A Markovian Model for Dynamic and Constrained Resource Allocation Problems

Camille Besse & Brahim Chaib-draa
Department of Computer Science and Software Engineering
Laval University, G1K 7P4, Quebec (Qc), Canada
{besse,chaib}@damas.ift.ilaval.ca
Full Paper: http://macsp.cbesse.net

Introduction

An autonomous agent, allocating stochastic resources to incoming tasks, faces increasingly complex situations when formulating its control policy. These situations are often constrained by limited resources of the agent, time limits, physical constraints or other agents. All these reasons explain why complexity and state space dimension increase exponentially in size of considered problem. Unfortunately, models that already exist either consider the sequential aspect of the environment, or its stochastic one or its constrained one. To the best of our knowledge, there is no model that take into account all these three aspects.

For example, dynamic constraint satisfaction problems (DCSP) have been introduced by Dechter & Dechter (1988) to address dynamic and constrained problems. However, in DCSPs, there is typically no transition model, and thus no concept of sequence of controls. On the other hand, Fargier, Lang, & Schiex (1996) proposed mixed CSPs (MCSPs), but this approach considers only the stochastic and the constrained aspects of the problem.

In this paper, we introduce a new model based on DCSPs and Markov decision processes to address constrained stochastic resource allocation (SRA) problems by using expressiveness and powerfulness of CSPs. We thus propose a framework which aims to model dynamic and stochastic environments for constrained resources allocation decisions and present some complexity and experimental results.

Background

A classical *constraint satisfaction problem* (CSP) is a triple $P = \langle \mathcal{X}, \mathcal{D}, \mathcal{C} \rangle$, where $\mathcal{X}$ is the set of *variables*, each of them can take its possible values in a *domain* $\mathcal{D}$ (supposed here finite), and $\mathcal{C}$ is a set of *constraints* restricting the possible values of some variables. We will denote by $\mathbb{S}(P) \subset \mathcal{D}^{|\mathcal{X}|}$ the set of all assignments satisfying all constraints $\mathcal{C}$ of P. Solving a CSP is equivalent to finding one element $\sigma \in \mathbb{S}(P)$.

According to Verfaillie & Jussien (2005) dynamic constraint satisfaction problem (DCSP) is a sequence of η CSPs P, each depending only on changes in the definition of the previous one.

Solving a DCSP consists in finding a satisfying solution $\sigma_i \in \mathbb{S}(P_i)$ for each i, $1 \leq i \leq \eta$. Most past work on DCSPs has been devoted to finding a solution σ_{i+1} based on σ_i. For instance, solution reuse and reasoning reuse (Verfaillie & Jussien, 2005) are two approaches that benefit from previous solutions to produce future ones. However, none of these approaches attempt to formalize the influence of past assignments on future ones as it can be in SRA problems.

Hence, dynamic CSPs are inadequate to address SRA problems. We thus present Markov decision processes which are a well known approach to model these influence before proposing a new framework based on the composition of these two approaches.

A Markov Decision Process (MDP) models a planning problem in which action outcomes are stochastic but the world state is fully observable. The agent is assumed to know a probability distribution for action outcomes. Formally, a MDP is a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$ where $\mathcal{S}$ and $\mathcal{A}$ are finite set of states and actions, $\mathcal{T}^a_{ss'} = Pr(s'|s,a)$ is the transition function and $\mathcal{R}^a_s \in \mathbb{R}$ is the reward function.

Solving an MDP aims to find an optimal policy $\pi^* : \mathcal{S} \mapsto \mathcal{A}$ that associates to each state $s \in \mathcal{S}$ an action $a \in \mathcal{A}$ that maximizes the expected reward accumulated.

Papadimitriou & Tsisiklis (1987) have shown that finding an optimal policy is one of the most difficult polynomial problems under some restrictions recalled later. Nevertheless, as entries can have an exponential size, this problem is one of the current major difficulties in the planning community. In fact, two main factors are responsible for this tractability problem: the exponential size of the state space and the branching factor between each state which makes search barely feasible. Indeed, many methods already exist that address the state space problem. For example, aggregation of Dearden & Boutilier (1997) and real-time exploration (e.g. FRTDP algorithm of Smith & Simmons (2006)). This is why this paper aims to cover the branching factor problem instead, by using constraints on actions that will thus limit state space explosion.

Markovian Constraint Satisfaction Problem

A markovian CSP (MaCSP) is a Markov Decision Process which describes the stochastic evolution of a dynamic constraint satisfaction problem. States represent possible configurations of the DCSP among its evolution, and actions

represent assignments of each configuration of this DCSP. Thus, in a MaCSP, the Markov property is satisfied such as a future configuration depends only on the previous configuration and the assignment chosen. Formally, a MaCSP is a tuple $\Phi = \langle \mathcal{S}, \Psi, \{\mathcal{A}\}, \mathcal{T}, \mathcal{R}, s_0, \mathfrak{G} \rangle$ where $\mathcal{S}$ is the state space of the underlying DCSP Ψ, $\mathcal{A}_i = \{\sigma_{s_i}, \sigma_{s_i} \in \mathbb{S}(P_{s_i})\}$ is the set of assignments of variables in each state $s_i, 1 \leq i \leq \eta$, $\mathcal{T}^{\sigma}_{s_i s_{i+1}} = Pr(s_{i+1}|s_i, \sigma)$ and $\mathcal{R}^{\sigma}_{s} \in \mathbb{R}$ are the transition and the reward functions as in MDPs.

In fact, a DCSP is a particular case of MaCSP where transition model $\mathcal{T}^{\sigma}_{s_i s_{i+1}}$ is not specified. A DCSP can then be naturally defined as a MaCSP in which transition model is deterministic whatever assignment chosen and reward function is $\{satisfied, unsatisfied\}$.

Solving a MaCSP consists in finding a consistant assignment σ_{s_i} for each state s_i defining a policy $\xi^* : \mathcal{S} \mapsto \mathcal{A}$ that associates for each state $s_i \in \mathcal{S}$ an assignment $\sigma_{s_i} \in \mathbb{S}(P_{s_i})$ over the finite horizon η. ξ^* is the optimal policy over the set of all policies Ξ such as:

$$\xi^* = \sup_{\xi \in \Xi} \left[\sum_{\sigma \in \mathcal{A}} \xi(s, \sigma) \sum_{s' \in \mathcal{S}} \mathcal{T}^{\sigma}_{ss'} \left[\mathcal{R}^{\sigma}_{s} + \gamma V_{\xi}(s') \right] \right] \quad (1)$$

Complexity

The complexity class of satisfying a dynamic CSP is easy to show since a dynamic CSP is a linear combination of CSPs in terms of complexity: As stated by Haralick *et al.* (1978), the problem of satisfiability of a constraint network as defined in previous section is NP-complete and thus the Dynamic Constraint Satisfaction Problem.

Furthermore, Papadimitriou & Tsisiklis (1987) have shown that, given an MDP $\mathcal{M}$, a horizon T, and an integer K, the problem of computing a policy in $\mathcal{M}$ under horizon T that yields total reward at least K is P-complete.

Thus, as stated for MDPs, it is necessary to place some restrictions in order to hold the upper bounds. First, $\eta \ll |\mathcal{S}|$ and $|\Psi| \ll |\mathcal{S}|$. Then, we also assume that tables for the transition and the reward function can be represented with a constant number of bits. Under these restrictions:

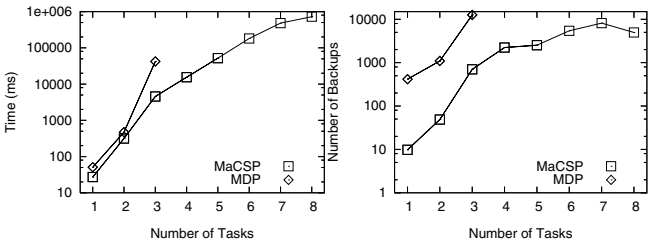
Theorem 1. *Given an MaCSP Φ, a horizon η, and an integer K, the problem of computing a policy in Φ under horizon η that yields total reward at least K is* NP*-complete.*

Proof. See full paper. □

Experimentations

A typical example of class of SRA problems addressed here is the Dynamic Weapon-Target Allocation (DWTA) problem described by Hosein, Athans, & Walton (1988) which have been shown to be NP-complete even in the static unconstrained case.

We have tested an algorithm based on FRTDP of Smith & Simmons (2006) where each state action space was first filtered by a constraint checking. The DWTA problem we consider was a naval defense problem with unreliable and constrained resources (see full paper for details). Comparison between MaCSP and standard MDP (figure 1) shows that MaCSP correctly adresses the branching factor.



(a) Time (ms) vs # Tasks (b) # Backups vs # Tasks

Figure 1: Performance comparison

Further Work

There are mainly two future research avenues for further work. First, factored MDP of Boutilier, Dearden, & Goldszmidt (2000) is a very well suited framework to apply MaCSP, with constraints between state variables, allowing also to prune non consistent states if *a priori* knowledge about environment is available. Second, Graphical Games Theory of Kearns, Littman, & Singh (2001) where constraints were recently applied to solve graphical games, so this framework could lead to stochastic graphical games with few efforts.

References

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 2000. Stochastic dynamic programming with factored representations. *Artificial Intelligence* 121(1-2):49–107.

Dearden, R., and Boutilier, C. 1997. Abstraction and Approximate Decision-Theoretic Planning. *Artificial Intelligence* 89(1–2):219–283.

Dechter, R., and Dechter, A. 1988. Belief Maintenance in Dynamic Constraint Networks. In *AAAI*, 37–42.

Fargier, H.; Lang, J.; and Schiex, T. 1996. Mixed Constraint Satisfaction: A Framework for Decision Problems under Incomplete Knowledge. In *AAAI/IAAI, Vol. 1*, 175–180.

Haralick, R. M.; Davis, L. S.; Rosenfeld, A.; and Milgram, D. L. 1978. Reduction Operations for Constraint Satisfaction. *Information Sciences* 14(3):199–219.

Hosein, P.; Athans, M.; and Walton, J. 1988. Dynamic Weapon-Target Assignment Problems with Vulnerable C2 nodes. In *Proceedings of the 1988 Command and Control Symposium*, 240–245.

Kearns, M. J.; Littman, M. L.; and Singh, S. P. 2001. Graphical models for game theory. In *UAI*, 253–260.

Papadimitriou, C., and Tsisiklis, J. N. 1987. The Complexity of Markov Decision Processes. *Math. Oper. Res.* 12(3):441–450.

Smith, T., and Simmons, R. G. 2006. Focused Real-Time Dynamic Programming for MDPs: Squeezing More Out of a Heuristic. In *AAAI/IAAI*.

Verfaillie, G., and Jussien, N. 2005. Constraint Solving in Uncertain and Dynamic Environments: A Survey. *Constraints* 10(3):253–281.