# Representing and Solving Factored Markov Decision Processes with Imprecise Probabilities


Karina V. Delgado
kvd@ime.usp.br

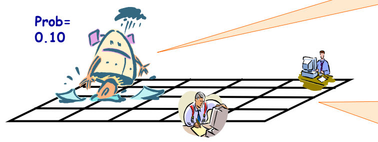
Leliane N. de Barros
leliane@ime.usp.br

Fabio G. Cozman
fgcozman@usp.br

Ricardo Shirota Filho
ricardo.shirota@poli.usp.br

IME - USP


The main approach to solve a probabilistic planning problem is modeling it as a Markov decision process (MDP).





**Abstract:** This paper investigates ***Factored Markov Decision Processes with imprecise Probabilities***: Markov Decision Processes where transition probabilities are imprecisely specified, and where their specifications does not deal directly with states, but rather with factored representations of states. We first define a Factored MDPIP, based on a multilinear formulation for MDPIPs; then we propose a novel approximated solution to generate maximin policies for Factored MDPIPs. We show that with this new algorithm we can solve relatively large MDPIP problems.


# Background

## MDP and a Factored Representation

MDP is defined by: $M = \langle T, S, A, R, P \rangle$.

(stages, states, transition probabilities, actions, rewards)

Let $V^*(s)$ be the optimal value of the state $s \in S$, for an agent that wants to maximize his expected reward. **The Bellman Optimality equation is:**

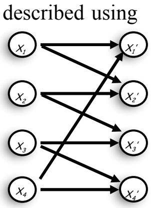
$$V^*(s) = \max_{a \in A} \{R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s')\}$$

modeled as a linear programming problem

$$\min_{V^*} : \sum_s V^*(s)$$
$$\text{s.t.} : V^*(s) \geq R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s'), \ \forall s \in S, a \in A$$

In a factored MDP, the state $\vec{x}$ are described using state variables $X_i$ for i=1..n and...

$$P(\vec{x}'|\vec{x},a) = \prod_{i=1}^{n} P(x'_i | pa(X'_i), a)$$




$$\hat{V}(\vec{x}) = \sum_{j=1}^{k} w_j h_j(\vec{x})$$

Approximated Value function

Basis functions

modeled as a linear programming problem:

$$\min_{V^*} : \sum_{\vec{x}} \sum_{i=1}^{k} w_i h_i(\vec{x})$$
$$\text{s.t.} : \sum_{i=1}^{k} w_i h_i(\vec{x}) \geq R(\vec{x},a) + \gamma \sum_{\vec{x}' \in S} P(\vec{x}'|\vec{x},a) \sum_{i=1}^{k} w_i h_i(\vec{x}'), \forall \vec{x} \in S, a \in A$$

Possible ways to solve it:

- Generating a smaller set of equivalent constraints: FactoredLPA algorithm (Guestrin 2003)
- Using general techniques to solve linear problems with large number of constraints.

## MDPIP

MDPIP is defined by : $M = \langle T, S, A, R, K \rangle$.
The credal conditional sets $K_a(s'|s)$, represented by linear inequations, to express all possible probability distributions.

**The Bellman optimality equation** for MDPIPs adopting the maximin criterion is:

$$V^*(s) = \max_{a \in A} \min_{P(s'|s,a) \in K_a(s'|s)} \{R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s')\}$$

modeled as a multilinear programming problem

Shirota, R.;et.al 2007

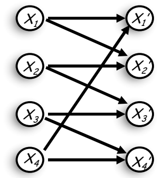
$$\min_{V^*, P} : \sum_s V^*(s)$$
$$\text{s.t.} : V^*(s) \geq R(s,a) + \gamma \sum_{s' \in S} P(s'|s,a) V^*(s'), \ \forall s \in S, a \in A$$
$$P(s'|s,a) \in K_a(s'|s)$$

# Factored MDPIP

## Definition and representation

In a factored MDPIP, states are defined using state variables $X_i$ for i = 1..n and the transitions are represented by Dynamic Credal Networks (DNC). The value and reward functions are factored as for a factored MDP.

A DCN has also two layers: one representing the actual state and other the next state. The DCN represents a joint credal set where each distribution satisfies the following expression:

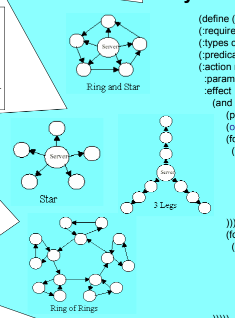



$$P(\vec{x}'|\vec{x},\vec{p},a) = \prod_{i=1}^{n} P(x'_i | pa(X'_i), \vec{p}, a)$$

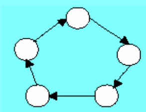
modeled as a multilinear programming problem

$$\min_{V^*, P} : \sum_{\vec{x}} \sum_{i=1}^{k} w_i h_i(\vec{x})$$
$$\text{s.t.} : \sum_{i=1}^{k} w_i h_i(\vec{x}) \geq R(\vec{x},a) + \gamma \sum_{\vec{x}' \in S} P(\vec{x}'|\vec{x},\vec{p},a) \sum_{i=1}^{k} w_i h_i(\vec{x}'),$$
$$\forall \vec{x} \in S, a \in A$$
$$P(\vec{x}'|\vec{x},\vec{p},a) \in K_a(\vec{x}'|\vec{x},\vec{p})$$
$$where \ : P(\vec{x}'|\vec{x},\vec{p},a) = \prod_{i=1}^{n} P(x'_i | pa(X'_i), \vec{p}, a)$$

**System Administrator Problem**




```
(define (sysadmin – unidirectional ring)
(:requirements :adl)
(:types comp)
(:predicates (up ?c)(conn ?c ?d))
(:action reboot
 :parameters (?x - comp)
 :effect
   (and (decrease (reward) 1)
      (probabilistic 0.9 (up ?x))
      (oneof
      (forall (?d - comp)
        (probabilistic
          0.6 (when (exists (?c - comp)
                (and (conn ?c ?d)
                  (not (up ?c))
                  (not (= ?x ?d))))
              (not (up ?d))
      )))
      (forall (?d - comp)
        (probabilistic
          0.8 (when (exists (?c - comp)
                (and (conn ?c ?d)
                  (not (up ?c))
                  (not (= ?x ?d))))
              (not (up ?d))
   )))))
)
```



Unidirectional Ring

## FactoredMPA: Solving a Factored MDPIP

Shortly, FactoredMPA:

1. simplifies the computation of each constraint applying the same Backprojection algorithm used by Guestrin for factored MDP.
2. calls the FactoredMP algorithm, an adaptation of the FactoredLP from Guestrin, to create a new and smaller equivalent set of constraints for the Multilinear Program.
3. in order to obtain $w_i$ and $\vec{p}$, it calls a nonlinear solver with the new equivalent problem.

# Results

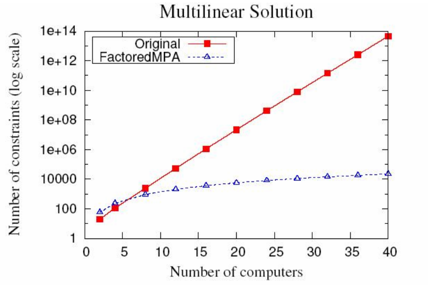



Figure 1: Comparing the number of constraints for the System Administrator domain (unidirectional ring) using the original constraints set Vs using the FactoredMPA generation of smaller equivalent constraints set. Problems with n computers implies in $2^n$ states.

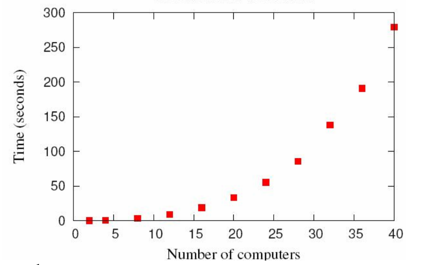



Figure 2: The number of constraints (Figure 1) and the CPU-time grows quadratically with the number of variables. Therefore we could solve MDPIP problems up to $2^{40}$ states.

## Conclusion

We have defined a Factored MDPIP problem based on Factored MDPs and an MDPIP multilinear formulation. Plus, we have developed a new algorithm, named FactoredMPA, to solve factored MDPIPs. This is the first solution in the literature to solve large MDPIP problems. Our experiments show that: (i) by exploiting the factored representation of an MDPIP problem, (ii) by making the assumption of a restrict scope for variable dependences, and (iii) by allowing approximated solutions, we can solve relatively large problems (up to $2^{40}$ states). We also showed that we can specify Factored MDPIPs in a first order language, an extension of PPDDL (*Probabilistic Planning Domain Definition Language*).

## References


•Cozman, F. G. 2000. Credal networks. AI 120:199–233.
•Guestrin, C. E. 2003. Planning under uncertainty in complex structured environments. Ph.D. Dissertation, Stanford University. Adviser-Daphne Koller.
•Shirota, R.;et.al 2007. Multilinear and integer programming for Markov decision processes with imprecise probabilities. In 5th ISIPTA, 395–404.
•Satia, J.K. and Lave, R.E. Markovian decision processes with uncertain transition probabilities. Operations Research, 21:728-740,1970.
•White III, C. C., and Eldeib, H. K. 1994. Markov decision processes with imprecise transition probabilities. Oper. Res. 42(4):739–749.



Universidade de São Paulo
Brasil
FAPESP
CNPq
CAPES