Integrated Cooperation and Competition in Multi-Agent Decision-Making

Kyle Hollins Wray[1]   **Akshat Kumar**[2]   **Shlomo Zilberstein**[1]

[1] College of Information and Computer Sciences, University of Massachusetts, Amherst, MA, USA
[2] School of Information Systems, Singapore Management University, Singapore

Abstract

Observing that many real-world sequential decision problems are not purely cooperative or purely competitive, we propose a new model—cooperative-competitive process (CCP)—that can simultaneously encapsulate both cooperation and competition. First, we discuss how the CCP model bridges the gap between cooperative and competitive models. Next, we investigate a specific class of group-dominant CCPs, in which agents cooperate to achieve a common goal as their primary objective, while also pursuing individual goals as a secondary objective. We provide an approximate solution for this class of problems that leverages stochastic finite-state controllers. The model is grounded in two multi-robot meeting and box-pushing domains that are implemented in simulation and demonstrated on two real robots.

Introduction

Multi-agent models for sequential decision making under uncertainty often fall into one of two distinct categories. They are either considered *cooperative*—when all the agents have a shared objective, or *competitive*—when each agent has a distinct private objective. Rich computational models and algorithms have been developed for each category, particularly the decentralized partially observable Markov decision process (Dec-POMDP) (Bernstein et al. 2002) for the cooperative case and the partially observable stochastic game (POSG) (Hansen, Bernstein, and Zilberstein 2004) for the competitive case. These models have proved useful for various applications such as cooperative Mars rovers (Becker et al. 2004) or competitive robotic tag (Emery-Montemerlo et al. 2004).

Real-world situations, however, tend to require a delicate balance between cooperation, focusing on societal interests, and competition, focusing on self-interest. For example, autonomous vehicles (Wray, Witwicki, and Zilberstein 2017) must cooperate with other vehicles while optimizing the resources needed to reach their own destination. Similarly, a personal assistant must be able to cooperatively schedule meetings that respect group preferences while optimizing the objectives of the user they represent (Varakantham, Maheswaran, and Tambe 2005). In fact, pure cooperation and pure competition are rare in real-world settings.

The overarching goal of this work is to formalize the notion of a *cooperative-competitive process* (CCP) by generalizing Dec-POMDPs and POSGs and viewing them as extreme points of a broad spectrum.

Work on integrated cooperative and competitive processes has been sparse. While single-agent multi-objective models have enjoyed recent interest (Roijers et al. 2013), multi-agent models have primarily focused either on a single objective for all agents (Bernstein et al. 2002; Kumar, Zilberstein, and Toussaint 2011; Nair et al. 2003) or a single objective for each agent (Monderer and Shapley 1996; Emery-Montemerlo et al. 2004; Gmytrasiewicz and Doshi 2005). TAEMS (Decker 1996) and evolutionary-based multi-agent systems (Cardon, Galinho, and Vacher 2000; Coello, Lamont, and Van Veldhuizen 2007) support multiple objectives, but they do not address sequential decision-making and partial observability. Semi-autonomous systems allow control to be transferred back and forth between an agent and a human (Wray, Pineda, and Zilberstein 2016), but do not include slack or competition. Vector-valued Dec-(PO)MDPs consider solving individual objectives, which encapsulate both the agent and group objectives, and then apply techniques to fuse policies to satisfy overall social welfare (Mouaddib, Boussard, and Bouzid 2007; Matignon, Jeanpierre, and Mouaddib 2012). This online approach does not consider a characterization of slack or game theory. Multi-objective distributed constraint optimization (MO-DCOP) and other subclasses extend DCOP problems to multiple objectives, but focus primarily on methods for scalarization into a single objective (Delle Fave et al. 2011; Roijers, Whiteson, and Oliehoek 2015).

Our primary contributions are: (1) the CCP model that seamlessly blends the established cooperative and competitive multi-agent models, allowing for a tunable "slack" parameter to control the amount of cooperation versus competition; (2) a novel approximate algorithm for the model using stochastic finite-state controllers (FSCs) that builds upon nonlinear programming (NLP) techniques for Dec-POMDPs (Amato, Bernstein, and Zilberstein 2010) and best-response dynamics for POSGs (Nair et al. 2003); (3) a formal analysis of CCP's generality and the correctness of the proposed algorithm; and (4) experimental evidence using robots in real world experiments to demonstrate the range of multi-agent interactions that the model captures.

Cooperative-Competitive Processes

We define the notion of a cooperative-competitive process that generalizes Dec-POMDPs and POSGs. Thanks to the use of a slack parameter, the new model can smoothly blend cooperation and competition, and capture a broad spectrum of scenarios.

Definition 1. A **cooperative-competitive process (CCP)** is represented by $\langle I, S, \vec{A}, \vec{\Omega}, T, O, \vec{R} \rangle$.

- $I = \{1, \ldots, n\}$ is a set of n agents.

- S is a set of states.

- $\vec{A} = A_1 \times \cdots \times A_n$ is a set of joint actions.

- $\vec{\Omega} = \Omega_1 \times \cdots \times \Omega_n$ is a set of joint observations.

- $T : S \times \vec{A} \times S \to [0,1]$ is a state transition function mapping factored state s and joint action a to successor state s' such that $T(s, \vec{a}, s') = Pr(s'|s, \vec{a})$.

- $O : \vec{A} \times S \times \vec{\Omega} \to [0,1]$ is an observation function mapping joint action $\vec{a}$ and successor state s' to a joint observation $\vec{\omega}$ such that $O(\vec{a}, s', \vec{\omega}) = Pr(\vec{\omega}|\vec{a}, s')$.

- $\vec{R} = [R_0, \ldots, R_n]^T$ is a vector of rewards; for each state s and action a, $R_0(s, \vec{a})$ is a *group reward* and $R_i(s, \vec{a})$ is an *individual reward* for each agent i.

A CCP operates at discrete time steps up to some *horizon* h, either *finite* ($h \in \mathbb{N}$) or *infinite* ($h = \infty$). Each agent i has only access to its private observation Ω_i after each joint action. Rewards are discounted over time by *discount factor* $\gamma \in (0,1]$. Unfortunately, infinite-horizon Dec-POMDPs, POSGs, and thus CCPs are undecidable like POMDPs, though they can be approximated. In POMDPs, however, a *belief* over the true state is a sufficient statistic for the entire history of actions and observations (Kaelbling, Littman, and Cassandra 1998). In Dec-POMDPs, POSGs, and CCPs, the full history of joint actions and joint observations is required (in the worst case) to determine optimal actions. This stems from the often insufficient local observations of each agent. Thus, an agent i has *local policy* $\pi_i : \mathcal{P}(\Omega_i^h) \to A_i$ that maps any history of observations to an action, with power set $\mathcal{P}(\cdot)$, and Ω_i to the h-th Cartesian power Ω_i^h. These are stored as *policy trees*: actions at each node with branching observations. A *joint policy* $\vec{\pi} = \langle \pi_1, \ldots, \pi_n \rangle$ is a tuple of the agents' local policies. We denote the set of all possible joint policies as $\vec{\Pi}$, and Π_i denoting the set of all possible local policies for agent i.

We characterize policies in terms of a *value function*, representing the expected reward following a policy. Formally, given policy $\vec{\pi}$ and initial belief $b^0 \in \triangle^{|S|}$, for each reward function R_k, $k \in \{0, \ldots, n\}$, the value is defined as:

$$V_k^{\vec{\pi}}(b^0) = \mathbb{E}\Big[\sum_{t=0}^{h-1} \gamma^t R_k^t | b^0, \vec{\pi} \Big] \tag{1}$$

with random variable R_k^t denoting the reward at t. We also denote the values of the group objective as $V_0(b^0)$, individual objectives as $V_i(b^0)$ for $i \in I$, and the value of the optimal policy as $V_k^*(b^0) = \max_{\vec{\pi} \in \vec{\Pi}} V_k^{\vec{\pi}}(b^0)$.

The Objective Functions There are two complimentary objective functions we consider for CCPs. They differ in their lexicographic preference for optimizing the group versus the individual performance. A group dominant objective function (Definition 2) for a CCP primarily maximizes the group reward, then secondarily maximizes the individual rewards by reducing the group's expected reward up to some allotted slack. Conversely, an individual dominant objective function (Definition 3) for a CCP primarily maximizes the individual rewards, then secondarily maximizes the group rewards, allowing for each agent to reduce their expected reward by some individual allotted slack.

These definitions borrow notation from game theory. For an agent i, let $-i = \langle 1, \ldots, i-1, i+1, \ldots, n \rangle$. This is used to refer to all other elements in a tuple or set except for agent i; for example, we have $\vec{\pi}_{-i} = \langle \pi_1, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_n \rangle$. Additionally, we consider *best responses*, denoted as BR, which refers to the best policy (or *strategy*) given that the policies of all other agents (or *players*) are fixed. A situation in which all players are playing a best response to one another is called a *Nash equilibrium* (Fudenberg and Tirole 1991). This best response may be in the form of either a deterministic selection of a policy ($\vec{\Pi} = \langle \mathcal{P}(\Omega_1^h), \ldots, \mathcal{P}(\Omega_n^h) \rangle$ as before) or a stochastic selection from a set of policies ($\vec{\Pi} = \triangle^{|\langle \mathcal{P}(\Omega_1^h), \ldots, \mathcal{P}(\Omega_n^h) \rangle|}$), also called *pure* or *mixed* strategies, respectively.

Definition 2. Given a CCP with initial belief $b^0 \in \triangle^{|S|}$, a **group-dominant CCP (GD-CCP)** with group slack $\delta \geq 0$ has an optimal policy $\vec{\pi}^*$ that satisfies:

$$\vec{\Pi}^G = \{\vec{\pi} \in \vec{\Pi} | V_0^*(b^0) - V_0^{\vec{\pi}}(b^0) \leq \delta\}$$

$$BR^G(\vec{\pi}_{-i}) = \{\pi_i \in \Pi_i^G | \forall \pi_i' \in \Pi_i^G, V_i^{\vec{\pi}}(b^0) \geq V_i^{\vec{\pi}'}(b^0)\}$$

$$\vec{\pi}^* \in \vec{\Pi}^* = \{\vec{\pi} \in \vec{\Pi}^G | \forall i, \pi_i \in BR^G(\vec{\pi}_{-i})\}. \tag{2}$$

Definition 3. Given a CCP with initial belief $b^0 \in \triangle^{|S|}$, an **individual-dominant CCP (ID-CCP)** with individual slacks $\delta_i \geq 0$ has an optimal policy $\vec{\pi}^*$ that satisfies:

$$BR^I(\vec{\pi}_{-i}) = \{\pi_i \in \Pi_i | \forall \pi_i' \in \Pi_i, V_i^{\vec{\pi}}(b^0) \geq V_i^{\vec{\pi}'}(b^0)\}$$

$$\vec{\Pi}^I = \{\vec{\pi} \in \vec{\Pi} | \forall i, \pi_i \in BR^I(\vec{\pi}_{-i}) \wedge V_i^*(b^0) - V_i^{\vec{\pi}}(b^0) \leq \delta_i\}$$

$$\vec{\pi}^* \in \vec{\Pi}^* = \{\vec{\pi} \in \vec{\Pi}^I | \max_{\vec{\pi}' \in \vec{\Pi}^I} V_0^{\vec{\pi}'}(b^0) = V_0^{\vec{\pi}}(b^0)\}. \tag{3}$$

Informally, a GD-CCP first identifies policies $\vec{\Pi}^G$ that maximize the expected group reward, then considers policies whose group value deviates up to the slack from the optimal value. Among these policies, the set of optimal policies $\vec{\Pi}^* \subseteq \vec{\Pi}^G$ are those that ensure that all agents are performing a best response to one another with respect to their individual objectives (i.e., a Nash equilibrium). ID-CCPs follow the same pattern but with a reversed preference, prioritizing individual objectives.

Scalable Solution for CCPs

For finite horizon CCPs, Definitions 2 and 3 can be computed by exhaustively enumerating all joint policies and

restricting the set of policies as shown. Hence, the equations represent a naive optimal algorithm for solving GD-CCPs and ID-CCPs. Additionally, clever policy tree pruning methods used for Dec-POMDPs (Bernstein et al. 2009) and POSGs (Hansen, Bernstein, and Zilberstein 2004) may be applied here as well. These algorithms exploit the fact that some policy trees or subtrees may be eliminated because their values are strictly dominated by another tree. As with Dec-POMDPs and POSGs, however, these algorithms do not scale well due to the sheer complexity of the problem.

Instead, other approximate solutions to Dec-POMDPs include bounded policy iteration (Dec-BPI) (Bernstein, Hansen, and Zilberstein 2005), joint-equilibrium-based search for policies (JESP) (Nair et al. 2003), and nonlinear programming (NLP) solutions using finite state controllers (FSCs) (Amato, Bernstein, and Zilberstein 2010). POSGs also enjoy approximation methods which use best-response dynamics (Oliehoek, Spaan, and Vlassis 2005), possibly in conjunction with model and policy approximations (e.g., Bayesian games) (Emery-Montemerlo et al. 2004).

This body of successful research forms the foundation of our approach. Our approximation algorithm for GD-CCPs uses FSC mathematical programming techniques to approximate the policy in synergy with best response dynamics as a basis to realistically solve large GD-CCPs. The solution also enjoys a very natural characterization of slack and results in a tractable solution with crisp theoretical results.

Hence, we use a *stochastic finite state controller (FSC)* for each agent i, represented by the tuple $\langle Q_i, \psi_i, \eta_i \rangle$ (Bernstein, Hansen, and Zilberstein 2005). Q_i denotes the finite set of controller nodes for agent i. The stochastic action selection function $\psi_i : Q_i \times A_i \to [0,1]$ is defined such that $\psi_i(q_i, a_i) = Pr(a_i|q_i)$ per controller node q_i and action a_i. The controller transition function $\eta_i : Q_i \times A_i \times \Omega_i \times Q_i \to [0,1]$ is defined such that $\eta_i(q_i, a_i, \omega_i, q_i') = Pr(q_i'|q_i, a_i, \omega_i)$ given that in controller node q_i, taking action a_i yielded observation ω_i, resulting in successor q_i'.

We denote $\vec{\pi} = \langle \vec{\psi}, \vec{\eta} \rangle$ as *policy*, with $\vec{\psi} = \langle \psi_1, \dots, \psi_n \rangle$ and $\vec{\eta} = \langle \eta_1, \dots, \eta_n \rangle$, given controller nodes $\vec{Q} = Q_1 \times \dots \times Q_n$.

Given the local FSCs for all agents, the *value* of objective $k \in \{0, \dots, n\}$ at nodes $\vec{q}$ and state s is (Amato, Bernstein, and Zilberstein 2010):

$$V_k(\vec{q}, s) = \sum_{\vec{a} \in \vec{A}} \prod_{i=1}^{n} \psi_i(q_i, a_i) \Big(R_k(s, \vec{a}) + \gamma \sum_{s' \in S} T(s, \vec{a}, s')$$

$$\sum_{\vec{\omega} \in \vec{\Omega}} O(\vec{a}, s', \vec{\omega}) \sum_{\vec{q}' \in \vec{Q}} \prod_{j=1}^{n} \eta_j(q_j, a_j, \omega_j, q_j') V_k(\vec{q}', s') \Big)$$

Algorithm 1 presents a scalable FSC solution to CCPs that assumes a given tuple of fixed size FSC nodes $\vec{Q}$. First, in Table 1 we compute the best approximate value, denoted $\hat{V}_0^*$, for the group objective (reward R_0) following the *nonlinear programming (NLP)* techniques developed by Amato, Bernstein, and Zilberstein (2010). Second, we select an agent i arbitrarily, defined by some function $select(I)$ that returns an agent $i \in I$ (e.g., random) as long as it does not starve the selection of any agents. Following this agent selection,

---

Algorithm 1 Approximate FSC Solution to GD-CCP

Require: $\langle S, \vec{A}, \vec{\Omega}, T, O, \vec{R} \rangle$: The GD-CCP.
Require: $\vec{Q} = \langle Q_1, \dots, Q_n \rangle$: The fixed controller nodes.
Require: $q^0 \in \vec{Q}$: The initial controller nodes.
Require: $b^0 \in \triangle^{|S|}$: The initial belief state.
Require: ϵ: The convergence criterion.
1: $\vec{\psi}^*, \vec{\eta}^*, \hat{V}_0 \leftarrow solve\_\pi_0(\vec{q}^0, b^0)$
2: $\hat{V}_0^*(b^0) \leftarrow \sum_{s \in S} b^0(s) \hat{V}_0(\vec{q}^0, s)$
3: $\hat{V}_i^*(b^0) \leftarrow 0, \quad \forall i \in \{1, \dots n\}$
4: $\vec{d} = \langle d_1, \dots, d_n \rangle \leftarrow \langle \epsilon, \dots, \epsilon \rangle$
5: **while** $\max_j d_j \geq \epsilon$ **do**
6: $\quad i \leftarrow select(I)$
7: $\quad \psi_i^*, \eta_i^*, \hat{V}_i, \hat{V}_0 \leftarrow solve\_\pi_i(i, \vec{q}^0, b^0, \vec{\psi}_{-i}^*, \vec{\eta}_{-i}^*, \hat{V}_0^*(b^0))$
8: $\quad \hat{V}_i^*(b^0)' \leftarrow \sum_{s \in S} b^0(s) \hat{V}_i(\vec{q}^0, s)$
9: $\quad d_i \leftarrow \max\{d_i, |\hat{V}_i^*(b^0)' - \hat{V}_i^*(b^0)|\}$
10: $\quad \hat{V}_i^*(b^0) \leftarrow \hat{V}_i^*(b^0)'$
11: $\quad \hat{V}_0^*(b^0) \leftarrow \sum_{s \in S} b^0(s) \hat{V}_0(\vec{q}^0, s)$
12: **end while**
13: **return** $\vec{\psi}^*, \vec{\eta}^*, \hat{V}_0^*(b^0), \dots, \hat{V}_n^*(b^0)$

---

we fix all other agents' FSCs, and compute a new local FSC for i that maximizes the individual expected reward R_i over time, denoted $\hat{V}_i^*$, as shown in Table 2. This is an iterative *best response dynamics* process. The FSC for each agent i is solved using a *cubically constrained linear program (CCLP)* (as written in Table 2) similar to the NLP. Two additional constraints are included. We must compute the effective $\hat{V}_0^\pi$ following the local deviations made by agent i. Using this, we constrain the deviation of this value up to the slack δ.

While written in its most general form for clarity, the CCLP follows the same logic as Amato, Bernstein, and Zilberstein (2010) use for their quadratically constrained linear program (QLCP) POMDP solution. Specifically, we distribute ψ_i to R_i and directly to $\eta_i \hat{V}_i$ inside the summation. We may then combine η_i and ψ_i as a single variable $Pr(q_i', a_i|q_i, \omega_i)$ via the transformations:

$$Pr(q_i', a_i|q_i, \omega_i) = \psi_i(q_i, a_i) \eta_i(q_i, a_i, \omega_i, q_i')$$

$$\sum_{q_i' \in Q_i} Pr(q_i', a_i|q_i, \omega_i) = \psi_i(q_i, a_i)$$

As with other forms of slack (Wray and Zilberstein 2015a), this algorithm only guarantees that the final joint policy $\hat{\pi}^*$ has a value $\hat{V}_0^{\hat{\pi}^*}$ that is within δ of the more preferred objective's value $\hat{V}_0^*$, which is approximate in this case with a fixed set of controller nodes. It is *not* within slack of the *true* optimal value V_0^*, since we obviously did not compute that; $\hat{\pi}^*$ is an approximate solution after all.

Theoretical Analysis

To begin, we prove a straight-forward but important property of an CCP in Definition 1, namely that they strictly generalize both Dec-POMDP and a POSG.

Proposition 1. For all Dec-POMDPs and POSGs, there exists an equivalent CCP.

```
Function:   solve_π₀
```

$$\text{Function:} \quad solve\_\pi_0$$
$$\text{Given:} \quad \vec{q}^0 \in \vec{Q} \quad b^0 \in \triangle^{|S|}$$
$$\text{Variables:} \quad \psi_i(q_i, a_i), \forall i, q_i, a_i \quad \eta_i(q_i, a_i, \omega_i, q_i'), \forall i, q_i, a_i, \omega_i, q_i' \quad \hat{V}_0(\vec{q}, s), \forall \vec{q}, s$$
$$\text{Maximize:} \quad \hat{V}_0^*(b^0) = \sum_{s \in S} b^0(s) \hat{V}_0(\vec{q}^0, s)$$

Bellman Constraint:
$$\hat{V}_0(\vec{q}, s) = \sum_{\vec{a} \in \vec{A}} \prod_{i=1}^n \psi_i(q_i, a_i) \Big[R_0(s, \vec{a}) + \gamma \sum_{s' \in S} T(s, \vec{a}, s') \sum_{\vec{\omega} \in \vec{\Omega}} O(\vec{a}, s', \vec{\omega}) \sum_{\vec{q}' \in Q} \prod_{i=1}^n \eta_i(q_i, a_i, \omega_i, q_i') \hat{V}_0(\vec{q}', s') \Big]$$

Probability Constraints:
$$\psi_i(q_i, a_i) \geq 0, \forall i, q_i, a_i \qquad \sum_{a_i \in A_i} \psi_i(q_i, a_i) = 1, \forall i, q_i$$
$$\eta_i(q_i, a_i, \omega_i, q_i') \geq 0, \forall i, q_i, a_i, \omega_i, q_i' \qquad \sum_{q_i' \in Q_i} \eta_i(q_i, a_i, \omega_i, q_i') = 1, \forall i, q_i, a_i, \omega_i$$

Table 1: The NLP for computing the solution to the group reward $\hat{V}_0^*$.

$$\text{Function:} \quad solve\_\pi_i$$
$$\text{Given:} \quad \text{Agent } i \in I \quad \vec{q}^0 \in \vec{Q} \quad b^0 \in \triangle^{|S|} \quad \psi_j(q_j, a_j), \forall j \neq i, q_j, a_j \quad \eta_j(q_j, a_j, \omega_j, q_j'), \forall j \neq i, q_j, a_j, \omega_j, q_j' \quad \hat{V}_0^*(b^0)$$
$$\text{Variables:} \quad \psi_i(q_i, a_i), \forall q_i, a_i \quad \eta_i(q_i, a_i, \omega_i, q_i'), \forall q_i, a_i, \omega_i, q_i' \quad \hat{V}_i(\vec{q}, s), \forall \vec{q}, s \quad \hat{V}_0(\vec{q}, s), \forall \vec{q}, s$$
$$\text{Maximize:} \quad \hat{V}_i^*(b^0) = \sum_{s \in S} b^0(s) \hat{V}_i(\vec{q}^0, s)$$

Bellman Constraint for Individual Objective:
$$\hat{V}_i(\vec{q}, s) = \sum_{\vec{a} \in \vec{A}} \prod_{j=1}^n \psi_j(q_j, a_j) \Big[R_i(s, \vec{a}) + \gamma \sum_{s' \in S} T(s, \vec{a}, s') \sum_{\vec{\omega} \in \vec{\Omega}} O(\vec{a}, s', \vec{\omega}) \sum_{\vec{q}' \in Q} \prod_{j=1}^n \eta_j(q_j, a_j, \omega_j, q_j') \hat{V}_i(\vec{q}', s') \Big]$$

Bellman Constraint for Group Objective:
$$\hat{V}_0(\vec{q}, s) = \sum_{\vec{a} \in \vec{A}} \prod_{j=1}^n \psi_j(q_j, a_j) \Big[R_0(s, \vec{a}) + \gamma \sum_{s' \in S} T(s, \vec{a}, s') \sum_{\vec{\omega} \in \vec{\Omega}} O(\vec{a}, s', \vec{\omega}) \sum_{\vec{q}' \in Q} \prod_{j=1}^n \eta_j(q_j, a_j, \omega_j, q_j') \hat{V}_0(\vec{q}', s') \Big]$$

Slack Constraint:
$$\hat{V}_0^*(b^0) - \sum_{s \in S} b^0(s) \hat{V}_0(\vec{q}^0, s) \leq \delta$$

Probability Constraints:
$$\psi_i(q_i, a_i) \geq 0, \forall q_i, a_i \qquad \sum_{a_i \in A_i} \psi_i(q_i, a_i) = 1, \forall q_i$$
$$\eta_i(q_i, a_i, \omega_i, q_i') \geq 0, \forall q_i, a_i, \omega_i, q_i' \qquad \sum_{q_i' \in Q_i} \eta_i(q_i, a_i, \omega_i, q_i') = 1, \forall q_i, a_i, \omega_i$$

Table 2: The NLP (written as CCLP for clarity) for computing the best response for agent i's reward, $\hat{V}_i^*$, given all others $-i$ are fixed.

Proof. For any Dec-POMDP $\mathcal{X}$ with rewards R, let a GD-CCP $\mathcal{Z}$ have identical S, $\vec{A}$, $\vec{O}$, T, and O with $\vec{R} = [R, R, \ldots, R]^T$ and slack $\delta = 0$. By Definition 2, $\vec{\Pi}^G = \{\vec{\pi} \in \vec{\Pi} | V_0^*(b^0) = V_0^{\vec{\pi}}(b^0)\}$ which are equivalent optimal policies for $\mathcal{X}$. Similarly, for a POSG $\mathcal{Y}$ with payoffs $\langle R_1, \ldots, R_n \rangle$, let a GD-CCP have $\vec{R} = [R_0, R_1, \ldots, R_n]^T$ with any R_0 with slack $\delta \to \infty$; e.g., infinite horizon has $\delta = \max_{s, \vec{a}} R_0(s, \vec{a})/(1 - \gamma)$ and finite horizon has $\delta = \max_{s, \vec{a}} R_0(s, \vec{a})h$. Thus, by Definition 2, $\vec{\Pi}^G = \vec{\Pi}$, resulting in the exclusively Nash equilibrium solution concept POSG $\mathcal{Y}$. Similar logic applies for ID-CCPs in the natural way. $\square$

Definition 5 presents a type of GD-CCP for which a *potential function* (Definition 4) exists. This class of problems extends the powerful and well-known *potential games* (Monderer and Shapley 1996). Intuitively, this class of GD-CCPs means that while individual rewards might differ and be competing, there is still a sense of a collective objective (likely distinct from R_0).

Definition 4. An (ordinal) **potential** for a GD-CCP is a function $\Phi: \vec{\Pi} \to \mathbb{R}$ such that for all agents i:

$$V_i^{\vec{\pi}}(b^0) - V_i^{\vec{\pi}'}(b^0) > 0 \quad \text{iff} \quad \Phi(\vec{\pi}) - \Phi(\vec{\pi}') > 0 \quad (4)$$

for all $\vec{\pi}, \vec{\pi}' \in \vec{\Pi}$ such that $\vec{\pi}_{-i} = \vec{\pi}'_{-i}$, with $b^0 \in \triangle^{|S|}$.

Definition 5. A **potential GD-CCP** is a GD-CCP for which there exists a potential function Φ.

Within this class of GD-CCPs, Proposition 2 proves that if a potential function exists then it induces a potential game for the individual rewards for *all* assignments of slack.

Proposition 2. For a potential GD-CCP and slack $\delta \geq 0$, the policy constraints induced by Equation 2 preserve potential Φ and induce an equivalent potential game denoted $\mathcal{N}$.

Proof. By Equation 2, $\vec{\Pi}^G \subseteq \vec{\Pi}$ which restricts according to slack δ. This induces an equivalent normal form game $\mathcal{N}$ with strategy profiles $\vec{\Pi}^G$ and utilities $\langle V_1^{\vec{\pi}}(b^0), \ldots, V_n^{\vec{\pi}}(b^0) \rangle$ for $\vec{\pi} \in \vec{\Pi}^G$ and initial belief b^0. By Definition 4, we are given potential Φ, thus $\mathcal{N}$ is also a potential game. $\square$

Next, we consider *finite potential* GD-CCPs (Definition 6) to facilitate convergence in potential games. Deterministic FSCs, commonly used in other work (Kumar, Mostafa, and Zilberstein 2016), produce finite potential GD-CCPs and can be solved as a mixed integer linear program (MILP). Another natural case is stochastic FSCs defined as bounded integer variables in Tables 1 and 2 with a normalization term.

Definition 6. A **finite potential GD-CCP** is a potential GD-CCP with finite sets $\Pi_i \in \vec{\Pi}$ for all $i \in I$.
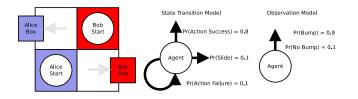
Figure 1: Two abstract meeting box-pushing domains: Battle Meeting and Prisoner Meeting. States: 2x2 Grid. Actions: None, N, S, E, W. Observations: No Bump and Bump. Stochastic movement and observations. Group Objective: Proportional to distance apart. Individual Objectives: Battle of the Sexes or Prisoner's Dilemma.

Now we present two propositions characterizing Algorithm 1. First, we show that the algorithm converges to a joint policy (i.e., pure strategy profile) given that it is a finite potential GD-CCP. Next, we show that this joint policy satisfies the slack constraint imposed by the model—a simple but crucial fact. Note that in both cases, we assign $\epsilon = 0$ because we consider finite potential GD-CCPs that guarantee convergence to equilibria in a finite number of iterations.

Proposition 3. Algorithm 1 with $\epsilon = 0$ converges to a joint policy $\vec{\pi}^*$ for finite potential GD-CCP.

Proof. Solving the NLP (Table 1) on Line 1, we obtain our slack constraint for the CCLPs. The algorithm then iterates from Line 5 to Line 12, until the policy does not change ($\vec{\pi} = \vec{\pi}'$) due to $\epsilon = 0$. For the selected agent each time, Line 7 computes a best response following Table 2's objective function. As described in Proposition 2, there exists an equivalent normal form game $\mathcal{N}$. Given (ordinal) potential Φ, this is a potential game for any slack δ. All finite ordinal potential games converge in a finite number of steps to a pure strategy Nash equilibrium following best response dynamics as proven by Monderer and Shapley (1996). $\qquad\square$

Proposition 4. Algorithm 1 with $\epsilon = 0$ produces joint policy $\vec{\pi}^*$ which satisfies the slack δ for finite potential GD-CCP.

Proof. By Line 7 and the Slack Constraint in Table 2, we have $\hat{V}_0^*(b^0) - \sum_{s \in S} b^0(s)\hat{V}_0(\vec{q}^0, s) \leq \delta$. By Line 6, this is true for all agents $i \in I$. $\qquad\square$

Experiments

Our experiments consider two GD-CCPs which both uniquely combine the spirit of the standard cooperative domain *Meeting in a Grid* (Amato, Bernstein, and Zilberstein 2010) with the spirit of two competitive domains: *Battle of the Sexes* and *Prisoner's Dilemma* (Fudenberg and Tirole 1991). In both cases, the agents are primarily rewarded proportional to their distance from one another. They allow for a relaxation of this objective up to some slack in order to push an agent-specific box for an individual reward. The Battle of the Sexes variant assigns a greater reward for "push" only if the agents are together. The Prisoner's Dilemma variant assigns "cooperate" as "do not push" and "defect" as "push". Figure 1 shows a visual example of the domains' models including the starting locations and individual box locations.

CCP Domains We consider two novel domains called **Battle Meeting** and **Prisoner Meeting**. In both, there are two agents $I = \{1, 2\}$ and the state space is $S = S_1 \times S_2$ with $S_i = \{top\_left, top\_right, bottom\_left, bottom\_right\}$. It has action space $A = A_1 \times A_2$ with $A_i = \{none, north, south, east, west\}$ and observation space $\Omega = \{no\_bump, bump\}$. The state transitions T are defined as found in Figure 1. For example, successful movement is 0.8, sliding to a neighboring cell is 0.1, and failure to move is 0.1; the agents always succeeds at not moving. Also, they have transition independence for movement, and self-loop with 0.9 probability when moving into a box or wall with a 0.1 probability of sliding to the neighbor cell. Observations are perfect for detecting a "bump" at a wall or box, but there is a 0.1 probability of a false positive detected while moving normally.

The group objective rewards R_0 are defined by $2 - d(s_1, d_2)$ where $d(s_1, s_2)$ is the Manhattan distance between agents' locations $s_1 \in S_1$ and $s_2 \in S_2$. Battle Meeting's individual objectives assign reward R_i to $2 - d(s_1, s_2)$ if agent i pushes its box, and half that if $-i$ pushes its box. Prisoner Meeting's individual objectives assign reward R_i to: 1 if both push their respective boxes, 3 if agent i pushes but $-i$ does not, 0 if i does not push but $-i$ does, and 2 if both do not push. In all cases, all three objectives have an additional small -0.1 penalty if any movement action is taken. Lastly, we have a discount factor of $\gamma = 0.95$.

Evaluation in Simulation We evaluate our approximate CCP algorithm in simulation with three different amounts of controller nodes with $|Q_i| \in \{2, 4, 6\}$ for each agent i in Figure 2. In each scenario, we evaluate the average discounted reward (ADR) vs. the allotted slack (δ). The ADR averages over 1000 trials for each scenario (i.e., each point). The standard error is provided as error bars. We omit the directly computed $\hat{V}_i^*(b^0)$ for π^* as they match the ADR values.

Figure 2 shows the results from the experiments. We solve the NLPs and CCLPs in Tables 1 and 2 using the NEOS Server (Czyzyk, Mesnier, and Moré 1998) running SNOPT (Gill, Murray, and Saunders 2005). Table 3 summarizes the timings (in seconds) for the group and individual objectives, as well as the entire solution time (in seconds), for all scenarios. We also include the number of iterations for the best response dynamics. We allowed for a maximum of 50 iterations, occasionally causing early termination of the best response dynamics. In a few cases, SNOPT can also fail to optimally solve the NLP, as it is an approximation, resulting in occasional sub-optimal policies. All other normal cases terminate when $\max_j d_j < \epsilon = 0.01$ as in Algorithm 1.

Evaluation on Two Robots We implemented the Prisoner Meeting and Battle Meeting domains on two real robot platforms (Figure 3). The objective is to evaluate the stochastic FSC performance on an actual system, as opposed to only an abstract notion of value. In prior work, this challenging but *crucial* step is unfortunately skipped when using FSCs or other approximation techniques. Conducting actual experiments ensures that the policies we compute are suitable for use in real systems and produce meaningful agent behaviors in practice.
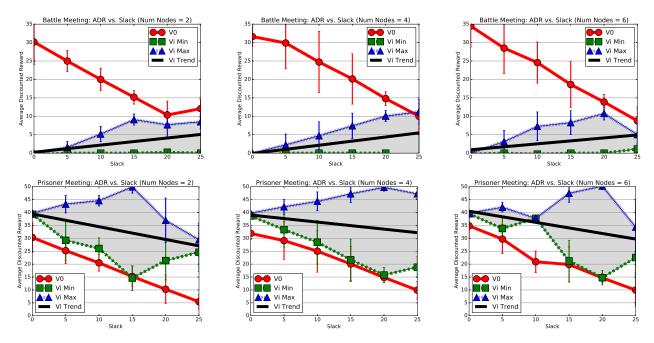
Figure 2: Results for the Battle Meeting (top) and Prisoner Meeting (bottom) domains. The average discounted reward (ADR) for the group objective (V0) and the min/max values for two agents' individual objectives (V1 and V2) are shown for different amounts of slack $\delta \in [0, 25]$.

Meeting	$|Q_i|$	Group	Individual	Iterations	Total
Battle	2	17.6	21.1	10.5	447.2
	4	322.4	49.3	24.3	2827.3
	6	181.5	196.9	16.5	10775.3
Prisoner	2	22.1	20.7	13.3	574.1
	4	256.8	39.1	26.2	2577.0
	6	177.3	182.2	38.3	11702.8

Table 3: Battle Meeting and Prisoner Meeting average computation time (seconds) for group and individual objectives, the average number of best response iterations, and the average total computation time (seconds).

Discussion

CCPs are designed to naturally blend between cooperation and competition with an intuitive meaning for how this structure operates in a given domain. The two example domains illustrate this intent by means of rewarding both agents meeting for the group while rewarding individual goals either in the form of Prisoner's Dilemma or Battle of the Sexes. The reward structure in each case directly matches the classic games from game theory, expanded into a richer state and action space. Interestingly, Prisoner's Dilemma and Battle of the Sexes are both classical potential games as well.

In both Prisoner and Battle Meeting, one agent is always able to exploit the other. The higher performing agent is determined primarily by their best response ordering and the approximations of SNOPT when solving the NLP (e.g., relaxed nonlinear infeasibilities), even though their initialization is entirely symmetric. The regions found in Figure 2 show the range of the agents' individual objectives' expected values. The trend lines are shown to reinforce the observation that it indeed produces a blended cooperation from meeting to either Battle of the Sexes (trend increases) or Prisoner's Dilemma (trend decreases). Additionally, the timings in Table 3 clearly show that the number of iterations before convergence is typically a reasonable number between 10 and 25 for most of our experiments. The time to compute a best response, however, scales rapidly depending on the number of controller nodes, as expected. Overall our results show that with zero slack the agents focus almost exclusively on meeting. As we increase slack, however, their preference shifts to first move their individual boxes then meet in the grid. Finally, as the slack increases the agents favor only pushing individual boxes, based on the objectives, and do not focus on the meeting objective.

Simulation and Robotic Experiments Battle Meeting is designed to demonstrate that as slack increases, the group objective *decreases*, and the individual objectives are able to *increase*. In Figure 2, agents cannot satisfy their individual objectives with $\delta = 0$. As $\delta \to 25$, the agents are able to focus more on box pushing and less on meeting in the grid as evident by the trend line and individual expected values. The robotic experiments in Figure 3 clearly show this behavior as well. With $\delta = 0$, the best decision is for one agent (Alice, arbitrarily chosen due to symmetric initial state) to risk the stochastic motion in an attempt to meet the other (Bob) in the grid. Conversely at $\delta = 25$, they *exactly* incarnate Battle of the Sexes wherein one agent pushes its box for a large reward (Bob, again essentially arbitrary) but the other agent merely joins (Alice) for a small reward.

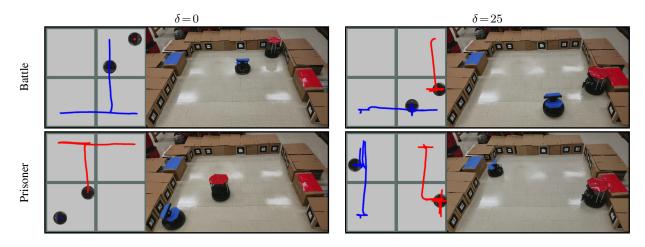Prisoner Meeting is designed instead to demonstrate a

Figure 3: Experiments with CCP on two real robot platforms. Battle Meeting (top row) and Prisoner Meeting (bottom row) are shown with slack $\delta = 0$ (left column) and $\delta = 25$ (right column) for both. The paths traversed by robots "Alice" (blue) and "Bob" (red) over the entire execution are shown as well. For each slack allotment, the robots adapt their policy to work together to satisfy the group task and/or complete their individual box-pushing tasks.

different behavior to clearly illustrate the classic Prisoner's Dilemma. As slack increases, the intent is to elicit the group objective to *decrease* but, very importantly, the individual objectives are *also expected to decrease*. We observe this desired behavior in Figure 2. With $\delta = 0$, the agents are *forced* into the classical Prisoner's Dilemma's "cooperate" actions—not pushing the boxes—without the option to "defect"—to push their boxes. As slack increases, they converge to the proper Nash equilibrium of "defect" for both players which results in lower values for both agents (Fudenberg and Tirole 1991). Again, the robotic experiments perfectly illustrate the expected behavior. With $\delta = 0$, a robot (Bob, again, essentially arbitrarily chosen) risks stochastic movement to meet the other (Alice). Interestingly, a slack of $\delta = 0$ forces both players in Prisoner's Dilemma to "cooperate" by not pushing their boxes. Once $\delta = 25$, however, both robots have enough slack to push their individual boxes, probabilistically ignoring meeting each other altogether.

In summary, the simulation results in Figure 2 and robotic experiments in Figure 3 demonstrate the intent of the CCP model: In all 6 cases shown, as slack increases, the group objective decreases up to the allotted slack, allowing the best response dynamics to optimize individual objectives.

Fair Distribution The CCP model's slack raises interesting questions of *fairness* (Rabin 1993). For example, is one agent in a GD-CCP allowed to consume all of the slack in order to maximize its own rewards at the expense of others? Or perhaps do we enforce some measure of fairness in the allocation of slack so that all agents have a chance to fairly maximize their rewards? As defined, we consider a shared slack to be divided amongst the group based on best response converging to a Nash equilibria. More fair methods may be employed that ensure each agent receives *some* share of the slack. For example, a formal definition of local and global fairness properties can ensure an even distribution (Burkhard 1993). Conversely, a form of reci-

procity solution concept may be used instead, possibly including corresponding reciprocity payoffs (Dufwenberg and Kirchsteiger 2004). These approaches, however, likely require many more best response iterations. Addressing this increased complexity while ensuring equal slack distribution is a topic of interest for future work.

Conclusion

It has long been a challenge to build a formal multi-agent model that seamlessly incorporates both cooperation and competition. The CCP model is shown to be an effective tool for capturing this intriguing marriage of the cooperative Dec-POMDP and the competitive POSG models. It enables an intuitive slack parameter to control the degree of cooperation/competition. A theoretically-grounded approximate algorithm uses FSCs to produce tractable results that are demonstrated to work on actual multi-robot systems.

Future work will focus on building more scalable algorithms (Wray and Zilberstein 2015b), potentially using other solution concepts and incorporating measures to ensure the fair distribution of slack among the agents. We are also interested in applying this framework to more expansive robotic domains that require modeling both cooperation and competition, such as autonomous vehicles. Towards this goal, we will provide our source code to support further development of models that generalize cooperation and competition under a unified approach.

Acknowledgments

References

Amato, C.; Bernstein, D. S.; and Zilberstein, S. 2010. Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs. *Autonomous Agents and Multi-Agent Systems* 21(3):293–320.

Becker, R.; Zilberstein, S.; Lesser, V.; and Goldman, C. V. 2004. Solving transition independent decentralized Markov decision processes. *Journal of Artificial Intelligence Research (JAIR)* 22:423–455.

Bernstein, D. S.; Givan, R.; Immerman, N.; and Zilberstein, S. 2002. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research* 27(4):819–840.

Bernstein, D. S.; Amato, C.; Hansen, E. A.; and Zilberstein, S. 2009. Policy iteration for decentralized control of Markov decision processes. *Journal of Artificial Intelligence Research (JAIR)* 34(1):89–132.

Bernstein, D. S.; Hansen, E. A.; and Zilberstein, S. 2005. Bounded policy iteration for decentralized POMDPs. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, 1287–1292.

Burkhard, H.-D. 1993. Liveness and fairness properties in multi-agent systems. In *Proceedings of the 13th International Joint Conference of Artificial Intelligence (IJCAI)*, 325–331.

Cardon, A.; Galinho, T.; and Vacher, J.-P. 2000. Genetic algorithms using multi-objectives in a multi-agent system. *Robotics and Autonomous Systems (RAS)* 33(2):179–190.

Coello, C. A.; Lamont, G. B.; and Van Veldhuizen, D. A. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer.

Czyzyk, J.; Mesnier, M. P.; and Moré, J. J. 1998. The NEOS Server. *IEEE Computational Science and Engineering* 5(3):68–75.

Decker, K. 1996. TAEMS: A framework for environment centered analysis & design of coordination mechanisms. In Hare, G. O., and Jennings, N., eds., *Foundations of Distributed Artificial Intelligence*. Wiley Inter-Science. 429–448.

Delle Fave, F. M.; Stranders, R.; Rogers, A.; and Jennings, N. R. 2011. Bounded decentralised coordination over multiple objectives. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 371–378.

Dufwenberg, M., and Kirchsteiger, G. 2004. A theory of sequential reciprocity. *Games and Economic Behavior* 47(2):268–298.

Emery-Montemerlo, R.; Gordon, G.; Schneider, J.; and Thrun, S. 2004. Approximate solutions for partially observable stochastic games with common payoffs. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 136–143.

Fudenberg, D., and Tirole, J. 1991. *Game Theory*. The MIT Press.

Gill, P. E.; Murray, W.; and Saunders, M. A. 2005. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Review* 47(1):99–131.

Gmytrasiewicz, P. J., and Doshi, P. 2005. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research (JAIR)* 24:49–79.

Hansen, E. A.; Bernstein, D. S.; and Zilberstein, S. 2004. Dynamic programming for partially observable stochastic games. In *Proceedings of the 19th AAAI Conference on Artificial Intelligence (AAAI)*, 709–715.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1):99–134.

Kumar, A.; Mostafa, H.; and Zilberstein, S. 2016. Dual formulations for optimizing Dec-POMDP controllers. In *Proceedings of the 26th International Conference on Automated Planning and Scheduling (ICAPS)*, 202–210.

Kumar, A.; Zilberstein, S.; and Toussaint, M. 2011. Scalable multiagent planning using probabilistic inference. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2140–2146.

Matignon, L.; Jeanpierre, L.; and Mouaddib, A.-I. 2012. Distributed value functions for multi-robot exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1544–1550.

Monderer, D., and Shapley, L. S. 1996. Potential games. *Games and Economic Behavior* 14(1):124–143.

Mouaddib, A.-I.; Boussard, M.; and Bouzid, M. 2007. Towards a formal framework for multi-objective multiagent planning. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 801–808.

Nair, R.; Tambe, M.; Yokoo, M.; Pynadath, D.; and Marsella, S. 2003. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, 705–711.

Oliehoek, F.; Spaan, M. T.; and Vlassis, N. 2005. Best-response play in partially observable card games. In *Proceedings of the 14th Annual Machine Learning Conference of Belgium and the Netherlands*, 45–50.

Rabin, M. 1993. Incorporating fairness into game theory and economics. *The American Economic Review* 1281–1302.

Roijers, D. M.; Vamplew, P.; Whiteson, S.; and Dazeley, R. 2013. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research (JAIR)* 48:67–113.

Roijers, D. M.; Whiteson, S.; and Oliehoek, F. A. 2015. Computing convex coverage sets for faster multi-objective coordination. *Journal of Artificial Intelligence Research (JAIR)* 52:399–443.

Varakantham, P.; Maheswaran, R.; and Tambe, M. 2005. Exploiting belief bounds: Practical POMDPs for personal assistant agents. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 978–985.

Wray, K. H., and Zilberstein, S. 2015a. Multi-objective POMDPs with lexicographic reward preferences. In *Proceedings of the 24th International Joint Conference of Artificial Intelligence (IJCAI)*, 1719–1725.

Wray, K. H., and Zilberstein, S. 2015b. A parallel point-based POMDP algorithm leveraging GPUs. In *Proceedings of the AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents*, 95–96.

Wray, K. H.; Pineda, L.; and Zilberstein, S. 2016. Hierarchical approach to transfer of control in semi-autonomous systems. In *Procceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 517–523.

Wray, K. H.; Witwicki, S. J.; and Zilberstein, S. 2017. Online decision-making for scalable autonomous systems. In *Proceedings of the 26th International Joint Conference of Artificial Intelligence (IJCAI)*, 4768–4774.