Sequential Decision Making for Intelligent Agents
Papers from the AAAI 2015 Fall Symposium

Nested Value Iteration for Partially Satisfiable Co-Safe LTL Specifications (Extended Abstract)

Bruno Lacerda, David Parker and Nick Hawes
School of Computer Science, University of Birmingham
Birmingham, United Kingdom
{b.lacerda, d.a.parker, n.a.hawes}@cs.bham.ac.uk

Overview We describe our recent work (Lacerda, Parker, and Hawes 2015) on cost-optimal policy generation, for co-safe linear temporal logic (LTL) specifications that are not satisfiable with probability one in a Markov decision process (MDP) model. We provide an overview of the approach to pose the problem as the optimisation of three standard objectives in a *trimmed product MDP*. Furthermore, we introduce a new approach for optimising the three objectives, in a decreasing order of priority, based on a "nested" value iteration, where one value table is kept for each objective.

The overall goal of our work is to generate policies that maximise the *probability of success* and minimise the *undiscounted expected cumulative cost* to achieve a task specified in the *co-safe* fragment of LTL (i.e., a task that can be completed in a finite horizon). Furthermore, we tackle the question of what to do when the task becomes unsatisfiable during execution. In many cases, even if the probability of satisfying the overall task is zero, it is still possible to fulfil part of it. An illustrative example is a robot that needs to navigate to every office in a building to perform security checks. During execution some doors might be closed, making offices inaccessible. This will make the overall task unsatisfiable, yet we still want the robot to check as many offices as it can. We formalise this notion as a progression reward defined over the LTL automaton.

Given an MDP and a co-safe LTL specification, we show that the problems of (i) maximising the probability of satisfying a co-safe LTL formula; (ii) maximising the progression reward (i.e., fulfilling as much of the formula as possible); and (iii) minimising a cost function while performing (i) and (ii) can be solved independently by standard techniques in a *trimmed product MDP*. The main steps of the construction of this MDP are depicted in Fig. 1.

The problems above are conflicting. In (Lacerda, Parker, and Hawes 2015), we implement *objective prioritisation* by using *multi-objective model checking* techniques. However, the use of such queries is quite inefficient: in our experiments, the calculation of the optimal policy for a problem where the trimmed product MDP has approximately 20,000 states and 100,000 transitions took ~9 minutes, in a standard laptop computer. Hence, here we introduce a new, more efficient, approach, based on a nested version of value iteration.

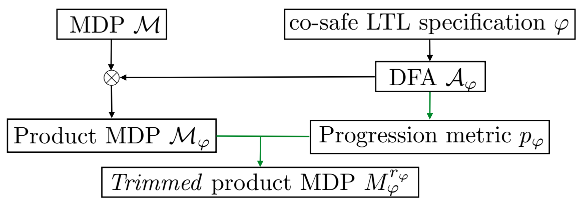
MDP $\mathcal{M}$
co-safe LTL specification φ
DFA $\mathcal{A}_\varphi$
$\otimes$
Product MDP $\mathcal{M}_\varphi$
Progression metric p_φ
Trimmed product MDP $M_\varphi^{r_\varphi}$

Figure 1: Diagram of the approach for the problem reduction, with our contributions represented by green arrows.

A Metric for Task Progression Co-safe LTL formulas φ over atomic propositions AP are defined by:
$\varphi ::= true \mid p \mid \neg p \mid \varphi \wedge \varphi \mid \mathtt{X}\,\varphi \mid \mathtt{F}\,\varphi \mid \varphi\,\mathtt{U}\,\varphi$, where $p \in AP$. The temporal operators X, F, and U, read "next", "eventually", and "until", respectively, allow for the specification of more general goals than the usual *state reachability* goals.

It is known (Kupferman and Vardi 2001) that for any co-safe LTL formula φ written over AP, we can build a deterministic finite automaton (DFA) $\mathcal{A}_\varphi = \langle Q, \overline{q}, Q_F, 2^{AP}, \delta_{\mathcal{A}_\varphi}\rangle$, where: Q is a finite set of states; $\overline{q} \in Q$ is the initial state; $Q_F \subseteq Q$ is the set of accepting states; 2^{AP} is the alphabet; and $\delta_{\mathcal{A}_\varphi} : Q \times 2^{AP} \to Q$ is a transition function. $\mathcal{A}_\varphi$ accepts exactly the sequences satisfying φ.

We propose a notion of *task progression* over $\mathcal{A}_\varphi$, defined from a distance metric $d_\varphi : Q \to \mathbb{Q}_{\geq 0}$ that maps each state of $\mathcal{A}_\varphi$ to a value representing how close we are to reaching an accepting state. Let $Suc_q \subseteq Q$ be the set of successors of state q, and $|\delta_{q,q'}| \in \{0, ..., 2^{|AP|}\}$ be the number of transitions from q to q'. We define $d_F(q_F) = 0$ for $q_F \in Q_F$, and $d_F(q) = \min_{q' \in Suc_q} \left\{ d_\varphi(q') + \frac{1}{|\delta_{q,q'}|} \right\}$ for $q \notin Q_F$ such that there is a sequence of transitions in $\mathcal{A}_\varphi$ that leads it from state q to state in Q_F. Furthermore, if there is no path from q to Q_F, we set $d_\varphi(q)$ to the maximum value $|Q|$.

This distance metric represents the number of states that need to be visited, starting in q, to reach a state in Q_F, balanced by the number of transitions between these states. It allows us to define the *progression* $p_\varphi : Q \times Q \to \mathbb{Q}_{\geq 0}$ between two successive states of $\mathcal{A}_\varphi$ as a measure of how much the distance is reduced by moving between q and q':

$$p_\varphi(q, q') = \begin{cases} \max\{0, d_\varphi(q) - d_\varphi(q')\} & \text{if } q' \in Suc_q \text{ and} \\ & q' \not\to^* q \\ 0 & \text{otherwise} \end{cases}$$

To guarantee convergence of infinite sums of values of p_φ, we require that the progression metric is non-negative, and that there are no cycles in the DFA with non-zero values.

Construction of the Trimmed Product Let $\mathcal{M} = \langle S, \overline{s}, A, \delta_{\mathcal{M}}, AP, Lab \rangle$ be an MDP, where: S is a finite set of states; $\overline{s} \in S$ is the initial state; A is a finite set of actions; $\delta_{\mathcal{M}} : S \times A \times S \to [0,1]$ is a probabilistic transition function, where $\sum_{s' \in S} \delta_{\mathcal{M}}(s,a,s') \in \{0,1\}$ for all $s \in S$, $a \in A$; AP is a set of atomic propositions; and $Lab : S \to 2^{AP}$ is a labelling function, such that $p \in Lab(s)$ iff p is true in $s \in S$, and $c : S \times A \to \mathbb{R}_{>0}$ be a cost structure over $\mathcal{M}$. Given a DFA $\mathcal{A}_\varphi = \langle Q, \overline{q}, Q_F, 2^{AP}, \delta_{\mathcal{A}_\varphi} \rangle$, one can build the cross-product $\mathcal{M}_\varphi = \mathcal{M} \otimes \mathcal{A}_\varphi = \langle S \times Q, \overline{s_\varphi}, A, \delta_{\mathcal{M}_\varphi}, AP, Lab_\varphi \rangle$, and extend the cost structure c to $\mathcal{M}_\varphi$ as $c_\varphi((s,q),a) = c(s,a)$. $\mathcal{M}_\varphi$ behaves like the original MDP $\mathcal{M}$, but is augmented with information about the satisfaction of φ. Once a path of $\mathcal{M}_\varphi$ reaches an *accepting state* (i.e., a state of the form (s, q_F)), it satisfies φ. The construction of the product MDP is well known and is such that $\mathcal{M}_\varphi$ preserves the probabilities of paths from $\mathcal{M}$ (see, e.g., (Baier and Katoen 2008)).

However, given that the probability of reaching an accepting state in $\mathcal{M}_\varphi$ is not one, the calculation of the minimal expected cost in this structure does not converge. In order to tackle this issue, we start by encoding p_φ as a reward structure $r_\varphi : (S \times Q) \times A \to \mathbb{Q}_{\geq 0}$:

$$r_\varphi((s,q),a) = \sum_{(s',q') \in S \times Q} \delta_{\mathcal{M}_\varphi}((s,q),a,(s',q')) p_\varphi(q,q')$$

The main insight for building the trimmed product MDP $\mathcal{M}_\varphi^{r_\varphi}$ is noticing that the progression metric is built in such a way that any infinite run of $\mathcal{M}_\varphi$ will eventually reach a state where no more progression reward can be gathered. This is due to the system reaching an accepting state, or a state where no more can be done towards the formula satisfaction. Thus, the trimming operation simply removes *all* states for which (i) the probability of gathering more progression reward is 0; and (ii) are not a successor of one such state. All transitions to these states are also removed from the structure. By doing this, we guarantee convergence of the calculation of the cumulative cost, because we removed *all* end components in the model with a positive cost.

Thus, our three objectives can be posed as standard problems in the trimmed product: (i) maximising the probability of satisfying φ can be reduced to maximising the probability of reaching an accepting state; (ii) maximising progression can be reduced to the infinite-horizon maximisation of the cumulative value of r_φ; and (iii) minimising the cost to reach a state where no more progression can be accumulated can be reduced to the infinite-horizon minimisation of the cumulative value of c_φ.

Nested Value Iteration Since the above problems are conflicting, we optimise them in decreasing order of priority.

Algorithm 1 keeps track of a value table per objective, and only uses the lower level priority objectives to decide which action to execute when the value for a pair of actions is the same for the higher-priority objective(s). Its termination is guaranteed due to the structure of the progression metric. It has similarities with (Teichteil-Königsbuch 2012; Kolobov, Mausam, and Weld 2012), which tackle a similar problem. However, they just take into account *single state reachability*, while we use co-safe LTL goals, which are more general, and introduce the notion of task progression.

Algorithm 1 NESTED VALUE ITERATION

```
Input: Trimmed product $\mathcal{M}_\varphi^{r_\varphi}$, cost $c_\varphi$, progression reward $r_\varphi$
Output: Optimal policy $\pi : S \times Q \to A$
 1: for all $s_\varphi \in S \times Q$ do
 2:     $V_p(s_\varphi) \leftarrow \begin{cases} 1 & \text{if } s_\varphi = (s, q_F), \text{where } q_F \in Q_F \\ 0 & \text{otherwise} \end{cases}$
 3:     $V_r(s_\varphi) \leftarrow 0$
 4:     $V_c(s_\varphi) \leftarrow 0$
 5: end for
 6: while $V_p$ or $V_r$ or $V_c$ have not converged do
 7:     for all $s_\varphi \in S \times Q$ do
 8:         for all $a \in A(s_\varphi)$ do
 9:             $v_p \leftarrow \sum_{s'_\varphi \in S \times Q} \delta_{\mathcal{M}_\varphi^{r_\varphi}}(s_\varphi, a, s'_\varphi) V_p(s'_\varphi)$
10:             $v_r \leftarrow r_\varphi(s_\varphi, a) + \sum_{s'_\varphi \in S \times Q} \delta_{\mathcal{M}_\varphi^{r_\varphi}}(s_\varphi, a, s'_\varphi) V_r(s'_\varphi)$
11:             $v_c \leftarrow c_\varphi(s_\varphi, a) + \sum_{s'_\varphi \in S \times Q} \delta_{\mathcal{M}_\varphi^{r_\varphi}}(s_\varphi, a, s'_\varphi) V_c(s'_\varphi)$
12:             if $v_p > V_p(s_\varphi)$ then
13:                 $V_p(s_\varphi) \leftarrow v_p$
14:                 $V_r(s_\varphi) \leftarrow v_r$
15:                 $V_c(s_\varphi) \leftarrow v_c$
16:                 $\pi(s_\varphi) \leftarrow a$
17:             else if $v_p = V_p(s_\varphi) \wedge v_r > V_r(s_\varphi)$ then
18:                 $V_r(s_\varphi) \leftarrow v_r$
19:                 $V_c(s_\varphi) \leftarrow v_c$
20:                 $\pi(s_\varphi) \leftarrow a$
21:             else if $v_p = V_p(s_\varphi) \wedge v_r = V_r(s_\varphi) \wedge v_c < V_c(s_\varphi)$ then
22:                 $V_c(s_\varphi) \leftarrow v_c$
23:                 $\pi(s_\varphi) \leftarrow a$
24:             end if
25:         end for
26:     end for
27: end while
```

This approach greatly improves the efficiency of policy generation for our problem, when compared with our previous approach based on contrained multi-objective queries. It solves our illustrative problem, that before took ~9 minutes, in ~10 seconds.

Acknowledgement This work has received funding from EU FP7 under grant agreement No 600623, STRANDS.

References

Baier, C., and Katoen, J.-P. 2008. *Principles of Model Checking*.

Kolobov, A.; Mausam; and Weld, D. 2012. A theory of goal-oriented MDPs with dead ends. In *Proc. of UAI '12*.

Kupferman, O., and Vardi, M. 2001. Model checking of safety properties. *Formal Methods in System Design* 19(3).

Lacerda, B.; Parker, D.; and Hawes, N. 2015. Optimal policy generation for partially satisfiable co-safe LTL specifications. In *Proc. of IJCAI '15*.

Teichteil-Königsbuch, F. 2012. Stochastic safest and shortest path problems. In *Proc. of AAAI '12*.