An Elegant and Efficient Implementation of Russian Dolls Search for Variable Weighted CSP

T. Benoist, M. Lemaître

{tbenoist}@bouygues.com

Bouygues e-lab, Paris, France
ONERA, Toulouse, France

Variable Weighted CSP

- **Definition:**
  - Additive WCSP where only unary constraints ($X_i=1$) are weighted
  - maximize $\Omega = \sum w_i X_i$
  - Subject to: $X_i \in \{0,1\}$ + selection constraints
- **Some VWCSPs:**
  - knapsack problems,
  - «soft scheduling »
  - prize-collecting TSP
  - …

e-lab ONERA

Example: *Select & Schedule*

10 candidate photos, each with gain $\boldsymbol{w_i}$

→Select the subset (*selection* variables X_i)

- of higher gain ($\Omega=\sum w_i X_i$)
- that can be scheduled (variables T_i) without violating transition times $t_{i\rightarrow j}$ constraints.

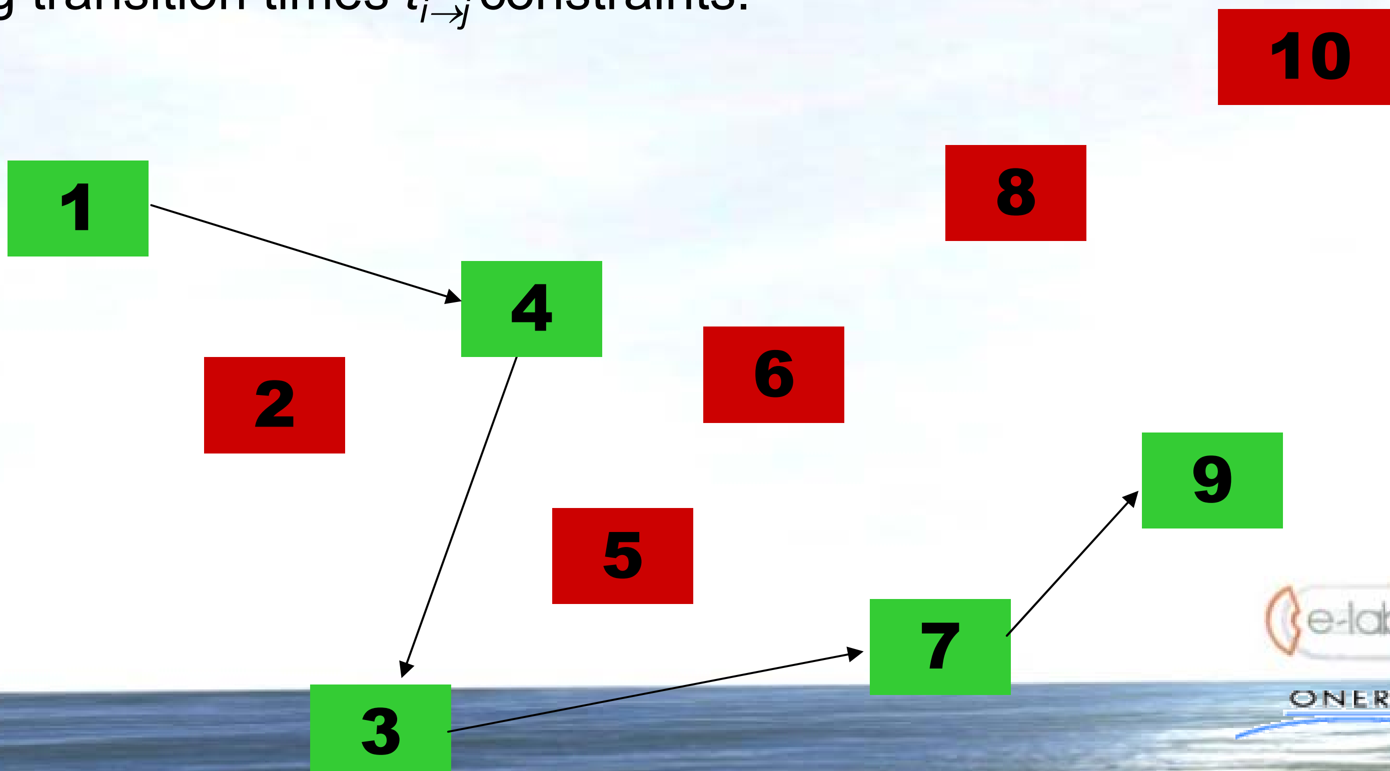
Simplified version of *SRSS [Verfaillie & Lemaître CP'01]*

Example: *Select & Schedule*

10 candidate photos, each with gain $\boldsymbol{w_i}$

→Select the subset (*selection* variables X_i)

- of higher gain ($\Omega=\sum w_i X_i$)
- that can be scheduled (variables T_i) without violating transition times $t_{i\rightarrow j}$ constraints.



Simplified version of *SRSS [Verfaillie & Lemaître CP'01]*

Russian Dolls Search

- Static order on selection variables
- Successive resolutions of nested sub problems

$Rds_{10}=\max(P_{10})$

Russian Dolls Search

- **Static order on selection variables**
- **Successive resolutions of nested sub problems**

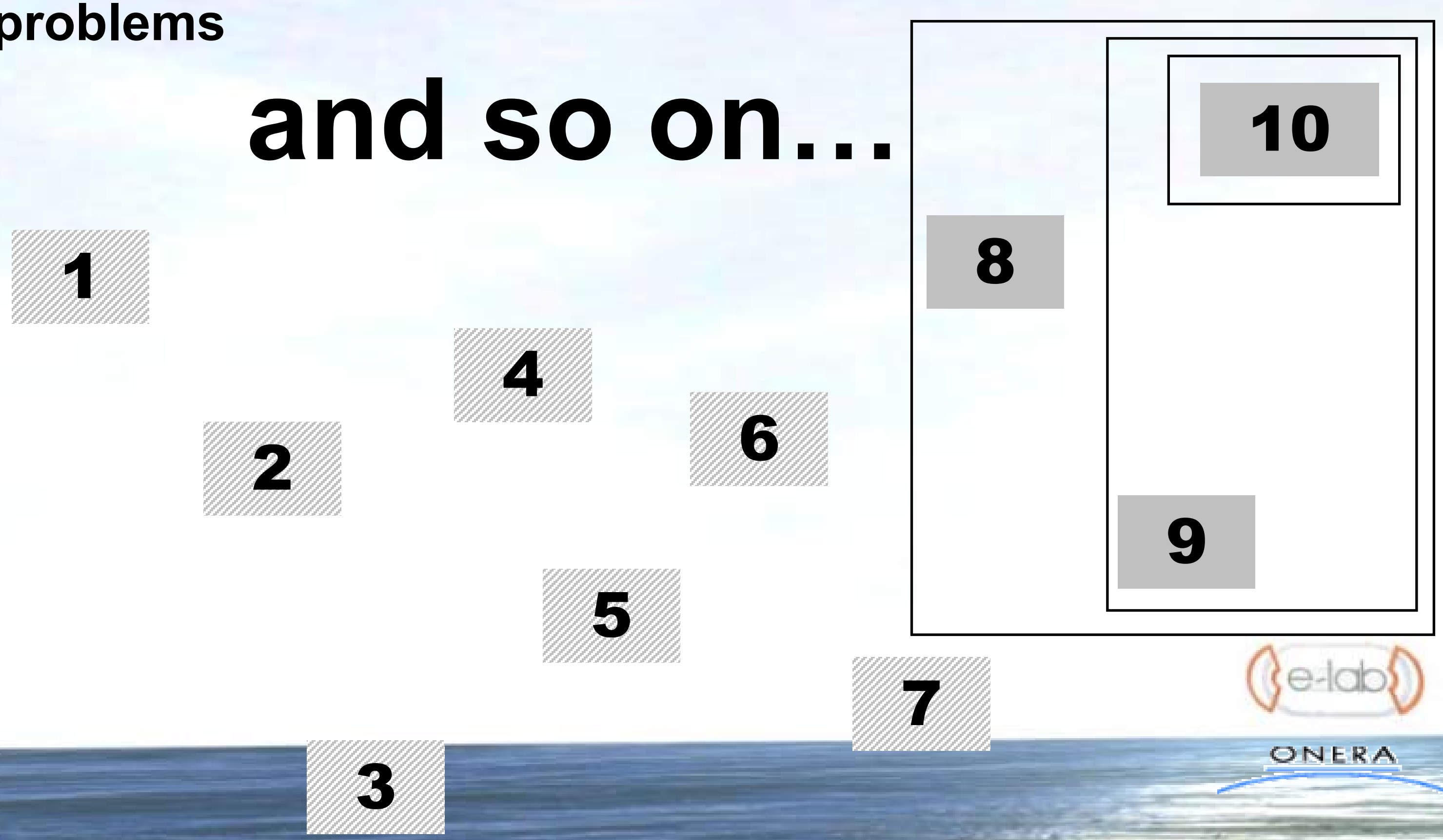
$$Rds_9 = \max(P_9)$$

Russian Dolls Search

- Static order on selection variables
- Successive resolutions of nested sub problems

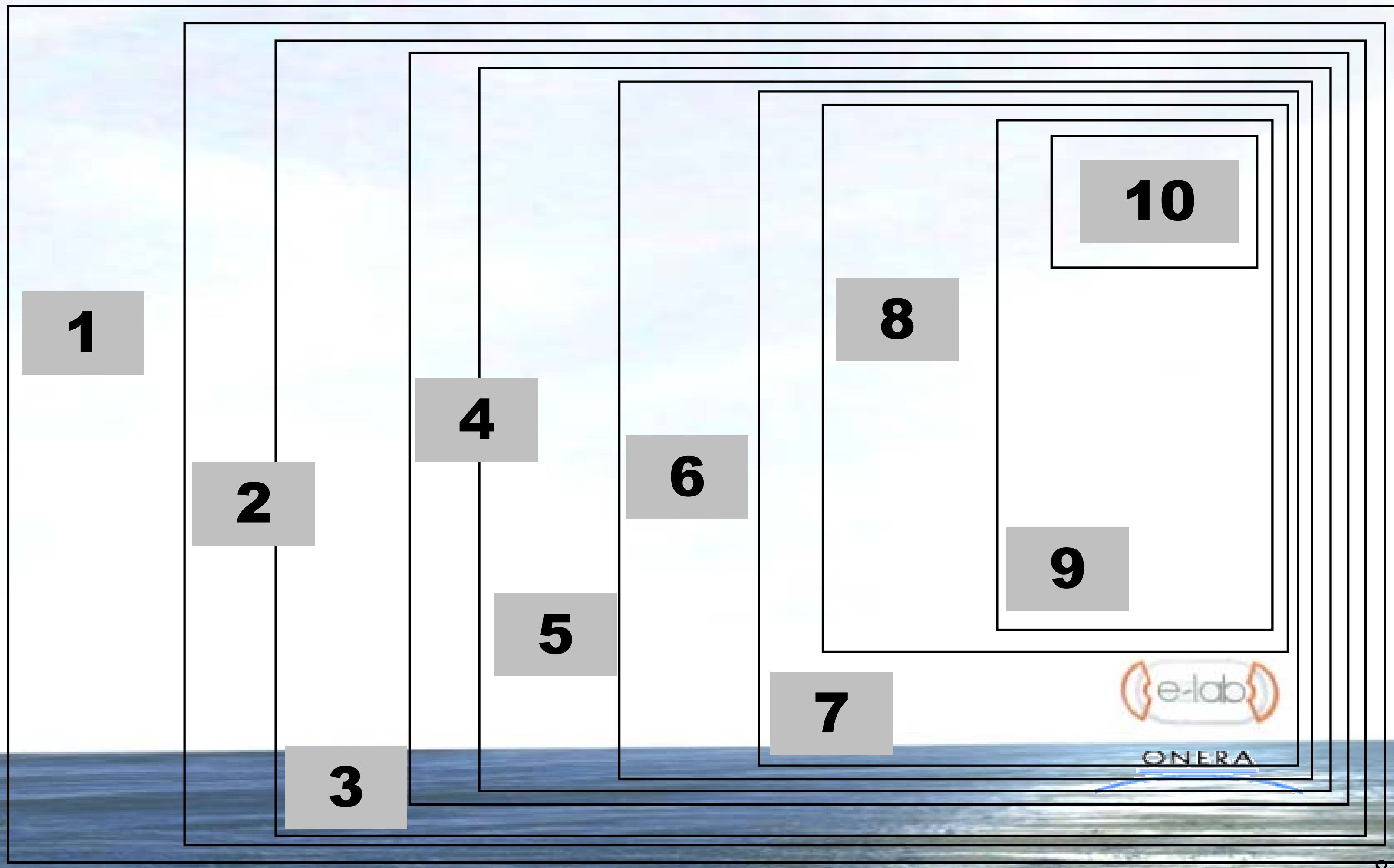
$Rds_8 = max(P_8)$

and so on…



Russian Dolls Search

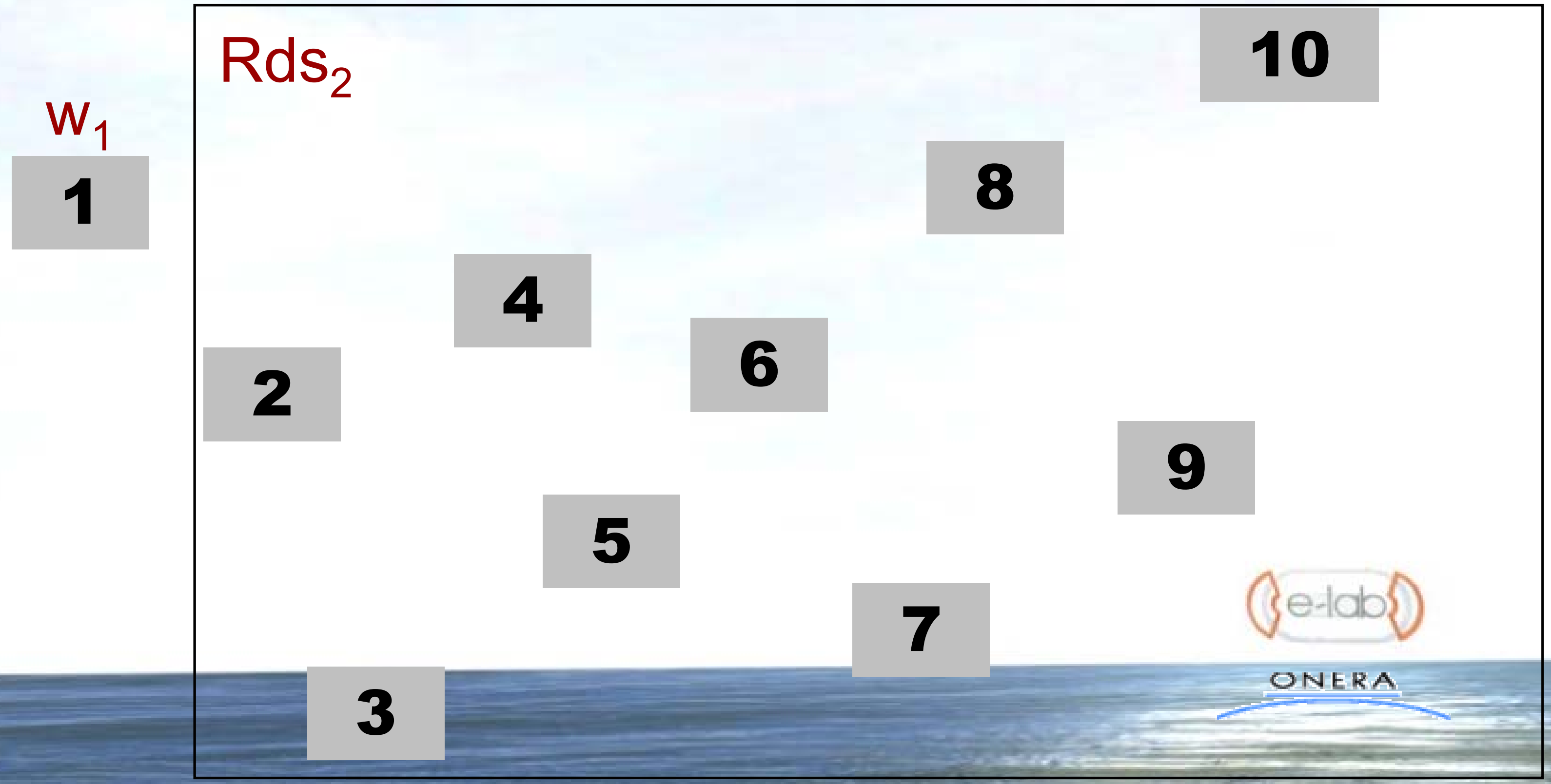
$Rds_1 = \max(P_1) = \max(P)$



RDS upper bound

During resolution of the last problem (P_1)
(once $P_{10}, P_9, \ldots, P_2$ have been solved)

Initial upper bound: $\Omega \leq w_1 + Rds_2$



RDS upper bound

10

8

1

4

6

2

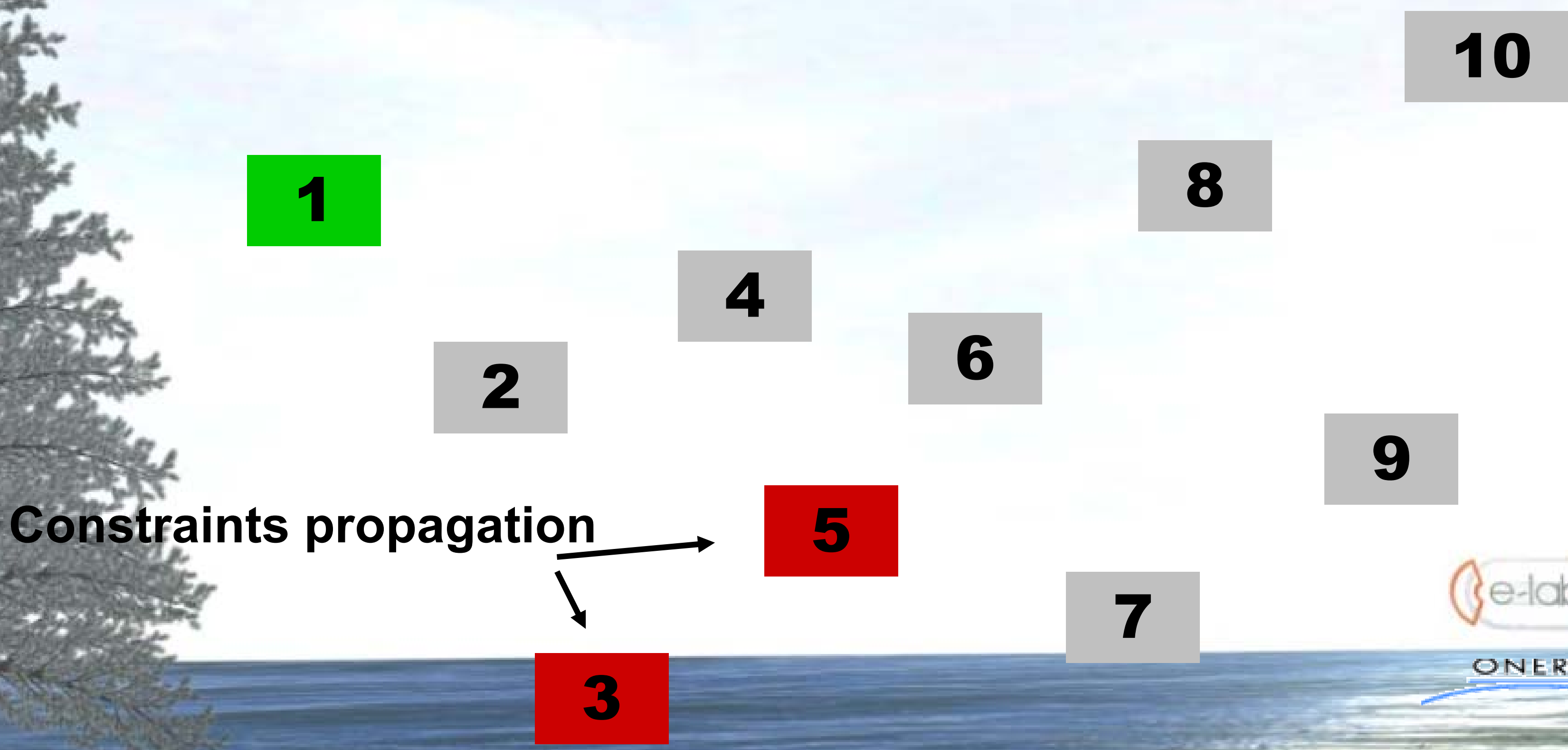
9

5

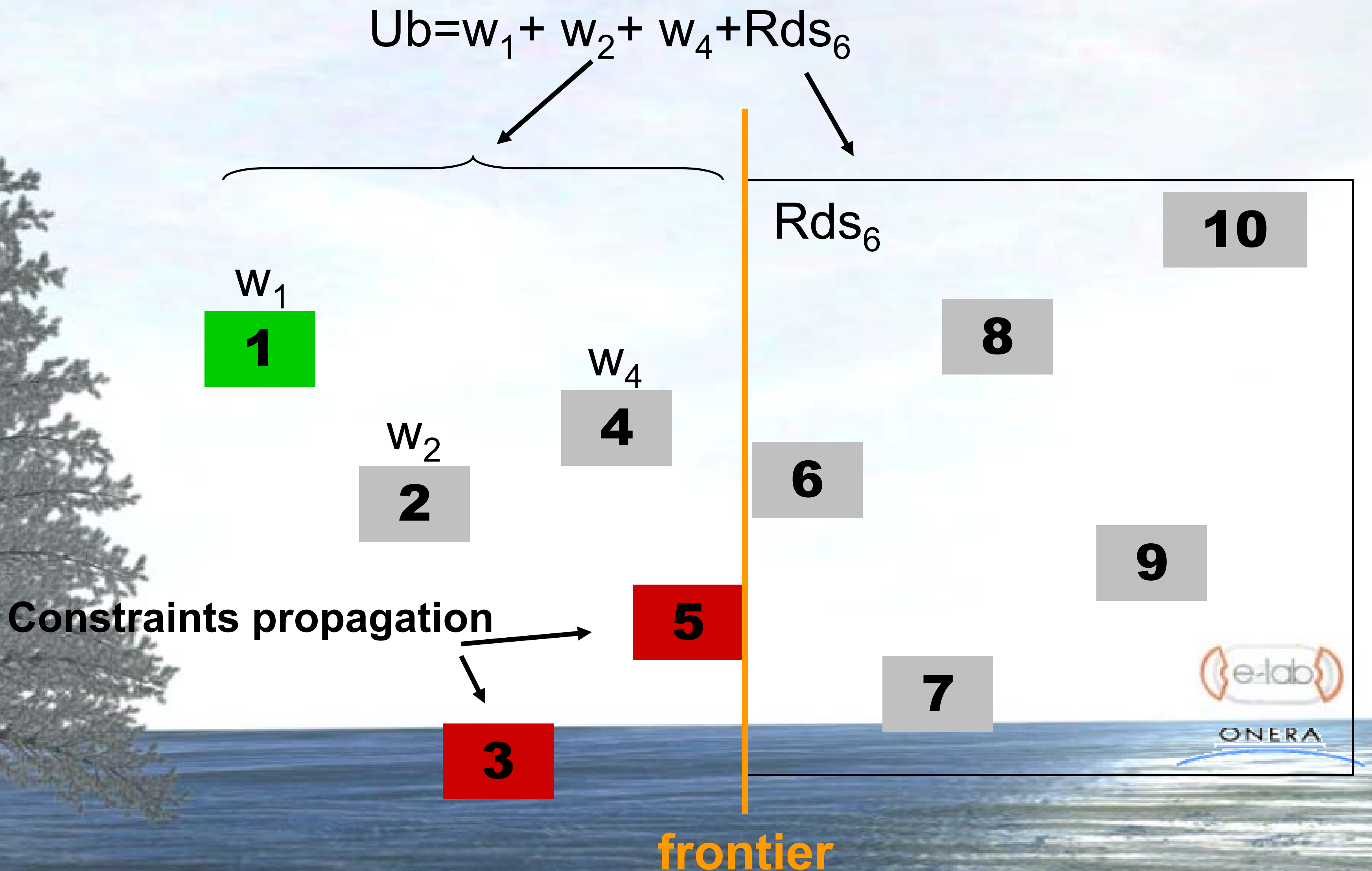
7

3

RDS upper bound



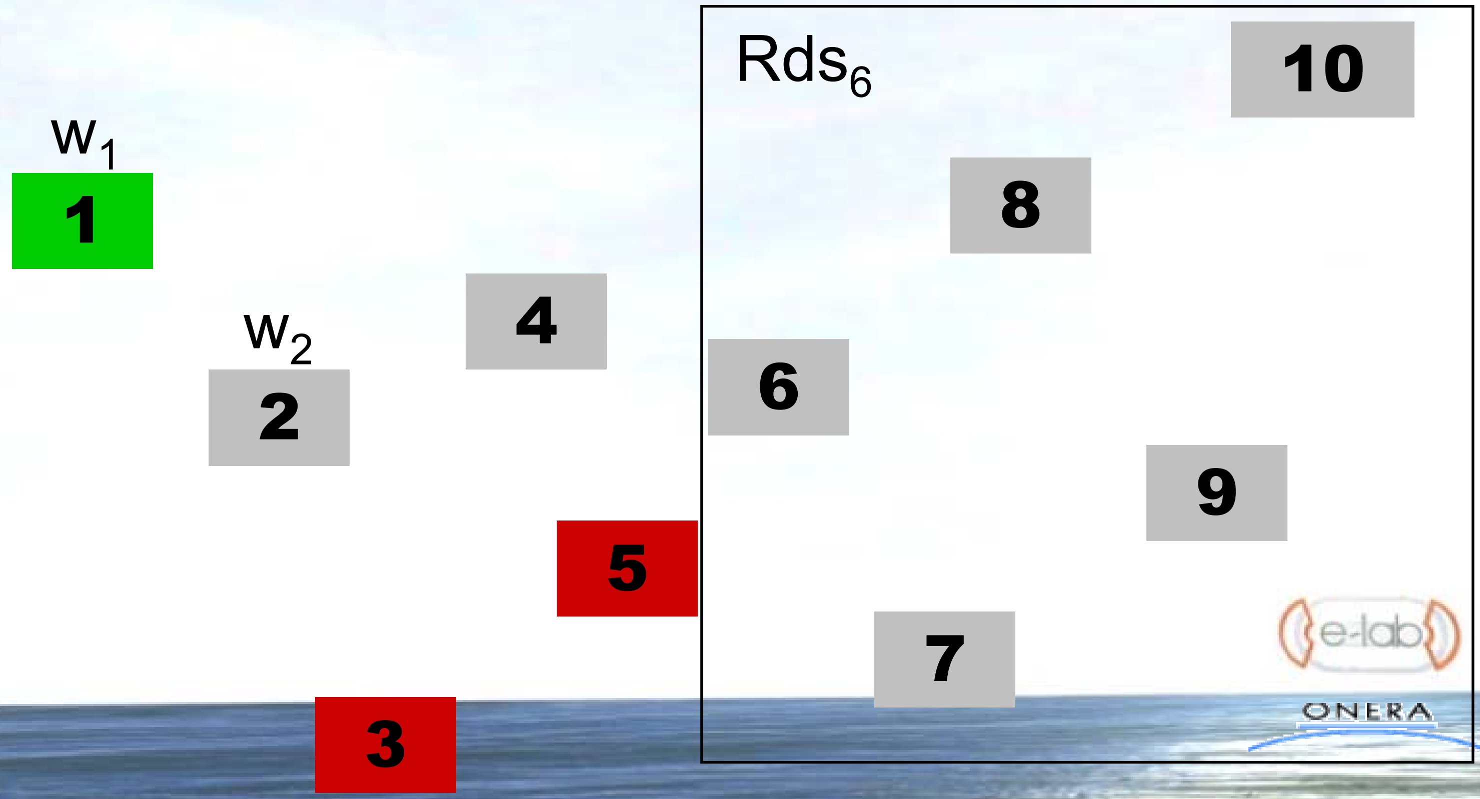
RDS upper bound



RDS cost-based filtering

Variable fixing rule:

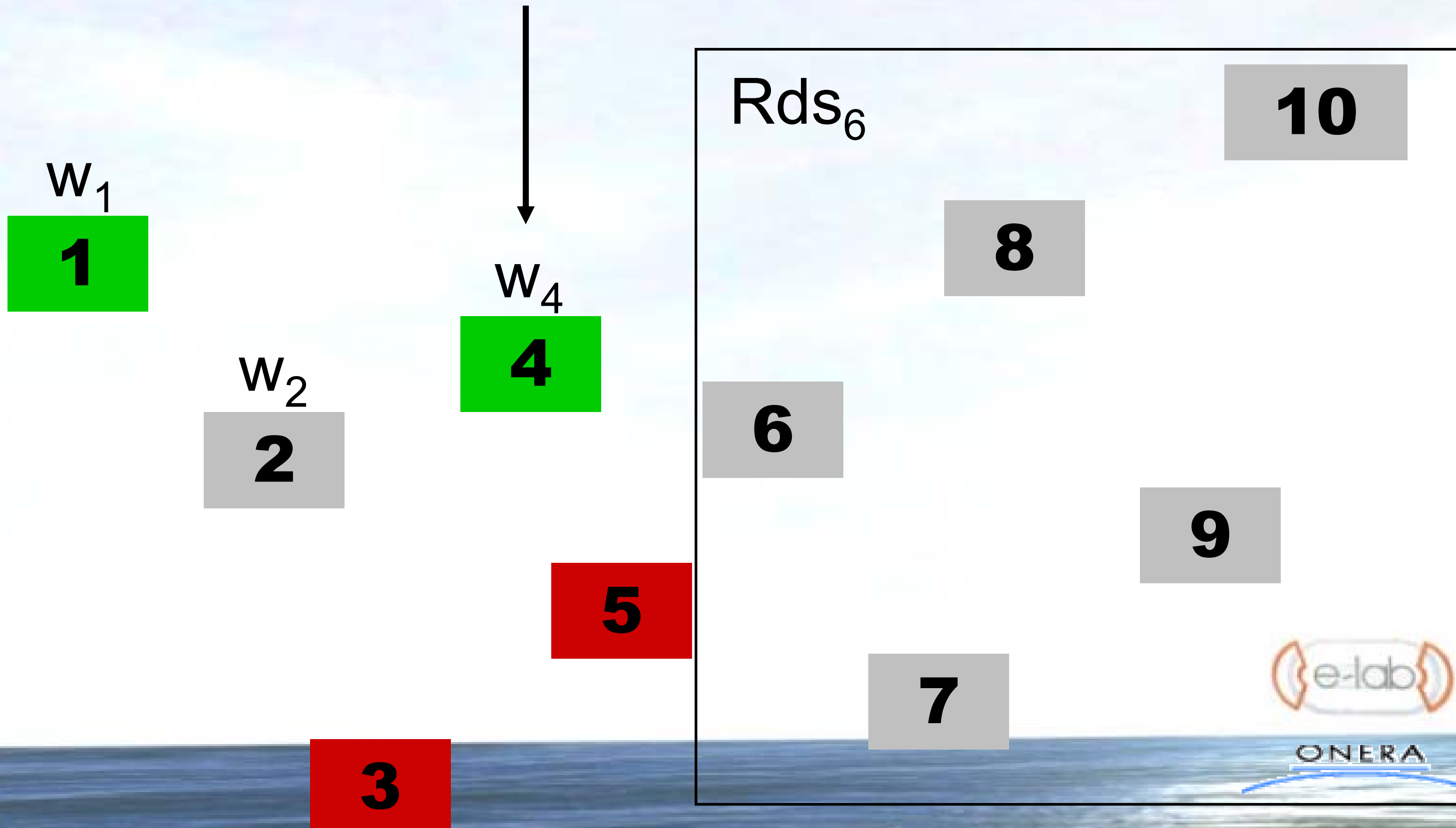
If w1+ w2+ Rds6 $\leq$ currentBest **then** X_4=1



RDS cost-based filtering

Variable fixing rule:

If w1+ w2+ Rds6 $\leq$ currentBest **then** X_4=1



LightRDS

- **Objective function:**
  - $\Omega = w_1X_1 + \ldots + w_5X_5 + \Omega_6$
  - $\Omega_6 = w_6X_6 + \ldots + w_{10}X_{10}$ with $\Omega_6 \leq Rds_6$
- **RDS filtering is naturally performed by these linear constraints**

Declarative implementation →

$$\Omega = \Omega_1$$

$$\Omega_i = w_iX_i + \Omega_{i+1}$$

$$\Omega_{10} = w_{10}X_{10}$$

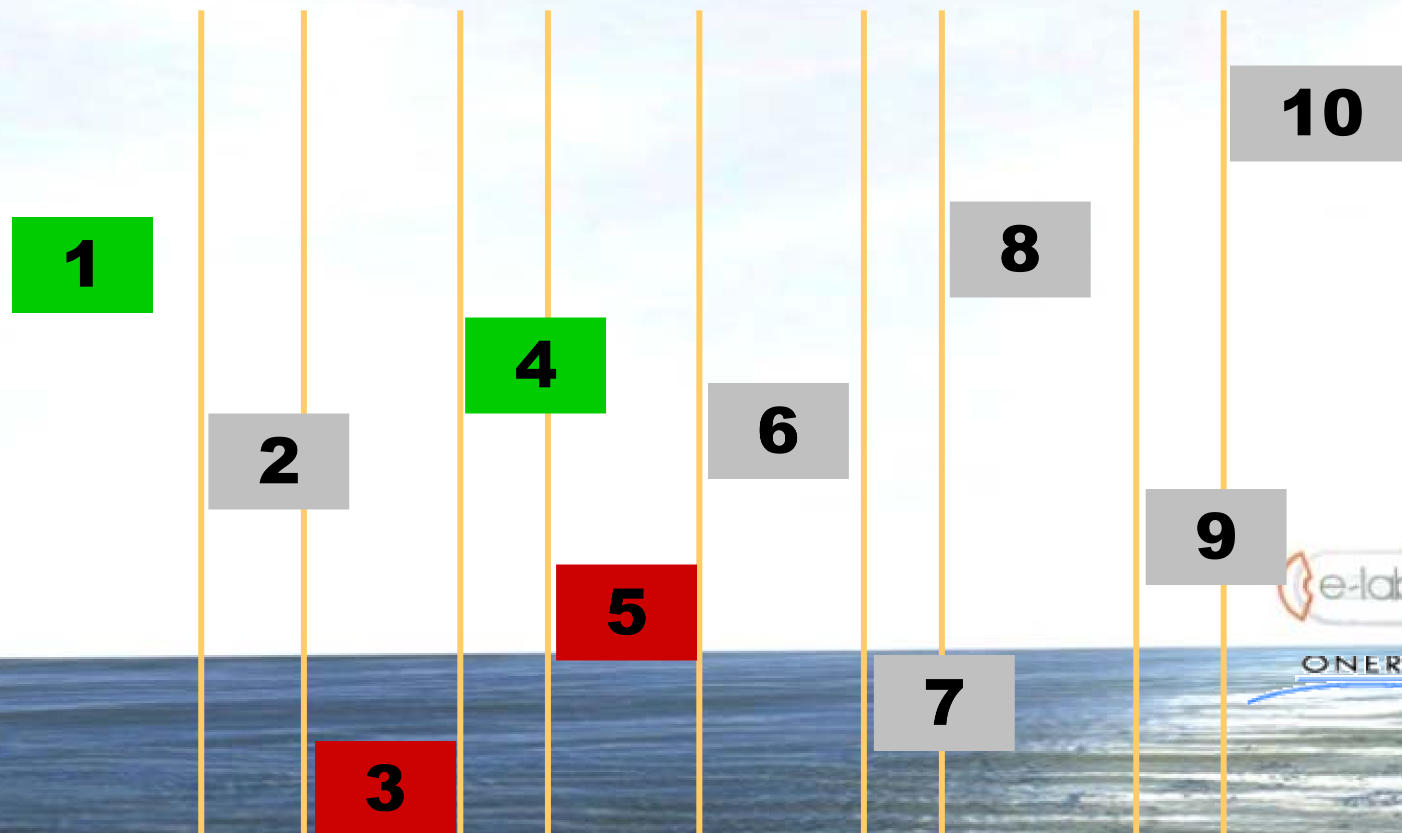
e-lab ONERA

Adding *10* variables and *10* ternary constraints

LightRDS

LightRDS filtering is strictly stronger

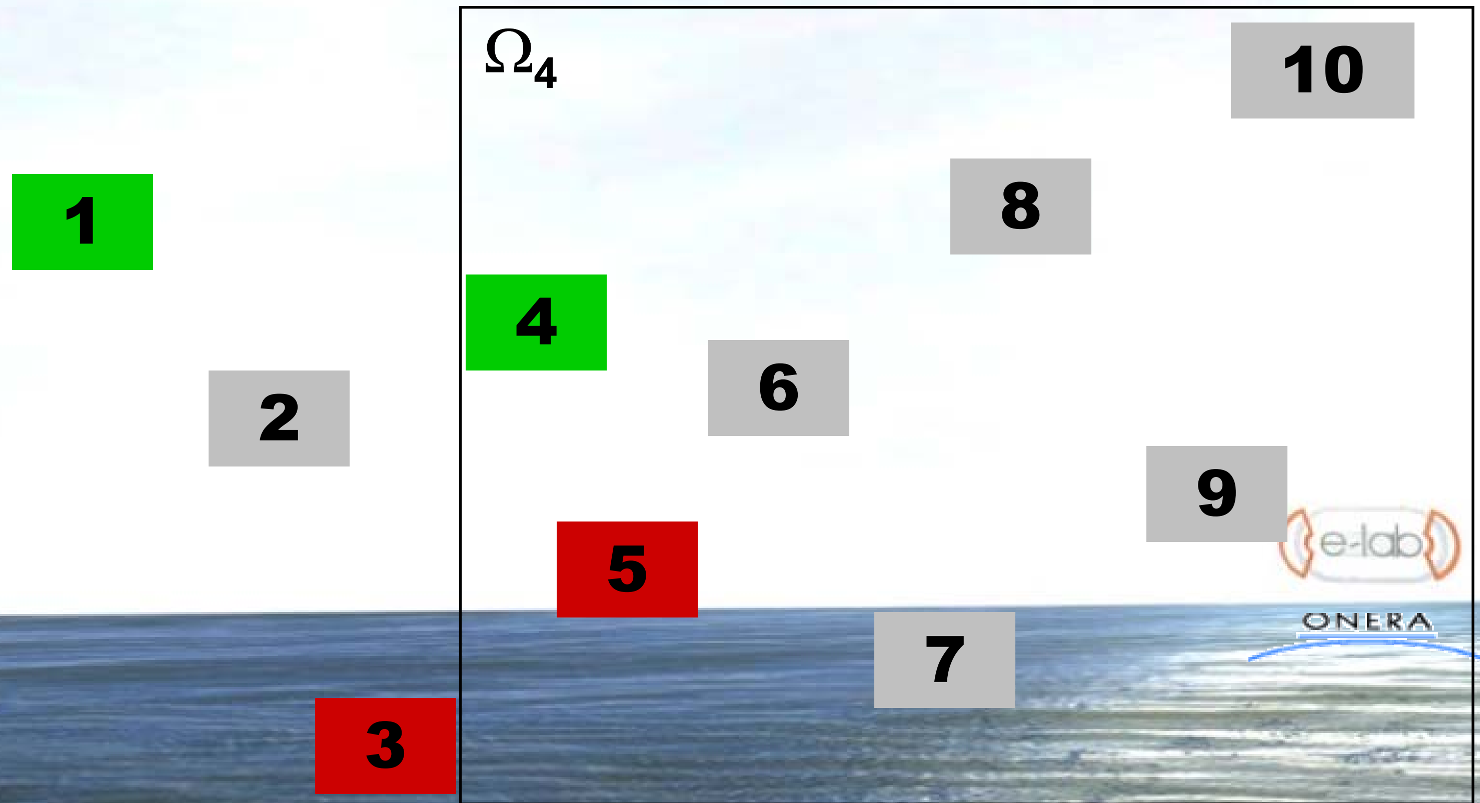
This is equivalent to using **all frontiers** simultaneously



LightRDS filtering is strictly stronger

If $Rds_4 < Rds_6 + w_4$ then frontier 4 would produce a better bound:

$$w_1 + w_2 + Rds_4 < w_1 + w_2 + w_4 + Rds_6$$

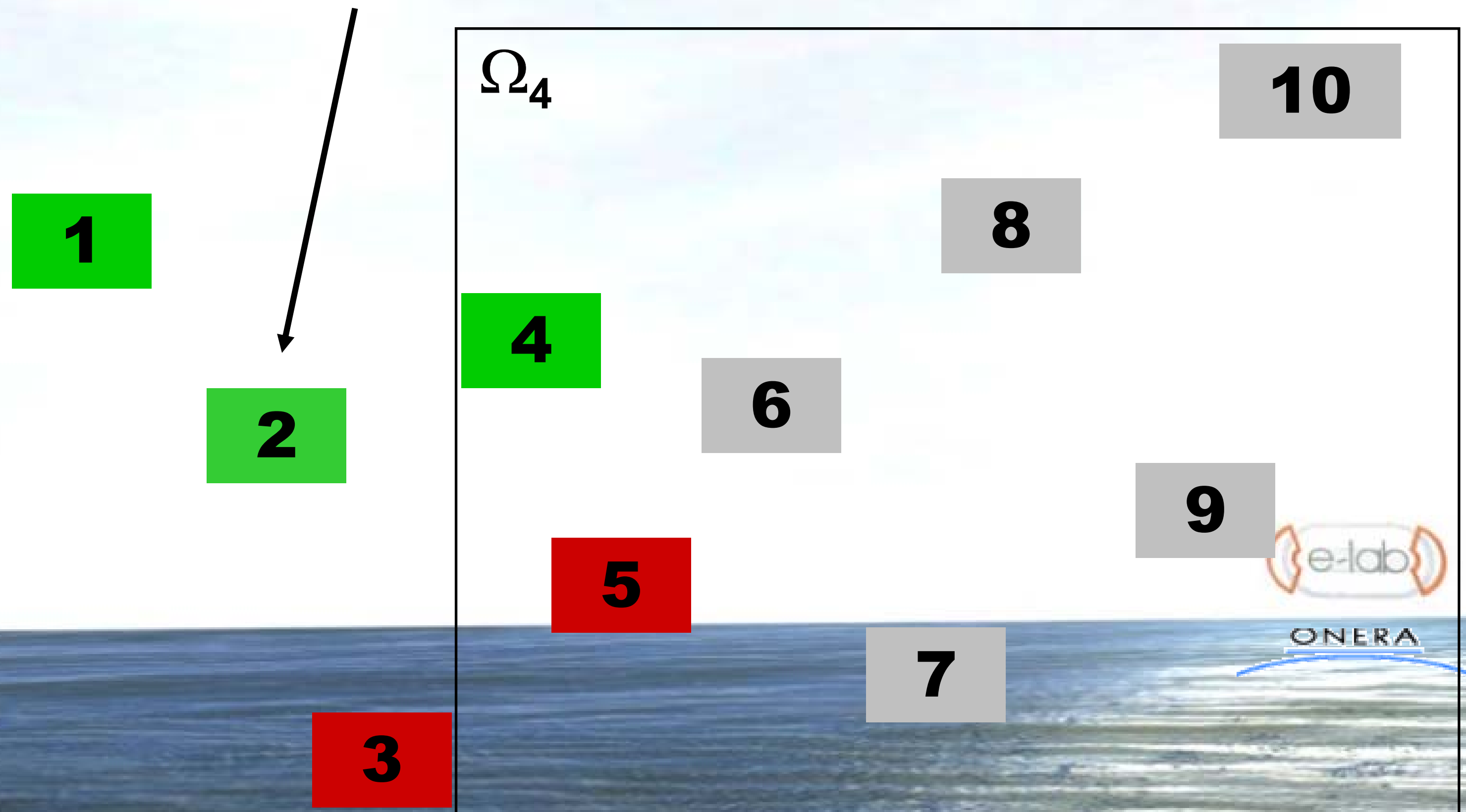


LightRDS filtering is strictly stronger

If $Rds_4 < Rds_6 + w_4$ then frontier 4 would produce a better bound:

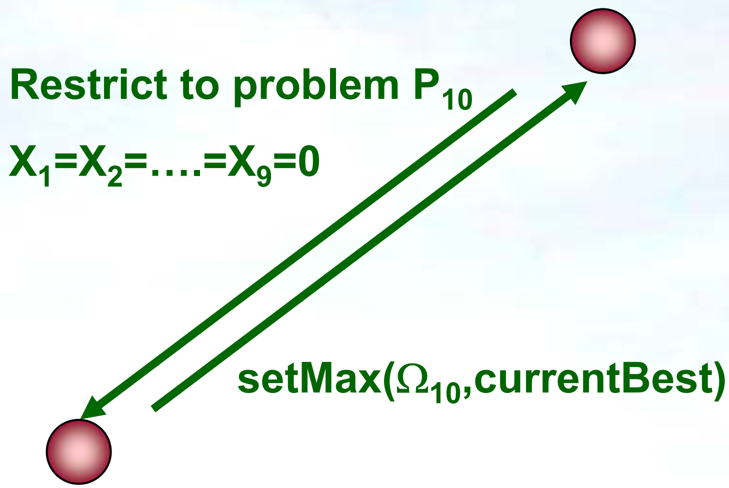
$$w_1 + w_2 + Rds_4 < w_1 + w_2 + w_4 + Rds_6$$

And possibly better filtering:



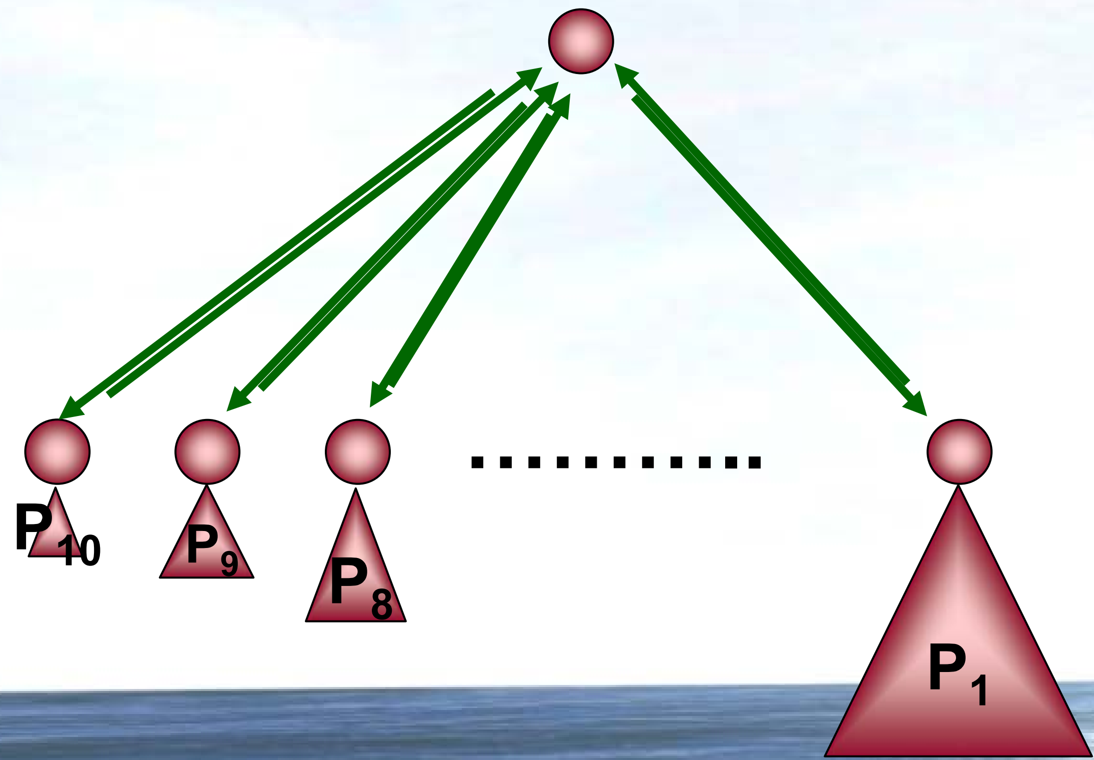
Control mechanism

→ Encapsulated in a special root choice point



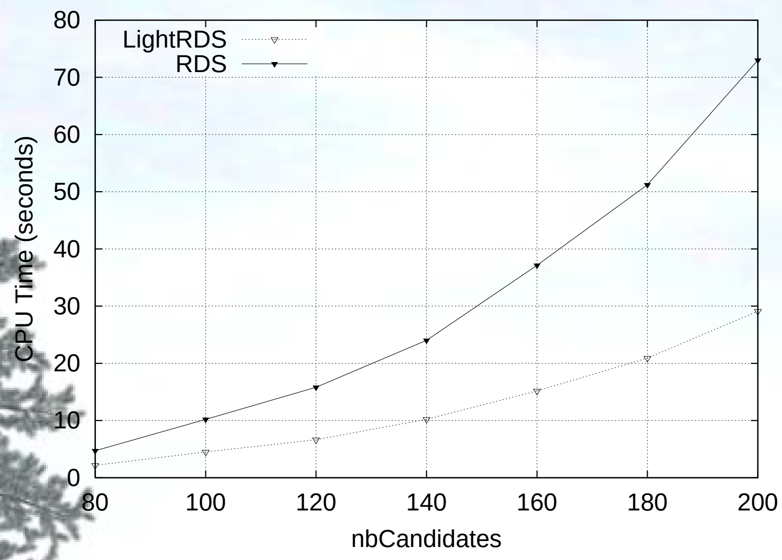
Control mechanism

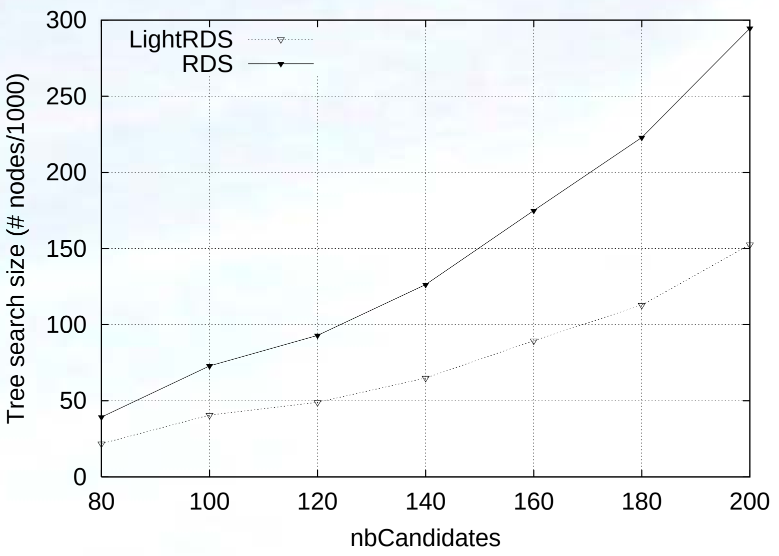
→ Encapsulated in a special root choice point



Computational Results

- **Satellite planning problem: ftp://ftp.cert.fr/pub/DCSD/CD/lemaitre/Choco/bep/**
- **CHOCO model**





Number of nodes and CPU time are divided by two

NOT extensible to WCSP

- **Forward Checking (FC)**

$$\sum w_c X_c = \sum_{c \in F^*} w_c X_c + \sum_{c \in F_1} w_c X_c + \sum_{c \in F_2} w_c X_c + \ldots + \sum_{c \in F_0} w_c X_c$$

- $c \in F^*$: Instantiated constraints
- $c \in F_1$: Constraints whose only non instantiated variable is V_1 — ***Bounded by*** $\min_{domain(V_1)} \sum_{c \in F_1} w_c X_c$
- $c \in F_2$: Constraints whose only non instantiated variable is V_2
- $c \in F_0$: Remaining constraints — ***Bounded by*** RDS bound of the largest subproblem $\subseteq F_0$

dynamic **partition of constraints**

→ *static* reformulation of the objective function seems impossible

e-lab ONERA

Conclusion

- **LightRDS is more efficient**
- **LightRDS is simple:**
  - Declarative implementation
  - No dedicated filtering to program
  - No frontier to manage
- **LightRDS can be tested in a few minutes**
  - on ***Variable*** *Weighted CSPs*
  - when the constraint graph has a small bandwidth