uc3m Universidad Carlos III de Madrid

[plg]

Carnegie Mellon University

icaps 2018 - Delft

IntEx Workshop

Task Monitoring and Rescheduling for Opportunity and Failure Management

José Carlos González, Manuela Veloso,

Fernando Fernández and Ángel García-Olaya

Planning and Learning Group

25 June 2018

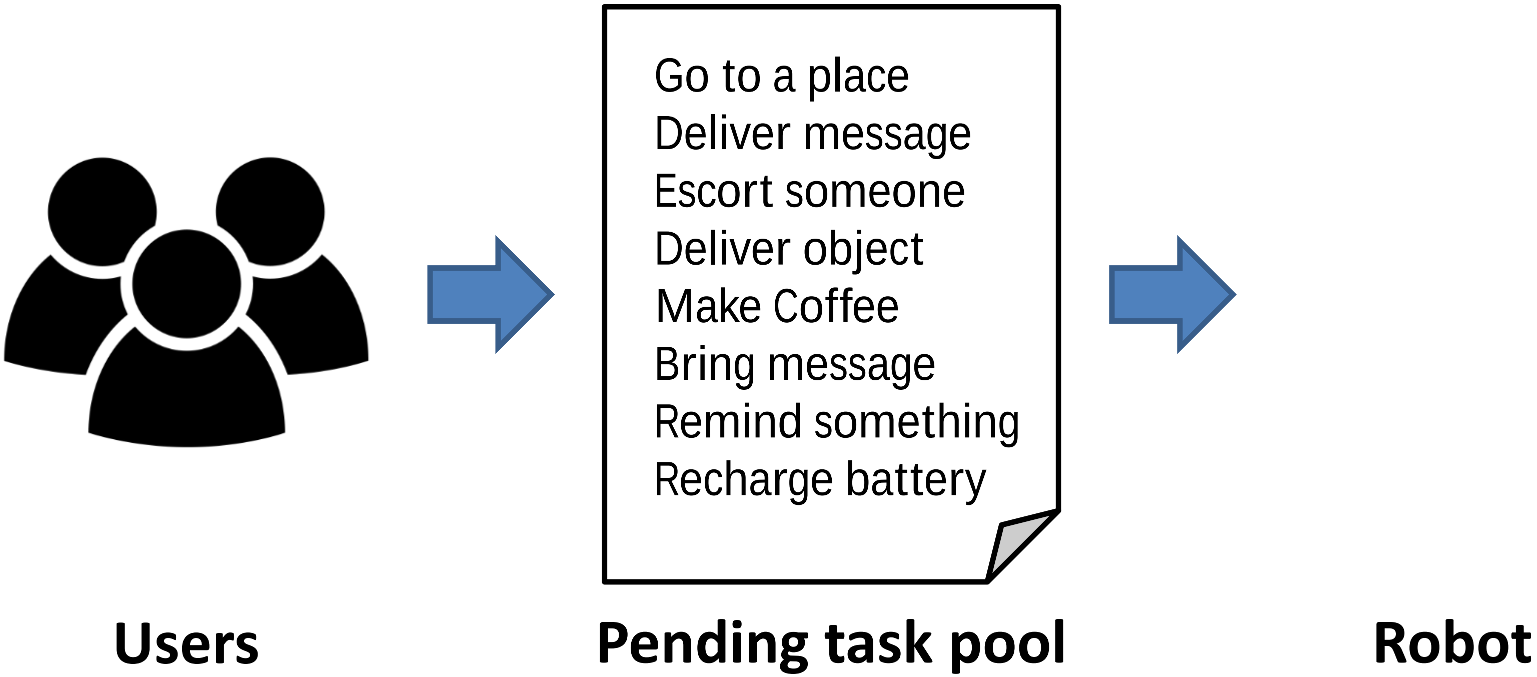
Computer Science Department

Introduction – Tasks of service robots

Å Robot must find a valid task schedule, and execute it

Å Several constraints per task

Å Users can add tasks anytime



Go to a place
Deliver message
Escort someone
Deliver object
Make Coffee
Bring message
Remind something
Recharge battery

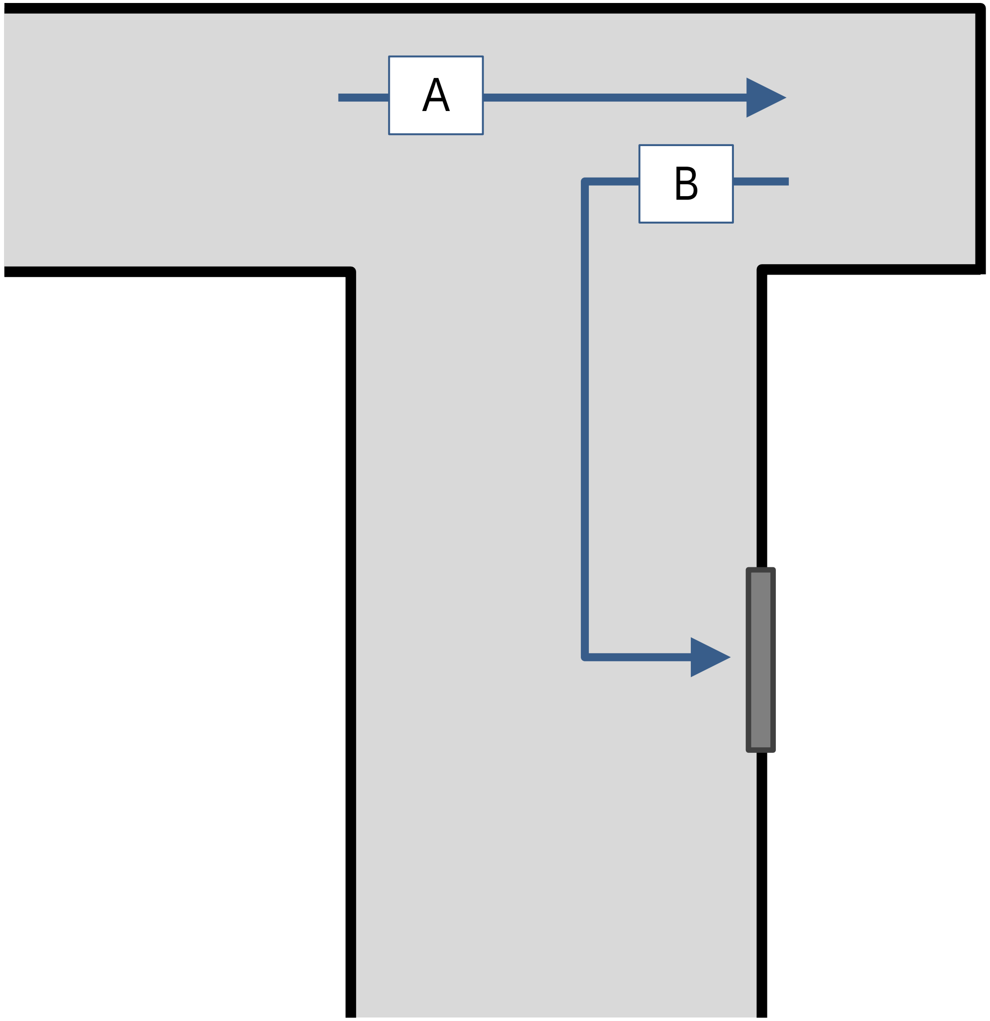
Users

Pending task pool

Robot

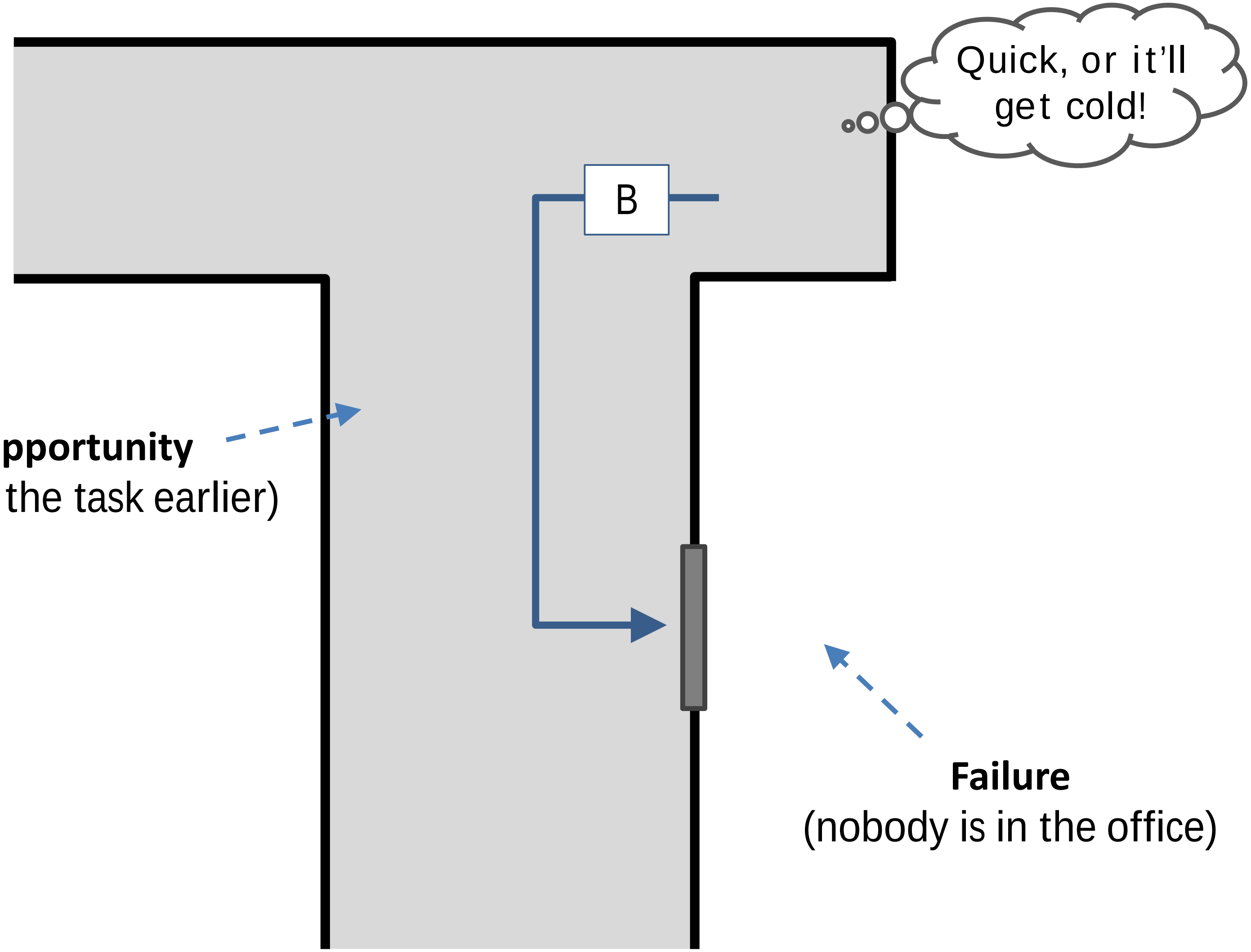
Introduction – Hot coffee delivering

Subtasks: A, B



Introduction – Hot coffee delivering

Subtasks: A, B

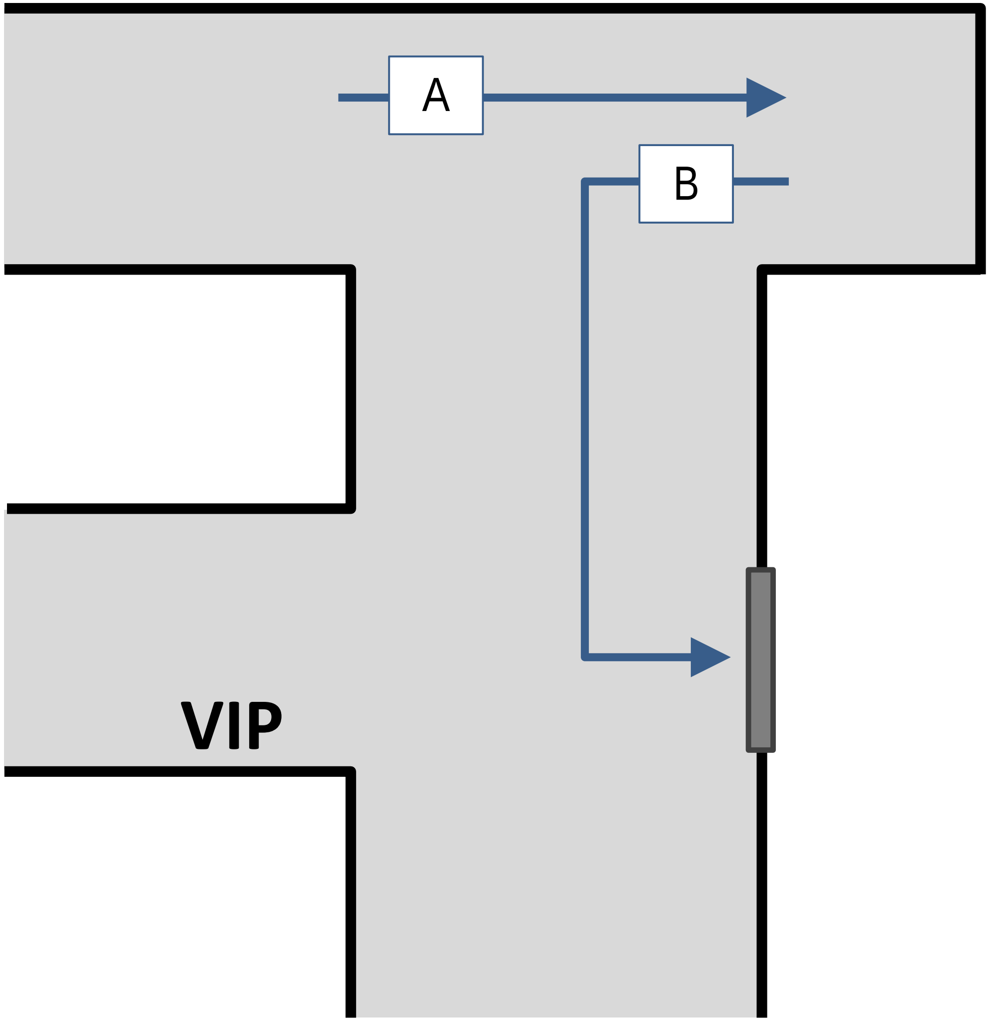


Opportunity
(finish the task earlier)

Failure
(nobody is in the office)

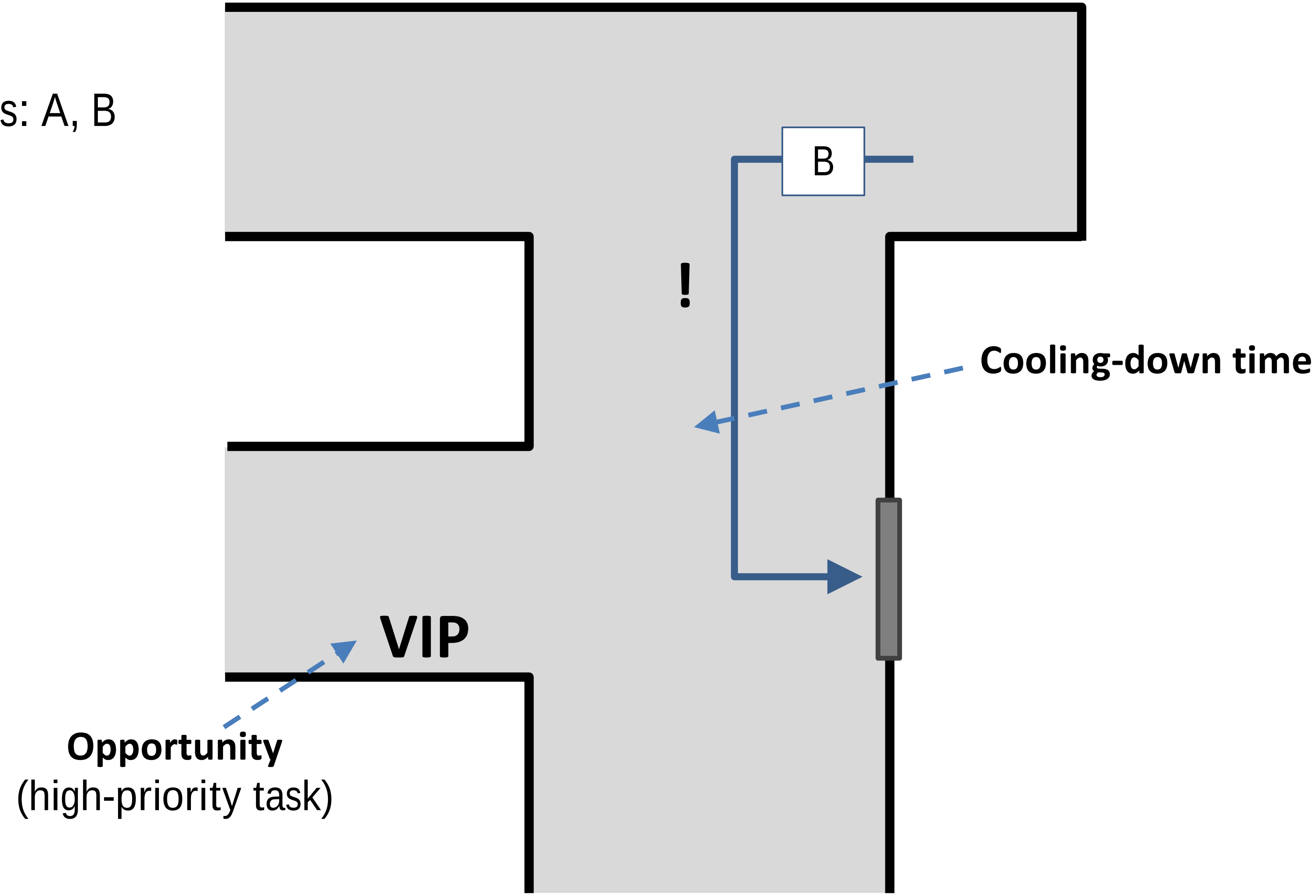
Introduction – Hot coffee delivering

Subtasks: A, B

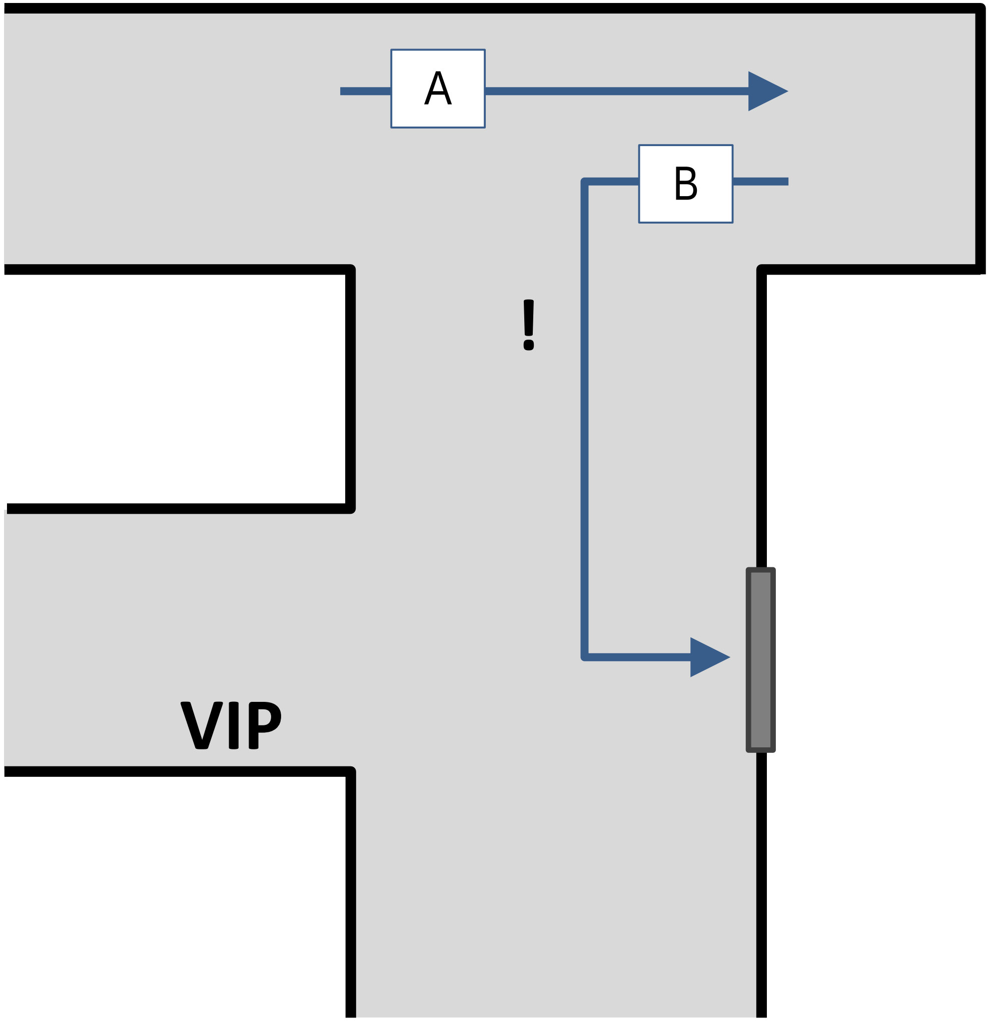


Introduction – Hot coffee delivering

Subtasks: A, B



Introduction – Hot coffee delivering



What to do now?

Å VIP first, then resume B

Å Redo A and B

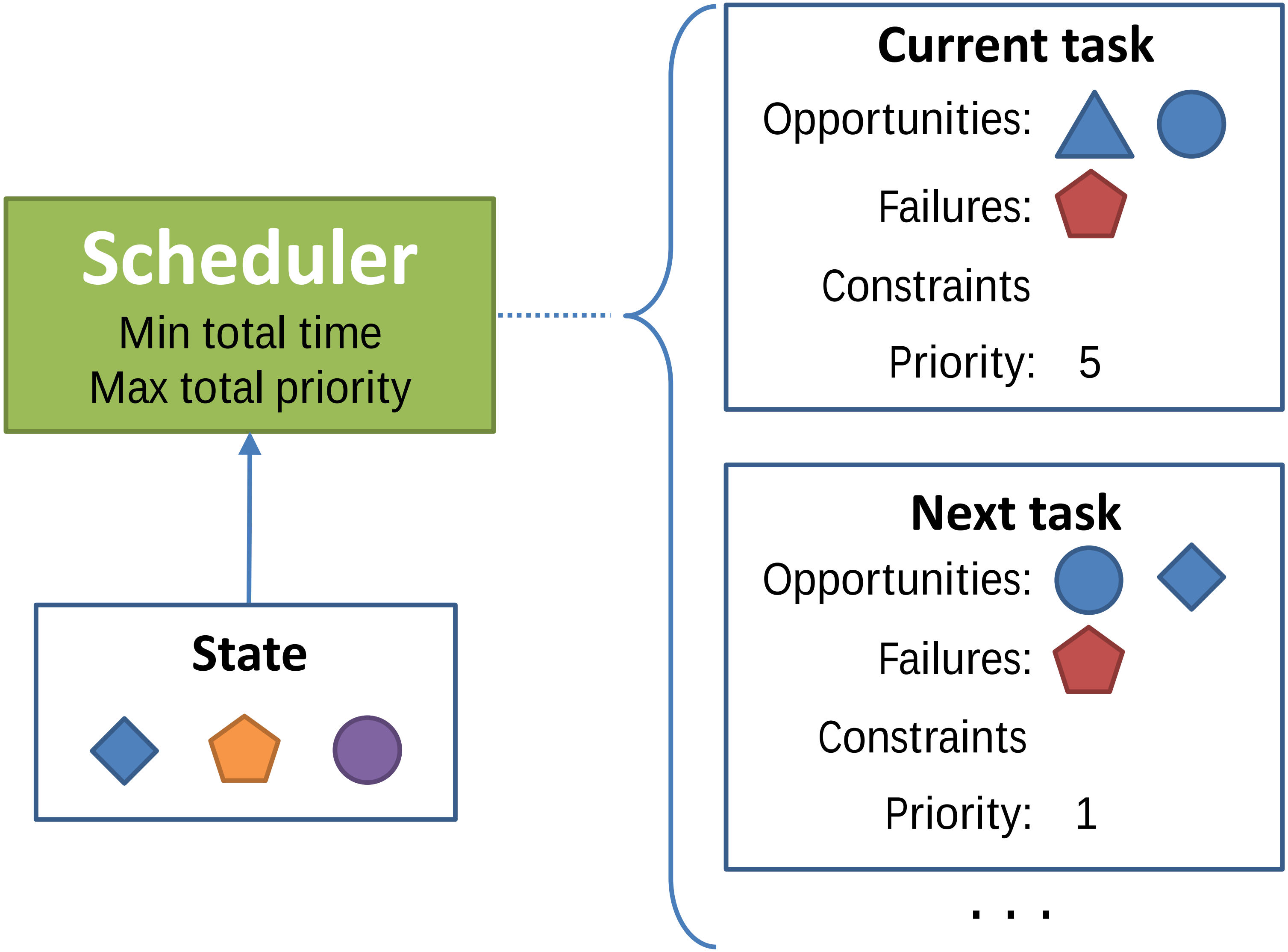
Å VIP after B

Å Cancel A and B

Å Cancel VIP

Å Try a quick VIP

Opportunities and Failures



Scheduler

Min total time

Max total priority

State

◆ ⬟ ●

Current task

Opportunities: ▲ ●

Failures: ⬟

Constraints

Priority: 5

Next task

Opportunities: ● ◆

Failures: ⬟

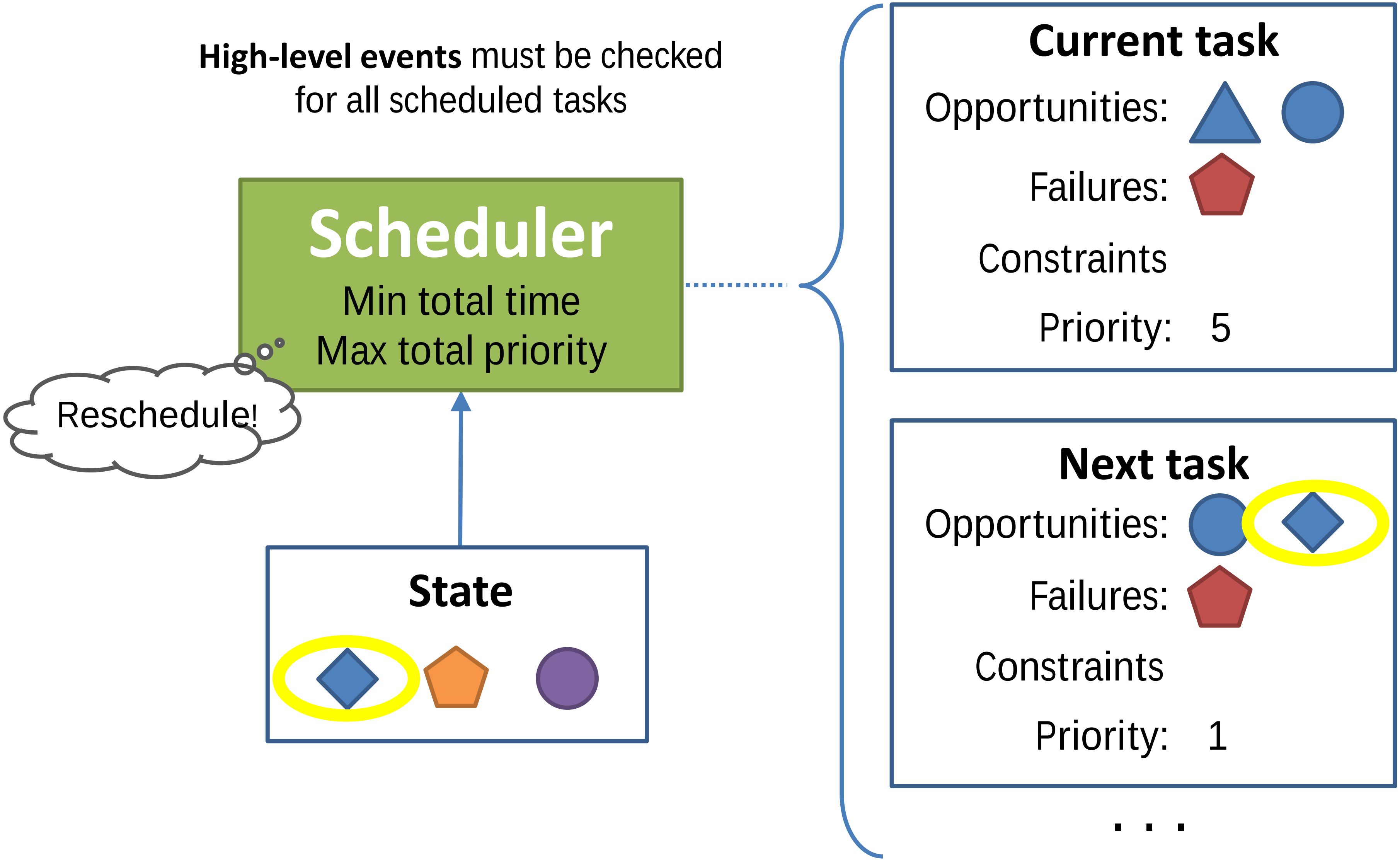
Constraints

Priority: 1

. . .

Opportunities and Failures

High-level events must be checked
for all scheduled tasks



Scheduler

Min total time
Max total priority

Reschedule!

State

◆ ⬟ ●

Current task

Opportunities: ▲ ●

Failures: ⬟

Constraints

Priority: 5

Next task

Opportunities: ● ◆

Failures: ⬟

Constraints

Priority: 1

. . .

Contribution and Related work

Å **Our contribution**

Á **Component** to handle high-level unexpected events among tasks

Á MIP model with dependent tasks and **cooling-down times**

Å Coltin, B.; Veloso, M. M.; and Ventura, R. 2011.
Dynamic user task scheduling for mobile robots

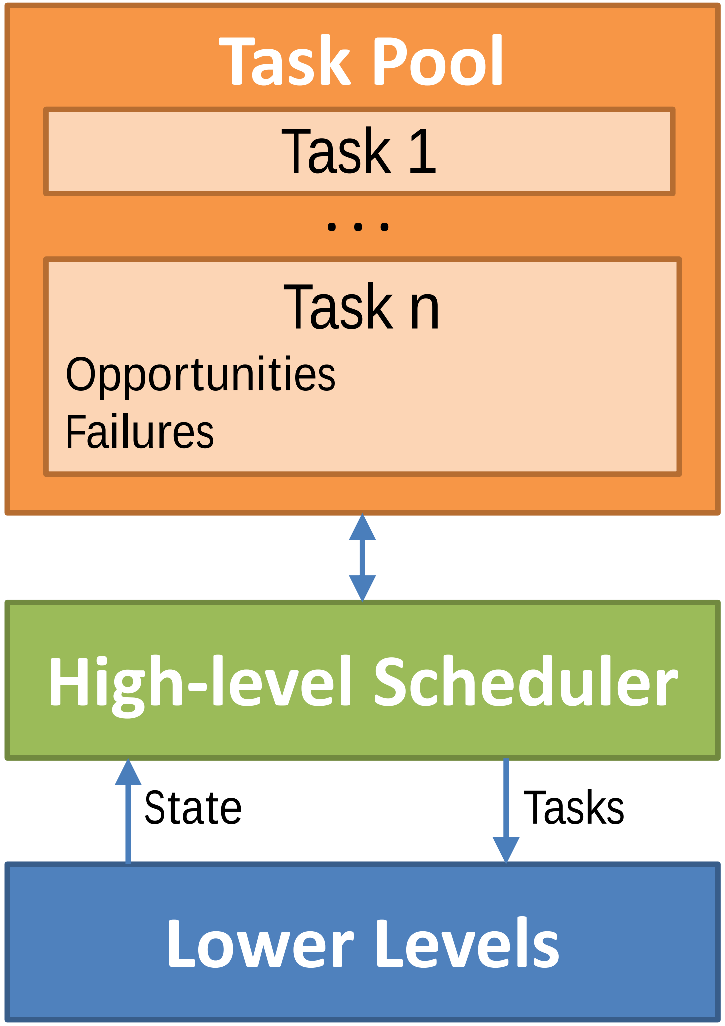
Á Fixed schedules with a Mixed Integer Programming (MIP) solver

Our starting point

Å Cashmore, M.; Fox, M.; Long, D.; et al. 2017.
Opportunistic Planning in Autonomous Underwater Missions

Å Schermerhorn, P.; Benton, J.; Scheutz, M.; et al. 2009.
Finding and Exploiting Goal Opportunities in Real-Time During Plan Execution

Monitoring model



Task Pool

Task 1

. . .

Task n

Opportunities

Failures

High-level Scheduler

State

Tasks

Lower Levels

Å **Updated states** received while subtasks are being executed

Å **Generic task attributes** *Opportunities* and *Failures*

Á Indicate parameters in the state that should remain invariant

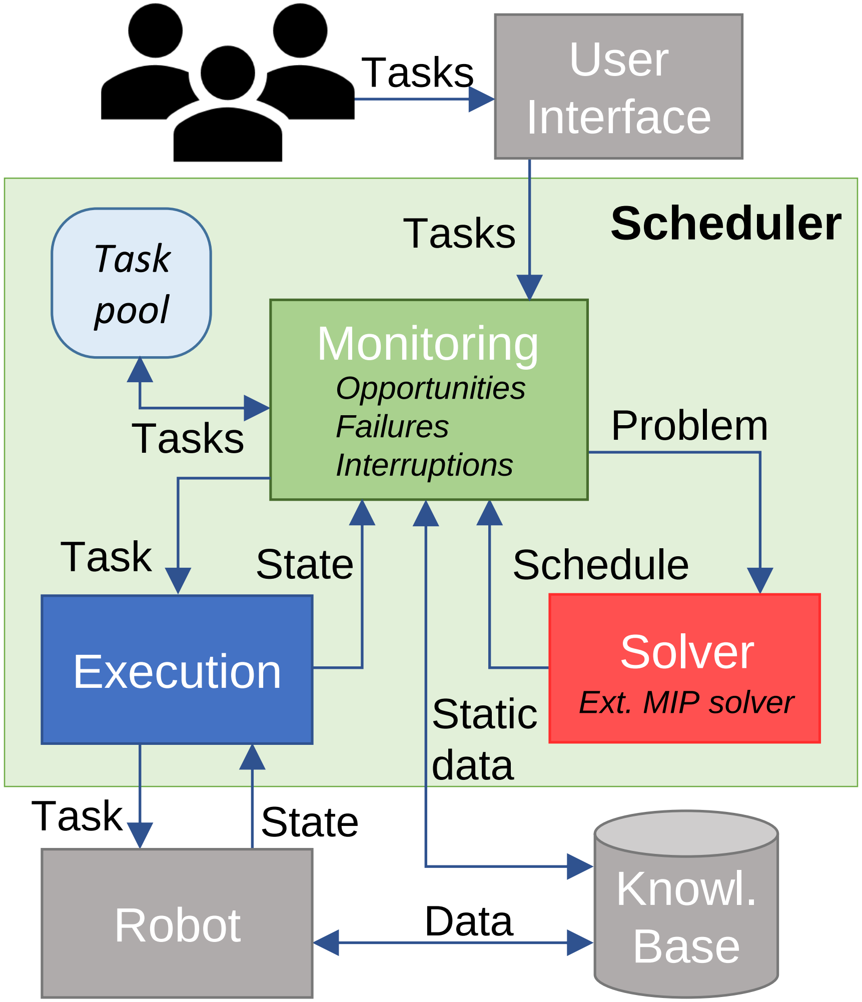
Á Used to trigger reschedulings

Å A **rescheduling** can

Á Add or remove tasks in the pool

Á Interrupt the current subtask

High-level Task Scheduler Architecture



Å **Multilevel global scheme**

Á Rescheduling for high-level events

Á Tasks sent to lower abstraction levels

Á States are generalized from lower levels

Task modeling and decomposition

		Task	Subtask-1	Subtask-2
User	Task type	DeliverDrink	MakeHotDrink	DeliverObject
	Task owner	Alice	*Alice*	*Alice*
	Location start	-	CoffeMaker	CoffeMaker
	Location end	-	CoffeMaker	AliceOffice
	Time start min	0	*0*	*0*
	Time end max	15	*15*	*15*
	Person target	Alice	-	Alice
	Object	HotCoffee	HotCoffee	HotCoffee
	Priority	-	10	10
Internal	Time operation	-	5	2
	Time cooldown	-	-	6
	Task depending	-	-	Subtask-1
	Opportunities	VIP	HotCoffee, *VIP*	Person target, *VIP*
	Failures	TO, BP	*TO, BP*	HotCoffee, *TO, BP*

MIP model with cooling-down time

Constraints:

$$w_i^{min} \leq s_i \leq w_i^{max} - o_i - d(l_i^s, l_i^e)$$
$$w_i^{min} + o_i + d(l_i^s, l_i^e) \leq e_i \leq w_i^{max}$$
$$\left.\begin{array}{l} Previous(i,j) \Rightarrow s_i < e_i < s_j \\ \neg Previous(i,j) \Rightarrow s_i < s_k < s_j \end{array}\right\} \cdots\cdots\blacktriangleright$$

Order and overlapping

$$Previous(i,j) \Rightarrow e_j \geq s_j + o_j + d(l_i^e, l_j^s) + d(l_j^s, l_j^e)$$
$$\left.\begin{array}{l} Depends(j,i) \Rightarrow e_i < s_j \\ Depends(j,i) \Rightarrow c_j \geq e_j - e_i \end{array}\right\} \cdots\cdots\blacktriangleright$$

Depending subtasks and cooling-down

Objective function:

$$\text{Minimize} \sum_{i=1}^{n} e_i p_i$$

Checks:

$$w_i^{min} + o_i + d(l_i^s . l_i^e) < w_i^{max}$$
$$c_i \geq o_i + d(l_i^s . l_i^e)$$

Solution types

- Á Proven optimal
- Á Suboptimal
- Á Not found
  - Unfeasible
  - Time limit

Positive integer parameters:

i, j, k: Any task of the pool
w^{min}: Minimum start time
w^{max}: Maximum end time
s: Start time (variable)
e: Ending time (variable)
o: Operation time
c: Cooling down time
p: Priority value higher than 0
l^s: Starting location
l^e: Ending location
$d(a, b)$: Distance (time estimation) between a and b

Binary parameters:

$Previous(i, j)$: Task i starts just before j (variable)
$Depends(j, i)$: Task j must start after i

Rescheduling policy

- If the scheduler **cannot find a suitable plan**
  - **Failures**: Monitoring cancels the next task
    - With the lowest priority first
    - Then the smallest time window that overlaps another
  - **Opportunities**:
    1. Tries to redo the current subtask later
    2. If it cannot, it tries to redo the whole task
    3. If it cannot, it evaluates whether to cancel the current task or the new task by maximizing the gain measure *g*

$$\boxed{\textbf{Gain:}\quad g = \sum_{i=1}^{n} p_i}$$

Sum of the priorities of the scheduled tasks

Experiments – CoBot robots

Å **Using the CoBot platform**

  Å Their task catalog

  Å Schedules work in the actual robot

Å **180 simulations**

  Å Scheduling times

  Å Quality

Experiments – Schedules

Å **Task decomposition** allows to optimize locations

Schedule 1

Task	Start	End
…	0	10
C1a	11	20
C2a	21	26
C1b	27	31
C2b	32	33
C3a	34	42
C3b	43	47
VIP	48	53
Cost	739	

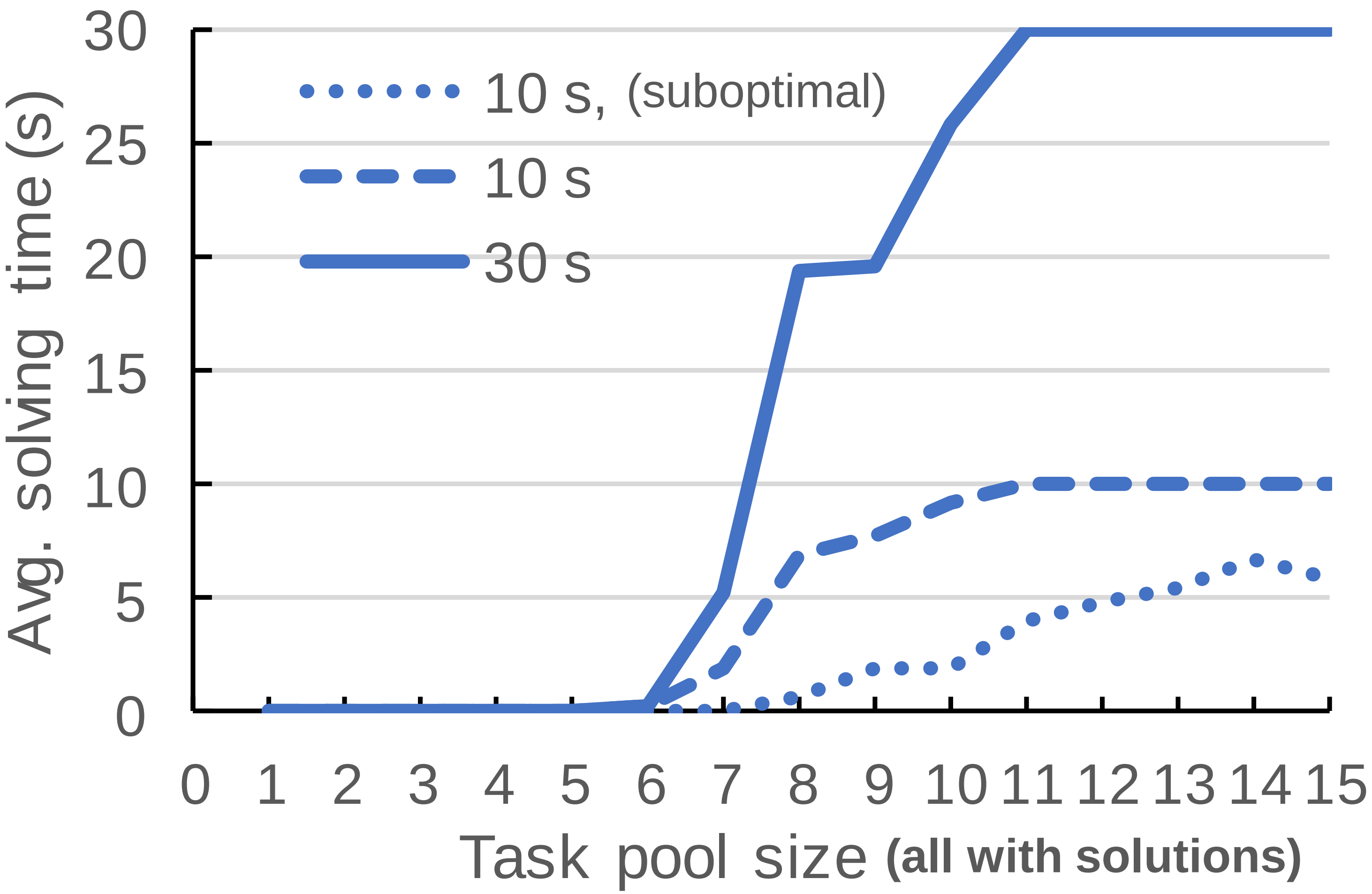
Schedule 2

Task	Start	End
…	0	10
C1a	11	20
C2a	21	26
C1b	27	31
C2b	32	33
VIP	34	39
C3a	40	45
C3b	46	50
Cost	605	

Schedule 3

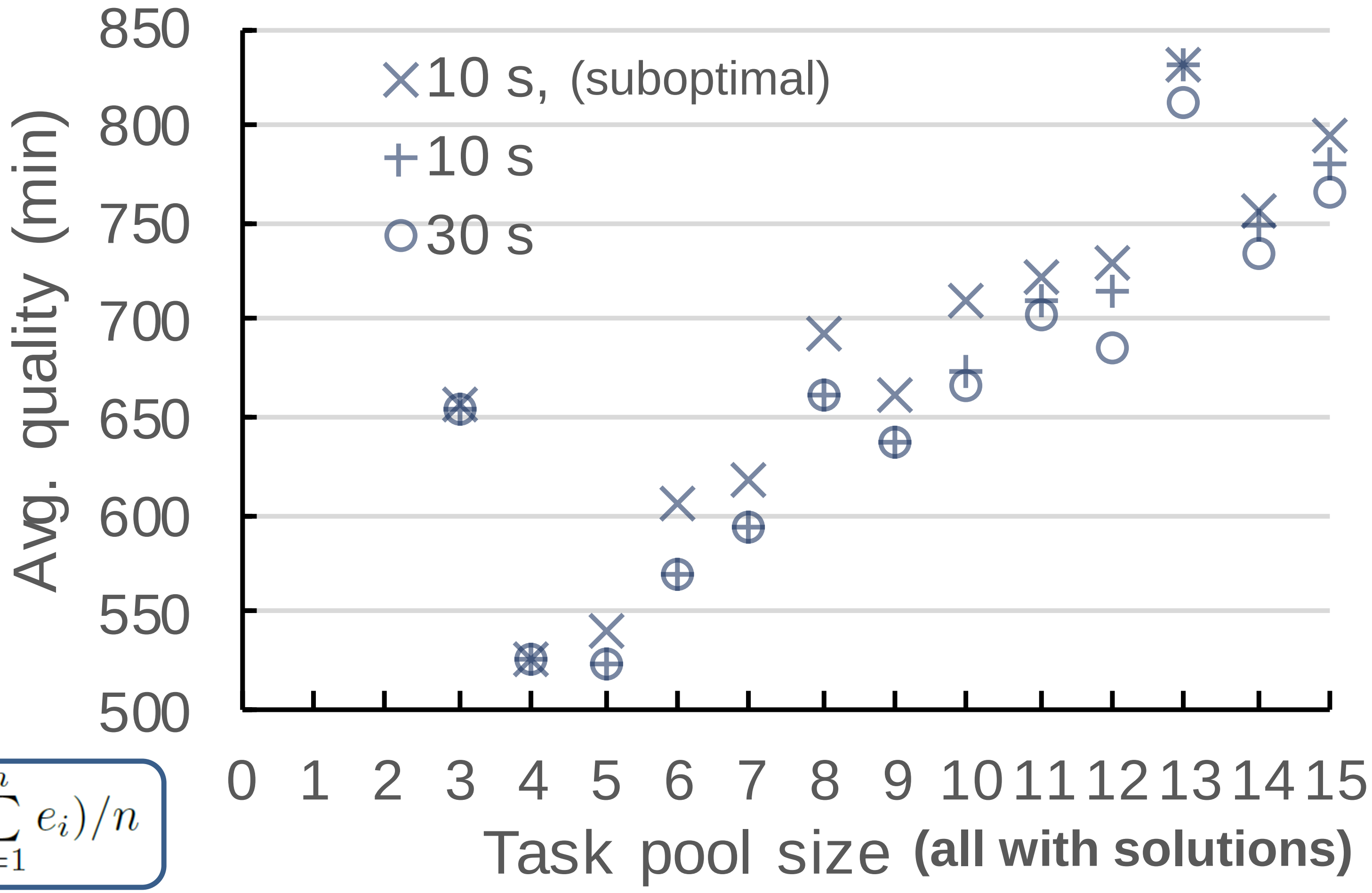
Task	Start	End
…	0	10
C1a	11	20
VIP	21	23
C2a	24	29
C1b	30	34
C2b	35	36
C3a	37	45
C3b	46	50
Cost	454	

Experiments – Solving time vs. Subtasks



Å Proven optimal solutions found up to size 10

Experiments – Quality vs. Subtasks



$$q = (\sum_{i=1}^{n} e_i)/n$$

Å vdzl·fʎúè ʎƴάмлă ădzō2LJúʎY l·fέ ʎă l·ŎŎŜLJúl·ōfŜ Ŧ2Ŋ úKŜ /2.2úΩă Ř2Y l·ʎƴ

Conclusions

Å **New architecture** of task execution, monitoring and rescheduling

- Á **Rescheduling** according to **opportunities and failures**
- Á **Interruption of tasks** in the middle of their execution
- Á **Future work**: integration with a generic hierarchical control architecture, independent from the planning/scheduling mechanism

Å **Improved MIP model**

- Á Able to deal with **cooling-down times** and dependent tasks
- Á Focused on the **quality of the solutions**
- Á **Quality can be affected in extreme conditions** with large task pools and fast solving times required
- Á **Future work**:
  - Transform some hard-constraints (time-window) into soft
  - Comparisons with other rescheduling systems

IntEx
Workshop

Task Monitoring and Rescheduling for Opportunity and Failure Management

José Carlos González, Manuela Veloso,

Fernando Fernández and Ángel García-Olaya

Planning and Learning Group

Thank you for your attention

25 June 2018

Computer Science Department

Opportunities and Failures

- **High-level events**
  - Affect the **current task and future tasks** in the schedule
  - **Interrupt tasks** in the middle of their execution

- **Opportunities**
  - Domain: can appear at any moment (VIP)
  - Specific: exclusive for a task (receipt of the coffee found earlier)

- **Failures**
  - Domain: same failure for several tasks (blocked paths, timeout)
  - Specific: exclusive for a task (coffee stolen)

Experimental sets

Experimental sets A>B>C

Å **A**: 480 random instances (task pools)

Å **B**: 12 solved instances per each pool size from 1-15 (180 in total)

Å **C**: 12 random instances per each pool size from 8-15 (96 in total)

Experiments – Solution types

Set A

Set B

Set C