Artificial Intelligence 216 (2014) 179–205

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Depth-based short-sighted stochastic shortest path problems

Felipe W. Trevizan *, Manuela M. Veloso

School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

ARTICLE INFO

Article history:
Received 12 November 2012
Received in revised form 13 June 2014
Accepted 6 July 2014
Available online 14 July 2014

Keywords:
Probabilistic planning
Stochastic Shortest Path Problems
Markov Decision Processes

ABSTRACT

Stochastic Shortest Path Problems (SSPs) are a common representation for probabilistic planning problems. Two approaches can be used to solve SSPs: (i) consider all probabilistically reachable states and (ii) plan only for a subset of these reachable states. Closed policies, the solutions obtained in the former approach, require significant computational effort, and they do not require replanning, i.e., the planner is never re-invoked. The second approach, employed by replanners, computes open policies, i.e., policies for a subset of the probabilistically reachable states. Therefore, when a state is reached in which the open policy is not defined, the replanner is reinvoked to compute a new open policy. In this article, we introduce a special case of SSPs, the depth-based short-sighted SSPs, in which every state has a nonzero probability of being reached using at most t actions. We also introduce the novel algorithm Short-Sighted Probabilistic Planner (SSiPP), which solves SSPs through depth-based short-sighted SSPs and guarantees that at least t actions can be executed without replanning. Therefore, SSiPP can compute both open and closed policies: as t increases, the returned policy approaches the behavior of a closed policy, and for t large enough, the returned policy is closed. Moreover, we present two extensions to SSiPP: Labeled-SSiPP and SSiPP-FF. The former extension incorporates a labeling mechanism to avoid revisiting states that have already converged. The latter extension combines SSiPP and determinizations to improve the performance of SSiPP in problems without dead ends. We also performed an extensive empirical evaluation of SSiPP and its extensions in several problems against state-of-the-art planners. The results show that (i) Labeled-SSiPP outperforms SSiPP and the considered planners in the task of finding the optimal solution when the problems have a low percentage of relevant states; and (ii) SSiPP-FF outperforms SSiPP in the task of quickly finding suboptimal solutions to problems without dead ends while performing similarly in problems with dead ends.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Stochastic Shortest Path Problems (SSP) [3] provide a convenient framework for modeling fully observable probabilistic planning problems. A solution to an SSP is a policy – a mapping from states to actions – that is guaranteed to reach a goal state when executed from the initial state of the given SSP. In this article, we address the question of how to improve the scalability of probabilistic planners when searching for (i) an optimal policy and (ii) a suboptimal policy given a time deadline.

* Corresponding author.
E-mail addresses: fwt@cs.cmu.edu (F.W. Trevizan), mmv@cs.cmu.edu (M.M. Veloso).

http://dx.doi.org/10.1016/j.artint.2014.07.001
0004-3702/© 2014 Elsevier B.V. All rights reserved.

One approach to computing optimal solutions to SSPs is to use value iteration and policy iteration algorithms, which are optimal [4]. Planners based on these algorithms return a closed policy, i.e., a policy that is defined at least over all the probabilistically reachable states of the given SSP. Assuming the model correctly captures the cost and uncertainty of the actions in the environment, closed policies are extremely powerful as their execution never "fails"; therefore, the planner is never reinvoked. Unfortunately, the computation of such policies is prohibitive in complexity as problems scale up. The efficiency of value-iteration-based probabilistic planners can be improved by combining asynchronous updates and heuristic search (e.g., Labeled RTDP [5]), resulting in optimal algorithms with convergence bounds. Although these techniques allow planners to compute compact policies, in the worst case, these policies are still linear in the size of the state space, which itself can be exponential in the size of the state or goals.

Different approaches have been proposed to efficiently find nonoptimal solutions to SSPs based on replanning. Replanners do not invest the computational effort to generate a closed policy, and instead compute an *open policy*, i.e., a policy that does not address all the probabilistically reachable states. Different methods can be employed to generate open policies, e.g., determinization [30,31], sampling [11,26], and finite horizon search [21,17]. During the execution, if a state not included in the open policy is reached, the replanner is reinvoked to compute a new open policy starting from the unpredicted state.

In this work, we introduce a new model, the depth-based short-sighted Stochastic Shortest Path Problems (short-sighted SSPs), a special case of SSPs in which every state has a nonzero probability of being reached using at most t actions. We also introduce the novel algorithm Short-Sighted Probabilistic Planner (SSiPP), which solves SSPs using short-sighted SSPs to represent subproblems of the original problem. We prove that the policies computed by SSiPP can be executed for at least t time steps without replanning; therefore, by varying the parameter t of the short-sighted SSPs, SSiPP can behave as either a probabilistic planner or a replanner: for small values of t, the SSiPP returns open policies and less replanning is necessary as t increases; and for t large enough, SSiPP returns closed policies. We provide an upper bound for t in which SSiPP is guaranteed to return closed policies.

We also present two extensions of SSiPP: Labeled-SSiPP and SSiPP-FF. Labeled-SSiPP improves the performance of SSiPP when searching for the optimal solution of SSPs by not revisiting states that have already converged. SSiPP-FF combines SSiPP and determinization to improve the efficiency of SSiPP when searching for a suboptimal solution under *small* time constraints, e.g., the International Probabilistic Planning Competition (IPPC) [33,7,8] rules.

Lastly, we extensively compare SSiPP, Labeled-SSiPP, and SSiPP-FF in different domains against the state-of-the-art probabilistic planners. Our results show that in the task of finding the optimal solution for an SSP, Labeled-SSiPP represents an improvement of SSiPP, and Labeled-SSiPP outperforms the other considered planners when the optimal policy encompasses a small fraction of the state space. For the task of quickly finding a suboptimal solution to an SSP, our results indicate that SSiPP-FF successfully combines the behavior of SSiPP and FF-Replan: for problems without dead ends, SSiPP-FF performance is similar to FF-Replan performance (the best planner for problems without dead ends); and for problems with dead ends, SSiPP-FF performs better than FF-Replan and similarly to SSiPP. Moreover, the solutions obtained by SSiPP-FF in problems without dead ends have better quality, i.e., lower average cost, than the solutions obtained by FF-Replan.

This article is organized as follows: Section 2 reviews the basic background on SSPs and the related work. Section 3 defines formally our novel model, the depth-based short-sighted SSPs, as well as its properties. Section 4 presents our main algorithms, namely SSiPP, Labeled-SSiPP, and SSiPP-FF, and their theoretical guarantees. Section 5 empirically evaluates SSiPP (and its extensions) against the state-of-the-art planners in two settings: search for the optimal solution (Section 5.2) and search for a solution using the IPPC rules (Section 5.3). Section 6 concludes the article.

2. Background and related work

This section introduces the basic concepts and notation used in this article (Section 2.1) and reviews the related work in probabilistic planning (Section 2.2).

2.1. Stochastic shortest path problem

A Stochastic Shortest Path Problem (SSP) [3] is a tuple $\mathbb{S} = \langle \mathsf{S}, s_0, \mathsf{G}, \mathsf{A}, P, C\rangle$, in which

- S is the finite set of states;
- $s_0 \in \mathsf{S}$ is the initial state;
- $\mathsf{G} \subseteq \mathsf{S}$ is the nonempty set of goal states;
- A is the finite set of actions;
- $P(s'|s,a)$ represents the probability that $s' \in \mathsf{S}$ is reached after applying action $a \in \mathsf{A}$ in state $s \in \mathsf{S}$; and
- $C(s,a,s') \in (0,+\infty)$ is the immediate cost incurred when state s' is reached after applying action a in state s. This function is required to be defined for all s, a, and s' in which $P(s'|s,a) > 0$.

In SSPs, an agent executes actions $a \in \mathsf{A}$ in discrete time steps at a state $s \in \mathsf{S}$. The chosen action a changes state s to state s' with probability $P(s'|s,a)$ and the cost $C(s,a,s')$ is incurred. If a goal state $s_G \in \mathsf{G}$ is reached, the problem finishes, i.e., no more actions need to be executed. The sequence of states $\mathcal{T} = \langle s_0, s_1, s_2, \ldots\rangle$ visited by the agent is called a trajectory,

Download English Version:

https://daneshyari.com/en/article/376879

Download Persian Version:

https://daneshyari.com/article/376879

Daneshyari.com