Using linear programming duality for solving finite horizon Dec-POMDPs

Raghav Aras, Alain Dutech, François Charpillet

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

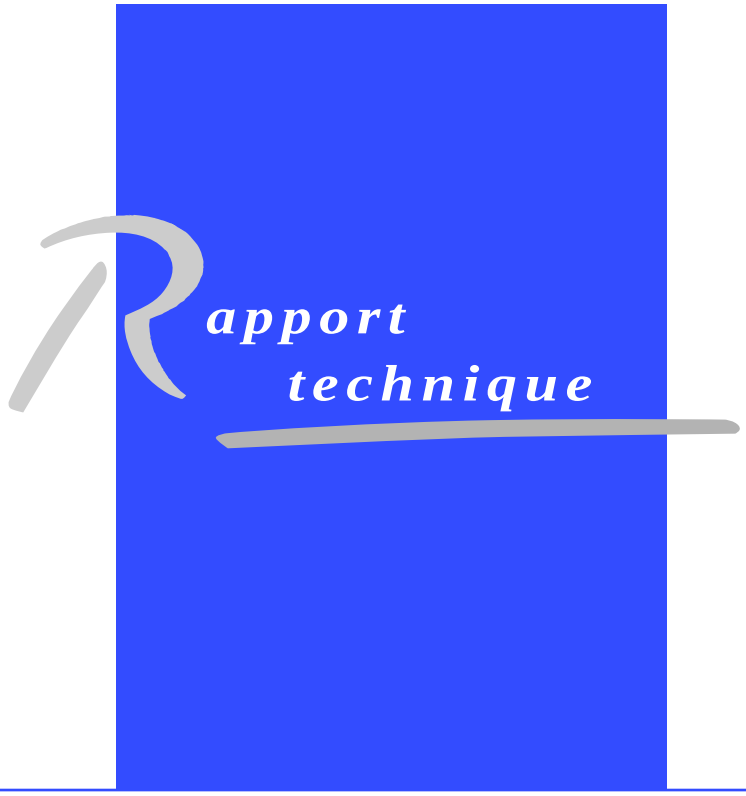
Using linear programming duality for solving finite horizon Dec-POMDPs

Raghav Aras — Alain Dutech — François Charpillet

N° 6641

September 2008

Thème NUM



ISSN 0249-0803 ISRN INRIA/RT--6641--FR+ENG

Using linear programming duality for solving finite horizon Dec-POMDPs

Raghav Aras , Alain Dutech , François Charpillet

Thème NUM —Systèmes numériques
Équipes-Projets MAIA

Rapport technique n° 6641 —September 2008 —24 pages

Abstract: This paper studies the problem of finding an optimal finite horizon joint policy for a decentralized partially observable Markov decision process (Dec-POMDP). We present a new algorithm for finding an optimal joint policy. The algorithm is based on the fact that the necessary condition for a joint policy to be optimal is that it be locally optimal (that is, a Nash equilibrium). Through the application of linear programming duality, the necessary condition can be transformed to a nonlinear program which can then further be transformed to a 0-1 mixed integer linear program (MILP) whose optimal solution is an optimal joint policy (in the sequence form). The proposed algorithm thus consists of solving this 0-1 MILP. Computational experience of the 0-1 MILP on two and three agent DEC-POMDPs gives mixed results. On some problems it is faster than existing algorithms, on others it is slower.

Key-words: multi-agent sequential decision making, decentralized Markov processes, partial observability, Dec-POMDPs

Centre de recherche INRIA Nancy – Grand Est
LORIA, Technopôle de Nancy-Brabois, Campus scientifique,
615, rue du Jardin Botanique, BP 101, 54602 Villers-Lès-Nancy
Téléphone : +33 3 83 59 30 00 —Télécopie : +33 3 83 27 83 19

exploiter la dualité des programmes linéaires pour résoudre des Dec-POMDPs à horizon fini

Résumé : Nous étudions le problème de la recherche d'une politique jointe optimale à horizon fini d'un processus de Markov décentralisé partiellement observé (Dec-POMDP). Nous proposons un nouveau algorithme pour trouver une politique jointe optimale. La condition nécessaire pour qu'une politque jointe soit optimale est qu'elle soit localement optimale (autrement dit, un équilibre de Nash). En appliquant le théorème de la dualité des programmes linéaires, cette condition se transforme à un programme non-linéaire qui peut être lui-même transformer à un programme linéaire en variables bivalentes et continues dont la solution optimale est une politique optimale jointe (sous forme séquentielle). L'algorithme proposé donc consiste de résoudre ce programme linéaire en variables bivalentes et continues. L'éxpérience computationelle de cet algorithme sur quelques problémes à deux et trois agents donne des résultats mixtes; l'algorithme est plus rapides que des algorithmes existants pour certains problèmes, moins rapides pour les autres.

Mots-clés : processus décisionnels multi-agents, processus décisionnels de Markov décentralisés, observabilité partielle, Dec-POMDPs

1 Introduction

Since its inception in the early 1970s, the POMDP model has enabled many problems from the domains of operations research and robotics involving single agent sequential decision making under uncertainty to be formulated, and thereby solved. The decentralized POMDP[2] (Dec-POMDP) model generalizes the POMDP model by allowing multi-agent sequential decision making under uncertainty. In a Dec-POMDP, a set of agents are required to cooperate to achieve a common objective. The common objective concerns the optimal control of a (discrete-state, discrete-time) Markov process whose state is hidden from the agents. They are required to control the process with only partial information about the state.

The focus of this paper is the finite horizon Dec-POMDP problem. In this problem, the agents are required to optimally control the given partially observable Markov process for a finite number of time periods. The objective is to choose such joint actions that maximize the expected sum of rewards obtainable in the given number of periods. A solution to this problem is a joint control policy that achieves this objective. A joint policy is a tuple of policies, one policy in the tuple per agent. A policy is a function or decision rule which a agent employs while choosing actions to control the Markov process. Finding an optimal finite horizon joint policy is a very hard problem. There are two equivalent ways in which its complexity can be stated. In terms of the parameters of a Dec-POMDP, the problem is NEXP-hard[2]. Alternatively, a tree representation of the problem can be generated from the parameters of a Dec-POMDP and in terms of the size of this tree, the problem is NP-hard[6].

In comparison, finding an optimal policy for a POMDP a is PSPACE-hard[10] problem (in terms of the parameters of a POMDP). The reason for the much lower complexity of a POMDP is that in a POMDP, an enumeration of the possible policies need not be undertaken for finding an optimal policy. The value function of the POMDP - the maximum reward-to-go function - once determined, allows the immediate inference of an optimal policy. The value function being piece-wise linear convex is relatively easy to determine. In a Dec-POMDP, the situation is quite different. The value function alone does not suffice. Given the value function, it is not possible to infer an optimal joint policy directly; we are obliged to search for the joint policy. This means that parallel to the computation of the value function, the joint policy inferable from it must also be determined. In the worst case, this entails the enumeration (and possibly storage) of all possible joint policies. Since the number of joint policies is doubly exponential in the horizon (besides being exponential in the number of agents), the time required in the worst case is doubly exponential in the horizon.

At the present time, four non-trivial algorithms capable of finding an optimal Dec-POMDP joint policy are to be found in the literature to our knowledge. These can be divided into two groups depending on the *form* of joint policy they find. Algorithms of the first group find an optimal joint policy in the *tree form*. This is also the form of a policy used by existing POMDP algorithms[11]. Three[5],[13],[12] of the four algorithms belong to this group. Ameliorated or generalized versions of these algorithms are also now available [9], [3]. These algorithms employ dynamic programming techniques such as backward induction or forward search to find an optimal joint policy. The remaining algorithm[1] belongs to the second group. It finds an optimal joint policy in the *sequence form*, a form of policy introduced by Koller, Megiddo and von Stengel (KMvS)[7] in the context of extensive games. This algorithm is in fact a 0-1 mixed integer linear program (MILP) whose optimal solution is an optimal joint policy.

While the four algorithms (and their improvements) are capable of solving only very small Dec-POMDPs, the 0-1 MILP has been found to be much faster in practice than the other algorithms.

In this paper, we present an addition to the second group. We present a new algorithm that finds an optimal joint policy in the sequence form. This algorithm is also a 0-1 MILP whose optimal solution is an optimal joint policy in the sequence form. While the existing 0-1 MILP (which we shall henceforth refer to as **M**) is conceived by casting the problem as an instance of combinatorial optimization, the 0-1 MILP presented in this paper, which we shall henceforth refer to as **M'**, is obtained by transforming the necessary condition for a joint policy to be optimal into a nonlinear program **N'** and thence by linearizing **N'**. We obtain the necessary condition by an applying the theorem of linear programming duality, although it can also be obtained by the Kuhn-Tucker theorem. The derivation of **N'** we present is more or less identical to the one presented by KMvS[7].

The necessary condition is that a joint policy have *zero regret* in order to be optimal. When a joint policy has zero regret, it means that it cannot be improved by changing the policy of only one agent in it. Thus, an optimal joint policy must fulfill two agendas not one:

(i) It must maximize value,

(ii) It must minimize regret (that is, bring it down to 0).

Fulfillment of the first agenda implies the fulfillment of the second as well, but the inverse does not hold. While existing algorithms (of both groups) focus only on the first agenda to find an optimal joint policy, the algorithm we present in this chapter uses both agendas to find an optimal joint policy.

The remainder of the paper has the following organization. In the next section, we review the Dec-POMDP model and the two forms of a joint policy. In Section 3 we derive the nonlinear program **N'** using the theorem of linear programming duality. In Section 5, we convert **N'** to **M'**. Here again, we present two versions of **M'**, one for the case where there are only two agents and one for the case where there are three or more agents. The two versions are different and are required because the 2 agents version is not extendable readily when 3 or more agents are present. In Section 6, computational experience of **M'** on sample Dec-POMDPs is presented. In Section 7, some conclusions are drawn about the work presented in this paper.

Most of the discussion in the paper shall be directed at the two-agent case. In some sections, we shall move to the three-agent case which shall serve as a surrogate for the 3 or more agents case.

2 The Dec-POMDP Model

The control of a Markov process by two agents in a decentralized manner is described as follows (the description easily generalizes to more than two agents). The control unfolds over discrete time periods. In each period, the process assumes a state from a set of possible states denoted by S. The state occupied by the process is hidden from the agents. In each period, each agent selects an action from the actions available to him and executes that action. The set of actions available to agent i is denoted by A_i. Upon the execution of the pair of actions chosen by the two agents, the period changes. The state occupied by the process in the new period is determined by its state in the previous period and the pair of actions chosen in the previous period. The probability that the state of the process in a period is s' if in the previous period it was s and the

actions chosen by the agents were respectively a_1 and a_2 is denoted by $\mathbb{P}(s, a_1, a_2, s')$. Thus for any period t,

$$\mathbb{P}(s, a_1, a_2, s') \quad = \quad \text{Prob.}(s^{t+1} = s' | s^t = s, a_1^t = a_1, a_2^t = a_2)$$

At the start of each period (except the very first one), each agent receives an observation that is determined by the pair of actions chosen by the agents in the previous period and the state of the process in the current period. The set of possible observation receivable by agent i is denoted by O_i. The probability that agents 1 and 2 receive observations o_1 and o_2 respectively in a period if the state of that period is s' and if the actions chosen by them in the previous period were respectively a_1 and a_2 is denoted by $\mathbb{G}(a_1, a_2, s', o_1, o_2)$. Thus for any period t,

$$\mathbb{G}(a_1, a_2, s', o_1, o_2) \quad = \quad \text{Prob.}(o_1^t = o_1, o_2^t = o_2 | s^t = s', a_1^{t-1} = a_1, a_2^{t-1} = a_2)$$

In each period, the agents' actions realize a reward. This reward depends on the actions chosen by the agents and the state of the process in that period. This reward is independent of the period and is perceived only at the end of the control horizon. The reward realized if the actions chosen by the agents in a period are respectively a_1 and a_2 and if the state of the process in that period is s is denoted by $R(s, a_1, a_2)$. The control horizon is the number of periods for which the agents wish to control the Markov process. It is denoted by T.

The manner in which an agent chooses actions for T periods is called his policy. The policy of an agent is a function of the information he is allowed to have about the process in each period. The information of an agent in a period is the sequence of observations he has received till that period and the sequence of actions he himself has taken till that period. No other information is available to the agent. The tuple of policies of the agents forms a joint policy. The finite horizon Dec-POMDP problem consists of finding an optimal T-period joint policy that maximizes the expected sum of rewards obtained in T periods given that state of the process in the first period is selected according to a probability distribution over S denoted by α. That is, we are required to find a pair of policies p_1^* and p_2^* that maximize the following expectation:

$$E\Big\{ \sum_{t=1}^{T} R(s^t, p_1^*(i_1^t), p_2^*(i_2^t)) \Big\}$$

where i_1^t and i_2^t denote respectively the information available to agents 1 and 2 in period t.

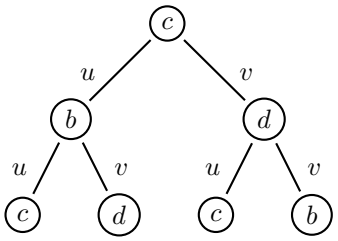
As stated in the introduction, two forms or representations of a policy are used by current algorithms, the tree form and the sequence form. Our algorithm uses the latter. We shall now describe these two forms with more attention to the latter.

2.1 Tree form of a policy

In the tree form, the policy is described as a function that maps sequences of observations receivable by the agent to actions of the agent. A policy in the tree form is defined as follows. For agent i, let $\overline{O}_i^t$ denote the set of sequences of t observations, $t \geq 0$, that can be formed from O_i. $\overline{O}_i^0$ contains only the empty sequence $\varnothing$. Thereby, a T-period policy of agent i is a function π that assigns, to each integer t = 0 to T - 1 and to each sequence of k observations $\overline{o} \in \overline{O}_i^t$ an action $\pi(\overline{o}) \in A_i$. In using π, the agent takes

action $\pi(\overline{o})$ if the sequence of observations he has received till a period is $\overline{o}$. In the first period, the agent takes action $\pi(\varnothing)$.

This definition of a policy results into a tree-like representation. An example of a 3-period policy in the tree form is given in the following figure.



b, c and d are actions of the agent and the set of observations of the agent is assumed to be $\{u, v\}$. In using this policy, the agent takes action c in the first period. In the second period, if he receives observation u, he takes action b and if he receives observation v he takes action d. He selects actions in a similar manner in the third period.

2.2 Sequence form of a policy

The sequence form of a policy is slightly less intuitive to grasp than the tree form. Since our algorithm is based on this form, we shall use a bit more space to describe it than the tree form. This form of a policy was introduced in work on solving games in extensive form by D. Koller, N. Megiddo and B. von Stengel[7].

In this form, the policy is described as a conditional probability distribution over the set of histories of the agent. We define a history of an agent to be a sequence of odd length in which the elements in odd positions are actions of the agent and those in even positions are observations of the agent. A history of length 1 of the agent is just an action of the agent. A T-policy in the sequence form accords a conditional probability (which we shall call a weight) to each history of length T or less of the agent. An agent takes actions according to the probabilities defined by this distribution. A policy in the sequence form can be represented as a vector or as a table whose entries are weights of the histories. The weight of history h in policy p is denoted by $p(h)$.

In a deterministic policy in the sequence form, the weight of each history is either 0 or 1. An agent uses a deterministic policy p in the sequence-form as follows. In each period, the agent takes an action as a function of the observation he receives in the period, and the history of actions taken and observations received till the previous period. If h_t is the history that has occurred till period t and if o is the observation received in period t + 1, then in period t + 1, the agent takes that action a for which $p(h_t oa)$ = 1; there will be only one such action.

An example of a 3-period deterministic policy in the sequence form is the following table. b, c and d are actions of the agent and the set of observations of the agent is assumed to be $\{u, v\}$. Only those histories that receive a weight of 1 are shown.

History	Weight	History	Weight
c	1	cub	1
cvd	1	$cubuc$	1
$cubvd$	1	$cvduc$	1
$cvdvb$	1		

Let this policy be denoted by p. The policy is for 3 periods because the longest history with weight 1 is of length 3 (the number of actions in a history counts as the length of the history). In following this policy, the agent takes action c in period 1. Then, if the observation he has received in period 2 is v, he takes action d in period 2 because $p(cvd)$ = 1. He does not take action b or c at this period because $p(cvb)$ = 0 and $p(cvc)$ = 0. Similarly, if in period 3, the observation received is u and the history that has occurred till the end of period 2 is cvd, then he takes action c because $p(cvduc)$ = 1; he does not take action b or d in this period because $p(cvdub)$ = 0 and $p(cvdud)$ = 0. Note that policy p is equivalent to the policy in the tree form given in the previous subsection in the sense that for a given sequence of observations, the agent takes the same action using either policy.

In a stochastic policy in the sequence form, the weight of a history is not restricted to 0 or 1 but can assume a value in the interval [0, 1]. The following table is an example of a 4-period stochastic policy in the sequence form. Again, only histories with nonzero weights in the policy are shown.

History	Weight	History	Weight
b	0.6	c	0.4
bub	0.6	cub	0.4
bvc	0.6	cvc	0.4
$bubub$	0.12	$bubuc$	0.48
$cubub$	0.08	$cubuc$	0.32
$bubvc$	0.6	$cubvc$	0.4
$bvcub$	0.6	$cvcub$	0.4
$bvcvb$	0.42	$bvcvc$	0.18
$cvcvb$	0.28	$cvcvc$	0.12

An agent uses a stochastic policy in the sequence form in the same manner as he uses a pure policy in the sequence form with one difference. When using a stochastic policy p in the sequence form, in period t + 1, upon receiving observation o, the agent chooses each action a with probability $p(h_toa)/p(h_t)$. Unlike the case of a pure policy in the sequence-form, here the denominator $p(h_t)$ need not be always 1.

Let the policy given in the table above be denoted by q. In using q, in period 1, the agent takes action b with probability 0.6 because $q(b)$ = 0.6 and action c with probability 0.4 because $q(c)$ = 0.4. In period 2, if he receives observation u and if he has taken action b in period 1, he takes action b with probability 1 because $q(bub)/q(b)$ = 0.6/0.6 = 1. Similarly, in period 2, if he receives observation u and if he has taken action c in period 1, he takes action b with probability 1 because $q(cub)/q(c)$ = 0.4/0.4 = 1. This means, that in period 2, regardless of the action taken in period 1, if he receives observation u, he takes action b. To take one more example, if in period 3, the observation received is v and the history that has occurred till the end of period 2 is cvc, then he takes action b with probability $q(cvcvb)/q(cvc)$ = 0.28/0.4 = 0.7, and he takes action c with probability $q(cvcvc)/q(cvc)$ = 0.12/0.4 = 0.3.

The weight $p(h)$ of a history h = (a^1, o^1, a^2, ..., o^{t-1}, a^t) of length t in fact represents the probability with which the agent takes actions a^1, a^2, ..., a^t in periods 1, 2, ..., t respectively if he receives observations o^1, o^2, ..., o^{t-1} in periods 2, 3, ..., t respectively. That is,

$$p(h) \quad = \quad \text{Prob.}(a^1, a^2, \ldots, a^t | o^1, o^2, \ldots, o^{t-1})$$

The weight of a history in a policy is independent of the Dec-POMDP model (i.e., it is free of the probabilities $\mathbb{P}$ and $\mathbb{G}$ as well as the initial state α).

The conditional probability distribution that a policy in the sequence form represents obeys three simple laws described below. Let $\mathcal{H}_i^t$ denote the set of all possible histories of length t of agent i. $\mathcal{H}_i^1$ is thus just the set A_i. Let $\mathcal{H}_i$ denote the set of all possible histories of lengths less than or equal to T of agent i. Let the set of histories of length T be denoted by $\mathcal{E}_i$. A history of length T shall be called a terminal history. Finally, let $\mathcal{N}_i$ denote the set $\mathcal{H}_i \backslash \mathcal{E}_i$, the set of nonterminal histories of agent i. Let the size of $\mathcal{H}_i$ be denoted by n_i. A policy in the sequence form of agent i can be represented by an n_i vector each of which entries is indexed by a history of the agent and contains the weight of that history. Then, an n_i-vector p is a policy in the sequence form of agent i if it obeys the following three laws,

$$\begin{aligned} \sum_{a \in A_i} p(a) &= 1 \\ -p(h) + \sum_{a \in A_i} p(hoa) &= 0, \quad \forall h \in \mathcal{N}_i, \forall o \in O_i \\ p(h) &\geq 0, \quad \forall h \in \mathcal{H}_i \end{aligned}$$

where hoa denotes the history obtained on concatenating o and a to h.

The first equation requires that the sum of weights accorded to histories of length 1 (i.e., actions) equal 1. This equation bootstraps the weight distribution for histories of longer lengths. The second equation enforces a legitimate conditional probability distribution. That is, for each history h and for each observation o, the sum of probabilities with which the agent takes an action in a period conditional on having in the past periods taken actions and received observations according to h and conditional on having received observation o in the current period, is equal to the weight accorded to h.

2.3 Value of a joint policy

The value of a joint policy is the expected sum of rewards it obtains in T periods. In the sequence form, the value of a joint policy is defined in terms of the weights and expected rewards of joint histories. A joint history is a pair of histories of the same length, one history in the pair per agent. A terminal joint history is a pair of terminal histories. The weight of a joint history in a joint policy is the product of the weights of histories of the joint history in the respective policies. The cost of a joint history is the expected sum of costs incurred by the joint actions of the joint history. The value $V(\alpha, p_1, p_2)$ of a joint policy (p_1, p_2) is defined as,

$$V(\alpha, p_1, p_2) = \sum_{h_1 \in \mathcal{E}_1} \sum_{h_2 \in \mathcal{E}_2} r(\alpha, h_1, h_2) p_1(h_1) p_2(h_2)$$

where $r(\alpha, h_1, h_2)$ denotes the expected reward of joint history (h_1, h_2).

The expected reward of a joint history (h_1, h_2) in turn is a product of two quanes, the probability $\Psi(\alpha, h_1, h_2)$ with which the agents receive the sequence of joint observations of the joint history conditional on the agents taking the sequence of joint actions of the joint history, and the sum $\mathcal{S}(\alpha, h_1, h_2)$ of expected rewards of the joint actions of the joint history:

$$r(\alpha, h_1, h_2) = \Psi(\alpha, h_1, h_2) \mathcal{S}(\alpha, h_1, h_2)$$

Let (h_1, h_2) be a joint history of length t, and let o^j and a^j denote respectively the jth joint observation and jth joint action of (h_1, h_2). Then, $\Psi(\alpha, h_1, h_2)$ is defined as follows.

$$\begin{aligned}\Psi(\alpha, h_1, h_2) &= \text{Prob.}(o^1, o^2, \ldots, o^{t-1} | \alpha, a^1, a^2, \ldots, a^{t-1}) \\ &= \prod_{k=1}^{t-1} \mathcal{T}(o^k | \alpha^{k-1}, a^k)\end{aligned}$$

where $\alpha^0 = \alpha$, and for each k = 1 to t - 1,

$$\mathcal{T}(o^k | \alpha^{k-1}, a^k) = \sum_{s \in S} \alpha^{k-1}(s) \sum_{s' \in S} \mathbb{P}(s, a^k, s') \mathbb{G}(a, s', o^k)$$

and for each k = 1 to t - 1,

$$\alpha^k(s') = \frac{\sum_{s \in S} \alpha^{k-1}(s) \mathbb{P}(s, a^k, s') \mathbb{G}(a, s', o^k)}{\mathcal{T}(o^k | \alpha^{k-1}, a^k)}, \quad \forall s' \in S$$

The sum of expected rewards of the joint actions of (h_1, h_2) is defined as,

$$\mathcal{S}(\alpha, h_1, h_2) = \sum_{k=1}^{t} \sum_{s \in S} \alpha^{k-1}(s) R(s, a^k)$$

where for each k = 1 to t - 1, α^{k-1} is defined as above.

3 A nonlinear program

We now have the elements required to define a nonlinear program (NLP) whose (globally) optimal solution is an optimal joint policy in the sequence form. This nonlinear program is as follows.

$$\text{Maximize} \quad \sum_{h_1 \in \mathcal{E}_1} \sum_{h_2 \in \mathcal{E}_2} r(\alpha, h_1, h_2) x_1(h_1) x_2(h_2)$$

Subject To,

$$\begin{aligned}\sum_{a \in A_i} x_i(a) &= 1, \quad i = 1, 2 \\ -x_i(h) + \sum_{a \in A_i} x_i(hoa) &= 0, \quad i = 1, 2, \forall h \in \mathcal{N}_i, \forall o \in O_i \\ x_i(h) &\geq 0, \quad i = 1, 2, \forall h \in \mathcal{H}_i\end{aligned}$$

This NLP is presented for the 2-agent case, but can be generalized to more than 2 agents. It therefore consists of two vectors x_1 and x_2 of variables. An optimal solution (x_1^*, x_2^*) to the program is an optimal T-period joint policy. The constraints of the program ensure that the vectors veritably represent T-period (possibly stochastic) policies in the sequence form of the two agents. We shall henceforth denote this NLP by **N**.

N contains two sets of constraints, one per agent. The sets are independent of one another in that neither uses variables appearing in the other. In the two-agent case, **N** is a bilinear program with separable constraints. The constraints of **N** are linear.

Together they form a convex set. The objective function of **N** is quadratic (in n-agent case, it is n-adic). A global maximum point of **N** is an optimal T-period joint policy. Unfortunately, finding an global maximum of a constrained nonconcave function is a very difficult problem. It is NP-hard. Graver still is the fact that there are no generalized methods that guarantee finding one. Most methods guarantee finding a local maximum point, but even that is an NP-hard problem.

$\big\{$A function f defined on a convex set M is said to be concave if for every w, $w' \in M$, and every real number $\beta \in [0,1]$, there holds,

$$f(\beta w + (1-\beta)w') \geq \beta f(w) + (1-\beta) f(w') \tag{1}$$

Geometrically, a function is concave if the line joining any two points on its graph lies nowhere above its graph. Given a function f defined over a convex set M, $w^* \in M$ is a local maximum point of f over M if there exists an $\epsilon > 0$ such that for all $w \in M$ within a distance of ϵ from w^* (i.e., $|w - w^*| < \epsilon$), $f(w^*) \geq f(w)$. Further, $w^* \in M$ is a global maximum point of f over M if for all $w \in M$, $f(w^*) \geq f(w)$.$\big\}$

Solving **N** directly is thus not a viable alternative. One approach is to linearize the objective function of **N**. One such linearization[1] results in a 0-1 MILP given as follows. The 3-agent case is shown for clarity.

$$\text{Maximize} \quad \sum_{h_1 \in \mathcal{E}_1} \sum_{h_2 \in \mathcal{E}_2} \sum_{h_3 \in \mathcal{E}_3} r(\alpha, h_1, h_2, h_3) z(h_1, h_2, h_3)$$

Subject To,

$$\begin{aligned}
\sum_{a \in A_i} x_i(a) &= 1, \quad i = 1, 2, 3 \\
-x_i(h) + \sum_{a \in A_i} x_i(hoa) &= 0, \quad i = 1, 2, 3, \forall h \in \mathcal{N}_i, \forall o \in O_i \\
x_i(h) &\geq 0, \quad i = 1, 2, 3, \forall h \in \mathcal{N}_i \\
x_i(h) &\in \{0,1\}, \quad i = 1, 2, 3, \forall h \in \mathcal{E}_i \\
(|O_2||O_3|)^{T-1} x_1(h) &= \sum_{h_2 \in \mathcal{E}_2} \sum_{h_3 \in \mathcal{E}_3} z(h, h_2, h_3), \quad \forall h \in \mathcal{E}_1 \\
(|O_1||O_3|)^{T-1} x_2(h) &= \sum_{h_1 \in \mathcal{E}_2} \sum_{h_3 \in \mathcal{E}_3} z(h_1, h, h_3), \quad \forall h \in \mathcal{E}_2 \\
(|O_1||O_2|)^{T-1} x_3(h) &= \sum_{h_1 \in \mathcal{E}_2} \sum_{h_2 \in \mathcal{E}_2} z(h_1, h_2, h), \quad \forall h \in \mathcal{E}_3 \\
\sum_{h_1 \in \mathcal{E}_1} \sum_{h_2 \in \mathcal{E}_2} \sum_{h_3 \in \mathcal{E}_3} z(h_1, h_2, h_3) &= (|O_1||O_2||O_3|)^{T-1} \\
z(h_1, h_2, h_3) &\in [0,1], \quad \forall h_1 \in \mathcal{E}_1, \forall h_2 \in \mathcal{E}_2, \forall h_3 \in \mathcal{E}_3
\end{aligned}$$

This MILP shall be henceforth referred to as **M**. An optimal solution to **M** is an optimal T-period pure joint policy. Thus unlike **N**, the solution set of **M** is restricted to the set of pure joint policies.

In order to achieve the linerization of **N**, **M** uses a variable $z(h_1, h_2, h_3)$ for every terminal joint history (h_1, h_2, h_3) in addition to the variables used by **N**. It places 0-1 integer constraints on the variables of terminal histories of each agent. The effect of these constraints is that in every solution to **M**, every x variable, be it representing a terminal history or a nonterminal one, has value 0 or 1.

Further, **M** introduces an additional set of constraints to ensure that the following double implication holds for every terminal joint history h_1, h_2, h_3),

$$z^*(h_1, h_2, h_3) = 1 \quad \Leftrightarrow \quad x^*(h_1) = 1, x^*(h_3) = 1, x^*(h_3) = 1$$

The additional constraints are based on the insight that in every Dec-POMDP, there exists an optimal joint policy that is pure, and the number of histories that receive a weight of 1 in a pure policy of an agent is fixed (the number of terminal histories of agent i that receive a weight of 1 in a policy of agent i is $|O_i|^{T-1}$).

The computational experience of **M** on standard Dec-POMDP problem shows that it is much faster algorithms of the first group, as mentioned in the introduction. The main drawback of **M** is that its size is much larger than **N**. While the number of variables and constraints in the latter is exponential in T and linear in the number of agents, the number of variables and constraints in the former is exponential in T and exponential in the number of agents.

In the next section, we introduce a new 0-1 MILP obtained not so much from linearizing **N**, but rather from inquiring into the necessary conditions for a joint policy to be optimal.

4 Necessary conditions for optimality

An optimal joint policy is also a locally optimal joint policy. Thus the necessary condition for a joint policy to be optimal is for it to be locally optimal. A locally optimal joint policy is one whose value cannot increase if the policy of only one agent (any agent) in it is changed. A joint policy (p_1, p_2) is locally optimal if there holds,

$$\begin{aligned} V(\alpha, p_1, p_2) &\geq V(\alpha, \hat{p}_1, p_2), \quad \forall \hat{p}_1 \neq p_1 \\ V(\alpha, p_1, p_2) &\geq V(\alpha, p_1, \hat{p}_2), \quad \forall \hat{p}_2 \neq p_2 \end{aligned}$$

Due to this condition, each policy in a locally optimal joint policy is called a best response to the remaining policies in the joint policy. The necessary condition for a joint policy to be optimal is that each policy in it be a best response (to the remaining policies). Hence, our problem reduces to deriving the mathematical expression for a joint policy in which each policy is a best response to the remaining policies.

The derivation of this expression can be approached from two fronts as mentioned in the introduction, by using the theorem of linear programming duality or alternatively by using the theory of Lagrange multipliers in the form of the Kuhn-Tucker theorem. We shall use linear programming duality. We begin the derivation by taking up the simpler case, that of finding a policy that is a best response to given policies of other agents. We consider the three-agent case. Let p_2 and p_3 be policies of agents 2 and 3 respectively. A policy p_1 of agent 1 is a best response to these policies if there holds,

$$V(\alpha, p_1, p_2, p_3) \geq V(\alpha, \hat{p}_1, p_2, p_3), \quad \forall \hat{p}_1 \neq p_1$$

A best response policy x_1 to (p_2, p_3) can be found by solving the following primal linear program (LP),

$$\text{Maximize} \quad \sum_{h_1 \in \mathcal{E}_1} \sum_{h_2 \in \mathcal{E}_2} \sum_{h_3 \in \mathcal{E}_3} r(\alpha, h_1, h_2, h_3) x_1(h_1) p_2(h_2) p_2(h_3)$$

Subject To,

$$\begin{aligned} \sum_{a \in A_1} x_1(a) &= 1 \\ -x_1(h) + \sum_{a \in A_1} x_1(hoa) &= 0, \quad \forall h \in \mathcal{N}_1, \forall o \in O_1 \\ x_1(h) &\geq 0, \quad \forall h \in \mathcal{H}_1 \end{aligned}$$

Every LP has a dual LP which solves a converse problem than the LP itself. The LP is called the primal to distinguish it from its dual. If the LP maximizes a function, the dual minimizes a function. In order to state the dual of the above LP, we shall require three concepts: an information set, the value of an information set and the regret of a history.

An information set ι of agent i is a sequence of even length in which the elements in odd positions are actions of the agent (members of A_i) and those in even positions are observations of the agent (members of O_i). Thus, despite its name, an information set is not a set but a sequence. An information set is so called because given an information set ι we can use it to group all possible joint histories that can occur from the agent's perspective given ι. In other words, ι circumscribes the agent's knowledge about which joint history may occur at the end of that period. The number of actions in an information set shall be called its length. An information set of length $t \geq 0$ has t actions and t observations. An information set of length 0 shall be called the null information set, denoted by $\varnothing$. An information set of length T - 1 shall be called a terminal information set. Information sets of lengths less than T - 1 shall be called nonterminal information sets. The information set of a history h, denoted by $\iota(h)$, is the information set obtained by dropping the last action in the history. Thus, history h is said to belong to information set $\iota(h)$. The set of information sets of length t of agent i shall be denoted by $\mathcal{I}_i^t$. The set of information sets of lengths less than or equal to T - 1 shall be denoted by $\mathcal{I}_i$.

The value of an information set of an agent is a relative term. It is a function of the policies of other agents. Given policies p_2 and p_3 of agents 2 and 3, the value $\lambda_1^*(\iota, p_2, p_3)$ of an information set ι of agent 1 is defined as follows. If ι is a terminal information set, then

$$\lambda_1^*(\iota, p_2, p_3) = \max_{h \in \iota} \sum_{h_2 \in \mathcal{E}_2} \sum_{h_3 \in \mathcal{E}_3} r(\alpha, (h, h_2, h_3)) p_2(h_2) p_3(h_3)$$

and if ι is a nonterminal information set then,

$$\lambda_1^*(\iota, p_2, p_3) = \max_{h \in \iota} \sum_{o \in O_1} \lambda_1^*(ho, p_2, p_3)$$

where ho denotes the information set obtained on concatenating o to h.

Then, the regret $\mu_1(h, p_2, p_3)$ of a history h of agent 1 is defined as follows. If h is a terminal history then,

$$\mu_1(h, p_2, p_3) = \lambda_1^*(\iota(h), p_2, p_3) - \sum_{h_2 \in \mathcal{E}_2} \sum_{h_3 \in \mathcal{E}_3} r(\alpha, (h, h_2, h_3)) p_2(h_2) p_3(h_3)$$

and if h is a nonterminal history then,

$$\mu_1(h, q_{-i}) = \lambda_1^*(\iota(h), p_2, p_3) - \sum_{o \in O_1} \lambda_1^*(ho, p_2, p_3)$$

Notice that by definition, the regret of a history cannot be a negative number. It is either 0 or greater than 0.

We can now state the dual of the primal LP described above. It is as follows.

$$\text{Minimize} \quad y_1(\varnothing)$$

Subject To,

$$\begin{aligned} y_1(\iota(h)) - \sum_{o \in O_1} y_1(ho) &\geq 0, \quad \forall h \in \mathcal{N}_1 \\ y_1(\iota(h)) - \sum_{h_2 \in \mathcal{E}_2} \sum_{h_3 \in \mathcal{E}_3} r(\alpha, (h, h_2, h_3)) p_2(h_2) p_3(h_3) &\geq 0, \quad \forall h \in \mathcal{E}_1 \\ y_1(\iota) &\in [-\infty, +\infty], \quad \forall \iota \in \mathcal{I}_1 \end{aligned}$$

where $\iota(h)$ denotes the information set to which h belongs. The dual has one free variable $y_1(\iota)$ for every information set of agent 1. It has one constraint per history of the agent. Note that the objective of the dual is to minimize only $y_1(\varnothing)$ because in the primal LP, the right-hand side of all the constraints except the very first one, is a 0. If we consider the definition of the value of an information set, we can see that the dual finds the value of every information set of agent i given p_2 and p_3. That is, for each $\iota \in \mathcal{I}_1$, $y_1^*(\iota)$ is the value of information set ι given p_2 and p_3.

Before applying the theorem of LP duality to the primal-dual pair given above, we will transform the dual slightly by introducing surplus variables that modify the inequality constraints into equality constraints. The transformed dual with these surplus variables is as follows:

$$\text{Minimize} \quad y_1(\varnothing)$$

Subject To,

$$\begin{aligned} y_1(\iota(h)) - \sum_{o \in O_1} y_1(ho) &= w_1(h), \quad \forall h \in \mathcal{N}_1 \\ y_1(\iota(h)) - \sum_{h_2 \in \mathcal{E}_2} \sum_{h_3 \in \mathcal{E}_3} r(\alpha, (h, h_2, h_3)) p_2(h_2) p_3(h_3) &= w_1(h), \quad \forall h \in \mathcal{E}_1 \\ y_1(\iota) &\in [-\infty, +\infty], \quad \forall \iota \in \mathcal{I}_1 \\ w_1(h) &\geq 0, \quad \forall h \in \mathcal{H}_1 \end{aligned}$$

As can be seen, the transformed dual is identical to the dual except that we have now used a variable $w_1(h)$ for each history h of agent 1; $w_1(h)$ represents the regret of history h. A solution of the transformed dual consists of values of information sets and regrets of histories. That is, for each $\iota \in \mathcal{I}_1$, $y_1^*(\iota)$ is the value of information set ι given p_2 and p_3 and for each $h \in \mathcal{H}_1$, $w_1^*(h)$ is the regret of history h given p_2 and p_3.

We shall now apply the theorem of linear programming duality[8] to the primal LP and the transformed dual LP. According to this theorem, if either a primal LP or its dual LP has a finite optimal solution, then so does the other, and the corresponding values of the objective functions are equal. Therefore, the first relationship we obtain on applying this theorem is that,

$$\sum_{h_1 \in \mathcal{E}_1} \sum_{h_2 \in \mathcal{E}_2} \sum_{h_3 \in \mathcal{E}_3} r(\alpha, (h_1, h_2, h_3)) x_1^*(h_1) p_2(h_2) p_3(h_3) = y_1^*(\varnothing)$$

Thus, the value of the joint policy (x_1^*, p_2, p_3) can be expressed either as,

$$V(\alpha,(x_1^*,p_2,p_3)) \quad = \quad \sum_{h_1\in\mathcal{E}_1}\sum_{h_2\in\mathcal{E}_2}\sum_{h_3\in\mathcal{E}_3} r(\alpha,(h_1,h_2,h_3))x_1^*(h_1)p_2(h_2)p_3(h_3)$$

or as,

$$V(\alpha,(x_1^*,p_2,p_3)) \quad = \quad y_1^*(\varnothing)$$

The second relationship we obtain (between weights of histories and values of information sets and regrets of histories) constitutes the necessary condition for optimality. Due to the constraints of the primal LP, there holds

$$y_1^*(\varnothing) \quad = \quad y_1^*(\varnothing)\big\{\sum_{a\in A_1} x_1^*(a)\big\} + \sum_{h\in\mathcal{N}_1}\sum_{o\in O_1} y_1^*(ho)\big\{-x_1^*(h) + \sum_{a\in A_i} x_1^*(hoa)\big\}$$

The above equation obtains because in the primal, the first term in the braces is 1 and each of the remaining terms in braces is 0. The right hand side of the above equation can be rewritten as,

$$\begin{aligned}
&\sum_{a\in A_1} x_1^*(a)\big\{y_1^*(\varnothing) - \sum_{o\in O_1} y_1^*(ao)\big\} + \\
\sum_{h\in\mathcal{N}_1\setminus A_1} x_1^*(h)\big\{y_1^*(\iota(h)) - \sum_{o\in O_1} y_1^*(ho)\big\} + \sum_{h\in\mathcal{E}_1} x_1^*(h)y_1^*(\iota(h)) \quad &= \\
\sum_{h\in\mathcal{N}_1} x_1^*(h)\big\{y_1^*(\iota(h)) - \sum_{o\in O_1} y_1^*(ho)\big\} + \sum_{h\in\mathcal{E}_1} x_1^*(h)y_1^*(\iota(h))&
\end{aligned}$$

Further, we have,

$$\begin{aligned}
\sum_{h\in\mathcal{E}_1}\big\{\sum_{h_2\in\mathcal{E}_2}\sum_{h_3\in\mathcal{E}_3} r(\alpha,(h,h_2,h_3))p_2(h_2)p_3(h_3)\big\}x_1^*(h) \quad &= \\
\sum_{h\in\mathcal{N}_1} x_1^*(h)\big\{y_1^*(\iota(h)) - \sum_{o\in O_1} y_1^*(ho)\big\} + \sum_{h\in\mathcal{E}_1} x_1^*(h)y_1^*(\iota(h))&
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\sum_{h\in\mathcal{N}_1} x_1^*(h)\big\{y_1^*(\iota(h)) - \sum_{o\in O_1} y_1^*(ho)\big\} + \\
\sum_{h\in\mathcal{E}_1} x_1^*(h)\big\{y_1^*(\iota(h)) - \sum_{h_2\in\mathcal{E}_2}\sum_{h_3\in\mathcal{E}_3} r(\alpha,(h,h_2,h_3))p_2(h_2)p_3(h_3)\big\} \quad &= \quad 0
\end{aligned}$$

Due to the constraints of the transformed dual, we can furthermore rewrite the above equation as simply,

$$\begin{aligned}
\sum_{h\in\mathcal{N}_1} x_1^*(h)w_1^*(h) + \sum_{h\in\mathcal{E}_1} x_1^*(h)w_1^*(h) \quad &= \quad 0 \\
\sum_{h\in\mathcal{H}_1} x_1^*(h)w_1^*(h) \quad &= \quad 0
\end{aligned}$$

This is a sum of n_1 products, n_1 being the size of $\mathcal{H}_1$. Each product in this sum is necessarily 0 because both $x_1(h)$ and $w_1(h)$ are constrained to be nonnegative in the primal and the dual respectively. Hence, this sum is equivalent to,

$$x_1^*(h)w_1^*(h) \quad = \quad 0, \quad \forall h\in\mathcal{H}_1 \tag{2}$$

Each equation $x_1^*(h)w_1^*(h)$ is called a complementarity constraint or an equilibrium constraint. Each complementarity constraint implies that either the weight of h be zero or its regret be 0. Both cannot be false at the same time. Therefore, we can conclude that the necessary condition for a policy p_1 of agent 1 to be a best response policy to policies p_2 and p_3 is that,

$$p_1(h)\mu_i(h, p_2, p_3) \quad = \quad 0, \quad \forall h \in \mathcal{H}_1$$

The necessary condition for joint policy to be an optimal joint policy is obtained by generalizing the reasoning of obtaining the necessary condition for a best response policy. That is, we must assume that each agent simultaneously with the other agents attempts to find a policy that is a best response to the policies of the other agents' policies. This implies that we must set up and solve simultaneously n pairs of primal-dual linear programs. The generalization yields the required necessary condition. In the three-agent case, the necessary condition for a joint policy (p_1, p_2, p_3) to be optimal is that,

$$\begin{aligned} p_1(h_1)\mu_1(h_1, p_2, p_3) &= 0, \quad \forall h_1 \in \mathcal{H}_1 \\ p_2(h_2)\mu_1(h_2, p_1, p_3) &= 0, \quad \forall h_2 \in \mathcal{H}_2 \\ p_3(h_3)\mu_1(h_3, p_1, p_2) &= 0, \quad \forall h_3 \in \mathcal{H}_3 \end{aligned}$$

The necessary condition obtained above can be transformed to the following non-linear program for finding an optimal 3-agent joint policy.

$$\text{Maximize} \quad y_1(\varnothing)$$

Subject To,

$$\begin{aligned} \sum_{a \in A_i} x_i(a) &= 1, \quad i = 1, 2, 3 \\ -x_i(h) + \sum_{a \in A_i} x_i(hoa) &= 0, \quad i = 1, 2, 3, \forall h \in \mathcal{N}_i, \forall o \in O_i \\ y_i(\iota(h)) - \sum_{o \in O_i} y_i(ho) &= w_i(h), \quad i = 1, 2, 3, \forall h \in \mathcal{N}_i \\ y_1(\iota(h_1)) - \sum_{h_2 \in \mathcal{E}_2} \sum_{h_3 \in \mathcal{E}_3} r(\alpha, (h_1, h_2, h_3)) x_2(h_2) x_3(h_3) &= w_1(h_1), \quad \forall h_1 \in \mathcal{E}_1 \\ y_2(\iota(h_2)) - \sum_{h_1 \in \mathcal{E}_1} \sum_{h_3 \in \mathcal{E}_3} r(\alpha, (h_1, h_2, h_3)) x_1(h_1) x_3(h_3) &= w_2(h_2), \quad \forall h_2 \in \mathcal{E}_2 \\ y_3(\iota(h_3)) - \sum_{h_1 \in \mathcal{E}_1} \sum_{h_2 \in \mathcal{E}_2} r(\alpha, (h_1, h_2, h_3)) x_1(h_1) x_2(h_2) &= w_3(h_3), \quad \forall h_3 \in \mathcal{E}_3 \\ x_i(h) w_i(h) &= 0, \quad i = 1, 2, 3, \forall h \in \mathcal{H}_i \\ x_i(h) &\geq 0, \quad i = 1, 2, 3, \forall h \in \mathcal{H}_i \\ w_i(h) &\geq 0, \quad i = 1, 2, 3, \forall h \in \mathcal{H}_i \\ y_i(\iota) &\in [-\infty, +\infty], \quad i = 1, 2, 3, \forall \iota \in \mathcal{I}_i \end{aligned}$$

We shall henceforth refer to this NLP as **N'**. Its 2-agent version or its generalization to more than 3 agents is easily obtained. Given a global maximum point (x^*, y^*, w^*) of this NLP, x^* = (x_1^*, x_2^*, x_3^*) is an optimal 3-agent joint policy. Note that $y_1(\varnothing)$ in the objective can be replaced by $y_2(\varnothing)$ or $y_2(\varnothing)$.

We enumerate the constraints of **N'** linewise as follows. The first two lines define the policies of the three agents. The next line defines the regrets of nonterminal histories of the three agents. The next three lines define the regrets of the terminal histories of the three agents. The next line defines the complementarity condition for the three agents. The remaining lines define the domains for the variables.

When the number of agents n = 2, the constraints of **N'** constitute a linear complementarity problem (LCP), and when $n > 2$, they constitute a nonlinear complementarity problem (NLCP). When n = 2, every constraint in **N'** is a linear constraint with the exception of the complementarity constraints, since they remain quadratic even in that case. When $n > 2$, every constraint in **N'** is a linear constraint with the exception of the complementarity constraints and the constraints defining the regrets of terminal histories since they contain a sum of nonlinear terms.

Due to the presence of nonlinear constraints in **N'** for both n = 2 and $n > 2$, we are not guaranteed to find a global maximum point of the program by solving the program directly. The program must be linearized. When n = 2, only the complementarity constraints must be linearized. When $n > 2$, the complementarity constraints as well as the nonlinear term appearing in constraints defining regrets of terminal histories must be linearized. The linearization of the complementarity constraints is the same be n = 2 or $n > 2$. This linearization is achieved by replacing each complementarity constraint by a pair of linear inequalities which use a common 0-1 variable. The result is a 0-1 MILP. For the $n > 2$ case, the conversion of the complementarity constraint must be followed by the linearization of the aforementioned nonlinear term (whose linerization is facilitated by the use of more variables and constraints). This results in a different 0-1 MILP. In the next section, we present both these 0-1 MILPs.

5 Conversion to 0-1 MILP

Consider a complementarity constraint ab = 0 in variables a and b. Assume that the lower bound on the values of a and b is 0. Let the upper bounds on the values of a and b be respectively u_a and u_b. Now let c be a 0-1 variable. That is, c is allowed to assume a value of either 0 or 1. Then, the complementarity constraint ab = 0 can be separated into the following equivalent pair of linear constraints,

$$a \leq u_a c \tag{3}$$
$$b \leq u_b(1-c) \tag{4}$$

In other words, if this pair of constraints is satisfied, then it is surely the case that $ab = 0$. This is easily verified. c can either be 0 or 1. If c = 0, then a will be set to 0 because a is constrained to be not more than $u_a c$ (and not less than 0); if c = 1, then b will be set to 0 since b is constrained to be not more than $u_b(1-c)$ (and not less than 0). In either case, $ab = 0$.

Now consider each complementarity constraint $x_i(h)w_i(h)$ = 0 from **N'**. We wish to separate this constraint into a pair of linear constraints. We recall that $x_i(h)$ represents the weight of h and $w_i(h)$ represents the regret of h. The first requirement to convert this constraint to a pair of linear constraints is that the lower bound on the values of the two terms be 0. This is indeed the case since $x_i(h)$ and $w_i(h)$ are both constrained to be nonnegative. Next, we require upper bounds on the weights of histories and regrets of histories. The upper bound on the value of $x_i(h)$ for each h is 1. For the upper bounds on the regrets of histories, we require some notation. An upper bound

of the regret of a history of an agent is the maximum value the regret of the history can assume regardless of the policies of the other agents. We denote the upper bound on the regret of a history h of agent i by $\mathcal{U}_i(h)$.

Given an upper bound $\mathcal{U}_i(h)$ on the regret of a history h of agent i, the complementarity constraint $x_i(h)w_i(h) = 0$ can separated into a pair of linear constraints by using a 0-1 variable $b_i(h)$ as follows,

$$\begin{aligned} x_i(h) &\leq 1 - b_i(h) \\ w_i &\leq \mathcal{U}_i(h)b_i(h) \\ b_i(h) &\in \{0,1\} \end{aligned}$$

Upper bounds on the regrets of histories are defined as follows. We state the definitions for agent 1; they apply by analogy to agents 2 and 3. In any policy p_1 of agent 1 there holds,

$$\sum_{h_1 \in \mathcal{E}_1} p_1(h) = |O_1|^{T-1}$$

Similarly, for any given polices p_2 and p_3 of agents 2 and 3, there holds,

$$\sum_{h_2 \in \mathcal{E}_2} \sum_{h_3 \in \mathcal{E}_3} p_2(h_2)p_3(h_3) = (|O_2||O_3|)^{T-1}$$

Since the regret of a terminal history h_1 of agent 1 given p_2 and p_3 is defined as,

$$\mu_1(h_1, p_2, p_3) = \max_{h \in \iota(h_1)} \sum_{h_2 \in \mathcal{E}_2} \sum_{h_3 \in \mathcal{E}_3} p_2(h_2)p_3(h_3)\big\{r(\alpha,(h,h_2,h_3)) - r(\alpha,(h_1,h_2,h_3))\big\}$$

we can conclude that the upper bound $\mathcal{U}_i(h_1)$ on the regret of a terminal history h_1 of agent 1 is,

$$\mathcal{U}_1(h_1) = (|O_2||O_3|)^{T-1}\{\max_{h \in \iota(h_1)} \max_{h_2 \in \mathcal{E}_2} \max_{h_3 \in \mathcal{E}_3} r(\alpha,(h,h_2,h_3)) - \min_{h'_2 \in \mathcal{E}_2} \min_{h'_3 \in \mathcal{E}_3} r(\alpha,(h,h'_2,h'_3))\}$$

Now let us consider the upper bounds on the regrets of nonterminal histories. Let ι be an information set of length t of agent 1. Let $\mathcal{E}_1(\iota) \subset \mathcal{E}_1$ denote the set of terminal histories of agent 1 such the first $2t$ elements of each history in the set are identical to ι. Let h be a history of length $t < T$ of agent 1. Let $\mathcal{E}_1(h) \subset \mathcal{E}_1$ denote the set of terminal histories such that the first $2t$ - 1 elements of each history in the set are identical to h. Since in any policy p_1 of agent 1, there holds,

$$\sum_{h' \in \mathcal{E}_1(h)} p_1(h') \leq |O_1|^{T-t}$$

we can conclude that the upper bound $\mathcal{U}_1(h_1)$ on the regret of a nonterminal history h_1 of length t agent 1 is,

$$\mathcal{U}_1(h_1) = L_1\{\max_{h \in \mathcal{E}_1(\iota(h_1))} \max_{h_2 \in \mathcal{E}_2} \max_{h_3 \in \mathcal{E}_3} r(\alpha,(h,h_2,h_3)) - \min_{h' \in \mathcal{E}_1(h_1)} \min_{h'_2 \in \mathcal{E}_2} \min_{h'_3 \in \mathcal{E}_3} r(\alpha,(h',h'_2,h'_3))\}$$

where,

$$L_1 = |O_1|^{T-t}(|O_2||O_3|)^{T-1}$$

Notice that if $t = T$ (that is, h_1 is terminal) this definition reduces to the definition of the upper bound on the regret of a terminal history.

With the definitions of upper bounds on regrets of histories in place, we can convert **N'** in the 2-agent case to a 0-1 MILP by linearizing all the complementarity constraints in it. This results in the following 0-1 MILP.

$$\text{Maximize} \quad y_1(\varnothing)$$

Subject To,

$$\begin{aligned}
\sum_{a \in A_i} x_i(a) &= 1, \quad i = 1, 2 \\
-x_i(h) + \sum_{a \in A_i} x_i(hoa) &= 0, \quad i = 1, 2, \forall h \in \mathcal{N}_i, \forall o \in O_i \\
y_i(\iota(h)) - \sum_{o \in O_i} y_i(ho) &= w_i(h), \quad i = 1, 2, \forall h \in \mathcal{N}_i \\
y_1(\iota(h_1)) - \sum_{h_2 \in \mathcal{E}_2} r(\alpha, (h_1, h_2)) x_2(h_2) &= w_1(h_1), \quad \forall h_1 \in \mathcal{E}_1 \\
y_2(\iota(h_2)) - \sum_{h_1 \in \mathcal{E}_1} r(\alpha, (h_1, h_2) x_1(h_1) &= w_2(h_2), \quad \forall h_2 \in \mathcal{E}_2 \\
x_i(h) &\leq 1 - b_i(h), \quad i = 1, 2, \forall h \in \mathcal{H}_i \\
w_i &\leq \mathcal{U}_i(h) b_i(h), \quad i = 1, 2, \forall h \in \mathcal{H}_i \\
x_i(h) &\geq 0, \quad i = 1, 2, \forall h \in \mathcal{H}_i \\
w_i(h) &\geq 0, \quad i = 1, 2, \forall h \in \mathcal{H}_i \\
y_i(\iota) &\in [-\infty, +\infty], \quad i = 1, 2, \forall \iota \in \mathcal{I}_i \\
b_i(h) &\in \{0, 1\}, \quad i = 1, 2, \forall h \in \mathcal{H}_i
\end{aligned}$$

We shall henceforth refer to this 0-1 MILP as **M'-2**. The variables of the program are the vectors x_i, w_i, b_i and y_i for each agent i. A solution (x^*, y^*, w^*, b^*) to **M'-2** consists of the following quantities. (i) An optimal joint policy $x^* = (x_1^*, x_2^*)$ which may be a stochastic. (ii) For each agent i = 1, 2, for each history $h \in \mathcal{H}_i$, $w_i^*(h)$, the regret of h given the policy x_{-i}^* of the other agent. (iii) For each agent i = 1, 2, for each information set $\iota \in \mathcal{I}_i$, $y_i^*(\iota)$, the value of ι given the policy x_{-i} of the other agent. (iv) For each agent i = 1, 2, the vector b_i^* simply tells us which histories receive a weight of 0 in x_i^*; each history h of agent i such that $b_i^*(h) = 1$ has zero weight in x_i^*. Note that we can replace $y_1(\varnothing)$ by $y_2(\varnothing)$ in the objective function without affecting the program. Given a solution (x^*, w^*, y^*, b^*) to **M'-2** $x^* = (x_1^*, x_2^*)$ is an optimal joint policy

Turning to the 3 (or more agents) case, the additions/changes to **N'** required to linearize it are as follows.

(i) We linearize the complementarity constraints as in the 2-agent case.

(ii) Doing so, the only nonlinear terms we are left with are those appearing in the constraints representing definitions of regrets of terminal histories. Each nonlinear term is a product of weights of terminal histories of a reduced terminal joint history (i.e., a joint history in which the history of one agent is absent). Therefore, for each agent, and for each reduced terminal joint history for that agent, we introduce a variable. Thus, in the 3-agent case, we introduce for agent 1 a variable $z_{23}(h_2, h_3)$, for each $h_2 \in \mathcal{E}_2$ and for each $h_3 \in \mathcal{E}_3$. Idem for the other two agents.

(iii) We place integer constraints on the x variables representing weights of terminal histories of each agent. This has the effect that the value of every x variable in a feasible solution to the program has a value of 0 or 1. In other words, every feasible solution to the program is a pure joint policy.

(iv) We introduce additional constraints that ensure in every solution to the program, for each $h_2 \in \mathcal{E}_2$ and for each $h_3 \in \mathcal{E}_3$, there holds

$$z^*_{23}(h_2, h_3) = 1 \quad \Leftrightarrow \quad x^*_2(h_2) = 1, x^*_3(h_3) = 1$$

With these additions/changes, **N'** changes to the following 0-1 MILP; the 3-agent case is given for clarity.

$$
\begin{aligned}
&\text{Maximize} && y_1(\varnothing) \quad \text{Subject To,}\\
\sum_{a\in A_i} x_i(a) &= 1, && i = 1,2,3\\
-x_i(h) + \sum_{a\in A_i} x_i(hoa) &= 0, && i = 1,2,3, \forall h \in \mathcal{N}_i, \forall o \in O_i\\
y_i(\iota(h)) - \sum_{o\in O_i} y_i(ho) &= w_i(h), && i = 1,2,3, \forall h \in \mathcal{N}_i\\
y_1(\iota(h_1)) - \sum_{h_2\in\mathcal{E}_2}\sum_{h_3\in\mathcal{E}_3} r(\alpha,(h_1,h_2,h_3))z_{23}(h_2,h_3) &= w_1(h_1), && \forall h_1 \in \mathcal{E}_1\\
y_2(\iota(h_2)) - \sum_{h_1\in\mathcal{E}_1}\sum_{h_3\in\mathcal{E}_3} r(\alpha,(h_1,h_2,h_3))z_{13}(h_1,h_3) &= w_2(h_2), && \forall h_2 \in \mathcal{E}_2\\
y_3(\iota(h_3)) - \sum_{h_1\in\mathcal{E}_1}\sum_{h_2\in\mathcal{E}_2} r(\alpha,(h_1,h_2,h_3))z_{12}(h_1,h_2) &= w_3(h_1), && \forall h_3 \in \mathcal{E}_3\\
\sum_{h_3\in\mathcal{E}_3} z_{23}(h_2,h_3) &= |O_3|^{T-1}x_2(h_2), && \forall h_2 \in \mathcal{E}_2\\
\sum_{h_2\in\mathcal{E}_2} z_{23}(h_2,h_3) &= |O_2|^{T-1}x_3(h_3), && \forall h_3 \in \mathcal{E}_3\\
\sum_{h_3\in\mathcal{E}_3} z_{13}(h_1,h_3) &= |O_3|^{T-1}x_1(h_1), && \forall h_1 \in \mathcal{E}_1\\
\sum_{h_1\in\mathcal{E}_1} z_{13}(h_1,h_3) &= |O_1|^{T-1}x_3(h_3), && \forall h_3 \in \mathcal{E}_3\\
\sum_{h_2\in\mathcal{E}_2} z_{12}(h_1,h_2) &= |O_2|^{T-1}x_1(h_1), && \forall h_1 \in \mathcal{E}_1\\
\sum_{h_1\in\mathcal{E}_1} z_{12}(h_1,h_2) &= |O_1|^{T-1}x_2(h_2), && \forall h_2 \in \mathcal{E}_2\\
\sum_{h_1\in\mathcal{E}_1}\sum_{h_2\in\mathcal{E}_2} z_{12}(h_1,h_2) &= (|O_1||O_2|)^{T-1}\\
\sum_{h_1\in\mathcal{E}_1}\sum_{h_3\in\mathcal{E}_3} z_{13}(h_1,h_3) &= (|O_1||O_3|)^{T-1}\\
\sum_{h_2\in\mathcal{E}_2}\sum_{h_3\in\mathcal{E}_3} z_{23}(h_2,h_3) &= (|O_2||O_3|)^{T-1}\\
x_i(h) &\le 1 - b_i(h), && i = 1,2,3, \forall h \in \mathcal{H}_i\\
w_i &\le \mathcal{U}_i(h)b_i(h), && i = 1,2,3, \forall h \in \mathcal{H}_i\\
x_i(h) &\ge 0, && i = 1,2,3, \forall h \in \mathcal{N}_i\\
x_i(h) &\in \{0,1\}, && i = 1,2,3, \forall h \in \mathcal{E}_i\\
w_i(h) &\ge 0, && i = 1,2,3, \forall h \in \mathcal{H}_i\\
y_i(\iota) &\in [-\infty,+\infty], && i = 1,2,3, \forall \iota \in \mathcal{I}_i\\
b_i(h) &\in \{0,1\}, && i = 1,2,3\, \forall h \in \mathcal{H}_i\\
z_{12}(h_1,h_2) &\in \{0,1\}, && \forall h_1 \in \mathcal{E}_1, \forall h_2 \in \mathcal{E}_2\\
z_{13}(h_1,h_3) &\in \{0,1\}, && \forall h_1 \in \mathcal{E}_1, \forall h_3 \in \mathcal{E}_3\\
z_{23}(h_2,h_3) &\in \{0,1\}, && \forall h_2 \in \mathcal{E}_2, \forall h_3 \in \mathcal{E}_3
\end{aligned}
$$

We shall henceforth refer to this program as **M'-3**. Given an optimal solution x^*, y^*, w^*, b^*, z^*) to this program, x^* = (x_1^*, x_2^*, x_3^*) constitutes an optimal T-period 3-agent joint policy.

The size of a program is given in the number of variables and constraints it has. In the case of a 0-1 MILP, the number of 0-1 variables in it is an additional and separate factor since the time taken to solve a 0-1 MILP through the branch and bound (BB) method is dependent on this number. In the 2-agent case, the total number of variables in **M** is exponentially higher than in **M'-2**. However, it has less 0-1 variables and less constraints than **M'-2**. So while the latter is overall a smaller program, it may not necessarily be faster to solve than the former. In the 3-agent case, again, we find that **M** has more variables than **M'-3** making it overall larger than **M'-3**, but the latter has more 0-1 variables and more constraints implying that its smaller size may not result in a lesser runtime. The computational experience of the two programs confirms this mixed picture.

6 Computational experience

We tested the two programs on different instances of three problems. Two of these are wellknown in the literature and are known as the multi-access broadcast channel (MABC) problem and the multi-agent tiger (MA-tiger) problem. The third problem consists of testing on a randomly generated Dec-POMDP. This problem has not yet been used by existing algorithms for testing purposes. MABC has 2 agents, 4 states, 2 actions and 2 observations per agent. MA-tiger has 2 agents, 2 states, 3 actions and 2 observations per agent. Two versions of the random problem were tested. We shall call these R1 and R2 respectively. R1 has 50 states, 2 actions and 2 observations per agent. R2 has 3 actions and 2 observations per agent. Of all the problems, only R1 was tested for the 3-agent case as well since R2 could not formulated in memory.

The time taken in seconds to solve these problems for the 2-agent case using **M**, **M'-2** as well as algorithms of the first group (those that are capable of solving these problems in reasonable time) is given in Table 1. We recall that algorithms of the first group are DP[5], MAA*[13], PBDP[12], GMAA*[9] and DP-LPC[3]. **M** and **M'-2** were solved using the branch and bound (BB) method implemented by the ILOG Cplex software. Note that the runtimes of the algorithms of the first group are given as reported in the literature - we have not implemented these algorithms ourselves. We kept a timeout of 1800 seconds (half an hour) when solving **M'-2** through the BB method.

As stated, **M** and **M'-3** were also tested on random problems, R1 and R2. Table 2 shows the runtimes of the two programs on 2-agent instances of R1 and R2. The times are averaged over 10 runs. Finally , **M** and **M'-3** were also tested on the 3-agent random problem R1. The times taken to solve instances of R1 using **M** and **M'-3** for the 3-agent, horizon 3 case are given in Table 3.

These results indicate that **M'** is not as fast as **M** on most of the problems tested, neither in the 2-agent case nor in the 3-agents case; only on R1 does it seem to have an average runtime that is lesser than **M**. Unlike **M**, the runtimes reported for **M'** are for the case where no heuristics have been introduced in it (such as pruning of dominated histories[1]). We surmise that the runtimes of **M'** can be vastly improved with the aid of such heuristics. We are currently working on incorporating heuristics into **M'**; this is not as straightforward as is the case with **M**.

Problem	Horizon T	Algorithm	Time taken (secs)
MABC	4	**M**	10.20
		M'-2	3.53
		GMAA*	0.03
		PBDP	2.00
		DP-LPC	4.59
		DP	900.00
		MAA*	5400.00
	5	**M**	25.00
		M'-2	$>$1800.00
		GMAA*	5.15
MA-Tiger	3	**M**	6.20
		M'-2	11.16
		GMAA*	0.02
		DP-LPC	1.79
		MAA*	4.00
	4	**M**	72.00
		M'-2	$>$1800.00
		DP-LPC	535.00
		GMAA*	3208.00

Table 1: 2-agent problems

Problem	Horizon T	Algorithm	Least time (secs)	Most time (secs)	Average	Std. deviation
R1	4	**M**	2.45	455	120.6	183.48
		M'-2	6.85	356	86.88	111.56
R2	3	**M**	1.45	10.46	4.95	3.98
		M'-2	5.06	12.53	7.28	2.43

Table 2: 2-agent problems

Algorithm	Least time (secs)	Most time (secs)	Average	Std. deviation
M	26	90	53.2	24.2
M'-3	754	2013	1173	715

Table 3: 3-agent problems

7 Conclusions

We have presented in this paper a new algorithm - a 0-1 MILP **M'** - for solving finite horizon Dec-POMDPs. The motivation for developing and implementing this 0-1 MILP rose on two accounts. First, an existing 0-1 MILP[1] **M** gave excellent results on standard Dec-POMDP problems such as the MABC and the MA-tiger problems. It solved these problems much faster than dynamic programming algorithms[5],[13],[12], indicating that perhaps mathematical programming was a more effective approach to solving Dec-POMDPs than dynamic programming. Second, by using linear programming duality, for the 2-agent case, a 0-1 MILP that is much smaller than **M** obtains. It was thought that the smaller size of such a program would translate into a much smaller runtime as well.

However, the computational experience has been mixed. The smaller size of the proposed program **M'** has not translated into a program that is always faster than **M**, or for that matter, the existing dynamic programming algorithms. In fact, it is generally slower than **M** while being generally as fast or faster than the dynamic programming algorithms.

The reason for the generally sluggish performance of **M'** could be two-fold. First, it is a much more complicated program than **M**. The latter is a simple program, attempting to find an optimal combination of histories, one that maximizes value. **M'** attempts to minimize total regret while maximizing value; this burdens it by having to find the values of all information sets as well as the regrets of all the histories. In other words, **M'** finds many more quantities besides an optimal joint policy while **M** merely finds an optimal joint policy. This can be thought to be the main factor slowing down **M'**. Second, because **M'** attempts to find all histories with zero regret, it has many more 0-1 variables than **M**. This gives the BB method a larger palette to choose a branching variable from when solving **M'** compared to when solving **M**.

As stated in the previous section, we conjecture that the performance of **M'** can be vastly improved by using heuristics such as pruning of dominated histories. This is our current endeavour.

Finally, we note that **M'** posseses a distinct advantage over **M** in that it can be used to find a Nash equilibrium in a partially observable stochastic game (POSG) while **M** cannot be.

References

[1] Raghav Aras, Alain Dutech, and François Charpillet. Mixed Integer Linear Programming For Exact Finite-Horizon Planning In Decentralized POMDPs. *In Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling (ICAPS 2007)*, 2007.

[2] Daniel Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The Complexity Of Decentralized Control Of Markov Decision Processes. *Mathematics of Operations Research*, 27(4):819 – 840, 2002.

[3] Abdeslam Boularias and Brahim Chaib-draa. Exact Dynamic Programming For Decentralized POMDPs with Lossless Policy Compression. *In Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2008)*, 2008.

[4] A.R. Cassandra. A Survey Of POMDP Applications. 1998.

[5] Eric Hansen, Daniel Bernstein, and Shlomo Zilberstein. Dynamic Programming For Partially Observable Stochastic Games. *In Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI 2004)*, pages 709 – 715, 2004.

[6] Daphne Koller and Nimrod Megiddo. The Complexity of Zero-Sum Games in Extensive Form. *Games and Economic Behavior*, 4:4:528–552, 1992.

[7] Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Fast Algorithms for Finding Randomized Strategies in Game Trees. *Proceedings of the 26th ACM Symposium on Theory of Computing (STOC '94)*, pages 750–759, 1994.

[8] D.G. Luenberger. *Linear and Nonlinear Programming*. Addion-Wesley Publishing Company, Reading, Massachussetts, 1984.

[9] Frans A. Oliehoek, Matthijs T.J. Spaan, and Nikos Vlassis. Optimal and Approximate Q-value Functions for Decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32:289 – 353, 2008.

[10] Christos H. Papadimitriou and John Tsitsiklis. The Complexity Of Markov Decision Processes. *Mathematics of Operations Research*, 12 (3):441 – 450, 1987.

[11] R.D. Smallwood and E.J. Sondik. The Optimal Control Of Partially Observable Markov Processes Over A Finite Horizon. *Operations Research*, 21(5):1071 – 1088, 1973.

[12] Daniel Szer and François Charpillet. Point-based Dynamic Programming for DEC-POMDPs. *In Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI 2006)*, 2006.

[13] Daniel Szer, François Charpillet, and Shlomo Zilberstein. MAA*: A Heuristic Search Algorithm For Solving Decentralized POMDPs. *In Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI 2005)*, pages 576 – 583, 2005.

Centre de recherche INRIA Nancy – Grand Est
LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Centre de recherche INRIA Bordeaux – Sud Ouest : Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex
Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifi que de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-0803