Learning Effect and Compound Activities in High Multiplicity RCPSP : Application to Satellite Production

Duc Anh Le (1) – Stéphanie Roussel (1) – Christophe Lecoutre (2) – Anouck Chan (1)

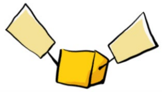
(1) ONERA – The French Aerospace Lab, DTIS/EDO

(2) Artois University, CRIL

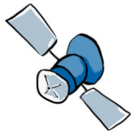
duc_anh.le@onera.fr

Table of Contents

Satellite Assembly Line



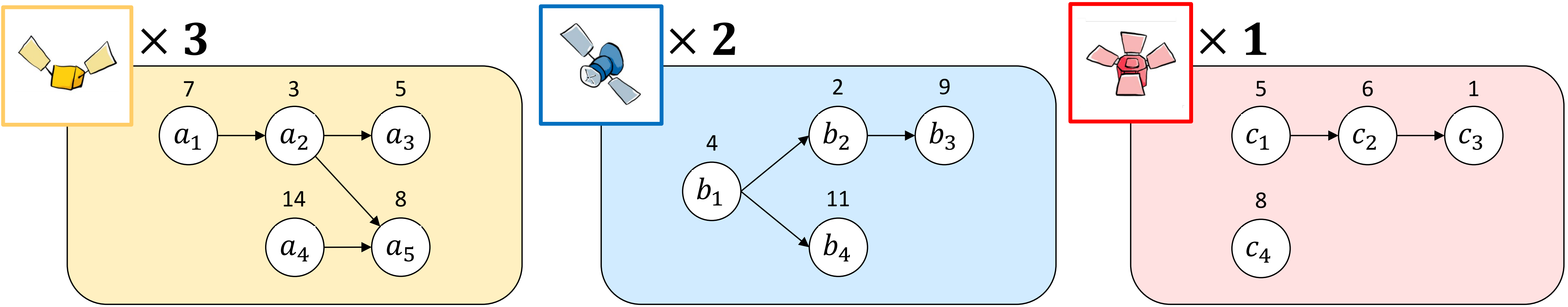
$\times$ 3



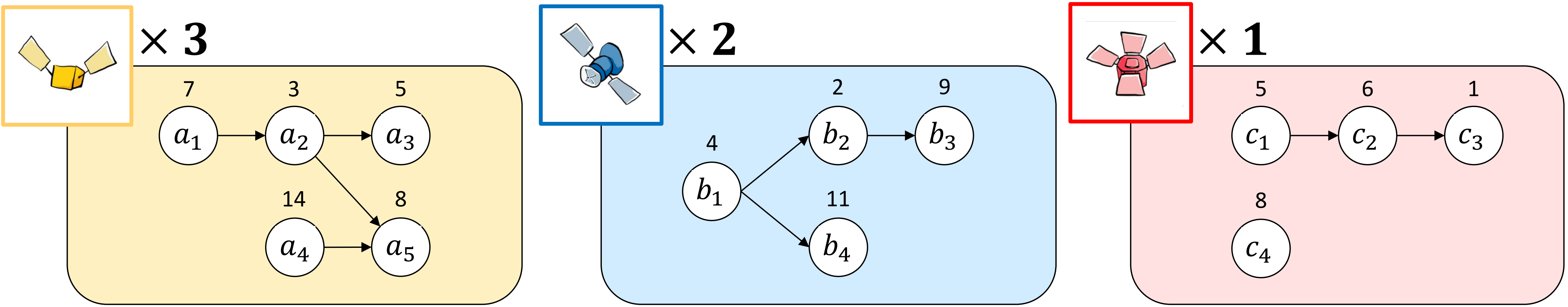
$\times$ 2

$\times$ 1

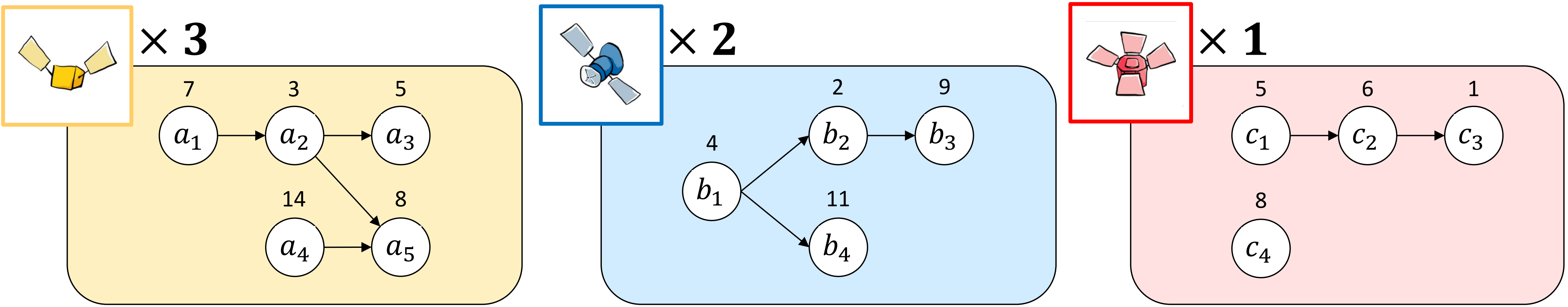
Satellite Assembly Line

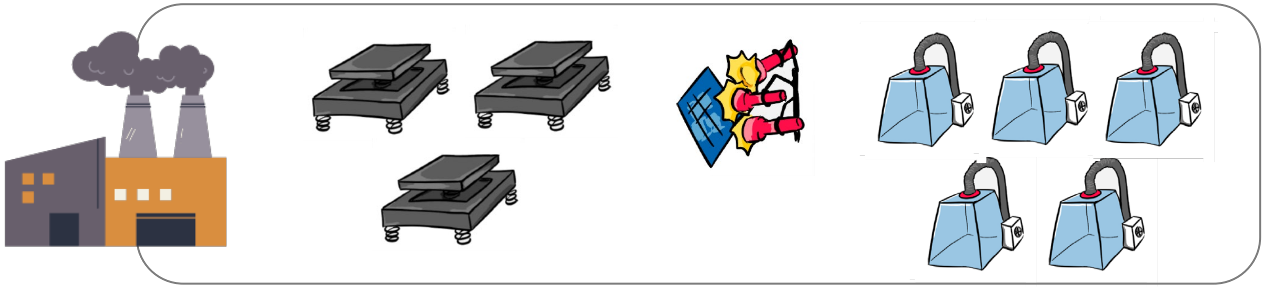


Satellite Assembly Line

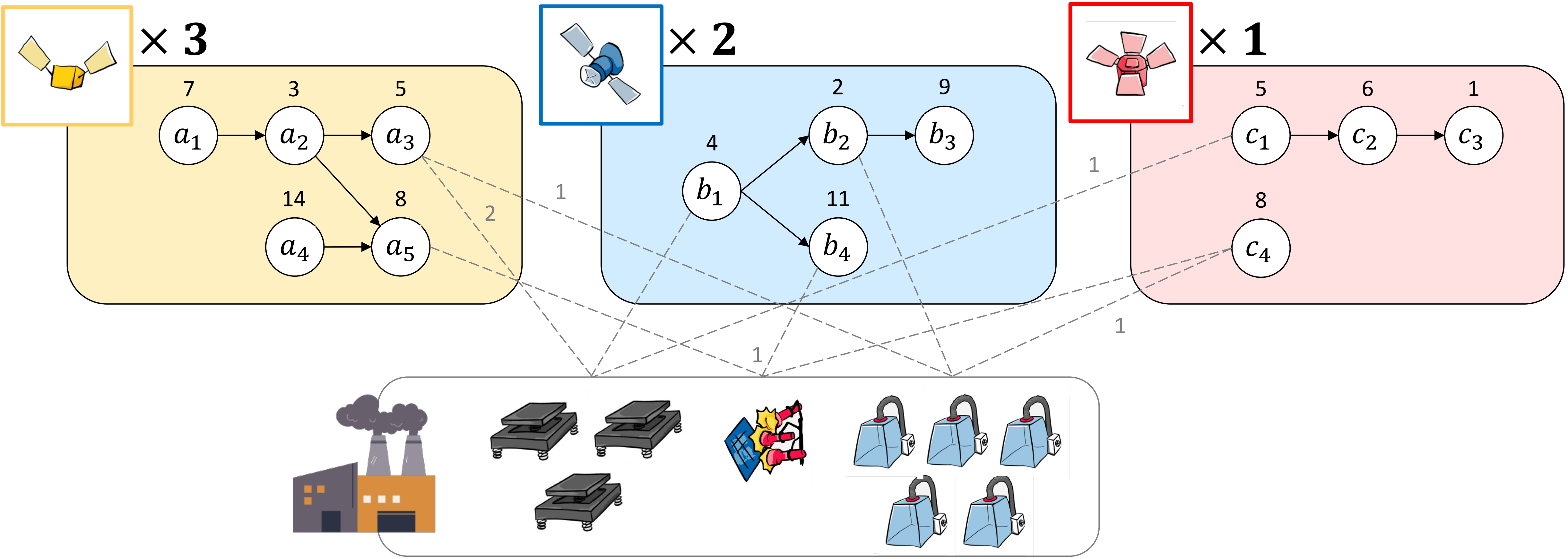


Satellite Assembly Line

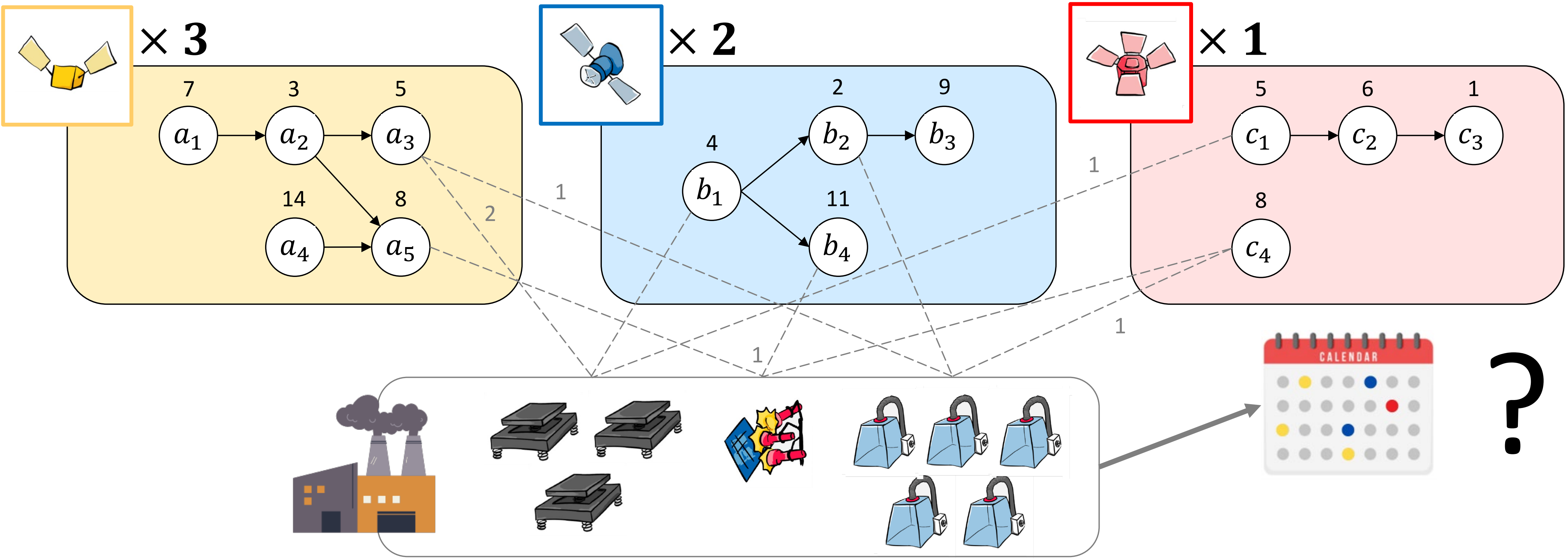




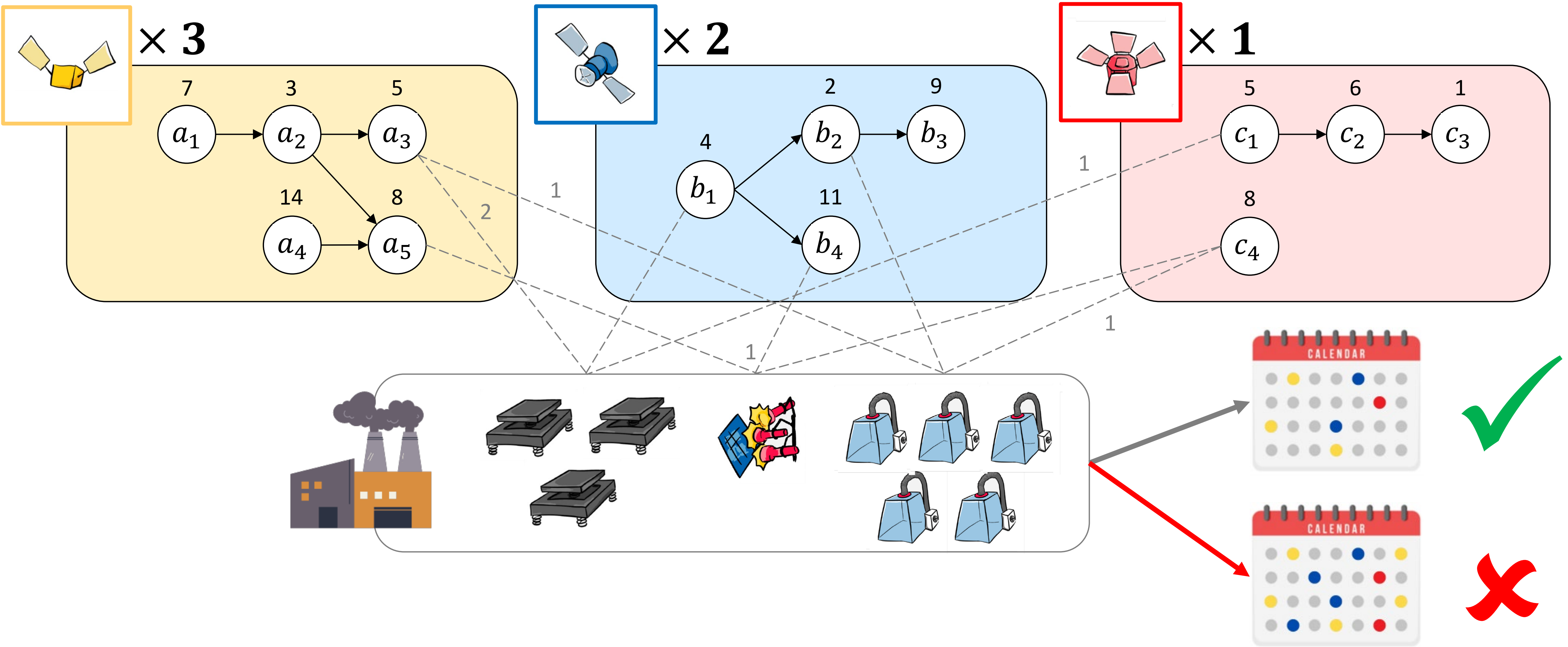
Satellite Assembly Line



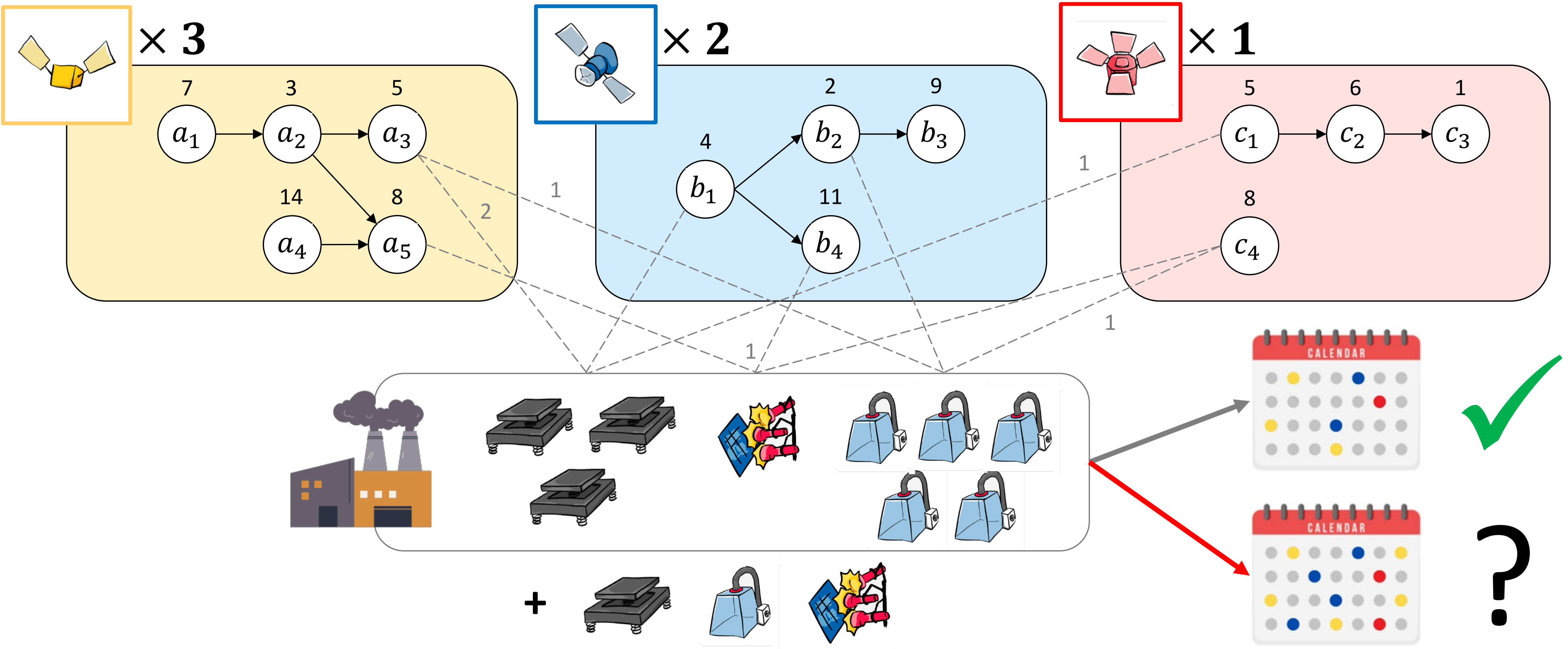
Satellite Assembly Line



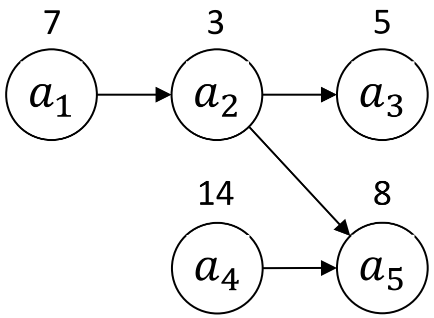
Satellite Assembly Line



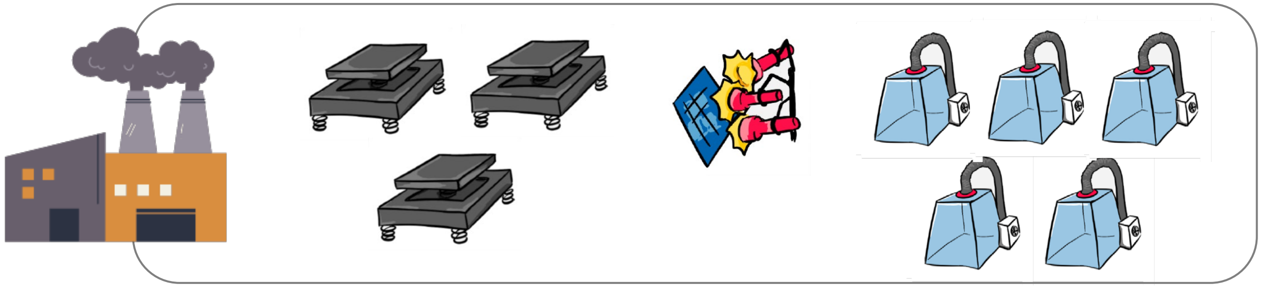
Satellite Assembly Line



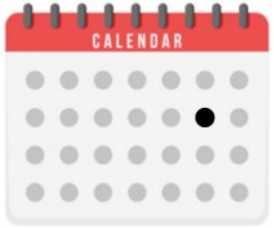
High Multiplicity RCPSP



Activities



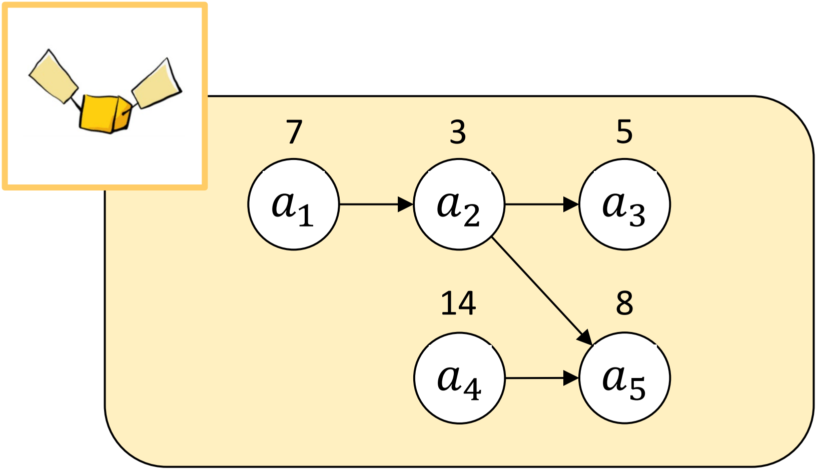
Resources



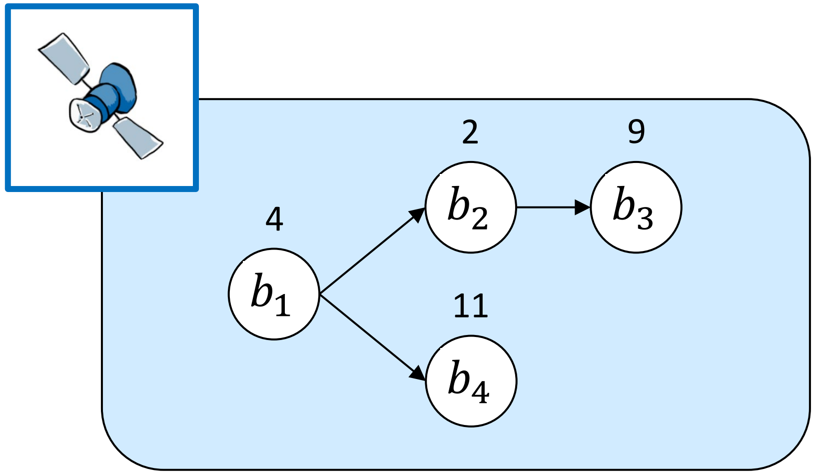
Due date

RCPSP

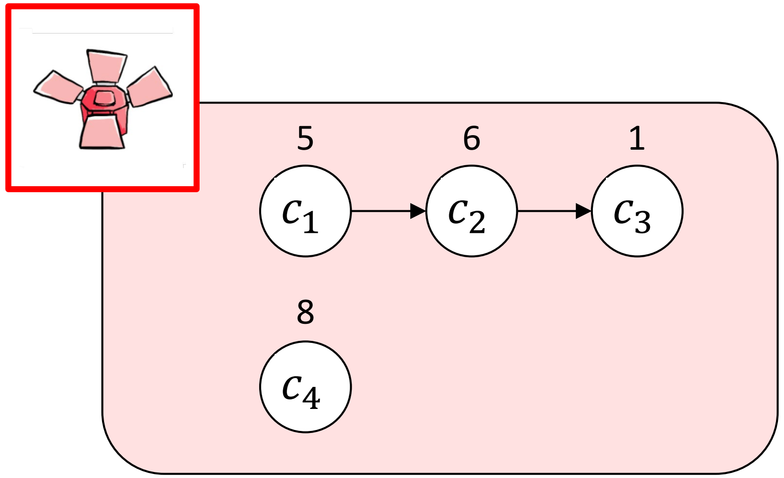
High Multiplicity RCPSP



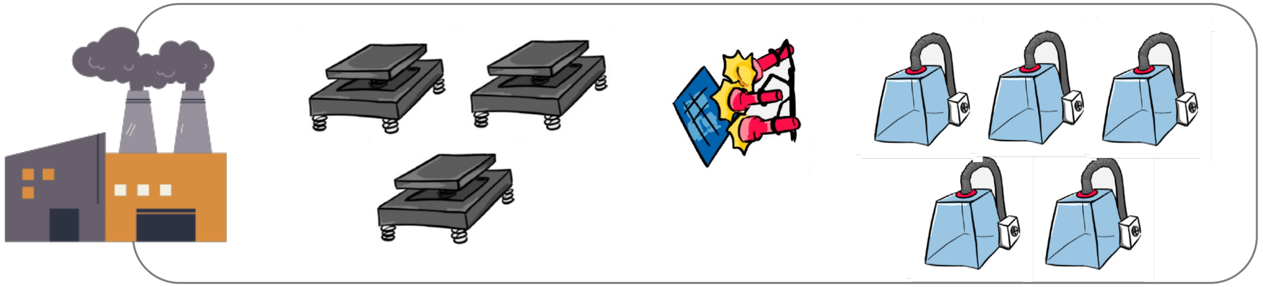
Project 1



Project 2



Project 3

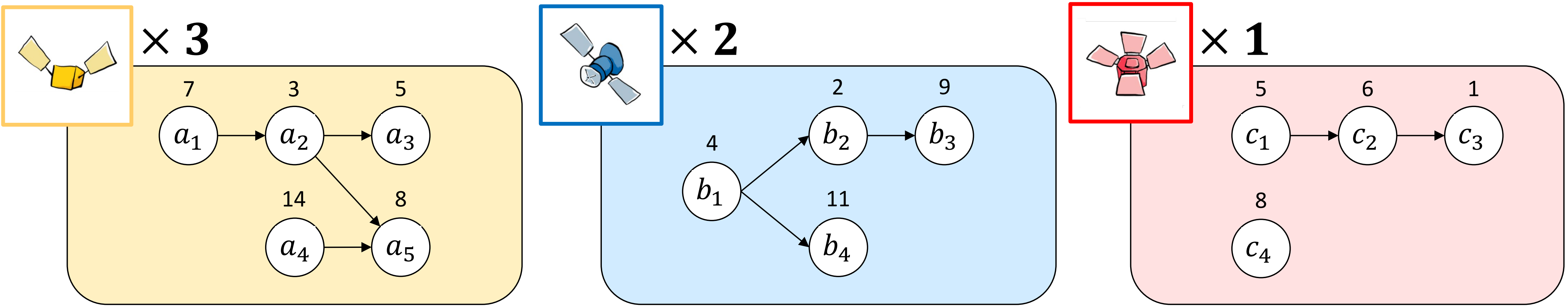


Resources

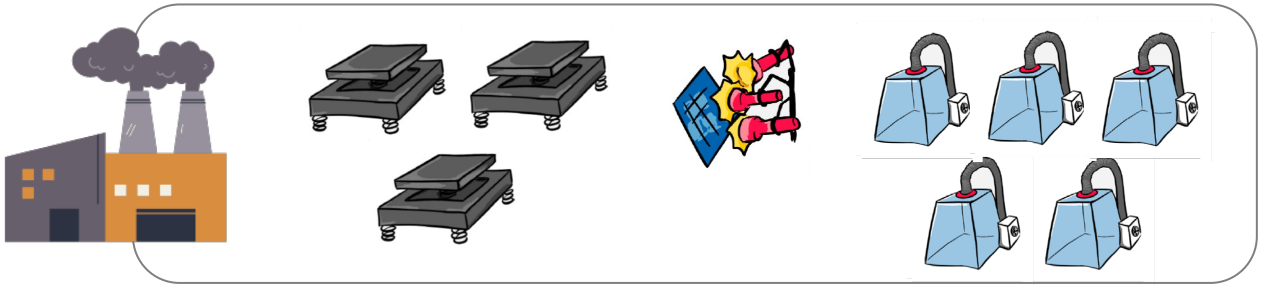
Due dates

MP-RCPSP

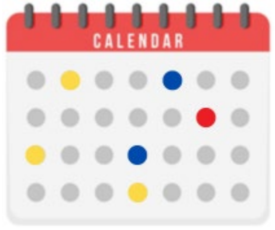
High Multiplicity RCPSP



Class 1 **Class 2** **Class 3**



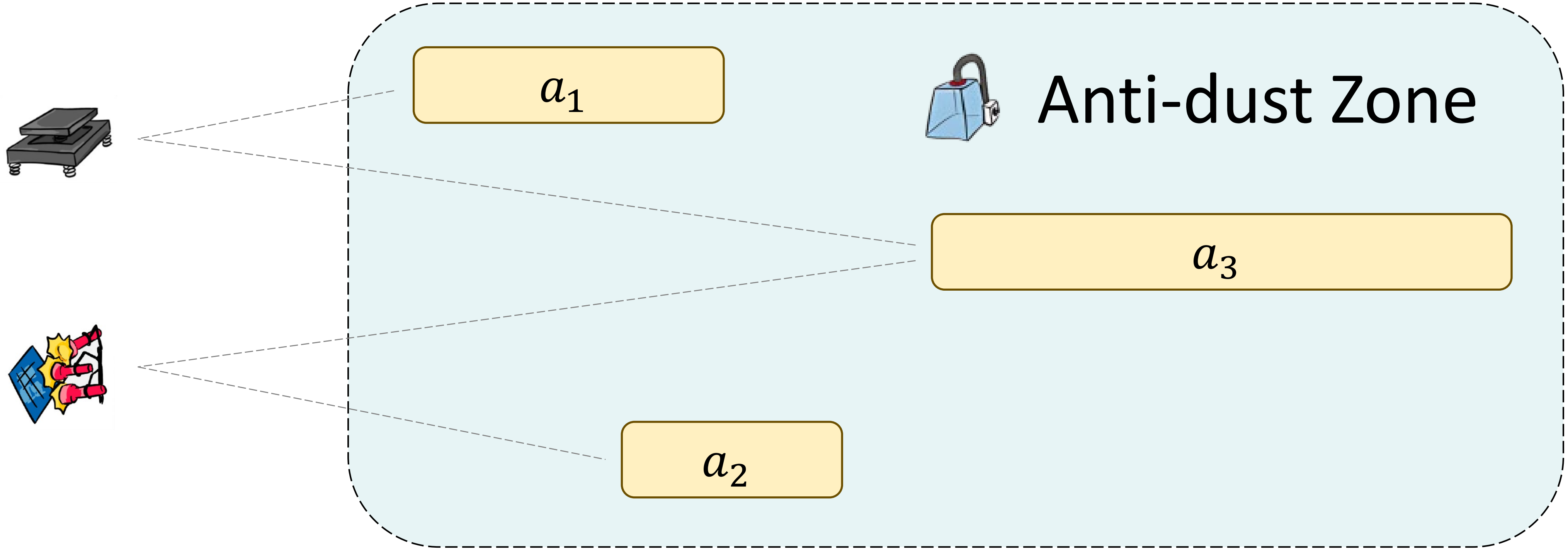
Resources



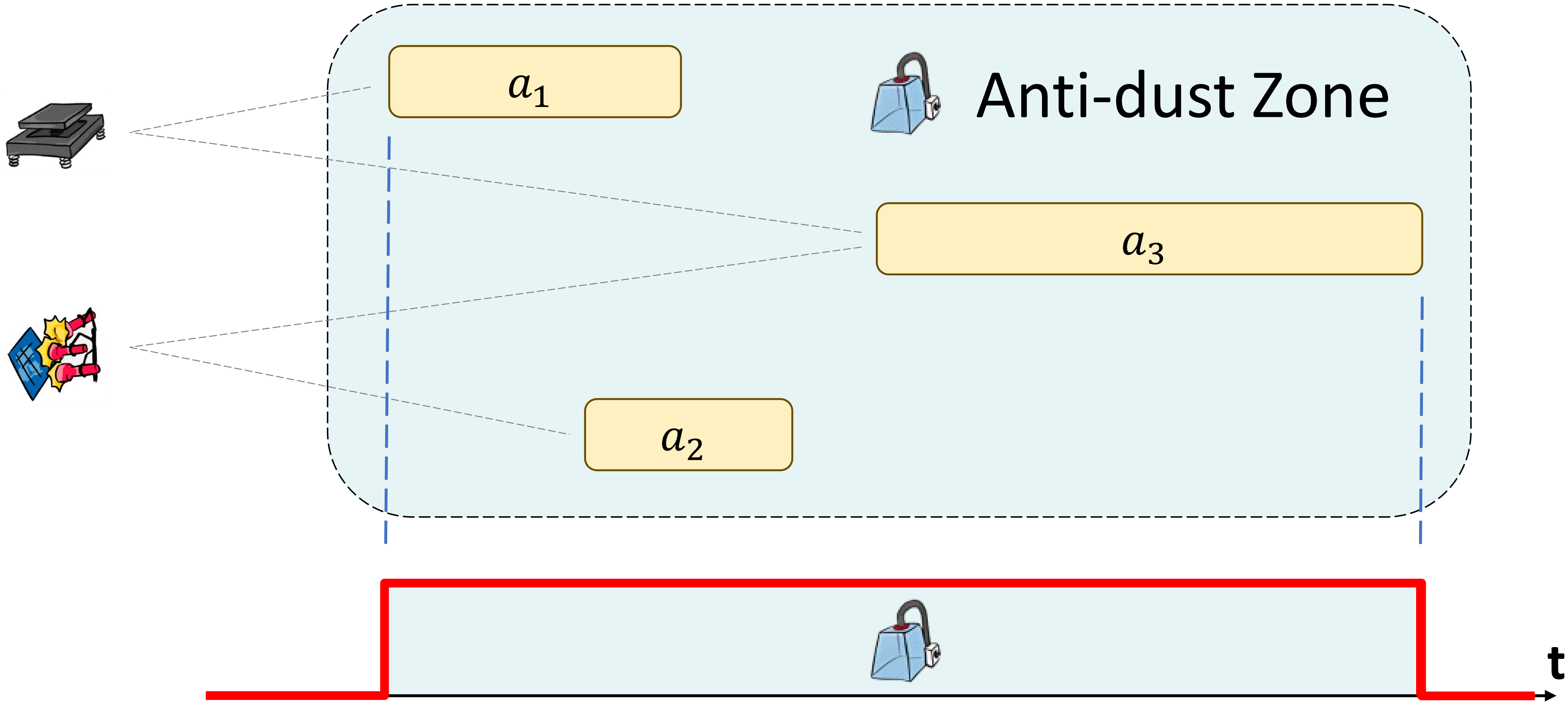
Due dates

HM-RCPSP

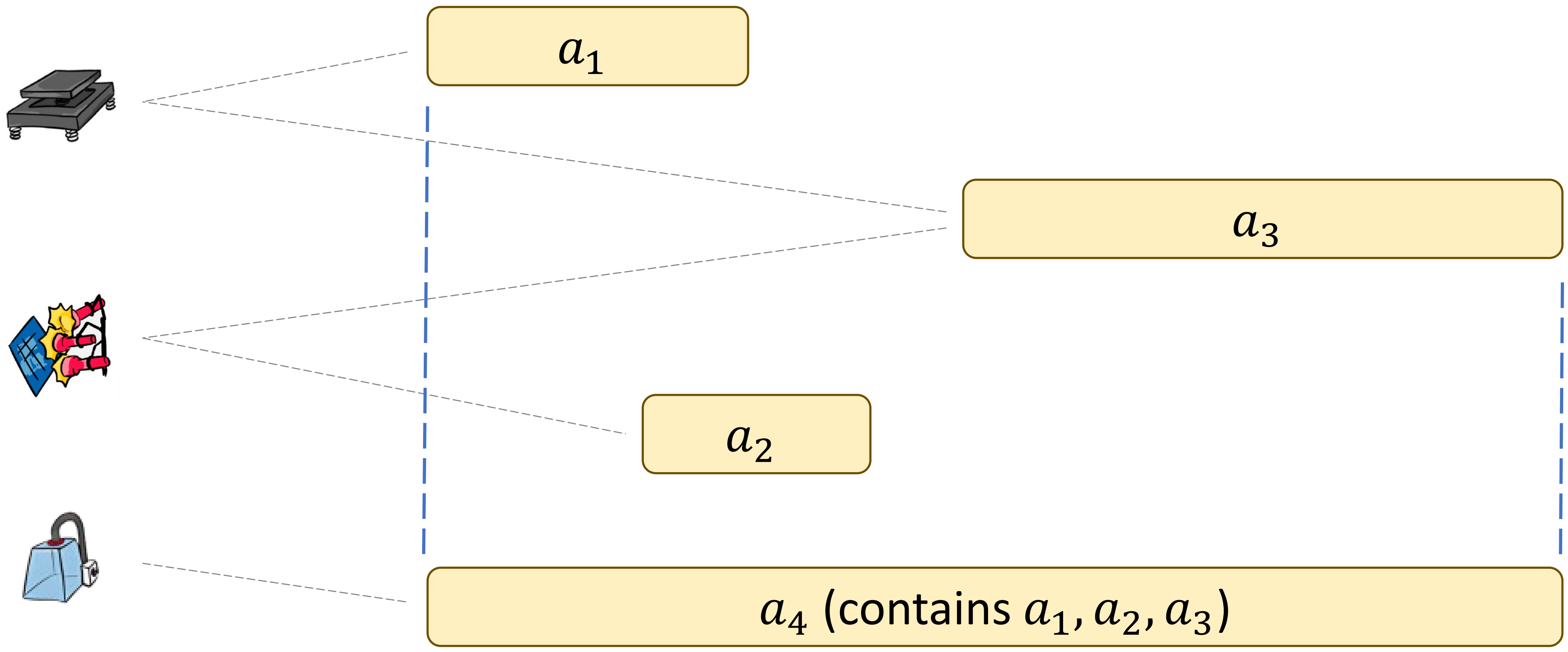
Compound Activity



Compound Activity

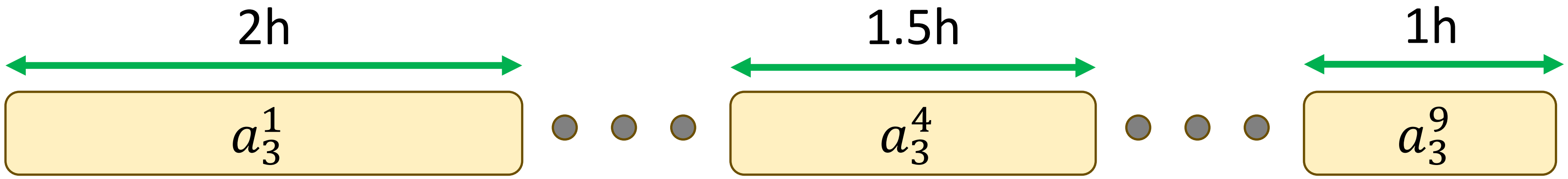


Compound Activity



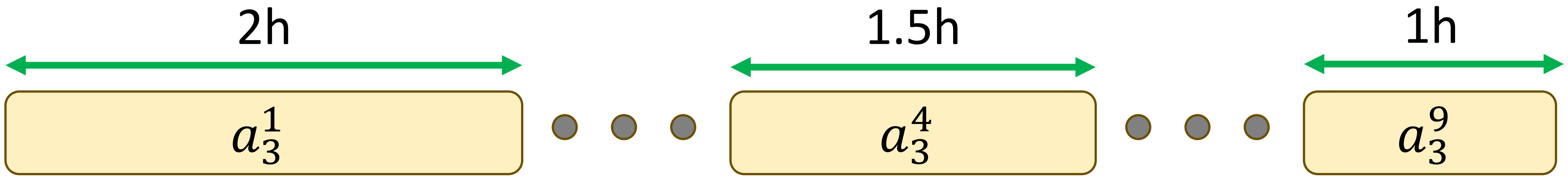
Learning Effect

- Acquiring experience ⇒ increase in efficiency



Learning Effect

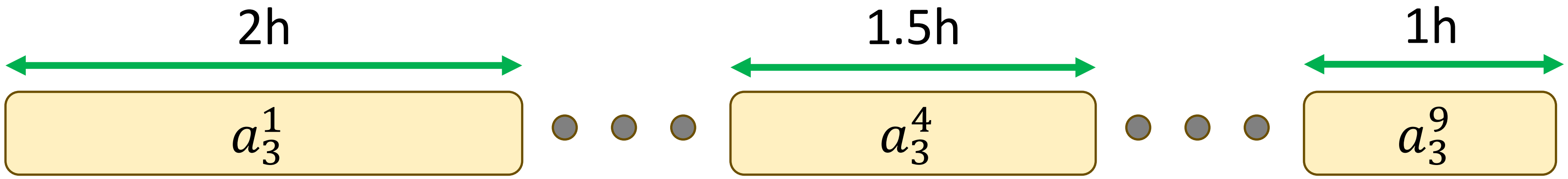
- Acquiring experience ⇒ increase in efficiency



- The execution of activities has an impact on the duration of others

Learning Effect

- Acquiring experience ⇒ increase in efficiency



- The execution of activities has an impact on the duration of others

⇒ Projects within class are interdependent

State of the Art

- RCPSP
- MP-RCPSP
- HM-RCPSP

State of the Art

- RCPSP
- MP-RCPSP
- HM-RCPSP

Learning effect

State of the Art

- RCPSP
- MP-RCPSP
- HM-RCPSP

Learning effect

- [Amor 2002]
- [Van Peteghem & Vanhoucke 2015]
- [Qin et al. 2016]
- [Hill et al. 2021]

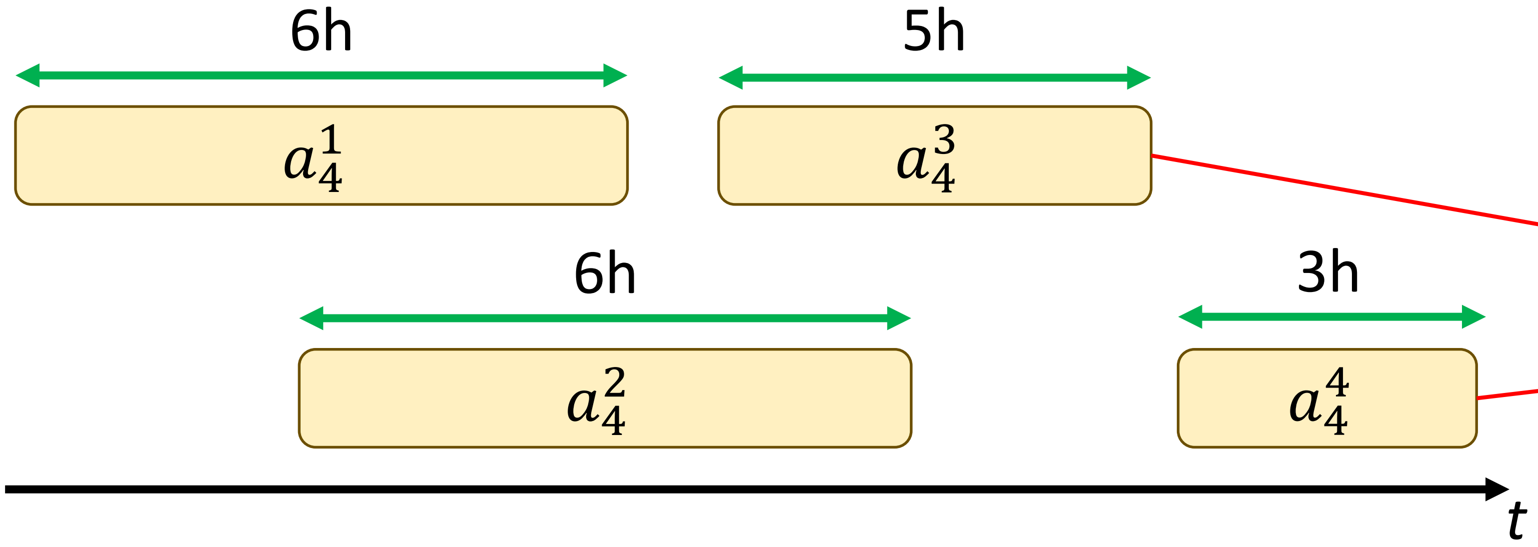
Objectives

- Bi-objective lexicographic scheduling problem:
  - 1st objective: minimize the sum of projects tardiness
  - 2nd objective: minimize the makespan of the entire schedule

Table of Contents

Learning Effect Modeling

- Monotonically decreasing function $dur : \mathbb{N} \rightarrow \mathbb{N}$



Completed executions	Duration (h)
0	6
1	5
2	4
3	3
4	3

HM-RCPSP/L-C

- Problem input:
  - Classes, projects and activities
  - Due dates
  - Learning effect
  - Resources

HM-RCPSP/L-C

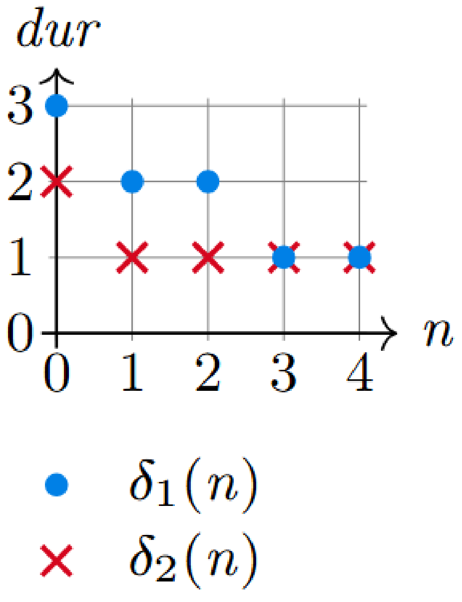
- Problem input:
  - Classes, projects and activities
  - Due dates
  - Learning effect
  - Resources
- Solution: a schedule / the start time of each instance of each activity, which satisfies
  - Precedence relation
  - Composition relation
  - Resource capacities
  - Duration function

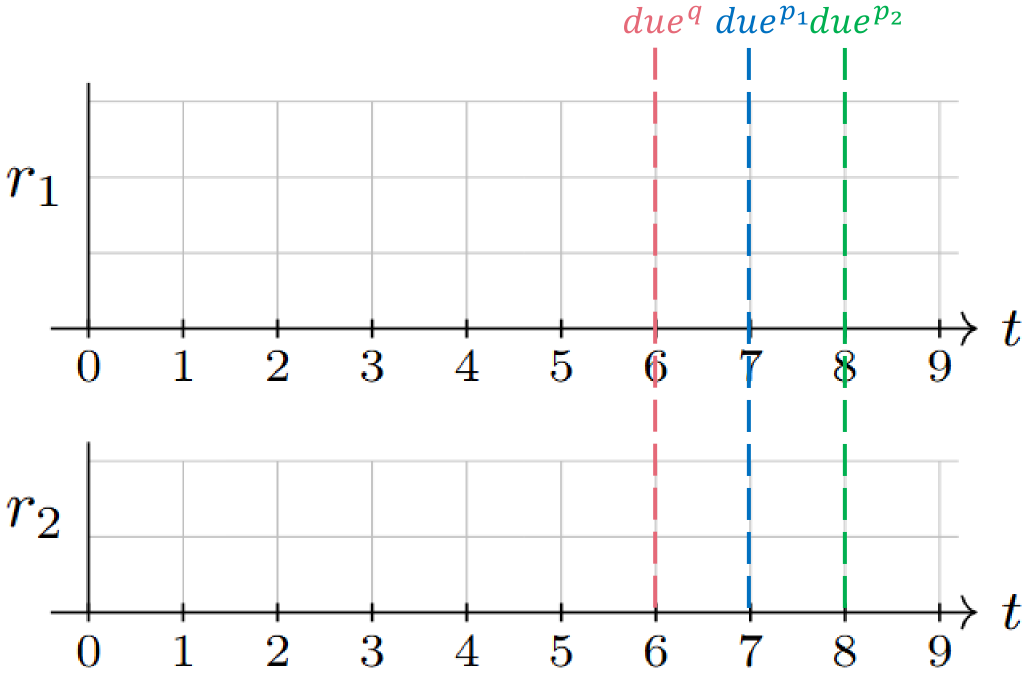
HM-RCPSP/L-C

- Problem input:
  - Classes, projects and activities
  - Due dates
  - Learning effect
  - Resources
- Solution: a schedule / the start time of each instance of each activity, which satisfies
  - Precedence relation
  - Composition relation
  - Resource capacities
  - Duration function
- Optimization criteria: sum of tardiness & makespan

Example

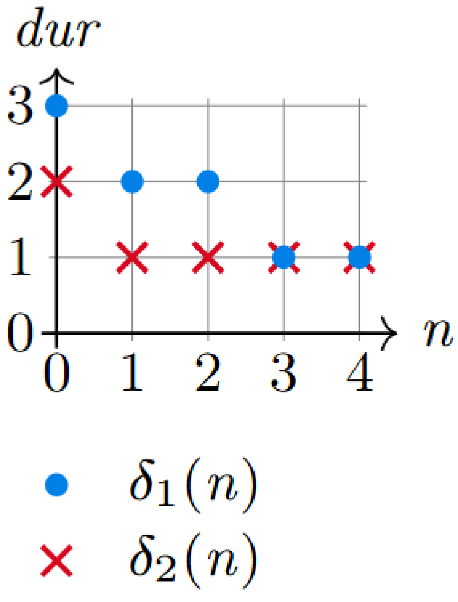
$\mathcal{C}$	$\mathcal{A}_c$	dur_a	r_1	r_2
c_1 p_1 p_2	a	δ_1	1	0
	b	δ_2	0	1
	c	δ_1	0	1
	d	-	1	0
c_2 q	e	δ_2	0	1
	f	δ_1	1	0
	g	-	1	0

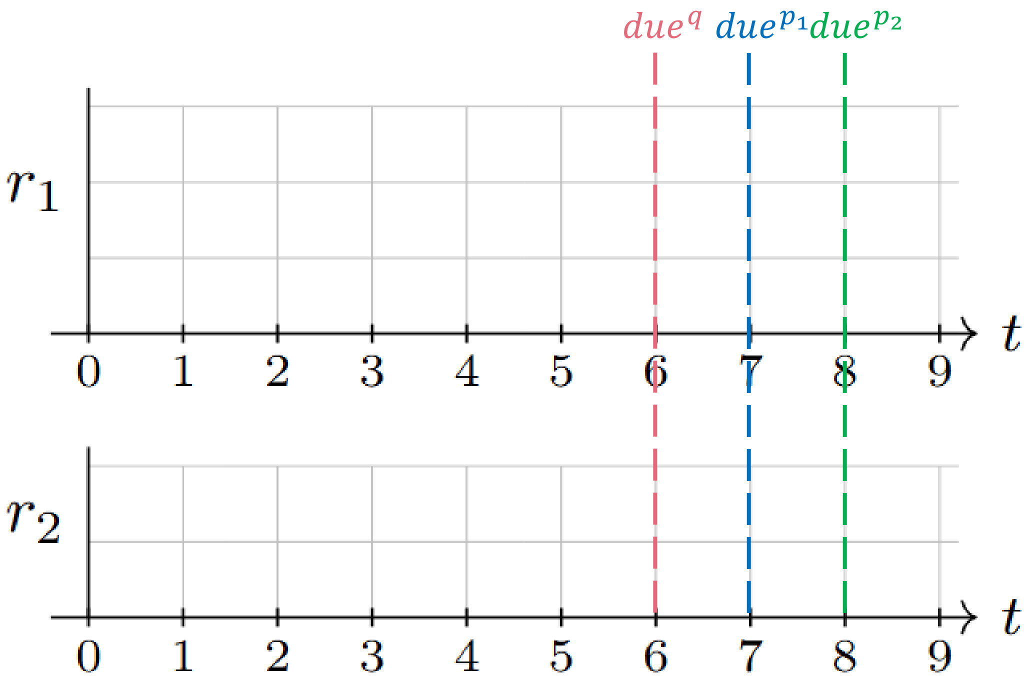




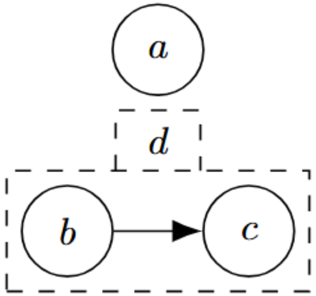
Example

$\mathcal{C}$	$\mathcal{A}_c$	dur_a	r_1	r_2
c_1 p_1 p_2	a	δ_1	1	0
	b	δ_2	0	1
	c	δ_1	0	1
	d	-	1	0
c_2 q	e	δ_2	0	1
	f	δ_1	1	0
	g	-	1	0

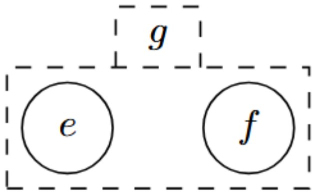




c_1 :

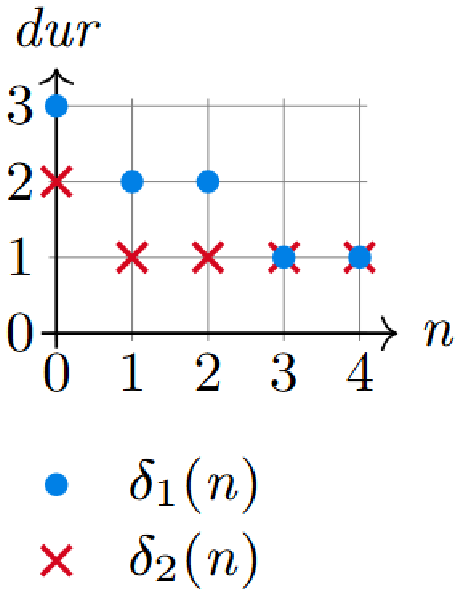


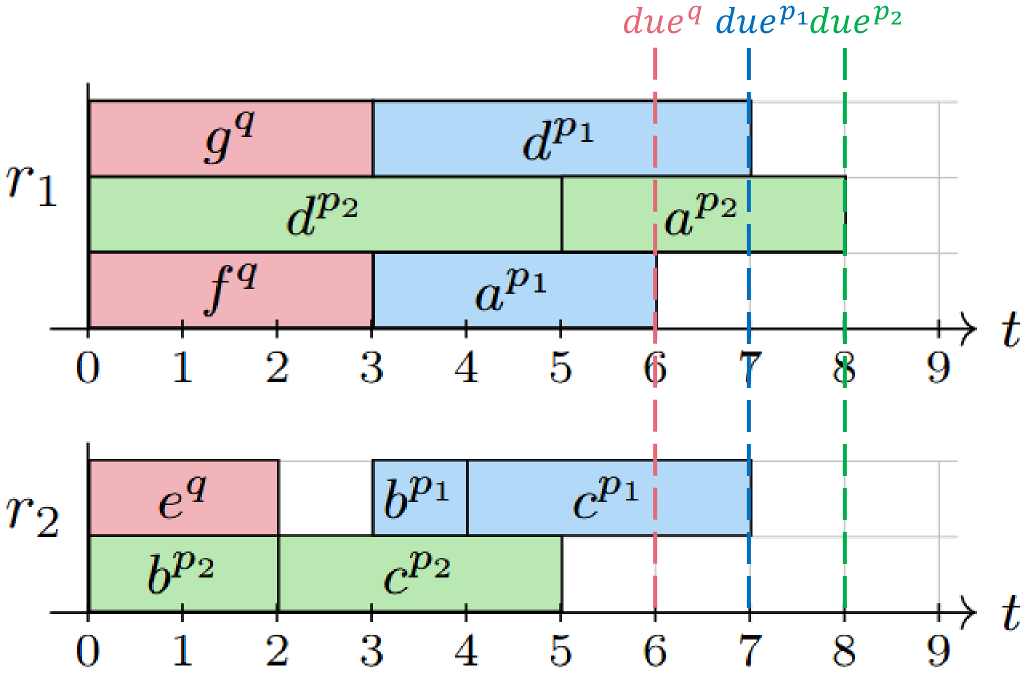
c_2 :



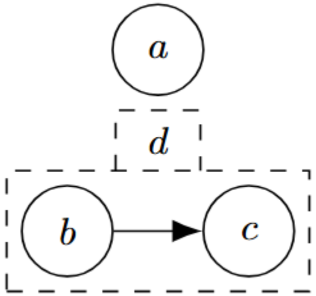
Example

$\mathcal{C}$	$\mathcal{A}_c$	dur_a	r_1	r_2
c_1 (p_1 p_2)	a	δ_1	1	0
	b	δ_2	0	1
	c	δ_1	0	1
	d	-	1	0
c_2 (q)	e	δ_2	0	1
	f	δ_1	1	0
	g	-	1	0

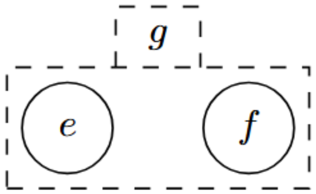




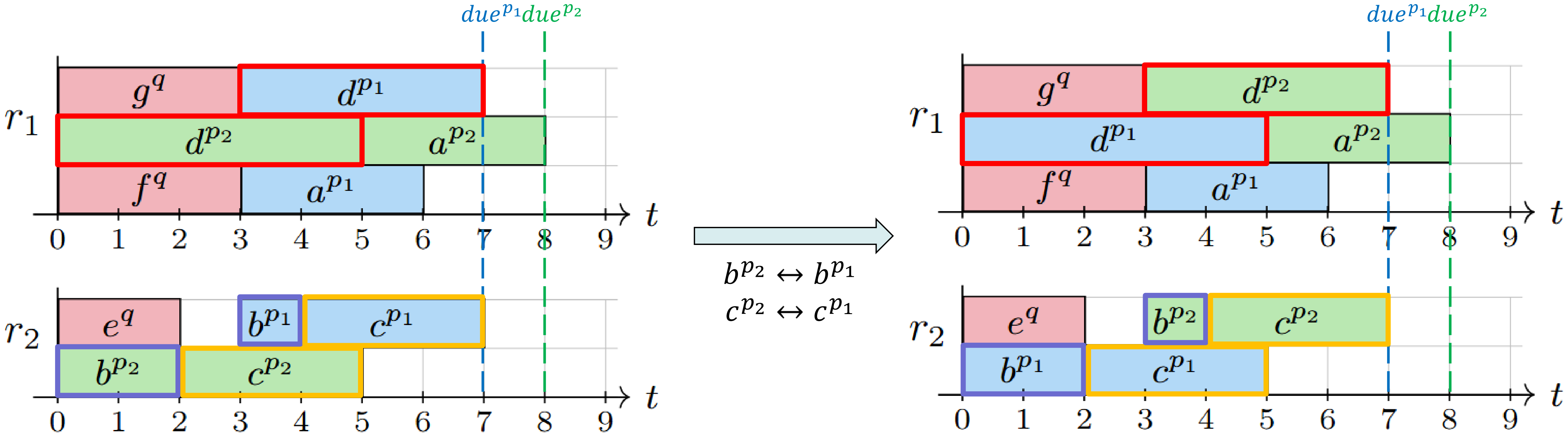
c_1 :



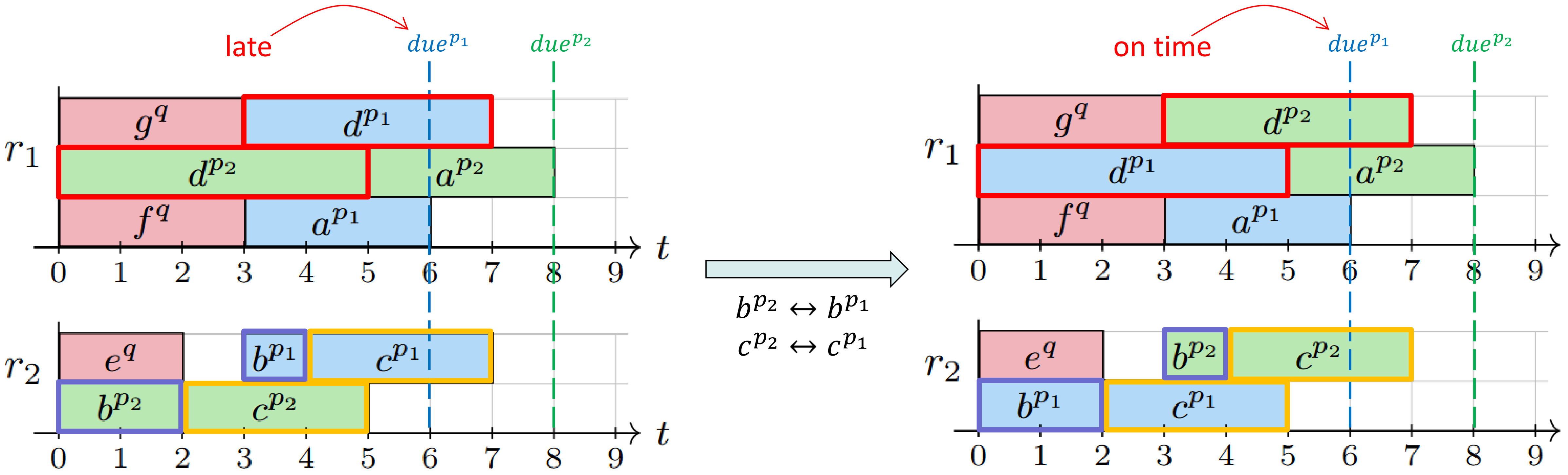
c_2 :



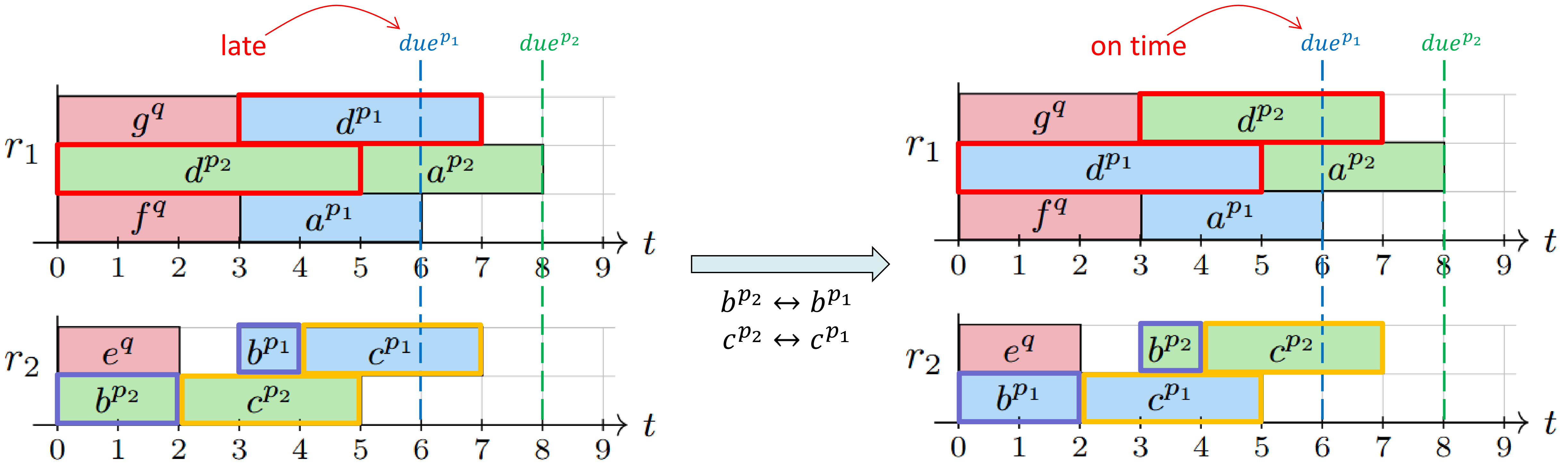
Symmetry Breaking



Symmetry Breaking



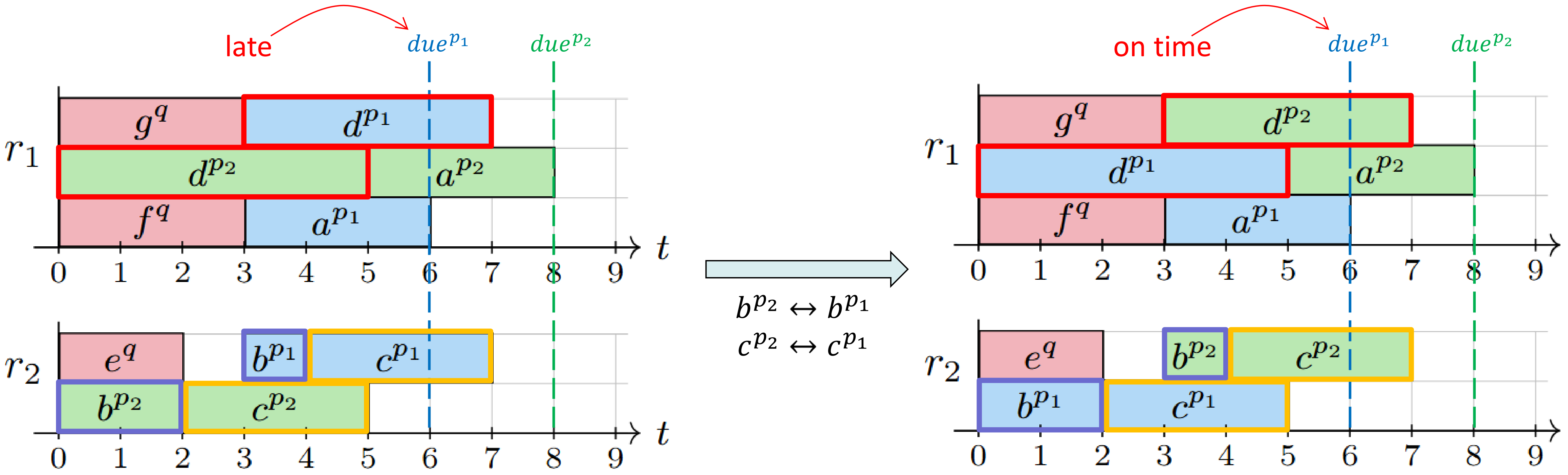
Symmetry Breaking



Theorem:

$$\boldsymbol{IF}\ due^{p_1} \leq due^{p_2}\ \boldsymbol{THEN}\ criteria(b^{p_1}\ before\ b^{p_2}) \leq criteria(b^{p_2}\ before\ b^{p_1})$$

Symmetry Breaking



Theorem:

$$\boldsymbol{IF}\ due^{p_1} \leq due^{p_2}\ \boldsymbol{THEN}\ criteria(b^{p_1}\ before\ b^{p_2}) \leq criteria(b^{p_2}\ before\ b^{p_1})$$

➡ Activities are performed in order of due date for same-class projects

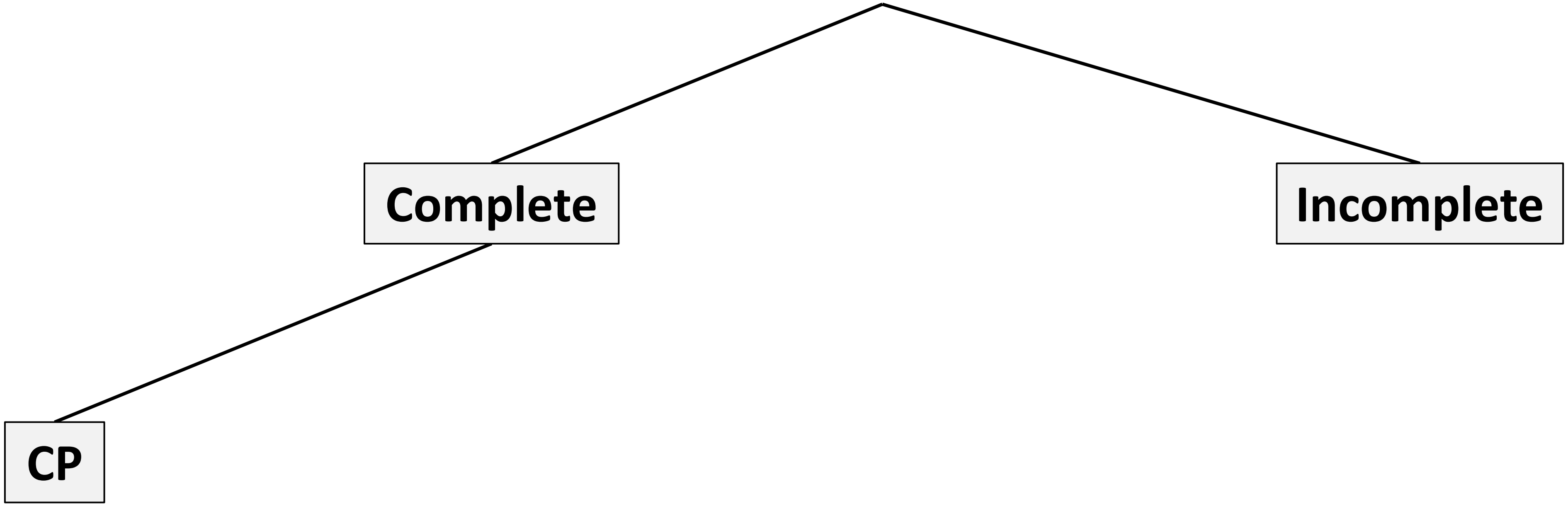
Table of Contents

Solving Approaches

Complete

Incomplete

Solving Approaches



Constraint Programming

For each activity a of each project p:

- Interval variable $\mathbf{itv}_{a,p}$ contains properties of this activity instance (starting time, duration)
- Integer variable $\mathbf{n}_{a,p}$ indicates the number of executions that have already been completed of this activity instance

Constraint Programming

$$minimize \left(\underbrace{\sum_{c \in C, p \in I_c, a \in A_c} \max(0, \mathrm{endOf}\left(\mathrm{itv}_{a,p}\right) - due_p)}_{\text{Tardiness}}, \underbrace{\max_{c \in C, p \in I_c, a \in A_c} \mathrm{endOf}\left(\mathrm{itv}_{a,p}\right)}_{\text{Makespan}} \right)$$

Constraint Programming

$$minimize \left(\underbrace{\sum_{c \in C, p \in I_c, a \in A_c} \max\left(0, \mathrm{endOf}\left(\mathrm{itv}_{a,p}\right) - due_p\right),}_{\text{Tardiness}} \underbrace{\max_{c \in C, p \in I_c, a \in A_c} \mathrm{endOf}\left(\mathrm{itv}_{a,p}\right)}_{\text{Makespan}} \right)$$

Classical RCPSP

$$\forall r \in R \qquad \sum_{c \in C, p \in I_c, a \in A_c} \mathrm{pulse}\left(\mathbf{itv}_{a,p}, cons_{r,a}\right) \leq capa_r \qquad (1)$$

$$\forall c \in C, \forall p \in I_c, \forall (a,b) \in Prec_c \qquad \mathrm{endBeforeStart}\left(\mathbf{itv}_{a,p}, \mathbf{itv}_{b,p}\right) \qquad (2)$$

Constraint Programming

$$minimize \left(\underbrace{\sum_{c\in C, p\in I_c, a\in A_c} \max\left(0, \mathrm{endOf}\left(\mathrm{itv}_{a,p}\right) - due_p\right)}_{\text{Tardiness}}, \underbrace{\max_{c\in C, p\in I_c, a\in A_c} \mathrm{endOf}\left(\mathrm{itv}_{a,p}\right)}_{\text{Makespan}} \right)$$

Classical RCPSP

$$\forall r \in R \qquad \textstyle\sum_{c\in C, p\in I_c, a\in A_c} \mathrm{pulse}\left(\mathbf{itv}_{a,p}, cons_{r,a}\right) \le capa_r \qquad (1)$$

$$\forall c \in C, \forall p \in I_c, \forall (a,b) \in Prec_c \qquad \mathrm{endBeforeStart}\left(\mathbf{itv}_{a,p}, \mathbf{itv}_{b,p}\right) \qquad (2)$$

Compound Activities

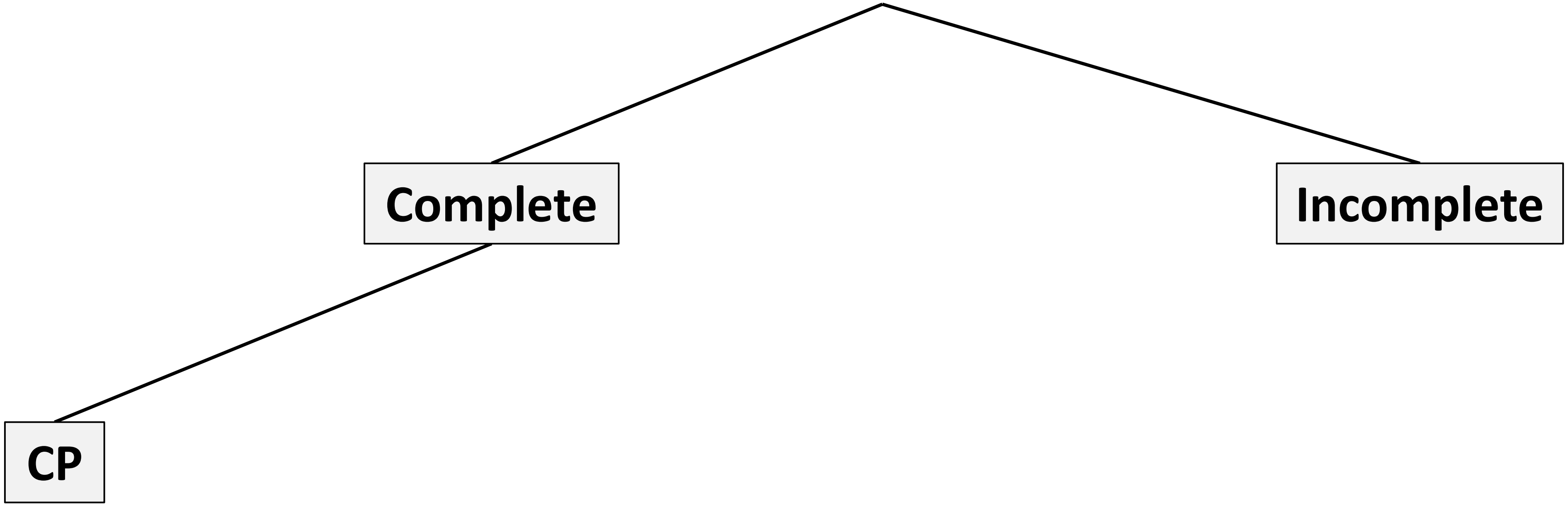
$$\forall c \in C, \forall p \in I_c, \forall a \in A_c^{Comp} \qquad \mathrm{span}\left(\mathbf{itv}_{a,p}, \left\{\mathbf{itv}_{b,p} \middle| (a,b) \in Comp_c\right\}\right) \qquad (3)$$

Constraint Programming

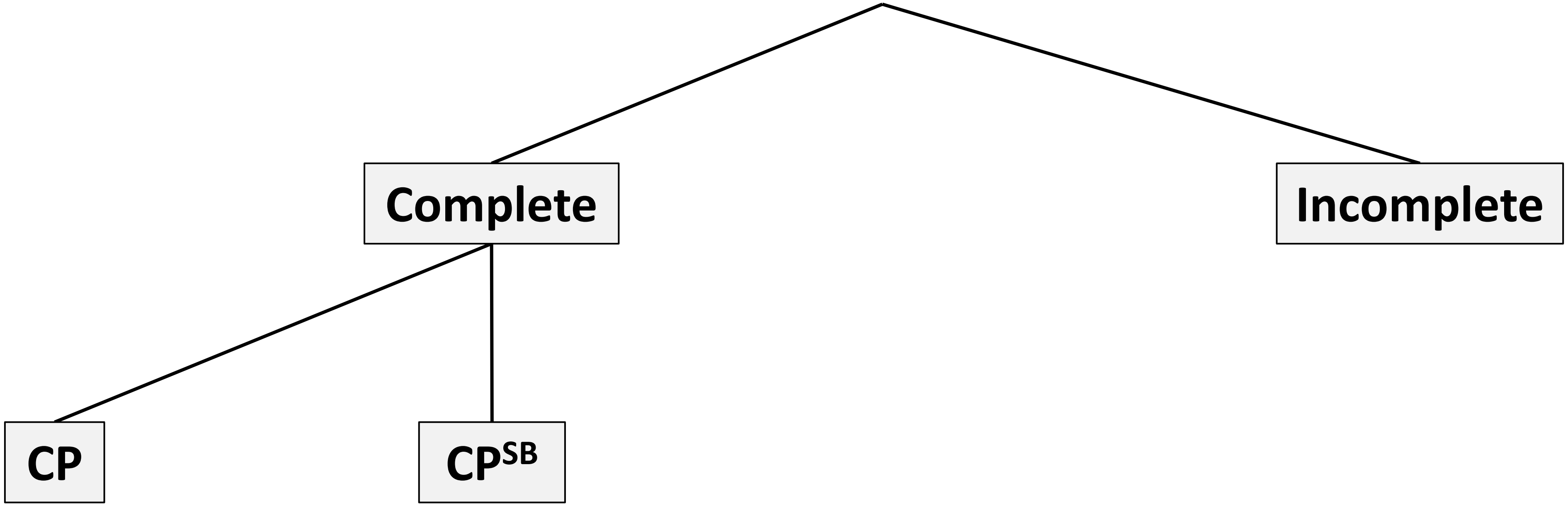
$$minimize \left(\underbrace{\sum_{c \in C, p \in I_c, a \in A_c} \max\left(0, \mathrm{endOf}\left(\mathrm{itv}_{a,p}\right) - due_p\right)}_{\text{Tardiness}}, \underbrace{\max_{c \in C, p \in I_c, a \in A_c} \mathrm{endOf}\left(\mathrm{itv}_{a,p}\right)}_{\text{Makespan}} \right)$$

Classical RCPSP	$\forall r \in R$	$\sum_{c \in C, p \in I_c, a \in A_c} \mathrm{pulse}\left(\mathbf{itv}_{a,p}, cons_{r,a}\right) \leq capa_r$	(1)
	$\forall c \in C, \forall p \in I_c, \forall (a,b) \in Prec_c$	$\mathrm{endBeforeStart}\left(\mathbf{itv}_{a,p}, \mathbf{itv}_{b,p}\right)$	(2)
Compound Activities	$\forall c \in C, \forall p \in I_c, \forall a \in A_c^{Comp}$	$\mathrm{span}\left(\mathbf{itv}_{a,p}, \left\{\mathbf{itv}_{b,p} \middle\| (a,b) \in Comp_c\right\}\right)$	(3)
Learning Effect	$\forall c \in C, \forall p \in I_c, \forall a \in A_c^{Atom}$	$\mathbf{n}_{a,p} = \sum_{q \in I_c} \left(\mathrm{endOf}\left(\mathbf{itv}_{a,q}\right) \leq \mathrm{startOf}\left(\mathbf{itv}_{a,p}\right)\right)$	(4)
	$\forall c \in C, \forall p \in I_c, \forall a \in A_c^{Atom}$	$\mathrm{lengthOf}\left(\mathbf{itv}_{a,p}\right) = dur_a\left(\mathbf{n}_{a,p}\right)$	(5)

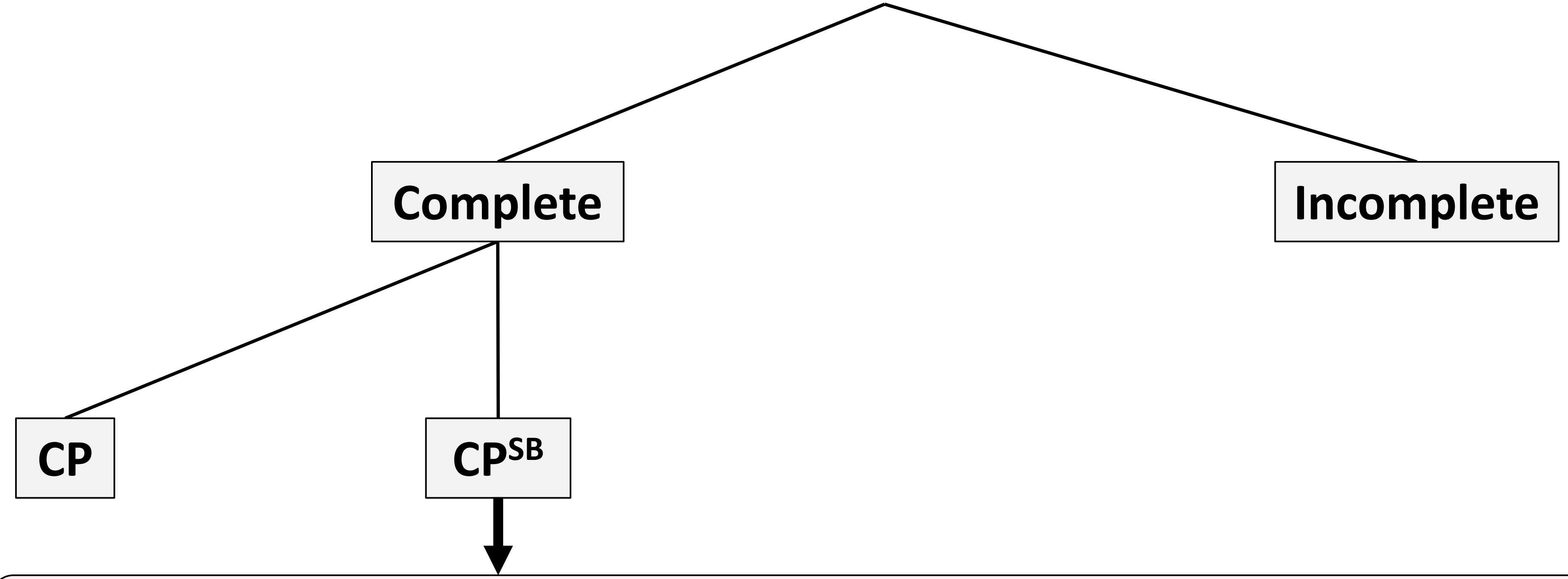
Solving Approaches



Solving Approaches

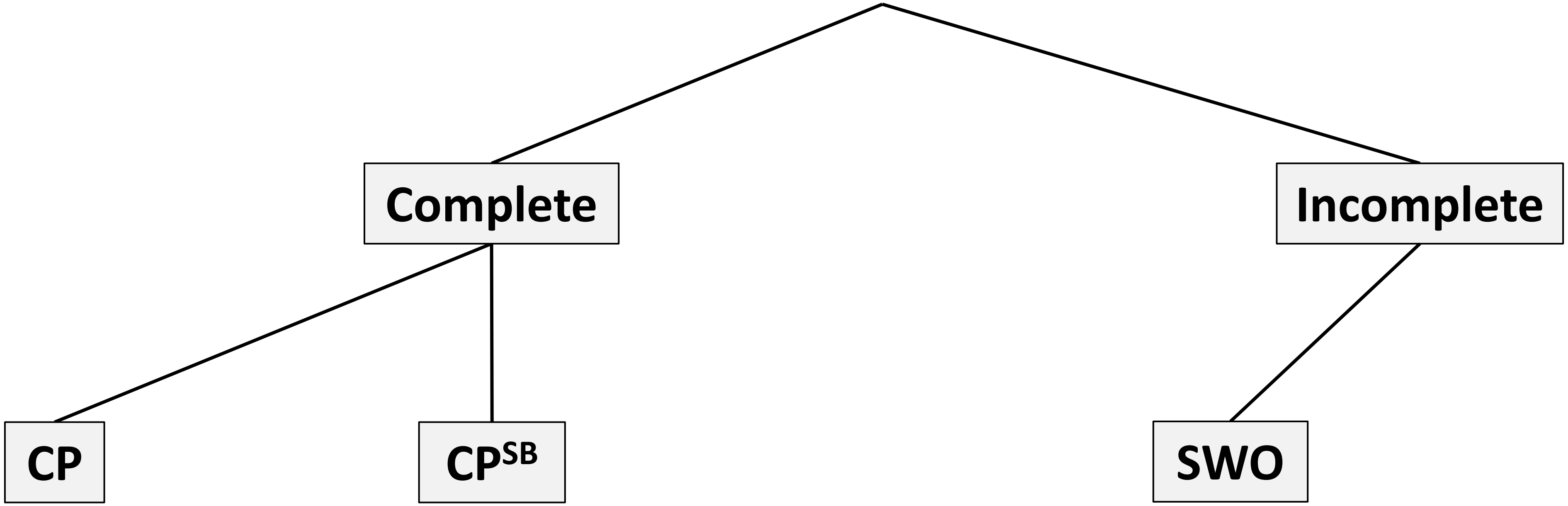


Solving Approaches

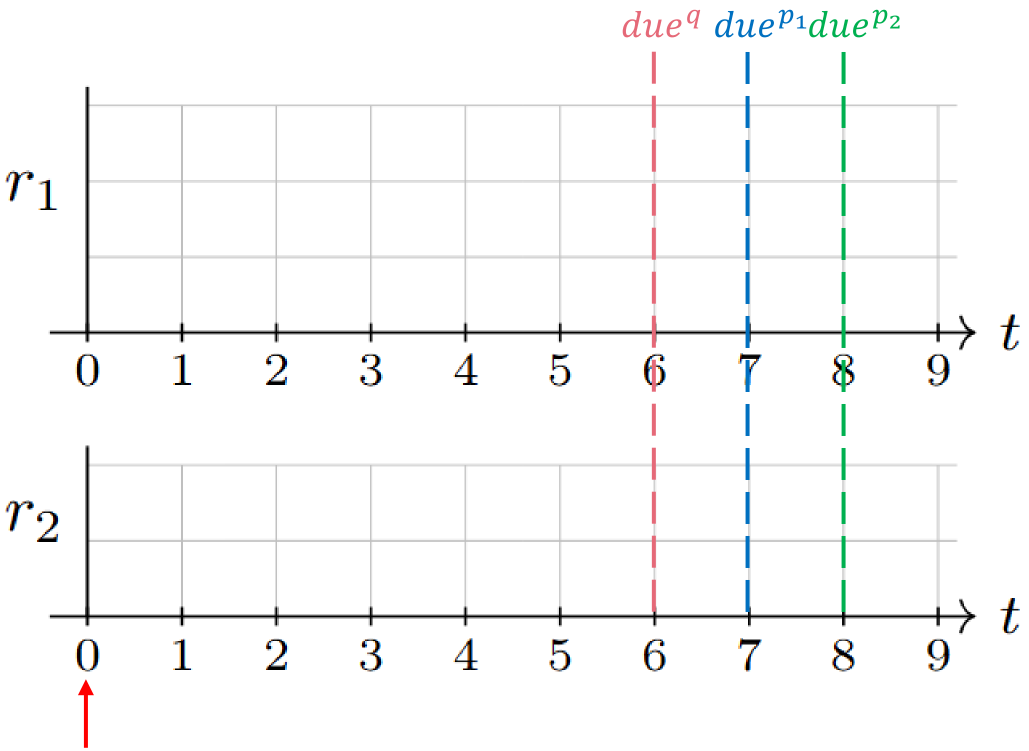


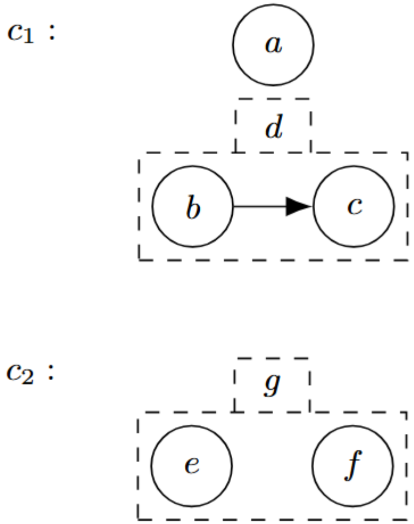
$\boxed{\textbf{CP}}$ **+** $\forall c \in C, \forall p, q \in I_c : p \prec_c q, \forall a \in A_c^{Atom}$ $\quad$ $\text{startBeforeStart}(\mathbf{itv}_{a,p}, \mathbf{itv}_{a,q})$

Solving Approaches

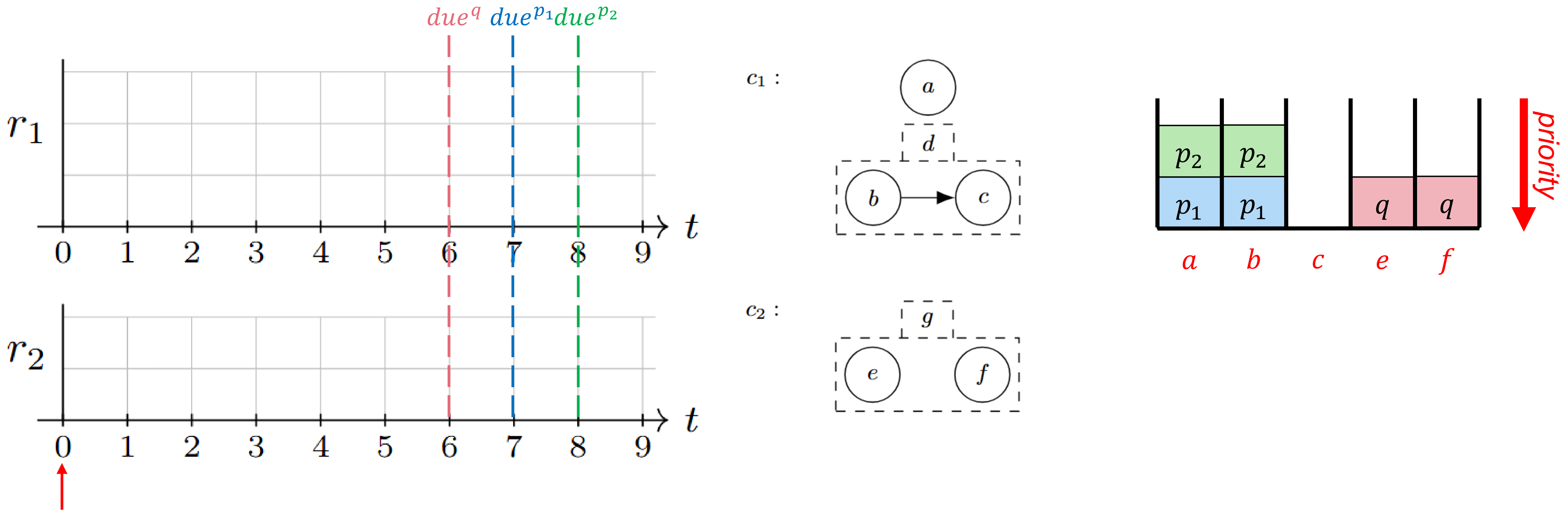


SWO (Greedy Algorithm)

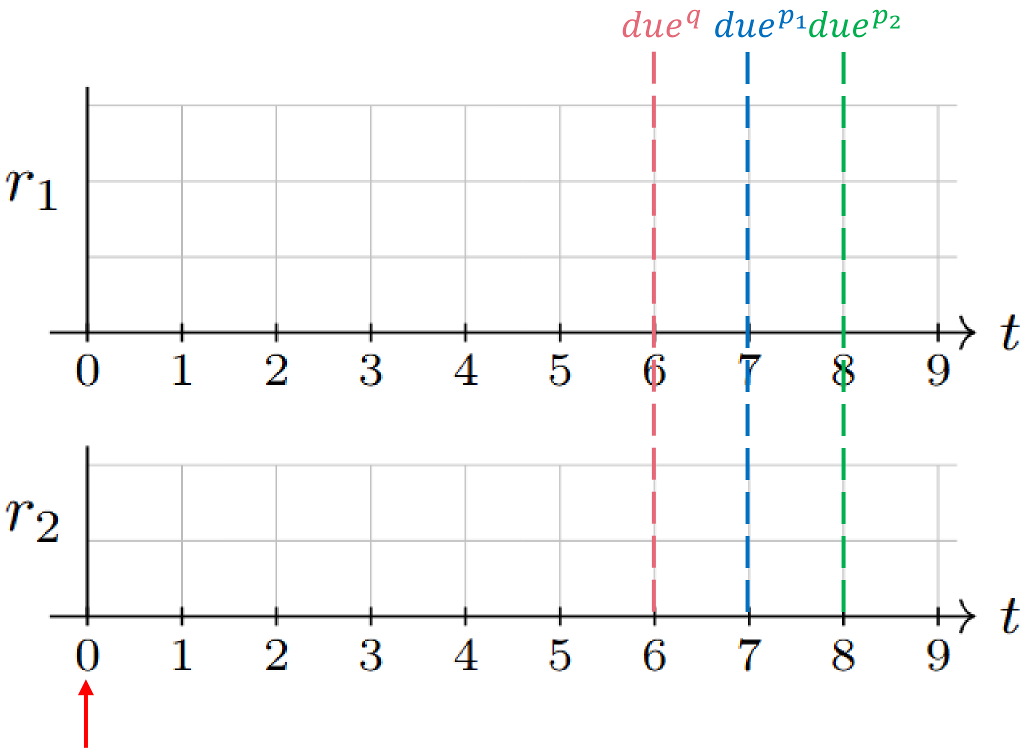


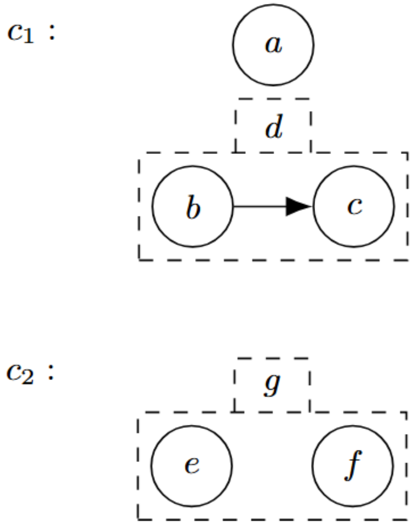


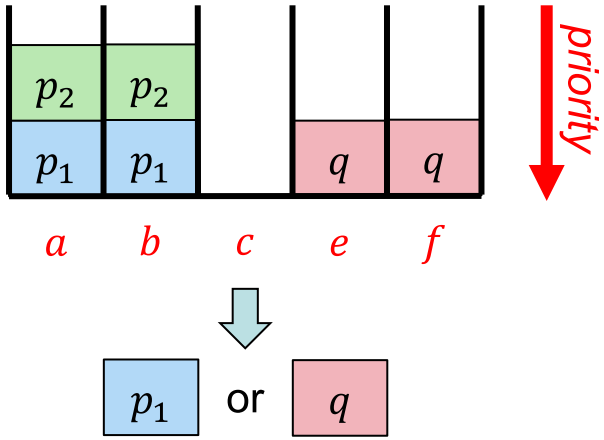
SWO (Greedy Algorithm)



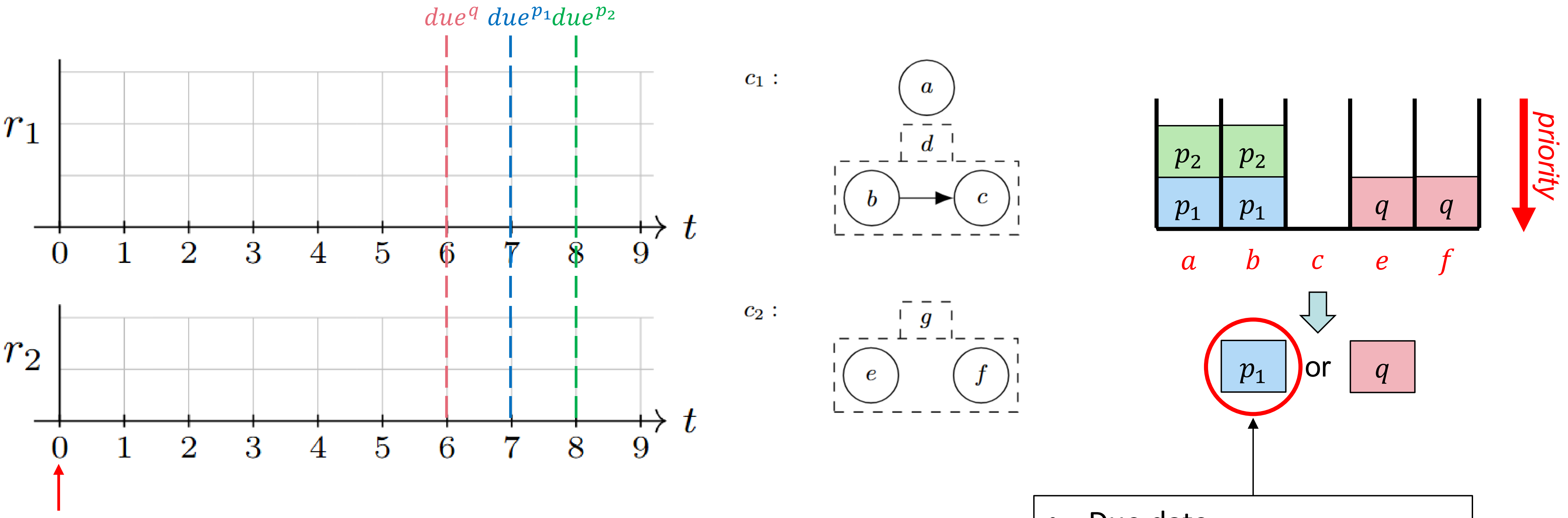
SWO (Greedy Algorithm)





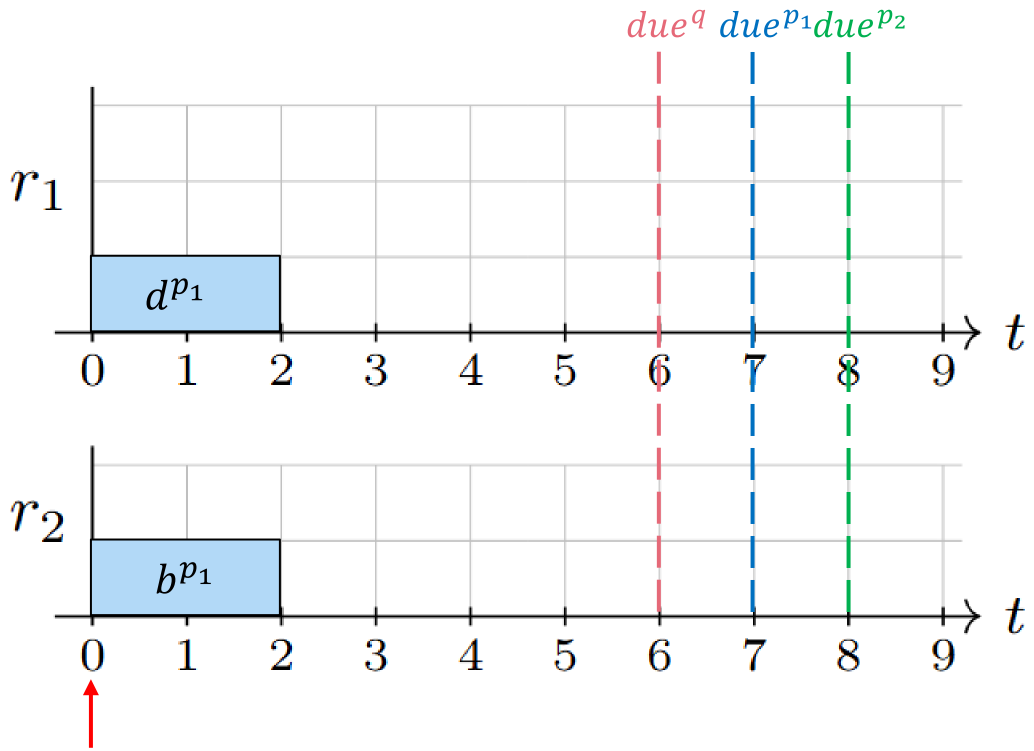


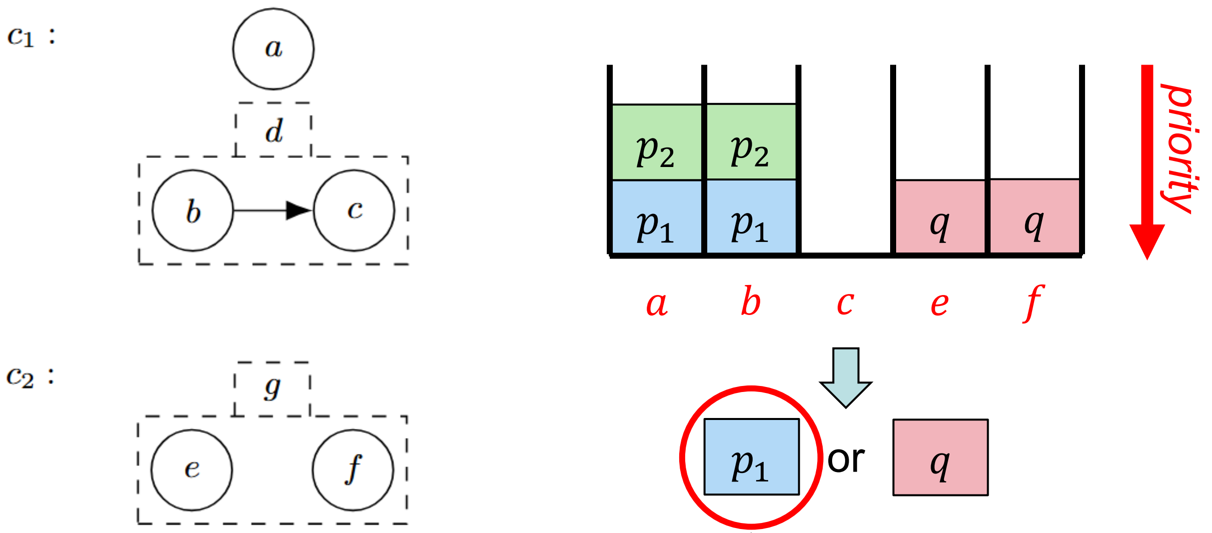
SWO (Greedy Algorithm)



- Due date
- Required time to complete
- Current time moment

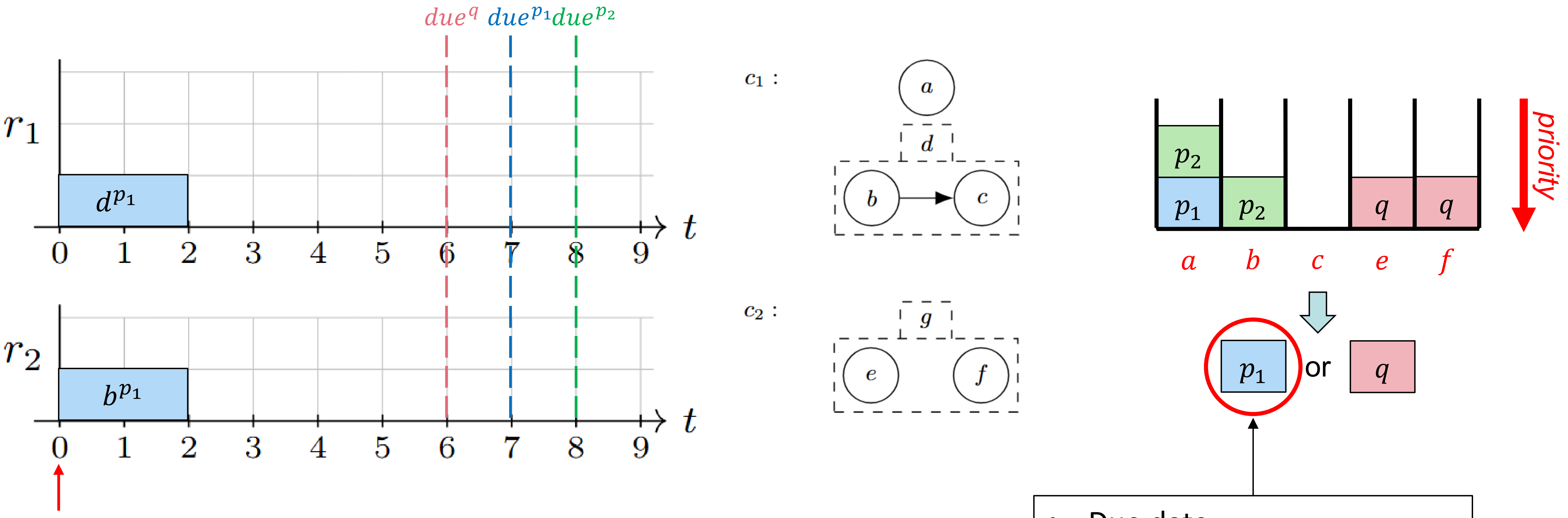
SWO (Greedy Algorithm)





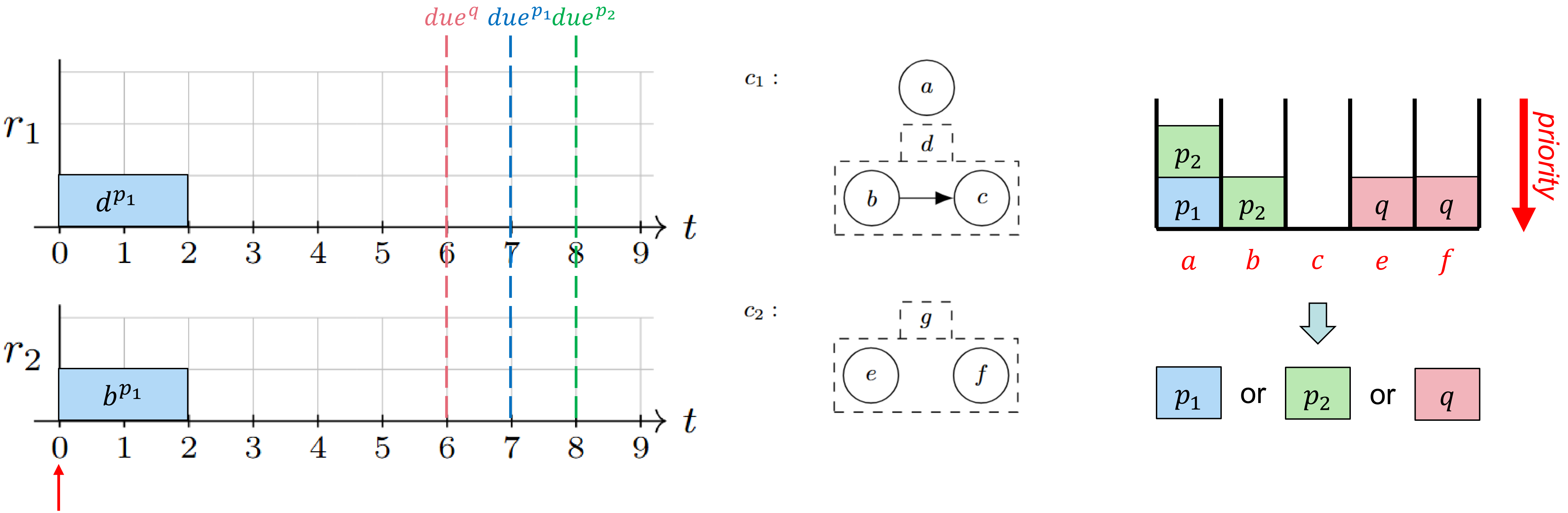
- Due date
- Required time to complete
- Current time moment

SWO (Greedy Algorithm)

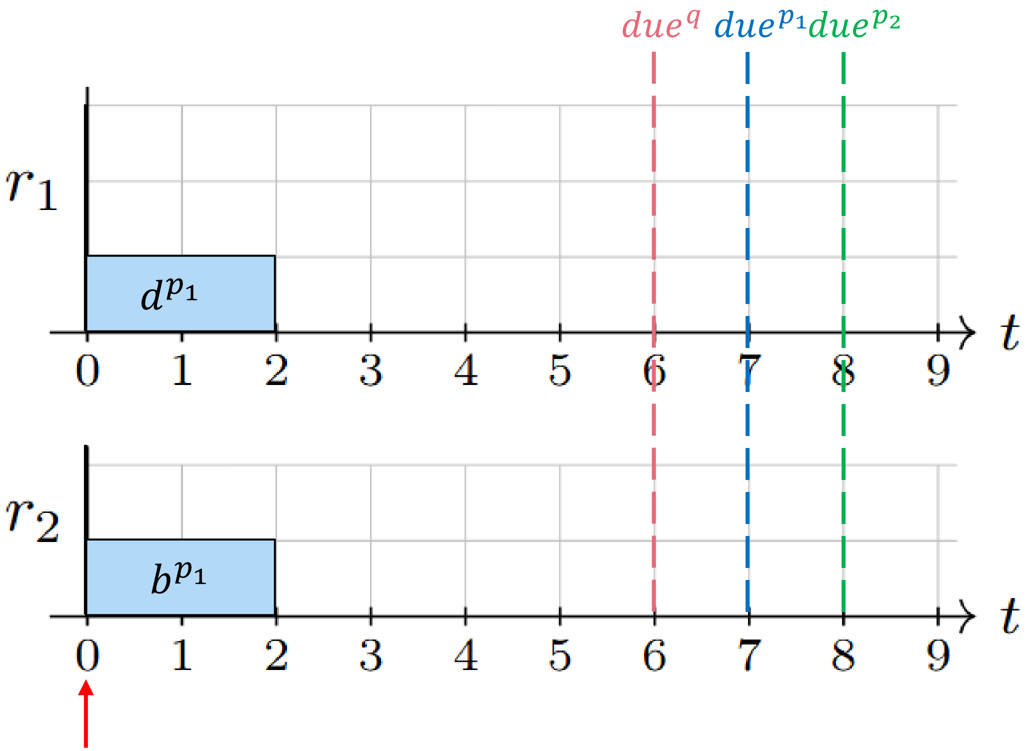


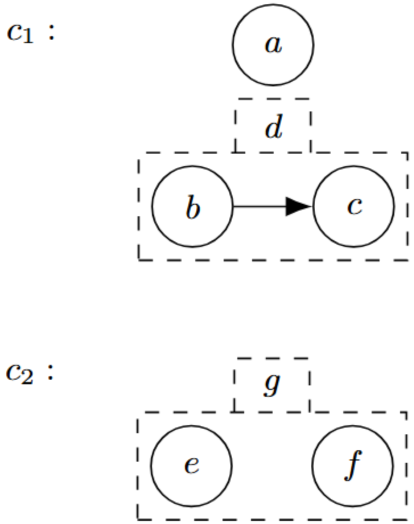
- Due date
- Required time to complete
- Current time moment

SWO (Greedy Algorithm)

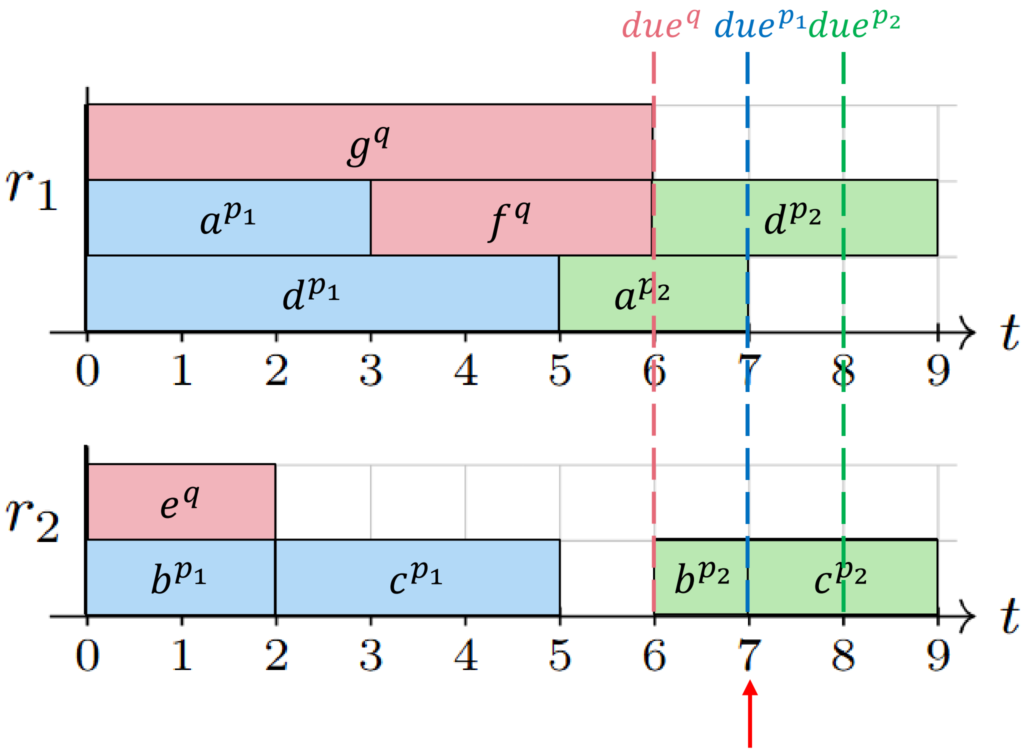


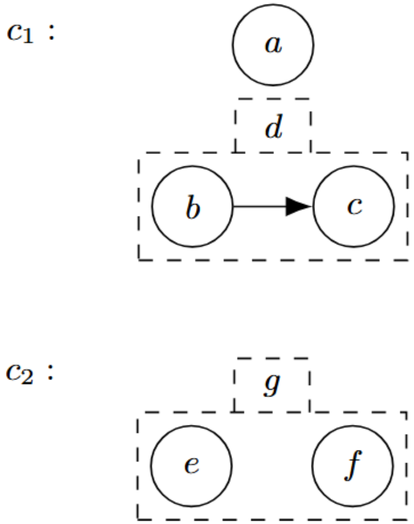
SWO (Greedy Algorithm)



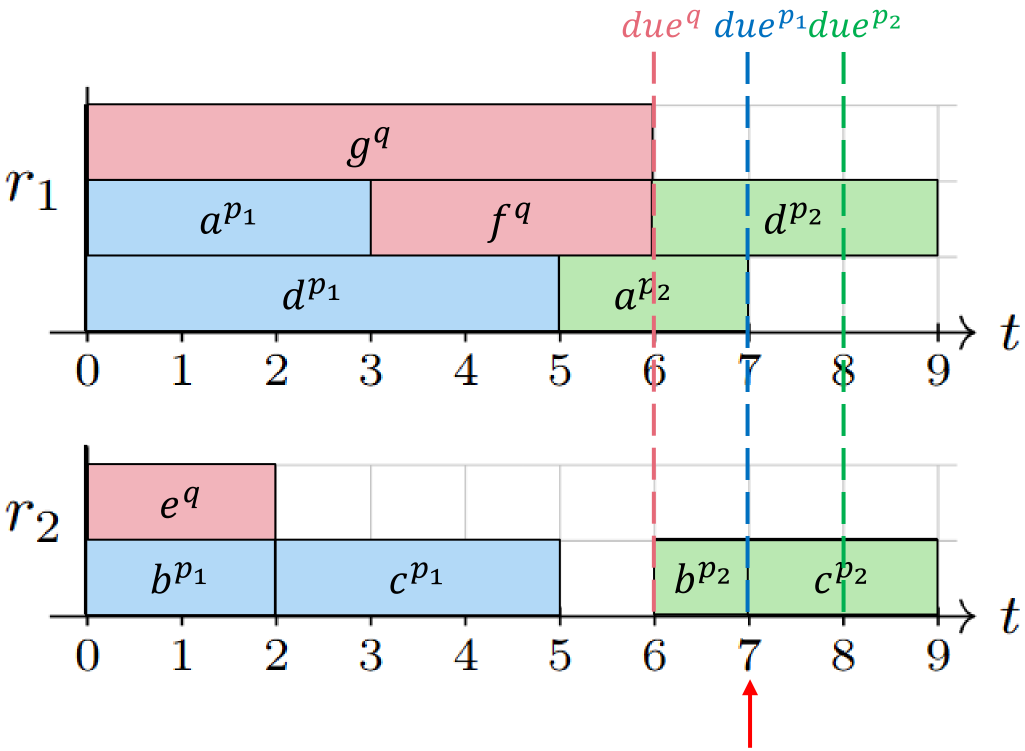


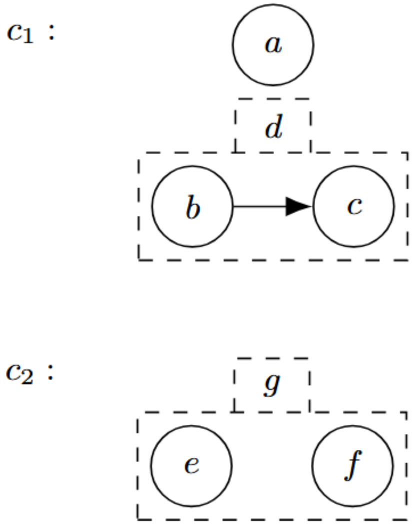
SWO (Greedy Algorithm)

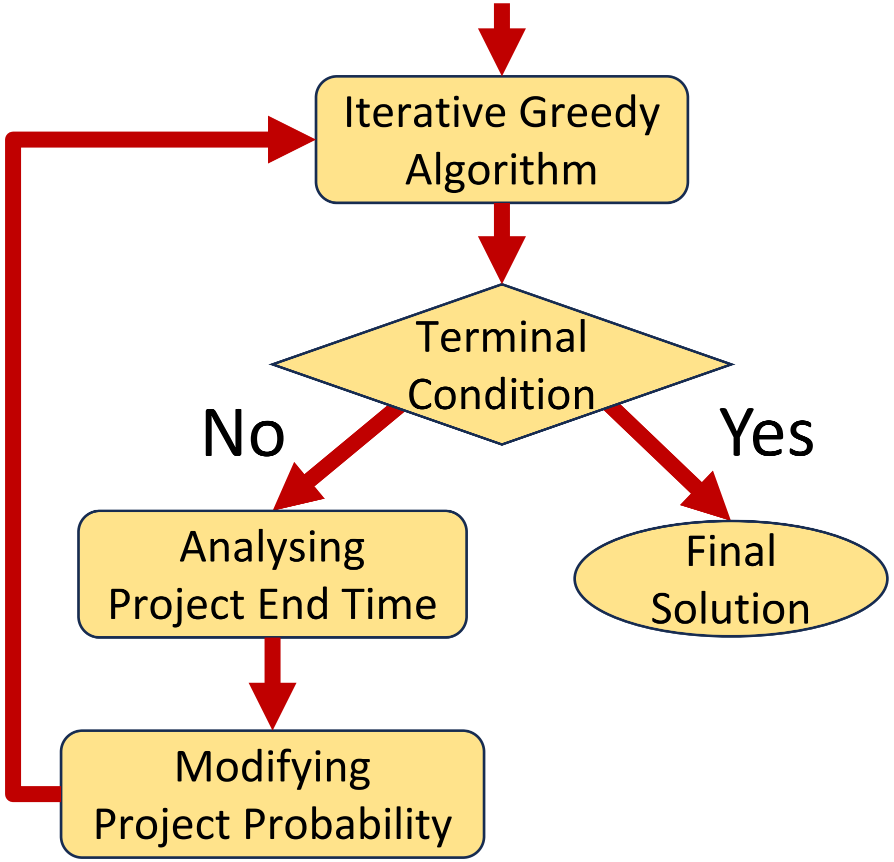




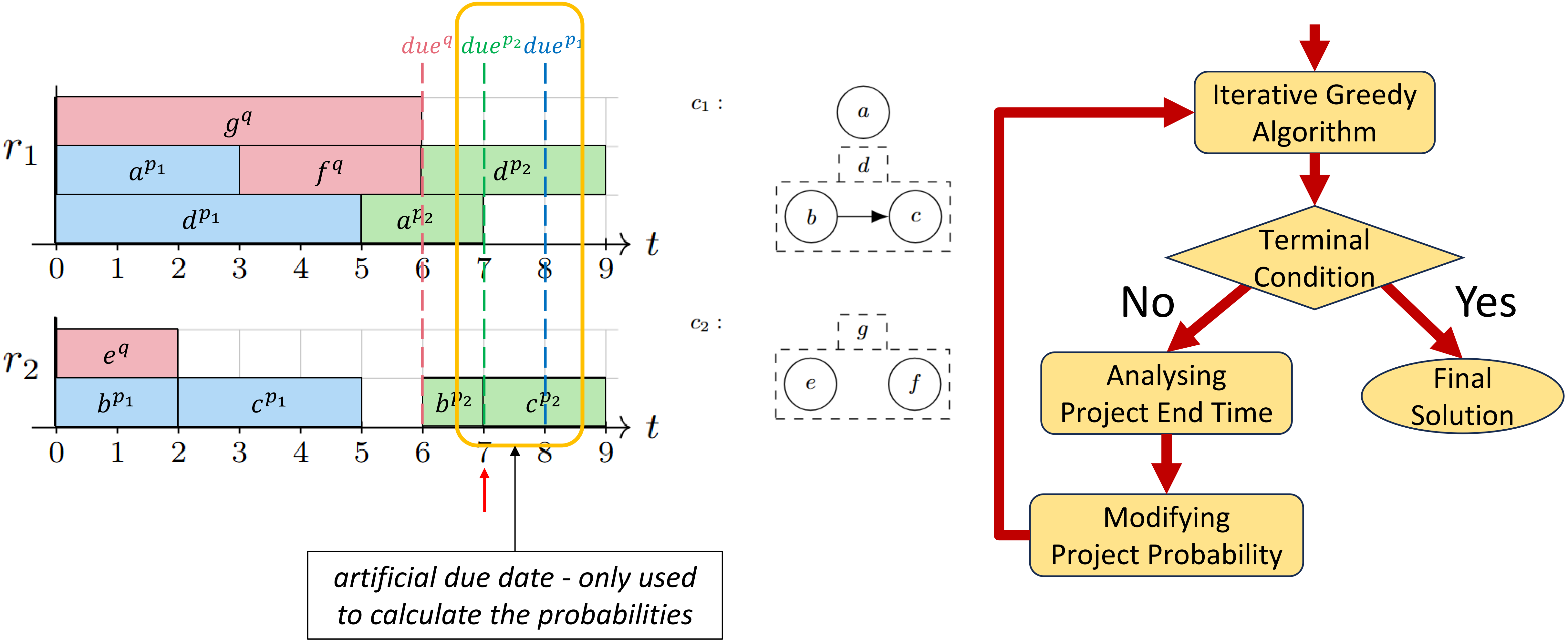
SWO (Greedy Algorithm)



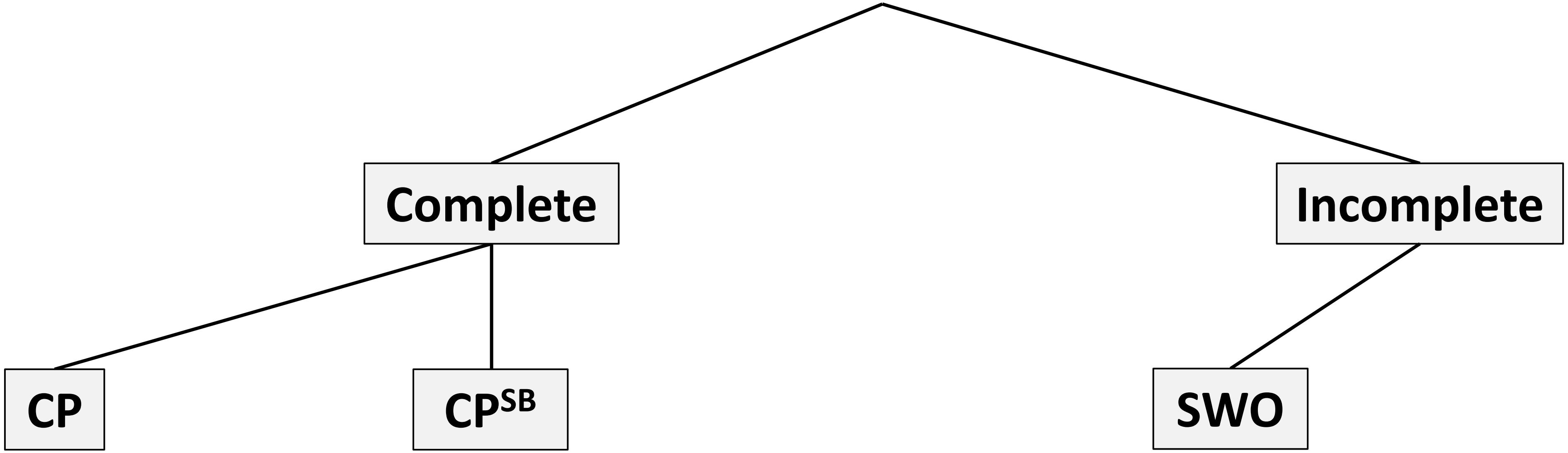




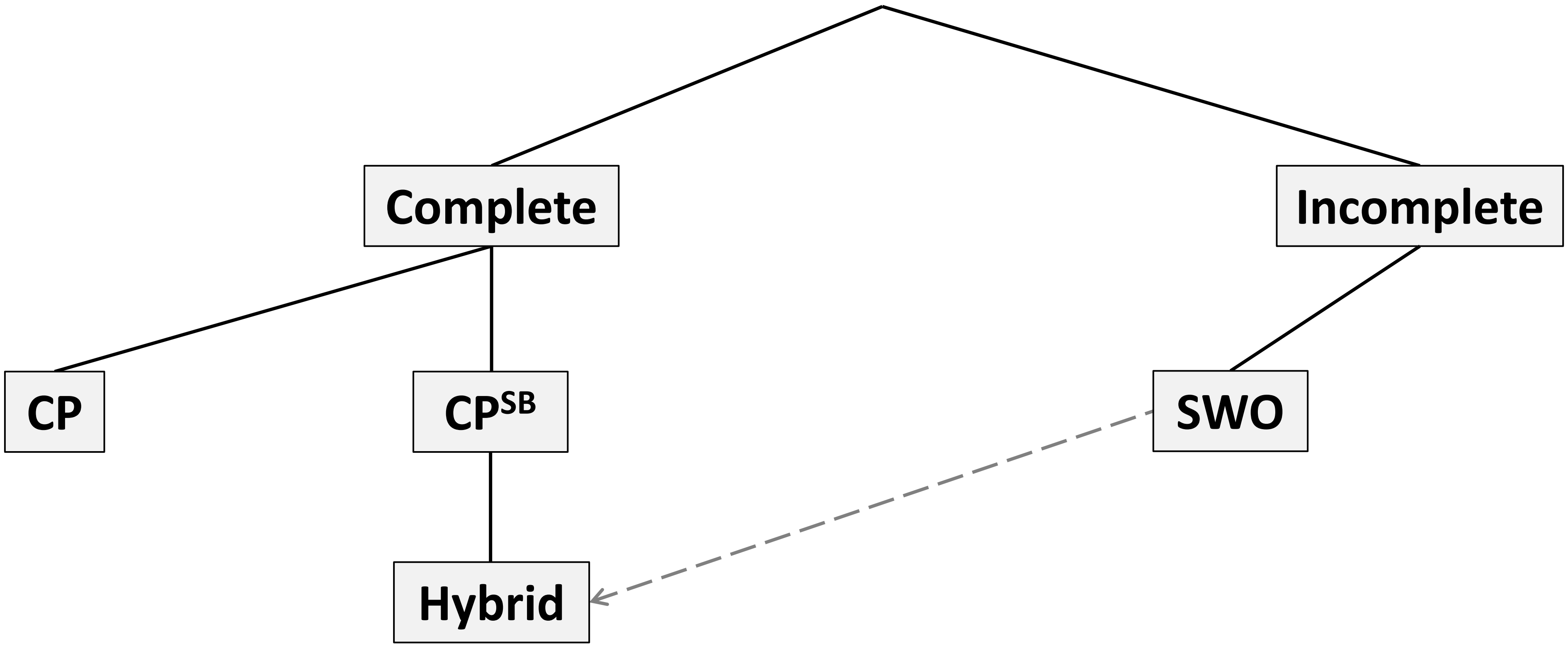
SWO (Greedy Algorithm)



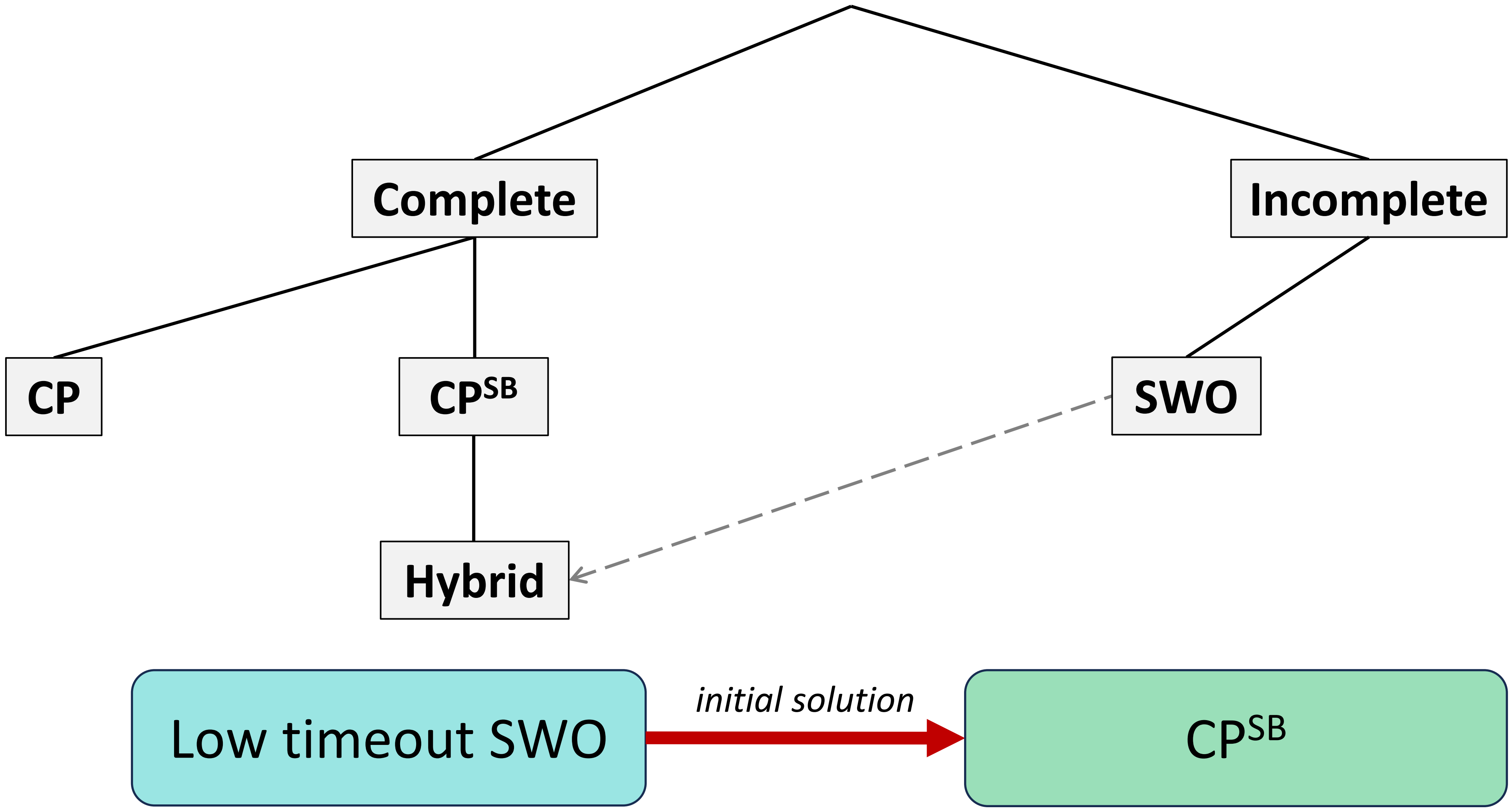
Solving Approaches



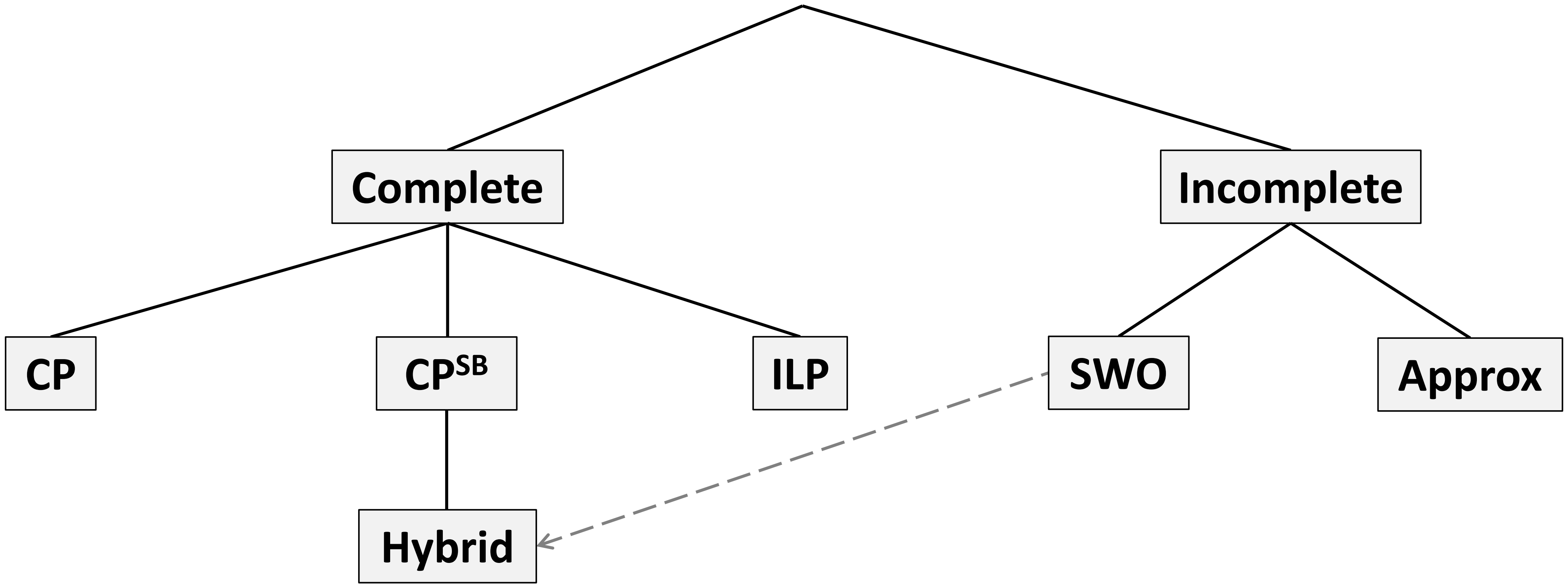
Solving Approaches



Solving Approaches



Solving Approaches



Solving Approaches

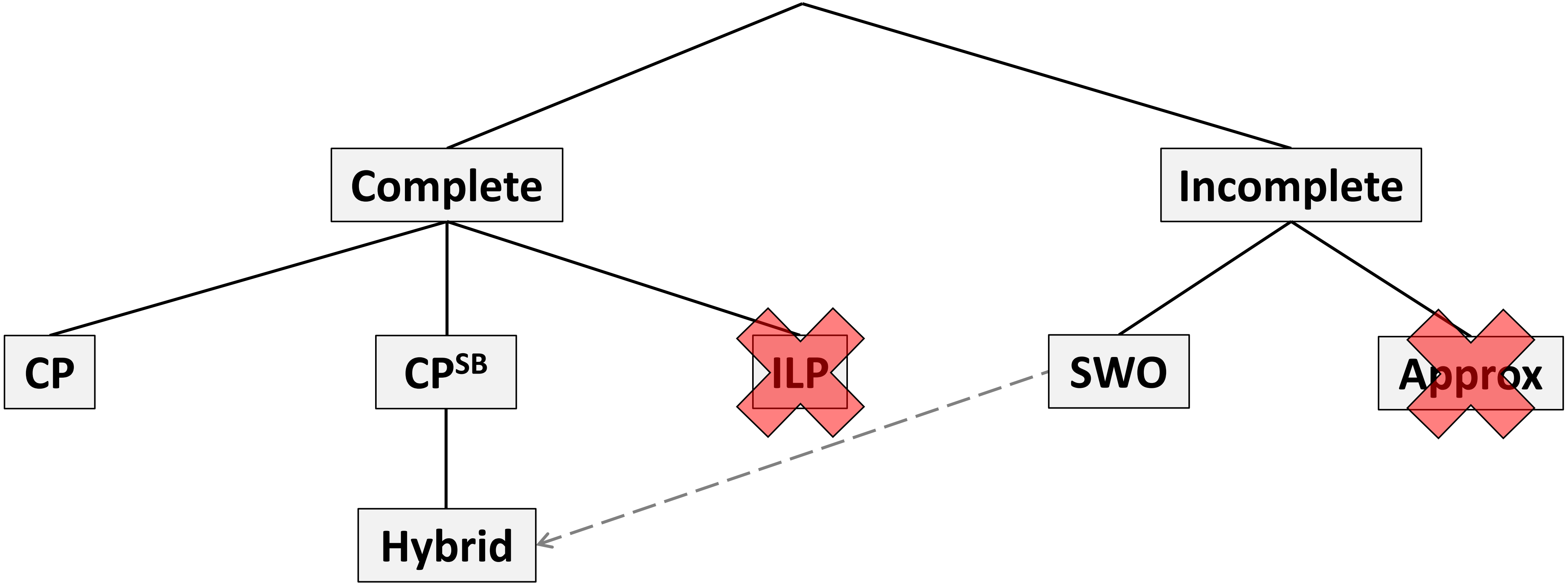


Table of Contents

Learning Curve

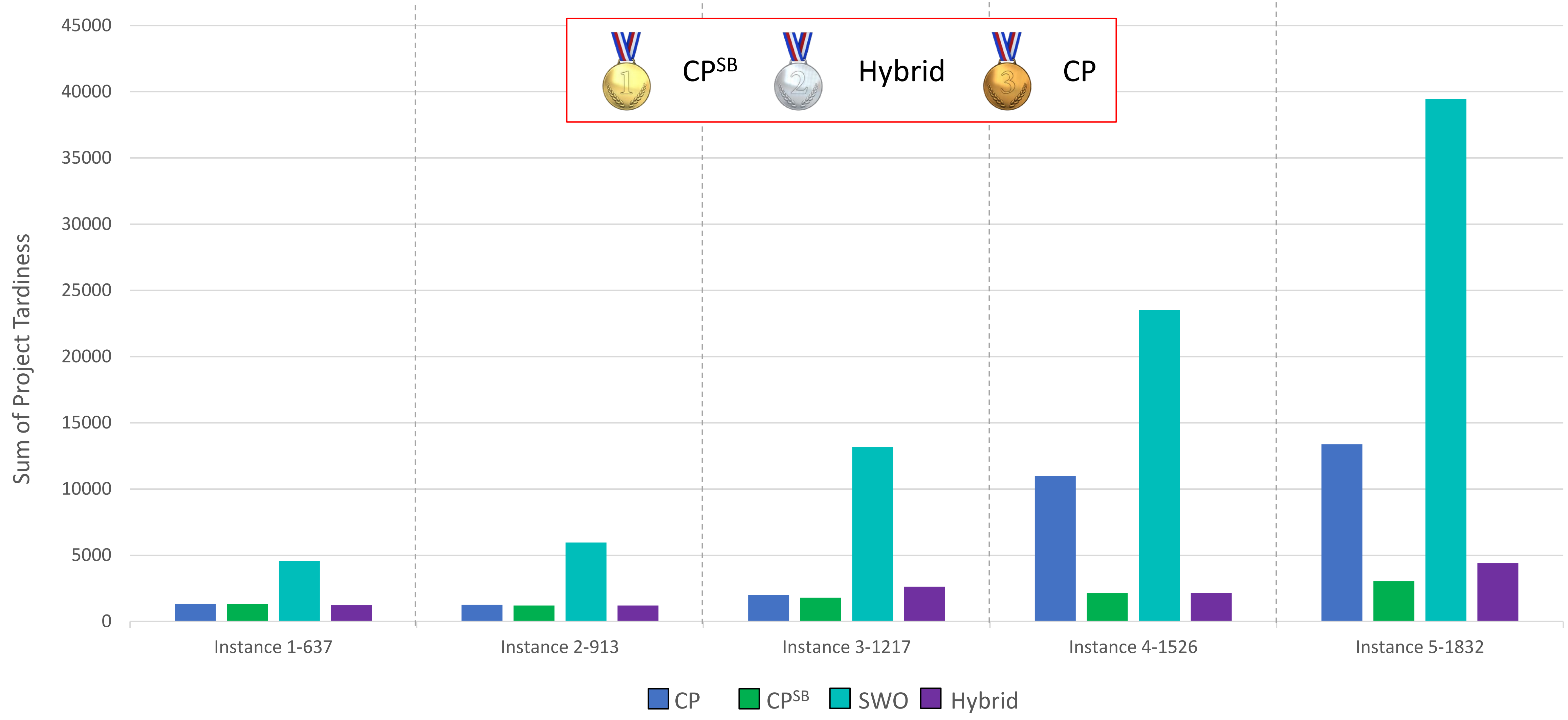
- $\forall n \in \mathbb{N}, \quad dur_a(n) = dur_a^\infty + \left\lceil (dur_a^0 - dur_a^\infty) * (n+1)^{\log_2 l} \right\rceil$

Industrial Benchmarks

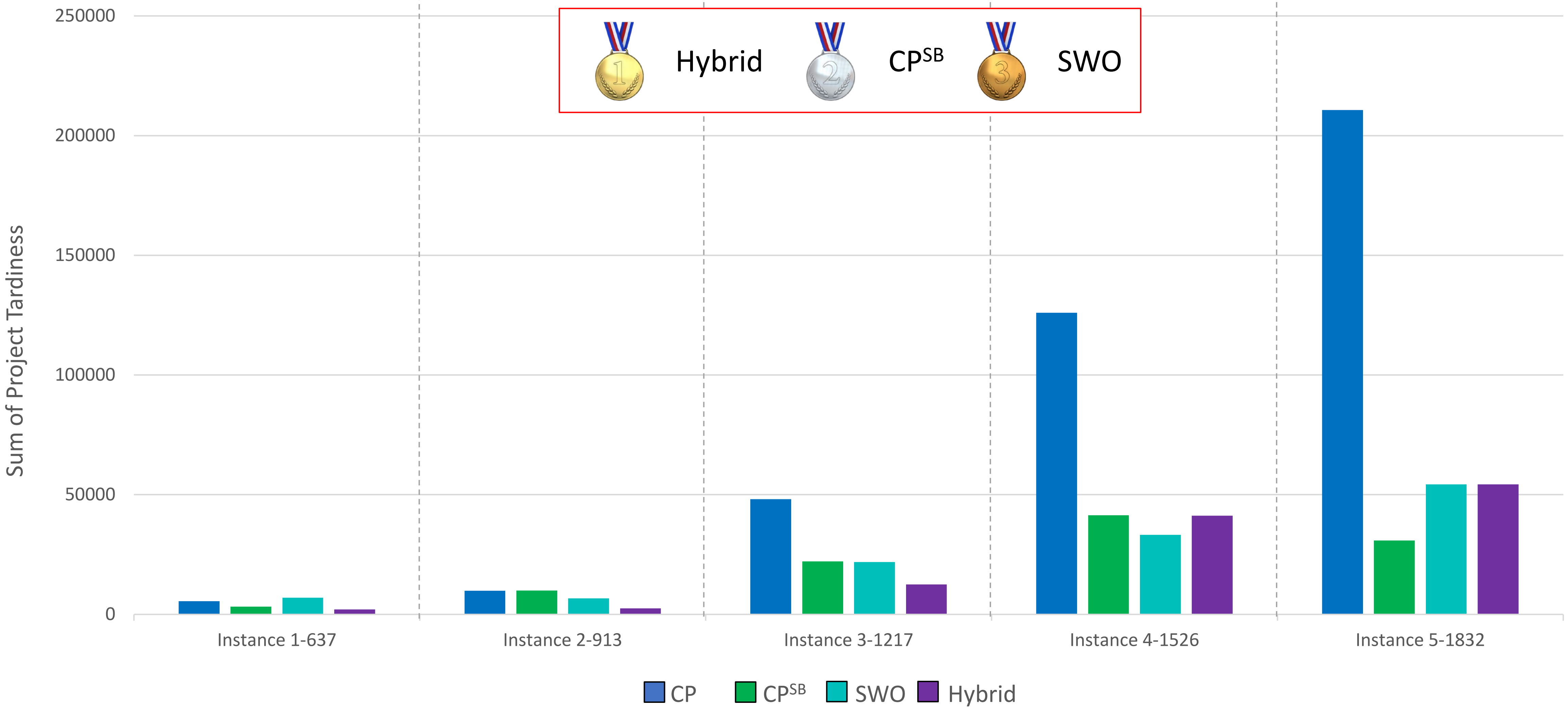
Instances		$\lvert C \rvert$	$\lvert A_c \rvert$	$\lvert A_c^{Comp} \rvert$	$\lvert I_c \rvert$	$\lvert R \rvert$	$\lvert capa_r \rvert$	l
Satellite	Original	3	≤ 30	≤ 3	≤ 5	40	≤ 16	$[0.05,0.95]$
	Extended	6			≤ 14			0.85

Test environment:
- IBM CP Optimizer 22.1.1 / Julia 1.10.1
- Timeout: 2h / 15s

Industrial Benchmarks (2h)



Industrial Benchmarks (15s)



PSP-based Benchmarks

1 class = 1 RCPSP from PSPLIB

- Several classes
- Multiplicity projects
- Random due dates
- Random learning rate
- Resource capacities = sum of PSPLIB resource capacities

Instances		$\lvert C\rvert$	$\lvert A_c\rvert$	$\lvert A_c^{Comp}\rvert$	$\lvert I_c\rvert$	$\lvert R\rvert$	$\lvert capa_r\rvert$	l
PSP-based	Small	$[\![2,4]\!]$	$\{30,60\}$	0	$[\![5,10]\!]$	4	$\sum_{i_{LIB}} capa_r^{i_{LIB}}$	$[0.45,0.95]$
	Large	$[\![5,7]\!]$	$\{60,90,120\}$					

PSP-based Benchmarks

- 50 instances
- Dataset available online

Timeout	Dataset	CP	CP^{SB}	SWO	Hybrid
2h	Small	21	21	0	21
	Large	17	16	0	16
15s	Small	22	16	0	19
	Large	15	4	6	3

Table of Contents

V. Conclusion

✓Learning effect is important in HM-RCPSP

✓CP-based approaches showed the best performance in tests

V. Conclusion

- ✓Learning effect is important in HM-RCPSP
- ✓CP-based approaches showed the best performance in tests

- ➢Explore another potential metaheuristic

V. Conclusion

- ✓ Learning effect is important in HM-RCPSP
- ✓ CP-based approaches showed the best performance in tests

- ➢ Explore another potential metaheuristic
- ➢ Dominance breaking variations

V. Conclusion

✓Learning effect is important in HM-RCPSP

✓CP-based approaches showed the best performance in tests

➢Explore another potential metaheuristic

➢Dominance breaking variations

➢Learning effect on similar activities between classes

V. Conclusion

- ✓ Learning effect is important in HM-RCPSP
- ✓ CP-based approaches showed the best performance in tests

- ➢ Explore another potential metaheuristic
- ➢ Dominance breaking variations
- ➢ Learning effect on similar activities between classes
- ➢ Uncertainty learning duration

Thank you for listening!