Efficiently Finding Ratio-Optimal Infinite Cycles in Doubly-Priced Timed Automata: Supplementary Material

Primary Keywords: *Temporal Planning*

This supplementary material provides additional insights to our benchmarks and some extended results.

Extended Results

Figure 1 shows how the quality of the solution evolves over time. This is the same data as in the paper, but without normalizing with respect to the optimal ratio.

In Table 1, extra data about the evaluation is shown (see caption).

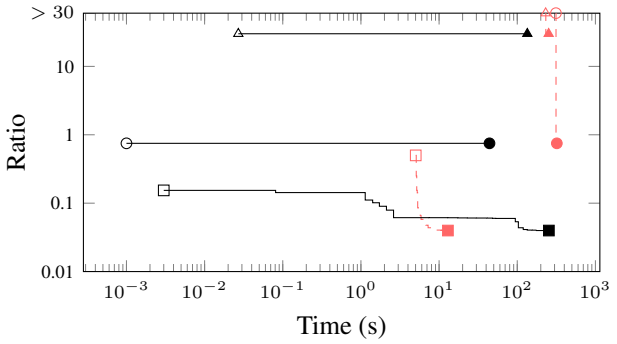


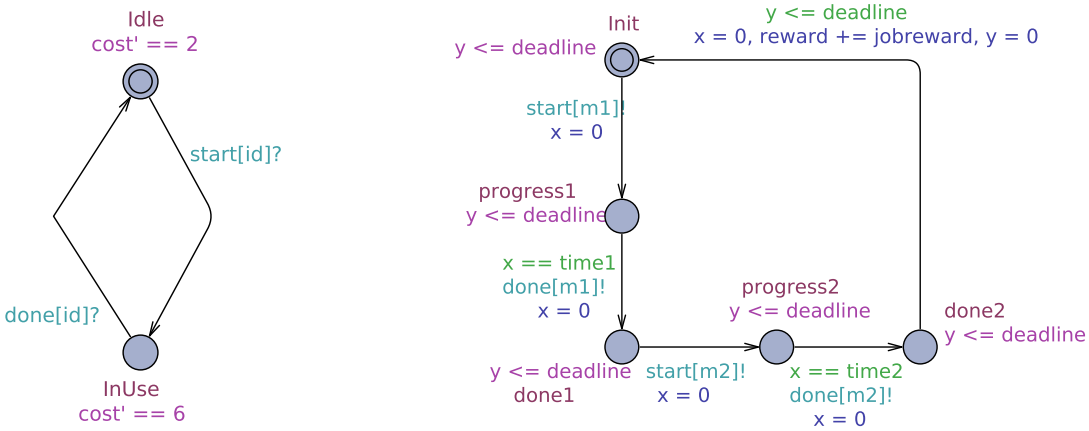
Figure 1: The ratio over time for the instances $J_{2,3}$ (triangle), $V_{2,1,2}$ (square), $S_{3,2}$ (circle). Black/solid is *CP-MCR* and red/dashed is *S-λD*.

Benchmarks

In Figures 2, 3, and 4, the graphical representations of the networks of timed automata used in the evaluation are shown.

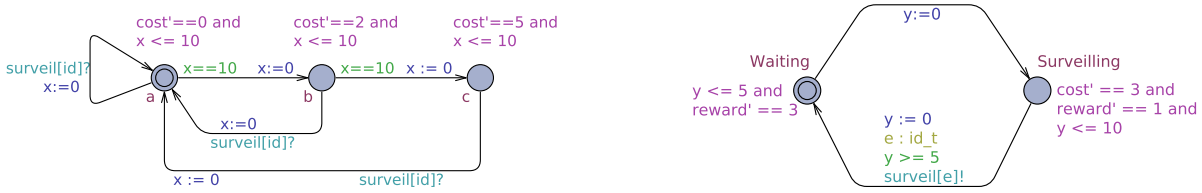
	CP-MCR							S-λD						
Inst.	Time	Memory	States	RT	Iter	SL	FCT	Time	Memory	States	RT	Iter	SL	FCT
$J_{2,1}$	**0.00**	**10**	173	**0.00**	2	5	0.000	**0.00**	11	12	**0.00**	2	5	0.000
$J_{2,2}$	0.29	30	20k	0.23	2	10	0.227	**0.07**	**12**	1k	**0.03**	4	10	0.002
$J_{2,3}$	252.29	2249	2M	232.00	3	15	230.863	**134.51**	**193**	225k	**0.03**	2	15	0.027
$J_{2,4}$	OOT	-	-	-	1	-	-	OOT	-	-	-	2	-	0.098
$J_{3,2}$	0.54	36	27k	0.35	2	10	0.348	**0.06**	**12**	1k	**0.01**	4	10	0.002
$J_{3,3}$	OOT	-	-	-	1	-	-	**494.80**	**301**	357k	**0.65**	3	25	0.164
$J_{3,4}$	OOT	-	-	-	1	-	-	OOT	-	-	-	5	-	4.649
$V_{1,1,1}$	**0.03**	**12**	1k	**0.02**	9	58	0.020	0.14	13	4k	0.10	13	58	0.000
$V_{1,1,2}$	**1.23**	**34**	20k	**1.15**	14	476	0.607	33.91	70	143k	18.74	14	476	0.002
$V_{1,1,3}$	**363.67**	**715**	542k	**359.67**	38	7118	123.726	OOT	-	-	-	5	-	0.255
$V_{1,2,1}$	**5.07**	**87**	60k	**3.97**	13	221	2.834	62.32	93	235k	27.46	16	221	0.003
$V_{1,2,2}$	**320.92**	**953**	694k	**287.45**	14	77	279.118	OOT	-	-	-	11	-	0.024
$V_{1,2,3}$	OOT	-	-	-	1	-	-	OOT	-	-	-	7	-	0.591
$V_{2,1,1}$	**0.23**	25	10k	**0.20**	10	31	0.169	0.54	**16**	20k	0.43	14	31	0.000
$V_{2,1,2}$	**12.97**	**162**	115k	**12.06**	16	257	5.025	253.93	374	1M	173.78	22	257	0.003
$V_{2,1,3}$	OOT	-	-	-	1	-	-	OOT	-	-	-	5	-	0.283
$S_{1,1}$	**0.00**	**10**	78	**0.00**	2	3	0.000	**0.00**	11	7	**0.00**	2	2	0.000
$S_{1,2}$	0.04	13	3k	0.03	7	6	0.032	**0.00**	**11**	119	**0.00**	4	6	0.001
$S_{1,3}$	1.70	74	63k	1.48	5	9	1.370	**0.21**	**14**	3k	**0.01**	4	11	0.002
$S_{1,4}$	70.81	641	636k	64.71	4	16	63.781	**11.35**	**40**	71k	**0.01**	2	16	0.005
$S_{1,5}$	OOT	-	-	-	1	-	-	**54.22**	**150**	213k	**0.03**	2	20	0.033
$S_{1,6}$	OOT	-	-	-	1	-	-	**49.94**	**306**	373k	**3.51**	2	24	3.509
$S_{1,7}$	OOT	-	-	-	1	-	-	**87.44**	**675**	702k	**N/A**	1	-	-
$S_{2,1}$	0.01	**11**	1k	0.01	3	4	0.009	**0.00**	11	39	**0.00**	2	4	0.000
$S_{2,2}$	2.15	67	50k	1.84	3	4	1.804	**0.14**	**12**	1k	**0.00**	2	5	0.001
$S_{2,3}$	OOT	-	-	-	1	-	-	**93.83**	**91**	102k	**0.01**	2	15	0.005
$S_{2,4}$	OOT	-	-	-	1	-	-	OOT	-	-	-	5	-	0.012
$S_{3,1}$	0.35	25	14k	0.29	3	6	0.280	**0.04**	**12**	629	**0.00**	2	6	0.001
$S_{3,2}$	320.94	975	770k	313.35	3	6	312.507	**44.15**	**50**	50k	**0.00**	2	7	0.001
$S_{3,3}$	OOT	-	-	-	1	-	-	OOT	-	-	-	2	-	0.003

Table 1: Results of comparison between *S-λD* and *CP-MCR*. Run with a 10-minute time limit and 10GB memory limit—OOT means that the time limit was exceed, the memory limit was never exceeded. The columns are (in order): the problem instance; the time in seconds until termination; the total memory usage in megabytes; the number of either concrete or symbolic states; the time in seconds it took for the algorithm to find the optimal *Ratio* (only written when the algorithm terminates); the number of iterations of the algorithm; the solution length; and finally the time to find the first solution.



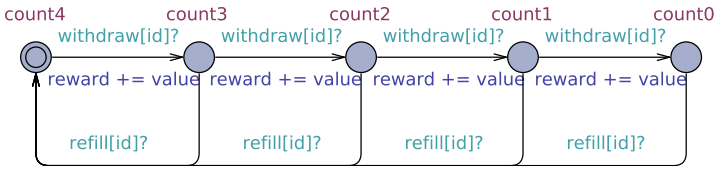
(a) Machine. (b) Job.

Figure 2: Job scheduling problem extended with cost and rewards. The aim is to schedule jobs on machines, where the cost is higher when running the machine and reward is gained by finishing a job.

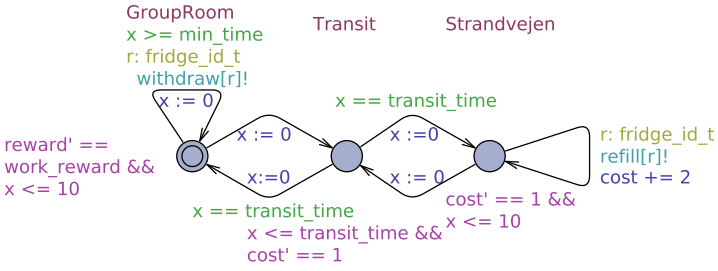


(a) Place. (b) Agent.

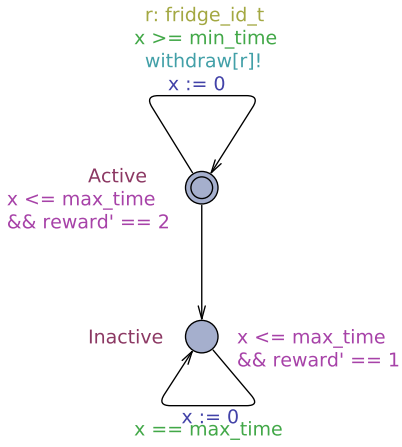
Figure 3: Surveillance scheduling problem. The aim is for a set of agents to surveil a set of locations, where leaving a location unsurveilled for a long period of time increases the cost, and the allowing the agent to do nothing increases the reward.



(a) Refrigerator.



(b) Volunteer.



(c) Consumer.

Figure 4: Volunteer scheduling problem. The aim is for a set of volunteering students to refill the refrigerators at the university social club, such that the consumers may consume their contents and remain active members. Active members; volunteers spending time studying; and dispensing items from the refrigerator yields reward, while refilling the refrigerator increases cost.