Dynamic Social Choice with Evolving Preferences[*]

David C. Parkes [†] Ariel D. Procaccia[‡]

Abstract

Social choice theory provides insights into a variety of collective decision making settings, but nowadays some of its tenets are challenged by Internet environments, which call for dynamic decision making under constantly changing preferences. In this paper we model the problem via Markov decision processes (MDP), where the states of the MDP coincide with preference profiles and a (deterministic, stationary) policy corresponds to a social choice function. We can therefore employ the axioms studied in the social choice literature as guidelines in the design of socially desirable policies. We present tractable algorithms that compute optimal policies under different prominent social choice constraints. Our machinery relies on techniques for exploiting symmetries and isomorphisms between MDPs.

1 Introduction

Social choice theory has its roots in the writings of the marquis de Condorcet and the chevalier de Borda, and over the centuries has evolved so that nowadays we have a comprehensive mathematical understanding of social decision making processes. However, social choice theory falls short in the context of today's online communities. The Internet and its myriad of applications has created a need for fast-paced, dynamic social decision making, which begs the question, is it possible to augment social choice theory to make it relevant for this new reality?

In our model of dynamic social decision making, a sequence of decisions must be made in the context of a population with constantly changing preferences, where the evolution of future preferences depends on past preferences and past decisions. Consider, for example, decision making in collaborative projects. Examples include collaborative computational science involving several groups of researchers, where the alternatives represent the different experiments that can be run on shared computational resources, and open source software projects such as Debian.[1] In collaborative science projects, it is natural that preferences over which experiment to run next change, and change in a way that depends, likely probabilistically, on previous experiments. Debian already employs an elaborate voting rule in order to facilitate decisions on issues ranging from the choice of a logo to which features to implement. Debian, as an example of open source software projects, naturally

[*]Between September 2010 and June 2011 this work was presented in seminars at the following institutions (listed in chronological order): MIT, Harvard, Brown, UC Berkeley, Tel Aviv University, Hebrew University, Technion, Weizmann Institute, Duke, Stanford, Carnegie Mellon, University of Toronto.

[†]School of Engineering and Applied Sciences, Harvard University, 33 Oxford Street, Cambridge, MA 02138, email: parkes@eecs.harvard.edu.

[‡]Computer Science Department, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, email: arielpro@cs.cmu.edu.

[1]`http://www.debian.org`

supports dynamic social choice, where the goal is to sequentially implement features in a way that improves the system as a whole, while keeping the many developers satisfied with the decision making process.

As a running example, we consider online public policy advocacy groups, which are quickly gaining popularity and influence on the web and in social networks such as Facebook (via the application Causes); to be concrete we focus on MoveOn (`www.moveon.org`). MoveOn boasts more than five million members but employs only a handful of staffers. Ideally the causes or issues that are advocated by MoveOn directly stem from the collective preferences of the members. A salient feature of MoveOn is that the time frame between deciding on a cause and acting on it is very short. Crucially, when a cause is chosen and advocated the preferences of the members will usually change, and this should have an impact on the next cause to be chosen. So, we are faced with a situation where both the current cause and the preferences of the members are constantly shifting. This calls for a consistent and socially desirable mechanism that sequentially selects the current cause given the current preferences of MoveOn members.

We believe that the scope of this work is wider, extending beyond social choice for Internet systems. Still in the political context, one can imagine providing a decision support system to a party leader in the US Senate or House of Representatives. The goal of the leader is to align the members of his party behind a bill. This is a complex process because the positions of party representatives are constantly shifting in response to polls and a variety of other decisions that are being made.

Dynamic social choice via factored MDPs. The common social choice setting concerns a set of agents (members, in the example) and a set of alternatives (causes or issues, in the example); the preferences of each agent are given by a ranking of the alternatives. A *preference profile* is a collection of the agents' preferences. The outcome is determined by a *social choice function*, which maps a given preference profile to the winning alternative.

We introduce dynamic preferences into this static setting by representing the preferences of each agent as a random variable whose current value is the agent's current ranking of the alternatives. Each agent is associated with a Markovian transition model that stochastically determines how the agent's preferences are updated based on the agent's current preferences and the currently selected alternative (c.f. [6] for dynamic incentive mechanisms). This model can be expressed using the formalism of factored Markov decision processes (MDP) [4]; we refer to the special case that we deal with as a *social choice MDP*.

One of the virtues of the social choice MDP model is that a state of the MDP, which represents the current value of each of the agents' random variables, corresponds to a preference profile. In other words, a static snapshot of the social choice MDP at any given time reduces, in a sense, to the traditional social choice setting. As in the traditional setting, the set of actions available in each state coincides with the set of alternatives.

We say that agents that have identical transition models share the same *type*. In our running example, we imagine MoveOn staffers categorizing their membership according to carefully selected features (e.g., "moderate" or "progressive") and eliciting the vector of features from each member. Each vector of features can then be associated with a transition model in an approximate way. Finding good features and constructing the associated transition models is a nontrivial problem that is beyond the scope of this paper; in the sequel we take the social choice MDP as a given (but see Section 6 for some discussion of this point).

A deterministic (stationary) policy in an MDP maps each state to the action taken in this state. The crucial insight, which will enable us to relate the dynamic setting to traditional social choice theory, is that *a deterministic policy in a social choice MDP corresponds to a social choice function.* This connection allows us to leverage the extensive literature on the design of desirable social choice functions. Specifically, social choice functions are usually compared on the basis of their axiomatic properties. For example, a social choice function is *Pareto optimal* if, whenever all the agents prefer x to y, y would not be elected. It is self evident that this property is highly desirable in the dynamic setting as well; the members of MoveOn would be outraged if at any point they unanimously preferred one cause to another but the organization aggregated their preferences to support the inferior cause. We will consider a number of natural axiomatic properties, seeking to impose them on the solution space of social choice MDPs.

The final component of a social choice MDP model is a reward function, which associates a given action taken in a given state with a reward; the goal is to optimize the infinite sum of discounted rewards. The existence of such an explicit objective is novel from a social choice perspective. For instance, the designer (MoveOn staffers in our example) may have a favorite alternative in mind, and receive a positive reward only when that alternative is selected. Alternatively, consider the function which rewards social consensus, e.g., provides a positive reward only when there is a Condorcet winner (an alternative that is preferred to every other alternative by a majority of agents) and this alternative is selected as the current action. This is a natural reward function in the context of our running example, as advocacy becomes much more powerful when millions of people are united behind a common cause.

Unfortunately, a policy that optimizes the discounted rewards may not satisfy basic social choice properties, and hence may be undesirable as a social decision making mechanism. The key algorithmic challenge is therefore:

> *Given a social choice MDP, tractably compute an optimal deterministic policy subject to given social choice constraints.*

Our results and techniques. Observe that the state space of a social choice MDP is huge; if there are n agents and m alternatives then its cardinality is $(m!)^n$. To make things manageable we assume (in our algorithmic results) that there is only a constant number of alternatives, i.e., $m = \mathcal{O}(1)$, and moreover that the number of types, that is, different agent transition models, is also bounded by a constant. Note that this still allows each agent of the same type to be in one of the $m!$ possible preference states. We wish to design algorithms that are polynomial time in n. We argue that these assumptions are consistent with our motivation: while the number of MoveOn members is in the millions, the number of causes is usually rather small, and the number of types is limited by the number of features, which must be individually elicited from each member.

Our results revolve around the symmetries induced by the types. Intuitively, two states are equivalent in terms of their preferences and next-state transitions if one can be obtained from the other by a permutation of the preferences of agents of the same transition type. In order to maintain the symmetries between such states we concentrate on *anonymous* reward functions, which are indifferent to the identities of the agents; both reward functions given as examples above are anonymous. We strongly rely on the techniques developed by Zinkevich and Balch [25], who treated symmetries via equivalence relations on states. In some of our proofs we take a somewhat different view of symmetries via isomorphisms between MDPs, à la Ravindran and Barto [20].

In Section 4 we identify social choice axioms that are *local*, in that they restrict individual states to certain actions in a way that is independent of the actions selected for other states. A local axiom is *anonymous* if it is indifferent to the identities of the agents. Some of the most prominent social choice axioms are local and anonymous, e.g., Pareto-optimality, unanimity (if all the agents rank alternative a first then a should be elected), and Condorcet-consistency (if a is a Condorcet winner then a should be elected). Our main result of Section 4 is that, given a social choice MDP, an anonymous reward function, and an anonymous local axiom, it is possible to compute an optimal policy that satisfies the axiom in polynomial time in n. The symmetric structure on the MDP, which follows from agent types and an anonymous reward function, is retained with local, anonymous axioms, which can be handled by imposing appropriate restrictions on the actions available in each set of equivalent states. On the way we develop rather general machinery that is used heavily in Section 5.

Section 5 tackles two very basic social choice axioms that are nonlocal: ontoness (every action is selected in some state) and nondictatorship (there is no agent whose most preferred alternative is selected in every state). We design polynomial time search algorithms that check different possibilities of restricting states to actions in a way that rules out undesirable policies. One of the obstacles we face is that these restrictions break the symmetries between states. Our algorithms therefore amount to a balancing act, where we must on one hand refine the equivalence relations on states and enable the imposition of particular restrictions, and on the other hand avoid very detailed refinements that lead to bad running time. We prove that, given a social choice MDP and an anonymous reward function, we can find an optimal policy that is onto or nondictatorial in polynomial time in n. Unlike the result mentioned above, here the policy is not uniformly optimal, i.e., is not optimal on every state, but is rather optimal under the assumption that the initial state is some given state; this assumption is easily justified in our running example, where the initial state can be elicited from MoveOn members. We believe that the proofs of the algorithms' properties are quite straightforward once the algorithms themselves are in place.

As an aside, in the sequel we insist on computing *deterministic* policies, since these map to social choice functions. In general, imposing constraints on MDPs can lead to optimal policies that are randomized, and although there is some discussion of randomized social choice functions in the literature (see, e.g., [12]) the vast majority of work concerns deterministic social choice functions. In fact, the axioms that we are interested in are not well-defined with respect to randomized social choice functions. Asking that the policy be deterministic can be seen as an additional constraint.

Related work. Our work is conceptually related to the literature on dynamic incentive mechanisms (see, e.g., [15] for an overview). This literature investigates two kinds of dynamics: where there is external uncertainty, i.e., the set of agents or alternatives is dynamically changing (see, e.g., [16]); and where there is internal uncertainty, i.e., the preferences of agents are dynamic (see, e.g., [6, 3]). Our model is inspired by the latter line of work, where the preferences of agents also evolve via an MDP. However, the focus of the work on dynamic incentive mechanisms is fundamentally different; the question is how to design monetary transfers to the agents in a way that incentivizes truthful reporting of preferences across time periods. This question is in fact moot in our framework (where in particular monetary transfers are unavailable), as it is well known that it is impossible to design reasonable social choice functions that are truthful [11, 21]. The work on *automated mechanism design* (see, e.g., [7]) is another strand of research in the mechanism design literature that is more distantly related to our work.

A body of work in the social choice literature studies a requirement known as *population monotonicity* in a dynamic population, which stipulates that if some agents arrive or leave, the remaining agents all strictly gain or all strictly lose, that is, there is some solidarity among the remaining agents; this idea originated in the work of Thomson [23]. More recently, Tennenholtz [22] studied a so-called *dynamic voting* setting where agents arrive and vote sequentially. He defines a dynamic social choice function that selects a social ranking after each agent has voted, and specifies some axioms that such a function must satisfy. The new axioms impose requirements on how the social ranking may change as additional votes are incorporated. Tennenholtz's main result is a characterization of a specific dynamic social choice function in terms of these axioms. Note that the foregoing papers are concerned with a dynamic population, with agent arrivals and departures, rather than agents with dynamically changing preferences. Nor do they consider a probabilistic model of a population of agents.

Jackson and Sonnenschein [14] consider a static population of agents and a sequence of decision problems, where agents' preferences are additively separable and independently distributed across the problems. They design a social choice mechanism that achieves ex ante Pareto efficiency and (approximate) truthful reporting in the limit, as more decision problems are linked together. Unlike our work, the focus is on incentive constraints and ex ante Pareto efficiency. Nor do these authors consider computational constraints. Moreover, this earlier model crucially relies on the independence of decision problems because the mechanism constrains agent reports to match the underlying distribution. In our dynamic social choice problem, agent preferences evolve according to the alternative selected in each period.

The problem of constructing optimal policies for *constrained MDPs* has received some attention (see, e.g., [1, 9, 10]). Unfortunately the constraints in question usually have a specific structure; to the best of our knowledge the constraints considered in previous work are not sufficiently general to capture our social choice constraints, and moreover, as mentioned above, we insist on deterministic policies.

A very recent paper by Boutilier and Procaccia [5] builds on our model of dynamic social choice to relate a notion of distances in social choice to an operational measure of of social desirability. Very informally, they show that a social choice function selects alternatives that are closest to being consensus winners if and only if that social choice function is an optimal policy of a specific, arguably natural, social choice MDP. The conceptual implication is that alternatives that are close to consensus can become winners faster in a dynamic process.

2 Preliminaries

In this section we formally introduce basic notions from the MDP literature and social choice theory.

2.1 Markov Decision Processes

Below we briefly review the basics of Markov decision processes; the reader is referred to Puterman [18] for more details.

A *Markov decision process (MDP)* is a 4-tuple $\mathcal{M} = (\mathcal{S}, A, R, P)$ where $\mathcal{S}$ is a finite set of states; A is a finite set of actions; $R : \mathcal{S} \times A \to \mathbb{R}$ is a reward function, where for $s \in \mathcal{S}$ and $a \in A$, $R(s, a)$ is the reward obtained when taking action a in state s; and P is the transition function, where $P(s'|s, a)$ is the probability of moving to state s' when action a is taken in state s. Note that the transition function is Markovian in that it depends only on the current state and not on the history.

A *deterministic policy* is a function $\pi : \mathcal{S} \to A$, which prescribes which action $\pi(s)$ is taken in state $s \in \mathcal{S}$. This definition implicitly assumes that the policy is also stationary, that is, it does not depend on the history.

There are several variations on how the cumulative reward is calculated. We consider the most common approach where there is an infinite horizon and a discount factor $\gamma \in [0, 1)$. Given an MDP $\mathcal{M}$, a policy π is associated with a value function $V_\pi : \mathcal{S} \to \mathbb{R}$, where $V_\pi(s)$ is the cumulative discounted reward that is obtained if the initial state is $s \in \mathcal{S}$ and the action prescribed by π is taken at each step. The optimal policy π^* satisfies the so-called Bellman equations [2],

$$\forall s \in \mathcal{S},\ V_{\pi^*}(s) = \max_{a \in A} \left[R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) \cdot V_{\pi^*}(s') \right].$$

It is known that for any (unconstrained) MDP there is an optimal policy that is deterministic. Such an optimal policy can be found in polynomial time, e.g., by computing the optimal values via linear programming and then greedily assigning actions that achieve the maximum at every state.

2.2 Social choice

Let $N = \{1, \ldots, n\}$ be a set of *agents*, and let A be a set of alternatives where $|A| = m$; we overload the notation A, as in the sequel the actions in our MDP coincide with the set of alternatives. Each agent is associated with strict linear preferences $\succ_i$ over the alternatives, that is, a strict ranking of the alternatives; $x \succ_i y$ means that agent i prefers x to y. Let $\mathcal{L} = \mathcal{L}(A)$ denote the set of strict linear preferences over A. A collection $\vec{\succ} = (\succ_1, \ldots, \succ_n) \in \mathcal{L}^n$ of the agents' preferences is called a *preference profile.* A *social choice function* is a function $f : \mathcal{L}^n \to A$ that designates a winning alternative given a preference profile. Given $\succ \in \mathcal{L}$ we denote by $\text{top}(\succ)$ the alternative that is most preferred in $\succ$.

A prominent approach in social choice theory compares social choice functions based on their axiomatic properties. Two properties are considered absolutely essential and are satisfied by all commonly studied social choice functions. A social choice function f is *onto* if for every $a \in A$ there exists $\vec{\succ} \in \mathcal{L}^n$ such that $f(\vec{\succ}) = a$, that is, every alternative can be elected. A social choice function f is *dictatorial* if there exists an agent $i \in N$ such that for every $\vec{\succ} \in \mathcal{L}^n$, $f(\vec{\succ}) = \text{top}(\succ_i)$; f is *nondictatorial* if there is no such agent.

Below we define some other prominent axioms; the first two axioms provide a notion of consensus, and require that a social choice function elect an alternative when this notion is satisfied. We say that $a^* \in A$ is a *Condorcet winner* in $\vec{\succ}$ if for all $a \in A \setminus \{a^*\}$, $|\{i \in N :\ a^* \succ_i a\}| > n/2$, that is, a majority of agents prefer a^* to any other alternative. A social choice function f is *Condorcet-consistent* if $f(\vec{\succ}) = a^*$ whenever a^* is a Condorcet winner in $\vec{\succ}$; it is *unanimous* if for every $\vec{\succ} \in \mathcal{L}^n$ such that $\text{top}(\succ_i) = a^*$ for every $i \in N$, $f(\vec{\succ}) = a^*$, i.e., it always elects an alternative that is ranked first by all the agents. A related axiom is *Pareto optimality* that requires that for every $\vec{\succ} \in \mathcal{L}$ where $x \succ_i y$ for all $i \in N$, $f(\vec{\succ}) \neq y$. Note that Condorcet-consistency and Pareto optimality both imply unanimity, and each of the three axioms implies ontoness. Furthermore, Condorcet consistency implies nondictatorship, but a dictatorship is Pareto optimal (and in particular unanimous).

3 The Model

Let $N = \{1, \ldots, n\}$ be the set of agents and A be the set of alternatives; denote $|A| = m$. We presently describe the MDP $\mathcal{M} = (\mathcal{S}, A, R, P)$ that we deal with. Specifically, we will describe $\mathcal{S}$,

A, and P, and leave the restrictions on R for later. The state transitions in our MDP are factored across agents and thus the MDP is a very special case of a *factored MDP* [4]; however, since we do not require much of the factored MDP formalism, we will describe our MDP directly while borrowing some notions from the factored MDP literature.

The set of actions A of our MDP coincides with the set of alternatives (which is also denoted by A). For each agent i we have a random variable X_i, which takes values in $\mathcal{L}$. The current value of X_i indicates the current preferences of agent i. A state $s \in \mathcal{S}$ defines a value $\succ_i \in \mathcal{L}$ for every variable X_i, $i \in N$. Therefore, each state is a preference profile, and the size of the state space is huge: $(m!)^n$. Given a state $s \in \mathcal{S}$, we denote by $s(i)$ the value of X_i in this state, that is, the preferences of agent i. Furthermore, given a permutation $\mu : N \to N$, we also denote by $\mu(s)$ the state such that $s(i) = \mu(s)(\mu(i))$ for all $i \in N$, that is, μ can also be seen as a permutation over states.

A central conceptual observation is that, under the above definition of the state space, a deterministic policy π coincides with a social choice function. A policy defines a mapping from preference profiles to alternatives. We can therefore seek policies that satisfy the traditional social choice axioms; this is indeed the focus of our technical results.

For each $i \in N$ we have a transition model P_i, where $P_i(\succ'_i \mid \succ_i, a)$ is the probability of X_i taking the value $\succ'_i$ when the current value is $\succ_i \in \mathcal{L}$ and the action $a \in A$ is taken. This models the way in which an agent's preferences change based on its current preferences and the selected alternative. We define the transition model of the MDP $\mathcal{M}$ by letting

$$P(s'|s,a) = \prod_{i \in N} P_i(s'(i)|s(i),a). \tag{1}$$

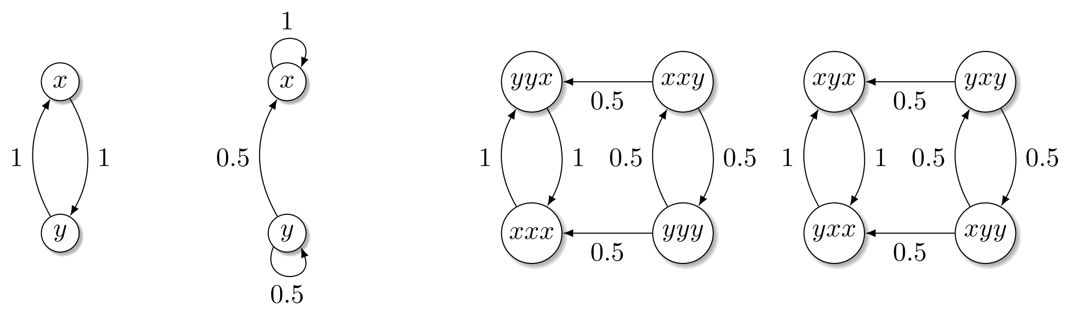
Intuitively, at every step we elect an alternative, and this choice affects the preferences of all the agents. Note that the preference ranking of each agent transitions independently given the selection of a particular alternative.

Definition 3.1. An MDP $\mathcal{M} = (\mathcal{S}, A, R, P)$ where $\mathcal{S}$, A, and P are as above, is called a *social choice MDP*.

In the sequel we assume that there are only t possible transition models, where $t \leq n$. We say that agents with the same transition model have the same *type*. Assuming a relatively small number of types will help us obtain efficient algorithms despite the huge state space. We let $\widehat{\Theta}$ be the partition of agents into types, where each $\theta \in \widehat{\Theta}$ is a set of agents with the same type; see Figure 1 for an illustration. In the following we will find it useful to construct partitions that are more refined, and sometimes coarser, than this basic type-based partition.

4 Symmetries and Local Axioms

Having defined the MDP that we are interested in, we ask whether it is possible to efficiently compute an optimal policy despite the large number of states. We will show that we can provide some positive answers by exploiting symmetries between states. Moreover, we will show that our arguments are sufficiently general to facilitate the computation of optimal policies that satisfy what we call *anonymous local axioms*. Some of the lemmata in this section are more general than what we need right now, but we will require their full power in subsequent sections.



(a) The two type transition models: type 1 (left) type 2 (right).

(b) The transition model of the MDP.

Figure 1: Types illustrated. In this example there are three agents and two alternatives, $A = \{x, y\}$. There are two types and $\widehat{\Theta} = \{\{1, 2\}, \{3\}\}$, that is, agents 1 and 2 are of type 1 and agent 3 is of type 2. The transition models of the two types are illustrated in (a). We only show the transitions that are associated with action x. A node is labeled with the top preference of an agent, e.g., a node labeled by x corresponds to the preference $x \succ y$. The transition model of the MDP is shown in (b), where again only the transitions that are associated with the action x are illustrated. A node is labeled by the top preference of the three agents, e.g., xyx corresponds to the preference profile where $x \succ_1 y, y \succ_2 x, x \succ_3 y$.

4.1 Exploiting symmetries

The symmetries in a social choice MDP stem from the identical transition models associated with agents of the same type. Intuitively, rather than concerning ourselves with which ranking is currently held by each agent, it should be enough to keep track of *how many agents of each type possess each ranking*. To make this precise we use a formalism introduced by Zinkevich and Balch [25]. Given an *equivalence relation* E over a set B, we denote by $E(x) = \{x' \in B : (x, x') \in E\}$ the *equivalence class* of $x \in B$, and by $\mathcal{E}$ the set of equivalence classes of E; in particular $E(x) \in \mathcal{E}$. $\mathcal{E}$ is a partition of set B.

Definition 4.1. Let $\mathcal{M} = (\mathcal{S}, A, R, P)$ be an MDP and let E be an equivalence relation over $\mathcal{S}$.

1. R is *symmetric with respect to* E if for all $(s, s') \in E$ and all $a \in A$, $R(s, a) = R(s', a)$.
2. P is *symmetric with respect to* E if for all $(s, s') \in E$, every $a \in A$, and every equivalence class $S \in \mathcal{E}$,
$$\sum_{s'' \in S} P(s''|s, a) = \sum_{s'' \in S} P(s''|s', a).$$
3. $\mathcal{M}$ is *symmetric with respect to* E if R and P are symmetric with respect to E.
4. A policy π is symmetric with respect to E if for all $(s, s') \in E$, $\pi(s) = \pi(s')$.

Intuitively, symmetry of an MDP with respect to an equivalence relation requires, for every action and every equivalence class on states, that both the reward and the probability of transitioning to any particular equivalence class, is independent of the exact state in the current equivalence class.

Zinkevich and Balch [25] show that if $\mathcal{M}$ is symmetric with respect to E then there is an optimal deterministic policy that is identical on the equivalence classes of $\mathcal{E}$, and thus symmetric with respect to E. This is useful because an optimal policy can be computed by contracting the state space of $\mathcal{M}$, replacing each equivalence class in $\mathcal{E}$ with one state. More formally, the following lemma is an adaptation of [25, Theorem 2].

Lemma 4.2 ([25])**.** *Let $\mathcal{M} = (\mathcal{S}, A, R, P)$ be an MDP and let E be an equivalence relation over $\mathcal{S}$. Assume that $\mathcal{M}$ is symmetric with respect to E. Then an optimal deterministic policy for $\mathcal{M}$ that is symmetric with respect to E can be computed in time that is polynomial in $|\mathcal{E}|$ and $|A|$.*

In order to employ Lemma 4.2 we must construct an equivalence relation for which the social choice MDP is symmetric. For this, we define a partition Θ on agents, that induces an equivalence relation E_Θ on states. As a special case, we will be interested in the partition $\widehat{\Theta}$ of agents into types but a formulation in terms of arbitrary partitions over the agents will prove useful in Section 5.

Given a partition Θ of agents, we define E_Θ as follows. For all $s, s' \in \mathcal{S}$, $(s, s') \in E_\Theta$ if and only if for all $\theta \in \Theta$ and $\succ \in \mathcal{L}$,

$$|\{i \in \theta : \; s(i) = \succ\}| = |\{i \in \theta : \; s'(i) = \succ\}|.$$

Informally, two states s and s' are equivalent given partition Θ on agents if one state can be obtained from the other by a permutation of preferences of agents in the same subset $\theta \in \Theta$. For example, if $\Theta = \{\{1\}, \{2\}, \{3\}\}$, and fully refined, then all states are distinct. If $\Theta = \{\{1, 2\}, \{3\}\}$, then any pair of states where agents 1 and 2 swap rankings are equivalent.

Note that for each $\theta \in \Theta$, each $\succ \in \mathcal{L}$, and any $s \in \mathcal{S}$, $|\{i \in \theta : \; s(i) = \succ\}| \in \{0, \ldots, n\}$. This immediately implies the following lemma.[2]

Lemma 4.3. *Let Θ be a partition of the agents such that $|\Theta| = k$. Then $|\mathcal{E}_\Theta| \leq (n+1)^{k(m!)}$.*

Thus, we see that when m is constant, the number of equivalence classes on states induced by a partition Θ of constant size k is polynomial in n.

Definition 4.4. Given a partition Θ of some (arbitrary) set, a partition Θ' is called a *refinement of* Θ if for every $\theta' \in \Theta'$ there exists $\theta \in \Theta$ such that $\theta' \subseteq \theta$.

We observe that refining the partition over the set of agents also refines the associated partition of states into equivalence classes. Formally:

Observation 4.5. *Let Θ be a partition of agents, and let Θ' be a refinement of Θ. Then $\mathcal{E}_{\Theta'}$ is a refinement of $\mathcal{E}_\Theta$, where these are the associated partitions on states.*

We wish to prove that a social choice MDP is symmetric with respect to equivalence classes induced by refinements of the partition of agents into types. The next lemma establishes this symmetry for the transition model.

Lemma 4.6. *Let $\mathcal{M} = (\mathcal{S}, A, R, P)$ be a social choice MDP, and let $\widehat{\Theta}$ be the partition of the agents into types. Let Θ be a refinement of $\widehat{\Theta}$. Then P is symmetric with respect to E_Θ.*

Proof. Let $s, s' \in \mathcal{S}$ such that $(s, s') \in E_\Theta$. Consider a permutation $\mu : N \to N$ such that for every $\theta \in \Theta$ and every $i \in N$, $\mu(i) \in \theta$, and $s' = \mu(s)$; μ is guaranteed to exist by the definition of E_Θ.

[2]The bound given in Lemma 4.3 is far from being tight, but it is sufficient for our purposes.

Since Θ is a refinement of $\widehat{\Theta}$, for every $\theta \in \Theta$ and every $i, j \in \theta$, $P_i \equiv P_j$, and in particular for every $i \in N$, $P_i \equiv P_{\mu(i)}$. It follows from (1) that

$$P(s''|s,a) = P(\mu(s'')|\mu(s),a) = P(\mu(s'')|s',a)$$

for all $s'' \in \mathcal{S}$. Therefore, for every $a \in A$ and every equivalence class $S \in \mathcal{E}_\Theta$,

$$\sum_{s'' \in S} P(s''|s,a) = \sum_{s'' \in S} P(\mu(s'')|s',a). \tag{2}$$

Next, recall that $\mu(i) \in \theta$ for each $\theta \in \Theta$ and $i \in \theta$, and hence for every $S \in \mathcal{E}_\Theta$ and $s'' \in S$ it holds that $(s'', \mu(s'')) \in E_\Theta$. We wish to claim that μ restricted to S is a permutation on S. It is sufficient to prove that μ is one-to-one; this is true since if $s''_1 \neq s''_2$ then there exists $i \in N$ such that $s''_1(i) \neq s''_2(i)$, and therefore $\mu(s''_1)(\mu(i)) \neq \mu(s''_2)(\mu(i))$. We conclude that

$$\sum_{s'' \in S} P(\mu(s'')|s',a) = \sum_{s'' \in S} P(s''|s',a). \tag{3}$$

The combination of (2) and (3) yields

$$\sum_{s'' \in S} P(s''|s,a) = \sum_{s'' \in S} P(s''|s',a),$$

as desired. $\square$

To provide intuition for Lemma 4.6, consider again the social choice MDP partially specified in Figure 1. In this example the equivalence relation induced by the partition on types is

$$\mathcal{E}_{\widehat{\Theta}} = \{\{xxx\}, \{xxy\}, \{xyx, yxx\}, \{xyy, yxy\}, \{yyx\}, \{yyy\}\}.$$

The lemma states that, e.g., considering a transition from a state in class 4 to a state in class 3,

$$P(xyx|xyy,a) + P(yxx|xyy,a) = P(xyx|yxy,a) + P(yxx|yxy,a).$$

Indeed, $P(xyx|xyy,a) + P(yxx|xyy,a) = 0 + 0.5 = 0.5$, and $P(xyx|yxy,a) + P(yxx|yxy,a) = 0.5 + 0 = 0.5$.

So far we have imposed no restrictions on the reward function R. Below we impose a strong restriction on R: we will require it to be anonymous, that is, it should be symmetric with respect to the equivalence relation $E_{\{N\}}$ induced by the coarsest partition of the agents, $\{N\}$. Equivalently, a reward function R is anonymous if for every permutation $\mu : N \to N$ on the agents, all $s \in \mathcal{S}$ and all $a \in A$, $R(s,a) = R(\mu(s),a)$. This is a stronger restriction than what we need technically, but we argue below that natural reward functions satisfy this property.

Definition 4.7. Let $\mathcal{M} = (\mathcal{S}, A, R, P)$ be a social choice MDP. A reward function $R : \mathcal{S} \times A \to \mathbb{R}$ is *anonymous* if it is symmetric with respect to $E_{\{N\}}$.

As a first example of an anonymous reward function, consider a designer who has a favorite alternative $a^* \in A$, and for every state $s \in \mathcal{S}$ we have

$$R(s,a) = \begin{cases} 1 & \text{if } a = a^* \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

In this case the reward function is completely indifferent to the state, and in particular anonymous. Clearly the optimal (unconstrained) policy would always choose the favorite alternative a^*, but this may be impossible when the policy is constrained via axioms, as we shall see later.

As a second natural example, consider a designer who wishes to reach a social consensus. As described in Section 2.2, three common notions of consensus are a *Condorcet winner*, a *majority winner* which is ranked first by a majority of agents, and (the stronger notion of) a *unanimous winner*, that is, an alternative ranked first by all agents. The reward function can then be, e.g., of the form

$$R(s,a) = \begin{cases} 1 & \text{if } a \text{ is a Condorcet winner in } s \\ 0 & \text{otherwise} \end{cases}$$

The abovementioned notions of social consensus are indifferent to permuting the agents and hence lead to anonymous reward functions.

The following observation is a direct corollary of Observation 4.5 and the fact that any partition on agents is a refinement of $\{N\}$.

Observation 4.8. *Let $\mathcal{M} = (\mathcal{S}, A, R, P)$ be a social choice MDP. Let Θ be any agent partition, and let $R : \mathcal{S} \times A \to \mathbb{R}$ be an anonymous reward function. Then R is symmetric with respect to E_Θ.*

To summarize, a social choice MDP with an anonymous reward function is symmetric with respect to refinements of $E_{\widehat{\Theta}}$ (by Lemma 4.6 and Observation 4.8), and this suggests a method for computing an optimal deterministic policy that is polynomial in n by Lemma 4.2.

We want to prove a more general result though, which will also enable the restriction of the actions available in some states. The purpose of handling restrictions is twofold. First, it will directly facilitate the computation of optimal policies that are consistent with certain local axioms. Second, it will prove crucial for the algorithms of Section 5 that handle non-local axioms.

Definition 4.9. Let $\mathcal{M} = (\mathcal{S}, A, R, P)$ be a social choice MDP.

1. A *restriction* is a subset $\Psi \subseteq \mathcal{S} \times A$. A deterministic policy π is Ψ-consistent if for every $s \in \mathcal{S}$, $\pi(s) \in \{a \in A : (s,a) \in \Psi\}$.

2. Let Θ be a partition of the agents. A restriction Ψ is symmetric with respect to induced equivalence relation E_Θ on states if for all $(s,s') \in E_\Theta$, and all $a \in A$, $(s,a) \in \Psi \Leftrightarrow (s',a) \in \Psi$.

That is, a restriction to particular actions in particular states is symmetric with respect to an equivalence relation if the same restriction holds across all equivalent states.

In some of the MDP literature a restriction is known as the set of admissible state-action pairs, and is considered to be a component of the MDP. For our purposes it is convenient to view the MDPs and restrictions separately, as we see the restriction as an external constraint that is imposed on the MDP.

We are finally ready to present this section's main result, which is technically simple despite the unavoidably elaborate formalism. A short discussion is in order, though, regarding the parameters of the problem. As mentioned above, the size of the state space of a social choice MDP is $(m!)^n$. Even if m is constant, this is still exponential in n. In order to obtain running time that is tractable with respect to the number of agents n we also assume that the number of types is constant. These restrictions, which are discussed in Section 1, underly all our algorithmic results, including the more involved results of Section 5.

Theorem 4.10. *Let $\mathcal{M} = (\mathcal{S}, A, R, P)$ be a social choice MDP, let R be an anonymous reward function, let $\widehat{\Theta}$ be the partition of the agents into types, let Θ be a refinement of $\widehat{\Theta}$, and let $\Psi \subseteq \mathcal{S} \times A$ be symmetric with respect to E_Θ. Furthermore, assume that $|A| = \mathcal{O}(1)$ and $|\Theta| = \mathcal{O}(1)$. Then an optimal deterministic Ψ-consistent policy for $\mathcal{M}$ (which will be symmetric with respect to E_Θ) can be computed in polynomial time in the number of agents n.*

Proof. We define $R' : \mathcal{S} \times A \to \mathbb{R}$ such that[3]

$$R'(s,a) = \begin{cases} R(s,a) & \text{if } \ (s,a) \in \Psi \\ -\infty & \text{otherwise} \end{cases}$$

From the facts that R is symmetric with respect to E_Θ (by Observation 4.8) and Ψ is symmetric with respect to E_Θ (by assumption), it follows that R' is also symmetric with respect to E_Θ. By Lemma 4.6, P is symmetric with respect to E_Θ. Let $\mathcal{M}' = (\mathcal{S}, A, R', P)$; we have that $\mathcal{M}'$ is symmetric with respect to E_Θ. By Lemma 4.2 we can find an optimal policy for $\mathcal{M}'$ in time polynomial in $|A| = \mathcal{O}(1)$ and $|\mathcal{E}_\Theta|$. By Lemma 4.3, using $|A| = \mathcal{O}(1)$ and $|\Theta| = \mathcal{O}(1)$, we have that $|\mathcal{E}_\Theta| \leq (n+1)^{\mathcal{O}(1)}$, therefore an optimal deterministic policy for $\mathcal{M}'$ can be computed in polynomial time. The definition of R' simply rules out the use of state-action pairs that are not in Ψ, hence this policy is an optimal deterministic Ψ-consistent policy for $\mathcal{M}$. $\square$

4.2 Anonymous local axioms

Theorem 4.10 provides us with the means to compute optimal policies that are consistent with restrictions that respect symmetries. Fortunately, many prominent social choice axioms can be expressed with just these kinds of restrictions, hence we can settle the question of computing optimal deterministic policies that satisfy such axioms without further ado.

Definition 4.11. An axiom is *local* if it can be represented as a restriction $\Psi \subseteq \mathcal{S} \times A$. A local axiom is *anonymous* if it is symmetric with respect to $E_{\{N\}}$.

Informally, a local axiom prescribes which actions are allowed in each state; it is local in the sense that this does not depend on the policy's choices in other states. Symmetry requires that the acceptability, or not, of an action does not depend on the names of agents. For example, Condorcet-consistency is a local axiom since it simply restricts the set of actions (to a singleton) in every state where a Condorcet winner exists. Condorcet-consistency is also anonymous: if an alternative is a Condorcet winner, it would remain one for any permutation of preferences amongst agents. More generally, we have the following observation.

Observation 4.12. *The following axioms are local and anonymous: Condorcet consistency, Pareto-optimality, and unanimity.*[4]

Now the following result is an immediate corollary of Theorem 4.10.

Corollary 4.13. *Let $\mathcal{M} = (\mathcal{S}, A, R, P)$ be a social choice MDP, let R be an anonymous reward function, let $\widehat{\Theta}$ be the partition of the agents into types, and let Ψ be the restriction that represents an anonymous local axiom. Furthermore, assume that $|A| = \mathcal{O}(1)$ and $|\widehat{\Theta}| = t = \mathcal{O}(1)$. Then an optimal deterministic Ψ-consistent policy for $\mathcal{M}$ can be computed in polynomial time in the number of agents n.*

[3]It is possible to replace $-\infty$ with a sufficiently negative constant, e.g., $-(1/(1-\gamma)) \max_{s,a} R(s,a)$.

[4]There are additional axioms that are local and anonymous, e.g., Smith consistency, mutual majority consistency, and invariant loss consistency; see [24] for a description of these axioms.

5 Nonlocal Axioms

The axioms that we were able to handle in Section 4 are local, in the sense that the restrictions on the available actions in a state are not conditional on the actions taken in other states. However, many important axioms do not possess this property. Ontoness, for example, restricts the action in a given state to be a only if a is not selected in any other state. Monotonicity restricts the action in a state to be a if there is another state where a is as preferred or less preferred by every agent relative to every other action (and the relative preference for all other actions is unchanged) and a is selected by the policy in the latter state.

In this section we present algorithms that compute an optimal deterministic policy under the restriction that they are either onto or nondictatorial. These nonlocal axioms stand out in that they admit a tractable guided search approach. For example, there are n dictatorial polices that must be ruled out in obtaining a nondictatorial policy. The technical difficulty, though, will be to exclude policies through search without causing an exponential growth in the number of equivalence classes.

It is important to note that when dealing with nonlocal axioms an optimal deterministic policy is not necessarily *uniformly optimal*, that is, it need not be simultaneously optimal on every state. This is not surprising, arising from the loss of the principle of optimality and the Markovian property when solving constrained MDPs and known from the constrained MDP literature [1].

To see this, consider a social choice MDP where the agents have only one type, and there is a uniform probability of transitioning from any state to any state under every action. Furthermore, assume that the reward function is the one given in (4), i.e., that of a designer with a favorite alternative a^*. Let π_1 be the policy that selects a^* in every state except s_1, and π_2 be the policy that selects a^* in every state except s_2. Given, e.g, a uniform distribution over states, clearly both π_1 and π_2 maximize $\sum_{s\in\mathcal{S}}(1/|\mathcal{S}|)V_\pi(s)$, but $V_{\pi_1}(s_1) < V_{\pi_2}(s_1)$ and $V_{\pi_2}(s_2) < V_{\pi_1}(s_2)$.

Hence, in the following we are interested in policies that are optimal with respect to a given state $\hat{s} \in \mathcal{S}$, that is, we wish to find a deterministic policy π that maximizes $V_\pi(\hat{s})$. Equivalently, we are interested in a situation where there is an initial distribution on states that is degenerate at one point.[5] We will assume that, given the initial partition into types $\widehat{\Theta}$, it holds that

$$E_{\widehat{\Theta}}(\hat{s}) = \{\hat{s}\}. \tag{5}$$

This assumption is without loss of generality as otherwise we can use the techniques presented below to refine $\widehat{\Theta}$ in a way that the equality holds.

In the sequel we will find it helpful to define a procedure $\textsc{Solve}(\mathcal{M}, \hat{s}, \Theta, \Psi)$, which receives as input a dynamic social choice *MDP* $\mathcal{M}$ with an anonymous reward function R, an initial state $\hat{s}$, a partition Θ of the agents that is a refinement of the partition $\widehat{\Theta}$ into types, and a restriction $\Psi \subseteq \mathcal{S} \times A$ that is symmetric with respect to E_Θ. The procedure finds an optimal deterministic Ψ-consistent policy π^* (which is symmetric with respect to E_Θ) using Theorem 4.10. Note that π^* maximizes the value on every state $s \in \mathcal{S}$ among all Ψ-consistent deterministic policies. The procedure then returns $V_{\pi^*}(\hat{s})$, the value of π^* on the initial state $\hat{s}$.

Our most useful tool in this section is a procedure for "carving out states". Specifically, we need to be able to restrict the feasible actions in a *specific*, given state. The problem is that restricting a specific state breaks the symmetries and therefore precludes solution via abstraction

[5] More general distributions lead to complications both in terms of their representation and in terms of maintaining symmetries.

Algorithm 1

```
1: procedure CARVE-OUT($\mathcal{M}, s, \Theta$)
2:     $\Theta' \leftarrow \emptyset$
3:     for each $\theta \in \Theta$ and each $\succ \in \mathcal{L}$ do
4:         $\theta' \leftarrow \{i \in \theta : \ s(i) = \succ\}$
5:         If $\theta' \neq \emptyset$ then $\Theta' \leftarrow \Theta' \cup \{\theta'\}$
6:     Return $\Theta'$
```

of the problem to the current equivalence classes. Put another way, if π is symmetric with respect to E, and we define Ψ' by removing some pairs (s, a) for a specific $s \in S$ where $|E(s)| > 1$, then Ψ' is not symmetric with respect to E. Nevertheless, Ψ' may be symmetric with respect to a refined partition. Indeed, we will recreate the symmetries by refining the equivalence relation in an appropriate way.

Consider the procedure Carve-Out, given as Algorithm 1. This procedure receives as input a social choice MDP $\mathcal{M}$, a state $s \in \mathcal{S}$ to be separated, and the current partition of the agents Θ. It refines Θ by partitioning each subset $\theta \in \Theta$ according to the rankings that the agents in θ hold in s, grouping agents with identical rankings together; see Figure 2 for an example.

The key observation is that Carve-Out refines the partition Θ of agents in a way that the given state s becomes a singleton equivalence class in the associated equivalence relation on states. We see this in Figure 2, depicting state s. No other state s' is equivalent to s under the equivalence class induced by the new partition on agents; any change in preference from s by any number of agents must lead to a different number of agents with the current preferences in at least one of the refined subsets of agents.

Lemma 5.1. *Let $\mathcal{M} = (\mathcal{S}, A, R, P)$ be a social choice MDP, let $s \in \mathcal{S}$, let Θ be a partition of the agents, and let $\Theta' =$ Carve-Out$(\mathcal{M}, s, \Theta)$. Then $E_{\Theta'}(s) = \{s\}$.*

Proof. Let $s' \in \mathcal{S} \backslash \{s\}$, so that there is an agent $i \in N$ with $s(i) \neq s'(i)$. Let $\theta' \in \Theta'$ such that $i \in \theta'$, and let $s(i) = \succ$. On one hand, it holds that for all $j \in \theta'$, $s(j) = \succ$, hence $|\{j \in \theta' : \ s(j) = \succ\}| = |\theta'|$. On the other hand, there exists $i \in \theta'$ such that $s'(i) \neq \succ$, i.e., $|\{j \in \theta' : \ s'(j) = \succ\}| < |\theta'|$. It follows that $(s, s') \notin E_{\Theta'}$. $\square$

Using Lemma 5.1 and the fact that Carve-Out refines the partition over agents and hence (by Observation 4.5) the partition over states, it is now obvious that Carve-out successfully recreates the symmetries; we make this statement formal in the following lemma.

1	2	3	4	5	6	7	8	9	10
x	x	y	y	z	z	z	z	x	z
y	y	z	z	y	y	y	x	y	y
z	z	x	x	x	x	x	y	z	x

Figure 2: Carve-Out illustrated. The preferences of the 10 agents in state s is shown, and an initial partition Θ is marked by thick lines, i.e., $\Theta = \{\{1, 2, 3, 4\}, \{5, 6, 7, 8\}, \{9, 10\}\}$. Letting $\Theta' =$ Carve-Out$(\mathcal{M}, s, \Theta)$, we have $\Theta' = \{\{1, 2\}, \{3, 4\}, \{5, 6, 7\}, \{8\}, \{9\}, \{10\}\}$. By this, no other state $s' \neq s$ can be equivalent to s.

Lemma 5.2. *Let $\mathcal{M} = (\mathcal{S}, A, R, P)$ be a social choice MDP, let $s \in \mathcal{S}$, let Θ be a partition of the agents, let $\Theta' = \text{Carve-Out}(\mathcal{M}, s, \Theta)$, let $\Psi \subseteq \mathcal{S} \times A$ be symmetric with respect to E_Θ, and let $\Psi' \subseteq \mathcal{S} \times A$ differ from Ψ only in the actions allowed in state s. Then Ψ' is symmetric with respect to $E_{\Theta'}$.*

We will require one last property of Carve-Out. This is by construction, since the procedure keeps agents with identical rankings grouped together.

Observation 5.3. *Let $\mathcal{M} = (\mathcal{S}, A, R, P)$ be a social choice MDP, let $s \in \mathcal{S}$, let Θ be a partition of the agents, and let $\Theta' = \text{Carve-Out}(\mathcal{M}, s, \Theta)$. Then $|\Theta'| \leq m!|\Theta|$.*

We continue by using this approach as a subroutine within a search algorithm for identifying optimal deterministic policies that are onto, and then nondictatorial.

5.1 Ontoness

We seek an algorithm for computing an optimal deterministic *onto* policy in polynomial time in n. Observe that an onto policy π must have at least one state s for each action a such that $\pi(s) = a$.

Let $A = \{a_1, \ldots, a_m\}$. A naïve algorithm would simply check all the possibilities of selecting a sequence of m states $\vec{S} = (s_1, \ldots, s_m)$, restricting each state s_k to a_k, for $k \in \{1, \ldots, m\}$, and return the best deterministic policy. There are two difficulties with this approach. First, restricting a specific state breaks the symmetries, but we have already introduced a method that circumvents this difficulty, namely the Carve-Out procedure. Second, since the state space is large we cannot check all possibilities of selecting m states; here again the symmetries come to the rescue.

Our solution is simple: when selecting the first state s_1 (to restrict to a_1), it is sufficient to consider one state per equivalence class in $E_{\widehat{\Theta}}$, and repartition the agents using Carve-Out. Given such a choice, then when selecting the second state s_2 (to restrict to a_2), we consider one state per equivalence class induced by the new partition of the agents, and repartition again; we continue selecting and repartitioning until we have selected m states. This leads to an idea for a search algorithm.

Our algorithm is formally given as Algorithm 2. The algorithm receives as input a social choice MDP $\mathcal{M}$, an initial state $\hat{s} \in \mathcal{S}$, and a partition $\widehat{\Theta}$ of the agents into types. The procedure Search recursively tries all possibilities of separating a state, one per each equivalence class, and for every possibility refines the partition of the agents using Carve-Out as described above. Search is initialized with ϵ, denoting an empty sequence of restricted states. For every sequence $\vec{S}$ of exactly m states selected by Search, the algorithm then constructs a restriction Ψ where each selected state is restricted to a different action; this step guarantees that any Ψ-consistent policy is onto. The algorithm then uses Solve to determine the value on $\hat{s}$ of the optimal deterministic Ψ-consistent policy. Note that it is easy to modify the algorithm so that the optimal policy itself is returned as well.

As we will show, a bound on the running time of the algorithm follows directly from our previous results. The main obstacle is establishing that the restricted search executed by the algorithm is indeed exhaustive.

Theorem 5.4. *Let $\mathcal{M} = (\mathcal{S}, A, R, P)$ be a social choice MDP, let R be an anonymous reward function, let $\widehat{\Theta}$ be the partition of the agents into types, and let $\hat{s} \in \mathcal{S}$. Furthermore, assume that $m = |A| = \mathcal{O}(1)$ and $|\widehat{\Theta}| = \mathcal{O}(1)$. Then:*

Algorithm 2 The social choice MDP $\mathcal{M}$ and the initial state $\hat{s}$ are treated as global variables.

```
1: procedure COMPUTE-ONTO($\mathcal{M} = (\mathcal{S}, A, R, P), \hat{s}, \widehat{\Theta}$)
2:     return SEARCH($\widehat{\Theta}, \epsilon$)

1: procedure SEARCH($\Theta, \vec{S}$)
2:     $k \leftarrow |\vec{S}| + 1$
3:     $M \leftarrow -\infty$
4:     for each $T \in \mathcal{E}_\Theta$ do
5:         select $s_k \in T \setminus \vec{S}$ arbitrarily (if exists)
6:         $\Theta_k \leftarrow$ CARVE-OUT($\mathcal{M}, s_k, \Theta$)
7:         if $k = m$ then
8:             Let $\Psi \subseteq \mathcal{S} \times A$ that restricts each of $\{s_1, \ldots, s_m\}$ to $\{a_1, \ldots, a_m\}$
9:             $M \leftarrow \max\{M,$ SOLVE$(\mathcal{M}, \hat{s}, \Theta_k, \Psi)\}$
10:        else
11:            $M \leftarrow \max\{M,$ SEARCH$(\Theta_k, \vec{S}.s_k)\}$
12:    return $M$
```

1. Compute-Onto($\mathcal{M}, \hat{s}, \widehat{\Theta}$) *can be implemented in polynomial time in n.*

2. Compute-Onto($\mathcal{M}, \hat{s}, \widehat{\Theta}$) *returns the value on $\hat{s}$ of the optimal deterministic onto policy.*

Proof. For part 1, we first claim that in each call to Solve($\mathcal{M}, \hat{s}, \Theta, \Psi$), the restriction Ψ that Search constructs in line 8 is symmetric with respect to E_Θ. Indeed, let $\mathcal{S} \times A = \Psi_0, \Psi_1, \ldots, \Psi_m = \Psi$ where each Ψ_k differs from Ψ_{k-1} only in that it restricts s_k to a_k. In addition, let $\widehat{\Theta} = \Theta_0, \Theta_1, \ldots, \Theta_m = \Theta$, where $\Theta_k =$ Carve-Out($\mathcal{M}, s_k, \Theta_{k-1}$). By inductively applying Lemma 5.2 we get that each Ψ_k is symmetric with respect to E_{Θ_k}, and in particular Ψ is symmetric with respect to E_Θ.

Now, it follows from Observation 5.3 that $|\Theta| \leq (m!)^m \cdot |\widehat{\Theta}| = \mathcal{O}(1)$, and hence by Lemma 4.3 $|\mathcal{E}_\Theta|$ is polynomial in n. Using Theorem 4.10 we conclude that Solve can be implemented in polynomial time.

It remains to bound the number of sequences $\vec{S}$ of restricted states that are searched by the algorithm, that is, the number of times Solve is executed. Consider the for loop in line 4 of Search. We have $|\mathcal{E}_{\Theta_{k-1}}|$ possibilities of choosing $T \in \mathcal{E}_{\Theta_{k-1}}$, from which s_k will be selected. As noted above, for all $k \in \{0, \ldots, m\}$ it holds that $|\Theta_k| = \mathcal{O}(1)$, and hence $|\mathcal{E}_{\Theta_k}|$ is polynomial in n for all $k \in \{0, \ldots, m\}$. Therefore, the total number of possible sequences is a product of a constant number of polynomials in n, which is polynomial in n.

We next turn to part 2 of the theorem. As mentioned above, the optimal deterministic onto policy π^* is such that for each $a_k \in A$ there is some $s_k \in \mathcal{S}$ such that $\pi^*(s_k) = a_k$. If Solve were called with the restriction induced by this sequence as input then it would return the optimal solution. Hence, it is sufficient to show that for every possible sequence of states there is an equivalent (under symmetry) sequence of states that is actually searched by Search.

More formally, let $\vec{S} = (s_1, \ldots, s_m)$ be a sequence of states. As before, consider the sequence of partitions $(\Theta_0, \Theta_1, \ldots, \Theta_m)$ where $\Theta_k =$ Carve-Out($\mathcal{M}, s_k, \Theta_{k-1}$).

Claim 5.5. *Let $\vec{S} = (s_1, \ldots, s_m)$ denote a sequence of states. There exists an equivalent sequence $\vec{S}' = (s'_1, \ldots, s'_m)$ that induces a sequence of partitions $(\Theta'_1, \ldots, \Theta'_m)$ where the following properties* *hold for each $k \in \{0, \ldots, m\}$:*

1. SEARCH *generates the sequence $(s'_1, \ldots, s'_k)$ (where for $k = 0$ we have an empty sequence).*

2. *There exists a permutation $\mu_k : N \to N$ such that:*

   (a) *For every $\theta \in \widehat{\Theta}$, $i \in \theta \Leftrightarrow \mu_k(i) \in \theta$ (i.e., the permutation respects types).*

   (b) *For each $l \in \{1, \ldots, k\}$, $s'_l = \mu_k(s_l)$ (i.e., the states are equivalent under permutation).*

Proof. We prove the claim by induction on k. For $k = 0$ the claim is trivial (by taking μ to be identity). Assume correctness for k, we will prove the claim for $k+1$. Consider $E_{\Theta'_k}(\mu_k(s_{k+1}))$. Let s'_{k+1} be the state that SEARCH$(\Theta'_k, (s'_1, \ldots, s'_k))$ selects from this class; then $(s'_1, \ldots, s'_{k+1})$ satisfies property 1. Let $\Theta'_{k+1} = $ CARVE-OUT$(\mathcal{M}, s'_{k+1}, \Theta'_k)$.

Next, we define a permutation μ'_k such that for all $\theta' \in \Theta'_k$ and $i \in \theta'$, $\mu'_k(i) \in \theta'$, and moreover $\mu'_k(\mu_k(s_{k+1})) = s'_{k+1}$. There exists such a permutation since $(\mu_k(s_{k+1}), s'_{k+1}) \in E_{\Theta'_k}$. Let $\mu_{k+1} = \mu'_k \circ \mu_k$, that is, $\mu_{k+1}(i) = \mu'_k(\mu_k(i))$.

We first claim that μ_{k+1} satisfies property 2(a). Let $i \in N$ such that $i \in \theta$ for some $\theta \in \widehat{\Theta}$. By property 2(a) of the induction assumption $\mu_k(i) \in \theta$, and since Θ'_k is a refinement of $\widehat{\Theta}$ this means that $\mu_k(i) \in \theta'$ where $\theta' \in \Theta'_k$ and $\theta' \subseteq \theta$. By the construction of μ'_k we have that $\mu'_k(\mu_k(i)) \in \theta'$, and hence $\mu_{k+1}(i) \in \theta$.

For property 2(b), it holds that $s'_{k+1} = \mu_{k+1}(s_{k+1})$ by definition, but we still have to show that $s'_l = \mu_{k+1}(s_l)$ for all $l \in \{1, \ldots, k\}$. Fix l. By property 2(b) of the induction assumption, $\mu_k(s_l) = s'_l$, and therefore

$$\mu_{k+1}(s_l) = \mu'_k \circ \mu_k(s_l) = \mu'_k(s'_l).$$

Left to show is $\mu'_k(s'_l) = s'_l$. Let $i \in N$ such that $i \in \theta'$, $\theta' \in \Theta'_l$. There are $\succ \in \mathcal{L}$ and $\theta'' \in \Theta'_{l-1}$ such that

$$\theta' = \{j \in \theta'' : \ s'_l(j) = \succ\},$$

that is, all the agents in θ' have the same preferences in s'_l.

Using the fact that Θ'_k is a refinement of Θ'_l and the construction of μ'_k we have that $\mu'_k(i) \in \theta'$. Therefore, $s'_l(\mu'_k(i)) = \succ = s'_l(i)$. Since this holds for every $i \in N$ we conclude that $\mu'_k(s'_l) = s'_l$. □

Using Claim 5.5 with $k = m$ we conclude that, for any $\vec{S} = (s_1, \ldots, s_m)$, there exists a sequence of states $\vec{S}' = (s'_1, \ldots, s'_m)$ that is searched by the algorithm, and a permutation $\mu : N \to N$ that preserves the partition $\widehat{\Theta}$, such that the induced permutation on states maps each s_l to s'_l, for all $l \in \{1, \ldots, m\}$.

Since μ simply shuffles agents of the same type it clearly holds that for all $s, s' \in \mathcal{S}$ and $a \in A$,

$$P(s'|s, a) = P(\mu(s')|\mu(s), a).$$

Furthermore, the assumption that R is anonymous implies that for all $s \in \mathcal{S}$ and $a \in A$, $R(s, a) = R(\mu(s), a)$. Finally, let Ψ be the restriction induced by $\vec{S}$ in line 8 of SEARCH, and let Ψ' be the restriction induced by $\vec{S}'$. Using $s'_k = \mu(s_k)$ for all $k = 1, \ldots, m$, it holds that $(s, a) \in \Psi$ if and only if $(\mu(s), a) \in \Psi'$.

The arguments above imply that μ induces an *isomorphism* between $(\mathcal{M}, \Psi)$ and $(\mathcal{M}, \Psi')$ [19, 20]; in other words, the permutation simply relabels the states. Let π^*_1 be an optimal Ψ-consistent policy for $\mathcal{M}$ and let π^*_2 be an optimal Ψ'-consistent policy for $\mathcal{M}$; then for all $s \in \mathcal{S}$, $V_{\pi^*_1}(s) = V_{\pi^*_2}(\mu(s))$.

We are now ready to argue that

$$\textsc{solve}(\mathcal{M}, \hat{s}, \Theta_m, \Psi) = \textsc{solve}(\mathcal{M}, \hat{s}, \Theta'_m, \Psi'), \tag{6}$$

where Θ_m is the refined agent partition given $\vec{S}$ and Θ'_m is the refined agent partition given $\vec{S}'$.

The only remaining issue is that both sides of (6) use the same $\hat{s}$ as input. However, from (5) we have that $E_{\widehat{\Theta}}(\hat{s}) = \{\hat{s}\}$, that is, for $\theta \in \widehat{\Theta}$ and $i, j \in \theta$, $\hat{s}(i) = \hat{s}(j)$. Recall that for $i \in \theta$ it holds that $\mu(i) \in \theta$, hence $\hat{s}(i) = \hat{s}(\mu(i))$ for all $i \in N$, and in particular $\hat{s} = \mu(\hat{s})$, that is,

$$\textsc{solve}(\mathcal{M}, \hat{s}, \Theta_m, \Psi) = V_{\pi_1^*}(\hat{s}) = V_{\pi_2^*}(\mu(\hat{s})) = V_{\pi_2^*}(\hat{s}) = \textsc{solve}(\mathcal{M}, \hat{s}, \Theta'_m, \Psi').$$

□

5.2 Nondictatorship

Our next goal is the computation of an optimal deterministic nondictatorial policy. Such a policy π must have the following property: for each $i \in N$ there is $s \in \mathcal{S}$ such that $\pi(s) \neq \text{top}(s(i))$.

This suggests a brute force approach similar to the one we used for ontoness: try all possibilities (up to symmetries) of choosing n states $s_1, \ldots, s_n$, and for each $i = 1, \ldots, n$ try all possibilities of restricting the policy to an action $a \in A \setminus \{\text{top}(s_i(i))\}$. An obvious difficulty with this suggestion is that even a single path in the search tree requires n calls to Carve-Out, and each such call increases the size of the partition of the agents. Ultimately the partition over the agents may become so fine that each agent is a singleton subset, and consequently the number of equivalence classes is too large.

Observe, though, that some restrictions rule out many dictators simultaneously. For example, if we restrict state $s \in \mathcal{S}$ to action $x \in A$ and there are $i, j \in N$ such that $y = \text{top}(s(i)) = \text{top}(s(j))$ and $y \neq x$ then both i and j are ruled out as dictators. Unfortunately, the search procedure described in the previous paragraph may lead us through a branch where each restriction rules out only one dictator, hence in general this observation is not sufficient.

A first insight is that it is possible to guide the search process rather than be guided by it. The idea is simple: the optimal deterministic nondictatorial policy can be computed as the best among the set of optimal deterministic nondictatorial policies that is obtained by restricting any particular state $s \in \mathcal{S}$ to each possible action $a \in A$ in turn. This suggests that instead of only considering restrictions that rule out a specific dictator, we might simply search all possible restrictions, but choosing the states to be restricted carefully.

Consider a state where $n/2$ agents rank $x \in A$ first and $n/2$ agents rank $y \in A$ first; by restricting this state to x we rule out the second subset of agents as dictators, by restricting it to y we rule out the first subset, and by restricting it to any other action we rule out dictators completely. By iteratively employing this idea we can rule out all the dictators in $\mathcal{O}(\log n)$ steps, but this is still too much, as in each step the size of the partition of the agents can increase by a constant factor.

In more detail, each step requires a call to Carve-Out, which in turn may increase the size of the partition of agents by a factor as large as $m!$. Even if the factor is only two, after $\log n$ executions the partition of agents would be of size n, i.e., each agent would be a singleton. This means that the number of induced equivalence classes would no longer be constant, and hence Theorem 4.10 would not guarantee a solution in polynomial time in n.

Our second insight circumvents this obstacle: instead of naïvely ruling out subsets of agents we rule out subsets $\theta \in \widehat{\Theta}$ in the initial type partition. After at most $|\widehat{\Theta}| - 1$ steps we will have at most one subset $\theta \in \widehat{\Theta}$ which can contain a dictator, but this subset may be very large.

At this point we can employ a third and final insight, formally given as the following lemma.

Lemma 5.6. *Let $\mathcal{M} = (\mathcal{S}, A, R, P)$ be a social choice MDP, let Θ be a partition of the agents, and let π be a policy that is symmetric with respect to E_Θ. If π is dictatorial with respect to agent $i \in \theta$, where $\theta \in \Theta$, then $|\theta| = 1$.*

Proof. Let π be a dictatorship of $i \in \theta$, and assume for contradiction that $|\theta| > 1$, hence there exists $j \in \theta \setminus \{i\}$. Next, let $s \in \mathcal{S}$ such that $\text{top}(s(i)) \neq \text{top}(s(j))$. Define $s' \in \mathcal{S}$ by flipping the votes of i and j, that is, $s'(i) = s(j)$, $s'(j) = s(i)$, and $s'(k) = s(k)$ for every $k \in N \setminus \{i, j\}$. It holds that $(s, s') \in E_\Theta$, and since π is symmetric with respect to E_Θ it holds that $\pi(s) = \pi(s')$. However, i is a dictator and therefore $\pi(s) = \text{top}(s(i))$ and $\pi(s') = \text{top}(s'(i)) = \text{top}(s(j)) \neq \text{top}(s(i)) = \pi(s)$, and a contradiction with symmetry. □

Lemma 5.6 implies that, conveniently, if we narrowed the possible dictators down to one subset $\theta \in \widehat{\Theta}$ and it contained at least two agents then we could compute an optimal policy that is symmetric with respect to $E_{\widehat{\Theta}}$, and this policy would not be dictatorial. In fact, if the subsets of agents in the type partition $\widehat{\Theta}$ are all size two or larger then an optimal (symmetric) deterministic policy would already be nondictatorial. By a carefully structured search algorithm, we compute the optimal nondictatorial policy in general problems.

Returning to the search approach, and considering the other case in which the possible dictators have been narrowed down to $\theta = \{i\}$, then it only remains to rule out i as a dictator, and we can do this using a brute force approach, by checking all possibilities (up to symmetries) of restricting a state to an alternative that is not most preferred by i.

The algorithm is formally given as Algorithm 3. Notice that $\mathcal{X}$ used in SEARCH is not a partition of the agents, rather it contains all the subsets of Θ that were not yet ruled out as dictators, and is initialized to $\widehat{\Theta}$. Lines 2–12 of SEARCH handle the case where there is only one subset that was not ruled out. If this subset is a singleton then we rule out the dictator using the brute force approach described above, where as if it is not (line 11) then Lemma 5.6 guarantees that SOLVE$(\mathcal{M}, \hat{s}, \widehat{\Theta}, \Psi)$ would return an optimal, nondictatorial policy.

Otherwise, an arbitrary θ in the remaining subsets of Θ is selected and a new state restriction is branched on that will preclude (on the basis of a restriction to this state) all of θ, all of $N \setminus \theta$, or every agent as a dictator. This is achieved in lines 13–20 of SEARCH. State s is selected so that restricting s to y rules out the agents in θ from being a dictator, restricting to x rules out all the agents except those in θ from being a dictator, and restricting to any action other than x, y rules out all dictatorial policies.[6] The following lemma highlights a crucial property of the choice of s.

Lemma 5.7. *Let $\mathcal{M} = (\mathcal{S}, A, P, R)$ be a social choice MDP, let $\widehat{\Theta}$ be the partition of the agents into types, let $\mathcal{X}$ be a set of subsets of N, let Ψ be symmetric with respect to $E_{\widehat{\Theta}}$, and let $s \in \mathcal{S}$ as constructed in line 16 of* SEARCH$(\mathcal{X}, \Psi)$ *and $\Psi_x, \Psi_y, \Psi_{-xy}$ as constructed in lines 17–19 of the procedure. Then $\Psi_x, \Psi_y, \Psi_{-xy}$ are symmetric with respect to $E_{\widehat{\Theta}}$.*

[6]It possible to modify the procedure in a way that rules out up to an $(m-1)/m$-fraction of the remaining subsets in Θ in each iteration, but the current presentation of the algorithm is clearer and technically sufficient for our purposes.

Algorithm 3 The social choice MDP $\mathcal{M}$, the initial state $\hat{s}$, and the partition of the agents into types $\widehat{\Theta}$ are treated as global variables.

```
1: procedure COMPUTE-NONDICT($\mathcal{M} = (\mathcal{S}, A, R, P), \hat{s}, \widehat{\Theta}$)
2:     return SEARCH($\widehat{\Theta}, \mathcal{S} \times A$)

1: procedure SEARCH($\mathcal{X}, \Psi$)
2:     if $\mathcal{X} = \{\theta\}$ then                                    ▷ Only one set left to preclude
3:         if $\theta = \{i\}$ then
4:             $M \leftarrow -\infty$
5:             for each $T \in \mathcal{E}_{\widehat{\Theta}}$ do
6:                 Select $s \in T$ arbitrarily
7:                 $\Theta \leftarrow$ CARVE-OUT($\mathcal{M}, s, \widehat{\Theta}$)
8:                 $\Psi' \leftarrow \Psi \setminus \{(s, \mathrm{top}(s(i)))\}$
9:                 $M \leftarrow \max\{M,$ SOLVE($\mathcal{M}, \hat{s}, \Theta, \Psi'$)$\}$
10:            return $M$
11:        else                                                          ▷ $|\theta| > 1$
12:            return SOLVE($\mathcal{M}, \hat{s}, \widehat{\Theta}, \Psi$)
13:    else
14:        Select arbitrary $x, y \in A$, $y \neq x$, and $\succ_x, \succ_y \in \mathcal{L}$ such that $\mathrm{top}(\succ_x) = x$ and $\mathrm{top}(\succ_y) = y$
15:        Select $\theta \in \mathcal{X}$ arbitrarily
16:        Select $s \in \mathcal{S}$ such that $s(i) = \succ_x$ for all $i \in \theta$ and $s(i) = \succ_y$ for all $i \in N \setminus \theta$
17:        $\Psi_x \leftarrow \Psi \setminus \{(s, a) : a \in A \setminus \{x\}\}$                    ▷ Restrict $s$ to $x$
18:        $\Psi_y \leftarrow \Psi \setminus \{(s, a) : a \in A \setminus \{y\}\}$                    ▷ Restrict $s$ to $y$
19:        $\Psi_{-xy} \leftarrow \Psi \setminus \{(s, x), (s, y)\}$                          ▷ Restrict $s$ to $A \setminus \{x, y\}$
20:        return $\max\{$SEARCH($\mathcal{X} \setminus \{\theta\}, \Psi_y$), SEARCH($\{\theta\}, \Psi_x$), SOLVE($\mathcal{M}, \hat{s}, \widehat{\Theta}, \Psi_{-xy}$)$\}$
```

To prove the lemma it is sufficient to show that $E_{\widehat{\Theta}}(s) = \{s\}$, and the proof of this equality is almost identical to the proof of Lemma 5.1 (and follows because the preferences in state s are the same for all agents with the same type). The desired properties of COMPUTE-NONDICT now easily follow from the discussion above and from the results of Section 4.

Theorem 5.8. *Let $\mathcal{M} = (\mathcal{S}, A, R, P)$ be a social choice MDP, let R be an anonymous reward function, let $\widehat{\Theta}$ be the partition of the agents into types, and let $\hat{s} \in \mathcal{S}$. Furthermore, assume that $m = |A| = \mathcal{O}(1)$ and $t = |\widehat{\Theta}| = \mathcal{O}(1)$. Then:*

1. COMPUTE-NONDICT$(\mathcal{M}, \hat{s}, \widehat{\Theta})$ *can be implemented in polynomial time in n.*
2. COMPUTE-NONDICT$(\mathcal{M}, \hat{s}, \widehat{\Theta})$ *returns the value on $\hat{s}$ of the optimal deterministic nondictatorial policy.*

Proof. For part 1, we first claim that all the executions of SOLVE can be implemented in polynomial time. It follows from Lemmata 5.1 and 5.7 that when the procedure is executed with a restriction Ψ and a partition Θ, Ψ is symmetric with respect to E_Θ. The partition given as input to SOLVE is either $\widehat{\Theta}$ or a refinement obtained via a single application of CARVE-OUT, i.e., its cardinality is at most $m!|\widehat{\Theta}|$ (using Observation 5.3). As before, the running time of SOLVE can now be bounded using Lemma 4.3 and Theorem 4.10.

In order to bound the total number of executions of Solve, note that our search tree has height at most $|\widehat{\Theta}|$ and a branching factor of at most two, therefore this number is constant given that $|\widehat{\Theta}| = \mathcal{O}(1)$.

We now turn to part 2 of the theorem. It is clear by the discussion above that the algorithm indeed returns the value on $\hat{s}$ of a policy that is nondictatorial; it remains to prove that this returned value is optimal. Let π^* be an optimal deterministic nondictatorial policy for $\mathcal{M}$ and the initial state $\hat{s}$. Crucially, for every state s defined in line 16 of Search it holds that

$$\{(s,a) : \ a \in A\} \subseteq \Psi_x \cup \Psi_y \cup \Psi_{-xy},$$

that is, for every $a \in A$ there is a recursive calls to Search where (s,a) is admissible, and nothing is precluded. In particular, for every s defined in Search we have that $(s, \pi^*(s)) \in \Psi_x \cup \Psi_y \cup \Psi_{-xy}$. By repeatedly following the branch of the search tree that admits $(s, \pi^*(s))$ we eventually reach a call to Solve.

Consider first the calls to Solve in lines 12 and 20 of Search. The restriction Ψ submitted as input to the procedure satisfies $\{(s, \pi^*(s)) : \ s \in \mathcal{S}\} \subseteq \Psi$, hence Solve optimizes over a space of policies that contains π^* and would return an optimal value.

Next, consider the call to Solve in line 9 of Search$(\mathcal{X}, \Psi)$, and let $i \in N$ be the agent such that $\theta = \{i\}$ in line 3. Since π^* is nondictatorial, there exists $s^* \in \mathcal{S}$ such that $\pi^*(s^*) \neq \text{top}(s^*(i))$. This is the state where the optimal policy does not pick the most-preferred alternative for agent i. Let $s \in \mathcal{S}$ be the state selected in line 6 such that $(s^*, s) \in E_{\widehat{\Theta}}$. Agent i is a singleton subset of $\widehat{\Theta}$, therefore $s(i) = s^*(i)$, and in particular $\text{top}(s(i)) = \text{top}(s^*(i))$. Let $\Psi' = \Psi \setminus \{(s, \text{top}(s(i))\}$ and $\Psi^* = \Psi \setminus \{(s^*, \text{top}(s^*(i))\} = \Psi \setminus \{(s^*, \text{top}(s(i))\}$.

Let $\mu : N \to N$ be a permutation on the agents such that for all $j \in N$ and all $\theta \in \widehat{\Theta}$, $j \in \theta$ if and only if $\mu(j) \in \theta$, and such that $\mu(s^*) = s$. Similarly to the proof of Theorem 5.4, μ induces an isomorphism between $(\mathcal{M}, \Psi^*)$ and $(\mathcal{M}, \Psi')$ that satisfies $\mu(\hat{s}) = \hat{s}$. Hence, letting $\Theta = \text{Carve-Out}(\mathcal{M}, s, \widehat{\Theta})$ and $\Theta^* = \text{Carve-Out}(\mathcal{M}, s^*, \widehat{\Theta})$, we have that

$$\text{Solve}(\mathcal{M}, \hat{s}, \Theta, \Psi') = \text{Solve}(\mathcal{M}, \hat{s}, \Theta^*, \Psi^*).$$

□

6 Discussion

Below we discuss several aspects of our work and propose some extensions.

Where do the agents' transition models come from? We have not addressed the challenge of constructing appropriate transition models for the agents, but rather these transition models are given as input to our algorithms. There are a variety of techniques that may be suitable, ranging from automated methods such as machine learning (see, e.g., [8]) and hidden Markov models, to marketing and psychological approaches. We wish to emphasize though that the purpose of this paper is to provide the first principled approach to decision making in environments where currently one is not available. Even a coarse partition of the agents into just several types, and subsequent application of our algorithmic results, would be an improvement over the status quo in organizations like MoveOn. Over time this initial partition can be gradually refined to yield better and better approximations of reality.

Applications of the model. We already mentioned a number of intriguing applications in the introduction. Applications should feature a population of stakeholders, in order to motivate the normative approach of social choice; e.g., in collaborative science, the scientists need to be motivated to remain an active part of the collaboration. The framework of dynamic social choice also requires a population with preferences that change in response to past decisions.

Recommendation systems are one such class of applications. Consider a recommendation system that, at any point, recommends a single product (say, among the products in some category). Users express their preferences by rating the products, and these ratings may be updated as recommended products are sampled by the users. The goal of the designer may be, for example, to recommend a specific product, but the recommendations must be made in a way that reflects the preferences of the users, that is, in a socially desirable way. In this case it is plausible to assume that the preferences of users only change by moving the currently recommended product up or down in a user's ranking.

Many recommendation systems provide a ranking of the products in a category; Tripadvisor (`www.tripadvisor.com`) is a prominent example. In this case the recommendation system should be modeled as a *social welfare function* rather than a social choice function. Fortunately, it is straightforward to suitably modify our model by replacing A with $\mathcal{L}(A)$ as the set of actions. Under the assumption $m = \mathcal{O}(1)$, which was required in our results regardless, we believe that our techniques and results would carry over to the modified model. That said, we note that an interesting aspect of websites like Tripadvisor is that the users only rate a small subset of the alternatives; an accurate model would have to take this aspect into account.

Dynamic agents and alternatives. An implicit assumption that we have made is that the sets of agents and alternatives remain fixed throughout the dynamic process. We believe that this assumption can be justified. Indeed, in some settings (like MoveOn) there are many agents and the overall proportion of types remains similar even though agents arrive and depart. In other settings (like scientific collaboration) the set of agents does indeed remain fixed.

The assumption about the fixed set of alternatives has more bite. Nevertheless, we argue that in our examples one can think of the set of alternatives as fixed by "bundling" together alternatives. For example, in political settings the big issues (the environment, human rights, religion, etc.) can remain fixed for decades, while the specific candidates who advocate and represent these issues may change. Going back to the Debian example, one can bundle specific features into fixed categories such as user interface, efficiency, etc.

Combinations of axioms. Our results were formulated with the goal of finding an optimal policy that satisfies a specific constraint. However, observe that requiring several local anonymous axioms corresponds to taking their intersection, and hence a combination of local anonymous axioms is itself a local anonymous axiom. We further believe it is possible to modify Algorithms 2 and 3 to find an optimal policy that satisfies any combination of ontoness, nondictatorship, and local anonymous axioms.

The Markovian assumption. We assume that agents' preferences are Markovian, that is, independent of the history. Note that technically this assumption is without loss of generality: one can expand the MDP to include a state for each possible history of bounded length. However, in such a model policies would no longer coincide with social choice functions, giving rise to decision making mechanisms that possibly depend on history, and leading to a need to adapt the classical axioms of social choice to this framework.

Computing optimal scoring rules. The family of (positional) scoring rules is perhaps the most prominent family of social choice functions. Each scoring rule is represented by an m-vector of real numbers $(\alpha_1, \ldots, \alpha_m)$, and each agent awards α_k points to the alternative it ranks in the k'th place; the alternative with most points wins the election. Plurality is the most prominent function in this family, with the vector $(1, 0, \ldots, 0)$, and Borda, with the vector $(m-1, m-2, \ldots, 0)$, is another example. It is natural to ask whether, given a social choice MDP, one can find an optimal scoring rule in polynomial time in n, assuming as before that $m = \mathcal{O}(1)$. In this case the answer is simple. Indeed, Procaccia et al. [17, Lemma 5.5] show that the number of distinct scoring rules is at most $\mathcal{O}((3n)^{m^2})$, and give a method of enumerating them. Therefore, one can find the optimal scoring rule by simply trying them all.

Dropping the assumptions about the number of alternatives and types. In general the state space of a social choice MDP is huge. We have circumvented this problem by assuming a constant number of alternatives and types and exploiting symmetries. It may be possible to replace these assumptions with different ones. For example, Guestrin et al. [13] present algorithms that exploit the structure in factored MDPs; they use an approximate value function that can be represented as a linear combination of basis functions, where each basis function is defined over a small number of variables. Unfortunately, their techniques do not seem to do well with respect to computing socially desirable policies, as nonlocal axioms such as ontoness couple the variables in a way that invalidates some key assumptions. This leaves open a challenging direction for future research: is it possible to develop new techniques for computing socially desirable policies under the assumption of an approximate value function representation, as in the work of Guestrin et al.?

On nonlocal axioms. Assuming a constant number of alternatives and types, the question of designing optimal policies that satisfy nonlocal axioms in polynomial time in n remains open, with the obvious exception of ontoness and nondictatorship. In the future we also plan to investigate heuristic techniques for tackling nonlocal axioms.

On nonreachable states. Consistent with the conceptualization of a policy as a social choice function we interpreted nonlocal axioms over all states, even states that are nonreachable given a distribution on initial states (recall that this is required for a well-defined optimal policy with nonlocal axioms). In particular, this allows for a policy to be nondictatorial while selecting the most preferred alternative for the same agent in all reachable preference profiles. Similarly, this insists that a policy is nonmonotonic even when it is monotonic with respect to all alternatives selected in all reachable states. In problems where the initial distribution does not have full support, for example placing all weight on one state, this leaves open the challenging problem of computing, for example, an optimal deterministic policy that is nondictatorial over the states that are reachable by the policy.

Acknowledgments

We warmly thank Felix Fischer and Ian Kash for an insightful remark that inspired Algorithm 3. We also thank Craig Boutilier, Ronald Ortner, and Satinder Singh for helpful discussions. This work is supported in part by NSF grant CCF-0915016.

References

[1] E. Altman. *Constrained Markov Decision Processes.* Chapman and Hall, 1999.

[2] R. E. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

[3] D. Bergemann and J. Välimäki. The dynamic pivot mechanism. *Econometrica*, 78:771–789, 2010.

[4] C. Boutilier, R. Dearden, and M. Goldszmidt. Exploiting structure in policy construction. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1104–1111, 1995.

[5] C. Boutilier and A. D. Procaccia. A dynamic rationalization of distance rationalizability. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1278–1284, 2012.

[6] R. Cavallo, D. C. Parkes, and S. Singh. Optimal coordinated planning amongst self-interested agents with private state. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 55–62, 2006.

[7] V. Conitzer and T. Sandholm. Complexity of mechanism design. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 103–110, 2002.

[8] E. Crawford and M. Veloso. Learning dynamic preferences in multi-agent meeting scheduling. In *Proceedings of the 5th IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT)*, pages 487–490, 2005.

[9] D. Dolgov and E. Durfee. Stationary deterministic policies for constrained MDPs with multiple rewards, costs, and discount factors. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1326–1331, 2005.

[10] D. Dolgov and E. Durfee. Resource allocation among agents with MDP-induced preferences. *Journal of Artificial Intelligence Research*, 27:505–549, 2006.

[11] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[12] A. Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica*, 45:665–681, 1977.

[13] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research*, 19:399–468, 2003.

[14] M. O. Jackson and H. F. Sonnenschein. Overcoming incentive constraints by linking decisions. *Econometrica*, 75:241–257, 2007.

[15] D. C. Parkes, R. Cavallo, F. Constantin, and S. Singh. Dynamic incentive mechanisms. *AI Magazine*, 31(4):79–94, 2010.

[16] D. C. Parkes and S. Singh. An MDP-based approach to online mechanism design. In *Proceedings of the 17th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 791–798, 2003.

[17] A. D. Procaccia, A. Zohar, Y. Peleg, and J. S. Rosenschein. The learnability of voting rules. *Artificial Intelligence*, 173(12–13):1133–1149, 2009.

[18] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, 1994.

[19] B. Ravindran and A. G. Barto. Symmetries and model minimization in markov decision processes. CMPSCI technical report 01-43, 2001.

[20] B. Ravindran and A. G. Barto. SMDP homomorphisms: An algebraic approach to abstraction in semi-Markov decision processes. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1011–1016, 2003.

[21] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[22] M. Tennenholtz. Transitive voting. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, pages 230–231, 2004.

[23] W. Thomson. The fair division of a fixed supply among a growing population. *Mathematics of Operations Research*, 8:319–326, 1983.

[24] N. Tideman. *Collective Decisions and Voting.* Ashgate, 2006.

[25] M. Zinkevich and T. Balch. Symmetry in Markov decision processes and its implications for single agent and multiagent learning. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*, pages 632–640, 2001.