j$ is polynomial

Consequence

We need to extract a SP-digraph of precedences from the time windows of the tasks \Rightarrow 2 algorithms



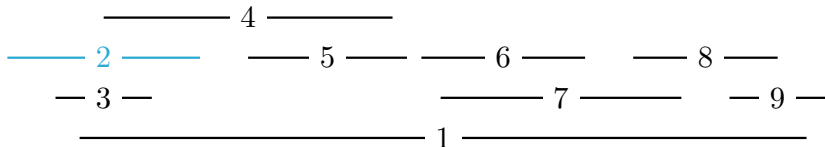
(a) Intervals representing the time windows of the tasks



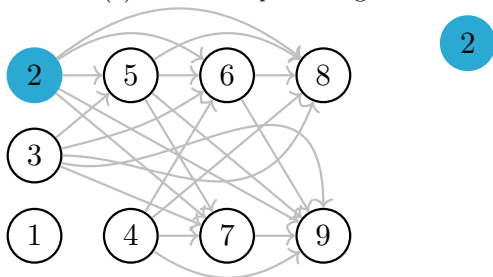
(b) Precedences digraph

(c) SP-digraph
obtained by CBA

Colouring Based Algorithm

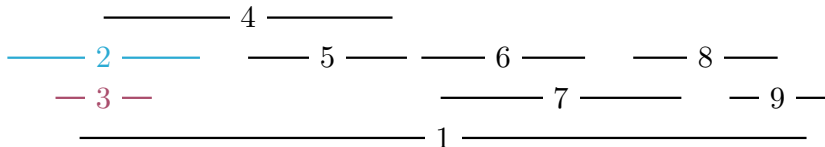


(a) Intervals representing the time windows of the tasks

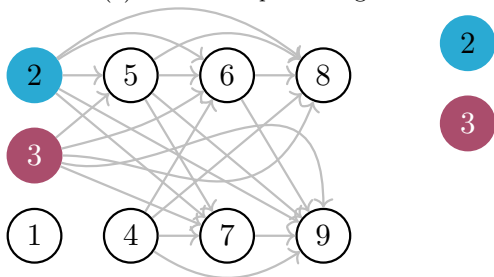


(b) Precedences digraph

(c) SP-digraph
obtained by CBA



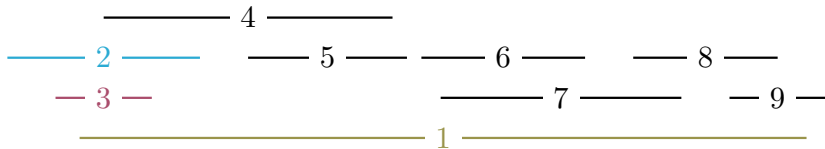
(a) Intervals representing the time windows of the tasks



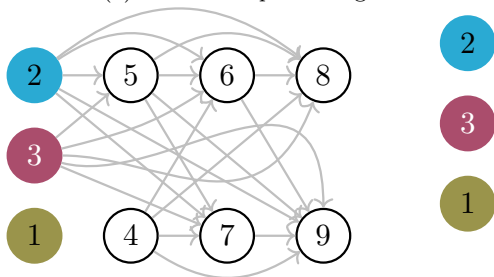
(b) Precedences digraph

(c) SP-digraph
obtained by CBA

Colouring Based Algorithm

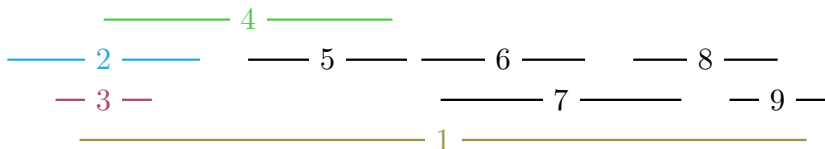


(a) Intervals representing the time windows of the tasks

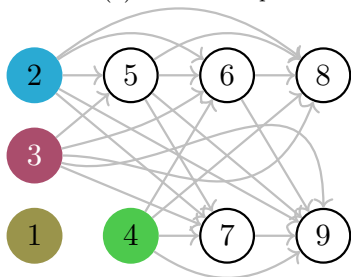


(b) Precedences digraph

(c) SP-digraph
obtained by CBA



(a) Intervals representing the time windows of the tasks



(b) Precedences digraph

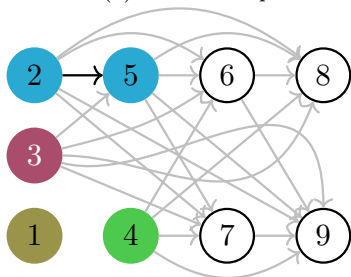


(c) SP-digraph
obtained by CBA

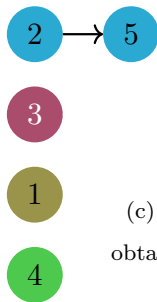
Colouring Based Algorithm



(a) Intervals representing the time windows of the tasks

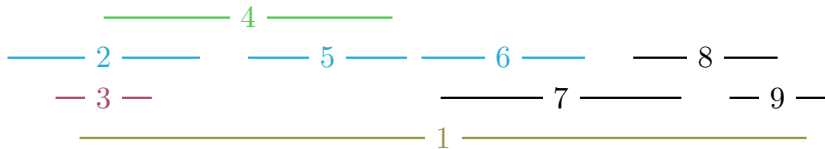


(b) Precedences digraph

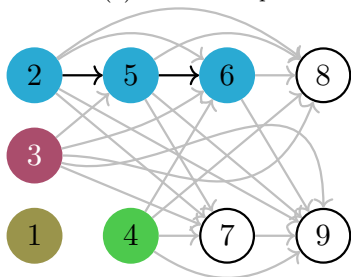


(c) SP-digraph
obtained by CBA

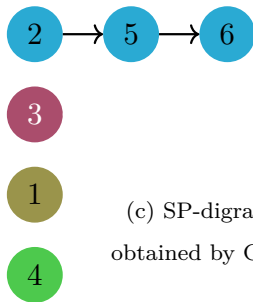
Colouring Based Algorithm



(a) Intervals representing the time windows of the tasks



(b) Precedences digraph

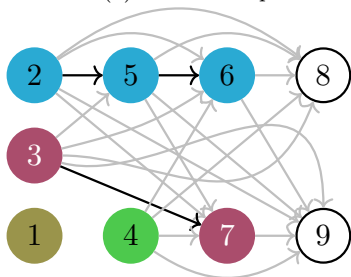


(c) SP-digraph
obtained by CBA

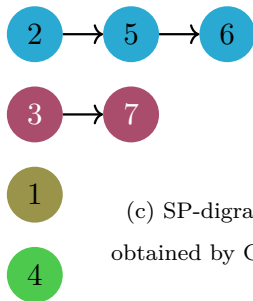
Colouring Based Algorithm



(a) Intervals representing the time windows of the tasks



(b) Precedences digraph

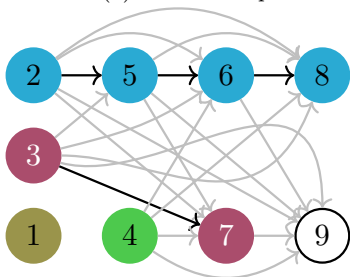


(c) SP-digraph
obtained by CBA

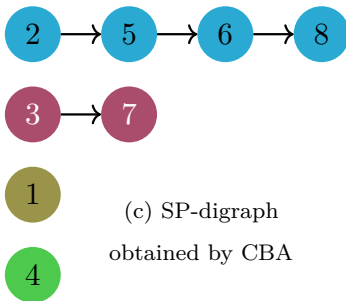
Colouring Based Algorithm



(a) Intervals representing the time windows of the tasks



(b) Precedences digraph

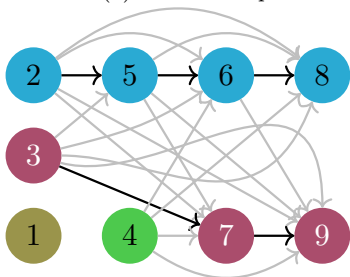


(c) SP-digraph
obtained by CBA

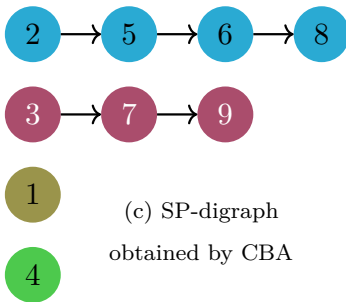
Colouring Based Algorithm



(a) Intervals representing the time windows of the tasks



(b) Precedences digraph

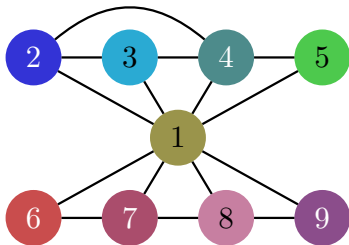


(c) SP-digraph
obtained by CBA

Minimal Separators Based Algorithm



(a) Intervals representing the time windows of the tasks

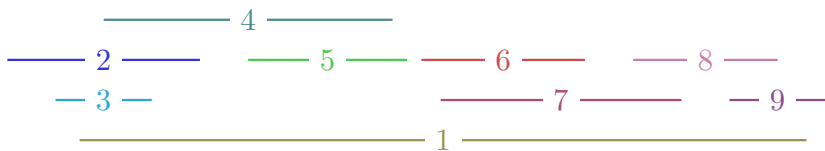


(b) Interval graph

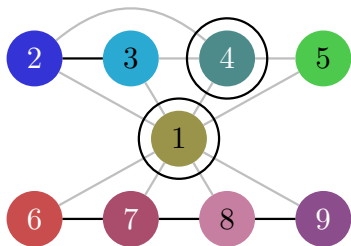
(c) SP-digraph obtained by MSBA

Minimal separators:

Minimal Separators Based Algorithm

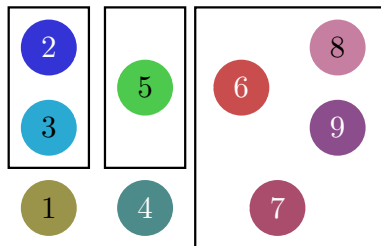


(a) Intervals representing the time windows of the tasks



(b) Interval graph

$C_1 \rightarrow C_2 \rightarrow C_3$



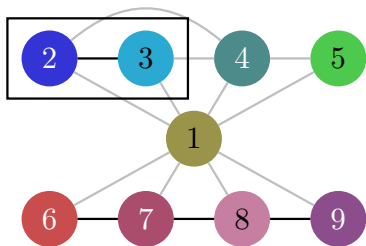
(c) SP-digraph obtained by MSBA

Minimal separators: $K_1 = \{1, 4\}$

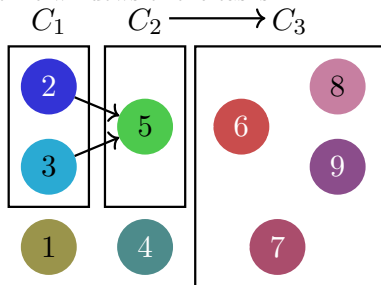
Minimal Separators Based Algorithm



(a) Intervals representing the time windows of the tasks



(b) Interval graph



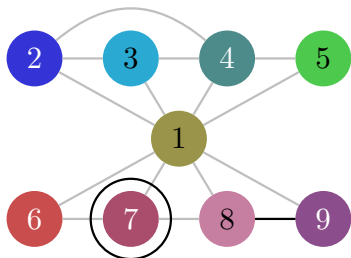
(c) SP-digraph obtained by MSBA

Minimal separators: $K_1 = \{1, 4\}$

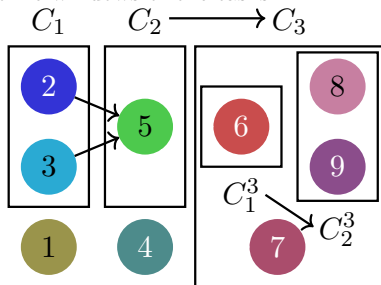
Minimal Separators Based Algorithm



(a) Intervals representing the time windows of the tasks



(b) Interval graph



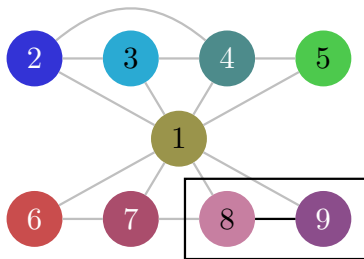
(c) SP-digraph obtained by MSBA

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

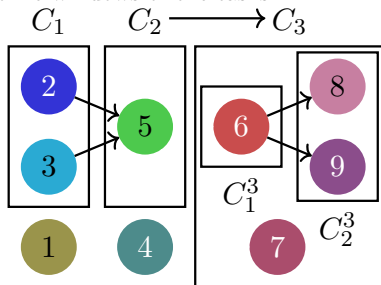
Minimal Separators Based Algorithm



(a) Intervals representing the time windows of the tasks



(b) Interval graph



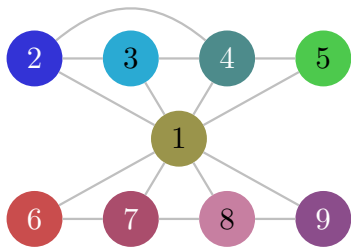
(c) SP-digraph obtained by MSBA

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

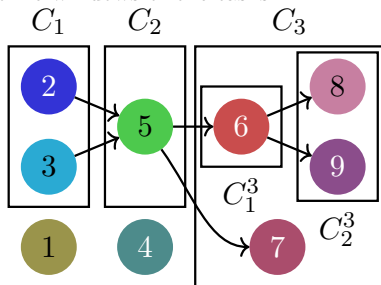
Minimal Separators Based Algorithm



(a) Intervals representing the time windows of the tasks



(b) Interval graph



(c) SP-digraph obtained by MSBA

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

Conclusion

- Relaxation $1|sp-graph| \sum w_j C_j$ is polynomial
- 2 algorithms for extracting an instance of $1|sp-graph| \sum w_j C_j$ from $1|r_j; d_j| \sum w_j C_j$

Prospects

- Implement new filtering algorithms for COMPLETION
- Compare the filtering algorithms
- We defined an extension of SP-digraphs: the Pseudo-SP digraphs (PSP-digraphs)
 \Rightarrow Complexity of $1|psp-graph| \sum w_j C_j$?

Conclusion

- Relaxation $1|sp-graph| \sum w_j C_j$ is polynomial
- 2 algorithms for extracting an instance of $1|sp-graph| \sum w_j C_j$ from $1|r_j; d_j| \sum w_j C_j$

Prospects

- Implement new filtering algorithms for COMPLETION
- Compare the filtering algorithms
- We defined an extension of SP-digraphs: the Pseudo-SP digraphs (PSP-digraphs)
 \Rightarrow Complexity of $1|psp-graph| \sum w_j C_j$?

Conclusion

- Relaxation $1|sp-graph| \sum w_j C_j$ is polynomial
- 2 algorithms for extracting an instance of $1|sp-graph| \sum w_j C_j$ from $1|r_j; d_j| \sum w_j C_j$

Prospects

- Implement new filtering algorithms for COMPLETION
- Compare the filtering algorithms
- We defined an extension of SP-digraphs: the Pseudo-SP digraphs (PSP-digraphs)
 \Rightarrow Complexity of $1|psp-graph| \sum w_j C_j$?

Conclusion

- Relaxation $1|sp-graph| \sum w_j C_j$ is polynomial
- 2 algorithms for extracting an instance of $1|sp-graph| \sum w_j C_j$ from $1|r_j; d_j| \sum w_j C_j$

Prospects

- Implement new filtering algorithms for COMPLETION
- Compare the filtering algorithms
- We defined an extension of SP-digraphs: the Pseudo-SP digraphs (PSP-digraphs)
 \Rightarrow Complexity of $1|psp-graph| \sum w_j C_j$?

Conclusion

- Relaxation $1|sp-graph| \sum w_j C_j$ is polynomial
- 2 algorithms for extracting an instance of $1|sp-graph| \sum w_j C_j$ from $1|r_j; d_j| \sum w_j C_j$

Prospects

- Implement new filtering algorithms for COMPLETION
- Compare the filtering algorithms
- We defined an extension of SP-digraphs: the Pseudo-SP digraphs (PSP-digraphs)
 \Rightarrow Complexity of $1|psp-graph| \sum w_j C_j$?