Optimal Coordination of Loosely-Coupled Self-Interested Robots

Ruggiero Cavallo
Div. Eng. & Applied Sci.
Harvard University
cavallo@eecs.harvard.edu

David C. Parkes
Div. Eng. & Applied Sci.
Harvard University
parkes@eecs.harvard.edu

Satinder Singh
Comp. Sci. & Eng.
University of Michigan
baveja@umich.edu

Abstract

We address the problem of optimally coordinating a group of loosely-coupled autonomous robots with private state information, when each robot is self-interested and acts only to maximize its own personal reward stream. The general solution we propose makes honest reporting of private information a best-response strategy and leads to the system-optimal outcome in equilibrium, while assuming the existence of a currency so that payments can be collected. We also provide a specialized mechanism for the case in which local robot models are Markov chains, using Gittins allocation indices to compute the system-optimal policy in time linear in the number of robots. The majority of the computation is distributed amongst the agents, with the coordinator primarily playing an enforcement role.

Introduction

To begin to address the problem of coordinating the behavior of individual robots in a group, one must first consider the circumstances under which that group has come into being, and the purpose each robot was created to serve. Currently, physical robots—to the extent that they exist—are almost always designed to serve very specific functions (e.g., "print the circuit", "vacuum the floor", etc.), and interaction with other robots is usually limited to purely cooperative settings. For instance, various rovers on Mars may be programmed to fulfill different goals, but in the end they are all there to do the bidding of the same group of scientists back on Earth.

Design of mechanisms for robot coordination has thus, naturally, focused on finding means of efficient communication and decision-making, with the assumption that individual robots are programmed to share information and perform tasks as the broader system-designer would like.

However, it is not difficult to call to mind current real-world scenarios involving software robots, or hypothetical future scenarios involving physical ones, in which a group of robots have been designed to serve very different purposes. That is, in addition to the typical interdependence that can exist between individual robots, there may also be *competition* within the group, leading to the possibility that individual robots will be programmed to behave strategically to maximize a utility function that is distinct from that of a broader system-designer (if present).

The problem of optimally coordinating the behavior of individuals in a group often essentially amounts to efficient communication (and perhaps determination) of relevant private information, so that local decisions can be made in a way that optimizes some global metric. Markets have been used throughout modern history to coordinate a wide array of human interactions: efficient allocation of food, hiring of labor, construction of municipal infrastructure, etc.

In robotics, markets have been used to minimize communication costs while achieving, e.g., desirable allocation of tasks to individuals (Zlot & Stentz 2006; Dias & Stentz 2001). But this "efficient communication" property of markets is actually just one of their key positive attributes. Certain markets also have the property of being robust against potential manipulation by self-interested agents; i.e., they act to align incentives of individuals with overall system-wide design goals. For example, coordination mechanisms with well aligned incentives can promote cooperative behavior amongst self-interested agents, with agents *choosing* to faithfully implement distributed planning algorithms and *choosing* to share truthful information about their local problem.

In this work we examine the design and application of truthful coordination mechanisms for multi-robot environments. We apply *mechanism design* to a multi-agent, sequential decision-making scenario, and we use the formalism of *Markov decision processes* (MDPs) to characterize agent models of the world. A significant portion of the paper is devoted to presenting work described in (Cavallo, Parkes, & Singh 2006), recast and specialized for robot domains.

In the next section we describe the problem we address in detail, and provide some necessary background in mechanism design and MDPs. We then present an optimal coordination mechanism for a general setting, and discuss the significant computational challenges that can arise. We present an important special case, that in which agent worlds can be modeled as Markov chains, for which computation can be (almost) completely distributed amongst the agents and optimal solutions are tractable. Finally, we make some concluding remarks.

Related Work

Brafman and Tennenholtz (1996) provide an early motivating scenario for self-interested robots, in the context of partially-controlled multi-agent systems. The authors consider a shared warehouse in which different robots, designed by different designers, need to coordinate on movements around the warehouse and placement of equipment.

Auction-based coordination mechanisms have been adopted for the coordination of *cooperative* multi-robot systems (Bererton *et al.* 2003; Tovey *et al.* 2005; Rabideau *et al.* 1999; Gerkey & Mataric 2002), adopting the perspective of using auctions as efficient *algorithms* for distributed planning and not for their incentive properties. This use parallels Wellman's seminal work on *market-oriented programming* (Wellman 1993), in which markets were adopted for distributed problem solving because of their ability to sustain optimal joint solutions while dealing with distributed private information. Prices provide concise aggregate summaries of the marginal effect of an agent's local action on the rest of the system.

A number of decomposition techniques for planning in stochastic domains, including methods specialized to multi-agent planning, are described in the literature (Kushner & Chen 1974; Boutilier 1999; Guestrin & Gordon 2002). These methods often work in the linear-programming formulation of the MDP planning problem, and leverage decomposition methods for large-scale linear programs, such as Benders and Dantzig-Wolfe decomposition (Lasdon 1970; Bradley, Hax, & Magnanti 1977).

Earlier work on online mechanism design (OMD) has considered dynamic environments, but with dynamic agent arrivals and departures, a single global state, and private information about agent rewards (Friedman & Parkes 2003; Parkes & Singh 2003). The persistence of agents coupled with the need for continued information from agents about their private state is what distinguishes the problem of coordinated planning from OMD. Dolgov and Durfee (2006) have studied resource allocation to self-interested agents with local problems modeled as MDPs, but in their setting this allocation is static and made in the initial period, and thus the incentive challenges are the same as those in standard (static) mechanism design.

Finally, (Parkes & Shneidman 2004) and (Shneidman & Parkes 2004) describe methods for distributing computation amongst self-interested agents in non-dynamic environments, while providing incentives so that agents will choose to faithfully perform the intended computation.

Set-Up and Background

A motivating story

Imagine a scenario, not too far in the future, in which a group of gold prospectors discovers that a particular 1-mile stretch of riverbed has significant gold deposits. At this point in time extraction of gold from riverbed has become highly automated via specialized robots. Each prospector owns a gold-searching robot and sends it to the river, at which point it acts autonomously until returning back to "the base" with its bounty. Certain portions of the riverbed are known to be more gold-laden than others, and robots are essentially in competition to work in the most desirable sections. To maintain order, a government enforcement agency seeks to coordinate the actions of the robot population. Moreover, the agency sees it as desirable that any chosen coordination scheme lead to the greatest possible gold harvest, with the specialisms of each robot matched to fit characteristics of different extraction tasks. How can these goals be achieved?

The problem we address, formally

We consider a scenario in which a group of n agents (we use "agent" and "robot" interchangeably) $I = \{1, \ldots, n\}$ interact with the world in various ways, each extracting reward at a rate dependent on the nature of its interaction, and each seeking to maximize its own reward over time.

More precisely, we assume each agent i's world model is represented as a Markov Decision Process (MDP) $M_i = <S_i, A_i, r_i, \tau_i>$, where:

- S_i is the set of all states of the world as it pertains to i
- A_i is the set of actions i is capable of executing
- $r_i : S_i \times A_i \rightarrow \mathcal{R}_{\geq 0}$ is a function specifying a real-valued reward for taking a particular action in a particular state
- $\tau_i : S_i \times A_i \times S_i \rightarrow [0, 1]$ is a transition function representing the probability that taking a given action in a given state will bring the world to any other state in the state space

Notice that there is uncertainty about the world, as represented in the potentially non-deterministic state transition function. A simple MDP example is portrayed in Figure 1. Here, the action space has just a single element (a single-action MDP is also called a *Markov chain*). There are three states, and transitions are non-deterministic.

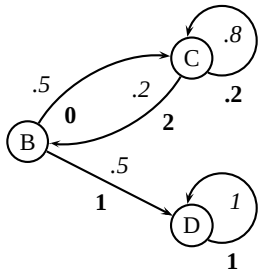


Figure 1: Example of a 3-state, single-action MDP (a Markov chain) with non-deterministic state transitions. Rewards are in bold font and transition probabilities in italics.

We assume an exponentially time-discounted valuation model in which a reward of x received t steps in the future is valued at $\gamma^t x$, where $0 \leq \gamma \leq 1$ is the discount factor. The goal of each agent is to maximize the expected discounted sum of rewards it receives over an infinite time-horizon. If the agent MDPs were completely independent, each agent i would then seek to execute a policy $\pi_i^* : S_i \rightarrow A_i$ such that:

$$\pi_i^* \in \arg\max_{\pi} \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r_i(s_i^t, \pi(s_i^t)) \,\Big|\, \pi\Big] \qquad (1)$$

where s_i^t is i's state at time t. Each agent i could determine π_i^* by, for instance, using value-iteration (see, e.g., Sutton & Barto 1998) to compute the optimal value function V_i^* for its MDP, from which the optimal action-choice can simply be read off:

$$\pi_i^*(s) \in \arg\max_{a \in A_i} \mathbb{E}\Big[r_i(s,a) + \gamma \sum_{s' \in S_i} \tau(s,a,s') V_i^*(s') \Big], \tag{2}$$

However, in a multi-agent setting there may be dependencies that exist between sets of agent behaviors. We consider a loose coupling in which interdependencies exist only through restrictions on joint actions (c.f. Singh & Cohn 1998). A natural example of joint-action feasibility constraints is when there is a shared resource required for execution, or when actions are location-dependent and only a single robot can be present in a given location at once.[1]

Taking maximization of system reward as our goal, a *coordinator* would like to enforce a policy that is optimal for the joint MDP M, which incorporates each of the component agent MDPs. Considering joint state space $S : S_1 \times \ldots \times S_n$ and action space $A : A_1 \times \ldots \times A_n$, we can define joint transition function τ and reward function r, where $\tau(s,a,s')$ is the probability that taking joint action a in joint state s brings the world to joint state s', and $r(s,a)$ is the total reward received by all agents when a is executed in s.[2] Assuming agents have a common discount factor, the coordinator's task at any time t_0 is to determine and execute the joint policy π^* that maximizes the discounted infinite sum of expected total system reward:

$$\pi^*(s) \in \arg\max_{\pi \in \Pi_f} \mathbb{E}\Big[\sum_{t=t_0}^{\infty} \sum_{i \in I} \gamma^{t-t_0} r_i(s_i^t, \pi(s^t)) \,\Big|\, \pi, s^{t_0} = s \Big],$$

for all states $s \in S$, searching across every π in Π_f, the set of all feasible joint policies (i.e., those that respect constraints on the joint actions).

Even when a coordinator capable of enforcing prescriptions for agent behavior is present, significant complications can arise if agents are *self-interested*. Agents typically hold some *private information*, knowledge of which is essential for optimal planning; for instance, the state that each agent is in at any point in time may not be publicly observable.[3] Thus the problem of coordinating a group of self-interested agents consists of providing appropriate incentives so that agents will choose to make truthful reports of local private information, in addition to the computational challenge of planning. This is the problem that can be addressed with an appropriate coordination *mechanism*.

Mechanism design for sequential environments

The field of mechanism design is concerned with bringing about globally-desirable outcomes, despite individuals in a system acting only to bring about locally-desirable ones. This requires finding a way to align the interests of each individual in the group with the welfare of the system as a whole. A typical way to do this is through transfer payments, assuming the existence of a currency.[4]

In a one-shot (i.e., single time-step) setting, a mechanism will typically consist of making some query to each agent regarding its private information, followed by selection of an outcome and determination of transfer payments according to the information that is reported. For instance, the basic Groves mechanism (Groves 1973) chooses the outcome that is optimal according to the information agents report, and sets the transfer payment for each agent equal to the value that all other agents (reportedly) reap from that selected outcome. The goals of all agents are completely aligned as each receives total payoff equivalent to the reward reportedly achieved by the entire group, so agents will report valuation information truthfully to allow the center to be successful in maximizing system reward.

The sequential environment we consider is more complex: here, at every time-step t an "outcome" decision must be made (i.e., a joint action a^t must be selected) and transfers may be executed, all of which can potentially depend on the entire execution history through t. We describe a *sequential coordination mechanism* $\Gamma = <\pi, T, \mu>$. which specifies a joint execution policy π, a transfer policy $T = T_1 \times \ldots \times T_n$ defining payments made to each agent, and a joint message space $\mu = \mu_1 \times \ldots \times \mu_n$ defining possible modes of communication from agents to coordinator. In the environments we consider, each agent's world model is considered common knowledge,[5] but there is private information consisting of each agent's local state; thus at every time-step a claim about each agent's current state will be solicited by the coordinator.

Each agent i has a strategy σ that maps a history h of the agent's state trajectory and transfer payments, and the current state s_i^t, to a message. That is, at time t agent i executes a strategy $\sigma_i(h, s_i^t) \in \mu_i$. In our coordination mechanism, truth revelation (in all states) is a *Markov Perfect Equilibrium* (MPE) (Maskin & Tirole 2001). The following is an informal definition.

[1] Another kind of interdependency, not considered here, could be through rewards, in settings in which one agent taking an action changes the reward to another agent for an action (consider another robot retrieving the gold before your robot).

[2] The actions adopted in this joint MDP to model interdependencies could be "macroactions" such as "go to section 1", with agents retaining autonomy on the sequence of actions that are interspersed in between macroactions (Sutton *et al.* 1999; Hauskrecht *et al.* 1998).

[3] One can similarly consider environments in which state is public but the reward functions (valuation information) are private.

[4] This need not be a "real" currency (such as dollars) as long as it enjoys the important properties of a currency (for instance as long as it is secure, transferable, and (relatively) stable).

[5] As discussed in the related paper (Cavallo, Parkes, & Singh 2006), this can be relaxed by including an initial step in which agents report their models to the center (which they will do truthfully in equilibrium), or finessed by distributing computation to agents.

Definition 1. (Markov Perfect Equilibrium) *A strategy profile $(\sigma_1^*, \ldots, \sigma_n^*)$ is an MPE if:*

a) (Perfect) *no agent can improve its expected utility by deviating in any state reachable either on or off the equilibrium path, given the other agents' strategies and the agent's belief about the other agents' private state and local MDP models;*

b) (Bayesian updating) *each agent updates its beliefs according to Bayes' rule where possible (e.g., while on the equilibrium path);*

c) (Markov) *an agent's strategy is conditioned only on the local state of the agent, and is history independent.*

In our environments Bayesian updating is unimportant because we bring truthful reporting into an MPE for all states, *whatever* the state, and thus for any private state of other agents. Moreover, since MPE is 'perfect', each agent can maximize its expected utility by truthful reporting even when other agents have previously deviated from truthful reporting.

We will refer to mechanisms that, like the basic Groves, achieve equilibrium outcomes that maximize total system reward as *system-optimal*. In a *truthful* mechanism, agents report true information in equilibrium. If a mechanism guarantees that no agent will be worse off (i.e., obtain negative net utility) from having participated, it is termed *ex post individual rational (IR)*; if the mechanism achieves this only in expectation, it is *ex ante IR*. Ex ante IR is a minimal requirement for inducing agents to participate in the mechanism. When the net payment made from the coordinator to the agents is guaranteed non-negative, a mechanism is termed *ex post budget balanced*; when this holds in expectation it is *ex ante budget balanced*; when net payments are exactly 0, a mechanism is *strongly budget balanced*.

Coordination in the General Setting

In this section we examine coordination mechanisms that are applicable to the general setting in which each agent's local problem is modeled as an MDP.

The first mechanism we describe is an extension of the basic Groves mechanism, introduced above, to a sequential, multi-agent coordination environment. Since agent MDPs are publicly known, optimal policy π^* can be computed; the challenge is that in order for the execution to be optimal decisions must reflect the *true* joint state at every time period.

Mechanism 1. (Sequential-Groves)

- *The planner computes an optimal joint policy π^*.*
- *At every time-step t:*
  1. *Each agent i reports to the planner a claim about its current state $\hat{s}_i^t$.*
  2. *The planner implements the joint action $a^t = \pi^*(\hat{s}^t)$.*
  3. *The planner pays each agent i a transfer:*

$$T_i(\hat{s}^t) = \sum_{j \in I \setminus \{i\}} r_j(\hat{s}_j^t, a_j^t)$$

Payments made by the coordinator to the agents are received immediately and as "reward", so an intrinsic reward of x plus a transfer payment of y at time t is valued equivalently to a reward of $x + y$ at t.

Theorem 1. *The Sequential-Groves mechanism is truthful, system-optimal, and ex post IR in Markov Perfect Equilibrium when agents have a common discount factor.*

Proof Sketch.[6] Let $\nu_i^{t_0}$ equal agent i's expected payoff at any time t_0 going forward, given the set of (known) agent MDPs $M = (M_1, \ldots, M_n)$ and current joint state s^{t_0} when all agents are reporting truthfully:

$$\nu_i^{t_0}(s^{t_0}, M) = \mathbb{E}_M\Big[\sum_{t=t_0}^{\infty}\Big\{\gamma^{t-t_0} r_i(s_i^t, \pi^*(s^t)) + \sum_{j \in I \setminus \{i\}} \gamma^{t-t_0} r_j(s_j^t, \pi^*(s^t))\Big\}\Big|\pi^*, s^{t_0} = s\Big]$$

$$= \mathbb{E}_M\Big[\sum_{t=t_0}^{\infty}\sum_{j \in I} \gamma^{t-t_0} r_j(s_j^t, \pi^*(s^t))\Big|\pi^*, s^{t_0} = s\Big]$$

This quantity is maximized, for all states s^{t_0} at all times t_0, by agent i reporting its true state when other agents do, because the joint policy will then maximize the expected utility to agent i (which is equal to the MDP value achieved by the joint policy). It is clear that this utility cannot be made greater by misreporting s_i^t, for any t, since the coordinator would then implement a policy that is based on faulty information, and thus potentially suboptimal.

The mechanism is trivially ex post IR, as each agent receives non-negative intrinsic reward from the world, and a grossly positive transfer payment from the coordinator. □

In the above, truthfulness and system-optimality follow from the fact that every agent's payoff is exactly equal to the payoff of the entire system. Since the coordinator's policy is designed to maximize this quantity, and since its only challenge in achieving this maximum is having access to accurate state information, agent payoffs are maximized when they report their current states truthfully.

If budget properties were of no concern, the *Sequential-Groves* mechanism would be quite satisfying; however, it will typically be extremely unrealistic to assume that a budget large enough to execute the specified payments will be available. Think of the gold-prospecting scenario: the coordinator would be making out massive payments on the order of n times the total value of the gold in the riverbed.

While the payments in *Sequential-Groves* ("Groves payments") are required in order to align the interests of all agents in the system, the Groves scheme fortunately also allows for imposition of a *charge* on each agent that can be used towards balancing the budget; this will do nothing to weaken the desirable equilibrium incentive properties of a coordination mechanism so long as the charge computed for each agent i is completely beyond i's influence.

The Vickrey Clarke Groves (VCG) mechanism for static settings specifies the charge for each agent i to be the total

[6] See (Cavallo, Parkes, & Singh 2006) for full proofs of all theorems in this paper (and other related ones).

reward that agents other than i *would have received* if i were not present (see, e.g., Jackson 2000); VCG thus has the appealing property that each agent's net payoff will equal its marginal contribution to total system welfare. Complications arise, however, when one tries to directly apply VCG to a sequential environment, as there are dependencies that exist between decisions made at one time-step and the space of possible outcomes that will be possible in future time-steps. Specifically, to preserve incentive properties we cannot use reported state information from agent i at *any* time-step throughout execution of the mechanism in determining i's charge. We propose the following variation on VCG for sequential coordination problems[7]:

Mechanism 2. (Sequential-VCG) *Identical to the Sequential-Groves mechanism, except at every time t, transfer payments are computed as follows:*

$$T_i(\hat{s}^t) = \sum_{j \in I \setminus \{i\}} r_j(\hat{s}_j^t, a_j^t) - (1-\gamma) V_{-i}^*(s^0)$$

Here, $V_{-i}^*(s^0)$ is the expected discounted sum of total value extracted for all agents *except* i, from time 0 under the system-optimal policy π^*, given models M. That is,

$$V_{-i}^*(s^0) = \mathbb{E}_M\Big[\sum_{t=0}^{\infty} \sum_{j \in I \setminus \{i\}} \gamma^t r_j(s_j^t, \pi^*(s^t)) \,\Big|\, \pi^*\Big]$$

Theorem 2. *The Sequential-VCG mechanism is truthful, system-optimal, ex ante IR, and ex ante strong budget-balanced in Markov Perfect Equilibrium when agents have a common discount factor.*

Proof. Truthfulness and system-optimality hold by truthfulness and system-optimality of *Sequential-Groves* plus the fact that each agent's charges are completely independent of reports that it makes. The expected payoff for each agent from time 0 (given models M) is as follows:

$$\nu_i^0(s, M) = \mathbb{E}_M\Big[\sum_{t=0}^{\infty} \gamma^t r_i(s_i^t, \pi^*(\hat{s}^t)) \,\Big|\, \pi^*, s^0 = s\Big] + \quad (3)$$

$$\mathbb{E}_M\Big[\sum_{t=0}^{\infty} \sum_{j \in I \setminus \{i\}} \gamma^t r_j(\hat{s}_j^t, \pi^*(\hat{s}^t)) \,\Big|\, \pi^*, s^0 = s\Big] - \quad (4)$$

$$\sum_{t=0}^{\infty} \gamma^t (1-\gamma) V_{-i}^*(s) \quad (5)$$

$$= \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t r_i(s_i^t, \pi^*(\hat{s}^t)) \,\Big|\, \pi^*, s^0 = s\Big] \quad (6)$$

The Groves payments (4) and the VCG charges (5) perfectly cancel out in expectation when agents report truthfully. As a result, net payments from the coordinator to the agents are 0, yielding ex ante strong budget-balance; total expected payoff for each agent i is exactly the (non-negative) intrinsic reward extracted by i under the system-optimal policy, so the mechanism is ex ante IR. □

[7]The *Sequential-VCG* mechanism diverges from a direct sequential analog of VCG in that charges computed for each agent i include hypothetical reward that i receives; this leads to stronger budget balance than would be achieved otherwise, and is possible here because we assume agent world models are public knowledge.

Realize that the flavor of IR achieved with this mechanism (and the specialized mechanism presented in the next section) is weak, that of *ex ante* IR. This is the cost that comes from performing mechanism design in these rich, dynamic environments where the "charge-back" payments collected from agents cannot be conditioned on the *actual* sequence of visited states.

However, in some domains a stronger form of IR will be possible. The *Sequential-VCG* mechanism will actually be ex ante IR from *any* time at which the agent MDPs are in a joint state (known to the planner) that is independent of anything that's ever been reported. Consider worlds in which a certain known-state is guaranteed to be visited repeatedly, for instance worlds that start in the same state every morning. In such cases we can provide ex ante IR periodically, rather than just once—agents will willingly "sign up" for the mechanism repeatedly, regardless of the interim execution, every time the known-state is visited.

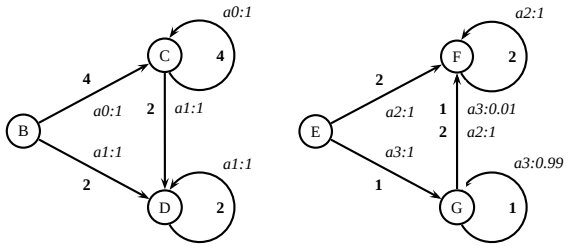
Similar examples can be provided if periodic "monitoring" is possible, so that the joint state is known for sure from time to time. In some robot environments this will be particularly relevant. One can imagine scenarios in which semi-autonomous robots are sent out in the field daily to perform some behaviors and make reports about their current location, physical state, etc.; sending a human observer out to verify the legitimacy of their claims may be expensive, but could be executed, say, once a day in order to realign the mechanism into ex ante IR for each agent going forward.

An example

We now illustrate why a coordination mechanism may be necessary, and how the one we propose works. Figure 2 depicts a 2-agent scenario, where each agent's world model has 3 states and 2 actions, and the initial states are B and E. We take discount factor $\gamma = 0.9$, and consider the coordination problem that arises when actions $a0$ and $a2$ cannot be performed simultaneously. We first construct the joint MDP, as in Figure 3, and then compute the system-optimal policy, given in Table 1.

The *Sequential-VCG* mechanism *pays* agent1 the reward agent2 reports having achieved each period, and vice versa. Under the system-optimal policy, with high probability for many time-steps from the beginning of execution agent1's payment will be 4 and agent2's payment will be 1. Each period the mechanism charges agent1 $(1-\gamma) \cdot V_{-2}^*(BE) = 3.8$, and charges agent2 $(1-\gamma) \cdot V_{-1}^*(BE) = 1.1$. The payments and charges cancel out exactly in expectation, leaving each agent with payoff equal to the intrinsic reward extracted under the system-optimal policy.

Now consider the case where the true joint state is CG. It is clear that the system-optimal policy executes joint action $a0a3$, as with very high probability reward 5 will be yielded each period going forward, while alternative $a1a2$ will yield reward 4 in all periods going forward. But notice that $a1a2$ would yield greater intrinsic reward for agent2



(a) Agent 1 (b) Agent 2

Figure 2: MDPs for a 2-agent world, each with 3 states.

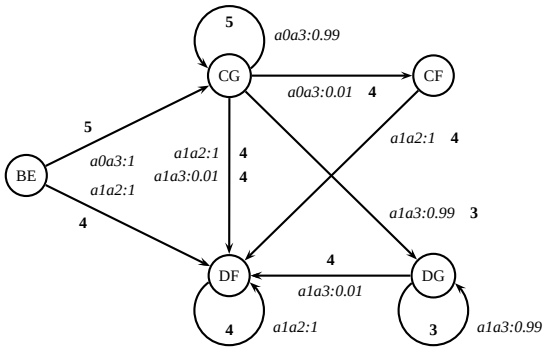


Figure 3: Joint MDP constructed from component MDPs illustrated in Figure 2. Joint state XY denotes agent1's MDP in state X and agent2's in state Y; joint action $axay$ denotes x taken by agent1 and y taken by agent2. Incompatibility of actions $a0$ and $a2$ is reflected by omission of $a0a2$.

state	π^*	V^*	V^*_{-1}	V^*_{-2}
BE	$a0a3$	49	11	38
CG	$a0a3$	49	11	38
DF	$a1a2$	40	20	20
DG	$a1a3$	31	11	20
CF	$a1a2$	40	20	20

Table 1: Optimal policy for joint MDP, and optimal value functions, with discount factor $\gamma = 0.9$.

than would $a0a3$, and if the system were in state CF then $a1a2$ would be the only possibility (since $a0a2$ is infeasible). If we did not make the Groves payments, aligning the interests of agent2 with that of the system as a whole, agent2 would have incentive to report being in state F rather than G when agent1 is in state C. Thus one can observe even in this simple example the essential role that the payments play in enabling realization of an optimal joint plan.

Efficiently solving groups of interdependent MDPs

In *Sequential-VCG*, we have a coordination mechanism that achieves system-optimality in equilibrium, while (in expectation) requiring no external budget to implement. The primary remaining challenge is computational tractability of determining an optimal policy. We explore in detail one generally tractable domain (Markov chains) in the next section, and here briefly describe promising solution methodologies that have been proposed for problem decomposition with cooperative agents. There are two important considerations in applying these methods in our setting with self interest:

a) are the decomposition methods already factored, or can they be re-factored, to ensure that agents have correct incentives to choose to follow them?

b) can the decomposition methods be leveraged to allow for planning without each agent in turn, in order to enable computation of payments?

Following Guestrin and Gordon (2002) we can divide prior work into that on *serial decomposition* in which one agent is active at any given time (Kushner & Chen 1974), and *parallel decomposition* in which multiple agents can be active at the same time (Singh & Cohn 1998; Meuleau *et al.* 1998).

Parallel decomposition is more relevant to multi-robot coordination. Singh and Cohn (1998) consider the same cross-product representation for the global MDP as we adopt, and place constraints on joint actions. Admissible estimates from subproblems are used to accelerate planning, however their algorithm is not fully factored in that it requires an explicit representation of the joint state space. Meuleau et al. (1998) specialize to settings in which the only coupling is via resource constraints, and are able to find approximate solutions to large problems through a combination of offline and online computation. Approximations can pose some new challenges in the context of self-interested agents, for instance causing the strong truthful equilibrium properties to unravel. Future work will need to explore these issues.

More recently, Guestrin and Gordon (2002) describe decomposition methods based on linear-programming decomposition techniques, such as those due to Benders and Dantzig-Wolfe (Lasdon 1970; Bradley, Hax, & Magnanti 1977, see). Dantzig-Wolfe decomposition methods often have a market interpretation, with complicating constraints between subproblems priced by a coordinator and used to modify local agent problems such that an optimal joint solution can be constructed (Dantzig & Wolfe 1960; Dantzig 1963). Indeed, Bererton et al. (2003) have recently provided an auction interpretation of the Dantzig-Wolfe decomposition for MDPs. None of these methods are incentive-compatible in our sense, and an important next step will investigate the integration of these methods into *Sequential-VCG* in order to handle self-interest.

Coordination in Markov Chain Settings

We now examine the case in which all local agent models are Markov chains (i.e., all are MDPs in which just a single action per agent is available for every local state), and in which only one can be activated at a time. In a Markov chain setting agents do not face an action-selection problem, but the coordination problem remains as a decision must be made at every time-step regarding which chain to activate. This setting is appealing because it allows a tractable coordinated planning algorithm based on index policies.

It is not hard to imagine settings in which robots have been programmed to behave deterministically given any particular state of the world; in such cases world models are

Markov chains. Consider, for instance, software robots that are using a super-computer to perform computational tasks on behalf of their designers; the way computation should proceed for any robot's task is completely known, but a decision (coordination) regarding which robot should be granted access to the super-computer must be made at every point in time. The state of each robot reflects the (non-deterministic) partial results from the computation performed so far.

We can formalize the specifics of this environment by positing that each agent i has an MDP with action space $A_i = \{a_i, a_{\text{null}}\}$, and that any admittable policy π specifies, at any time t, a joint action a^t in which all but one agent's action is a_{null}. For convenience we write $\pi(s) = i$ to denote that policy π activates agent i's Markov chain when the world is in joint state s. We let $r(s_i)$ denote the reward i receives when its chain is activated in state s_i.

Gittins (1974) showed that in this setting (minus the self-interest) optimal planning is tractable. Specifically, he showed that one can compute an index (which we will call the *Gittins index*) independently for each Markov chain given its current state, such that the optimal policy consists of always activating the chain with highest index. In this way the computational complexity of computing an optimal policy grows only linearly in the number of agents.

Theorem 3. (Gittins & Jones 1974; Gittins 1989) *Given Markov chains $M_1, \ldots, M_n$ in states $(s_1, \ldots, s_n)$ respectively, there exist independent functions $G_1(M_1, s_1), \ldots, G_n(M_n, s_n)$ such that the optimal policy $\pi^*(s) = \arg\max_i G_i(M_i, s_i)$.*

Several methods of computing Gittins indices are known. For instance, in (Katehakis & Veinott 1987) a special type of two-action, k-state MDP is formulated for every state in a k-state Markov chain, the optimal value of which corresponds to the Gittins index.

Besides computational tractability, the decomposition aspect of Gittins' solution is of particular interest in a multi-agent setting, as almost all computation can be distributed amongst the agents. In a robotics setting, if each robot is capable of computing its Gittins indices, the only coordination necessary is to determine which index is highest at every time-step, and to potentially compute and execute transfer payments to properly align agent incentives.

To compute VCG charges in this setting the coordinator must determine which Markov chain *would have* been activated in n hypothetical worlds, in which each agent is removed in turn. In the world without some agent i, the only difference in the optimal policy is that whenever i's Gittins index is highest, the Markov chain with second highest index is chosen instead. We can compute the expected value achieved by the system in such a world by *simulating* what would have happened. Again, it does not retain the right incentive properties to use the *actual* (real-world) indices to determine an agent's marginal effect on the other agents.

Consider simulation of a policy that is optimal in a world without i, and let $X_{\pi^*_{-i}}$ be the simulated sample trajectory. Let $r(X, t)$ denote the system-reward during the t^{th} step of trajectory X. We propose the following mechanism for optimal coordination in Markov chain settings when agents have computational capacity, where m sample trajectories[8] $\{X^1_{\pi^*_{-i}}, \ldots X^m_{\pi^*_{-i}}\}$ are maintained for every agent i:

Mechanism 3. (Distributed-Gittins-VCG)

- *Each agent i computes and reports a claim to the planner about Gittins indices $\hat{G}_i(M_i, s_i), \forall s_i \in S_i$*
- *At every time-step t:*
  1. *Each agent i reports to the planner a claim about its current state $\hat{s}^t_i$.*
  2. *The planner activates Markov chain:*

$$i^* \in \arg\max_{i \in I} \{\hat{G}_i(M_i, \hat{s}^t_i)\}$$

  and simulates the next action in each of the $n \cdot m$ sample trajectories.
  3. *The planner pays each agent i a transfer:*

$$T_i(\hat{s}^t) = \begin{cases} -\sum_{k=1}^{m} \dfrac{r(X^k_{\pi^*_{-i}}, t)}{m} & \text{for } i^* \\ r(\hat{s}^t_{i^*}) - \sum_{k=1}^{m} \dfrac{r(X^k_{\pi^*_{-j}}, t)}{m} & \text{for } j \in I \setminus \{i^*\} \end{cases}$$

Theorem 4. *The Distributed-Gittins-VCG mechanism is truthful, system-optimal, ex ante IR, and ex ante weak budget-balanced in Markov Perfect Equilibrium when agents have a common discount factor.*

Proof Sketch. As in the general setting, each agent receives Groves payments equal to the total reward received by other agents (here, only one agent receives reward per time-step). Agent i's charge term is again independent of i's reports, as information only from the other agents is used in simulating sample trajectory $X_{\pi^*_{-i}}$. Agents thus want system-welfare to be maximized, which brings truthful reporting of both reward and Gittins index information into equilibrium. In expectation the charges computed for each agent i will fall between 0 and the intrinsic reward received by i, as a policy that is optimal without considering i cannot be better for the entire system than one that takes all agents into account. This yields ex ante IR and weak budget-balance. □

In the version of *Distributed-Gittins-VCG* we have presented, agents compute and communicate Gittins indices up front, but this is not necessary; the mechanism properties maintain if we elicit index information *online*. That is, we can instead ask agents for the index of their current state at each time-step (along with indices for sample trajectories). See (Cavallo, Parkes, & Singh 2006) for a full discussion.

Conclusions

In this paper we addressed the problem of coordinating a group of self-interested robots in a way that yields maximum total social welfare. We have provided solutions that "disarm" the impact of self-interest on the behavior of robots,

[8] As m is increased the variance of the samples will decrease, but any $m \geq 1$ will achieve the properties in Theorem 4.

transforming competitive environments into "team games". Importantly, the methods we propose do not, in expectation, require any external budget to implement. The methods are applicable to a wide array of domains, including current scenarios where software robots compete for control of a shared resource and future scenarios of physical robot coordination problems where self-interest is a factor.

The specific algorithm used in determination of the system-optimal joint execution policy is not important to the incentive properties our proposals achieve, and distributed algorithms are possible. In the Markov chain setting, we proposed a Gittins index-based policy computation method that has several desirable properties. In this mechanism the system-optimal policy can be computed in time linear in the number of robots, and the computation is almost completely distributed amongst the robots themselves.

There are many interesting directions for future work. We are currently examining mechanisms that have desirable equilibrium properties even when the policy followed is suboptimal; such mechanisms are of interest because they would work with approximate MDP solutions. In addition, we are interested in finding a synthesis between known MDP decomposition methods and our mechanism framework, as well as developing concise methods for value representation in resource- and action-constrained settings. It will also be interesting to investigate alternate models of agent coupling, for instance with interactions through states and rewards.

References

Bererton, C.; Gordon, G.; Thrun, S.; and Khosla, P. 2003. Auction mechanism design for multi-robot coordination. In *Proc. 17th Annual Conf. on Neural Inf. Processing Systems (NIPS'03)*.

Boutilier, C. 1999. Sequential optimality and coordination in multiagent systems. In *Proceedings of the 16th International Joint Conference on Artificial In telligence (IJCAI'99)*, 478–485.

Bradley, S.; Hax, A.; and Magnanti, T. 1977. *Applied Mathematical Programming*. Addison-Wesley.

Cavallo, R.; Parkes, D. C.; and Singh, S. 2006. Optimal coordinated planning amongst self-interested agents with private state. In *Proceedings of the Twenty-second Annual Conference on Uncertainty in Artificial Intelligence (UAI'06)*.

Dantzig, G., and Wolfe, P. 1960. Decomposition principle for dynamic programs. *Operations Research* 8(1):101–111.

Dantzig, G. 1963. *Linear Programming and Extensions*. Princeton University Press.

Dias, M., and Stentz, A. 2001. A market approach to multirobot coordination. Technical Report, CMU-RI-TR-01-26.

Dolgov, D. A., and Durfee, E. H. 2006. Resource allocation among agents with preferences induced by factored mdps. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-06)*.

Friedman, E., and Parkes, D. C. 2003. Pricing WiFi at Starbucks–Issues in online mechanism design. In *Fourth ACM Conf. on Electronic Commerce (EC'03)*, 240–241.

Gerkey, B. P., and Mataric, M. J. 2002. Sold!: Auction methods for multi-robot coordination. *IEEE Transactions on Robotics and Automation, Special Issue on Multi-robot Systems* 18(5):758–768.

Gittins, J. C., and Jones, D. M. 1974. A dynamic allocation index for the sequential design of experiments. In *Progress in Statistics*, 241–266. J. Gani et al.

Gittins, J. C. 1989. *Multi-armed Bandit Allocation Indices*. New York: Wiley.

Groves, T. 1973. Incentives in teams. *Econometrica* 41:617–631.

Guestrin, C., and Gordon, G. 2002. Distributed planning in hierarchical factored MDPs. In *Proc. of the Eighteenth Annual Conf. on Uncertainty in Artificial Intelligence (UAI'02)*, 197–206.

Hauskrecht, M.; Meuleau, N.; Boutilier, C.; Kaelbling, L. P.; and Dean, T. 1998. Hierarchical solution of markov decision processes using macro-actions. In *Proc. of the Fourteenth Annual Conf. on Uncertainty in Artificial Intelligence (UAI'98)*, 220–229.

Jackson, M. O. 2000. *Mechanism Theory, In The Encyclopedia of Life Support Systems*. EOLSS Publishers.

Katehakis, M. N., and Veinott, A. F. 1987. The multi-armed bandit problem: decomposition and computation. *Mathematics of Operations Research* 22(2):262–268.

Kushner, H. J., and Chen, C.-H. 1974. Decomposition of systems governed by markov chains. *IEEE Transactions on Automatic Control* 19(5):501–507.

Lasdon, L. 1970. *Optimization theory for large systems*. MacMillan Publishing Company.

Maskin, E., and Tirole, J. 2001. Markov perfect equilibrium: I. observable actions. *Journal of Economic Theory* 100(2):191–219.

Meuleau, N.; Hauskrecht, M.; Kim, K.-E.; Peshkin, L.; Kaelbling, L.; Dean, T.; and Boutilier, C. 1998. Solving very large weakly coupled markov decision processes. In *AAAI/IAAI*, 165–172.

Parkes, D. C., and Shneidman, J. 2004. Distributed implementations of vickrey-clarke-groves mechanisms. In *Proc. 3rd Int. Joint Conf. on Autonomous Agents and Multi Agent Systems*, 261–268.

Parkes, D. C., and Singh, S. 2003. An MDP-based approach to Online Mechanism Design. In *Proc. 17th Annual Conf. on Neural Information Processing Systems (NIPS'03)*.

Rabideau, G.; Estlin, T.; Chien, S.; and Barrett, A. 1999. A comparison of coordinated planning methods for cooperating rovers. In *Proceedings of AIAA Space Technology Conference*.

R.Brafman, and M.Tennenholtz. 1996. On partially-controlled multiagent systems. *Journal of Artificial Intelligence Research* 4:477–507.

Shneidman, J., and Parkes, D. C. 2004. Specification faithfulness in networks with rational nodes. In *Proc. 23rd ACM Symp. on Principles of Distributed Computing (PODC'04)*.

Singh, S., and Cohn, D. 1998. How to dynamically merge markov decision processes. In *Advances in Neural Information Processing Systems 10 (NIPS)*, 1057–1063.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Sutton, R.; Singh, S.; Precup, D.; and Ravindran, B. 1999. Improved switching among temporally abstract actions. In *Advances in Neural Information Processing Systems 11 (NIPS)*, 1066–1072.

Tovey, C.; Lagoudakis, M. G.; Jain, S.; and Koenig, S. 2005. Generation of bidding rules for auction-based robot coordination. In *Proc. of the 3rd International Multi-Robot Systems Workshop*.

Wellman, M. P. 1993. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research* 1:1–23.

Zlot, R., and Stentz, A. 2006. Market-based multirobot coordination for complex tasks. *International Journal of Robotics Research, Special Issue on the 4th International Conference on Field and Service Robotics*, 25(1):73–101.