Solving Transition-Independent Multi-agent MDPs with Sparse Interactions

Joris Scharpff
Delft University of Technology, The Netherlands

Diederik M. Roijers
University of Amsterdam, The Netherlands

Frans A. Oliehoek
University of Amsterdam, The Netherlands
University of Liverpool, United Kingdom

Matthijs T. J. Spaan and **Mathijs M. de Weerdt**
Delft University of Technology, The Netherlands

Abstract

In cooperative multi-agent sequential decision making under uncertainty, agents must coordinate to find an optimal joint policy that maximises joint value. Typical algorithms exploit additive structure in the value function, but in the fully-observable multi-agent MDP (MMDP) setting such structure is not present. We propose a new optimal solver for transition-independent MMDPs, in which agents can only affect their own state but their reward depends on joint transitions. We represent these dependencies compactly in *conditional return graphs (CRGs)*. Using CRGs the value of a joint policy and the bounds on partially specified joint policies can be efficiently computed. We propose CoRe, a novel branch-and-bound policy search algorithm building on CRGs. CoRe typically requires less runtime than available alternatives and finds solutions to previously unsolvable problems.

Introduction

When cooperative teams of agents are planning in uncertain domains, they must coordinate to maximise their (joint) team value. In several problem domains, such as traffic light control (Bakker et al. 2010), system monitoring (Guestrin, Koller, and Parr 2002), multi-robot planning (Messias, Spaan, and Lima 2013) or maintenance planning (Scharpff et al. 2013), the full state of the environment is assumed to be known to each agent. Such *centralised* planning problems can be formalised as multi-agent Markov decision processes (MMDPs) (Boutilier 1996), in which the availability of complete and perfect information leads to highly-coordinated policies. However, these models suffer from exponential joint action spaces as well as a state that is typically exponential in the number of agents. In this paper, we identify a significant MMDP sub-class whose structure we compactly represent and exploit via locally-computed upper and lower bounds on the optimal policy value.

In problem domains with local observations, sub-classes of *decentralised* models exist that admit a value function that is exactly factored into additive components (Becker et al. 2003; Nair et al. 2005; Witwicki and Durfee 2010) and more general classes admit upper bounds on the value function that are factored (Oliehoek, Spaan, and Witwicki 2015). In centralised models however, the possibility of a factored value function can be ruled out in general: by observing the full state, agents can predict the actions of others better than when only observing a local state. This directly implies that the value function depends on the full state.

A class of problems that exhibits particular structure is that of task-based planning problems, such as the *maintenance planning problem* (MPP) from (Scharpff et al. 2013). In the MPP every agent needs to plan and complete its own set of road maintenance tasks at minimal (private) maintenance cost. Each task is performed only once and may delay with a known probability. As maintenance causes disruption to traffic, agents are collectively fined relative to the (super-additive) hindrance from their *joint actions*. Although agents plan autonomously, they depend on others via these fines and must therefore coordinate. Still, such *reward interactions* are typically sparse: they apply only to certain combinations of maintenance tasks, e.g., in the same area, and often involve only a few agents. Moreover, when an agent has performed its maintenance tasks that potentially interfere with others, it will no longer interact with any of the other agents.

Our main goal is to identify and exploit such structure in centralised models, for which we consider *transition independent* MMDPs (TI-MMDPs). In TI-MMDPS, agent rewards depend on joint states and actions, but transition probabilities are individual. Our key insight is that we can exploit the reward structure of TI-MMDPs by decomposing the *returns* of all execution histories (i.e., all possible state/action sequences from the initial time step to the planning horizon) into components that depend on local states and actions.

We build on three key observations. 1) Contrary to the optimal value function, returns *can* be decomposed without loss of optimality, as they depend only on local states and actions of execution sequences. This allows for a compact representation of rewards and efficiently computable bounds on the optimal policy value via a data structure we call the *conditional return graph* (CRG). 2) In TI-MMDPs agent interactions are often sparse and/or local, for instance in the domains mentioned before, typically resulting in very compact CRGs. 3) In many (e.g., task-modelling) problems the state space is transient, i.e., states can only be visited once, leading to a directed, acyclic transition graph. With our first two key observations this often gives rise to *conditional reward independence*, i.e. the absence of further reward interactions, and enables agent decoupling during policy search.

Here we propose *conditional return policy search* (CoRe), a branch-and-bound policy search algorithm for TI-MMDPs employing CRGs, and show that it is effective when reward interactions between agents are sparse. We evaluate CoRe on instances of the aforementioned MPP with uncertain outcomes and very large state spaces. We demonstrate that CoRe evaluates only a fraction of the policy search space and thus finds optimal policies for previously unsolvable instances and requires less runtime than its alternatives.

Related work

Scalability is a major challenge in multi-agent planning under uncertainty. In response, two important lines of work have been developed. One line proposed approximate solutions by imposing and exploiting an additive structure in the value function (Guestrin, Koller, and Parr 2002). This approach has been applied in a range of stochastic planning settings, fully and partially observable alike, both single-agent (Koller and Parr 1999; Parr 1998) and multi-agent (Guestrin, Venkataraman, and Koller 2002; Kok and Vlassis 2004; Oliehoek, Whiteson, and Spaan 2013). The drawback of such methods is that typically no bounds on the efficiency loss can be given. We focus on optimal solutions, required when dealing with strategic behaviour in a mechanism (Cavallo, Parkes, and Singh 2006; Scharpff et al. 2013).

This is part of another line of work that has not sacrificed optimality, but instead targets problem sub-classes with exploitable structure (Becker et al. 2003; Becker, Zilberstein, and Lesser 2004; Mostafa and Lesser 2009; Witwicki and Durfee 2010). In particular, several methods that similarly exploit additive structure in the value function have been shown exact, simply because the value functions of these sub-classes are guaranteed to have such shape (Nair et al. 2005; Oliehoek et al. 2008; Varakantham et al. 2007). However, all these approaches are for decentralised models in which actions are conditioned only on *local* observations. Consequentially, optimal policies for decentralised models typically yield lower value than the optimal policies for their fully-observable counterparts (shown in our experiments).

Our focus is on transition-independent problems, suitable for multi-agent problems in which the effects of activities of agents are (assumed) independent. In domains where agents directly influence each other, e.g., by manipulating shared state variables, this assumption is violated. Still, transition independence allows agent coordination at a task level, as in the MPP, and is both practically relevant and not uncommon in literature (Becker et al. 2003; Spaan, Gordon, and Vlassis 2006; Melo and Veloso 2011; Dibangoye et al. 2013).

Another type of interaction between agents is through limited (global) resources required for certain actions. While this introduces a global coupling, some scalability is achievable (Meuleau et al. 1998). Whether context-specific and conditional agent independence remains exploitable in the presence of such resources in TI-MMDPs is yet unclear.

Model

We consider a (fully-observable) *transition-independent, multi-agent Markov decision process (TI-MMDP)* with a finite horizon of length h, and no discounting of rewards.

Definition 1. *A TI-MMDP is a tuple* $\langle N, S, A, T, \mathcal{R}\rangle$*:*

$N = \{1, ..., n\}$ *is a set of* n *enumerated agents;*

$S = S^1 \times ... \times S^n$ *is the agent-factored state space, which is the Cartesian product of* n *factored states spaces* S^i *(composed of features* $f \in F$*, i.e.,* $s^i = \{f^i_x, f^i_y, \ldots\}$*);*

$A = A^1 \times ... \times A^n$ *is the joint action space, which is the Cartesian product of the* n *local action spaces* A^i*;*

$T(s, \vec{a}, \hat{s}) = \prod_{i \in N} T^i(s^i, a^i, \hat{s}^i)$ *defines a transition probability, which is the product of the local transition probabilities due to transition independence; and*

$\mathcal{R}$ *is the set of reward functions over transitions that we assume w.l.o.g. is structured as* $\{R^e | e \subseteq N\}$*. When* $e = \{i\}$*,* R^i *is the* local reward *function for agent* i*, and when* $|e| > 1$*,* R^e *is called an* interaction reward*. The total team reward per time step, given a joint state* s*, joint action* $\vec{a}$ *and new joint state* $\hat{s}$*, is the sum of all rewards:* $R(s, \vec{a}, \hat{s}) = \sum_{R^e \in \mathcal{R}} R^e(\{s^j\}_{j \in e}, \{\vec{a}^j\}_{j \in e}, \{\hat{s}^j\}_{j \in e})$.

Two agents i and j are called *dependent* when there exists a reward function with both agents in its scope, e.g., a two-agent reward function $R^{i,j}(\{s^i, s^j\}, \{a^i, a^j\}, \{\hat{s}^i, \hat{s}^j\})$ could describe the super-additive hindrance that results when agents in the MPP do concurrent maintenance on two nearby roads. We focus on problems with *sparse interaction rewards*, i.e., reward functions R^e with non-zero rewards for a small subset of the local joint actions (e.g., $\mathcal{A}^{ij} \subset A^i \times A^j$) or only a few agents in its scope. Of course, sparseness is not a binary property: the maximal number of actions with non-zero interaction rewards and participating agents (respectively α and w in Theorem 1) determine the level of sparsity. Note that this is not a restriction but rather a classification of problems that benefit most from our approach.

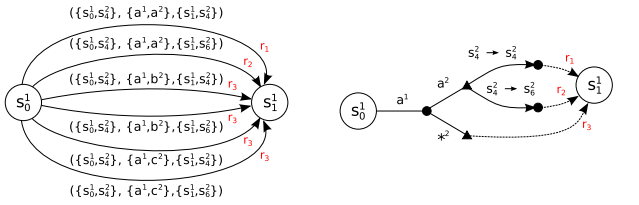
The goal in a TI-MMDP is to find the optimal joint policy π^* which actions $\vec{a}$ maximise the expected sum of rewards $V^*(s_t)$, expressed by the Bellman equation:

$$\max_{\vec{a}_t} \sum_{s_{t+1} \in S} T(s_t, \vec{a}_t, s_{t+1}) \big(\sum_{R^e \in \mathcal{R}} R^e(s^e_t, \vec{a}^e_t, s^e_{t+1}) + V^*(s_{t+1}) \big). \tag{1}$$

At the last timestep there are no future rewards, so $V^*(s_h) = 0$ for every $s_h \in S$. Although $V^*(s_t)$ can be computed through a series of maximisations over the planning period, e.g., via dynamic programming, it cannot be written as a sum of independent local value functions without losing optimality (Koller and Parr 1999).

Instead, we factor the *returns* of *execution sequences*, the sum of rewards obtained from following state/action sequences, which is optimality preserving. We denote an execution sequence up until time t as $\theta_t = [s_0, \vec{a}_0, ..., s_{t-1}, \vec{a}_{t-1}, s_t]$ and its return is the sum of its rewards: $\sum_{x=0}^{t-1} R(s_{\theta,x}, \vec{a}_{\theta,x}, s_{\theta,x+1})$, where $s_{\theta,x}$, $\vec{a}_{\theta,x}$ and $s_{\theta,x+1}$ respectively denote the state and joint action at time x, and the resulting state at time $x+1$ in this sequence. A seemingly trivial but important observation is that the return of an execution sequence can be written as the sum of local functions:

$$Z(\theta_t) = \sum_{R^e \in \mathcal{R}} \sum_{x=0}^{t-1} R^e(s^e_{\theta,x}, \vec{a}^e_{\theta,x}, s^e_{\theta,x+1}), \tag{2}$$



(a) (b)

Figure 1: Example of a transition for one agent of a two-agent problem where (a) shows the complete state/transition graph with unique rewards r_x and (b) the equivalent but more compact CRG when R_1 only depends on a_1^2.

where $s^e_{\theta,x}$, $\vec{a}^e_{\theta,x}$ and $s^e_{\theta,x+1}$ denote local states and actions from θ_t that are relevant for R^e. Contrary to the optimal value function, (2) is additive in the reward components and can thus be computed locally. To compute the expected policy value using (2), we sum the expected return $Pr(\theta_h)Z(\theta_h)$ of all future execution sequences θ_h reachable under policy π starting at s_0 (denoted $\theta_h|\pi, s_0$):

$$V^\pi(s_0) = \sum_{\theta_h|\pi,s_0} Z(\theta_h) \prod_{t=0}^{h-1} T(s_{\theta,t}, \pi(s_{\theta,t}), s_{\theta,t+1}). \quad (3)$$

Now, (3) is structured such that it expresses the value in terms of additively factored terms ($Z(\theta_h)$). However, comparing (1) and (3), we see that the price for this is that we no longer are expressing the *optimal* value function, but that of a given policy π. In fact, (3) corresponds to an equation for *policy evaluation*. It is thus not a basis for dynamic programming, but it is usable for policy search. Although policy search methods have their own problems in scaling to large problems, we show that the structure of (3) can be leveraged.

Conditional Return Graphs

We now partition the reward function into additive components $\mathcal{R}_i$ and assign them to agents. The *local* reward for an agent $i \in N$ is given by $\mathcal{R}_i = \{R^i\} \cup \mathcal{R}^e_i$, where $\mathcal{R}^e_i$ are the interaction rewards assigned to i (restricted to R^e where $i \in e$). The sets $\mathcal{R}_i$ are disjoint sub-sets of the reward functions $\mathcal{R}$. Then, a conditional return graph for agent i is a data structure that represents all possible *local* returns, for all possible *local* execution histories. Particularly, it is a directed acyclic graph (DAG) with a layer for every stage $t = 0, \ldots, h-1$ of the decision process. Each layer contains nodes corresponding to the reachable *local states* $s^i \in S^i$ of agent i at that stage. As the goal is to include interaction rewards, the CRG includes for every local state s^i, local action a^i, and successor state $\hat{s}^i$ a representation of all transitions $(s^e, \vec{a}^e, \hat{s}^e)$ for which $s^i \in s^e$, $a_i \in \vec{a}^e$, and $\hat{s}^i \in \hat{s}^e$.

While a direct representation of these transitions captures all the rewards possible, with the example DAG of Fig. 1a as a result, we can achieve a much more compact representation by exploiting sparse interaction rewards, enabling us to group many joint actions $\vec{a}^e$ leading to the same rewards. To make this explicit, we first define which actions of the neighbouring agents are important for $\mathcal{R}_i$. Given the partition of rewards $\mathcal{R}_i$ and an action a^i for an agent i, the *dependent actions* of an agent $j \neq i$ are

$$\mathcal{A}^i(a^i, j) = \{a^j \in A^j : \exists (R^e \in \mathcal{R}_i)\ \exists (s^e, \vec{a}^e, \hat{s}^e)\ a^i \in \vec{a}^e \wedge a^j \in \vec{a}^e \wedge R^e(s^e, \vec{a}^e, \hat{s}^e) \neq 0\}.$$

Actions by other agents that are not dependent are (made) anonymous in the CRG for agent i, since they do not influence the reward from the functions in R^i. A *conditional return graph* (CRG) ϕ_i for agent i is then defined as follows.

Definition 2 (Conditional Return Graph ϕ_i)**.** *Given a disjoint partitioning $\bigcup_{i \in N} \mathcal{R}_i$ of rewards, the* Conditional Return Graph (CRG) *ϕ_i is a* directed acyclic graph *with for every stage t of the decision process a node for every reachable local state s_i, and for every local transition $(s^i, a^i, \hat{s}^i)$, a tree compactly representing all transitions of the agents in scope in $\mathcal{R}_i$. The tree consists of two parts: an action tree that specifies all dependent local joint actions, and an influence tree, that contains the relevant local state transitions included in the respective joint action.*

The state s^i is connected to the root node v of an action tree by an arc labeled with action a^i. The action tree with root node v is defined recursively on the remaining $N' = N \setminus \{i\}$ agents as follows:

1. *If $N' \neq \emptyset$ take some $j \in N'$, otherwise stop.*
2. *For every $a^j \in \mathcal{A}^i(a^i, j)$, create an internal node connected from v and labeled with a^j.*
3. *Create one internal node to represent all actions of agent j not in $\mathcal{A}^i(a^i, j)$ (if any), labeled with $*^j$.*
4. *For each child create a subtree with $N' = N' \setminus \{j\}$ using the same procedure.*

Each leaf u of the action tree is the root node of an influence tree. Starting again from $N' = N \setminus \{i\}$:

1. *If $N' \neq \emptyset$ take some $j \in N'$, otherwise stop.*
2. *If the path from s^i to the present node contains an action $a^j \in \mathcal{A}^i(a^i, j)$, create child nodes to represent all local state transitions of agent j compatible with a^j, with arcs labeled $(s^j, \hat{s}^j) \in \{(s^j, \hat{s}^j) : T^j(s^j, a^j, \hat{s}^j) > 0\}$.*
3. *For each child create a sub-tree with $N' = N' \setminus \{j\}$ using the same procedure.*

Finally, we add for each leaf node of the influence tree an arc to the local state node $\hat{s}^i$ labeled with the transition reward.

The labels on the path to a leaf node of an influence tree, via a leaf node of the action tree, sufficiently specify the joint transitions of the agents in scope of the functions $R^e \in \mathcal{R}^i$, such that we can compute the reward $\sum_{R^e \in \mathcal{R}_i} R^e(s^e, \vec{a}^e, \hat{s}^e)$. Note that for each R^e for which an action is chosen that is not in $\mathcal{A}(a^i, j)$ (a wildcard in the action tree), the interaction reward must be 0 by definition.

In Fig. 1b an example CRG is illustrated. The local state nodes are displayed as circles; the internal nodes as black dots and action tree leaves as black triangles. The action arcs are labelled a^1, a^2 and 'wildcard' $*^2$, whereas influence arcs are labelled $(s^2_4 \to s^2_4)$ and $(s^2_4 \to s^2_6)$. Note that Def. 2 captures the general case, but often it suffices to consider transitions $(s^i \cup F^{e \setminus i}, \vec{a}^e, \hat{s}^i \cup \hat{F}^{e \setminus i})$, where $F^{e \setminus i}$ is the set of state

features on which the reward functions $\mathcal{R}_i$ depend. This is a further abstraction: only feature influence arcs are needed, typically resulting in much less arcs (see Fig. 2 later).

Now we investigate the maximal size of the CRGs. Let $|S^{max}| = \max_{i \in N} |S^i|$, $|A^{max}| = \max_{i \in N} |A^i|$, $w = \max_{R^e \in \mathcal{R}^e} |e| - 1$, i.e., the maximal interaction function scope size, and $\alpha = \max_{i,j \in N} \max_{a^i \in A^i} |\mathcal{A}^i(a^i, j)|$ the largest dependent action set size. First note that the full joint policy search space is $\Theta(h|S^{max}|^{2n}|A^{max}|^n)$, however we show that the use of CRGs can greatly reduce this.

Theorem 1. *The maximal size of a CRG is*

$$O(\, h \cdot |A^{max}||S^{max}|^2 \cdot (\alpha |S^{max}|^2)^w \,). \tag{4}$$

Proof. A CRG has as many layers as the planning horizon h. In the worst case, in every stage there are $|S^{max}|$ local state nodes, each connected to at most $|S^{max}|$ next-stage local state nodes via multiple arcs. The number of action arcs between two local state nodes s^i and $\hat{s}^i$ is at most $|A^i|$ times the maximal number of dependent actions, α^w. Finally, the number of influence arcs is bounded by $(|S^{max}|^w)^2$. □

Note that in general all actions can be dependent, in which case the size of all n CRGs combined is $O(nh|S^{max}|^{2+2w}|A^{max}|^{1+w})$; typically still much more compact than the full joint policy search space unless $w \approx |N|$. For many problems however, the interaction rewards are more sparse and $\alpha^w \ll |A^{max}|^w$. Moreover, (4) gives an upper bound on the CRG size in general, for a specific CRG ϕ_i this bound is often expressed more tightly by $O(h \cdot |A^i||S^i|^2 \cdot \prod_{j \in N} (\max_{a^i \in A^i} |\mathcal{A}^i(a^i, j)||S^j|^2))$, or even $|F|$ instead of $|S|$ when conditioning rewards on state features is sufficient.

In addition to storing rewards compactly, we use CRGs to bound the optimal policy value. Specifically, the maximal (resp. minimal) return from a joint state s_t onwards, is an upper (resp. lower) bound on the attainable reward. Moreover, the sum of bounds on local returns bounds the global return and thus the global optimal value. We define the bounds recursively:

$$U(s^i) = \max_{(s^e, \vec{a}^e_t, \hat{s}^e) \in \phi_i(s^i)} \left(\mathcal{R}_i(s^e, \vec{a}^e_t, \hat{s}^e) + U(\hat{s}^i) \right), \tag{5}$$

such that $\phi_i(s^i)$ denotes the set of local transitions available from state $s^i \in s^e$ (ending in $\hat{s}^i \in \hat{s}^e$). The bound on the optimal value for a joint transition $(s, \vec{a}, \hat{s})$ of all agents is

$$U(s, \vec{a}_t, \hat{s}) = \sum\nolimits_{i \in N} \left(\mathcal{R}_i(s^e, \vec{a}^e_t, \hat{s}^e) + U(\hat{s}^i) \right), \tag{6}$$

and lower bound L is defined similarly over minimal returns.

Furthermore, CRGs can exploit independence in local reward functions as a result of past decisions. In many task-modelling MMDPs, e.g., those mentioned in the introduction, actions can be performed a limited amount of times, after which reward interactions involving that action no longer occur. When an agent can no longer perform dependent actions, the expected value of the remaining decisions is found through local optimisation. More generally, when dependencies between groups of agents no longer occur, the policy search space can be decoupled into independent components for which a policy may be found separately while their combination is still globally optimal.

Algorithm 1: $\texttt{CoRe}(\Phi, \theta^N_t, h, N)$

Input: CRGs Φ, execution sequence θ^N_t, horizon h, agents N

if $t = h$ **then return** 0;
$V^* \leftarrow 0$
foreach *conditionally independent subset* $N' \subseteq N$ *given* θ^N_t **do**
  // Compute weighted sums of bounds:
  $\forall \vec{a}^{N'}_t : U(s^{N'}_{\theta,t}, \vec{a}^{N'}_t) \leftarrow \sum_{s^{N'}_{t+1}} T(s^{N'}_{\theta,t}, \vec{a}^{N'}_t, s^{N'}_{t+1}) U(s^{N'}_{\theta,t}, \vec{a}^{N'}_t, s^{N'}_{t+1})$
  $L_{max} \leftarrow \max_{\vec{a}_t} \sum_{s^{N'}_{t+1}} T(s^{N'}_{\theta,t}, \vec{a}^{N'}_t, s^{N'}_{t+1}) L(s^{N'}_{\theta,t}, \vec{a}^{N'}_t, s^{N'}_{t+1})$
  // Find joint action maximising expected reward
  foreach $\vec{a}^{N'}_t$ *for which* $U(s^{N'}_{\theta,t}, \vec{a}^{N'}_t) \geq L_{max}$ **do**
    $V_{\vec{a}^{N'}_t} \leftarrow 0$
    foreach $s^{N'}_{t+1}$ *reachable from* $s^{N'}_{\theta,t}$ *and* $\vec{a}^{N'}_t$ **do**
      $V_{\vec{a}^{N'}_t} \mathrel{+}= T(s^{N'}_{\theta,t}, \vec{a}^{N'}_t, s^{N'}_{t+1}) \Big(R(s^{N'}_{\theta,t}, \vec{a}^{N'}_t, s^{N'}_{t+1}) + \texttt{CoRe}(\Phi, \theta^{N'}_t \oplus [\vec{a}^{N'}_t, s^{N'}_{t+1}], h, N') \Big)$
    $L_{max} \leftarrow \max(V_{\vec{a}^{N'}_t}, L_{max})$; // update lb
  $V^* \mathrel{+}= \max_{\vec{a}^{N'}_t} V_{\vec{a}^{N'}_t}$
return V^*

Definition 3 (Conditional Reward Independence)**.** *Given an execution sequence θ_t, two agents $i, j \in N$ are* conditionally reward independent*, denoted $CRI(i, j, \theta_t)$, if for all future states $s_t, s_{t+1} \in S$ and every future joint action $\vec{a}_t \in A$:*

$$\forall R^e \in \mathcal{R} \; s.t. \; \{i, j\} \subseteq e : \sum\nolimits_{x=t}^{h-1} R^e(s_x, \vec{a}_x, s_{x+1}) = 0.$$

Although reward independence is concluded from joint execution sequence θ_t, some independence can be detected from the local execution sequence θ^i_t only, for example when agent i completes its dependent actions. This *local conditional reward independence* occurs when $\forall j \in N : CRI(i, j, \theta^i_t)$ and is easily detected from the state during CRG generation. For each such state s^i, we find optimal policy $\pi^*_i(s^i)$ and add only the optimal transitions to the CRG.

Together this leads to the *Conditional Return Policy Search (CoRe)* (Alg. 1). CoRe performs a branch-and-bound search over the joint policy space, represented as a DAG with nodes s_t and edges $\langle \vec{a}_t, \hat{s}_{t+1} \rangle$, such that finding a joint policy corresponds to selecting a subset of action arcs from the CRGs (corresponding to $\vec{a}_t$ and $\hat{s}_{t+1}$). First, however, the CRGs ϕ_i are constructed for the local rewards $\mathcal{R}_i$ of each agent $i \in N$, assigned heuristically to obtain balanced CRGs. The generation of the CRGs follows Def. 2 using a recursive procedure, during which we store bounds (Eq. 5). During the subsequent policy search CoRe detects when subsets of agents, $N' \subset N$, become conditionally reward independent, and recurses on these subsets separately.

Theorem 2 (CoRe Correctness)**.** *Given TI-MMDP $M = \langle N, S, A, T, \mathcal{R} \rangle$ with (implicit) initial state s_0, CoRe always returns the optimal MMDP policy value $V^*(s_0)$ (Eq. 1).*

Proof. (Sketch) Conditional reward independence enables optimal decoupling of policy search, the bounds are admissible with respect to the optimal policy value and our pruning does not exclude optimal execution sequences.[1] □

[1]The full proof can be found in the extended version of this paper, available at `http://arxiv.org/abs/1511.09047`.

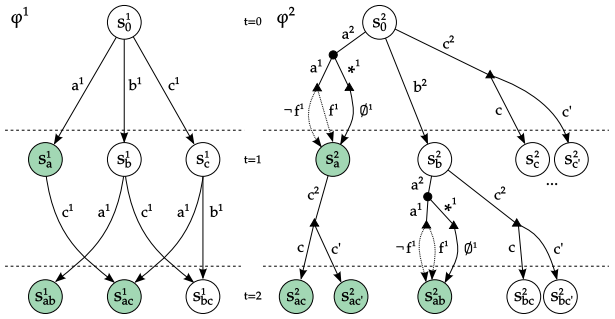


Figure 2: The CRGs of the two agents. We omit the branches for a^2 and b^2 from states s^2_c and $s^2_{c'}$. The highlighted states are locally reward independent (reward arcs are omitted).

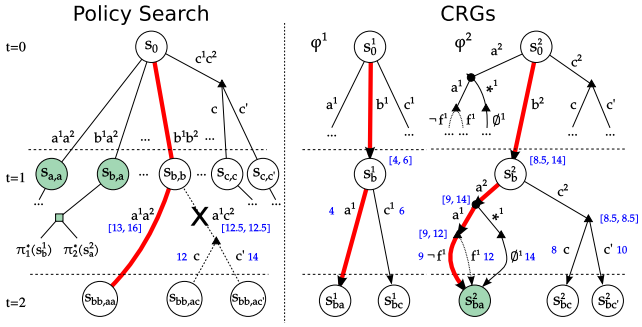


Figure 3: Example of policy evaluation. The left graph shows (a part of) the policy search tree with joint states and joint actions, and the right graph the CRGs per agent.

CoRe Example

We present a two-agent example problem in which both agents have actions a, b and c, but every action can be performed only once within a 2-step horizon. Action c^2 of agent 2 is (for ease of exposition) the only stochastic action with outcomes c and c', and corresponding probabilities 0.75 and 0.25. There is only one interaction, between actions a^1 and a^2, and the reward depends on feature f^1 of agent 1 being set from $f^1?$ to f^1 or $\neg f^1$. Thus we have one interaction reward function with rewards $R^{1,2}(f^1?, \{a^1, a^2\}, f^1)$ and $R^{1,2}(f^1?, \{a^1, a^2\}, \neg f^1)$, and local rewards R^1 and R^2.

Figure 2 illustrates the two CRGs. On the left is the CRG ϕ^1 of agent 1 with only its local reward R^1, while the CRG of agent 2 includes both the reward interaction function $R^{1,2}$ and its local reward R^2. Notice that only when sequences start with action a^2 additional arcs are included in CRG ϕ^2 to account for reward interactions. The sequence starting with a^2 is followed by an after-state node with two arcs: one for agent 1 performing a^1 and one for its other actions, $*^1 = \{b^1, c^1\}$. The interaction reward depends on what feature f^1 is (stochastically) set to, thus the influence arcs f^1 and $\neg f^1$. As the interaction reward only occurs when $\{a^1, a^2\}$ is executed, the fully-specified after-state node after a^2 and $*^1$ (the triangle below it) has a no-influence arc $\varnothing^1$. All other transitions are reward independent and captured by local transitions (s^2_0, b^2, s^2_b) and (s^2_0, c^2, s^2_c). Locally independent states are highlighted green, from which only the optimal action transitions are kept in the CRG, e.g., only action arc c^1 (and not b^1) is included from s^1_a.

An example of CoRe policy search is shown in Figure 3, with the policy search space on the left and the CRGs on the right, now annotated with return bounds. Only several of the branches of the full DAG and CRGs are shown to preserve clarity. At $t = 0$, there are 9 joint actions with 12 result states, while the CRGs need only 3 + 4 states and 3 + 6 transitions to represent all rewards. The execution sequence θ_h that is evaluated is highlighted in thick red. This sequence starts with non-dependent actions $\{b^1, b^2\}$, resulting in joint state $s_{b,b}$ (ignore the bounds in blue for now). The execution sequence at $t = 1$ is thus $\theta_1 = [s_0, \{b^1, b^2\}, s_{b,b}]$. In the CRGs the corresponding transitions to states s^1_b and s^2_b are shown. Now for $t = 1$ CoRe is evaluating joint action $\{a^1, a^2\}$ that is reward-interacting and thus the value of state feature f^1 is required to determine the transition in ϕ^2 (here chosen arbitrarily as $\neg f^1$). The corresponding execution sequence (of agent 2) is therefore $\theta^2_2 = [s^2_0, \{b^1, b^2\}, s^2_b \cup \{f^1?\}, \{a^1, a^2\}, s^2_{ba} \cup \{\neg f^1\}]$. If agent 1 had chosen c^1 instead, we would traverse the branch $*^1$ leading to state s^2_{ba} without reward interactions.

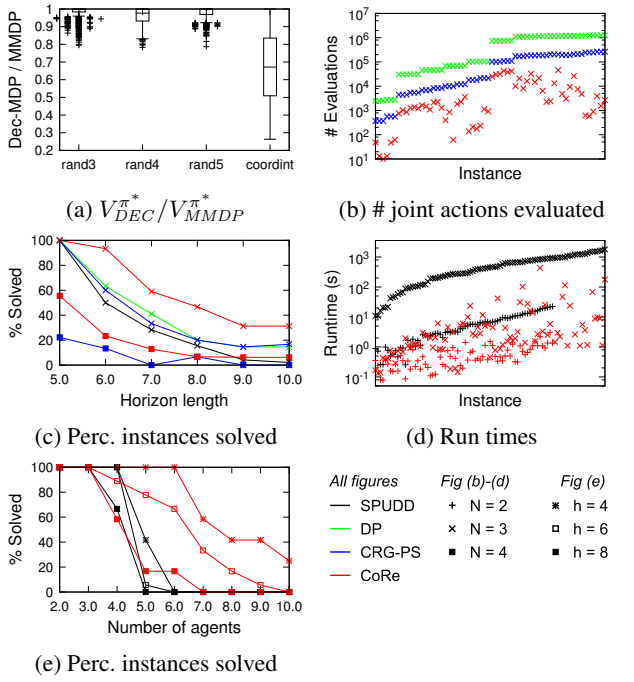
Branch-and-bound is shown (in blue) for state $s_{b,b}$, with the rewards labelled on transitions and their bounds at the nodes. The bounds for joint actions $\{a^1, a^2\}$ and $\{a^1, c^2\}$ are $[13, 16]$ and $[12.5, 12.5]$, respectively, found by summing the CRG bounds, hence $\{a^1, c^2\}$ can be pruned. Note that we can compute the expected value of $\{a^1, c^2\}$ in the CRG, but not that of $\{a^1, a^2\}$ because agent 2 does not know the value of f^1 or the probability of a^1 during CRG generation.

Conditional reward independence occurs in the green states of the policy search tree. After joint action $\{b^1, a^2\}$, the agents will no longer interact (a^2 is done) and thus the problem is decoupled. From state $s_{b,a}$ CoRe finds optimal policies $\pi^*_1(s^1_b)$ and $\pi^*_2(s^2_a)$ and combines them into an optimal joint policy $\pi^*(s_{b,a}) = \langle \pi^*_1(s^1_b), \pi^*_2(s^2_a) \rangle$.

Evaluation

In our experiments we find optimal policies for the *maintenance planning problem* (MPP, see the introduction) that minimise the (time-dependent) maintenance costs and economic losses due to traffic hindrance. Using this domain we conduct three experiments with CoRe to study 1) the expected value when solving centrally versus decentralised methods, 2) the impact on the number of joint actions evaluated and 3) the scalability in terms of agents.

First, we compare with a decentralised baseline by treating the problem as a (transition and observation independent) Dec-MDP (Becker et al. 2003) in which agents can only observe their local state. Although the (TI-)Dec-MDP model is fundamentally different from TI-MMDP – in the latter decisions are coordinated on *joint* (i.e., global) observations – the advances in Dec-MDP solution methods (Dibangoye et al. 2013) may be useful for TI-MMDP problems if they can deliver sufficient quality policies. That is, since they assume less information available, the value of Dec-MDP policies will *at best* equal that of their MMDP counterparts, but in practice the expected value obtained from following a decentralised policy may be lower. We in-



(a) $V^{\pi^*}_{DEC}/V^{\pi^*}_{MMDP}$ (b) # joint actions evaluated

(c) Perc. instances solved (d) Run times

(e) Perc. instances solved

Figure 4: Experimental results

vestigate if this is the case in our first experiment, which compares the expected value of optimal MMDP policies found by CoRe with optimal Dec-MDP policies, as found by the GMAA-ICE* algorithm (Oliehoek et al. 2013).

For this initial experiment we use two benchmark sets: `rand[h]`, 3 sets of 1000 random two-agent problems with horizons $h \in [3, 4, 5]$, and `coordint`, a set of 1000 coordination-intensive instances where poor coordination results in low expected values. Figure 4a shows the ratio $V^{\pi^*}_{DEC}/V^{\pi^*}_{MMDP}$. In the random instances the expected values of both policies equal in approximately half of the instances. For coordination-intensive instances `coordint` decentralised policies result in worse results – on average the reward loss is about 33%, but it can be 75% – demonstrating that decentralised policies are inadequate for our purposes.

In our remaining experiments we used a random test set `mpp` with 2, 3 and 4-agent problems (400 each) with 3 maintenance tasks, planning horizons 5 to 10, random delay probabilities and binary reward interactions. We compare CoRe against the current state-of-the-art MPP method from (Scharpff et al. 2013), solving a compact encoding of the problem through value iteration (SPUDD) (Hoey et al. 1999), and a dynamic programming algorithm that maximises Eq. 1 with added domain knowledge to quickly identify and prune infeasible branches. We included CRG policy search without bounds (CRG-PS) to study the impact of branch-and-bound.

Figure 4b shows the search space size reduction by CRGs in this domain. Our CRG-enabled algorithm (CRG-PS, blue) approximately decimates the number of evaluated joint actions compared to the DP method (green). Furthermore, when value bounds are used (CoRe, red), this number is reduced even more, although its effect varies per instance.

Figure 4c shows the percentage of problems from the `mpp` test set that are solved within 30 minutes per method (all two-agent instances were solved and hence omitted). CoRe solves more instances than SPUDD (black) of the 3 agent problems (cross marks), and only CRG-PS and CoRe solve 4-agent instances. This is because CRGs successfully exploit the conditional action independence that decouples the agents *for most of the planning decisions*. Only when reward interactions may occur actions are coordinated.

As CoRe achieves a greater coverage than SPUDD, we compare runtimes only for instances successfully solved by the latter (Fig. 4d). We order the instances on their SPUDD runtime (causing the apparent dispersion in CoRe runtimes) and plot runtimes of both. CoRe solves almost all instances faster than SPUDD, both with 2 and 3 agents. CoRe failed on 3.4% of the instances solved by SPUDD whereas SPUDD failed 63.9% of the instances that CoRe solved.

Finally, to study the agent-scalability of CoRe, we generated a test set `pyra` with a pyramid-like reward interaction structure: every first action of the k-th agent depends on the first action of agent $2k$ and agent $2k + 1$. Figure 4e shows the percentage of solved instances from the `pyra` test for various problem horizons. Whereas previous state-of-the-art solved instances up to only 5 agents, CoRe successfully solved about a quarter of the 10 agent problems ($h = 4$) and overall solves many of the previously unsolvable instances.

Conclusions and Future Work

In this work, we focus on optimally (and centrally) solving fully-observable, stochastic planning problems where agents are dependent only through interaction rewards. We partition individual and interaction rewards per agent in *conditional return graphs*, a compact and efficient data structure when interactions are sparse and/or non-recurrent. We propose a conditional return policy search algorithm (CoRe) that uses reward bounds based on CRGs to reduce the search space size, shown to be by orders of magnitude in the maintenance planning domain. This enables CoRe to overall decrease the runtime required compared to the previously best approach and solve instances previously deemed unsolvable.

We consider only optimal solutions, but CRGs can be combined with approximation in several ways. First, the reward structure of the problem itself may be approximated. For instance, the reward function approximation of (Koller and Parr 1999) can be applied to increase reward sparsity, or CRG paths with relatively small reward differences may be grouped, trading off a (bounded) reward loss for compactness. Secondly, the CRG bounds directly lead to a bounded-approximation scheme, usable in for instance the approximate multi-objective method of (Roijers et al. 2014). Lastly, CRGs can be implemented in any (approximate) TI-MMDP algorithm or, vice versa, any existing approximation scheme for MMDP that preserves TI can be used within CoRe.

Although we focused on transition-independent MMDPs, CRGs may be interesting for general MMDPs when transition dependencies are sparse. This would require including dependent-state transitions in the CRGs similar to reward-interaction paths and is considered to be future work.

Acknowledgements This research is supported by the NWO DTC-NCAP (#612.001.109), Next Generation Infrastructures/Almende BV and NWO VENI (#639.021.336) projects.

References

Bakker, B.; Whiteson, S.; Kester, L.; and Groen, F. 2010. *Traffic Light Control by Multiagent Reinforcement Learning Systems*. Studies in Computational Intelligence, Springer. chapter Interactive Collaborative Information Systems, 475–510.

Becker, R.; Zilberstein, S.; Lesser, V.; and Goldman, C. V. 2003. Transition-independent decentralized Markov decision processes. In *Proceedings of the Int. Conf. on Autonomous Agents and Multiagent Systems*, 41–48.

Becker, R.; Zilberstein, S.; and Lesser, V. 2004. Decentralized Markov decision processes with event-driven interactions. In *Proceedings of the Int. Conf. on Autonomous Agents and Multiagent Systems*, 302–309.

Boutilier, C. 1996. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the Int. Conf. on Theoretical Aspects of Rationality and Knowledge*.

Cavallo, R.; Parkes, D. C.; and Singh, S. 2006. Optimal coordinated planning amongst self-interested agents with private state. In *Proceedings of Uncertainty in Artificial Intelligence*.

Dibangoye, J. S.; Amato, C.; Doniec, A.; and Charpillet, F. 2013. Producing efficient error-bounded solutions for transition independent decentralized MDPs. In *Proceedings of the Int. Conf. on Autonomous Agents and Multiagent Systems*.

Guestrin, C.; Koller, D.; and Parr, R. 2002. Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems 14*. MIT Press.

Guestrin, C.; Venkataraman, S.; and Koller, D. 2002. Context-specific multiagent coordination and planning with factored MDPs. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, 253–259.

Hoey, J.; St-Aubin, R.; Hu, A.; and Boutilier, C. 1999. SPUDD: Stochastic planning using decision diagrams. *Proceedings of Uncertainty in Artificial Intelligence*.

Kok, J. R., and Vlassis, N. 2004. Sparse cooperative Q-learning. In *Proceedings of the Int. Conf. on Machine Learning*, 481–488.

Koller, D., and Parr, R. 1999. Computing factored value functions for policies in structured MDPs. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1332–1339.

Melo, F. S., and Veloso, M. 2011. Decentralized MDPs with sparse interactions. *Artificial Intelligence* 175(11):1757–1789.

Messias, J. V.; Spaan, M. T. J.; and Lima, P. U. 2013. GSMDPs for multi-robot sequential decision-making. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 1408–1414.

Meuleau, N.; Hauskrecht, M.; Kim, K.-E.; Peshkin, L.; Kaelbling, L. P.; Dean, T. L.; and Boutilier, C. 1998. Solving very large weakly coupled Markov decision processes. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 165–172.

Mostafa, H., and Lesser, V. 2009. Offline planning for communication by exploiting structured interactions in decentralized MDPs. In *Proceedings of the International Joint Conference on Web Intelligence and Intelligent Agent Technologies*, volume 2, 193–200.

Nair, R.; Varakantham, P.; Tambe, M.; and Yokoo, M. 2005. Networked distributed POMDPs: A synthesis of distributed constraint optimization and POMDPs. *Proceedings of the Twentieth National Conference on Artificial Intelligence*.

Oliehoek, F. A.; Spaan, M. T. J.; Whiteson, S.; and Vlassis, N. 2008. Exploiting locality of interaction in factored Dec-POMDPs. In *Proceedings of the Int. Conf. on Autonomous Agents and Multiagent Systems*, 517–524.

Oliehoek, F. A.; Spaan, M. T. J.; Amato, C.; and Whiteson, S. 2013. Incremental clustering and expansion for faster optimal planning in decentralized POMDPs. *Journal of Artificial Intelligence Research* 46:449–509.

Oliehoek, F. A.; Spaan, M. T. J.; and Witwicki, S. J. 2015. Factored upper bounds for multiagent planning problems under uncertainty with non-factored value functions. In *Proc. of International Joint Conference on Artificial Intelligence*, 1645–1651.

Oliehoek, F. A.; Whiteson, S.; and Spaan, M. T. J. 2013. Approximate solutions for factored Dec-POMDPs with many agents. In *Proceedings of the Int. Conf. on Autonomous Agents and Multiagent Systems*, 563–570.

Parr, R. 1998. Flexible decomposition algorithms for weakly coupled Markov decision problems. In *Proceedings of Uncertainty in Artificial Intelligence*, 422–430. Morgan Kaufmann Publishers Inc.

Roijers, D. M.; Scharpff, J.; Spaan, M. T. J.; Oliehoek, F. A.; De Weerdt, M.; and Whiteson, S. 2014. Bounded approximations for linear multi-objective planning under uncertainty. In *Proceedings of the Int. Conf. on Automated Planning and Scheduling*, 262–270.

Scharpff, J.; Spaan, M. T. J.; de Weerdt, M.; and Volker, L. 2013. Planning under uncertainty for coordinating infrastructural maintenance. In *Proceedings of the Int. Conf. on Automated Planning and Scheduling*, 425–433.

Spaan, M. T. J.; Gordon, G. J.; and Vlassis, N. 2006. Decentralized planning under uncertainty for teams of communicating agents. In *Proceedings of the Int. Conf. on Autonomous Agents and Multiagent Systems*.

Varakantham, P.; Marecki, J.; Yabu, Y.; Tambe, M.; and Yokoo, M. 2007. Letting loose a SPIDER on a network of POMDPs: Generating quality guaranteed policies. In *Proceedings of the Int. Conf. on Autonomous Agents and Multiagent Systems*.

Witwicki, S. J., and Durfee, E. H. 2010. Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In *Proceedings of the Int. Conf. on Automated Planning and Scheduling*, 185–192.