AST 2015
Automated Systems & Technologies
25-26 May 2015 • St. Petersburg, Russia

DISTRIBUTED TREE SEARCH ALGORITHMS FOR POMDPS

Sergey Zobnin, Evgeniya Sheremetova, Vyacheslav Shkodyrev

Control Systems and Technologies Department
Peter the Great St. Petersburg Polytechnic University
Grazhdansky Pr., 28
195220, St. Petersburg, Russia
e-mail address of lead author: e.g. sergei.s.zobnin@gmail.com

Abstract

In the era of distributed computing, algorithms and technologies, traditional decision-making research communities still focus mostly on speeding up algorithms via non-trivial optimizations on a single machine rather than speeding up via scaling-out. In this paper, we make a step towards design and implementation of highly scalable decision-making systems based on well-formalized decision-making frameworks with strong mathematical background. One of such frameworks that we consider, Partially Observable Markov Decision Processes (POMDPs), provides capabilities of handling multiple types of uncertainties at the same time: uncertainty of observations and uncertainty of actions. Model expressiveness comes at a high cost: POMDPs are computationally complex and this circumstance becomes a serious obstacle to production usage of POMDPs despite of their features. We focus on design and implementation of distributed tree search algorithms for solving POMDPs using modern distributed computing framework, Apache Spark, that provides a highly performant implementation of a Map-Reduce paradigm. We provide a set of map-reduce algorithms that are capable of solving large POMDP problems in a highly scalable fashion, which results in small execution times given enough computational power. We perform research of scalability characteristics of devised algorithms via theoretical analysis and experiments on a distributed installation of Apache Spark. We

analyze convergence of created algorithms via comparison to results of centralized algorithms execution. Our experimental results show almost linear scalability of our algorithms.

INTRODUCTION

Decision making techniques become widely used in different domains. Various approaches and models of performing decision exist. Decision inference is a computationally complex process, which can exploit scalability to improve execution characteristics of algorithms. The main contribution of this paper is a development of general distributed algorithm for solving POMDP problems. The paper is structured as follows: a short overview of POMDP is given and then a novel approach of solving POMDPs in a distributed manner is presented.

POMDP MODEL AND TRADITIONAL SOLUTION TECHNIQUES

Core notation

POMDPs are a proven technique of making value-oriented decisions in presence of observation and action uncertainty. POMDP models a process of making decisions under two types of uncertainties (action uncertainties and observation uncertainties) via utility-based approach (rewards are given for being in particular states of the system) over possibly infinite event horizons.

POMDP is defined as (S, A, Z, T, O, R), where

- S is a set of states,
- A is a set of actions,
- Z is a set of observations,
- T is transition function,
- O is an observation function,
- R is the reward function.

POMDP usage is beneficial because of model expressiveness, which is enough to model problems of different domains (POMDP applications list includes many items starting from UAV control [1] and ending with spoken dialog management system [2]).

Value function

Value function is a mathematical equation that bounds POMDP model entities into single decision making framework.

$$V^*(b) = \max_{a \in A}\left[R(b,a) + \gamma \sum_{z \in Z} \mathrm{P}(z|b,a) V^*\big(\tau(b,a,o)\big)\right] \quad (1)$$

where γ is the discount factor, τ is the belief state transition function, b is a belief state, a probability distribution over all possible system states. $\mathrm{P}(z|b,a)$ is a joint conditional probability distribution over observations and actions, which is defined as follows:

$$\mathrm{P}(z|b,a) = \sum_{s' \in S} \mathrm{O}(\mathrm{s}', \mathrm{a}, \mathrm{z}) * \sum_{s \in S} T(\mathrm{s}, \mathrm{a}, \mathrm{s}')\mathrm{b}(\mathrm{s}) \quad (2)$$

The value function in form (1) defines a decision making process over an infinite horizon, implicitly bound by the discount factor which makes further decisions less important. See [3] for a comprehensive description of the model.

Solution techniques

There are two main approaches of solving POMDP: an offline and online approach. Offline algorithms assume separation of planning and execution phase. In the planning phase dynamic programing procedures are executed to compute the value function (1) for all possible belief states. In execution phase precomputed values are used to infer the decisions. Example of such algorithm is [4]. Online algorithms allow combination of planning and execution phase via transformation of dynamic programming algorithms into tree based algorithms. A tree search procedures, with some modifications, are performed and best action may be chose in any moment. For online algorithms reference see [3].

TOWARDS MAP-REDUCE ALGORITHMS FOR POMDPS

Standard online solution techniques for POMDPs are not suitable for Map-reduce approach due to requirements of a shared data structure usage, which greatly reduces scalability capabilities. We developed an approach that reduces number of required synchronisations that increases parallelism of an algorithm.

The devised approach introduces a modification of a standard tree-expansion procedure of online algorithm. The modification applies distributed-system approach of state propagation to minimise required synchronization via additional memory usage. In order to define a basis and correctness of approach a value function for several steps ahead should be considered.

$$\begin{aligned} V^*(b) = \max_{a\in A}\Bigg[R(b,a)(\ +\gamma \sum_{z\in Z} \mathrm{P}(z|b,a) \\ * \max_{a\in A}\Bigg[R(b',a) +\ \gamma \sum_{z\in Z} \mathrm{P}(z|b',a)V^*(b'')\Bigg]\Bigg] \end{aligned} \tag{3}$$

In (3) the value function for two steps ahead case is given. Traditional online algorithms assume tree expansion with afterwards backpropagation of nodes' values to the root.

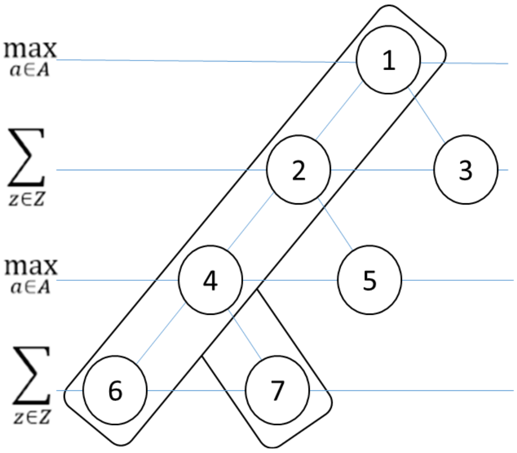
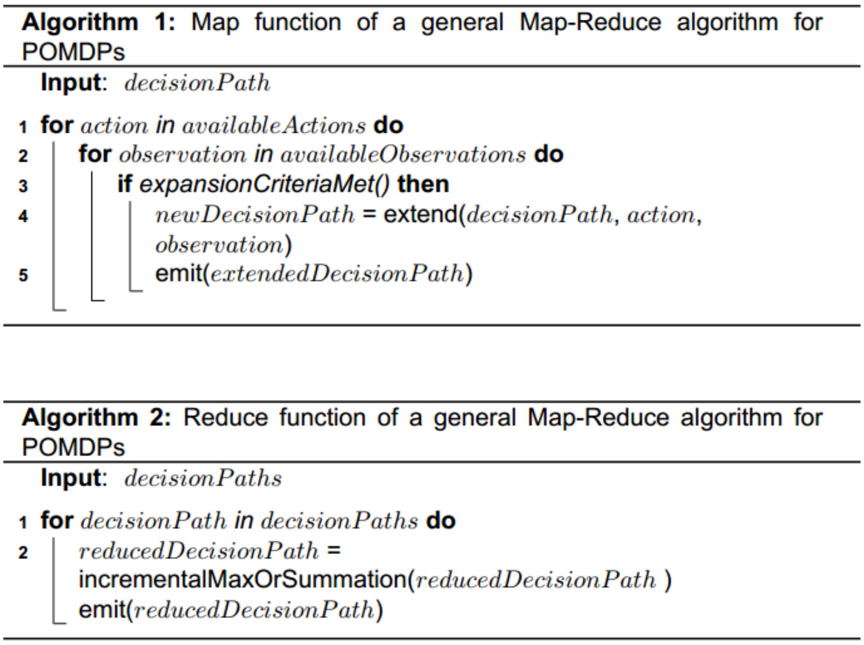


Figure 1. Expansion of decision trees and values backpropagation.

The backpropagation procedure, stated in Figure 1, is a main problem of algorithms scalability. When implemented based on a tree-like structure, the backpropagation procedure requires synchronization both on maximization and summation phases. An alternative approach that we propose is to convert the standard tree-based backpropagation procedure into multiple reduction phases of a distributed algorithm in map-reduce.

We define an additional notation to allow highly distributed algorithms for POMDPS. Let $decisionPath$ be a path from the POMDP decision root to a particular node with associated state. Each decision path contains a set of decisions on actions and received observations with associated values. We define a generalized algorithm for distributed solution of POMDPs, the algorithm is presented in a Figure 2.



Algorithm 1: Map function of a general Map-Reduce algorithm for POMDPs

```
Input: $decisionPath$
for $action$ in $availableActions$ do
    for $observation$ in $availableObservations$ do
        if expansionCriteriaMet() then
            $newDecisionPath$ = extend($decisionPath$, $action$, $observation$)
            emit($extendedDecisionPath$)
```

Algorithm 2: Reduce function of a general Map-Reduce algorithm for POMDPs

```
Input: $decisionPaths$
for $decisionPath$ in $decisionPaths$ do
    $reducedDecisionPath$ = incrementalMaxOrSummation($reducedDecisionPath$ )
    emit($reducedDecisionPath$)
```

Figure 2. General distributed algorithm for POMDPs.

In Map-Reduce two functions should be provided to organize distributed computations. Figure 2 presents both Map and Reduce function for distributed tree-based algorithms for solving POMDPs.

FURTHER WORK

A set of distributed algorithms will be developed based on a developed framework for distributed POMDP algorithms. The following approach modifications will be considered and evaluated.

- Usage of heuristic function in the map phase that will reduce number of emitted: far not all decision paths should be extended. Extension of only most promising paths will be performed and that will decrease RAM usage
- Usage of sampling techniques inside a map function to perform state exploration in a direction of most likely states. This technique will further reduce RAM usage
- Usage of state space clustering techniques that will allow belief space reduction: instead of performing new expansions for particular combinations of actions and observation it may make sense to reuse existing tree structure

- In order to reduce RAM consumption it may make sense not to forward all decision path information forward in map phase, rather than drop part of information in favour of additional computations. In this way only a subset of decision path will be kept in memory

SUMMARY (or CONCLUSIONS)

In this paper a novel scheme of POMDP solution is presented. Correctness of the general approach was verified via trivial POMDP solving algorithm that were developed on top of this approach. However these algorithms are far from production readiness and additional techniques, presented on further work, need to be incorporated to achieve good performance as well as scalability, gained by our approach.

REFERENCES

[1] Miller, Scott A., Zachary A. Harris, and Edwin KP Chong. "A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking." EURASIP Journal on Advances in Signal Processing 2009 (2009): 2.

[2] Young, Steve. "Using POMDPs for Dialog Management." SLT. 2006.

[3] Ross, Stéphane, et al. "Online planning algorithms for POMDPs." Journal of Artificial Intelligence Research (2008): 663-704.

[4] Pineau, Joelle, Geoff Gordon, and Sebastian Thrun. "Point-based value iteration: An anytime algorithm for POMDPs." IJCAI. Vol. 3. 2003.