Bounding the Probability of Resource Constraint Violations in Multi-Agent MDPs Supplementary Material

Frits de Nijs and **Erwin Walraven** and **Mathijs M. de Weerdt** and **Matthijs T. J. Spaan**
Delft University of Technology
Mekelweg 4, 2628 CD
Delft, The Netherlands

This document contains a full description and pseudocode of the Column Generation algorithm. We also provide more details about Lottery domain used in the experiments.

Column Generation Pseudocode

Algorithm 1 shows how columns can be incrementally added to the master LP. The algorithm is an adaptation of the method introduced by Yost and Washburn (2000). Since columns correspond to policies, we will use both terms interchangeably. For each MDP M_i the columns are stored in the set Z_i, and the algorithm adds initial policies on lines 3–7, which does not consume any resources. The latter is necessary to ensure that the linear program has a feasible solution satisfying the constraints. The master LP is solved on line 10 to obtain dual prices $\lambda_{j,t}$, after which new columns are generated for each MDP on lines 14–23. The function call on line 17 is a call to our pruning subroutine, shown in Algorithm 2. The algorithm uses the alternative reward function $G_{i,t}$ on line 19. However, the expected reward computed on line 20 does not involve cost components that were subtracted in $G_{i,t}$. The algorithm keeps track of a lower bound ϕ_l and upper bound ϕ_u on the optimal objective value ϕ to detect convergence. Eventually the algorithm returns a probability distribution over policies for each MDP M_i, which is represented by tuples in a set Y_i.

```
input  : $Z_i$, $\{W_{i,1}, \dots, W_{i,\rho}\}$, $\delta$
output : column set $Z'_i$
1  $Z'_i \leftarrow W_{i,\rho} \cup W_{i,\rho-1} \cup \dots \cup W_{i,\rho-(\delta-1)}$
2  $E \leftarrow Z_i \setminus Z'_i$
3  foreach $\pi \in E$ do
4      remove column from master LP: $Z_i \leftarrow Z_i \setminus \{\pi\}$
5  end
6  return $Z'_i$
```

Algorithm 2: Column pruning function `prune`.

```
input  : MDP $M_i$ for each agent $i$, prune parameter $\delta$
output : policies for each $M_i$
1  $\phi_l \leftarrow -\infty,\ \phi_u \leftarrow \infty,\ \rho \leftarrow 0,\ W_{i,\rho} \leftarrow \emptyset\ \forall i$
2  initialize empty master LP: $Z_i \leftarrow \emptyset\ \forall i$
3  foreach $i = 1, \dots, n$ do
4      $\pi_i \leftarrow$ policy for $M_i$ consuming no resources
5      compute $V_{i,\pi_i}$ and $C^{j,t}_{i,\pi_i}$ using $\pi_i$ $(\forall j, t)$
6      add column: $Z_i \leftarrow Z_i \cup \{\pi_i\}$
7  end
8  do
9      $\rho \leftarrow \rho + 1$
10     solve the master LP to obtain $\lambda_{j,t}$ $(\forall j, t)$
11     $\phi_l \leftarrow$ objective value of the master LP
12     $\phi_{l,\rho} \leftarrow \phi_l$
13     $\phi_u \leftarrow \sum_{j,t} \lambda_{j,t} L_{j,t}$
14     foreach $i = 1, \dots, n$ do
15         $W_{i,\rho} \leftarrow \{\pi_i \mid \pi_i \in Z_i \text{ and } x_{i,\pi_i} > 0\}$
16         if $\rho > 1 \wedge \phi_{l,\rho} > \phi_{l,\rho-1}$ then
17             $Z_i \leftarrow$ prune($Z_i, \{W_{i,1}, \dots, W_{i,\rho}\}, \delta$)
18         end
19         solve $M_i$ using $G_{i,t}$ to obtain $\pi_i$
20         compute $V_{i,\pi_i}$ and $C^{j,t}_{i,\pi_i}$ using $\pi_i$ $(\forall j, t)$
21         add column: $Z_i \leftarrow Z_i \cup \{\pi_i\}$
22         $\phi_u \leftarrow \phi_u + (V_{i,\pi_i} - \sum_{j,t} \lambda_{j,t} C^{j,t}_{i,\pi_i})$
23     end
24 while $\phi_u - \phi_l > 0$;
25 $Y_i \leftarrow \{(\pi_i, x_{i,\pi_i}) \mid \pi_i \in Z_i \text{ and } x_{i,\pi_i} > 0\}\ \forall i$
26 return $\{Y_1, \dots, Y_n\}$
```

Algorithm 1: Column Generation method

Copyright © 2017, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

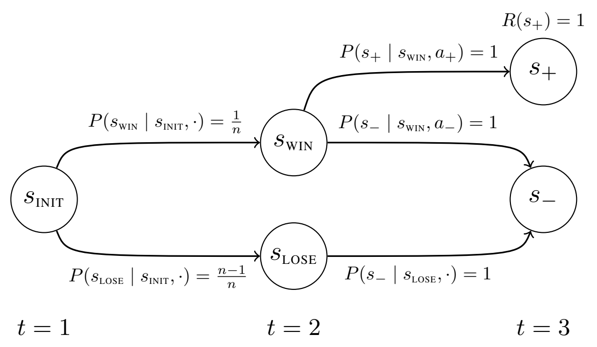
$R(s_+) = 1$
$P(s_+ \mid s_{\text{WIN}}, a_+) = 1$
s_+
$P(s_{\text{WIN}} \mid s_{\text{INIT}}, \cdot) = \frac{1}{n}$
s_{WIN}
$P(s_- \mid s_{\text{WIN}}, a_-) = 1$
s_{INIT}
s_-
s_{LOSE}
$P(s_{\text{LOSE}} \mid s_{\text{INIT}}, \cdot) = \frac{n-1}{n}$
$P(s_- \mid s_{\text{LOSE}}, \cdot) = 1$
$t = 1$
$t = 2$
$t = 3$

Figure 1: Graphical description of the *Lottery* agents' MDP model, presenting the transition and reward functions. Unspecified transitions and rewards have value 0.

Lottery Domain

The *Lottery* problem consists of n identical agents, which are jointly trying to win a lottery. The agents know the odds of the lottery, so that they are able to purchase sufficient tickets to win the jackpot *in expectation*. They divide the load of purchasing the tickets equally amongst themselves, so that each agent has probability $\frac{1}{n}$ to own a winning ticket. The lottery draw is distributed, so that each agent can individually determine whether or not it has a winning ticket. Agents owning winning tickets are expected to coordinate their actions so that only one redeems the indivisible prize.

Formally, the Lottery problem is modeled as an MMDP with Global Resource Constraints as follows:

$$\begin{aligned} S_i &= \langle s_{\text{INIT}}, s_{\text{LOSE}}, s_{\text{WIN}}, s_-, s_+ \rangle \\ A_i &= \langle a_-, a_+ \rangle \quad C_{i,1}(\cdot, a_-) = 0 \\ h &= 3 \qquad\quad\;\; C_{i,1}(\cdot, a_+) = 1 \\ s^{i,1} &= s_{\text{INIT}} \qquad\qquad\; L_{1,t} = 1 \end{aligned} \tag{1}$$

Figure 1 presents the transition function and the rewards of the model. The action a_+ is used to claim the prize, which is constrained by the resource limit function to be used by at most 1 agent. Using action a_+ in the state s_{WIN} indicating ownership of a winning ticket results in reaching the only state with positive reward, s_+.

The goal of the agents is to have one of them use the resource to reach s_+, to distribute the reward across all of them. The problem the agents face is that they cannot know a priori which agent(s) will transition to state s_{WIN} since that is up to chance. All they know is that in expectation, one of them will reach s_{WIN}.

References

Yost, K. A., and Washburn, A. R. 2000. The LP/POMDP Marriage: Optimization with Imperfect Information. *Naval Research Logistics* 47(8):607–619.