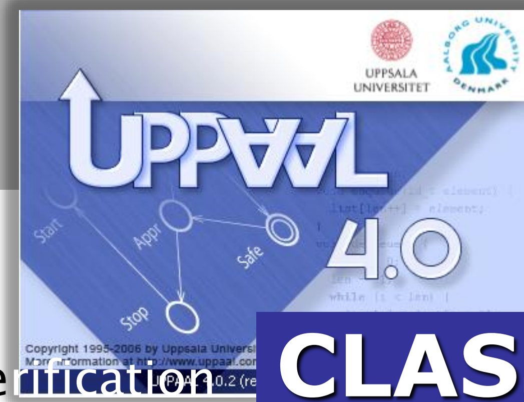
Planning
via
Model Checking

AALBORG UNIVERSITET

“Plan”

- Timed Automata
  - Model Checking
  - Time-optimal Planning
  - Zones
- Priced Timed Automata
  - Cost-optimal Planning
  - Priced Zones A*
- Timed Games
  - Dynamic Planning
  - Strategies, Zones
- Stochastic Timed Automata
  - Performance Analysis
  - Statistical Model Checking
- Stochastic Priced Timed Games
  - Expected Cost Optimal Adaptive Planning
  - Strategies
  - Reinforcement Learning
- Applications



Verification **CLASSIC** 1995

Optimization **CORA** 2001

Synthesis **TIGA** 2005

Component **ECDAR** 2010

Testing **TRON** 2004

Performance Analysis **SMC** 2011

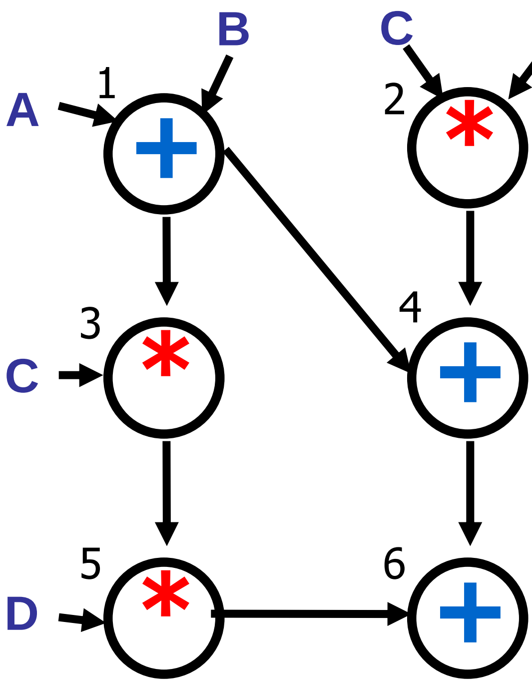
STRATEGO 2014

Priced Timed Automata

Cost-Optimal Planning

AALBORG UNIVERSITET

Task Graph Scheduling – Revisited



Compute :

(D * (C * (A + B)) + ((A + B) + (C * D))

using 2 processors

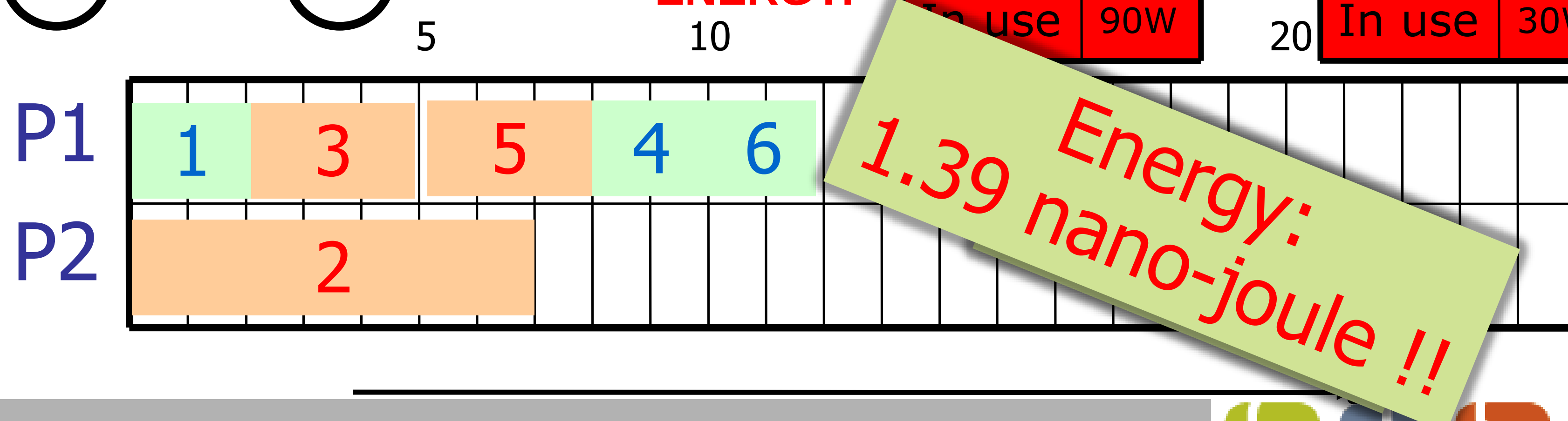
P1 (fast)

+	2ps
*	3ps
Idle	10W
In use	90W

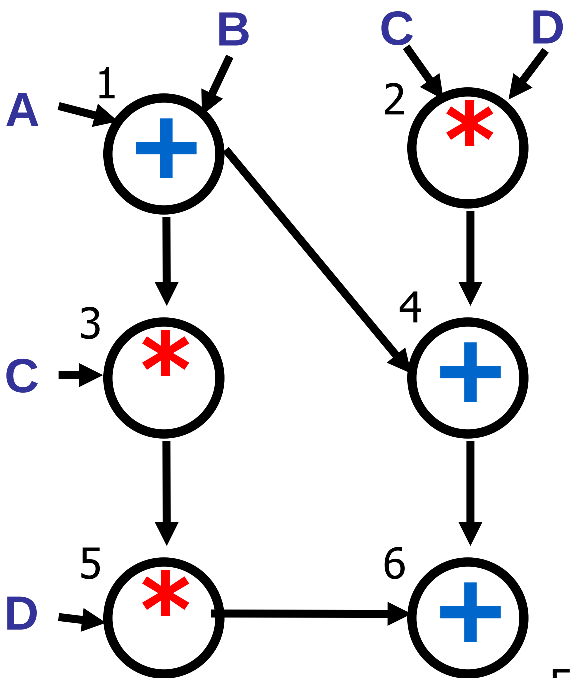
P2 (slow)

+	5ps
*	7ps
Idle	20W
In use	30W

ENERGY:



Task Graph Scheduling – Revisited

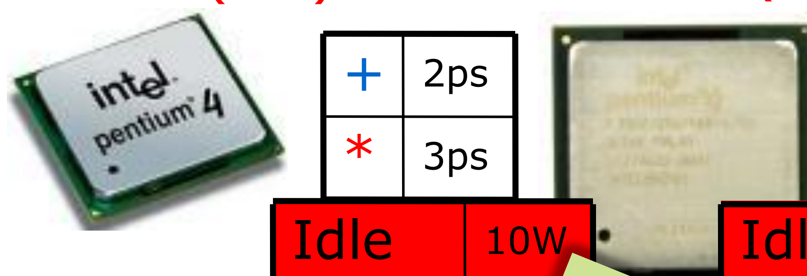


Compute :

(D * (C * (A + B)) + ((A + B) + (C * D))

using 2 processors

P1 (fast) P2 (slow)



P1:

+	2ps
*	3ps

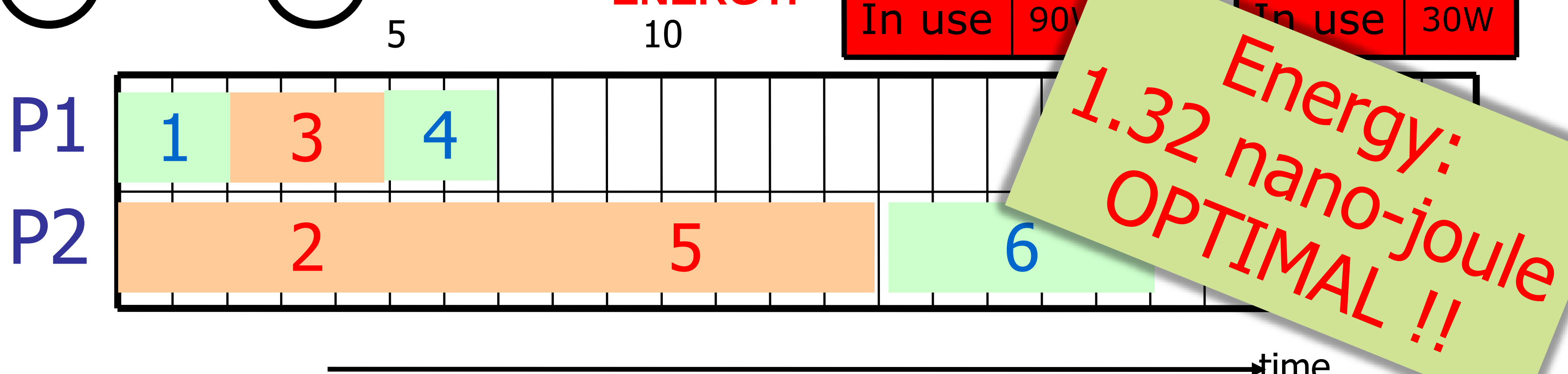
Idle	10W
In use	90W

P2:

+	5ps
*	7ps

Idle	20W
In use	30W

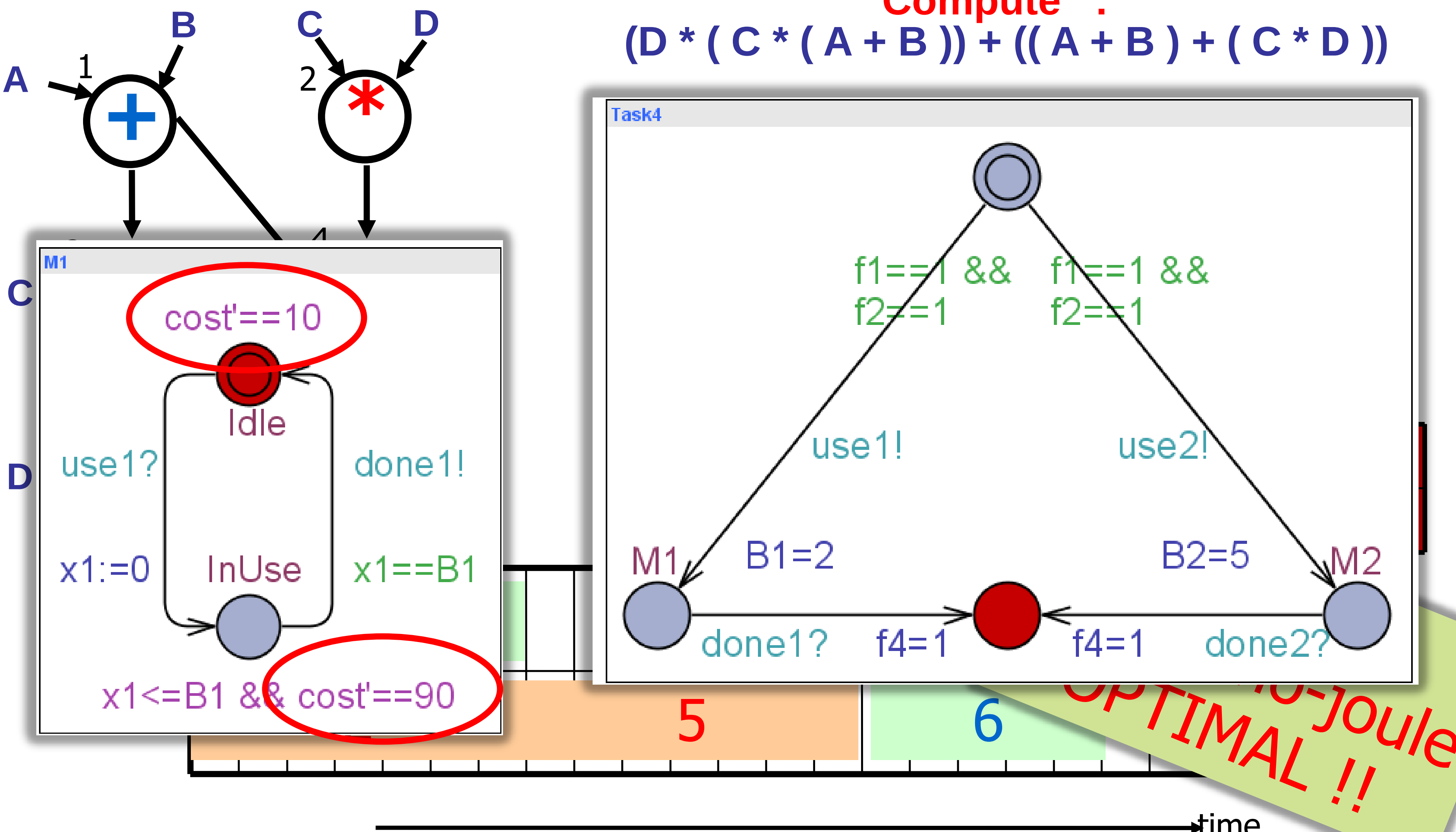
ENERGY:



Energy:
1.32 nano-joule
OPTIMAL !!

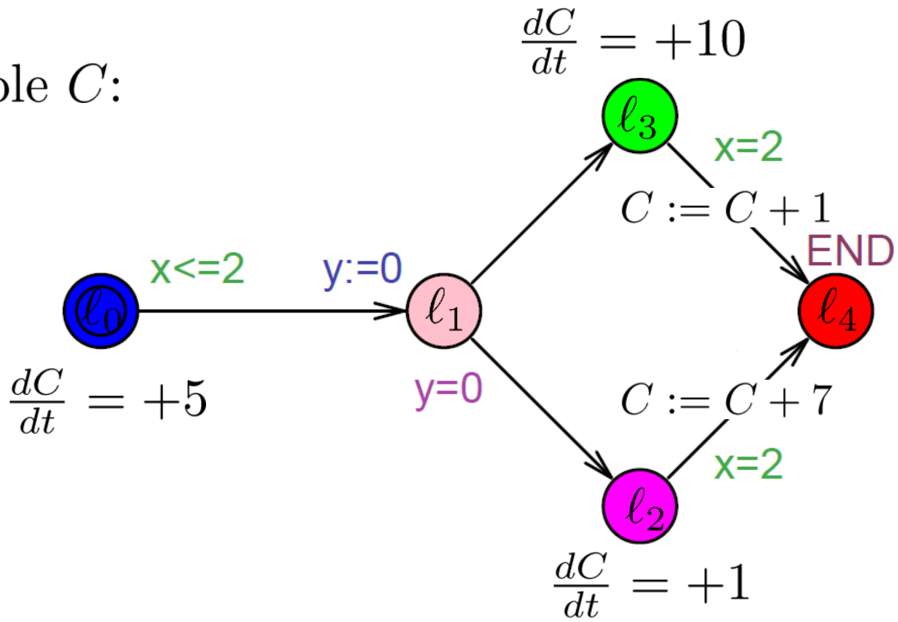
Task Graph Scheduling – Revisited

Compute :
(D * (C * (A + B)) + ((A + B) + (C * D))



A simple example

Observer variable C:



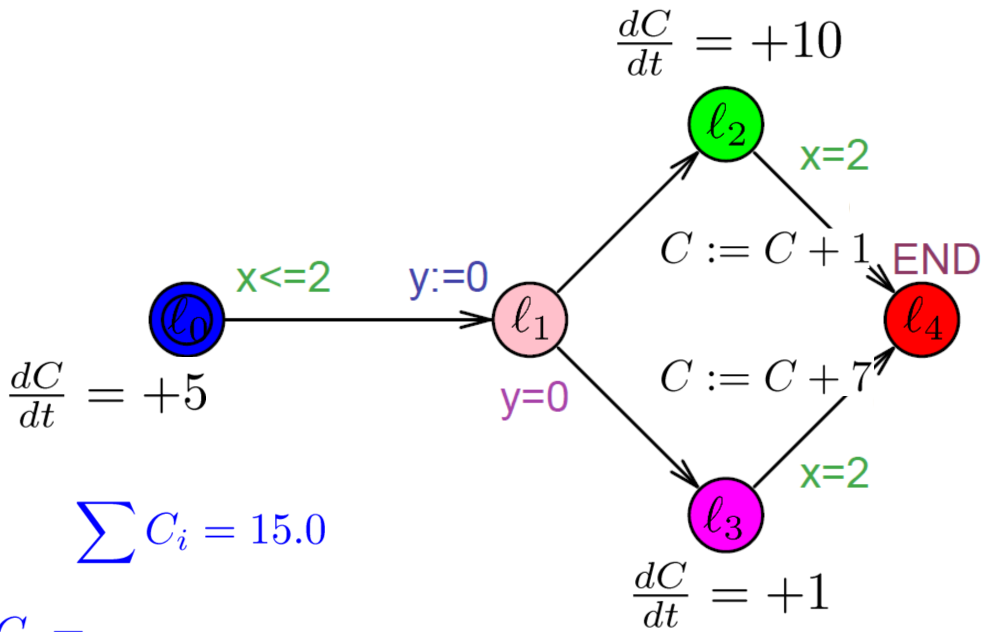
$$(\ell_0, [0,0]) \xrightarrow{1.9}_{9.5} (\ell_0, [1.9, 1.9]) \rightarrow_0 (\ell_1, [1.9, 0]) \rightarrow_0 (\ell_2, [1.9, 0]) \xrightarrow{0.1}_{0.1} (\ell_2, [2, 0.1]) \rightarrow_7 (\ell_4, [2, 0.1])$$

$$\sum C_i = 16.6$$

$$(\ell_0, [0,0]) \xrightarrow{1.2}_{6.0} (\ell_0, [1.2, 1.2]) \rightarrow_0 (\ell_1, [1.2, 0]) \rightarrow_0 (\ell_3, [1.2, 0]) \xrightarrow{0.8}_{8.0} (\ell_3, [2, 0.8]) \rightarrow_1 (\ell_4, [2, 0.8])$$

$$\sum C_i = 15.0$$

A simple example



$\sum C_i = 16.6$ $\sum C_i = 15.0$

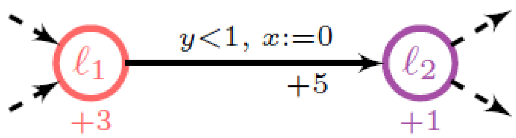
$\sum C_i = ..$

Q: What is cheapest cost for reaching ℓ_4 ?

$$\inf_{0 \leq t \leq 2} \min\{5t + 10(2-t) + 1, 5t + (2-t) + 4\} = 9$$

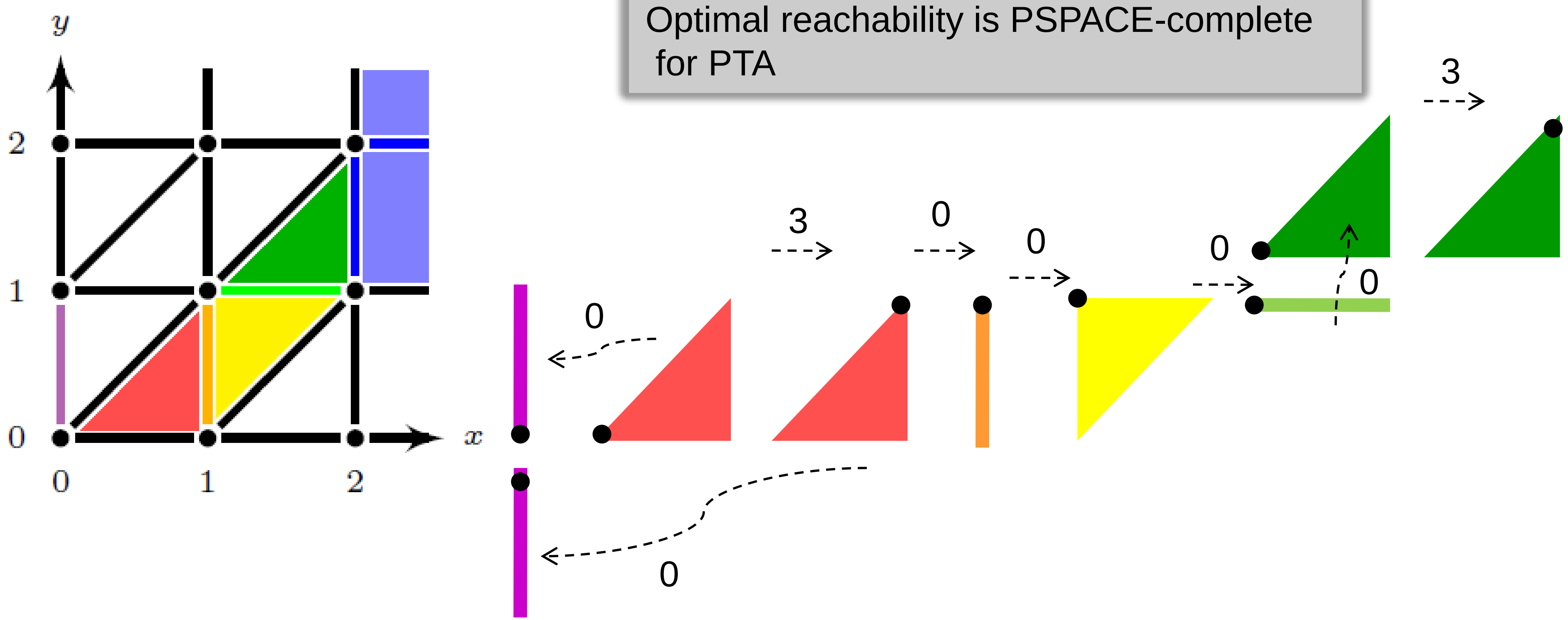
➜ **strategy:** leave immediately ℓ_0, go to ℓ_3, and wait there 2 t.u.

Corner Point Regions



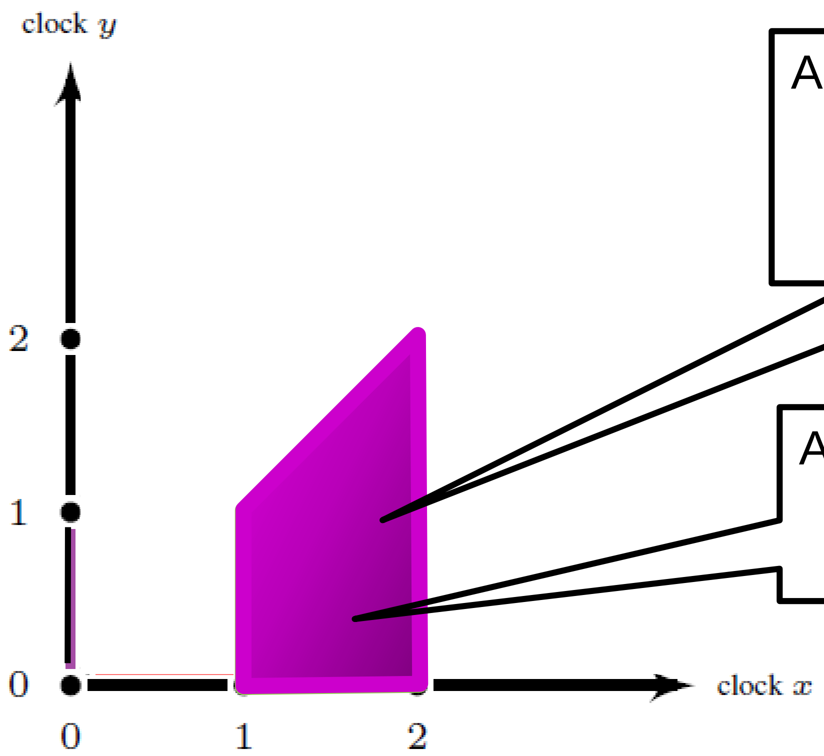
THM [Behrmann, Fehnker ..01] [Alur,Torre,Pappas 01]
Optimal reachability is decidable for PTA

THM [Bouyer, Brojaue, Briuere, Raskin 07]
Optimal reachability is PSPACE-complete for PTA



Priced Zones

[CAV01]



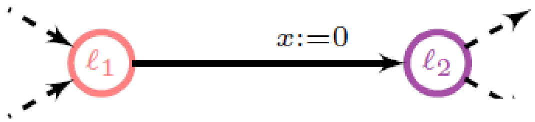
A zone **Z**:

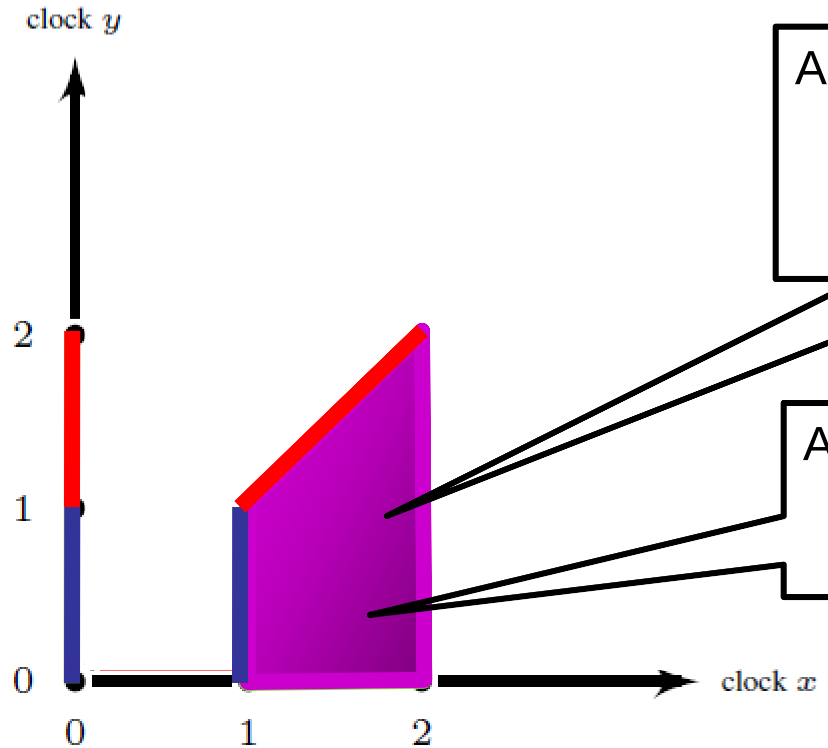
$1 \leq x \leq 2 \quad \wedge$
$0 \leq y \leq 2 \quad \wedge$
$x - y \geq 0$

A cost function **C**

$C(x,y) =$
$2 \cdot x - 1 \cdot y + 3$

Priced Zones – Reset [CAV01]





Z[x=0]**:**
x=0 $\wedge$
$0 \leq y \leq 2$

C = 1·y + 3

C= -1·y + 5

A zone **Z**:
$1 \leq x \leq 2 \quad \wedge$
$0 \leq y \leq 2 \quad \wedge$
$x - y \geq 0$

A cost function **C**
C(x,y) =
2·x - 1·y + 3

Symbolic Branch & Bound Algorithm

Cost := ∞

Passed := $\emptyset$

Waiting := $\{(l_0, Z_0)\}$

while Waiting $\neq \emptyset$ **do**

  select (l, Z) from Waiting

  if $l = l_g$ and minCost(Z) < Cost **then**

    Cost := minCost(Z)

  if minCost(Z) + Rem$_{(l,Z)} \geq$ Co **the**

  if for all (l, Z') in Passed: $Z' \not\leq Z$ **then**

    add (l, Z) to Passed

    add all (l', Z') with $(l, Z) \rightarrow (l', Z')$

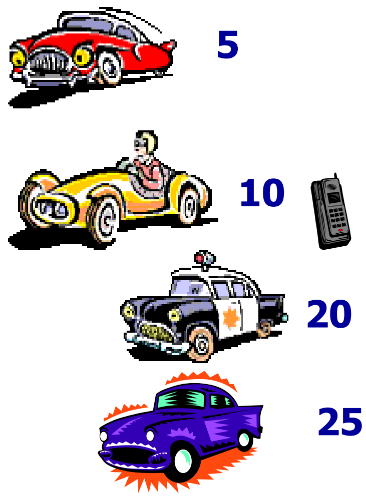
return Cost

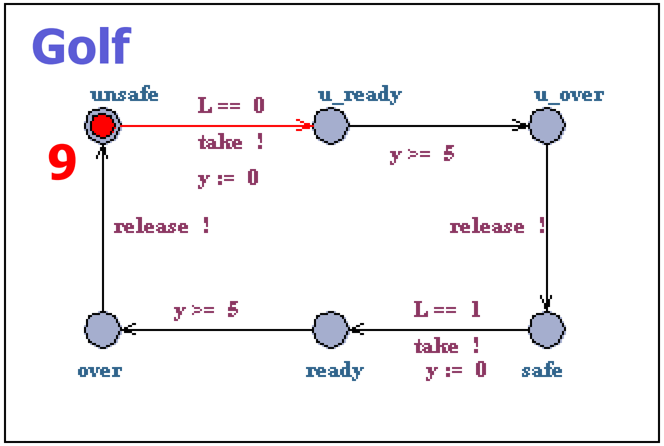
$Z' \leq Z$

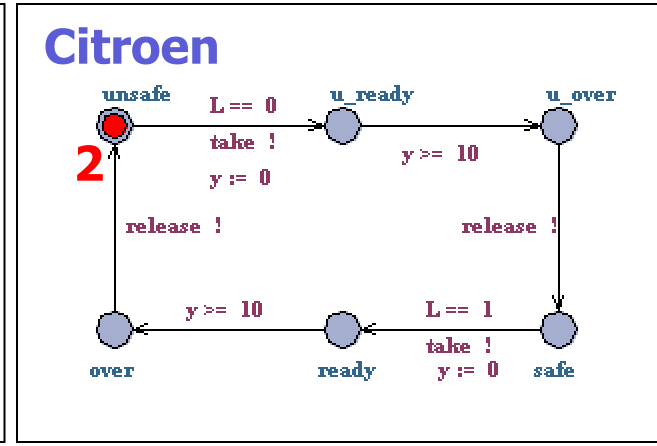
Z′ **is bigger & cheaper than** ***Z***

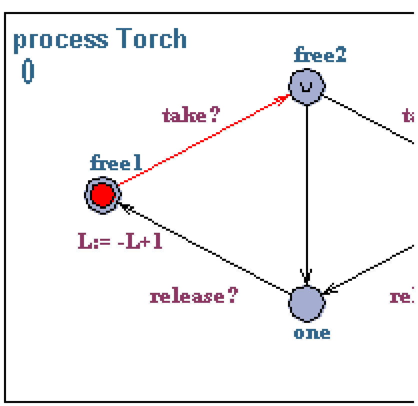
$\leq$ is a well-quasi ordering which guarantees termination!

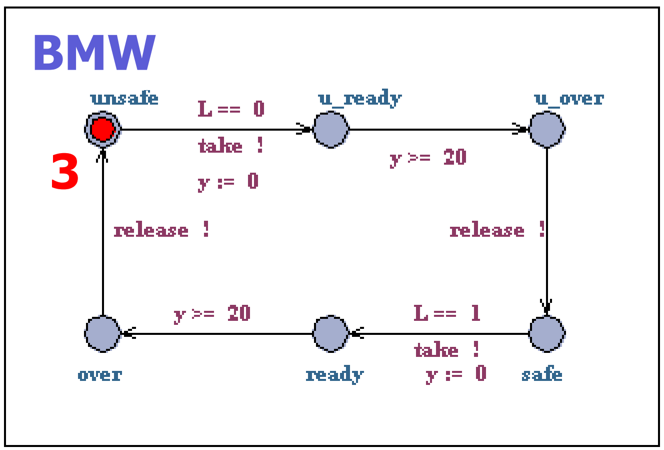
EXAMPLE: Optimal rescue plan for cars with different subscription rates for city driving !

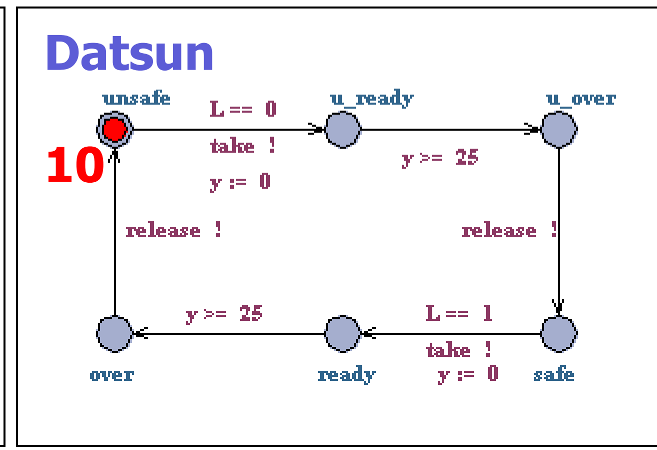










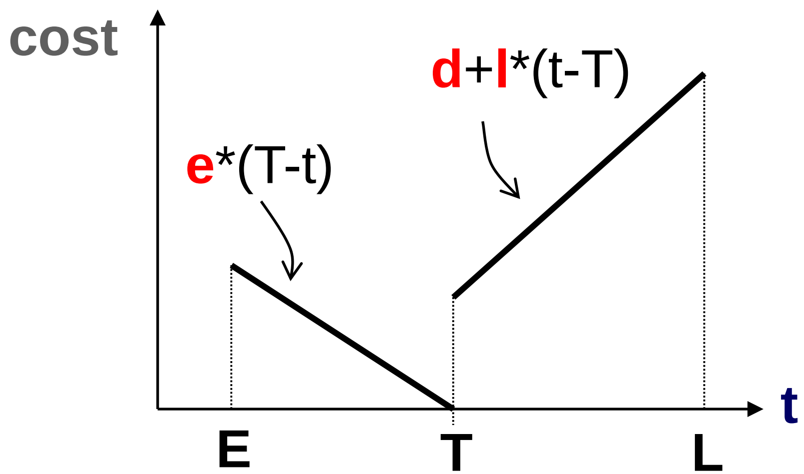


OPTIMAL PLAN HAS ACCUMULATED COST=195 and TOTAL TIME=65!

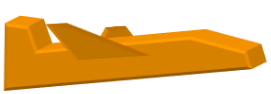
Experiments

COST-rates				SCHEDULE	COST	TIME	#Expl	#Pop'd
G	C	B	D					
Min Time				CG> G< BD> C< CG>		60	1762 1538	2638
1	1	1	1	CG> G< BG> G< GD>	55	65	252	378
9	2	3	10	GD> G< CG> G< BG>	195	65	149	233
1	2	3	4	CG> G< BD> C< CG>	140	60	232	350
1	2	3	10	CD> C< CB> C< CG>	170	65	263	408
1	20	30	40	BD> B< CB> C< CG>	975 1085	85 time<85	-	-
0	0	0	0	-	0	-	406	447

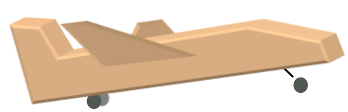
Example: Aircraft Landing

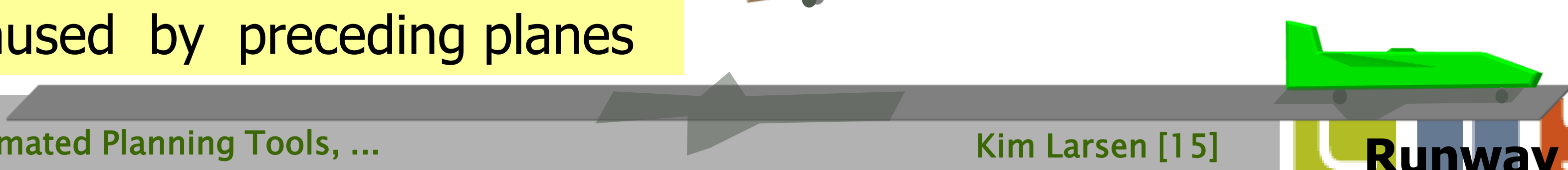


E earliest landing time
T target time
L latest time
e cost rate for being early
l cost rate for being late
d fixed cost for being late



Planes have to keep separation distance to avoid turbulences caused by preceding planes





Modelling ALP with MILP

[Beasley00]

minimize $\sum_{i=1}^{P} (e_i \, \alpha_i + d_i \, \beta_i)$

where

t_i: landing time of plane i

α_i : how early plane i lands before target T_i

β_i : how late plane i lands after target T_i

δ_{ij} : if i lands before j then 0 otherwise 1

Modelling ALP with MILP

[Beasley00]

$E_i \leq t_i \leq L_i$: Land only in the permitted slot

$\delta_{ij} + \delta_{ji} = 1$: Either or

$t_i \geq t_j + S_{ij} - M\delta_{ij}$: Respect the turbulence

$\alpha_i \geq T_i - t_i$: Bigger than the difference between T_i and t_i

$0 \leq \alpha_i \leq T_i - E_i$: The relevant interval

$\beta_i \geq t_i - T_i$: Bigger than the difference between t_i and T_i

$0 \leq \beta_i \leq L_i - T_i$: The relevant intervals

$t_i = T_i - \alpha_i + \beta_i$: Guaranteeing the right relationships

- Total formulation
  - $3P$ continuous variables
  - $2P(P-1) + PR$ binary variables
  - $4P + (4+R)P(P-1)/2$ constraints
- 2 runways, 30 planes $\Longrightarrow$ 1890 variables 2730 constraints
- Additional tightening constraints to help LP relaxation

minimize $\sum_{i=1}^{P} (e_i \alpha_i + d_i \beta_i)$

where

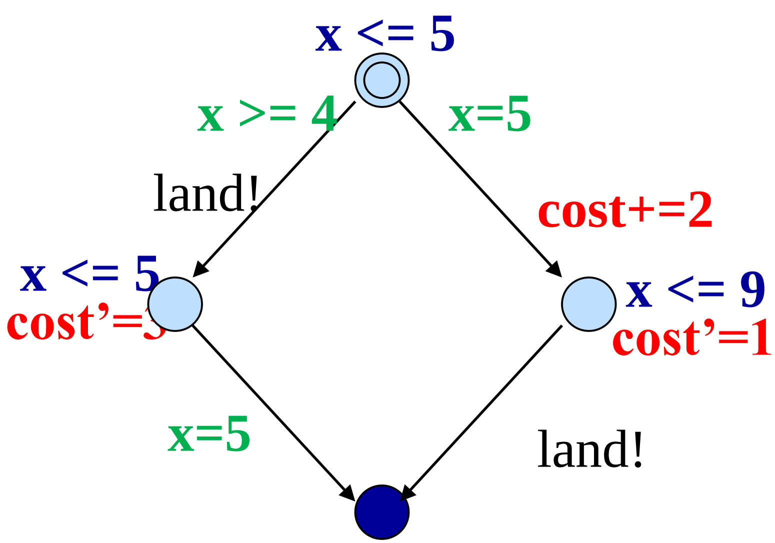
t_i: landing time of plane i

α_i : how early plane i lands before target T_i

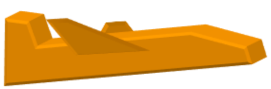
β_i : how late plane i lands after target T_i

δ_{ij} : if i lands before then 0 otherwise 1

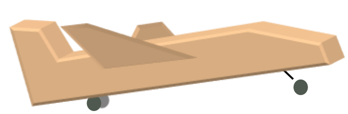
Aircraft Landing – PTA

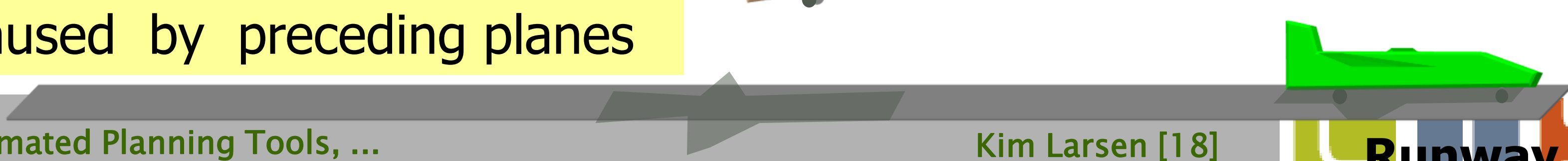


4 earliest landing time
5 target time
9 latest time
3 cost rate for being early
1 cost rate for being late
2 fixed cost for being late



Planes have to keep separation distance to avoid turbulences caused by preceding planes





Aircraft Landing

Source of examples:
Baesley et al'2000

	problem instance	1	2	3	4	5	6	7
	number of planes	10	15	20	20	20	30	44
	number of types	2	2	2	2	2	4	2
1	optimal value	700	1480	820	2520	3100	24442	1550
	explored states	481	2149	920	5693	15069	122	662
	cputime (secs)	4.19	25.30	11.05	87.67	220.22	0.60	4.27
2	optimal value	90	210	60	640	650	554	0
	explored states	1218	1797	669	28821	47993	9035	92
	cputime (secs)	17.87	39.92	11.02	755.84	1085.08	123.72	1.06
3	optimal value	0	0	0	130	170	0	N/A
	explored states	24	46	84	207715	189602	62	
	cputime (secs)	0.36	0.70	1.71	14786.19	12461.47	0.68	
4	optimal value	N/A	N/A	N/A	0	0	N/A	N/A
	explored states				65	64		
	cputime (secs)				1.97	1.53		

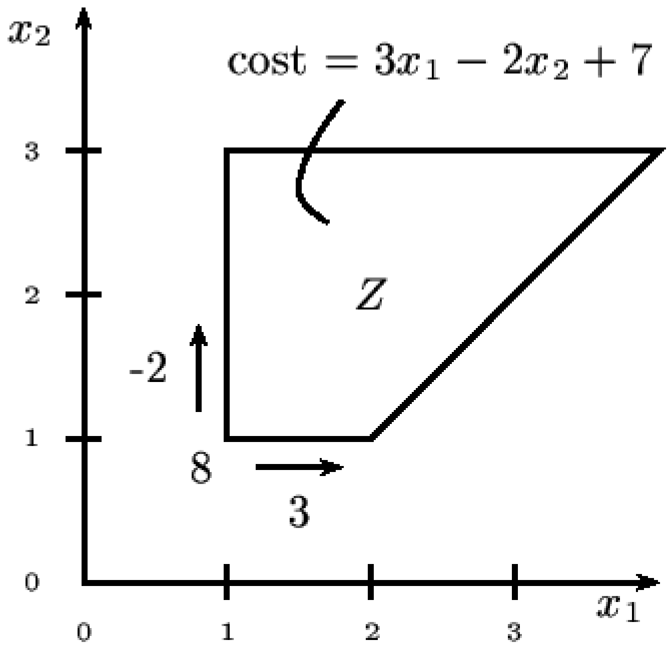
Symbolic Branch & Bound Algorithm

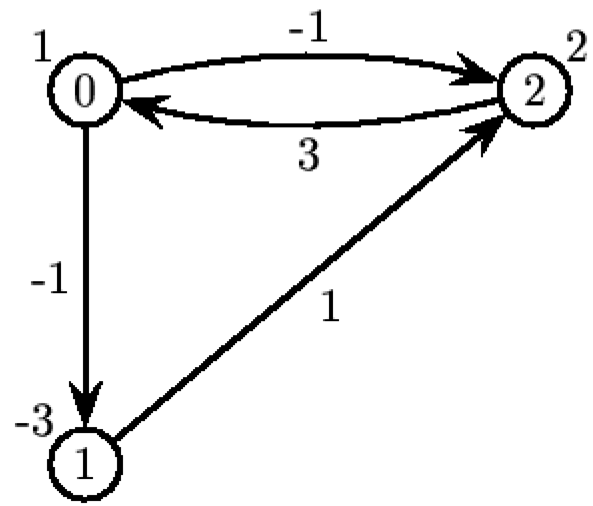
Cost := ∞
Passed := $\emptyset$
Waiting := $\{(l_0, Z_0)\}$
while Waiting $\neq \emptyset$ **do**
  select (l, Z) from Waiting
  if $l = l_g$ and minCost(Z) < Cost **then**
    Cost := minCost(Z)
  if minCost(Z) + Rem$_{(l,Z)} \geq$ Cost **then** break
  if for all (l, Z') in Passed: $Z' \not\subseteq Z$ **then**
    add (l, Z) to Passed
    add all (l', Z') with $(l, Z) \rightarrow (l', Z')$ to Waiting
return Cost

Zone based
Linear Programming Problems
→(dualize)
Min Cost Flow

Zone LP → Min Cost Flow

Exploiting duality



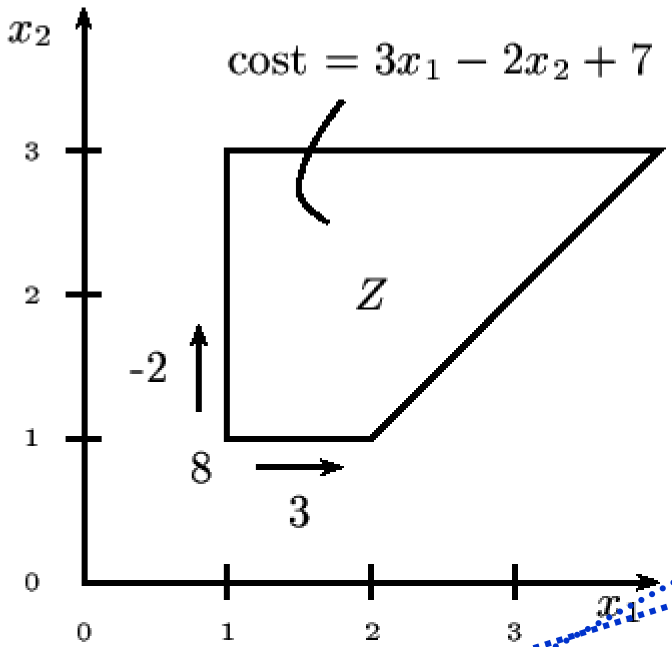


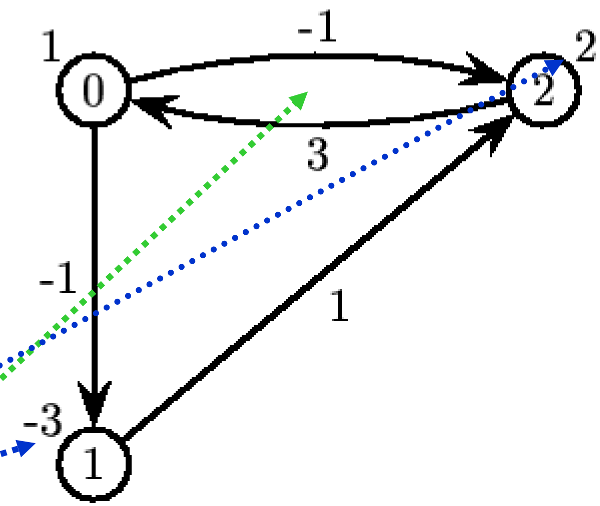
minimize $3x_1-2x_2+7$
when $x_1-x_2 \leq 1$
$1 \leq x_2 \leq 3$
$x_2 \geq 1$

minimize $3y_{2,0}-y_{0,2}+y_{1,2} - y_{0,1}$
when $y_{2,0}-y_{0,1}-y_{0,2}=1$
$y_{0,2}+y_{1,2}=2$
$y_{0,1}-y_{1,2}=-3$

Zone LP → Min Cost Flow

Exploiting duality





minimize $3x_1 - 2x_2 + 7$

when $x_1 - x_2 \leq 1$

$1 \leq x_2 \leq 3$

$x_2 \geq 1$

minimize $3y_{2,0} - y_{0,2} + y_{1,2} - y_{0,1}$

when $y_{2,0} - y_{0,1} - y_{0,2} = 1$

$y_{0,2} + y_{1,2} = 2$

$y_{0,1} - y_{1,2} = -3$

Aircraft Landing (revisited)

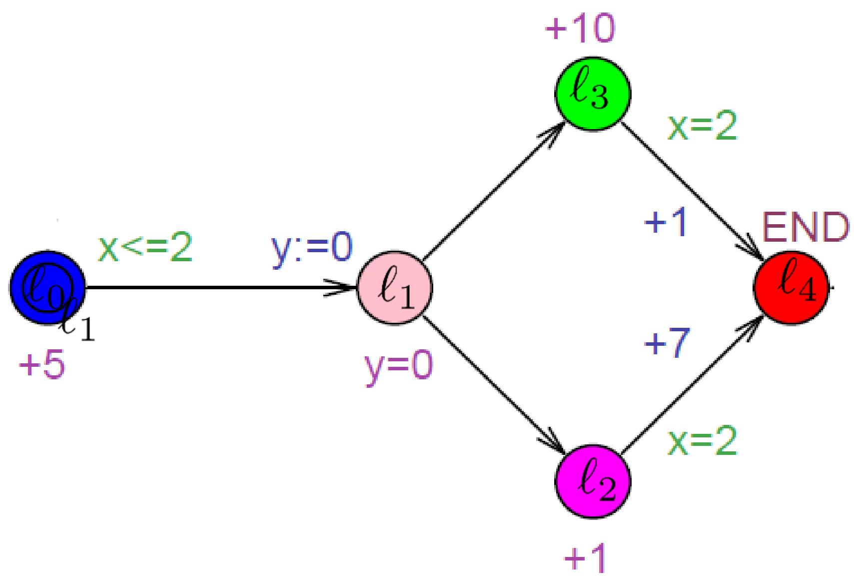
[TACAS04]

Using MCF/Netsimplex

RW	Planes	10	15	20	20	20	30	44
	Types	2	2	2	2	2	4	2
1	simplex	0.844s	5.210s	2.135s	17.888s	44.878s	0.451s	0.670s
	netsimplex	0.156s	0.657s	0.369s	2.363s	5.503s	0.127s	0.322s
factor		**5.41**	**7.93**	**5.79**	**7.57**	**8.16**	**3.55**	**2.08**
2	simplex	2.577s	7.436s	2.175s	94.357s	120.004s	2.322s	0.264s
	netsimplex	0.332s	1.036s	0.436s	13.376s	18.033s	0.600s	0.179s
factor		**8.00**	**7.18**	**4.99**	**7.054**	**6.65**	**3.87**	**1.474**
3	simplex	0.120s	0.181s	0.357s	740.100s	516.678s	0.166s	N/A
	netsimplex	0.064s	0.104s	0.129s	170.176s	124.805s	0.079s	N/A
factor		**1.87**	**1.74**	**2.77**	**4.34**	**4.14**	**2.10**	
4	simplex	N/A	N/A	N/A	1.603s	0.318s	N/A	N/A
	netsimplex	N/A	N/A	N/A	0.378s	0.093s	N/A	N/A
factor					**4.24**	**3.42**		

A. Loebel (2000). MCF Version 1.2 - A network simplex implementation. (http://www.zib.de)

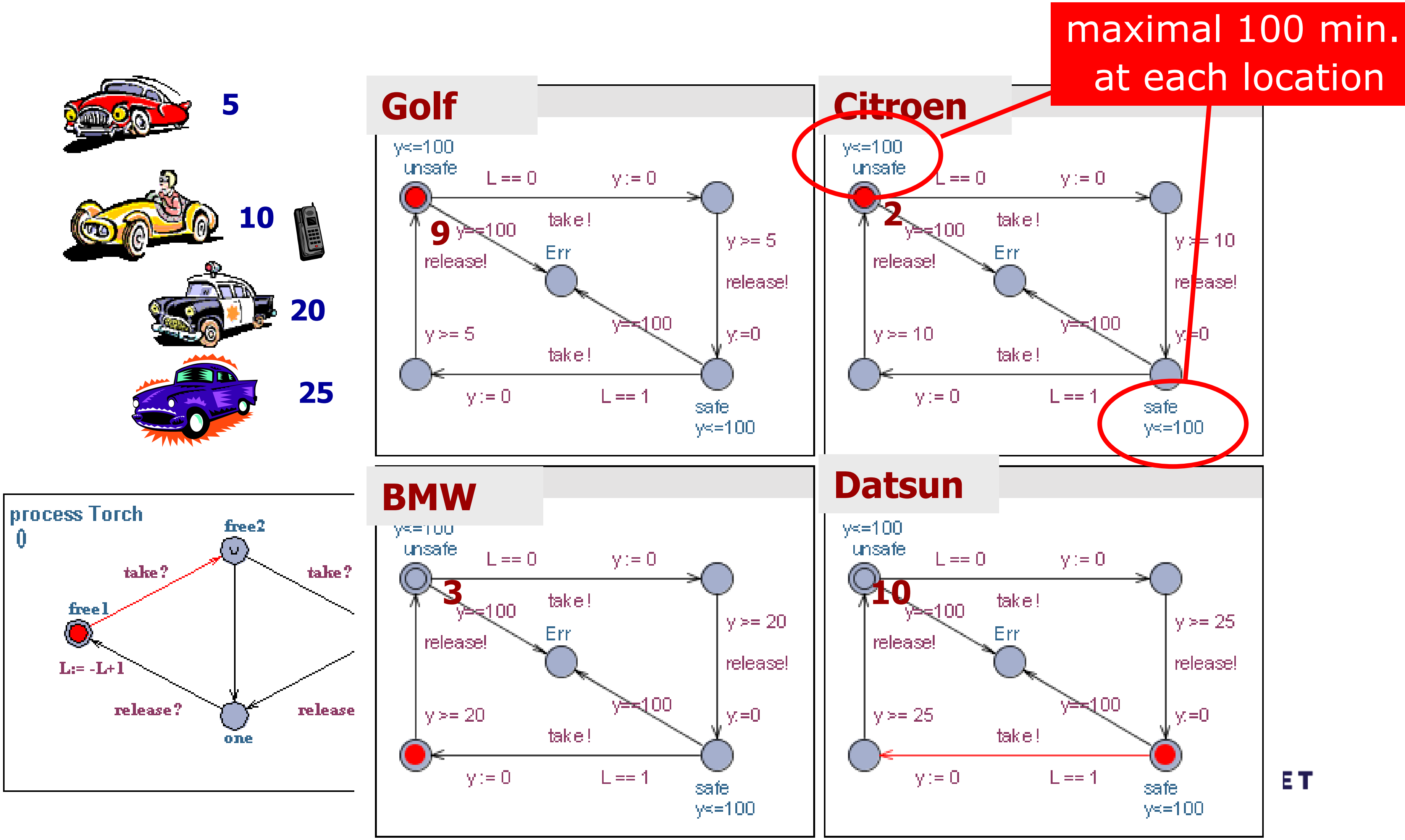
Optimal Schedule



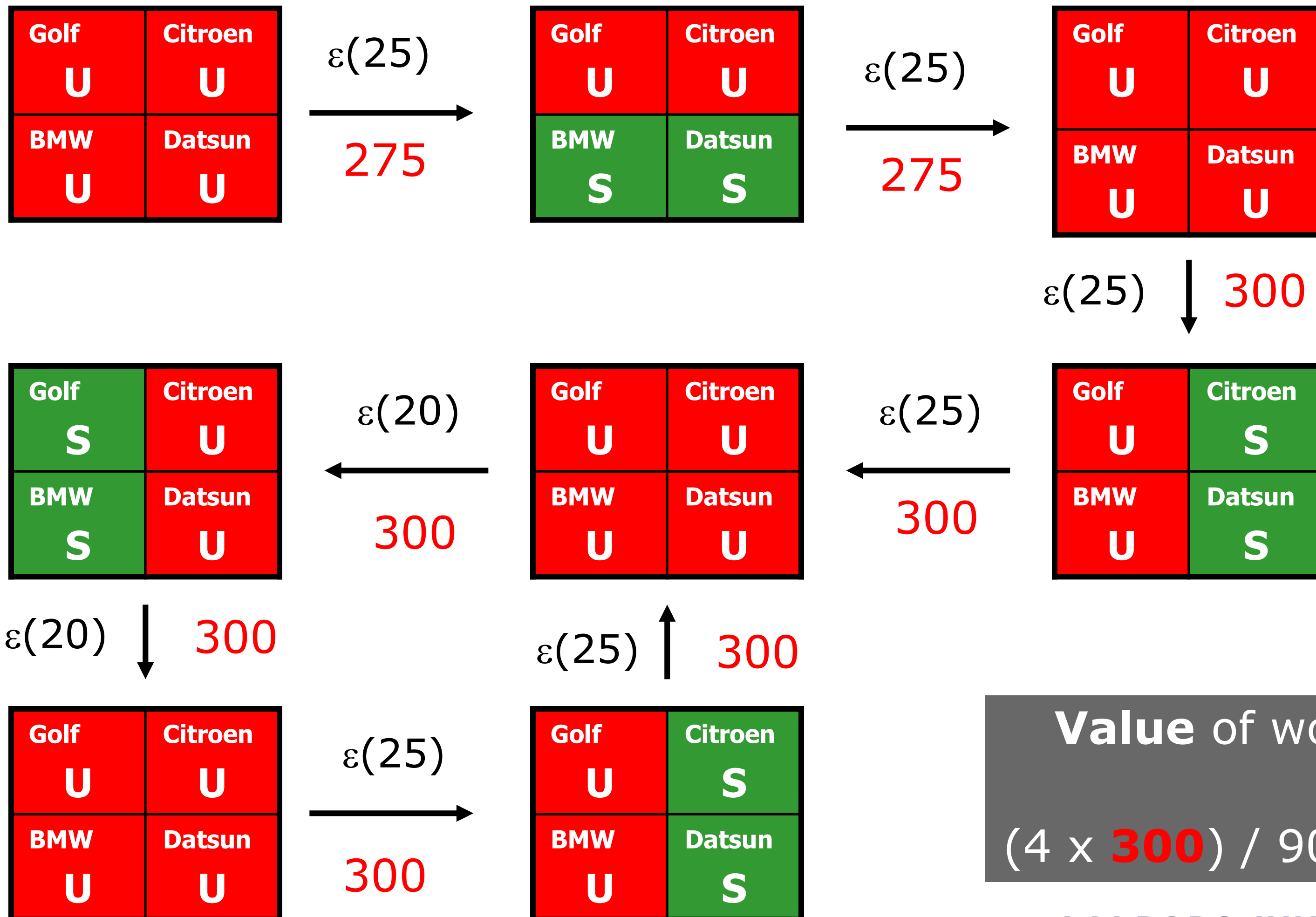
$$(\ell_0,[0,0]) \xrightarrow{1.2}_{6.0} (\ell_0,[1.2,1.2]) \rightarrow_{0} (\ell_1,[1.2,0]) \rightarrow_{0}$$
$$(\ell_3,[1.2,0]) \xrightarrow{0.8}_{8.0} (\ell_3,[2,0.8]) \rightarrow_{1} (\ell_4,[2,0.8])$$
$$\rightarrow_{2.0} (\ell_0,[0,0])$$

$$\boxed{\sum_i C_i / \sum_i t_i = 17/2 = 8.5}$$

EXAMPLE: Optimal WORK plan for cars with different subscription rates for city driving !



Workplan I



Value of workplan:

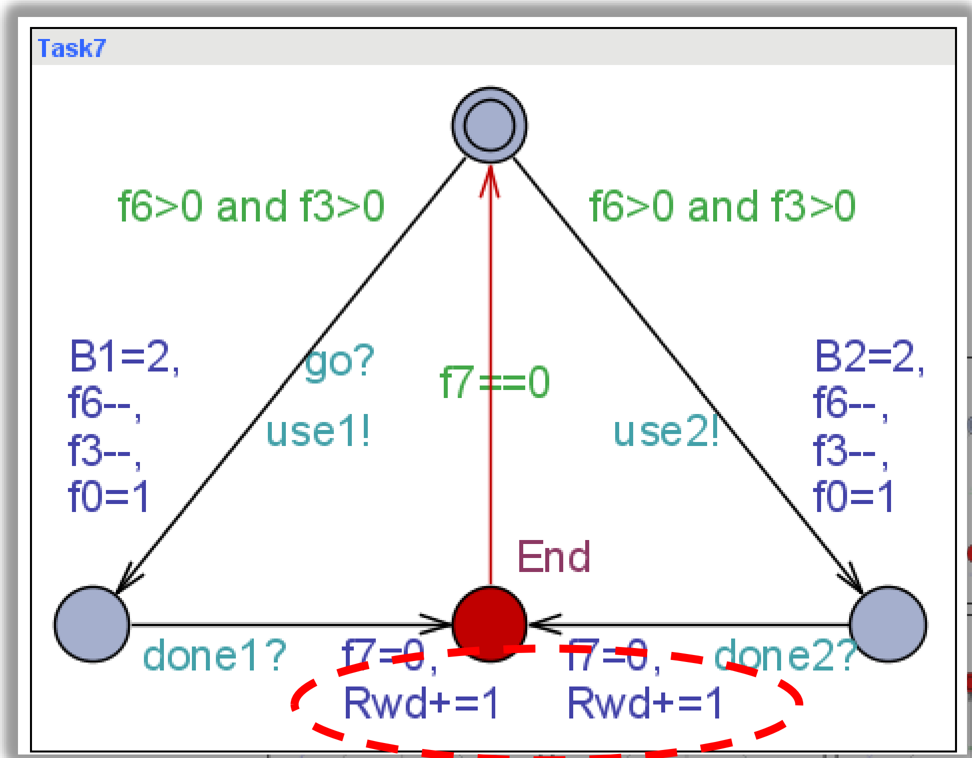
(4 x **300**) / 90 = **13.33**

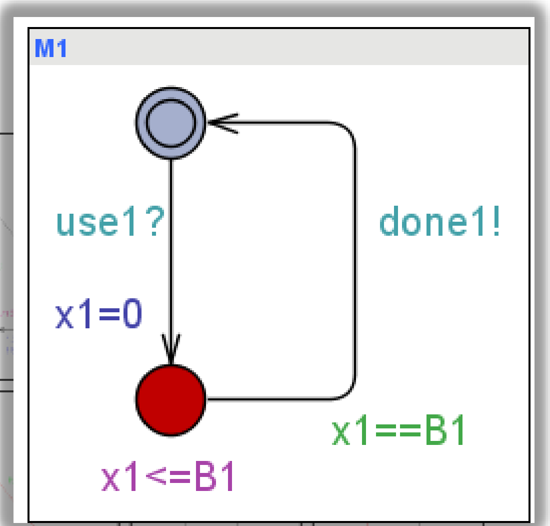
Workplan II

Value of workplan:

560 / 100 = **5.6**

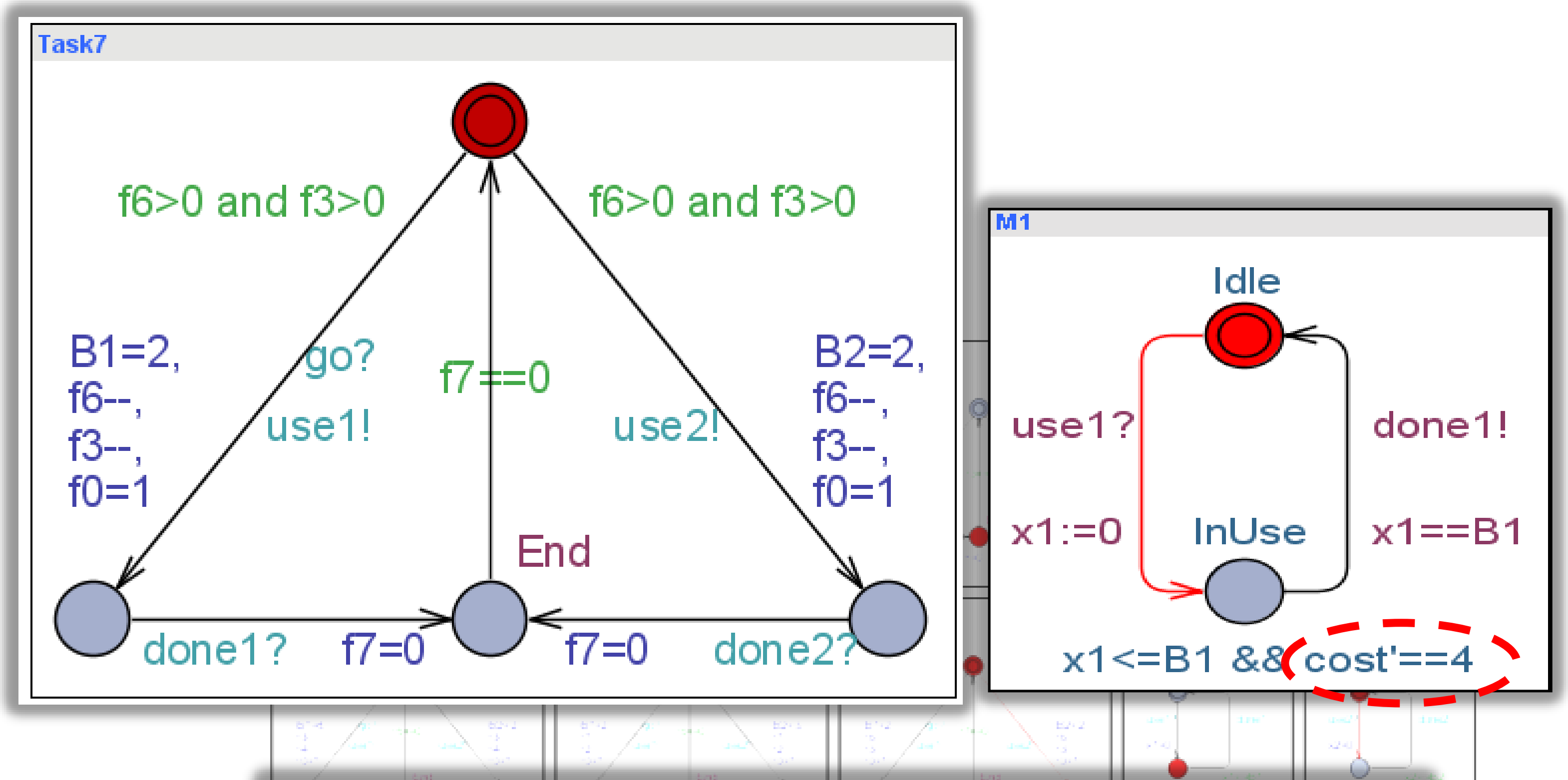
Optimal Infinite Scheduling





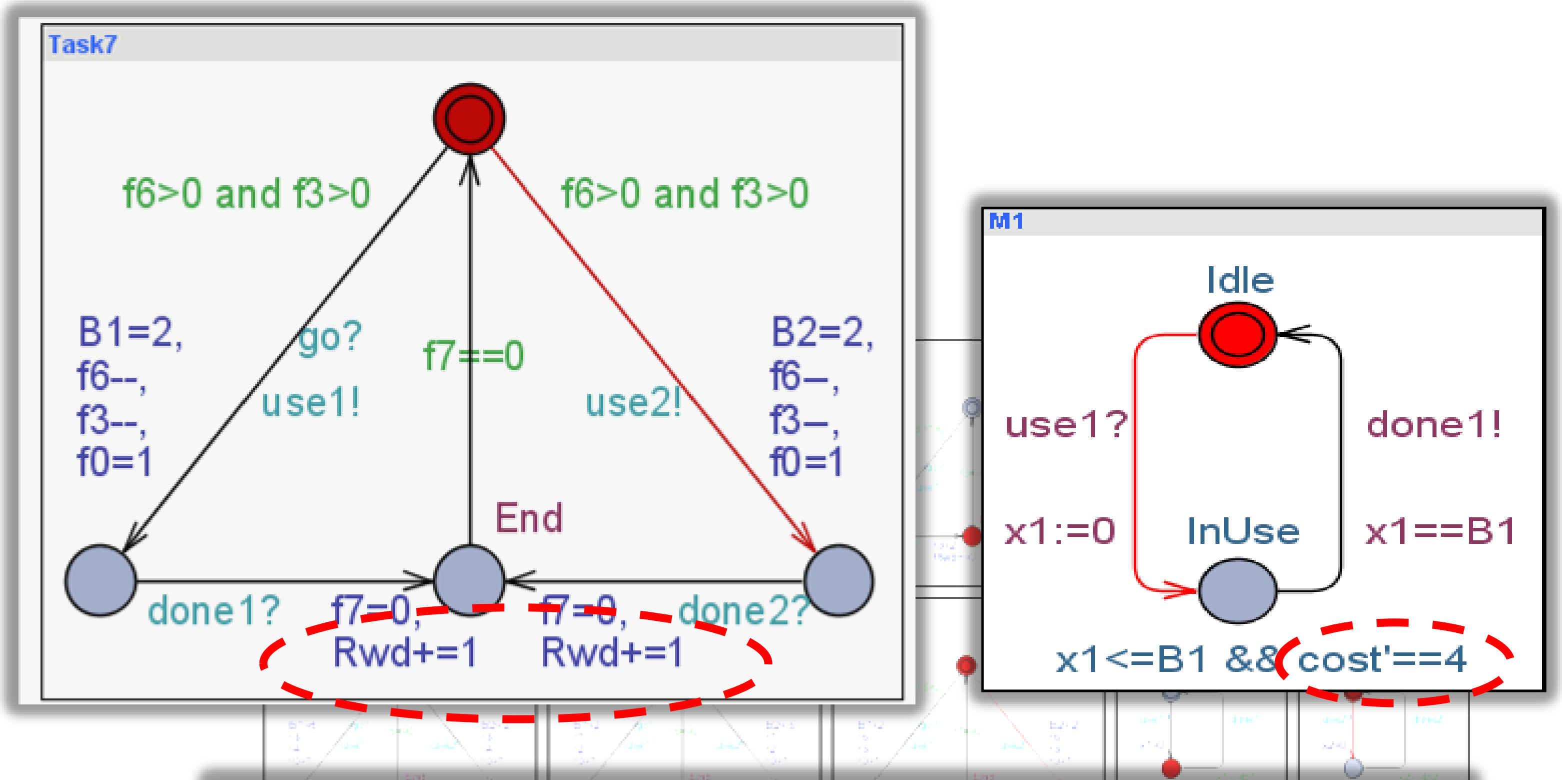
Maximize throughput:
i.e. maximize Reward / Time in the long run!

Optimal Infinite Scheduling



Minimize Energy Consumption:
i.e. minimize Cost / Time in the long run

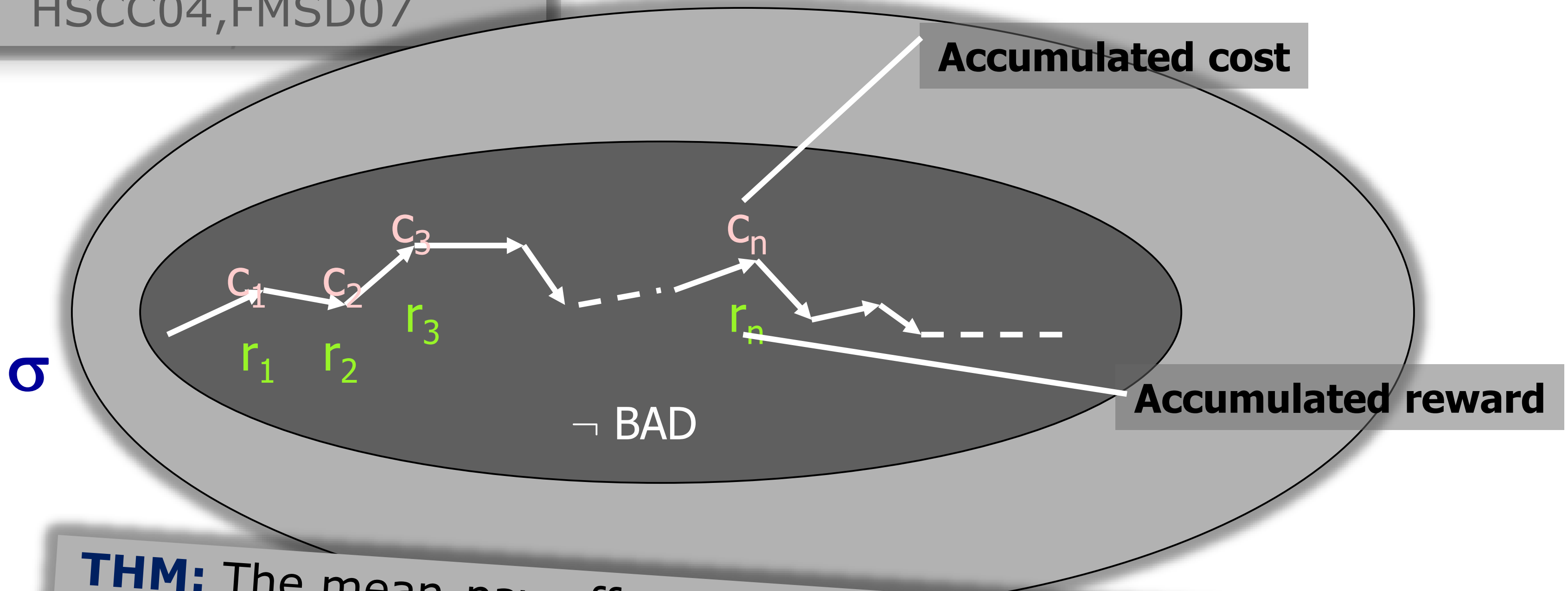
Optimal Infinite Scheduling



Maximize throughput:
i.e. maximize Reward / Cost in the long run

Mean Pay-Off Optimality

Bouyer, Brinksma, Larsen: HSCC04,FMSD07



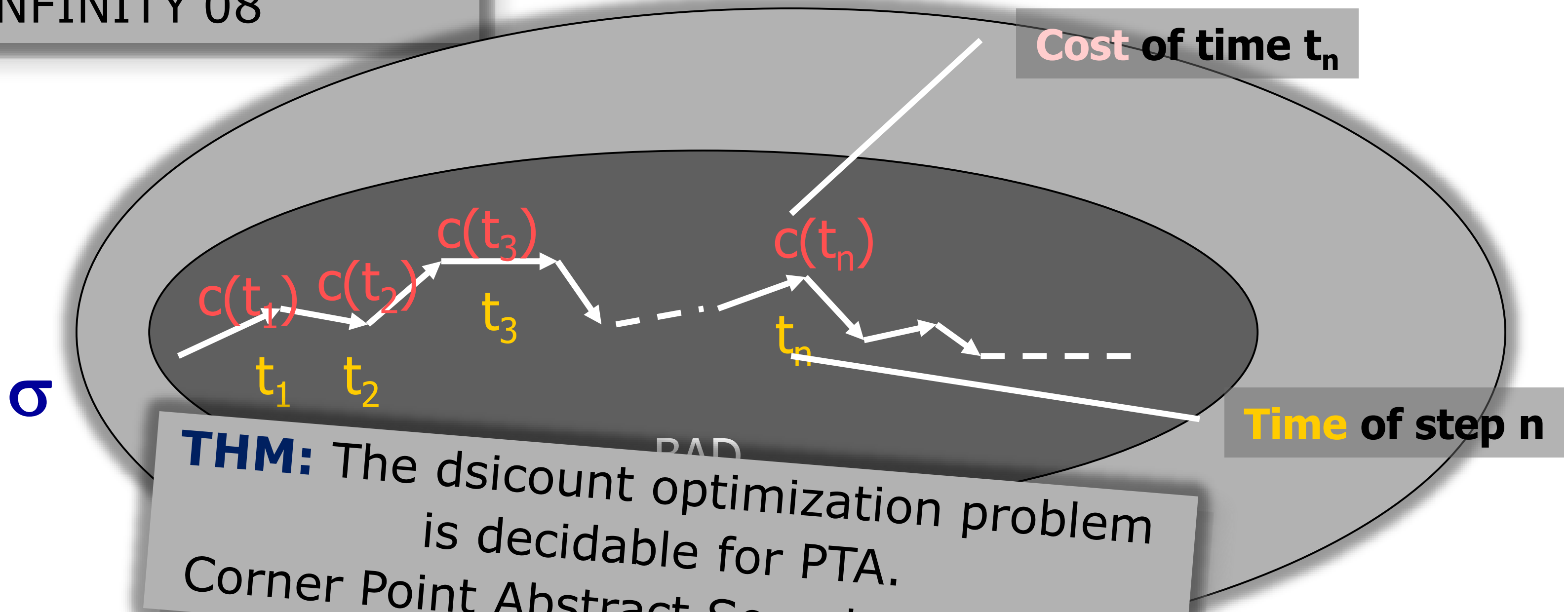
THM: The mean-pay off optimization problem is decidable (and PSPACE-complete) for PTA.
Corner Point Abstract Sound & Complete

Optimal Schedule σ^*: $val(\sigma^*) = \inf_\sigma val(\sigma)$

Discount Optimality

$\lambda < 1$: discounting factor

Larsen, Fahrenberg: INFINITY'08



THM: The dsicount optimization problem is decidable for PTA.
Corner Point Abstract Sound & Complete

Value of path σ: $\quad \mathrm{val}(\sigma) = \int_{t=0}^{t=\infty} c(t)\lambda^t dt$

Optimal Schedule σ^*: $\quad \mathrm{val}(\sigma^*) = \inf_\sigma \mathrm{val}(\sigma)$

Soundness of Corner Point Abstraction

Lemma
Let Z be a (bounded, closed) zone and let f be a(well-defined) function over Z defined by:

$$f : (t_1, \ldots, t_n) \mapsto \frac{a_1 t_1 + \cdots + a_n t_n + a}{c_1 t_1 + \cdots + c_n t_n + d}$$

then $\inf_Z f$ is obtained at a corner-point of Z (with integer coefficients).

Lemma
Let Z be a (bounded, closed) zone and let f be a function over Z defined by:

$$f : (t_1, \ldots, t_n) \mapsto a_1 \lambda^{t_1} + \cdots a_n \lambda^{t_n} + a$$

then $\inf_Z f$ is obtained at a corner-point of Z (with integer coefficients).

Optimal Conditional Reachability

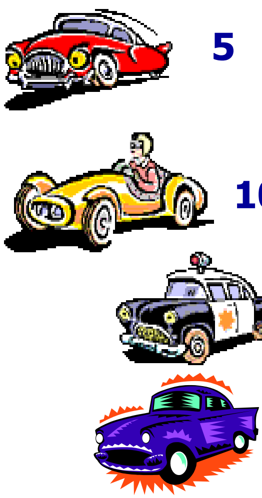
with Jacob I. Rasmussen

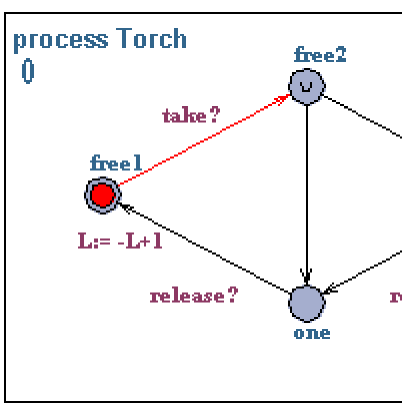
CISS
CENTER FOR INDLEJREDE SOFTWARE SYSTEMER

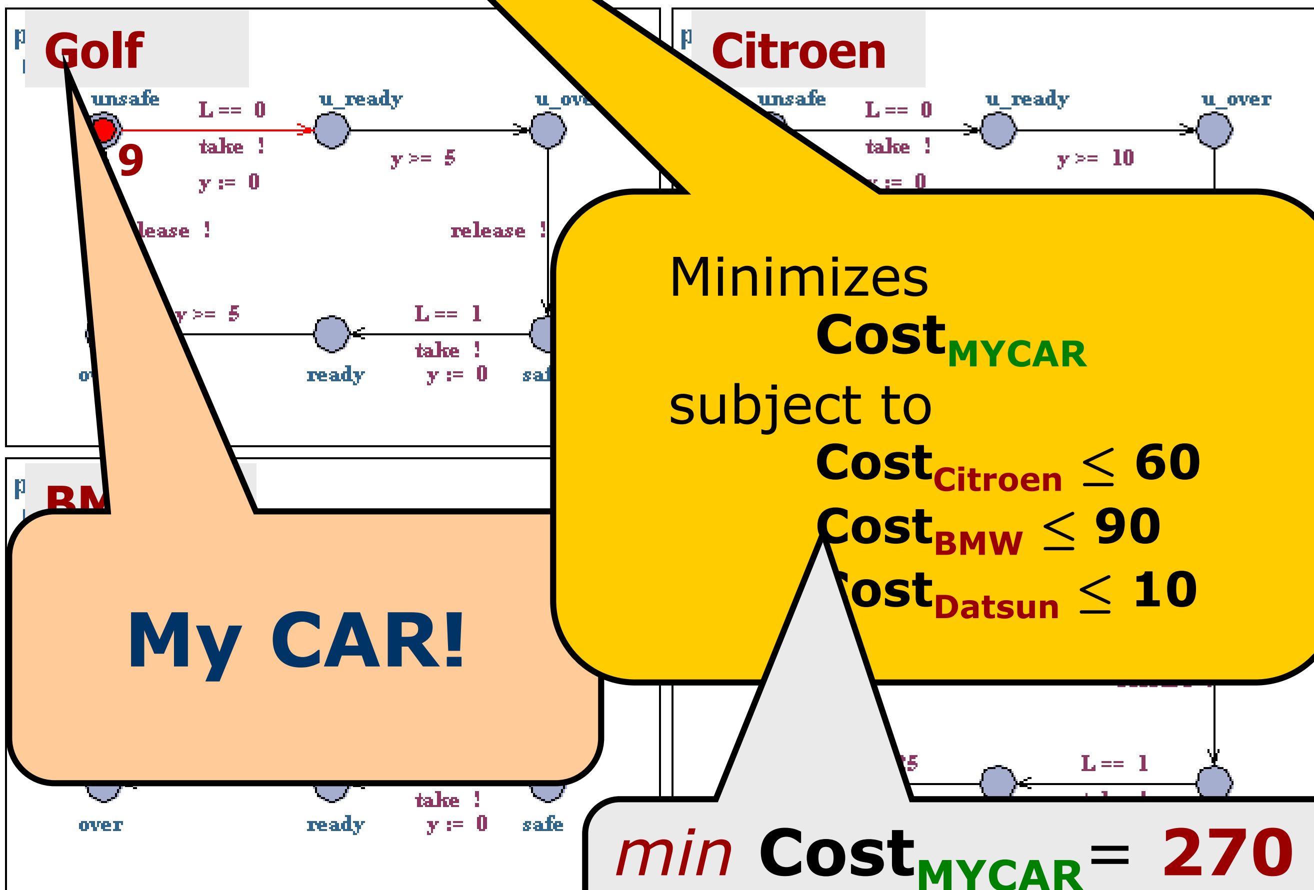
EXAMPLE: Optimal rescue plan for cars with different subscription rates for city driving !

CONDITIONAL

UNSAFE







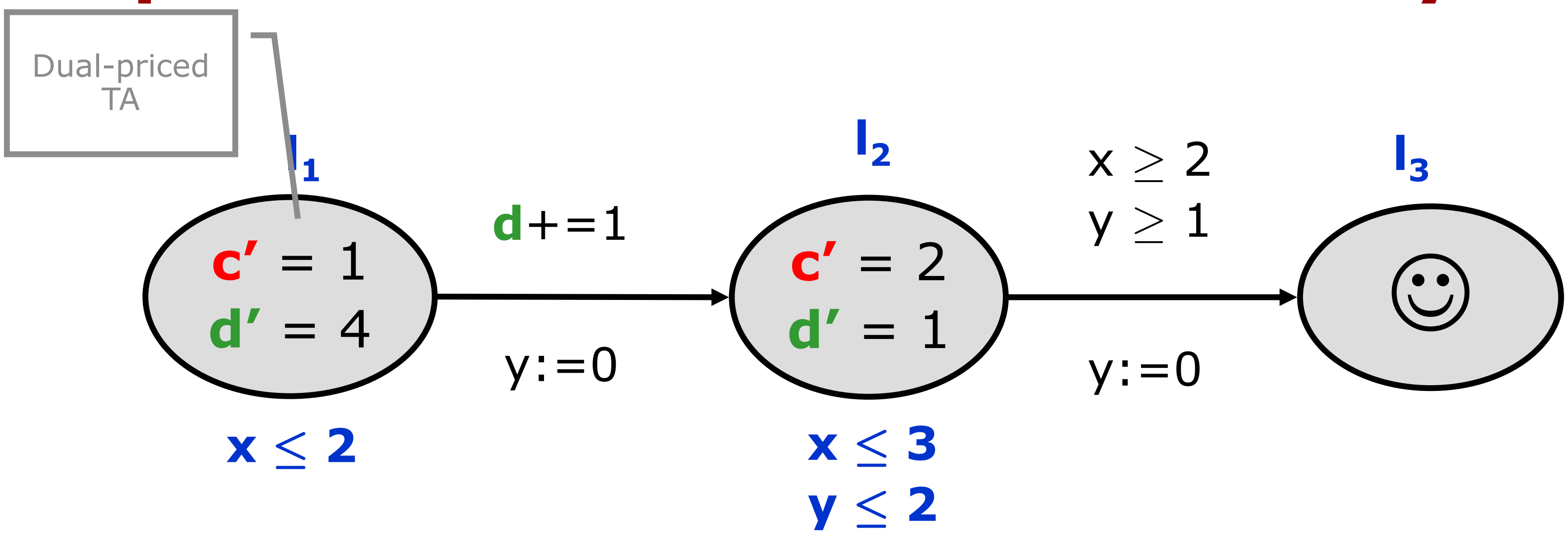
My CAR!

Minimizes
$\mathbf{Cost}_{MYCAR}$
subject to
$\mathbf{Cost}_{Citroen} \leq 60$
$\mathbf{Cost}_{BMW} \leq 90$
$\mathbf{Cost}_{Datsun} \leq 10$

min $\mathbf{Cost}_{MYCAR} = 270$
time = 70

UCb

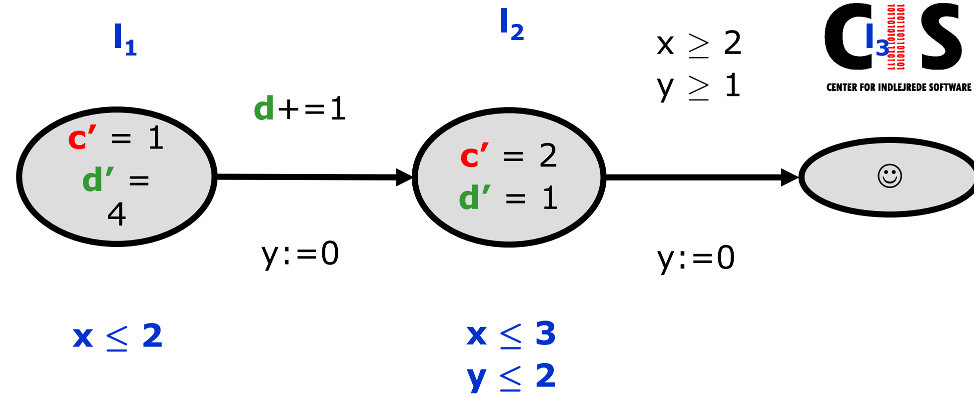
Optimal *Conditional* Reachability

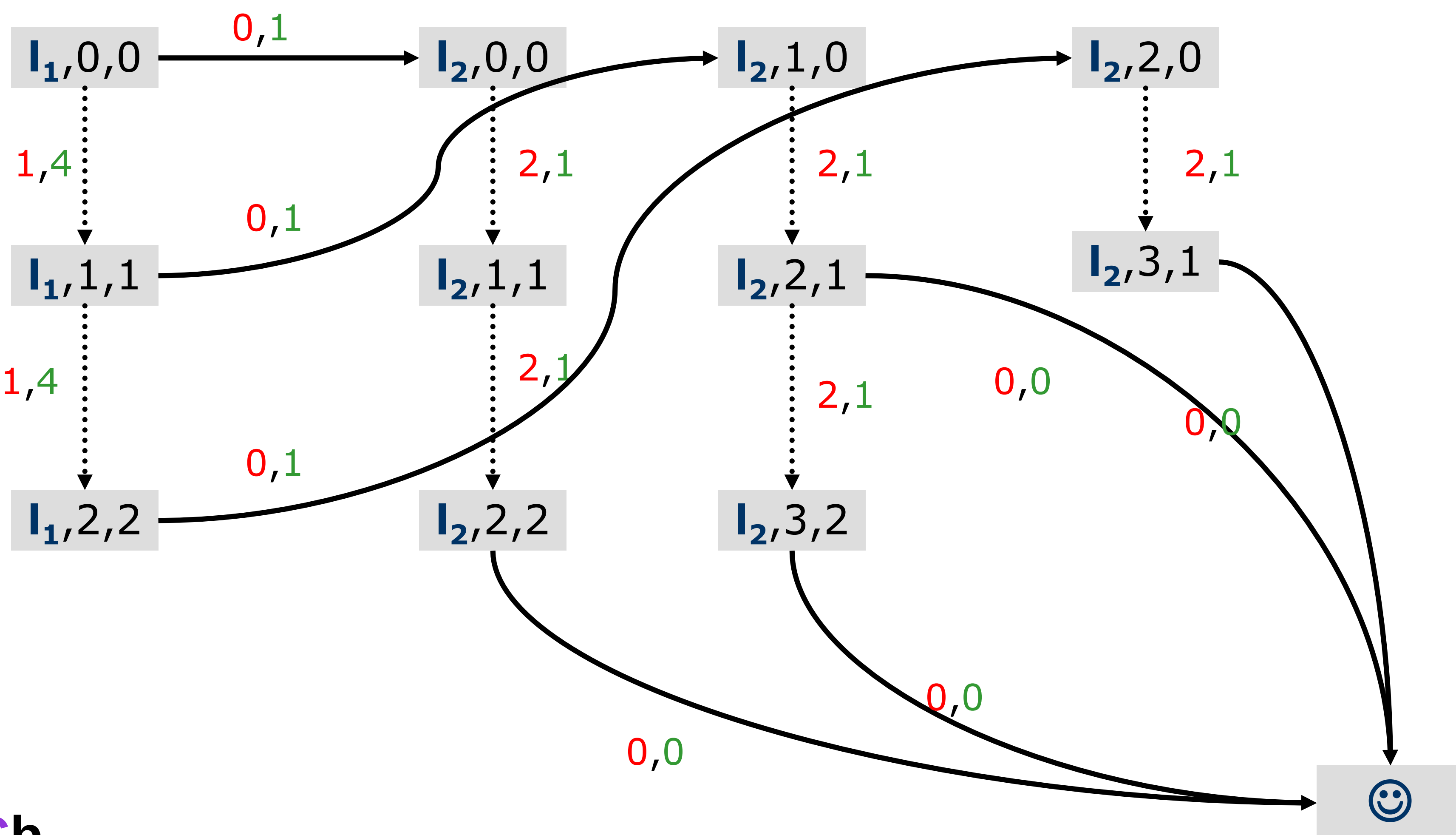


PROBLEM:
Reach l_3 in a way which
minimizes c
subject to $d \leq 4$

SOLUTION:
$c = 11/3 \rightarrow$
wait 1/3 in l_1; goto l_2;
wait 5/3 in l_2; goto l_3

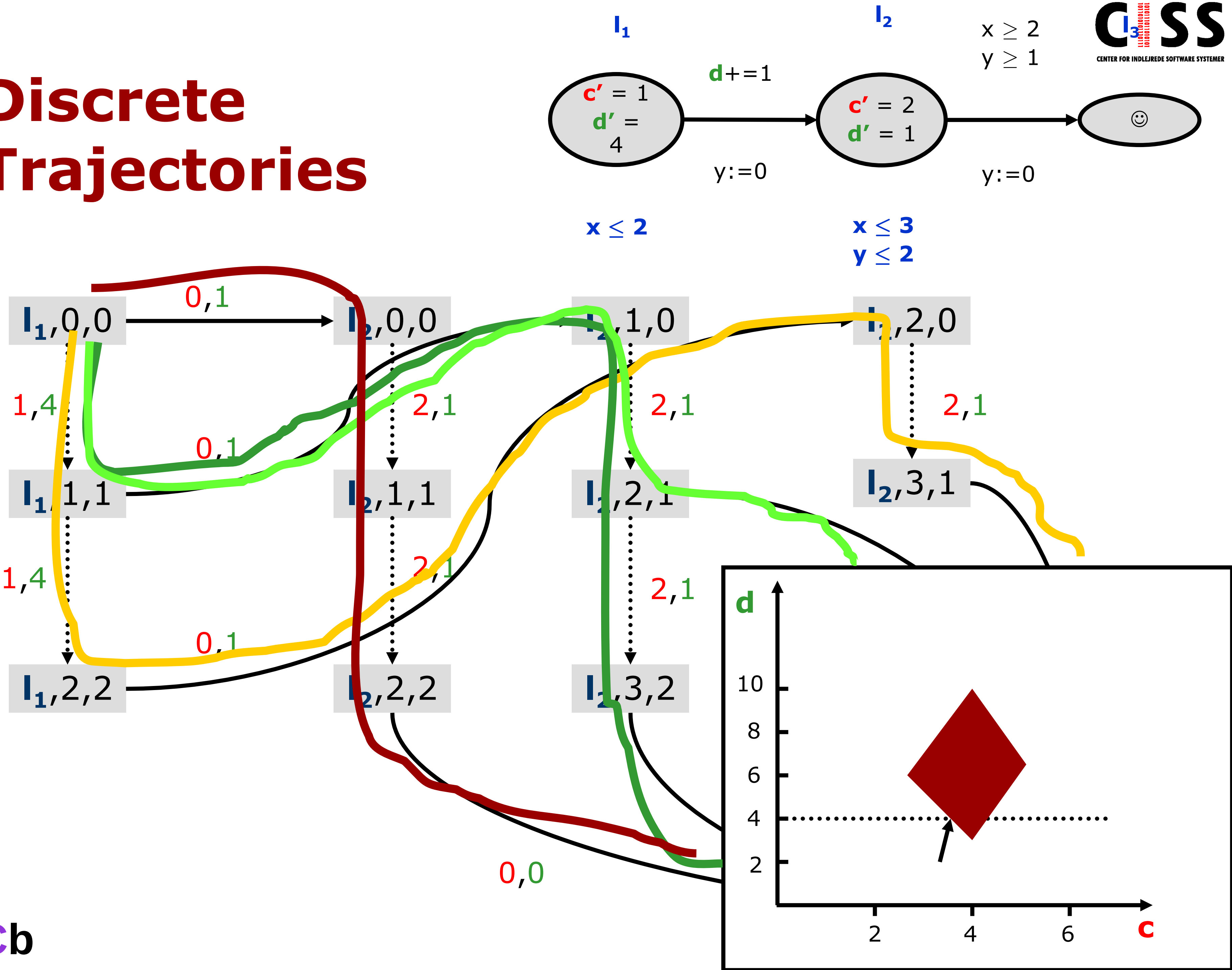
Discrete Trajectories





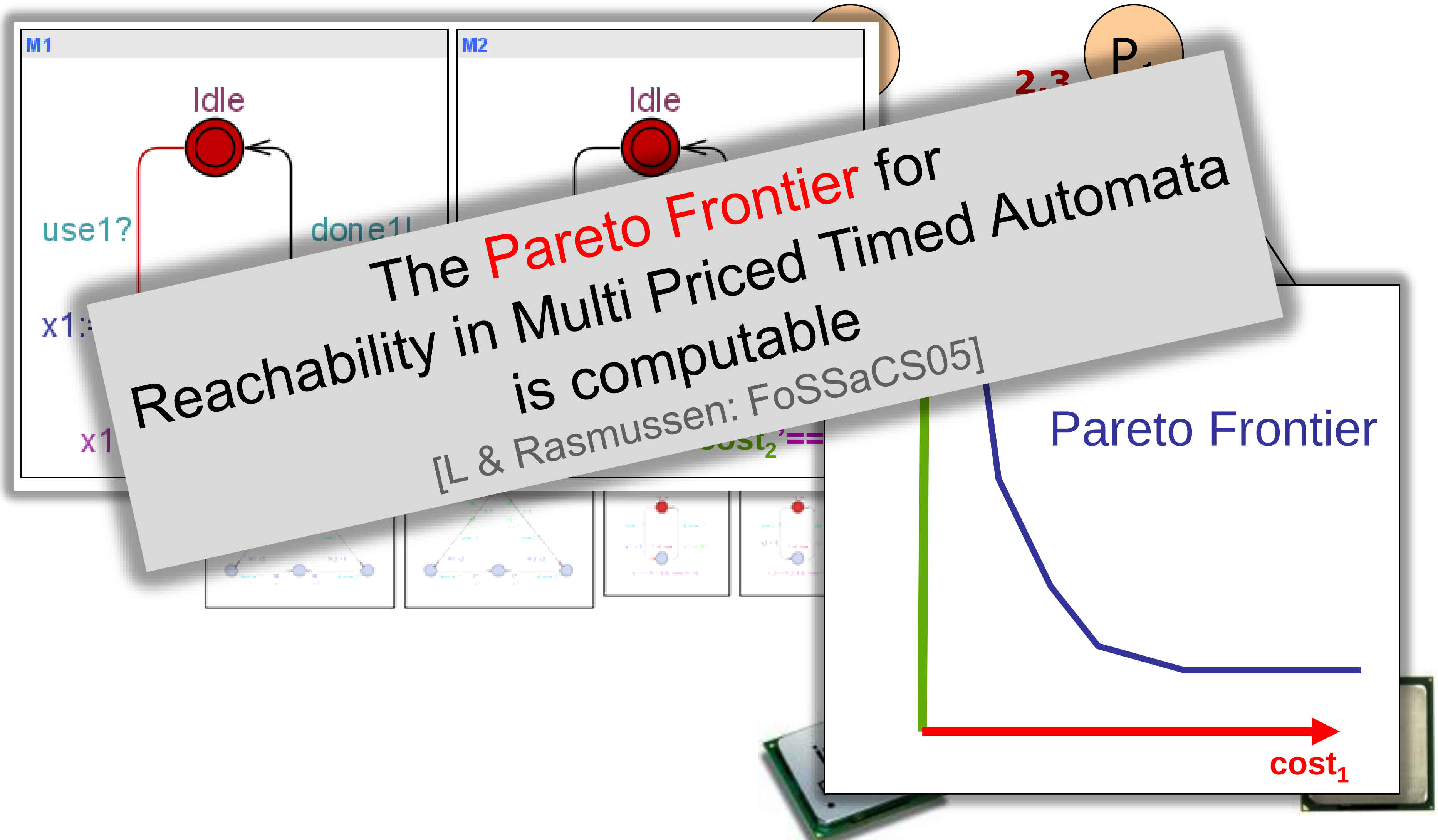
UCb

Discrete Trajectories



UCb

Multiple Objective Scheduling



The Pareto Frontier for Reachability in Multi Priced Timed Automata is computable

[L & Rasmussen: FoSSaCS05]