# Deriving Consensus in Multiagent Systems

Eithan Ephrati
Computer Science Department
University of Pittsburgh
Pittsburgh, PA
tantush@cs.pitt.edu

Jeffrey S. Rosenschein
Institute of Computer Science
The Hebrew University
Jerusalem, Israel
jeff@cs.huji.ac.il

**Abstract**

Consider the designers of a multiagent environment, who are charged with establishing the rules by which agents in an encounter will interact. Once the rules of encounter have been determined, each builder of each agent is free to design his own machine any way that he wants. However, the rules that were established will certainly affect the choices he makes in building his own agent.

In this article we suggest an economic decision process that can be used to derive multiagent consensus, namely, the Clarke Tax mechanism [6]. Consensus is reached through the process of voting; each agent expresses its preferences, and a group choice mechanism is used to select the result. Clarke Tax-like mechanisms provide a set of attractive alternatives for the designers of multiagent environments, particularly if those environments consist of individually motivated heterogeneous agents.

The Clarke Tax mechanism has many desirable properties such as non-manipulability, individual rationality, and maximization of the agents' global utility. However, though theoretically attractive, the Clarke Tax presents a number of difficulties when one attempts to use it in practical implementations. This article examines how the Clarke Tax could be used as an effective consensus mechanism in domains consisting of automated agents. In particular, we consider how agents can come to a consensus without needing to reveal full information about their preferences, and without needing to generate alternatives prior to the voting process.

## 1 Introduction

Multiagent systems have recently emerged as an important new focus for research in artificial intelligence (AI). The same factors that made distributed computer systems the norm, namely advances in miniaturization and networking technology, have had an effect on the kinds of sophisticated autonomous systems that designers can build. When the domain has been suitable, designers have found compelling reasons to consider distributed AI solutions.

Suitability of the domain might involve a natural geographic distribution in the sources of incoming data (so, for example, much research has focused on distributed systems of sensors carrying out cooperative data fusion [10, 14]), or a natural functional distribution best addressed through modular problem solvers (such as, for example, the modules at work in the Pilot's Associate project [80]).

All this work on cooperative problem solvers arises as designers try to figure out the most effective way of dealing with difficult domains. Centralized solutions might be possible, but require a prohibitive amount of communication overhead, or require too much coordination from the diverse group of designers themselves. There are other kinds of problems, however, where distribution is not simply a *tool* to be used in designing a problem solving system—it is an innate feature of the system itself. The Internet, for example, provides an extreme example of a real-world distributed environment. Systems of agents operating effectively in such an environment must cope appropriately with distribution.

There has been a great deal of research in recent years on the common language that such heterogeneous computer systems can use to communicate with one another (efforts such as KIF, the Knowledge Interchange Format, fall into this category [28]). Our research assumes that this kind of common language already exists. The questions we ask revolve around *how* such a language will be used. Once agents share a language and are able to communicate, how will they choose to interact?

Consider the designers of a multiagent environment, who are charged with establishing the rules by which agents in an encounter will interact. Once the rules of encounter have been determined, each builder of each agent is free to design his own machine any way that he wants. However, the rules that were established will certainly affect the choices he makes in building his own agent (these environment design issues are examined, for example, in [71]).

This article is concerned with interaction mechanisms that can ensure that the agreements reached by individually motivated agents, and the process by which these agreements are reached, display certain desirable properties. The need for heterogeneous agents to reach agreement is increasingly a part of real-world distributed systems. The techniques we explore here provide tools for building environments capable of inducing beneficial coordination.

## 1.1 Examples of Applications

Consider the world's current telecommunication network infrastructure. Various physical pieces of this network are controlled by different companies and organizations. The management of these pieces is basically automated, and computers decide on the routing of messages and data between points. At times, messages may be more efficiently routed when resources (communication lines, short and long term storage, etc.) from several networks are exploited. The machines that can make the decision to share resources and route their packets effectively are getting their jobs done better than machines that do not cooperate (and sometimes, individual machines will not be able to carry out their tasks at all without help from another resource-controlling machine). Here, decentralization among machines, the fact that several heterogeneous, distinct problem solvers are interacting, is not an artificial design decision, imposed from above. It is an intrinsic part of the system, an unavoidable consequence of the fact that the resources these machines control belong to different individuals or companies.

Another example might involve an automated method for choosing a carrier for long-distance telephone calls. In this system, when a customer lifts the handset and dials a long-distance call, a microprocessor within the phone automatically collects bids from the various carriers. Each company's computer simultaneously declares the price per minute at which it is willing to carry the call. The phone's microprocessor collects these bids, and decides which company wins. The exact method by which the carrier is chosen, and how much it is paid, can drastically affect the appropriate bidding strategy for each of the telephone company computers [72]. The designers of the overall environment can induce certain kinds of behavior on the part of the heterogeneous computers by establishing particular rules for the interaction.

As another example, consider meeting scheduling software. The problem solving that such software carries out cannot be done in isolation; the very nature of the task it performs involves interaction among machines. Distribution here arises as a natural aspect of, indeed a requirement of, the environment. A person who uses such software might be able to specify her preferences regarding alternative schedules. The interaction of scheduling agents is an encounter that is sensitive to the strategies being used. For example, one might conceal certain time slots so as to force a meeting at ones most convenient time. The mechanisms that establish how these programs set meetings can fundamentally alter the strategies of their users (and the outcome of their interaction).

The techniques explored in this article can be directly applied to solving problems like those discussed above. For example, our approach has already been used to design a meeting scheduling system that induces its users to reveal their true scheduling preferences, while efficiently establishing an optimal global schedule [23].

## 1.2 Distributed Artificial Intelligence

Multiagent activity is obviously facilitated by, and sometimes requires, agreement by the agents as to how they will act in the world. Reaching such a consensus has been a major concern of research in Distributed Artificial Intelligence (DAI), and various alternative methods of achieving consensus among groups of autonomous agents have been suggested in the DAI literature.

Within the field of DAI, a distinction is sometimes made between two paradigms: *Distributed Problem Solving* (DPS) and *Multiagent Systems* (MAS) [15].[1] Within the first paradigm, agents are assumed to be created by the same designer (or group of designers), and thus, work together to solve common goals. Using economic terms, the agents have a common preference profile which yields an identical utility function. The common goal(s)

[1]These terms are falling into disuse, as "multiagent systems" becomes the all-encompassing term for multiagent research. The distinction between research emphases discussed in the article, however, remains. In the meantime, we'll continue using these terms in the absence of superior alternatives.

may be decomposed into subgoals that are allocated to different agents in the group. Coordination is needed to allow efficient distributed activity towards the achievement of the global goal. While pursuing their tasks, agents may communicate, share their knowledge, and help each other.

In the second paradigm, that of Multiagent Systems, agents may have their own private goals and act selfishly towards the achievement of these goals. Therefore each agent may have its private profile of preferences, and a distinct individual utility function. An assumption can be made that agents will help one another only when it is in their own best interests to do so.

The distinction between Distributed Problem Solving and Multiagent Systems should really be seen more as a distinction between research agendas, rather than between running systems. Certainly it will not always be obvious to an outside observer whether a given distributed system falls into one paradigm or the other. A single designer may have built his agents to act competitively, believing it improves overall system efficiency. Similarly, individually motivated agents might be seen sharing information and helping one another, because they have determined that it is in their own best interests to act that way. However, the research questions asked by a researcher in DPS may be distinct from those asked by an MAS researcher (despite a good deal of overlap in their research agendas). In particular, if a DPS researcher can show that acting in a particular way is good for the system as a whole, he can *impose* this behavior on all the agents in the system at design time. For the MAS researcher, such an alternative is unavailable. At best, he might be able to design aspects of the environment that motivate all the (selfish) agents to act in a certain way. This need for *indirect* incentives is one element that distinguishes MAS research from DPS research.

The research reported on in this article is solidly within the area of Multiagent Systems. We consider multiagent societies that consist of heterogeneous agents that have been manufactured by *different* designers. Therefore we take the point of view that each agent acts in the best interests of its own designer, i.e., each agent acts rationally to maximize its own subjective expected utility (in the sense of Savage [77], who simultaneously axiomatized utility and subjective behavior). This is also in keeping with a large (and growing) body of work within artificial intelligence that attributes rationality (or explores the consequences of attributing rationality) to autonomous agents. See, for example, [46, 25, 70, 12, 42, 34].

### 1.3 Overview of this Article

In this article we present a method for reaching consensus based on the Clarke Tax mechanism [6, 7] (CTm), and consider how this mechanism could be used among rational automated agents. Parts of this work have appeared previously in [18, 22, 20, 19].

In Section 2.1 we present the model of interaction that this article addresses, followed by an illustrative example and some initial definitions. Section 2.4 introduces the CTm. Section 3 suggests ways for agents to assess the worth of alternative states (a necessary ingredient to establishing preferences among alternatives).

The CTm has several major drawbacks for those who would want to use it in a practical implementation. In each of the subsequent sections of this article, we suggest a solution to one or more of these issues:

1. **The set of final candidates over which the agents vote is assumed to be determined in advance of the vote** — It is not obvious how this set should be generated, especially in a non-expensive way and without full knowledge of the individual agents' goals.

2. **Agents are required to calculate and reveal their exact and entire profile of preferences over the entire set of candidates** — Although this was actually the consequence desired by the original inventor of the CTm, the designers of autonomous agents might prefer to conceal some information regarding their preferences. We use the CTm for the purposes of reaching consensus, and want the preferences that agents reveal to be true; however, we (as designers of the mechanism) would prefer the agents not to have to reveal *all* their preferences (the original CTm inventor did want them to reveal all their preferences). Moreover, the *exact* calculation of the entire set of preferences may be computationally expensive and under some circumstances extremely difficult.

3. **All agents have equal influence on the decision procedure** — Although desirable in many human scenarios, this fact might not be acceptable to all the designers of the agents involved.

4. **The mechanism is sensitive to possible coalition formation** — Groups of agents may collude in order to bend the groups' decision in their favor. Although rational, such behavior may distort the outcome and cause it to be inefficient from a global (and perhaps local) perspective.

Generation of candidate states by a central planner (issue 1) is discussed in Section 4. Determining this set of final candidates may be quite expensive computationally, and requires a full revelation of the individual goals. In Section 5 we address these two problems. We present a new approach to deriving multiagent plans by employing a variation of the CTm that eliminates the need to generate candidates ahead of time (i.e., an alternative approach to issue 1). In addition, the method maintains the agents' privacy more effectively (issue 2); agents can reach consensus in a Clarke Tax-like voting procedure, without having to reveal full preferences and goals (unless that is actually necessary for consensus to be reached). Agents iteratively converge to a *plan* that brings the group to a state maximizing social welfare. Section 5.6 shows how to relax one assumption of the original decision procedure, namely that agents have equal influence (issue 3). In Section 5.7 we suggest a heuristic refinement of the plan aggregation process.

Section 6 considers the situation where final candidates already exist, but we are interested in a method of protecting agents' privacy and saving agents the burden of calculating their full preferences over all alternatives (unless that is necessary to reach consensus). This is a more direct treatment of issue 2, without the need to deal with candidate generation.

Section 7 considers the fragility of the CTm with respect to possible coalition formation (issue 4).

We conclude with a brief review of related work in Section 8.

Appendix B describes in greater detail the desirable solution properties for which we are looking, followed by a some useful background material from voting theory, economics, and game theory.

# 2 The Clarke Tax

The CTm was originally presented in the economics literature as a way to solve the so-called *free rider problem* [64]. This problem was considered to be unsolvable prior to the introduction of the CTm.[2]

The original economic scenario was concerned with the way some central agency (government, project manager, etc.) could choose between two possible projects ("public goods") based on the individual preferences of the members of the society. Once the decision is reached, each member of society is required to contribute money towards its realization based on the preferences that the member stated. It was soon realized that rational individuals will tend to declare untruthful preferences so as to be able to pay less once the decision is reached (i.e., understate their true preferences). These untruthful members that pay less are getting a "free ride" from the others. Such behavior may, under many circumstances, yield an inefficient global decision. This phenomenon of it being rational to falsely understate preferences has been overcome by the Clarke Tax mechanism. For that reason it is also called a "preference revelation process."

Although we are concerned with somewhat different scenarios, the CTm is also a very desirable decision procedure for deriving consensus (coordination) in multiagent environments.

## 2.1 The Scenario

Imagine a group $\mathcal{A}$ of $N$ agents (possibly created by different designers) operating in a world currently in the state $s_0$, facing the decision of what to do next. One way of formulating this problem is to consider that the agents are trying to agree into which member of the set $\mathcal{S}$ of $m$ possible states the current world should be transformed.

Each agent in $\mathcal{A}$ has its private goal. This goal gives rise to a *worth*, or utility, that it associates with each state. In turn that worth induces a preference relation over states and plans. Agent $i$'s true worth for state $k$ will be denoted by $w_i(k)$. The preferences *declared* by an agent might differ from his true preferences. A decision procedure that chooses one state from $\mathcal{S}$ is a function from the agents' declared preferences to a member of the set $\{1, ..., m\}$.

[2] Actually, a more general family of mechanisms, related to the Clarke Tax, was independently introduced by Groves [40] and received wider recognition. The underlying idea on which this family of mechanisms relies was first discovered by Vickrey in the context of a public auction [90].

It maps the agents' declared preferences into a group decision as to how the world will be transformed.

We are looking for an efficient decision procedure that will enable the agents to reach consensus and agree on a final state.

## 2.2 An Example

Consider a simple scenario in the slotted blocks world.[3] There are four slots $(a, b, c, d)$, five blocks $(1, 2, 3, 4, 5)$, and the world is described by the relations: $\mathbf{On}(Obj_1, Obj_2)$—$Obj_1$ is stacked onto $Obj_2$; $\mathbf{Clear}(Obj)$—there is no object on $Obj$; and $\mathbf{At}(Obj, Slot)$—$Obj$ is located at $Slot$. The function $\mathbf{loc}(Obj)$ returns the location (slot) of $Obj$.

**Figure 1**

There are three agents operating in the world. The start state is shown at the far left of Figure 1. As further represented in that figure, these agents have (respectively) the following goals:
$g_1 = \{At(4, c), At(2, b)\}\langle Worth = 12\rangle$,
$g_2 = \{On(2, 4), On(5, 2)\}\langle Worth = 14\rangle$,
$g_3 = \{On(3, 2), At(2, c)\}\langle Worth = 16\rangle$.
Slots themselves function as (stationary) objects (e.g., block 1 in slot $b$ could be described by $On(1, b)$).

There is only one available operator: $\mathbf{Move}(Obj_1, Obj_2)$—place $Obj_1$ onto $Obj_2$. This operator can be characterized by the following STRIPS-like lists:
$[Prec: Clear(Obj_1),\ Clear(Obj_2),\ On(Obj_1, Obj_x)]$,
$[Del: On(Obj_1, Obj_x),\ Clear(Obj_2),\ At\ (Obj_1, loc(Obj_1))]$,
$[Add: On(Obj_1, Obj_2),\ At(Obj_1, loc(Obj_2)))]$.
Assume that when a single agent performs the $Move$ operation there is a cost of 4, while if two agents $Move$ an object together the operation costs a total of 3 (1.5 each).

The problem is to find a plan to be carried out by all three agents that will bring the world to a compromise state that is in consensus. Preferably it will be a state that maximizes the group's utility (by some definition of group utility).

[3] While the blocks world is inappropriate for studying many real-world issues in robotics, it remains broadly suitable for the study of abstract goal interactions, which is how we use it.

## 2.3 Basic Definitions

Throughout this article we use the following assumptions and definitions:

- There exists some kind of monetary system in the multiagent environment that allows side payments between agents. In other words, one agent can pay another and compensate him for doing some action, accepting some result, etc. We do not have to actually assume the existence of explicit currency; agents can trade using promised actions, for example, instead. The important point is that agreements can be facilitated by side payments, with utility transferred from one agent to another.

  Each agent $a_i$ has its own goal $g_i$, which is a set of predicates. $w_i(s)$ is the worth (or utility) that agent $i$ assigns to the state $s$. If $s \models g_i$ then $w_i(s)$ is equal to the worth that $i$ associates with his goal, $w_i(g_i)$. The agents' utility functions are additively separable. That is, an agent is indifferent between establishing a state $s$, that it assesses a worth $w_i(s)$, and establishing a state $q$, that it assesses a worth $w_i(q)$, if the agent (in the latter case) is receiving a side payment of $w_i(s) - w_i(q)$.

- The plan $P(s_0 \rightsquigarrow s_k)$ is the sequence $s_0 \overset{op-1}{\rightsquigarrow} s_1 \overset{op-2}{\rightsquigarrow} \ldots s_{k-1} \overset{op-k}{\rightsquigarrow} s_k$. $l(P(s_0 \rightsquigarrow s_k))$ is the length ($k$) of the plan that starts in $s_0$ and ends in $s_k$.

- $C(i, s_0 \rightsquigarrow s_g)$ denotes the minimal cost that it would take for a single agent $i$, in state $s_0$, to bring about any state $s_g$ that satisfies $g$ ($s_g \models g$). $P^*(i, s_0 \rightsquigarrow s_g)$ denotes the set of all plans that will bring about $s_g$ with this minimal cost.

  $C(s_0 \rightsquigarrow s_1)$ is the minimal cost needed to move the world from $s_0$ into $s_1$, using any combination of (multi-)agent actions. Given a plan $P(s_0 \rightsquigarrow s_1)$, $C_i(s_0 \rightsquigarrow s_1)$ is $i$'s share of the work in that plan.

## 2.4 Clarke's Bidding Mechanism

The CTm is one of many one-shot voting-by-bid mechanisms that were invented within the fields of voting theory and economics. Unlike these other mechanisms, the CTm is non-manipulative; the basic idea of Clarke's Mechanism is to make sure that each voter has only one dominant strategy, telling the truth. This phenomenon is established by slightly changing the classic sealed-bid mechanism:[4] instead of simply collecting the bid of the winning bidder, each agent is fined with a tax. The tax equals the portion of his bid that made a difference to the outcome. The example in Figure 2 shows how to calculate this tax. Each row of the table shows several pieces of information regarding an agent. First, his preferences for each state are listed. Then, the total score that each state would have gotten, had the agent *not* voted, are listed. An asterisk marks the winning choice in each situation.

[4] In the sealed-bid mechanism, agents bid by submitting a secret offer; the agent with the highest bid wins. It can be shown that this mechanism is manipulable; agents might benefit by submitting false bids.

**Figure 2**

For example, when all the agents voted, state $s_3$ was chosen. If $a_2$ had not voted, $s_1$ would have been chosen. The score in this situation would have been $\langle 17, -22, 5 \rangle$, and $s_1$ would have beaten $s_3$ by 12. Thus, agent $a_2$ has affected the outcome by his vote, and he has affected it by a "magnitude" of 12; he is therefore fined 12. Agents $a_1$, $a_3$, and $a_5$ are not fined because even without the vote of each of them (separately), $s_3$ would still have been chosen.

Given this scheme, revealing true preferences is the dominant strategy. An agent that overbids (so that some given state will win) risks having to pay a tax larger than his true preferences warrant. Similarly, the only way to pay less tax is to actually change the outcome—and any agent that underbids (to change the outcome and save himself some tax) will always come out behind; the saved tax will never compensate him for his lost utility.[5]

Note the distinction between private preferences and public behavior. It is relatively easy to monitor and enforce public behavior; so, for example, it will be publically known if an agent owes tax, and paying the tax can then be enforced. The private preferences of an agent are entirely different. There is no way to observe the private preferences of an agent, or directly know if its stated preferences match its private preferences. Thus, there is the need for an indirect mechanism like the CTm. But once such a mechanism is in place, we assume corresponding public behavior (like tax paying) can be ensured.

## 2.5 Attributes of the CTm

The Clarke Tax decision mechanism is appealing because it satisfies many desirable properties of social decision mechanisms (for a wide-ranging discussion of common solution concepts for social decision processes, see Appendix B).

First, it is not manipulable by individuals—any other declaration of preferences is *dominated* by declaring the truth. Therefore, it saves each agent from the computational complexity of guessing what the others' preferences and strategies are, what the negotiation set is, and how it can be manipulated. This simplicity of strategy is highly desirable in the design of automated agents. The agents tell the truth out of their own self-interest—there is no need to assume that agents will act benevolently by design. Thus, the process answers both the "simplicity" and "stability" criteria (see Appendix B).

[5] For a formal proof that revealing true preferences is the dominant strategy, consider Proof 2 in Appendix A where $z_i = 1$ for all $i$.

A second advantage of the technique is that it satisfies several desirable criteria, including the "Condorcet winner" (—a choice that would have beaten every other choice in pair-wise votes is guaranteed to be chosen by the mechanism [26]), "monotonicity" (—by giving an alternative a higher value, an agent cannot undermine the alternative's selection), "independence of irrelevant alternatives" (—removal of any "unchosen" alternative from the set of alternatives will not change the outcome [88]), "individual rationality" (an agent may only gain utility by taking part in the process), "anonymity" and "neutrality" (—the identity of a voter or the name of an alternative has no influence on the outcome), "expressiveness" (—preferences are expressed using the actual cardinal utilities), it is relatively simple, and finally the process can be designed to preserve privacy since the actual choice function uses only the total sum of preferences.

A third advantage is that the alternative chosen by the Clarke Tax mechanism answers a social welfare criterion similar to the summation criterion mentioned in Appendix B [89]. In fact, the Clarke mechanism is just one member of the family of Groves mechanisms [40]. It has been proven in Economics that any decision mechanism that chooses a state with the same properties as the CTm does, and that also has telling the truth as a dominant strategy, belongs to this family [36]. However, the CTm requires the least amount of tax to be paid, from among the members of this decision mechanism family [54]. It guarantees the best minimal utility level for each of the participants, and is the only mechanism within this family that has no free rider problem [63] (i.e., an agent will not be tempted to avoid the process, hoping to benefit from the decision without the risk of paying the tax).

# 3 Calculation of Preferences

Since telling the truth is the dominant strategy when the Clarke tax is being used, it is in each agent's interest to compute his true preferences over the potential alternative states. The preference profile is based on the worth that each agent assigns to each alternative state. The way by which (true) worth is associated with states obviously has a crucial influence on the outcome of the group decision. In this section we suggest several conceptually different approaches for an agent to determine the worth of a given state.

As an example, consider the scenario in the blocks world as described in Figure 1. For the sake of simplicity, assume that each $Move$ action costs 1, and that $w_i(g_i) = C_i(s_0 \rightsquigarrow g_i)$. In that case, $w_1(g_1) = 2$, $w_2(g_2) = 3$, and $w_3(g_3) = 4$. Assume that the agents in state $s_0$ are faced with choosing among six alternative future states as shown in Figure 3 (we will later discuss how alternatives are to be generated).

**Figure 3**

## 3.1 The Assessment of a Single State's Worth

For the "worth" of a given state to be meaningful, it should be a function of the initial state, the state itself, and the agent's goal (for a similar argument, see [24]). It is easier for a social consensus procedure to find compromise consensus states when agents' worth functions are not binary (i.e., agents will be partially satisfied by states, see for example [94, 19]).

According to the "all-or-nothing" approach, the agent assigns the full value of his goal to any state that satisfies it, and zero otherwise. In the example in Figure 3, $s_4$ would be chosen, causing $a_3$ to pay a tax of 3. In the general case, the state that satisfies the single most valuable private goal will be chosen, unless there is a state that fully satisfies more than one goal. This approach suffers from the fact that an agent cannot assign relative weights to the alternatives, and no mutual compromise can be achieved.

A more flexible approach ("partial satisfaction") is for the agent to give each state a worth that represents the portion of the agent's goal that the state satisfies, i.e., which predicates in the agent's composite goal are satisfied in the state. Assume that each of the agents' goal predicates contributes equally to the worth associated with a state. In the example, $s_4$ is again chosen, but $a_3$ pays a tax of only 1.5. This approach is superior in the sense that compromise can be achieved via a state that partially satisfies a group of different goals. But in addition to preventing the agent from ranking bad alternatives (since there are no negative valuations), the method can be misleading. Consider, for example, $a_2$. His evaluation of $s_1$ (1.5) is based on the fact that $s_1$ satisfies $On(5,2)$, while any attempt to achieve his *other* subgoal, $On(2,4)$, will require the violation of this predicate.

Yet a third approach ("future cost") is to evaluate a state by taking into consideration the cost of the agent's eventually achieving his full goal, given that state: $w_i(k) = w_i(g_i) - C_i(s_k \rightsquigarrow g_i)$. Consider $a_3$ calculating the worth of $s_1$. Given $s_0$, he could achieve his goal using four $Move$ operations; our assumption is thus that his goal's value is 4. Given $s_1$, however, he would need five $Move$ operations, $Move(5,d)$, $Move(4,5)$, $Move(3,4)$, $Move(2,c)$ and $Move(3,2)$. He is therefore "worse off" by 1, and gives $s_1$ a worth of $-1$. In the example in Figure 3, this yields the following true worths for each agent: $\langle 2,0,1,0,-2,2\rangle$, $\langle 0,3,2,1,1,0\rangle$, $\langle -1,2,3,4,1,1\rangle$. $s_3$ (which is only one $Move$ operation distant from *all* the agents' goals) is chosen, and no tax is collected.

The second and third approaches above allow greater compromise than does the first, yielding states that are guaranteed to have greater social welfare. However, they may not improve the outcome for individual agents, who therefore might prefer to operate according to the more rigid first approach.

In some sense, this last method guarantees a "fair" consensus (where all agents are approximately equally distant from their ultimate goals). If it is important that some agent's goal be fully satisfied, a coin can be tossed to determine which of the agents will continue and fulfill his complete goal. Given a distribution of labor, the utility of an agent using this scheme may be greater than it would be if we had a lottery to select one agent, then let that agent bring about his own goal.[6]

However, the valuation of a state based on its closeness to a desired final state is obviously domain dependent. In some situations, there may really exist the possibility of arriving in the final state at some future time; in other situations, the value of a state may really be related to its proximity to some desired final state, even if the final state is not ever achieved (a chip manufacturer might be increasingly pleased, the fewer faults appear in his chips, even if he can never reach the ultimate desired situation where there are no faults). However, this is clearly an inappropriate technique in some domains, where intermediate states, viewed as final, have worths unrelated to their distance from the goal.

## 3.2 "Progressive" Worth Functions

Let's consider a slightly more sophisticated way of evaluating the worth of arbitrary states. Since an agent may have to contribute to the social effort of reaching any particular (intermediate) state, it also makes sense to take into account his share of the initial work, and to decrease his evaluation of worth appropriately.

A straightforward worth function for an arbitrary state $s$ might be built as follows: take the worth of a goal state (assumed to be available), subtract the cost of the single-agent plan to get from $s$ to the goal, then subtract the agent's share of the cost of the multiagent plan to get from start state $s_0$ to $s$. This gives us the worth of $s$: $w_i(s) = w_i(g_i) - C(i, s \rightsquigarrow g_i) - C_i(s_0 \rightsquigarrow s)$.

Note, however, that using the above equation the worth of $s_0$ for an agent would simply be the worth of his goal, minus the cost of his one-agent plan to reach that goal (in a one-agent scenario this would be true for every state). Thus, since the evaluation function does not capture the notion of progress in the plan, a rational agent would have no motivation to carry out his plan at all.[7]

There are several ways to refine the worth function so as to solve this problem. One way is by making the "future-cost" (i.e., $w_i(g_i) - C(i, s \rightsquigarrow g_i)$) more sensitive to the progress of the plan (that is, weighted so that states far away from the goal are assigned proportionately less worth). A simple approach is to take into consideration only that fraction of the goal's

[6] See [93] for an example of a similar scenario. Two agents agree to cooperate to an intermediate state that satisfies neither, then flip a coin to see who, alone, continues to his own goal. There, the cooperation is brought about by negotiation instead of voting.

[7] This is reminiscent of the Little Nell planning paradox, where an agent solves his problem by deriving a suitable plan, but has no need to carry it out once he identifies that it indeed solves his problem [58].

worth that reflects the amount of work already done to achieve it ($\approx w_i(g_i) \times [l(P(s_0 \rightsquigarrow s))/l(P(s \rightsquigarrow g_i))$. Another way is to give greater weight to the cost of operators that are located further along in the plan ($\approx w_i(g_i) - \sum_1^k k \times C(op_k)$). Or, assuming that each operator has a probability ($pr(op_k)$) associated with its success, we could use $\approx (\prod_1^k pr(op_k) \times w_i(g_i)) - C(i, s \rightsquigarrow g_i)$. These evaluations may be further refined by having costs or probability of success associated with the constraints that enable a plan (see [42] and [47] for richer probabilistic approaches).

There is an important restriction that we would like to place on the class of worth functions used by our agents. In Section 5, we will be introducing a pruning method that allows multiple agents to cut off search for optimal states when the group's global utility function decreases during the search. Therefore, we require that there be no local minima in the *global* utility function, or that the maximal depth of such a minima be known in advance. Formally, we use the following definitions:

- A set of worth functions will be called *strictly progressive* if between any two states $s_i$ and $s_k$, such that $U(s_k) \geq U(s_i)$, there exists no state, $s_j$, such that $U(s_j) < U(s_i)$.
- Similarly, let $\delta$ be the maximal gap bound between any local maximum of $U$ and any local minimum that follows it, of *any* given set of worth functions. We then say that the set is $\delta$-progressive.

Under many conditions, the above worth functions are strictly progressive. For example, when we use the "future cost" technique, it would be sufficient to assume that $\forall i C(s_1 \rightsquigarrow s_2) \leq C(i, s_1 \rightsquigarrow s_2)$). The decision procedure we propose in Section 5 is most suitable for progressive worth functions.

# 4 Centralized Generation of Alternatives

The selection of the candidate states (among which the agents will vote) plays a crucial role in the voting process.[8] Given a group of agents with fixed goals, choosing different candidates can result in wildly different outcomes. The question thus arises of how these candidate states are to be generated. It is desirable that this generation process be a function of the agents' goals. In this section, we describe a heuristic approach to generate alternative consensus states (i.e., candidate states) by a central planner, given the agents' goals. In Section 5, we introduce an iterative multiagent agent planning algorithm to derive such states in a way that requires the agents to reveal only the minimal necessary information about their private goals.

The generation of candidate states should aspire to choosing states maximal with respect to the satisfaction of agents' goals. Let $\mathcal{P}_{\mathcal{A}}^{U} = \cup_{a_i \in \mathcal{A}}(g_i)$ be the set of all the predicates appearing in all the agents' goals. Usually this does not specify a real-world state, since in

[8] The agents cannot vote over all possible states, as the large number of such states would make the voting process intractable.

the general case there are contradictory predicates among different agents' goals (otherwise, this state is guaranteed to be chosen).

We want it to be the case that each $s_k$ in the set of candidate states satisfies the following definition: $s_k = \{p \mid p \in \mathcal{P}^U_{\mathcal{A}}$ and $p$ is consistent with $s_k\}$. Thus, each $s_k$ is a maximal feasible subset of $\mathcal{P}^U_{\mathcal{A}}$, a fixed point with respect to the predicates' consistency.

To check consistency, we assume a set of axioms over the domain predicates by which inconsistency can be discovered. In the example above we might have
$On(Obj, t) \Rightarrow At(Obj, t)$
$[At(Obj_1, t) \wedge On(Obj_2, Obj_1)] \Rightarrow At(Obj_2, t)$
$[At(Obj, t_1) \wedge At(Obj, t_2) \wedge (t_1 \neq t_2)] \Rightarrow False$
to establish the inconsistency of a set such as $\{At(2, b), On(2, 5), At(5, c)\}$.[9]

Note that this process of generation has several features. First, the procedure guarantees for each $i$ the existence of at least one $s_k$ such that $g_i \subseteq s_k$. Second, each agent is motivated to hand the generator his true goal. Declaring $\tilde{g}_i \supset g_i$ might prevent the generation of compromise states that benefit $a_i$, or cause the generation of states preferable to other agents (resulting in the selection of a worse alternative than otherwise would have been chosen). Declaring $\{\tilde{g}_i \mid (\tilde{g}_i \cap g_i) \subset g_i$ or $(\tilde{g}_i \cap g_i) = \emptyset\}$ may prevent the generation of any $s_k$ that satisfies $g_i$, as well as preventing the generation of other states preferred by $a_i$ which otherwise could have been chosen. In either case, $a_i$ cannot hope to improve on his utility.[10]

Note that the phase of candidate generation is completely distinct from the Clarke Tax voting phase that follows it. An agent could declare goals that are used in generating candidates, and then vote in ways that contradict its declared desires. Note also that the technique above assumes the collection of information regarding agents' goals in a central location. This, of course, may be undesirable in a distributed system because of bottlenecks and communication overhead. In Section 5, we suggest several techniques for distributing the generation of alternatives among agents.

## 4.1 Additional Criteria in Candidate Generation

Candidate state generation can be refined by taking into consideration several additional criteria that avoid dominated states. These additional criteria are sometimes related to the approach agents will use to evaluate the worth of candidate states.

First, the generator can exclude states $\tilde{s}_k$ such that $\exists s_k[\forall i(w_i(k) \geq w_i(\tilde{k})) \wedge (C(s_0 \rightsquigarrow s_k) < C(s_0 \rightsquigarrow \tilde{s}_k))]$. In other words, the generator can exclude a candidate state if there

[9] Of course, it is not clear how such axioms should be generated [16, 32, 33]. In any case, the use of such axioms to check consistency produces an NP-hard problem, unless the database consists of Horn clauses. It seems more reasonable to consider the use of such consistency axioms as an idealized mechanism, and any realistic system would have to make do with heuristic approximations.

[10] This contrasts with cases where an agent can benefit from misrepresenting his goals, as in [71]. The key difference here is that agents will eventually engage in a non-manipulable voting process, namely one based on the Clarke Tax mechanism, which eliminates the possibility of useful deception in the candidate generation phase as well.

is another of equivalent value that is easier to reach. In the example in Figure 3, this test causes the elimination of the state $\{At(3,c),\ At(4,b),\ On(2,4),\ On(5,2)\}$ in favor of $s_2$.

If the agents are going to evaluate candidate states using the "partial satisfaction" criterion, the generator can exclude $\tilde{s}_k$ such that $\exists s_k[(\tilde{s}_k \cap \mathcal{P}^U_\mathcal{A}) \subset (s_k \cap \mathcal{P}^U_\mathcal{A})]$. In other words, the generator will exclude a candidate that specifies states that are a superset of another candidate's states. In the example, this would exclude $s_3$ in favor of $s_2$.

If the agents are going to evaluate candidate states using the "future cost" criterion, the generator can eliminate states $\tilde{s}_k$ such that $\exists s_k[\forall i(C_i(s_k \rightsquigarrow g_i) \leq C_i(\tilde{s}_k \rightsquigarrow g_i)) \wedge \exists i(C_i(s_k \rightsquigarrow g_i) < C_i(\tilde{s}_k \rightsquigarrow g_i))]$. In other words, the generator can exclude a candidate that, for all agents, is "more expensive" than another candidate.[11] In the example, such a test would eliminate the state $s_5$ in favor of $s_4$.

One might suppose that if it is known ahead of time how candidate states are going to be evaluated, the Clarke Tax voting phase itself becomes redundant. By extension of elimination procedures such as those above, the generator could just compute the optimal state. For instance, using the "future cost" criterion, it might directly generate the $s_k$ that minimizes $\sum_i^n C_i(s_k \rightsquigarrow g_i)$, and using the "partial satisfaction" criterion, it might directly choose the $s_k$ that is the maximal (with respect to number of predicates) consistent subset of $\mathcal{P}^U_\mathcal{A}$.

However, such extensions to the generation method are not always desirable. If the state generator uses them, the agents will sometimes be motivated to declare false goals. For example, if $a_1$ declares his goal to be $\{At(4,c), At(2,b), On(4,3), On(5,2)\}$ (whose predicates are a superset of his original goal), $s_1$ becomes dominant over all the other states if the generator uses either of the two global extensions considered above. Thus $s_1$ would automatically be chosen, and $a_1$ achieves a higher utility by lying. The two-phase method we propose, where candidates are generated and then agents vote using a Clarke Tax mechanism, eliminates such undesirable effects.

# 5 Dynamic Search for Alternatives as Multiagent Planning

The central generation of alternative consensus states suffers from two significant drawbacks. First, the entire set of alternative states is to be calculated prior to the voting procedure and therefore all agents have to reveal their goals before the vote takes place. Second, the participating agents are expected to declare their exact and entire set of utilities and preferences over this set of alternatives. Indeed, in the original economic scenario, one objective of the mechanism was to let the "planner" know the complete set of preferences. The designers of automated agents might have an interest, however, in reaching the right decision, but ensuring agents' privacy as much as possible; at the same time, they would like the agents to generate only alternatives that are likely to be in the consensus (for reasons of

[11] Actually, more or equally expensive for all agents, and more expensive for at least one.

efficiency).

To accomplish both these aims, we can make repetitive use of votes that employ the Clarke Tax mechanism and converge to a consensus decision. In the following sections, we present a novel iterative voting procedure, based on the original Clarke Tax mechanism, that enables agents to reach a decision of maximal social utility with only partial information.

We employ a dynamic search by a group of agents for all states that maximize their social utility. At each step, the agents vote (using the CTm) about the *next* joint action to be taken (i.e., what the next state will be for the group). This technique has several desirable properties: agents need not fully reveal their preferences, and the set of alternative final states need not be generated in advance of a vote.

In effect, the voting procedure gives rise to a new multiagent planning technique. Through a process of group constraint aggregation, the agents iteratively converge to a plan that brings the group to a state maximizing social welfare. Another advantage of the process is that it has lower complexity than the original centralized process from the previous section.

## 5.1 Notation

We use the following definitions in the description of our planning procedure and the embedded stepwise CTm:

- At each step of the process, additional alternative states are generated. $\mathcal{S}^k$ denotes the set of states that have been generated up to the $k$-th iteration of the process.

- The function $v_i : \mathcal{S}^k \rightarrow \mathbb{R}$, returns the *true* normalized worth (to $a_i$) of each state ($\forall i v_i(s) = w_i(s) - w_i(s_0)$). The function $d_i^k(j)$ returns the *declared* worth of state $s_j$ by agent $a_i$ at step $k$. $\vec{d_i^k}$ denotes the vector $\langle d_i^k(1), d_i^k(2), \ldots, d_i^k(m)\rangle$, the agent's declared worth over all alternatives.

- The profile of preferences declared by all agents at step $k$ is denoted by $D_n^k$, where $D_{\neg i}^k$ denotes this set excluding $i$'s preferences, such that $D_n^k = (D_{\neg i}^k, \vec{d_i^k})$.

- The choice function $f : D_n^k \rightarrow \mathcal{S}^k$ returns the state that is the *maximizer* of $\sum_{i=1}^n d_i^k(s)$, where $s$ is an element of $\mathcal{S}^k$.

- The tax imposed on $i$ at step $k$ is $t_i^k(f(D_n^k)) = \sum_{j \neq i} d_j^k(f(D_{\neg i}^k)) - \sum_{j \neq i} d_j^k(f(D_{\neg i}^k, \vec{d_i^k}))$, if this value is positive. Otherwise, $t_i^k$ will be zero. Therefore, the utility $u_i^k(f(D_n^k))$ of agent $i$ with respect to the chosen alternative is $w_i(f(D_n^k)) - t_i^k(f(D_n^k))$.

- $\mathcal{I}(P)$ is the set of absolutely necessary constraints needed for the plan $P$ to succeed. In accordance with the partial order over these constraints,[12] we divide $\mathcal{I}(P)$ into subsets

[12]Constraints are temporally ordered sets of the domain's predicates associated with the appropriate limitations on their codesignation [5]. In a STRIPS-like planning system, these "constraints" are actually the preconditions required by operators.

of constraints. Each such subset within $\mathcal{I}(P)$ comprises all the constraints that can be satisfied within $j$ (optimal) steps, and are necessary at some subsequent step after $j$.

We denote $\mathcal{I}(P)$'s components by $\cup_j I^j(P)$, such that $I^j$ includes all the constraints that can be satisfied at the $j$th step of the plan, and are necessary at some step $\geq j$. For any $j \geq l(P)$, we define $I^j$ to be the goal achieved by $P$.

- A constraint $I \in I^j$ is said to be *temporary* if later in the plan there is a constraint $\bar{I} \in I^k$ (where $k > j$) that denies it ($I \wedge \bar{I} \models False$). We say that a set of apparently conflicting constraints $G$ is semi-consistent ($G \not\models_{tmp} False$) if the removal of temporary constraints makes it consistent (see Section 5.4 for an example). A *virtual state* is a state that is specified by a semi-consistent set of predicates. In such a state, semi-consistent predicates may co-exist. Given a virtual state $v$ we define $r(v)$ to be the set of real states that it maps onto, that is, all maximally consistent subsets of the predicates in $v$. Note that for any real state $s$, $r(s) = s$.

- $P(G)$ denotes the set of the cheapest "grounded" plans (what Chapman calls "complete" plans [5]) of the temporally ordered set of constraints $G$. Each such plan results in a certain real or virtual state. We signify the set of all these states (the states induced by $G$) by $s(G)$ (we are only concerned with the cheapest states that satisfy $G$, in other words, $s(G) = \{s \mid s \models G \wedge C(s_0 \rightsquigarrow s) = \min_{k \models G} C(s_0 \rightsquigarrow k)\}$ ). $f_{ollow}(G)$ is defined to be the set of constraints that can be satisfied by invoking at most one operator on any state in $s(G)$ ($f_{ollow}(G) = \{I \mid \exists op \exists P[op(P(G)) \models I]\}$).

## 5.2 General Overview of the Search

The algorithm, presented in this section, allows a group of agents to find the state that maximizes their social welfare. The underlying idea is the *dynamic generation of alternatives* to locate the most desirable state for the society. At each step, all agents reveal additional information about their private goals. The current set of candidate states is then expanded and (possibly) pruned to comprise the new set of candidate states. The process continues until no new states are generated with higher social utility.

The entire process is illustrated in Figure 4. As represented by gray arrows in the upper part of the figure, each agent has a private set of constraints. Given the initial state $s_0$, by incrementally satisfying this set, its private goal will be achieved. These private sets of constraints are aggregated iteratively. At each step, each agent may try to impose more of its private constraints on the group's decision. This is done by declaring alternative feasible private extensions to all "live" sets of aggregated constraints.

**Figure 4**

The search for the global plan is through the space of states. A partial plan in the queue of alternative plans is identified by the states that it induces. Given some specific initial state, the agents will go through the following loop until they derive all the plans that reach states with maximal social welfare.

**Until** all maximal social welfare states have been found, do:

1. Identify all the promising successor states that can be reached from the first path in the queue. Each successor state represents a feasible set of constraints (propositions) that can be satisfied given that path. The relevant sets are determined by aggregating the sets of constraints that each agent may declare, based on its individual plan.
2. Determine the (fraction of the global) multiagent plan that achieves each successor state.
3. Based on the individual plans and the actual path that leads to each successor, determine the heuristic value of each newly formed path.
4. Add new successors into the queue according to their heuristic values.

The search procedure, to proceed correctly, needs an accurate value for the social utility of each candidate state (i.e., the search space of alternatives is pruned dynamically by the social welfare criterion). To provide this value, the agents vote over the set of candidates at each step. The Clarke Tax mechanism is then used at the end of the procedure, to ensure that agents' votes throughout will be honest.

The search method is in effect a parallel $A^*$ search. For example, the value of a state (its social utility) its taken to be the sum of individual estimates of the distance to the goal state (the $h'$ component) and the actual cost of achieving it (the $g$ component). But parallel searches are carried out simultaneously in each promising direction (wherever social utility does not decrease). When the search encounters a direction where social utility decreases, the search is terminated (that path is pruned). This method is illustrated in the lower part of Figure 4: each set of aggregated constraints has corresponding (real or virtual) states that it induces. Starting at the initial state, $s_0$, at each step of the process, all "live" descendents are considered. Then their possible successors are generated. A state that reduces the social utility (by more than $\delta$) in comparison to its parent, is pruned.

The agents are participating in many intermediate votes. Since each intermediate vote is only over a subset of candidates, there is the possibility that an agent will "shift" his vote by a constant, keeping a single round's preferences accurate while undermining inter-vote comparisons. For example, an agent voting over $s_0$ and $s_1$ might give the first a value of 0 and the second a value of 5. If that agent subsequently voted over $s_2$ and $s_3$, he might assign the first 0 and the second 8—but if he had voted over all four alternatives at once, $s_2$ and $s_3$ would have been shifted up by 3, giving votes of 0, 5, 3, and 11 to the candidates. The Clarke tax which is levied at the end of the procedure requires that artificial shifting does not occur. Therefore, we will require that all votes be relative to some "benchmark": we include $s_0$ in the set of alternatives at every step, and require that each agent give worth 0 to that state. If each agent gives his true preferences to the other states relative to $s_0$ ($v_i(s)$), then the score of each state $s$ in the vote is exactly $U^0(s)$.

## 5.3 The Algorithm

This section describes the algorithm in more detail, along with a running example. At each step of the procedure, agents try to impose more of their private constraints on the group's aggregated set of sets of constraints. Since agents want to maximize their own utility, they will impose as many constraints as they can at each step.

As an example, assume a simple scenario of the slotted blocks world as described in Figure 5 (we use the same operators and predicates as described in Section 2.2). There are three blocks (1,2,3) and two agents ($a_1, a_2$). The initial state is $\{On(1,b), On(3,1), On(2,c)\}$. The agent's goals are (respectively) $g_1 = \{On(1,2)\}$ and $g_2 = \{On(2,3)\}$.[13]

**Figure 5**

The set of all unpruned sets of constraints at step $k$ is denoted by $\mathcal{G}^k$ (its constituent sets will be denoted by $G_j^k$, where $j$ is simply an index over those sets). $\mathcal{G}'^k$ denotes the set $\mathcal{G}^k$ before it is pruned. The exact procedure is defined as follows:

**Initialize:** At step 0 each agent $i$ finds $\mathcal{I}_i = \mathcal{I}(P^*(i, s_0 \rightsquigarrow g_i))$—the temporally ordered sets of constraints imposed by any optimal skeleton plan that achieves the goal $g_i$. The virtual

[13] This is reminiscent of Sussman's Anomaly in the single-agent planning scenario—where the plan to achieve one sub-goal obstructs the plan that achieves the other [85]. Although the final state that satisfies both agents' goals in our example costs 9 to reach, and is therefore relatively expensive, it is the state that meets the social welfare criterion.

set of alternatives is initialized to be the empty set, and its induced set of states is initialized to be $s_0$ ($\mathcal{G}^0 = \emptyset$ and $\mathcal{S}^0 = \{s_0\}$).

In our example, we have:[14] $\mathcal{I}_1 = \{[\mathtt{C}(2)] \cup [\mathtt{C}(2), C(1)] \cup [O(1,2)]\}$ (this ordered set induces the plan $\langle M(3,\bar{2}), M(1,2)\rangle$) and $\mathcal{I}_2 = \{[\mathtt{C}(3), C(2)] \cup [O(2,3)]\}$ (inducing the plan $\langle M(2,3)\rangle$). Note that $\mathtt{C}(2)$ and $\mathtt{C}(3)$ are temporary since they are denied later by the induced plan.

1. **Loop to Step 5:**
   At step $k$ each agent may declare $I_i^k \subseteq I_i^{j+1}$ only if he has already fully declared $I_i^j$ (for any j) and the declaration is "feasible," i.e., it is possible to reach a state that satisfies those constraints: $\exists G[(G \in \mathcal{G}^k) \wedge (\bigcup_{n=1}^{j} I_i^j \subseteq G) \wedge (I_i^k \subseteq f_{ollow}(G))]$.

   $i$ is allowed to try to impose elements of his "next" private subset of constraints on the group decision only if they are still relevant and his previous constraints were accepted by the group. Notice that it is in $i$'s best interest to give a true declaration, since bringing about $I_i^{j+1}$ is useless without establishing $I_i^j$ first. Such insincere declarations can also be easily tracked by the other agents, since an insincerely declared constraint will not be used subsequently.

   At the first step, each agent $i$ will declare $I_{g_i}^1$. In our example, this will be $I_1^1 = [\mathtt{C}(2)]$ and $I_2^1 = [\mathtt{C}(3), C(2)]$.

   At the second step, $a_1$ declares $I_1^{G_1^1}$, which in this example is equal to $I_1^2 = [\mathtt{C}(2), C(1)]$. Similarly, $a_2$ declares $E_{g_2}^2 = [O(2,3)]$. (Both are in $F_{ollow}(\mathcal{A}^1)$, which contains only one subset.)

   At the third step, $a_1$ declares $[O(1,2)]$ and $a_2$ declares $[O(2,3)]$ (his final goal, which already appears in $\mathcal{G}^2$).

2. The set $\mathcal{N}(\mathcal{G}^k)$ of all possible extensions of "live" sets in $\mathcal{G}^k$ is set to be the union of all agents' declarations at step $k$, i.e., $\bigcup_i I_i^k$. The set $\mathcal{N}(\mathcal{G}^k)$ is not necessarily consistent. Then, for each set of constraints $G_j^k \in \mathcal{G}^k$, we generate all the maximally consistent or semi-consistent extensions of $G_j^k$ with elements of $\mathcal{N}(\mathcal{G}^k)$) (i.e., $\{G_j^k \cup \{I \mid (I \in \mathcal{N}(\mathcal{G}^k)) \wedge ((I \cup G_j^{k+1}) \not\models_{tmp} False)\}\}$). $\mathcal{G}'^{k+1}$ denotes the union of all these extensions.

   At the first step, the aggregated set of constraints is $\{\mathtt{C}(3), C(2)\}$.

   At the second step, both declarations may coexist consistently, and there is therefore only one successor to the previous set of constraints: $\{\mathtt{C}(2), C(1), O(2,3)\}$.

   At the third step, the aggregated set is $\{O(1,2), O(2,3)\}$, and it satisfies both agents' goals.

[14] We will use the first letter to denote the full operator predicate, and $\bar{y}$ to denote any location excluding $y$'s. We will use a typewriter font to denote temporary constraints.

3. At this stage, all extensions are evaluated to enable the pruning of sets that reduce social utility (by more than the bound $\delta$). The evaluation is done by first evaluating all the states that an extension induces:

   (a) Let $S$ be the set of all newly formed alternative states with respect to the new sets of constraints (i.e., $S = \{s \mid G \in \mathcal{G}'^{k+1} \wedge s \in r(s(G))\} \setminus \mathcal{S}^k$). $\mathcal{S}^{k+1}$ is set to be $\mathcal{S}^k \cup S$.

   (b) Each agent gives its vote to each state in $S \cup \{s_0\}$. The worth of a real state is simply the sum of individual worths given to that state. The worth of a virtual state $v$ in $\mathcal{G}'^{k+1}$ is computed as follows. For each state $v$ there is a set of real states that it maps onto, namely $r(v)$. For each agent and virtual state, there is a real state with maximal worth. The worth of the virtual state is taken to be the sum of these maximal worth states in $r(v)$, over all agents ($\sum_i \max_{s \in r(v)} w_i(s)$).

   (c) The normalized social utility ($U^0(s)$) of each state is found by subtracting its cost $C(s_0 \rightsquigarrow s)$ from its worth. The utility of the set of constraints $G$ is defined to be highest utility given to any of the states that it "induces" ($U^0(G) = \max_{s \in s(G)} U^0(s)$).

4. The next set of sets of constraints is pruned to contain only sets that do not decrease, by more than $\delta$, the social utility with respect to the set from which they were formed. Formally, $\mathcal{G}^{k+1} = \{G_j^{k+1} \mid G_j^{k+1} \in \mathcal{G}_j'^{k+1} \wedge [U^0(G_j^{k+1}) \geq U^0(G_j^k) - \delta]\}$.

   The first set of aggregated constraints is satisfied by the initial state, and thus induces the null plan (with cost of zero). The value given to that first set is the value that agents assign to the initial state.

   The second set can be achieved by the plan $\langle M(3,a)\rangle$, with cost of 3. In order for the set not to be pruned, it must be the case that the agents value the set by at least $3 - \delta$ more than the initial set.

   The third set is the set that satisfies the social welfare criterion (as we stated above), and therefore (by definition) will have higher value than previous steps (and not be pruned).

5. **Loop Termination Test:**
   The process ends when $\mathcal{G}^{k+1} = \mathcal{G}^k$ (and, thus, $\mathcal{S}^{k+1} = \mathcal{S}^k$). The votes given by each agent to each state in $\mathcal{S}^k$ are then combined using a CTm to determine $\hat{\mathcal{S}}^k$, the set of states that maximizes the social utility ($\hat{\mathcal{S}}^k = \{s \mid s \in \mathcal{S}^k \wedge \forall s^*[s^* \in \mathcal{S}^k \Rightarrow U^0(s^*) \leq U^0(s)]\}$). Among these states one is randomly chosen and each agent is fined the Clarke Tax with respect to this choice.

To realize the true social utility of each state, each agent has to reveal his true preferences during the vote at each step. The Clarke Tax makes this behavior dominant (i.e., preferable to any other behavior). As described in the last step (5) of the procedure above, in order to choose the *final* consensus state, all the votes are gathered together in a Clarke Tax vote.

The state to be chosen must be a real (not virtual) state; therefore, virtual states may not be included in the final vote, and thus not in any of the intermediate votes that determine it. For example, an agent might find incentives to manipulate its vote on a virtual state, knowing that this vote would not be subject to the Clarke Tax. For that reason, the virtual states are treated in a special way (step 3b above).

Note that since $s_0$ is included in each $\mathcal{S}^k$, each agent must give the same bid (i.e., 0) for $s_0$, and thus his true preferences are always (at any step) bounded to reflect the worth of the other states in comparison to $w_i(s_0)$ (see Section B.2). For that same reason, agents actually have to determine only the worth of the newly formed states (step 3a of the procedure). Given these considerations, the following lemma proves that honesty is the dominant strategy. The theorem that follows then states some of the desirable attributes of the procedure.

**Lemma 1** *At* any *step $k$ of the procedure, $i$'s best strategy is to vote over the alternatives at that step ($\mathcal{S}^k$) according to his* true *normalized preferences $\vec{v}_i$ (that is, subtracting $w_i(s_0)$ from $w_i(s)$ for any $s$).*

**Proof:** All proofs of lemmas and theorems in this article appear in Appendix A.

The lemma above demonstrates a subtle consequence of the "independence of irrelevant alternatives" property of the original CTm. An agent cannot change the winning alternative simply by causing other alternatives to be pruned. Moreover, directly causing another alternative to win (either by overbidding that alternative, or underbidding the rightful winner) will expose the agent to the Clarke Tax.

**Theorem 1** *Given any set of $\delta$-progressive individual worth functions, and the corresponding $\delta$, this mechanism finds all states that satisfy the Social Welfare Criterion. Consensus will be reached after $O(\max_{s \in SW(\mathcal{S})} l(P(s_0 \rightsquigarrow s)))$ steps (the order of the length of the plan that derives the most distant state that answers the Social Welfare Criterion, from the initial state).*

Note that the search is intended to find all states that satisfy the Social Welfare Criterion. If we were to interrupt the search after the first such state were found, we would not necessarily have found the shortest path to such a state (i.e., our $A^*$ search is not admissible with respect to the length of the path).

In comparison with the classic CTm, the procedure has the following advantages: **(a)** agents are required to submit only the minimally "conflict-sufficient" information about preferences over alternatives, thus maintaining a certain amount of privacy; **(b)** preferences will be calculated and submitted only for "feasible" alternatives (preferences over infeasible alternatives need not be revealed); **(c)** alternatives are generated by the *entire* voting group *dynamically* (making the procedure more distributed, and computationally more tractable).

The search process is also superior to the central generation of alternatives for several additional reasons: **(a)** conflicts and "positive" interactions are addressed within a unified framework; **(b)** the branching factor of the search space is strictly constrained by the individual plans' constraints (the kind of "cone" that is shown in the upper part of Figure 4); **(c)** the $A^*$-type algorithm uses a relatively good heuristic function, because it is derived "bottom-up" from the plans that the agents have already generated (not simply an artificial $h'$ function); **(d)** generation of successors in the search tree is split up among the agents (each doing a part of the search for a successor).

## 5.4 Specification and Aggregation of Constraints

There are several important aspects of, and requirements for, the procedure above:

- The procedure's success depends on each agent identifying and declaring only the absolutely necessary constraints needed for its individual plan to succeed.
- It is necessary that *temporary* conflicts among the agents' plans not cause deadlocks (that is, if there is a temporal order that can later resolve them). This phenomenon is achieved in our procedure by allowing the existence of semi-consistent sets of constraints and then voting on the virtual states that they induce.
- The generation of consensus sets of constraints is based on the aggregation of individual sets of constraints. This is not always trivial to do.

Figure 6 shows three simple scenarios of the slotted blocks world that illustrate the above issues. In these examples, two agents can achieve a consensus state that fully satisfies both agent's goals. There are three slots, and some blocks in each world. Only the *Move* operator is available and it costs 2.

**Figure 6**

### 5.4.1 Specification of Constraints

An important requirement for the success of the procedure is that each agent identify and declare only the absolutely necessary constraints needed for its individual plan to succeed. As an example consider the first scenario in Figure 6. Two agents want to achieve the following goals: $g_1 = \{A(1,a), C(1)\}$ and $g_2 = \{O(3,2), A(2,b)\}$. These two goals can coexist, as shown in the final goal state. To achieve his goal $a_1$ need not take any action.

$a_2$, on the other hand, has to follow $\langle M(2,x), M(3,\bar{2}), M(2,b), M(3,b)\rangle$.[15] The only way for the second step of this plan to be completed is by stacking either block 3 or block 2 onto block 1. Although neither of these operations has to do with $a_2$'s final goal, by including either of these alternatives as one of its constraints, $a_2$ would encounter $a_1$'s opposition, thus (perhaps) preventing his own goal from being achieved.

However, by realizing that the purpose of the second *Move* operator is just to achieve $A(3,\bar{b})$, the danger of conflict may be avoided. Thus, if $a_2$ declares his set of constraints to be $[C(2)] \cup [C(2), C(3)] \cup [C(2), C(3), A(3,\bar{b})] \cup [C(2), C(3), A(2,b)] \cup [A(2,b), O(3,2)]$, no conflict arises. At any step, $a_1$'s declaration equals his full goal's description. Therefore, at the third step of the procedure, the aggregated set of constraints would become $[C(2), C(3), A(3,\bar{b}), C(1), A(1,a)]$, which is fully satisfied by the state $\{A(1,a), A(2,b), A(3,c)\}$. This state is one step distant from the goal state. The described goal state will thus be generated at the next step. Note that $A(2,\bar{b})$, which $a_2$'s plan achieves after the first step, is not required for further progress, and therefore should not be included in the set of constraints at all. Conflict was avoided because the duration of the temporary conflict between the two agents' plans was short enough to be resolved by invoking one more operator.

### 5.4.2 Aggregation of Temporary Constraints

Unfortunately, this is not always the case. In the second scenario, $a_1$'s goal is the same, but $a_2$'s goal is now $\{O(5,4), O(3,2), A(2,b), A(4,b)\}$. As in the previous example, there is no way for him to achieve his goal without trying to temporarily violate $a_1$'s goal. But in contrast to that example, there is no way to specify $a_2$'s constraints in a way that would hide the conflict. Here, for example, $A(2,\bar{b})$, although temporary, is crucial for $a_2$'s plan's success. Were the process to consider sets of constraints to be relevant only as long as they were consistent, the process might stop after two steps, avoiding the goal state from even being considered.

It is therefore necessary that *temporary* conflicts between agents' plans not cause deadlocks (that is, if there is a temporal order that can later resolve them). This phenomenon is achieved by allowing the existence of semi-consistent sets of constraints, and allowing voting over the virtual states that they induce. For that reason, agents should recognize what their temporary constraints are (terms that will be violated by their own future actions). Identifying the temporary constraints of his own plan, $a_2$'s set of constraints ($I_2$) would then become:[16]
$[\mathtt{C(2)}] \cup [\mathtt{C(2)}, \mathtt{A(2,\bar{b})}] \cup [\mathtt{A(2,\bar{b})}, \mathtt{C(4)}, \mathtt{A(3,\bar{2}\bar{b})}] \cup [\mathtt{A(2,\bar{b})}, \mathtt{C(4)}, \mathtt{A(3,\bar{2}\bar{b})}, \mathtt{C(5)}, \mathtt{A(4,\bar{b})}] \cup$
$[\mathtt{A(2,\bar{b})}, \mathtt{C(4)}, \mathtt{A(3,\bar{2}\bar{b})}, \mathtt{C(5)}, \mathtt{A(5,\bar{4}\bar{b})}] \cup [\mathtt{A(2,\bar{b})}, \mathtt{C(4)}, \mathtt{A(3,\bar{2}\bar{b})}, \mathtt{C(5)}, \mathtt{A(5,\bar{4}\bar{b})}, \mathtt{C(2)}, A(4,b)] \cup$
$[\mathtt{A(3,\bar{2}\bar{b})}, C(2), A(4,b), O(5,4), C(3)] \cup [C(2), A(4,b), O(5,4), C(3), A(2,b)] \cup$
$[A(4,b), O(5,4), O(3,2), A(2,b)]$.

As can be seen in this specification, all the constraints that contradict $a_1$'s goal are temporary. Therefore, even though $a_2$'s plan actually violates $a_1$'s goal, the mutual goal

[15] The first letter denotes the full operator predicate, and $\bar{y}$ denotes any location excluding $y$'s.

[16] Remember that we use typewriter style to denote temporary constraints.

state is reachable. Since all first eight sets of combined constraints (in the first eight steps) are semi-consistent, they are taken into account and voted on. In addition, since all the induced virtual states are of increasing value ($a_1$ is always satisfied with his goal not being violated, and $a_2$ gets closer to his own goal), the sets are not pruned. Thus, the process continues, leading in the ninth step to the (real) mutual goal state.

### 5.4.3 Aggregation of Functional Constraints

The generation of consensus sets of constraints is based on the aggregation of the individual sets of constraints. As the third example in Figure 6 shows, this is not always trivial. Here, there is a ceiling that makes it impossible for more than two blocks to be stacked. This time $a_1$'s goal is $A(1, b)$ while $g_2 = A(4, b)$. We assume that the function $fs(b)$ returns the available free space (in number of blocks) left at slot $b$. As in any scenario, $s_0$ satisfies $I_i^1$ of both agents.[17] Following the second step of their plans, each of the agents has as temporary constraints $\texttt{fs(b)} \geq \texttt{1}$ and $C(x)$, where $x$ is the block each wants to move onto slot $b$. These constraints enable each agent to move "his" block to slot $b$ after removing block 2.

The aggregation of these constraints requires careful analysis. First, it must be recognized when the aggregated constraints of two identical terms such as $fs(x) \geq 1$ will be $fs(x) \geq 2$, or simply $fs(x) \geq 1$. Had block 4 been located on block 1 in the initial state of our example, the first solution would be appropriate. Following it in the given scenario, however, would yield the aggregated set of constraints in the second step to be $[C(1), C(4), fs(b) \geq 2]$. The induced states of this set cost 10 move operators, while the actual plan that achieves the mutual goal costs only 5. The problem here is that both constraints are *temporary* (each agent needs a free space for one block only momentarily). Thus, in this case, the aggregated free space should stay 1, leading to a state which is only three *move* steps distant from the initial state. Unfortunately, it not clear how in general this subtle analysis is to be done (see [5] for further discussion on the difficulty of handling functional constraints).

### 5.4.4 Worth Evaluation of Sets of Constraints

The process also counts on the fact that the agents' worth functions increase monotonically with the number of satisfied constraints (within the maximal gap bound $\delta$). This requirement yields the non-decreasing value of the extended set of constraints. Thus, the pruning of unpromising states is enabled. However, the fact that the individual worth functions monotonically increase may sometimes lead to counter-intuitive results. Consider again the last example. Each agent can achieve his own goal by two $Move$ steps, whereas to achieve both goals it takes five. Therefore, as long as the cost of a multiagent $Move$ is not less than $\frac{4}{5}$ of a single agent $Move$, $s_0$ will be the optimal state using our social welfare criterion.

[17] $I_1^1 = [C(1), C(2), \texttt{fs(c)} \geq \texttt{1}]$ and $I_2^1 = [C(4), C(2), \texttt{fs(a)} \geq \texttt{1}]$. Note that by its nature the free space constraint is temporary, since it always is a precondition of an action that violates it!

## 5.5 Using the Procedure on Our Example

We now follow the iterative Clarke Tax procedure to solve the first problem presented in Section 2.2. We assume that the cost of reaching a state is divided equally among the agents (by side-payments if necessary), and that each agent $i$ uses the (strictly progressive) worth function $w_i(s) = w_i(g_i) - C(i, s \rightsquigarrow g_i)$. From the agents' individual plans, we get the following constraints. As before, the first letter denotes the full predicate, $\bar{y}$ denotes any location excluding $y$'s, and temporary constraints are written in typewriter style.

$I_1 = [C(2), C(4)](= I_1^1) \cup [C(4), A(2, b)](= I_1^2) \cup [A(4, c), At(2, b)](= I_1^3)$[18]

$I_2 = [\mathtt{C(2)}, \mathtt{C(4)}] \cup [\mathtt{C(4)}, \mathtt{C(2)}, C(5)] \cup [C(2), C(5), O(2, 4)] \cup [O(2, 4), O(5, 2)]$

$I_3 = [\mathtt{C(2)}] \cup [\mathtt{C(2)}, C(3)] \cup [C(2), C(3), \mathtt{A(3, \bar{c})}] \cup [\mathtt{C(2)}, C(3), A(2, c)] \cup [A(2, c), O(3, 2)]$

**Figure 7**

Figure 7 presents the induced states at each step (in this example, all the generated states are real). At the first step, each agent declares $I_i^1$. $\mathcal{G}^1$, the set that includes all possible consensus sets of constraints, has only one member: $G_1^1 = \{[C(2), C(4)]\}$. $s(G_1^1)$ also has only one member, $s_0$. Agents then vote on this state, and it receives a score of 6 (for example, $a_2$ goal's worth is 14, and to achieve it the agent would perform $\langle M(4, \bar{c}), M(2, 4), M(5, 2)\rangle$ at a cost of 12; therefore, he values $s_0$ as 2). The normalized value is (by definition) 0.

At the second step, each agent hands in $I_i^2$ (which is in $f_{ollow}(G_1^1)$ for each $i$). Since all these constraints coexist consistently, $\mathcal{G}^2 = G_1^1 \cup [C(2), C(4), C(5), C(3), A(2, b)]$. This set induces the single state $s_1$ ($= s(G_1^2)$) as described in Figure 7. Note that there are many other states that could satisfy this set of constraints, but $s_1$ has the *minimal cost*. This state can be achieved by $\langle Move(4, a), Move(2, 1)\rangle$; therefore, the state costs 6. Subtracting this cost from the normalized worth values given by each agent (4 in this case by all three agents), the state scores 6. Since this score is greater than that of the preceding state $s_0$, the process continues.

At the third step, the new added constraints generate 3 possible maximally consistent extensions: $G_1^3 = [A(2, b), A(4, c), C(2), C(5), C(3), A(3, \bar{c})]$ inducing $s_2$,
$G_2^3 = [O(2, 4), A(2, b), C(2), C(5), C(3), A(3, \bar{c})]$ inducing $s_3$,

[18] $a_1$'s plan is $\langle Move(2, 1), Move(4, 3)\rangle$. The first operation may be invoked if the constraint $C(2)$ is satisfied. This constraint is satisfied by $s_0$, and therefore is included in $I_1^1$. $C(4)$ is needed for a future operator, but it is also satisfied by $s_0$, and therefore it too is included in $I_1^1$. The first operator establishes the constraint $A(2, b)$, which is necessary at all future times for the plan to succeed, so it is included in any future set of constraints.

and $G_3^3$ =[$O(2,4), A(4,c), C(2), C(5)$,$C(3), A(3,\bar{c})$] inducing $s_4$ and $s_5$. These induced states respectively score 11, 3, 12, and 12; $G_2^3$, which decreases the social utility, is therefore pruned.

At the fourth step, the two remaining extensions are extended further; $a_1$ hands in $\{A(2,b), A(4,c)\}$ (which is in $f_{ollow}(G_1^3)$), $a_2$ hands in $\{O(2,4), O(5,2)\}$ and $a_3$ $\{C(3), C(2), A(2,c)\}$ (both in $f_{ollow}(G_3^3)$). These constraints yield six different extensions that again induce the states $s_2$, $s_3$, $s_4$, $s_5$ and the new states $s_6$ and $s_7$ (that score respectively $-5$ and 5). Therefore, only the sets that induce $s_4$ and $s_5$ can be further extended (by $a_3$) to induce $s_8$. Although $s_8$ fully satisfies $a_3$'s goal, it scores only 5 and the process ends.

Figure 8 shows the entire search tree.

**Figure 8**

All intermediate votes are now gathered for the final vote, which employs the Clarke mechanism as described in Table 9. The format of the table follows the one given in Section 2.4. Each row of the table shows several pieces of information regarding an agent. First, his preferences for each state are listed. Then, the total score that each state would have received, had the agent *not* voted, are listed. An asterisk marks the winning choice in each situation.

**Figure 9**

As can be seen from the table, both $s_4$ and $s_5$ maximize the social welfare utility (both are one operation distant from each of the agents' goals). Both $a_2$ and $a_3$ are taxed 2. $a_2$ improves its utility by 2 and $a_3$ by 6 ($a_1$'s utility is not improved with respect to $s_0$). The group's social utility is therefore improved by 8.

## 5.6 Power and Influence Among Heterogeneous Agents

Up to now, we have assumed that all agents have equal capabilities and use an identical evaluation function. This is, however, not the typical real-world situation. Assume, for example, that $a_1$ is an older model agent and it's harder for him to carry a block; thus the operator $Move(Obj_1, Obj_2)$ has cost for $a_1$ of 6 instead of 4. His worth assessment for $s_0$ will thus become 0, and his entire set of normalized preferences is changed to be $\langle 4,9,-9,2,2,-15,-5,-5 \rangle$. This vote causes the chosen state to be $s_2$ with a score of 15. $a_1$ pays a tax of 6 for a state that he values as 9, and thus his own utility is improved by 3 $(9 - 6)$; in the original scenario his utility was not improved at all.

On the other hand, $a_2$ and $a_3$ improve their utilities by 1 and 5 respectively, while in the original scenario their utilities were improved by 2 and 6 respectively. Interestingly, the overall social utility actually increases in this case, and although achieving his goal alone would leave him with 0 utility, $a_1$ gains from the group activity. Notice, however, that $a_1$ cannot gain any profit by *pretending* to be disabled since his *true* preferences stay as before. If he changed the outcome by insincere voting as above, he would end up paying a tax of 6 for a state that he really values at 5.

The same phenomenon may be caused for a totally different reason. $a_1$'s goal's worth is 12. However, this relatively high worth is not expressed in the worth assessment functions that we assumed to be used by the three agents (because the functions are normalized). $a_i$ would rather use the following evaluation function: $w_i(s) = -C_i(s_0 \rightsquigarrow s) + w_i(g_i) \times [l(P(a_i, s_0 \rightsquigarrow g_i))] / [l(P(a_i, s \rightsquigarrow g_i))]$. Using this function his vote will again be changed to be $\langle 4,9,-9,2,2,-15,-5,-5 \rangle$, yielding the same outcome as above. In fact, by using any worth assessment function that "spreads" the worth of the goal over progress towards its achievement, we have that the higher $w_i(g_i)$ is, the more agent $i$ influences the group.

Fortunately, the influence of an agent on the social decision may be easily controlled without losing the power of the mechanism as an effective preference revealer [35] (of course by assigning weights to agents we lose the possibility of anonymity). This control may be achieved by giving an influence weight $z_i (\in \mathbb{R}^+)$ to each agent $a_i$. Given $a_i$'s normalized preferences $\vec{v_i}$, each $v_i(s_k)$ is divided by $z_i$; only then is the choice function invoked. Then, in order to keep *truth* as the dominant strategy, the calculated tax (according to the weighted votes) is multiplied by $z_i$.

**Lemma 2 (due to I. J. Good)** *Even when influence weights are used along with agent preferences, it is still agent i's best strategy, at each step k of the procedure, to vote over the alternatives at that step ($\mathcal{S}^k$) according to his* true *normalized preferences $\vec{v_i}$ (that is, subtracting $w_i(s_0)$ from $w_i(s)$ for any s).*

As an example, consider again the vote in Table 9. If it is desired that $a_2$'s influence on the voting process be four times greater relative to the others ($z_2 = 4$), then instead of his original vote we get $\langle 8, 4, 20, 16, 16, 36, 28, -4 \rangle$. The score of the states will become $\langle 12, 14, 18, 24, 24, 22, 26, 2 \rangle$, which yields the selection of $s_7$ (that fully satisfies $a_2$'s goal). $a_2$'s apparent tax becomes 12, but he will actually be taxed 3 (= 12/4). $a_3$ will pay a tax of 2, and $a_1$ a tax of 0; the social utility will therefore decrease to 4. Setting $z_1$ to be $\frac{1}{3}$ can serve to solve the problem that was described in the first example of this section, where we might wish to perturb the influence that agents have on the decision procedure, such as when their capabilities (or the costs they associate with actions) are different.

## 5.7 Heuristic Pruning of the Search

Although superior to the centralized process, the dynamic search process from Section 5.3 is very complex. Our primary concern has been the finding of *all* states that may be in consensus, rather than the complexity of finding all of them. However, if we relax this demand, and are satisfied with only one state, or with states that are "close" to the actual consensus states, the complexity of the process may be reduced significantly.

One way is to guide the search by using the actual $A^*$ algorithm. With this approach, the evaluation of each alternative state will be as before. However, at each iterative step of extension in the algorithm (Step 1) we choose for further extension only the most promising state according to the evaluation function ($f' = g + h'$). Note that if the heuristic evaluation is accurate, then a consensus state will be found after at most $O(\max_{s \in SW(\mathcal{S})} l(P(s_0 \leadsto s)))$ steps, where at each step only one set of aggregated constraints is being extended. However, since subplans will tend to interfere with one another (both in "positive" ways [overlapping constraints], and in "negative" ways [conflicting constraints], similar to what are called positive and negative threats in POCL planning [57]) we must search further so as to overcome the misjudgment of the heuristic function, or be satisfied with the first state reached, which may be sub-optimal.

Another computationally expensive component of each step is the generation of the actual optimal plan that derives each intermediate state ($s$). Here again, we may relax the demand for optimality and consider the plan that is induced by the aggregated set to be the optimal one. Following this approach in the example shown in Section 5.5, the cost of deriving the final consensus state would become 21 instead of 15 (seven $Move$ operators instead of just five).

Another way of reducing complexity in determining the optimal plan at each intermediate state is to exploit the computational power of participating agents. With this approach, each agent is assigned a subgoal that corresponds to his "contribution" to the state in question $s^i$ (such that $\wedge_i s^i \models s$). The agents then follow a variant of the main algorithm so as to derive the plan that generates the state. In the example, the division into subgoals of the final state would become: $s^1 = \{A(4,c)\}$, $s^2 = \{A(2,c), O(2,4), C(5)\}$, $s^3 = \{A(3,c), O(3,2)\}$. Note that in this case, there is no need to consider semi-consistent sets of aggregated constraints (and their corresponding virtual states) since all subgoals are coherent. For that same reason, the heuristic evaluation is much more accurate.

Following this approach, the actual multiagent plan may be constructed during the process itself. The construction may be made at Step 1 of the algorithm. Each agent bids for each action that each extension implies. The bid is based on the sequence of $i$'s previous actions in the extended set. Thus, the minimal cost sequence may be determined (to ensure honest bidding it is possible to employ the Vickrey mechanism [90]).

Yet another way to prune the search process is to define a dominance relation over alternative extensions at each step, and ignore dominated extensions. An example might be to prefer an extension that satisfies in full the declaration of one agent, over one that only partially satisfies two different declarations. Or, given an aggregated set $A$, we may prefer its extension $E$ that induces a grounded constraint $I$ over another extension $\bar{E}$ that induces a grounded constraint $\bar{I}$ (such that $I \wedge \bar{I} \models F_{a}lse$) if $I \in A$ while $\bar{I} \notin A$ or if $\bar{I}$ appears in some agent's final goal, while $I$ does not (intuitively, we can in this way avoid directions in which constraints are added that are merely stumbling blocks). Following this heuristic in Example 5.5, states $s_3$ and $s_6$ would not be considered since they will be dominated by states $s_2$ and $s_7$ respectively. In many cases $\bar{E} \setminus \bar{I}$ would be preferable to $\bar{E}$ itself. Following Example 5.5 again, $s_1$ will contain $\{On(4,5)\}$ instead of $\{On(4,a), C(5)\}$, and the "premature" removal of block 4 will be avoided.

# 6 Partial Revelation of Preferences

One significant drawback of the CTm (as mentioned above), is the fact that the participating agents are expected to calculate and declare their exact and entire set of utilities and preferences over the set of alternatives. One of our interests, however, might be to reach the right decision, but ensure agents' privacy as much as possible. Below, we introduce iterative variations of the Clarke mechanism that allow agents to reach consensus while calculating and revealing only partial information about their preferences.

## 6.1 An Example

As an example, consider a group of four scheduling agents, attempting to establish a schedule of meetings for their owners. The four individuals whose interests the agents represent have varying preferences regarding possible schedules. For example, one of the individuals very much does not want to meet his supervisor (the individual is unprepared for the meeting). However, he would prefer that this preference remain private.

**Figure 10**

As shown in Figure 10, there are four possible alternative schedules. The agents have preferences over these alternatives. For example, $a_2$ mostly prefers schedule $s_2$, finds $s_1$ to be a relatively good choice, but is strongly against $s_3$ and $s_4$. Since the decision is to be taken via the Clarke Tax, it is in each agent's best interest to declare its true preferences. The table shows how, using the Clarke Tax, they settle on schedule $s_2$, with agent $a_2$ paying a tax of 3.

However, the mechanism has publically uncovered all of the agents' preferences. It may be the case that agents (or their controlling owners) wish to conceal their true evaluations of the various schedules. Although $a_3$ doesn't want the schedule $s_3$, for example, it doesn't mean that $a_3$ wants to publically admit that fact.

This section presents an iterative, dynamic variation of the original mechanism. The basic idea is that instead of a "one shot" voting procedure, the vote takes place as a sequence of steps. At each step, a voter can either change the group vote in favor of one alternative by a fixed amount $\delta$, or opt out of the process. This method manages to preserve the agents' privacy with regard to preferences as much as possible. In addition, if the group can come to a quick consensus, agents do not have to fully explore their preferences over other alternatives. The longer the decision takes to make, the more refined calculation is needed.

## 6.2 Notation

Here we summarize the notation that we will be using in the procedure specification below. Some of it parallels the notation presented above for the embedded stepwise Clarke Tax mechanism (Section 5.1).

- The function $w_i : \mathcal{S} \rightarrow \mathbb{R}$, returns the *true* worth of each alternative to $a_i$. $w_i(s_i^{min})$ is the worth of the state that $i$ values the least.
- $v_i^r(s)$ is the value $i$ adds to the state $s$ at round $r$; $d_i^r(s) = \sum_{j=1}^{r} v_i^j(s)$ is the total sum given so far by $i$ to alternative $s$. $\vec{d_i^r}$ denotes the vector $\langle d_i^r(1), d_i^r(2), \ldots, d_i^r(m)\rangle$. We denote by $\vec{d}[d_k \leftarrow d'_k]$ the vector that is created by replacing the $k$th element of the vector $\vec{d}$ by $d'_k$.
- The set of preferences declared by all agents at round $r$ is denoted by $D_n^r$, where $D_{\neg i}^r$ denotes this set excluding $i$'s preferences, such that $D_n^r = (D_{\neg i}^r, \vec{d_i^r})$.
- $\Sigma^r(s) = \sum_{i=1}^{n} d_i^r(s)$ is the total score of $s$ at step $r$.
- $\tilde{s}^r$ denotes the choice at round $r$ (i.e., $\tilde{s}$ is the *maximizer* of $\Sigma^r(s)$).
- The tax imposed on $i$ at round $r$ is $t_i^r(f(D_n^r)) = \sum_{j \neq i}^{n} d_j^r(f(D_{\neg i}^r)) - \sum_{j \neq i}^{n} d_j^r(f(D_{\neg i}^r, \vec{d_i^r}))$.
- $u_i^r(s) = w_i(s) - t_i^r(s)$ is the utility that $i$ gets from choice $s$ at round $r$ (where $t_i^r$ is the tax that is imposed on $i$ with respect to round $r$).

## 6.3 The Partial Revelation Voting Procedure

Imagine a voter who is being asked to help choose among three outcomes, A, B, and C. His true preferences can be represented by the vector $\langle 7, 0, 13\rangle$. However, the voting procedure is going to protect the secrecy of these preferences as much as possible. Prior to the first round, this agent's "stated" preferences are initialized as $\langle 0, 0, 0\rangle$. At each step, the agent can specify a positive increase in one or more elements in the vector, subject to the following restriction: he can attempt to change the current group choice to some other outcome by amount $\delta$, and in parallel increase other outcomes (if he wishes). Alternatively, the agent can remain silent, or (if his true preferences won't change the group outcome), he can reveal all his preferences and/or opt out of future steps.

For example, let's say that at a certain step the group choice appears to be $\langle 25, 20, 23\rangle$, and our agent's stated preference (so far, from previous rounds) is $\langle 2, 0, 3\rangle$. Assume that $\delta = 1$. On the next round, our agent can specify the following increase: $\langle 0, 0, 3\rangle$ and thus try to change the outcome in favor of alternative $C$ by 1. All the agents' votes are combined in a linear sum, and a new choice (for this step) results.

Formally, the partial revelation voting procedure is defined as follows:

1. At step 0 all the preferences for all agents are assumed to equal zero. The choice is (arbitrarily) defined to be $s_1$.
2. At step $k$, all agents simultaneously vote. Each agent's vote consists of a vector that specifies the positive increase in his preferences. Each agent can either:
   (a) Keep silent;
   (b) Increase its vote for an outcome different than the current choice ($\tilde{s}^k$), in a way that would change the outcome by $\delta$ had he been the only voter. If the agent opts to attempt to change the group vote on a particular outcome by $\delta$, he is allowed to also increase his vote for other alternatives in parallel;
   (c) If the agent's true preferences will apparently not change the group outcome, he may choose not to take part in any subsequent steps of the voting procedure, and may or may not choose to reveal his entire set of preferences (as he wishes).
3. The new choice $\tilde{s}^{k+1}$ is taken to be the one that gets the highest score at round $k$ (i.e., the choice function $f$ returns $\{s \mid s \in \mathcal{A} \wedge s \text{ is the } \textit{maximizer} \text{ of } \Sigma^r(s)\}$).
4. If there is no change in the vote at step $k$ ($\tilde{s}^k = \tilde{s}^{k+1}$) then the process stops, and $\tilde{s}^k$ is chosen.
5. At all times, each agent must have at least one alternative that is declared to have a worth of zero; this effectively normalizes his preferences.
6. At the end of the process each agent is fined with the Clarke Tax, calculated with respect to all the agents' preferences reached at the last step. Let $r$ be the final round; then the tax paid by $i$ after round $r$ will be $t_i^r(f(D_n^r)) = \sum_{j\neq i}^n d_j^r(f(D_{\neg i}^r)) - \sum_{j\neq i}^n d_j^r(f(D_{\neg i}^r, \vec{d}_i))$.

Below, we will prove certain properties about this iterated voting procedure. First, however, we state a useful lemma, that an agent's best strategy in voting with this procedure is to consistently maintain a balanced distance between his own vote and his true preferences, for every outcome (for which this is possible). So, for example, if his true preferences are $\langle 7, 0, 13 \rangle$, and he is increasing his C vote to sum to 9, his A vote should in parallel be increased to sum to 3.

**Lemma 3** *Let $\vec{\sigma_i^r}$ denote i's declared preferences at round r, such that for each r, there is a constant $\nabla_i^r$ with the following property: for each alternative state s, the true worth of the state differs from $\sigma_i^r(s)$ by that constant. The dominant strategy for voting in the iterative procedure above is to always choose such a $\vec{\sigma_i^r}$ that maintains that constant relationship with the agent's true preferences.*

### 6.3.1 Observations on the Optimal Voting Strategy

Let $\hat{s}_i^r$ denote the alternative that maximizes $a_i$'s utility and let $\Delta_i^r = \Sigma^r(\tilde{s}_i^r) - \Sigma^r(\hat{s}_i^r) + \delta$; this is the amount needed to change the social choice from the current choice to an agent's own choice ($\hat{s}_i^r$). In effect, $\hat{s}_i^r$ is the maximizer of $u_i(f(D_{\neg i}^r, \vec{d_i^r}[d_k \leftarrow (d_i^r(k) + \Delta_i^r)]))$. It is evident from the proof of the above lemma and requirement 5 of the procedure, that $i$'s vote for any alternative $s$ cannot exceed $w_i(s) - w_i(s_i^{min})$. Combined with the constraint that the agent can only increase its vote for each outcome at most by $\Delta_i^r$ (requirement 2b of the procedure) yields $i$'s best response at round $r$ to be:
**(a)** If $\hat{s}_i^r = \tilde{s}^r$ keep silent (the current choice maximizes $i$'s utility), else if $w_i(\hat{s}_i^r) - w_i(s_i^{min}) \leq d_i^r(\hat{s}_i^r) + \Delta_i^r$ then $v_i^r(\hat{s}_i^r) = w_i(\hat{s}_i^r) - w_i(s_i^{min}) - d_i^r(\hat{s}_i^r)$ ($i$ reaches the "ceiling" of his preferences and must opt out) else $v_i^r(\hat{s}_i^r) = \Delta_i^r$ (i.e., $\delta$-change the vote in favor of $\hat{s}_i^r$ within the limitations of the voting procedure). Note that the agent cannot get a "free ride" by keeping silent, since if someone else is going to carry out the job for him, his tax will decrease to zero in any case.
**(b)** Let $\nabla_i^r = d_i^r(\hat{s}_i^r) + \Delta_i^r - d_i^r(k) - w_i(\hat{s}_i^r)$. Then for each $k \neq \hat{s}_i^r$ $v_i^r(k) = \max[0, w_i(k) + \nabla_i^r]$ (i.e., update his vote to most adequately reflect his *true* preferences with respect to $\hat{s}_i^r$ within the limitations imposed by the voting procedure).

We now state certain properties about the iterated voting procedure presented above:

**Theorem 2** *The partial revelation voting procedure has the following properties:*

1. *The chosen alternative is identical to the one chosen under the regular CTm, within $\delta$ (i.e., if two choices differ by less than $\delta$, the wrong one may be chosen);*

2. *On the average, agents have to reveal less information than in the original setting, and those who are indifferent between alternatives reveal their preferences before those who assign greatly different values to alternatives. In addition, the "amount of information" needed is inversely proportional to the identity of preferences held by the agents (i.e., the closer the agents are to one another's preferences, the less needs to be revealed).*

3. *Each agent has a maximum "spread" between alternatives with highest and lowest worth. Call the largest such spread among the group of voters $Q$. The process will stop after at most $Q/\delta + 1$ rounds.*

## 6.4 Running the Procedure on Our Example

The table in Figure 11 shows how the process works for the scheduling agents described in Section 6.1.

At the first round, the choice is set arbitrarily to be $s_0$. No tax is to be paid by any agent, and therefore each agent $i$ gets a utility of $w_i(s_1)$. Besides $a_4$ (who keeps silent), all other agents prefer to pay $\delta(= 2)$ in order to change the group's vote to be their favorite choice. Since both $a_2$ and $a_3$ prefer $s_2$, $s_2$ wins the highest score and becomes the choice at round 1. Now only $a_1$ and $a_4$ are motivated to bring about a change. To change the outcome of the vote, the score of any competing alternative must reach 6. Therefore, $a_1$ increases his vote for $s_4$ by 4. $a_4$, on the other hand, cannot vote more than 4 for $s_1$, and therefore opts out and reveals his entire (normalized) set of preferences. Thus, at round 2, the choice is $s_4$ where $a_1$ is to be taxed by 5 (for a state that he values 11). At the third round, both $a_2$ and $a_3$ increase their votes for $s_2$ by $3(= 6 - 5 + 2)$, and the group choice turns out to be $s_2$. Since no agent can improve its utility by changing his vote (though $a_1$ prefers outcome $s_4$ [as can be seen in Figure 10 above], he doesn't prefer it enough to change the overall group vote) the process ends; at round 4 everybody keeps silent. Notice that only $a_4$ (who is almost indifferent) ended up revealing (at round 2) his entire (normalized) set of preferences. All other agents have maintained some privacy. For example, $a_3$ and $a_4$ will never know that $a_2$ had significantly preferred $s_1$ over $s_3$ and $s_4$.

## 6.5 Determining $\delta$

For the vote to give an accurate result, $\delta$ should be small enough to allow each agent to express his preferences, i.e., $\delta$ should be less than or equal to the minimal difference between any agent's preferences: $\delta = \min_{s,k\in\mathcal{S},i\in\mathcal{A}} \mid w_i(s) - w_i(k) \mid$. One way of determining a good $\delta$ is the following: at step $-1$, before the voting commences, each agent declares the $\delta$ he prefers, and the minimal value is chosen. An agent has no reason to lie, since a too small $\delta$ wastes time, and a too big one destroys his influence.

To make convergence to a solution more efficient, the value of $\delta$ can also be determined dynamically. At each step, each agent who is not opting out can determine the $\delta$ that he would like to use, and the system-wide $\delta$ will be the minimal of all agents' $\delta$'s. In this process, each agent's best suggestion at round $r > 1$ is to let $\delta_i^r$ be $min_{k'} w_i(k) - w_i(k')$ where $k$ is the minimizer of $\{d_i^r(k) \mid d_i^r(k) > 0\}$ and at round 1 let $\delta$ equal the gap between his most preferred choice and his second preferred one. In the example we get: $\delta_1^1 = 8$, $\delta_2^1 = 6$, $\delta_3^1 = 4$ and $\delta_1^1 = 2$, and therefore $\delta^1 = 2$.

**Figure 11**

# 7 Coalitions

As with most other voting procedures, the original CTm is sensitive to coalitions [84]. In other words, the attributes of the mechanism (such as the important fact that telling the truth is the dominant strategy) are not maintained when agents can enter into agreements before the vote is taken. So when a subset of agents collude, they can exploit the process, getting the outcome they want without having to pay for it.

As an example consider two agents that would rather have some alternative $Y$ to be chosen instead of $X$. To accomplish this, each of the agents could overbid $Y$'s worth such that his declaration alone would be sufficient to make $Y$ be the chosen alternative. This way (assuming that the agents know the sufficient value and are not facing any other coalition) $Y$ will be chosen with none of the agents having to pay any tax (since neither agent *alone* caused $Y$ to be chosen, which is the condition upon which a tax is levied). In the general case, the larger their distortion is, the surer they are of forcing their preferred social decision, but the more risk they must accept if their assumptions about the others' votes proves erroneous.

More specifically, consider the vote described in Figure 2. $a_2$ and $a_4$ caused their favorable state ($s_3$) to be chosen, but are fined with a tax. If both agree before the vote and declare (for instance) $d_2 = \langle -45, 12, 33\rangle$, $d_4 = \langle -24, -15, 39\rangle$, $s_3$ is still chosen, but both agents avoid any tax payment.

Even worse, consider another possible coalition that might form. Both $a_1$ and $a_5$ prefer $s_1$ over the chosen $s_3$. If they come to a pre-vote agreement, they might both declare $d_1 = \langle 69, -33, -36\rangle$ and $d_2 = \langle 86, 2, -88\rangle$ and guarantee the group choice of $s_1$ (instead of $s_3$), without paying any tax.

Such crafty schemes may be easily dealt with if the formation of coalitions is known *a priori*, since by treating each coalition as a single agent, truth-telling remains each coalition's dominant strategy. But if formation of coalitions may occur secretly, the danger of the process being manipulated by these coalitions always exists in Clarke Tax mechanisms.

Unfortunately, it was proven in [3, 37], that there exists no successful preference revelation mechanism (i.e., such that truth telling is each agent's dominant strategy and the ultimate choice of the group is the one that maximizes the social welfare) that is immune to coalitions. Moreover, even if the size of possible coalitions is limited to equal some fixed $1 < c < n$, no such mechanism exists. On the other hand, for a fixed coalition, the expected gain from cheating, as compared to telling the truth, decreases with the number of agents in the population. And if the population is taken to be a random sample in a fixed distribution, the probability that a cheating coalition of size $\leq \sqrt{n}$ will gain any fixed positive utility approaches zero as the sample size grows [37].

Thus, the sensitivity of the mechanism to coalitions cannot be handled without some loss in optimality. One way is to limit the magnitude of bids allowable to each agent such that they will not exceed a predefined limit. The risk of such a limitation, of course, is that it might force an agent to understate his true preferences. It might at times be reasonable to set the limit based on the agent's declared goal, for example, setting the limit to be no more than $C(s_0 \rightsquigarrow s_k)$. But again, this might tempt the agent to declare a more expensive goal (although such dishonesty might be tracked by later observing his actual bids).

Another way is to charge a tax on winners (i.e., agents whose desired outcomes were chosen by the group) who would otherwise, in the regular CTm, not pay any tax (this will cause a deviation from the optimal equilibrium). Let $A_w$ denote the set of agents whose highest bid was chosen, and who ended up not having to pay any tax. For each agent $a_i$ in this group, we then calculate his tax by the profile $(\mathcal{D}_{\neg a_i}, D_w)$ (the vote with $a_i$ but without the other members of $A_w$). Using this additional tax mechanism, agents might still form coalitions to reduce their own tax, but such coalitions will not alter the chosen state. Although by telling the truth each member of this group is guaranteed to to pay less than the utility he gains from the chosen state, it is possible to reduce this tax in proportion to the portion of members of $A_w$ in the voting society (for example multiplying the tax by $1- \mid A_w \mid /N$).

## 7.1 Coalitions in the Partial Revelation Voting Procedure

As with the original CTm, our iterative voting procedure is also sensitive to coalitions: two or more agents could act in a coordinated way so that neither has to pay any tax. In certain ways, the iterated procedure can be exploited by coalitions even more easily than the original Clarke Tax, because there is a built-in mechanism for coordinating moves with your partner: keep cooperatively altering the outcome to be your choice, and there's no tax (in the original Clarke Tax, agents would need both to set up their own coordination prior to voting, and second-guess the votes of non-coalition members).

We can alter the simultaneous iterative procedure so that it is done sequentially, one agent voting after the other, with otherwise the same rules as those given above. In this case, the procedure is immune to coalitions, because there can be only one agent who is causing the change in outcome (i.e., not two or more agents simultaneously). The agent that changes the outcome, and he alone, will pay the tax.

However, there is a problem with the procedure, the recurrent "free-rider problem" that was mentioned above. An agent may be tempted not to vote honestly so that he won't be the one to pay the tax, in the hope that some other agent will still (later) cause the original agent's most-preferred outcome to be chosen. If an agent counts on someone after him changing the outcome, and that doesn't happen, then the "incorrect" (non-optimal) choice may be made by the group. The possibility of an incorrect outcome, however, only occurs when agents act on possibly incorrect beliefs—when they restrict themselves to acting on definite *knowledge* about other agents, the correct outcome will result from the voting procedure (though agents still get free rides). This sequential iterative procedure thus guarantees that as long as an agent operates relative only to his definite *knowledge* about the other agents, the correct outcome will be chosen and no coalition can manipulate it.

# 8 Related Work

In other research, we have considered related issues of how to use the Clarke Tax mechanism in automated systems. In [21] we explored how to distribute the voting mechanism so that it no longer requires a central vote counter. In [18] we considered the issue of how to use the tax that is collected (which cannot be used for the benefit of the voters themselves), and how to distribute the workload of a global plan among the agents. In [17] we analyzed how standard cryptographic techniques can be used as another means to maintain privacy regarding agent preferences. All of this work makes fundamental use of voting theory and economic mechanisms. A survey of relevant issues can be found in [84, 68, 54, 41].

Although the use of (economic) voting mechanisms to derive consensus in multiagent systems is novel within artificial intelligence, the issues of reaching consensus, and in a broader sense coordination, in multiagent environments, is the main concern of researchers in Distributed Artificial Intelligence. In this section we briefly review some of the most relevant work from that field.

Most of the DAI work on solving problems of coordination has been carried out by researchers within the area of *Distributed Problem Solving* (DPS). One common approach is to use a central coordinator. This approach includes *Centralized Planning*, where one central agent generates the global plan and then hands out pieces of that plan to be performed by the participating agents [48, 9, 67, 69]. Within this centralized approach, some work falls within the category of *synchronization of pre-existing plans* [29, 30]. The assumption there is that individual plans are first created, and then submitted to a central planner that is responsible for coordination. In [49] a fast probabilistic approach to solving this kind of coordination is suggested.

There are many methods for relaxing these centralized solutions to coordination. One common way is to have some hierarchy of coordinating agents, such that each agent is responsible for the coordination of those who are below it in the hierarchy. Coordination is then done through some communication process. One approach that uses this hierarchical technique is that of *Partial Global Plans* [14, 13]. Within this framework, the group's activity is modeled as a network and there is a distinction between three hierarchical types of plans (information nodes). Climbing the hierarchy, each node has a more global and long-term perspective on the multiagent activity. Some other approaches allow the the hierarchy to be dynamically changed, as in [83].

The underlying working assumption of the DPS paradigm, that the agents inhabiting the multiagent environment are centrally designed, also gave rise to some frameworks that impose coordination as an integral part of the environment. A notable member of this class is the work on *Artificial Social Systems* [62] that was inspired by the *Multi-Entity* [61] model. It uses the society metaphor to design robots that are to operate in a loosely-coupled fashion. A formal definition and basic semantics for artificial social systems were presented. This model was augmented with a set of restrictions on agents' actions to define the actions that an agent may "legally" perform. This concept gave rise to the term *social laws* which was introduced in [78]. A set of social laws is meant to enable efficient interaction among agents that adhere to them. In [78] the usefulness of social traffic laws within the domain of mobile robots was illustrated. The main research assumed that social laws are being generated off-line (i.e., prior to the actual interaction) and that the participating agents are to (benevolently) obey the given laws. In more recent work, the idea of dynamic emergence of such laws is discussed [79].

Note the difference between this work on social laws, and the mechanism design that we've been exploring in this article. The research on social laws concerns itself with the public behavior of agents, and not at all with the private preferences held by these agents. It is assumed that the desired public behavior can be directly imposed on the agents.

Another approach taken is to establish cooperation through the formalization of agents' intentions. Agents have to take their beliefs and intentions into consideration as they collaboratively plan [39, 38]. The collaborative agents build full plans from partial plans that they can alter dynamically over time.

Closer to our approach are frameworks that explicitly address the need for agents to reach an agreement or consensus. Typically, consensus is reached through a process of goal revelation and information exchange, loosely categorized under the label *negotiation* [70, 14, 87, 52, 8].

As an example, the PERSUADER system [86, 87] uses negotiation to find a compromise that is acceptable to the agents in conflict. Their goals might not be totally satisfied by the final agreement. The negotiation process can be seen as a search in a dynamic space consisting of the agents' beliefs about other agents' beliefs. This space changes dynamically as the agents' proposals are revealed.

Another example is the Multi-Fireboss Phoenix system [60]. Planning (the actions needed to assess and contain fires) is performed by several spatially distributed agents. The system addresses, through a sophisticated negotiation protocol, the dynamic allocation of resources.[19]

Less work has been done within DAI on the subject of reaching a consensus in the context of *Multiagent Systems* (MAS). One approach has been to use negotiation in the sense of Game Theory and (more specifically) Bargaining Theory [65, 92, 43, 55, 73, 74].

[19] Our approach would solve this problem in a direct manner, without negotiation. At each time interval, the agents would vote over the possible relevant distributions (one step of the algorithm per time interval). Given the individual utilities, the accurate distribution of resources would be chosen that maximizes the social utility (minimizes the damage according to the group's perspective). In addition, there is no need to assume that the agents are benevolent.

One game-theoretic method for coordinating the activities of autonomous agents, within MAS, is the *Recursive Modeling Method* [34]. Each agent models the other agents in a recursive manner and thus acquires probabilistic knowledge about the expected utility values that the other agents have about their preferences, abilities, and the world. Each agent looks for an action that will maximize its individual utility, by estimating the others' expected utility values. The uncertainties are represented as probabilistic distributions.

An alternative game-theoretic model, in which time is taken into consideration, appears in [53]. In general, the agents have a common goal to achieve. Each wants to do as little as possible to help carry out this goal. The agents are assumed to have full information, to be rational, and to commit themselves to the agreements they have reached. The *set of all possible agreements* is assumed to exist and to include all the pairs of work sharing that will satisfy a goal.

Another approach to handling negotiation in MAS has been explored in [71]. Within this framework, agents converge to a single choice in a so-called *negotiation set*. This negotiation set is the group of all agreements that exhibit the properties of positive utility for all agents, and Pareto Optimality.

The disadvantage of these classic approaches to negotiation, compared with our approach, is that they place a large computational burden on the negotiating agents. The agents must compute all the elements in the negotiation set, a computation which may be non-trivial. All agents must also have sufficient information about one another's preferences in order to compute the negotiation set. Agents need to consider all their and their opponents' possible strategies to determine their best response. And, finally, much of the work on negotiation has treated only two-agent consensus. Since, given the agents' preferences and the optimality criterion, determining the optimal choice is a matter of direct computation, the substantive role of the negotiation process is to reveal preferences. Our method for uncovering the true preferences of agents does away with the need for this kind of negotiation.

Another way of bypassing negotiation, and one that is quite similar to our own approach, is to use *Market-Like Mechanisms*. There have been several attempts, both inside of artificial intelligence and outside, to consider market mechanisms as a way of revealing agents' true preferences (and thus efficiently allocate resources). Notable among the AI work is that of Smith's Contract Net [82, 83], Malone's Enterprise system [56], the work of Miller and Drexler on Agoric Open Systems [59], and Wellman's WALRAS system [91].

The Contract Net is a high-level communication protocol for a Distributed Problem Solving system. Its aim is to facilitate the distribution of the tasks among the processors (nodes) that operate in the system. The collection of nodes is itself the *contract net*. A contract between two nodes is established so that tasks can be executed. Each node in the net can act either as a *manager* or as a *contractor*. A task that has been assigned to a node in the net can be further decomposed by the contractor. An agent can be a manager for one task and he can execute another task as a contractor, even simultaneously. A contract is established by a bidding scheme that includes the announcement of the potential manager and the bids sent by the potential contractors. A recent formalization of the bidding and awarding decision process that was originally described informally appears in [75], where the formalization is based on marginal cost calculation according to local agent criteria.

Enterprise is a system that was built using a variation of this protocol. The protocol is used to schedule tasks among different processors that are connected in a local area network. The personal workstations are dedicated to their owners but while idle, they serve as general purpose machines. The *Distributed Scheduling Protocol* is used to locate the best available machine to perform a task. This protocol is similar to that of the Contract Net. The main difference is in the (more well-defined) assignment criteria. In Enterprise, the contractors select the tasks, announced by the managers, according to the task's priority, and the managers select their contractors according to the time completion estimates of the contractors.

The concept of contracts was also used in [50]. One agent will have a contract with another when it wants some of its tasks to be done by the other. It might be the case that the first agent cannot perform his task, or that the other can do it better. A reward method based on a monetary system to convince agents to accept the contracts is used. The mechanism of sub-contracting is evaluated according to the *simplicity*, *Pareto Optimality*, and *stability* criteria.

Another system that takes an economic approach to solve a problem distributed among several agents, based on a price mechanism, has been proposed by Wellman [91]. There are two types of agents: *consumers* and *producers* that buy and sell goods. Producers may transform some kind of goods into others according to their production function. Each type of agent has an initial allocation of goods. Both types try to maximize their utility — consumers try to consume as much as they can, while producers maximize their profit. Each distinct good has an *auction* associated with it. The agents can get the good by submitting bids in the auction for that good. These bids specify a correspondence between prices and quantities of the good that the agent wants to demand or to supply. The market is in equilibrium with respect to some commodity, when the current price for that commodity is clearing regarding the current bids. A price is clearing when the quantity of the good that has been demanded is balanced by the quantity that has been supplied. The system presented there, WALRAS, computes for each market the equilibrium price.

# 9 Conclusions

In this article we have presented the Clarke Tax mechanism (CTm) as a plausible tool for deriving consensus in Multiagent Systems. The Clarke Tax is a voting procedure with several highly desirable characteristics for automated agents: it encourages truth-telling, and results in a choice that exhibits maximal social utility. We have addressed several fundamental implementation problems that arise when considering the employment of the mechanism in practical real-world systems. These problems included the generation of alternatives, the assessment of alternatives' worth, power and influence among the participating agents, and formation of coalitions. Another major issue that was treated was the potential desire of agents to keep their preferences private, as far as possible. While the original mechanism requires revealing full preferences, we developed a method that maintains agent privacy, while preserving other positive attributes of the CTm.

We introduced a novel voting procedure that enables a group of agents to construct a joint plan that results in a final state that maximizes social welfare for the group. Conflicts among agents are incrementally dealt with; agents iteratively search for a final state that maximizes the entire group's utility, incrementally constructing a plan to achieve that state.

We also introduced an alternative simultaneous, iterative voting procedure that enables agents to reach the decision of highest social utility with only partial information, presented the optimal strategy for voting in this procedure, and proved several desirable properties of the procedure. We also introduced a sequential version of the voting procedure that discouraged coalitions (which remains a general problem with using the Clarke Tax mechanism).

Systems comprised of individually motivated automated agents are likely to become increasingly common. These agents will need to resolve conflict and reach consensus to carry out their tasks effectively. Techniques such as the ones we have explored provide powerful tools for this coordination of multiagent activity. Moreover, by approaching the question of multiagent environment design formally, we are able to construct alternative protocols and precisely characterize their properties.

# Acknowledgments

We would like to thank Motty Perry and Ed Durfee for useful discussions about this material. The latter was especially helpful, suggesting privacy and distribution issues of the Clarke Tax mechanism as particularly important for the DAI community.

This research was partially supported by the Israeli Ministry of Science and Technology (Grant 032-8284), the Israel Science Foundation (Grant 032-7517), the Air Force Office of Scientific Research (Contract F49620-92-J-0422), by the Rome Laboratory (RL) of the Air Force Material Command and the Defense Advanced Research Projects Agency (Contract F30602-93-C-0038), and by an NSF Young Investigator's Award (IRI-9258392) to Prof. Martha Pollack.

# References

[1] Anat R. Adami and Motty Perry. Joint project without commitment. *Review of Economic Studies*, 58:259–276, 1991.

[2] K. J. Arrow. *Social Choice and Individual Values.* John Wiley, New York, 1963.

[3] E. Bennet and D. Conn. The group incentive properties of mechanisms for the provision of public goods. *Public Choice*, 24(2):95–102, 1977.

[4] Ken Binmore. *Fun and Games, A Text on Game Theory.* D. C. Heath and Company, Lexington, Massachusetts, 1992.

[5] D. Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32(3):333–377, July 1987.

[6] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, Fall 1971.

[7] E. H. Clarke. Multipart pricing of public goods: An example. In S. Muskin, editor, *Public Prices for Public Products*. Urban Inst., Washington, 1972.

[8] S. E. Conry, R. A. Meyer, and V. R. Lesser. Multistage negotiation in distributed planning. In Alan H. Bond and Les Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 367–384. Morgan Kaufmann, San Mateo, California, 1988.

[9] D. Corkill. Hierarchical planning in a distributed environment. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 168–175, Tokyo, August 1979.

[10] D. D. Corkill. *A Framework for Organizational Self-Design in Distributed Problem Solving Networks*. PhD thesis, University of Massachusetts, Amherst, MA, December 1982.

[11] J. Doyle and M. P. Wellman. Impediments to universal preference-based default theories. *Artificial Intelligence*, 49(1–3):97–128, 1992.

[12] Jon Doyle. Rationality and its role in reasoning. *Computational Intelligence*, 8(2):376–409, May 1992.

[13] E. H. Durfee and V. R. Lesser. Negotiating task decomposition and allocation using partial global planning. In Les Gasser and Michael N. Huhns, editors, *Distributed Artificial Intelligence, Vol. II*, pages 229–243. Morgan Kaufmann, San Mateo, California, 1989.

[14] Edmund H. Durfee. *Coordination of Distributed Problem Solvers*. Kluwer Academic Publishers, Boston, 1988.

[15] Edmund H. Durfee and Jeffrey S. Rosenschein. Distributed problem solving and multi-agent systems: Comparisons and examples. In *Proceedings of the Thirteenth International Workshop on Distributed Artificial Intelligence*, pages 94–104, Olympic Peninsula, Washington, July 1994.

[16] J. Elgot-Drapkin, M. Miller, and D. Perlis. The two frame problems. In *Proceedings of the 1987 Workshop on the Frame Problem in Artificial Intelligence*, pages 23–28, Lawrence, Kansas, April 1987.

[17] E. Ephrati. *Planning and Consensus among Autonomous Agents*. PhD thesis, The Hebrew University of Jerusalem, Jerusalem, Israel, October 1993.

[18] E. Ephrati and J. S. Rosenschein. The Clarke Tax as a consensus mechanism among automated agents. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 173–178, Anaheim, California, July 1991.

[19] E. Ephrati and J. S. Rosenschein. Multi-agent planning as search for a consensus that maximizes social welfare. In *Pre-Proceedings of the Fourth European Workshop on Modeling Autonomous Agents in a Multi-Agent World*, Rome, Italy, July 1992.

[20] E. Ephrati and J. S. Rosenschein. Reaching agreement through partial revelation of preferences. In *Proceedings of the Tenth European Conference on Artificial Intelligence*, pages 229–233, Vienna, Austria, August 1992. John Wiley & Sons.

[21] E. Ephrati and J. S. Rosenschein. Distributed consensus mechanisms for self-interested heterogeneous agents. In *Proceedings of the First International Conference on Intelligent and Cooperative Information Systems*, pages 71–79, Rotterdam, The Netherlands, May 1993. IEEE Computer Society Press.

[22] E. Ephrati and J. S. Rosenschein. Multi-agent planning as a dynamic search for social consensus. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 423–429, Chambery, France, August 1993.

[23] E. Ephrati, G. Zlotkin, and J. S. Rosenschein. Meet your destiny: A non-manipulable scheduler. In *Proceedings of the ACM 1994 Conference on Computer Supported Cooperative Work*, pages 359–371, Chapel Hill, North Carolina, October 1994.

[24] O. Etzioni. Embedding decision-analytic control in a learning architecture. *Artificial Intelligence*, 49:129–159, 1991.

[25] Jerome A. Feldman and Robert F. Sproull. Decision theory and Artificial Intelligence II: The hungry monkey. *Cognitive Science*, 1(2):158–192, April 1977.

[26] P. C. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33:469–489, 1977.

[27] Drew Fudenberg and Jean Tirole. *Game Theory*. The MIT Press, Cambridge, Massachusetts, 1992.

[28] M. R. Genesereth and R. E. Fikes. Knowledge interchange format version 3 reference manual. Technical Note Logic-92-1, Stanford University Logic Group, 1992.

[29] M. Georgeff. Communication and interaction in multi-agent planning. In *Proceedings of the National Conference on Artificial Intelligence*, pages 125–129, Washington, D.C., August 1983.

[30] M. Georgeff. A theory of action for multi-agent planning. In *Proceedings of the National Conference on Artificial Intelligence*, pages 121–125, Austin, Texas, August 1984.

[31] A. Gibbard. Manipulation of voting schemes: a general result. *Econometrica*, 41:587–602, 1973.

[32] Matthew L. Ginsberg and David E. Smith. Reasoning about action I: A possible worlds approach. In *Proceedings of the 1987 Workshop on the Frame Problem in Artificial Intelligence*, pages 233–258, Lawrence, Kansas, April 1987.

[33] Matthew L. Ginsberg and David E. Smith. Reasoning about action II: The qualification problem. In *Proceedings of the 1987 Workshop on the Frame Problem in Artificial Intelligence*, pages 259–258, Lawrence, Kansas, April 1987.

[34] Piotr J. Gmytrasiewicz, Edmund H. Durfee, and David K. Wehe. A decision-theoretic approach to coordinating multiagent interactions. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*, pages 62–68, August 1991.

[35] I. J. Good. Justice in voting by demand revelation. *Public Choice*, 29(2):65–70, 1977.

[36] J. Green, E. Kohlberg, and J. J. Laffont. Partial equilibrium approach to the free rider problem. *Journal of Public Economics*, 6:375–394, 1976.

[37] J. Green and J. J. Laffont. Coalition incentive compatibility. *Review of Economic Studies*, 46:243–254, 1979.

[38] B. J. Grosz and Sarit Kraus. Collaborative plans for group activities. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 367–373, Chambery, France, 1993.

[39] B. J. Grosz and C. Sidner. Plans for discourse. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication*. MIT Press, 1990.

[40] Theodor Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[41] Theodor Groves. On theories of incentive compatible choice with compensation. In Werner Hildenbrand, editor, *Advances in Economic Theory*, pages 1–29. Cambridge University Press, Cambridge, 1980.

[42] Peter Haddawy and Steven Hanks. Issues in decision-theoretic planning: Symbolic goals and numeric utilities. In *Proceedings of the 1990 DARPA Workshop on Innovative Approaches to Planning, Scheduling, and Control*, pages 48–58, November 1990.

[43] John C. Harsanyi. Approaches to the bargaining problem before and after the theory of games: a critical discussion of Zeuthen's, Hick's and Nash theories. *Econometrica*, (24):144–157, 1956.

[44] John C. Harsanyi. Utilitarian morality in a world of very half-hearted altruists. In Walter P. Heller, Ross M. Starr, and David A. Starrett, editors, *Social choice and public decision making*, chapter 3, pages 57–73. Cambridge University Press, Cambridge, London, 1986.

[45] L. Hurwics. On informationally decentralized systems. In R. Rander and C.B. McQuire, editors, *Decision an organization*. North-Holland, 1972.

[46] W. Jacobs and M. Kiefer. Robot decisions based on maximizing utility. In *Proceedings of the Third International Joint Conference on Artificial Intelligence*, pages 402–411, August 1973.

[47] K. Kanazawa and T. Dean. A model for projection and action. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 985–990, 1989.

[48] Matthew J. Katz and J. S. Rosenschein. Verifying plans for multiple agents. *Journal of Experimental and Theoretical Artificial Intelligence*, 5:39–56, 1993.

[49] Andrew P. Kosoresow. A fast first-cut protocol for agent coordination. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 237–242, Washington, DC, July 1993.

[50] S. Kraus. Agents contracting tasks in non-collaborative environments. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 243–248, 1993.

[51] S. Kraus, J. Wilkenfeld, and G. Zlotkin. Multiagent negotiation under time constraints. Computer Science Technical Report Series CS-TR-2975, University of Maryland, College Park, Maryland, October 1992.

[52] Sarit Kraus, E. Ephrati, and Daniel Lehmann. Negotiation in a non-cooperative environment. *Journal of Experimental and Theoretical Artificial Intelligence*, 3(4):255–282, October–December 1991.

[53] Sarit Kraus, Jonathan Wilkenfeld, and Gilad Zlotkin. Multiagent negotiation under time constraints. *Artificial Intelligence*, 75(2):297–345, June 1995.

[54] J. J. Laffont and E. Maskin. The theory of incentives: an overview. In Hildenbrand Werner, editor, *Advances in Economic theory*, chapter 3, pages 31–94. Cambridge University Press, Cambridge, 1980.

[55] R. Duncan Luce and Howard Raiffa. *Games and Decisions*. John Wiley & Sons, New York, 1957.

[56] Thomas W. Malone, Richard E. Fikes, and M. T. Howard. Enterprise: A market-like task scheduler for distributed computing environments. In B. A. Huberman, editor, *The Ecology of Computation*, pages 177–205. North-Holland Publishing Company, Amsterdam, 1988.

[57] David McAllester and David Rosenblitt. Systematic nonlinear planning. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 634–639, Anaheim, California, July 1991.

[58] D. McDermott. Planning and acting. *Cognitive Science*, 2:71–109, 1978.

[59] M. S. Miller and K. E. Drexler. Markets and computation: Agoric open systems. In B. A. Huberman, editor, *The Ecology of Computation*, pages 133–176. North-Holland Publishing Company, Amsterdam, 1988.

[60] T. Moehlman and V. Lesser. Cooperative planning and decentralized negotiation in Multi-Fireboss Phoenix. In *Proceedings of the Workshop on Innovative Approaches to Planning, Scheduling and Control*, pages 144–159, San Diego, California, November 1990.

[61] Y. Moses and M. Tennenholtz. On cooperation in a multi-entity model. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 918–923, Detroit, Michigan, August 1989.

[62] Y. Moses and M. Tennenholtz. On computational aspects of artificial social systems. In *Proceedings of the 11th International Workshop on Distributed Artificial Intelligence*, pages 267–283, Glen Arbor, Michigan, February 1992.

[63] Herve Moulin. Characterization of the pivotal mechanism. *Journal of Public Economics*, 31:53–78, 1986.

[64] T. Muench and M. Walker. Identifying the free rider problem. In J. J. Laffont, editor, *Aggregation and revelation of preferences*, chapter I(4), pages 61–87. North-Holland, 1979.

[65] J. F. Nash. The bargaining problem. *Econometrica*, 28:155–162, 1950.

[66] J. F. Nash. Two-person cooperative games. *Econometrica*, 21:128–140, 1953.

[67] Richard N. Pelavin. A formal approach to planning with concurrent actions and external events. Technical Report TR 254, University of Rochester, 1988.

[68] Bezalel Peleg. *Game Theoretic Analysis of Voting in Committees.* Cambridge University Press, Cambridge, 1984.

[69] J. S. Rosenschein. Synchronization of multi-agent plans. In *Proceedings of the National Conference on Artificial Intelligence*, pages 115–119, Pittsburgh, Pennsylvania, August 1982.

[70] J. S. Rosenschein and M. R. Genesereth. Deals among rational agents. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 91–99, Los Angeles, August 1985.

[71] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*. MIT Press, Cambridge, Massachusetts, 1994.

[72] J. S. Rosenschein and G. Zlotkin. Consenting agents: Designing conventions for automated negotiation. *AI Magazine*, 15(3), Fall 1994.

[73] Alvin E. Roth. *Axiomatic Models of Bargaining*. Springer-Verlag, Berlin, 1979.

[74] Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, (50):97–109, 1982.

[75] T. Sandholm. An implementation of the contract net protocol based on marginal calculations. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 256–262, 1993.

[76] N. A. Satterthwaite. Strategy-proofness and Arrow's conditions: existence and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[77] L. I. Savage. *The foundations of statistics*. John Wiley, New York, 1964.

[78] Y. Shoham and M. Tennenholtz. On the synthesis of usful social laws for artificial agent societies. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 276–281, San Jose, California, July 1992.

[79] Yoav Shoham and Moshe Tennenholtz. Emergent conventions in multi-agent systems: initial experimental results and observations (preliminary report). In *Principles of knowledge representation and reasoning: Proceedings of the Third International Conference*, Cambridge, Massachusetts, October 1992.

[80] D. Smith and M. Broadwell. Plan coordination in support of expert systems integration. In *Proceedings of the DARPA Knowledge-Based planning Workshop*, pages 12.1–12.6, December 1987.

[81] J. S. Smith. Aggregation of preferences with variable electorate. *Econometrica*, 41:1027–41, 1973.

[82] Reid G. Smith. *A Framework for Problem Solving in a Distributed Processing Environment*. PhD thesis, Stanford University, 1978.

[83] Reid G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, December 1980.

[84] Philip D. Straffin, Jr. *Topics in the Theory of Voting*. The UMAP Expository Monograph Series. Birkhäuser, Boston, 1980.

[85] G. J. Sussman. *A Computational Model of Skill Acquisition.* American Elsevier, New York, 1975.

[86] Katia P. Sycara. Resolving goal conflicts via negotiation. In *Proceedings of the Seventh National Conference on Artificial Intelligence*, pages 245–250, St. Paul, Minnesota, August 1988.

[87] Katia P. Sycara. Argumentation: Planning other agents' plans. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 517–523, Detroit, Michigan, August 1989.

[88] T. N. Tideman and G. Tullock. A new and superior process for making social choice. *Journal of Political Economy*, 84(6):1145–1159, 1976.

[89] G. Tullock. The demand-revealing process as a welfare indicator. *Public Choice*, 29(2):51–63, 1977.

[90] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

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

[92] Frederik Zeuthen. *Problems of Monopoly and Economic Walfare.* G. Routledge & Sons, London, 1930.

[93] G. Zlotkin and J. S. Rosenschein. Negotiation and conflict resolution in non-cooperative domains. In *Proceedings of the National Conference on Artificial Intelligence*, pages 100–105, Boston, Massachusetts, August 1990.

[94] G. Zlotkin and J. S. Rosenschein. Negotiation and goal relaxation. In Y. Demazeau and J. P. Muller, editors, *Decentralized A. I. 2, Proceedings of the Second European Workshop on Modeling Autonomous Agents in a Multi-Agent World*, pages 273–286. Elsevier Science Publishers B.V./North-Holland, 1991.

# A Proofs

We here present proofs of all the theorems and lemmas that appear in the article.

**Proof of Lemma 1:** To show that declaring $\vec{v_i^k}$ is the dominant strategy, we have to show that agent $a_i$'s utility from declaring it is greater than any other declaration $\vec{d_i^k}$:

$u_i^k(f(D_{\neg i}^k, \vec{v_i^k})) - u_i^k(f(D_{\neg i}^k, \vec{d_i^k})) =$

Expanding the utilities $u$ into worth minus tax, we get:
$w_i(f(D_{\neg i}^k, \vec{v_i^k})) - t_i^k(f(D_{\neg i}^k, \vec{v_i^k}))\ - w_i(f(D_{\neg i}^k, \vec{d_i^k})) + t_i^k(f(D_{\neg i}^k, \vec{d_i^k})) =$

Expanding the tax $t$ into its components, we get:
$w_i(f(D_{\neg i}^k, \vec{v_i^k})) - \sum_{j\neq i} d_j^k(f(D_{\neg i}^k)) + \sum_{j\neq i} d_j^k(f(D_{\neg i}^k, \vec{v_i^k}))$
$-w_i(f(D_{\neg i}^k, \vec{d_i^k})) + \sum_{j\neq i} d_j^k(f(D_{\neg i}^k)) - \sum_{j\neq i} d_j^k(f(D_{\neg i}^k, \vec{d_i^k})) =$

Eliminating equivalent plus and minus terms and adding and subtracting $w_i(s_0)$, we get:
$w_i(f(D_{\neg i}^k, \vec{v_i^k})) - w_i(s_0) + \sum_{j\neq i} d_j^k(f(D_{\neg i}^k, \vec{v_i^k}))\ - [w_i(f(D_{\neg i}^k, \vec{d_i^k})) - w_i(s_0) + \sum_{j\neq i} d_j^k(f(D_{\neg i}^k, \vec{d_i^k}))] =$

By $v_i^k$'s definition:
$v_i(f(D_{\neg i}^k, \vec{v_i^k})) + \sum_{j\neq i} d_j^k(f(D_{\neg i}^k, \vec{v_i^k}))\ - [v_i(f(D_{\neg i}^k, \vec{d_i^k})) + \sum_{j\neq i} d_j^k(f(D_{\neg i}^k, \vec{d_i^k}))] =$

Let $\hat{s}^k$ be the state that maximizes $v_i^k(s^k) + \sum_{j\neq i} d_j^k(s)$ (i.e., $\hat{s}^k = f(D_{\neg i}^k, \vec{v_i^k})$), and let $\hat{k}^k$ be the state that maximizes $\sum_{i=1}^n d_i^k(s)$ (i.e., $\hat{k}^k = f(D_{\neg i}^k, \vec{d_i^k})$). Then (by the definition of the choice function $f$) we get:

$v_i(\hat{s}^k) + \sum_{j\neq i} d_j^k(\hat{s}^k) - [v_i(\hat{k}^k) + \sum_{j\neq i} d_j^k(\hat{k}^k)] =$

By $\hat{s}^k$'s definition:
$max_{k,s}[v_i(s^k) + \sum_{j\neq i} d_j^k(s^k)] - [v_i(\hat{k}^k) + \sum_{j\neq i} d_j^k(\hat{k}^k)] \geq 0$ □

**Proof of Theorem 1:**

The general idea is that each set of aggregated constraints is extended by the search process until the "limit of consistency"; therefore, any maximal consistent set of aggregated constraints will eventually be reached. But it is the case that any state that answers the Social Welfare Criterion is induced by a maximal set of consistent constraints. Therefore, the search process will reach sets of constraints that induce all Social Welfare states.

For simplicity, we assume that the set of individual worth functions is strictly progressive. The generalization of the proof to $\delta$-progressive worth functions is straightforward.

Let $s \in SW(\mathcal{S})$ and $K$ be the last step. We need to prove that $s \in \hat{\mathcal{S}}^K$ (the final set of states in step 5 of the algorithm).

Let $G = \mathcal{I}(s_0 \rightsquigarrow s)$ be the set of constraints that induces $s$, and let $l = l(P(G))$ be the length of the plan that results in $s$. We will prove a stronger claim: *for any given (imposed) $n$, the procedure finds all states $\hat{s}$ such that $l(P(s_0 \rightsquigarrow \hat{s})) \leq n$ and $\hat{s}$ answers the social welfare criterion with regard to that additional constraint.*

Let $\mathcal{S}\rceil^n$ denote all states that are n-distant (or less) from $s_0$. Since, given $n$, each such

$\hat{s}$ is in $SW(\mathcal{S}\rceil^n)$, we have (by the definition of $SW$ for any other state $s' \in \mathcal{S}\rceil^n$, $s' \neq \hat{s}$) that $U^0(\hat{s}) \geq U^0(s')$. Therefore, if $\hat{s}$ is a member of any $\mathcal{S}^k\rceil^n$ it will be a member of $\hat{\mathcal{S}}^K\rceil^n$. It is thus sufficient to prove that for some $k$, $\hat{s}$ belongs to $\mathcal{S}^k\rceil^n$. Because of step 3a in the procedure, it is sufficient to prove (for any given $n$) that $\hat{s}$ is generated at some stage $k$ of the procedure, that is, there exists a stage $k$ at which a set of constraints that induces $\hat{s}$ is generated. Or, using the procedure's notation: $\exists k \exists G^{\hat{s}}[G^{\hat{s}} \in \mathcal{G}'^k \wedge s \in s(G^{\hat{s}})]$. The proof is by induction on $n$:

1. $n = 1$:

   (a) $l = 0$, so $\hat{s} = s_0$ and $G^{\hat{s}} = \emptyset$. From the procedure it follows that $G^{\hat{s}} \in \mathcal{G}^0$, and obviously $s_0 \in s(\emptyset)$. In effect, since by definition $\forall i \forall s_0 \models I_i^1$, it is also true that $\mathcal{G}'^1 = \mathcal{G}^1 = G^1 = \cup_i I_i^1$ and $s_0 \in s(G^1)$.

   (b) $l = 1$; $\hat{s}$ can be achieved from $s_0$ by one operator. $U^0(s_0) \leq U^0(\hat{s})$, so it must be the case that $\hat{s}$ is preferred to $s_0$ by some agents $A \subseteq \mathcal{A}$ (i.e., for each member of $A$ $w_i(\hat{s}) \geq w_i(s_0)$ holds). To win higher worth, a state must satisfy more constraints of the individual plans (we assume progressive worth functions). Let the set of additional constraints be denoted by $I^+$. It therefore follows that $I^+ \subseteq \cup_{i \in A} I_i$. Since $s_0$ enables $I^+$ ($I^+ \in f_{ollow}(s_0)$), then actually $I^+ \in \cup_{i \in A} I_i^2 \subseteq \mathcal{N}(\mathcal{G}^0)$) (this is because agents want to maximize their own utility, and thus, impose as many constraints as they can at each step). Therefore, $I^+ \in \mathcal{G}'^2$ and $I^+$ is the required $G^{\hat{s}}$.

2. Assume that the claim holds for any $l \leq n-1$ and that $l(P(G^{\hat{s}})) = n$. Let $\check{\mathcal{G}}$ denote all the sets that could proceed $G^{\hat{s}}$, i.e., sets of constraints $\check{G}$ such that $[\check{G} \subset G^{\hat{s}}] \wedge [\check{G} \cup f_{ollow}(\check{G}) = G^{\hat{s}}] \wedge [l(P(\check{G})) \leq n-1]$).

   For each state $s$ which is induced by the sets in $\check{\mathcal{G}}$, it holds that $U^0(s) \leq U^0(\hat{s})$. It must then be the case that $\hat{s}$ satisfies more constraints than any such $s$.

   Let $I^+(s)$ denote the set of these additional constraints. From the monotonicity of the worth functions with regard to satisfied constraints, it follows that there must be at least one state, $\check{s}$, in $s(\check{\mathcal{G}})$ (real or virtual) that is a member of $SW(\mathcal{S}\rceil^r)$ (where $r = l(P(s_0 \rightsquigarrow \check{s})))$. $\hat{s}$ is superior to that state since it satisfies $I^+(\check{s})$, but the only way for these additional constraints to be satisfied is by invoking some operators in $\check{s}$. That is, given the restriction that plans must not exceed length $r$, $\check{s}$ is a maximizer of the social welfare.

   On the other hand, since $\check{s} \in SW(\mathcal{S}\rceil^r)$, it follows from the induction assumption that any such $\check{s}$ will be found by the procedure at some step $\check{k}$. Also, since $\check{s} \in SW(\mathcal{S}\rceil^r)$, it does not reduce the social utility of any of its preceding states, and thus $\check{G}$ is not pruned in the $\check{k}$th step. Let $A \subseteq \mathcal{A}$ be the group of agents that prefer $\hat{s}$ to $\check{s}$. For each member of $A$, $w_i(\hat{s}) \geq w_i(\check{s})$.

It must, then, be the case (strictly progressive worth functions) that $I^+(\check{s}) \subseteq \cup_{i \in A} I_i$. Since $\check{s}$ enables $I^+(\check{s})$ (by definition, $\forall I \in I^+(\check{S}) I \in f_{ollow}(\check{s})$), $I^+(\check{s}) \in \cup_{i \in A} I_i^{\check{k}+1}$, and therefore $(I^+(\check{s}) \cup \check{G}) \in \mathcal{G}'^{\check{k}+1}$ and $\hat{s} \in s(\check{G} \cup I^+(\check{s}))$.

The number of steps in our search is bounded by $O(\max_{s \in SW(\mathcal{S})} l(P(s_0 \leadsto s)))$. At any step of the procedure, each set of constraints $(G)$ is either being pruned or extended. The extension will be by at least one constraint that belongs to $f_{ollow}(G)$. Eventually, the furthest state of maximal social welfare will be found. Any extensions on the set of constraints that induces this state will be pruned at the next step. Therefore, the process cannot proceed longer than the number of constraints needed to achieve this state. □

**Proof of Lemma 3:** To show that declaring $\vec{\sigma_i^r}$ is the dominant strategy, we have to show that the utility of agent $a_i$ from declaring it is greater than any other declaration $\vec{d_i^r}$:

$u_i^r(f(D_{\neg i}^r, \vec{\sigma_i^r})) - u_i^r(f(D_{\neg i}^r, \vec{d_i^r})) =$

Expanding the utilities $u$ into worth minus tax, we get:
$w_i(f(D_{\neg i}^r, \vec{\sigma_i^r})) - t_i^r(f(D_{\neg i}^r, \vec{\sigma_i^r})) - w_i(f(D_{\neg i}^r, \vec{d_i^r})) + t_i^r(f(D_{\neg i}^r, \vec{d_i^r})) =$

Expanding the tax $t$ into its components, we get:
$w_i(f(D_{\neg i}^r, \vec{\sigma_i^r})) - \sum_{j \neq i} d_j^r(f(D_{\neg i}^r)) + \sum_{j \neq i} d_j^r(f(D_{\neg i}^r, \vec{\sigma_i^r}))$
$-w_i(f(D_{\neg i}^r, \vec{d_i^r})) + \sum_{j \neq i} d_j^r(f(D_{\neg i}^r)) - \sum_{j \neq i} d_j^r(f(D_{\neg i}^r, \vec{d_i^r})) =$

Eliminating equivalent plus and minus terms, we get:
$w_i(f(D_{\neg i}^r, \vec{\sigma_i^r})) + \sum_{j \neq i} d_j^r(f(D_{\neg i}^r, \vec{\sigma_i^r})) - [w_i(f(D_{\neg i}^r, \vec{d_i^r})) + \sum_{j \neq i} d_j^r(f(D_{\neg i}^r, \vec{d_i^r}))] =$

By addition and subtraction of $\nabla_i^r$ , we get:
$w_i(f(D_{\neg i}^r, \vec{\sigma_i^r})) - \nabla_i^r + \sum_{j \neq i} d_j^r(f(D_{\neg i}^r, \vec{\sigma_i^r})) - [w_i(f(D_{\neg i}^r, \vec{d_i^r})) - \nabla_i^r + \sum_{j \neq i} d_j^r(f(D_{\neg i}^r, \vec{d_i^r}))] =$

By $\sigma_i^r$'s definition ($\sigma_i^r(s) = w_i(s) - \nabla_i^r$ for any $s$), we get:
$\sigma_i^r(f(D_{\neg i}^r, \vec{\sigma_i^r})) + \sum_{j \neq i} d_j^r(f(D_{\neg i}^r, \vec{\sigma_i^r})) - [\sigma_i^r(f(D_{\neg i}^r, \vec{d_i^r})) + \sum_{j \neq i} d_j^r(f(D_{\neg i}^r, \vec{d_i^r}))] =$

Let $\hat{s}^r$ be the maximizer of $\sigma_i^r(s) + \sum_{j \neq i} d_j^r(s)$ ($\hat{s}^r = f(D_{\neg i}^r, \vec{\sigma_i^r})$), and let $\hat{k}^r$ be the maximizer of $\sum_{i=1}^n d_i^r(s)$ ($\hat{k}^r = f(D_{\neg i}^r, \vec{d_i^r})$). Then (by the definition of the choice function $f$) we get:

$\sigma_i^r(\hat{s}^r) + \sum_{j \neq i} d_j^r(\hat{s}^r) - [\sigma_i^r(\hat{k}^r) + \sum_{j \neq i} d_j^r(\hat{k}^r)] =$

By $\hat{s}^r$'s definition:
$max_s[\sigma_i^r(s) + \sum_{j \neq i} d_j^r(s)] - [\sigma_i^r(\hat{k}^r) + \sum_{j \neq i} d_j^r(\hat{k}^r)] \geq 0$ □

**Proof of Theorem 2:** (1) Assume to the contrary that the voting procedure stops at round $R$, and $s^* \neq \tilde{s}^R$ where $s^*$ is the choice of the Clarke Tax mechanism. It must then be the case that there exists at least one $i$ such that $w_i(s^*) - w_i(\tilde{s}^R) > d_i^R(s^*) - d_i^R(\tilde{s}^R)$, that is $w_i(s^*) - d_i^R(s^*) > w_i(\tilde{s}^R) - d_i^R(\tilde{s}^R)$. It must be the case that there is a round $r \leq R$ in which the vote $v_i^r(\tilde{s}^R)$ is different than $i$'s optimal behavior (above); this would contradict $i$'s behavior as a utility maximizer.
(2) It is sufficient to consider the extreme cases where all agents' best alternative is $s_1$ (the arbitrary choice), where the vote will end at the first round with no information revealed at all, or the case where all prefer the same alternative $s_k$, and the vote ends in the second round with $s_k$ being the choice (scoring $n \times \delta$). Since the gap $max_{s \in \mathcal{S}} w_i(s) - w_i(s_i^{min})$ of an indifferent agent is (relatively) close to zero, such an agent will lose influence after a small number of rounds and will have to opt out, revealing his entire set of preferences. The bigger this spread is, the longer the agent can participate. (It would require further analysis to show the precise correlation between disagreement and information revelation.)
(3) The stopping condition follows directly from the fact that $i$'s influence on any group choice $s^*$ cannot exceed $w_i(s^*) - w_i(s_i^{min}) - \delta$ (which is $i$'s extreme influence on the vote); it follows from the procedure that at each round (excluding the first and last rounds) the score of the winning alternative at that round increases by at least $\delta$ (as long as the voting process continues). Thus, for any $i$, the "cost" of changing the vote increases by at least $\delta$ at each round. Therefore, in the extreme case, after at most $(w_i(s^*) - w_i(s_i^{min}))/\delta + 1$ rounds $i$'s vote for $s^*$ reaches $w_i(s^*) - w_i(s_i^{min})$ and $i$ must opt out. Thus, for any $i$, the vote cannot last more than $(\max_{s \in \mathcal{S}}[w_i(s) - w_i(s_i^{min})])/\delta + 1$. Notice that it is evident from the above argument that the closer the agents are to one another's preferences, the sooner the process will end. □

**Proof of Lemma 2:**[20] To show that declaring $\vec{v_i}$ is the dominant strategy, we have to show that the utility of agent $a_i$ from declaring it is greater than any other declaration $\vec{d_i^k}$:

$u_i(f(D_{\neg i}, \frac{1}{z_i}\vec{v_i})) - u_i(f(D_{\neg i}, \frac{1}{z_i}\vec{d_i})) =$

Expanding the utilities $u_i$ into worth minus tax, we get
$w_i(f(D_{\neg i}, \frac{1}{z_i}\vec{v_i})) - z_i \times t_i(f(D_{\neg i}, \frac{1}{z_i}\vec{v_i})) - w_i(f(D_{\neg i}, \frac{1}{z_i}\vec{d_i})) + z_i \times t_i(f(D_{\neg i}, \frac{1}{z_i}\vec{d_i})) =$

Expanding the tax $t_i$ into its components, we get
$w_i(f(D_{\neg i}, \frac{1}{z_i}\vec{v_i})) - z_i \times \sum_{j \neq i} d_j(f(D_{\neg i})) + z_i \times \sum_{j \neq i} d_j(f(D_{\neg i}, \frac{1}{z_i}\vec{v_i}))$
$-w_i(f(D_{\neg i}, \frac{1}{z_i}\vec{d_i})) + z_i \times \sum_{j \neq i} d_j(f(D_{\neg i})) - z_i \times \sum_{j \neq i} d_j(f(D_{\neg i}, \frac{1}{z_i}\vec{d_i})) =$

Eliminating equivalent plus and minus terms and adding and subtracting $w_i(s_0)$, we get
$w_i(f(D_{\neg i}, \frac{1}{z_i}\vec{v_i})) - w_i(s_0) + z_i \times \sum_{j \neq i} d_j(f(D_{\neg i}, \frac{1}{z_i}\vec{v_i}))$

[20] The lemma was originally proven in [35] using case analysis; the proof here is original.

$-[w_i(f(D_{\neg i}, \frac{1}{z_i}\vec{d_i})) - w_i(s_0) + z_i \times \sum_{j \neq i} d_j(f(D_{\neg i}, \frac{1}{z_i}\vec{d_i}))] =$

By $v_i$'s definition (for any $s$, $v_i(s) = w_i(s) - w_i(s_0)$):
$v_i(f(D_{\neg i}, \frac{1}{z_i}\vec{v_i})) + z_i \times \sum_{j \neq i} d_j(f(D_{\neg i}, \frac{1}{z_i}\vec{v_i}))$ $-[v_i(f(D_{\neg i}, \frac{1}{z_i}\vec{d_i})) + z_i \times \sum_{j \neq i} d_j(f(D_{\neg i}, \frac{1}{z_i}\vec{d_i}))] =$

By factoring out $z_i$, we get
$z_i \times [\frac{1}{z_i} v_i(f(D_{\neg i}, \frac{1}{z_i}\vec{v_i})) + \sum_{j \neq i} d_j(f(D_{\neg i}, \frac{1}{z_i}\vec{v_i}))$ $-[\frac{1}{z_i} v_i(f(D_{\neg i}, \frac{1}{z_i}\vec{d_i})) + \sum_{j \neq i} d_j(f(D_{\neg i}, \frac{1}{z_i}\vec{d_i}))]] =$

Let $\hat{s}$ be the maximizer of $\frac{1}{z_i} v_i(s) + \sum_{j \neq i} d_j(s)$ ($\hat{s} = f(D_{\neg i}, \frac{1}{z_i}\vec{v_i})$), and let $\hat{k}$ be the maximizer of $\frac{1}{z_i} d_i(s) + \sum_{j \neq i} d_i(s)$ ($\hat{k} = f(D_{\neg i}, \frac{1}{z_i}\vec{d_i})$). By the definition of the choice function $f$, we get

$z_i \times [\frac{1}{z_i} v_i(\hat{s}) + \sum_{j \neq i} d_j(\hat{s}) - \frac{1}{z_i}[v_i(\hat{k}) + \sum_{j \neq i} d_j(\hat{k})]] =$

By $\hat{s}$'s definition ($\hat{k}$ is *not necessarily* maximizing over $\frac{1}{z_i}\vec{v_i}$):
$z_i \times [max_s[\frac{1}{z_i} v_i(s) + \sum_{j \neq i} d_j(s)] - [\frac{1}{z_i} v_i(\hat{k}) + \sum_{j \neq i} d_j(\hat{k})]] \geq 0$ □

# B Basic Concepts of Solution for Social Decision Processes

A common requirement of any decision function is that it should be "optimal" in some sense. Different kinds of desirable attributes of decision functions that characterize optimality have been suggested in game theory, economics, and voting theory. Typically, the attributes are concerned with the influence of an individual agent on the outcome, and the impact of the outcome on the individual. Some common criteria include Pareto Optimality, Fairness, and Individual Rationality.[21] In this section we briefly summarize the most common criteria.

In general, the optimality of the decision process may be viewed with respect to two main aspects/categories. Below we mention some of the more common criteria that have been addressed in the literature [68, 84, 55, 45]:

1. Attributes of the resulting decision:

   (a) Global optimality — The chosen alternative should be optimal in some global sense. The most common requirement is that the decision will be **Pareto Optimal**, meaning that it is impossible to change the decision in a way that will make some agents better off without making some other agents worse off. This same attribute is sometimes called Unanimity, to denote that if alternative $X$ is preferred over some other alternative $Y$ by all agents (unanimously) than $Y$ should

[21] The issues of solution criteria are discussed in [11], in the context of combining various default theories.

not be chosen. Within the "pareto frontier" (all the Pareto Optimal decisions), there are many additional criteria of optimality based on Social Welfare theory. We discuss this issue further in Section B.2.

(b) Condorcet Winner — The chosen alternative should beat any other alternative in a pairwise contest. There are several weaker versions of this demand, such as the Condorcet Loser Criterion (if an alternative $Y$ would lose in pairwise contests with every other alternative, $Y$ should not be chosen). The Generalized Condorcet Criterion [81] demands that if the alternatives can be partitioned into two sets $A$ and $B$ such that every alternative in $A$ beats every alternative in $B$ in a pairwise contest, then the process should not end up choosing an alternative from $B$. Note that the generalization implies both of the previous criteria.

(c) Majority Criterion — If the majority of agents have an alternative $X$ as their first choice, the decision process should choose $X$. Another, stronger version states that if $X$ is preferred to $Y$ by a majority of agents, then $Y$ should not be the ultimate choice of the group.

(d) Nash's Independence of Irrelevant Alternatives — If out of a set of alternatives, $X$ is chosen, then if any other alternative $Y$ is removed from the set, $X$ will still be chosen. In other words, the choice should remain unchanged when the group is presented with a subset of the original group that includes the original choice. A weaker version (Arrow's Independence of Irrelevant Alternatives) demands that the choice will be independent of any potential alternative that is not included in the current set of alternatives (the collective preferences over any pair depend only upon individual preferences over that pair).

(e) Monotonicity — If $X$ is to be chosen by the process and one or more agents change their preferences *in favor* of $X$ (without changing the order of preferences over the other alternatives), then $X$ should remain the choice. Issue Monotonicity requires that the utility of the choice that is taken, given the entire set of alternatives, will not be lower than the utility of a choice that was based on any subset of the entire set.

2. Attributes of the process itself:

(a) Individual Rationality — An agent may only gain (utility) by taking part in the process (compared to not participating).

(b) Simplicity — The process should be simple in two respects: the computational complexity of determining the choice, and the individual computational complexity of determining each agent's behavior (strategy) in light of the rules of the process.

(c) Stability — The behavior of the participants should converge to an equilibrium point, and remain insensitive to minor perturbations of strategy among the players (in Section B.3 we describe the main notion of equilibrium from Game Theory).

(d) Privacy Preserving ("Information Decentralization") — The behavior of each individual should depend on as little information as possible regarding the others (preferences and behavior), and the choice function should depend on as much of a global view as possible (in contrast to taking into account interactions among individual preferences). For example, it is considered preferable if one's behavior can be determined according to others' aggregated behavior instead of having to take into account the individual behavior of each other member [45].

(e) Decentralization — The degree of distribution affects the likelihood of a bottleneck, the fragility of the process, and the need for a central decision maker.

(f) Expressiveness — Most decision mechanisms consider only the Ordinal preferences of the agents. The magnitude by which some alternative is preferred to another cannot be expressed. However, there are mechanisms that allow more powerful rating of preferences, such as rating the alternatives by points, or the assignment of actual (cardinal) utilities to preferences. The more expressive the rating is, the more informed can be the choice that is made.

(g) Symmetry — Given the possible permutations of the agents' roles in the process, the outcome should remain the same regardless of these permutations. The strongest kind of symmetry is *Anonymity*, which says that the process answers all possible symmetries (the identity of an agent has absolutely no influence on the outcome).

(h) Neutrality — The name of alternatives does not matter; if we exchange two candidates $a$ and $b$ in the ordering of every agent, then the outcome of the process should change accordingly.

## B.1 General Results from Social Welfare Theory

There is a large body of work in voting theory and social welfare theory that considers how groups make decisions when there is no transferable utility (no "money" that can be extracted from, or paid to, agents). In Social Welfare theory there exists the distinction between a *Social Welfare Function* (SWF) and a *Social Choice Function* (SCF). The first assigns a social ordering over the entire set of alternatives based on the individual preference profiles, while the second chooses *one* alternative from the set of alternatives.

In a classic paper, Arrow [2] introduced several appealing SWF attributes, and proved that they cannot be simultaneously satisfied. In our own case, however, we are interested in SCFs. There are many SCFs that reach a Pareto Optimal decision, but they suffer from a major drawback: they are *manipulable*, which means that an agent can benefit by declaring a preference other than his true preference. Thus, a rational agent will tend to manipulate the process.

Therefore, the stability of the choice mechanism is essential in our case. Since we assume that agents are rational and self-interested, we are concerned about their misrepresenting preferences to manipulate group decisions. Unfortunately, a theorem due to Gibbard [31] and Satterthwaite [76] states that any non-manipulable SCF that ranges over more than two alternatives is dictatorial. This means that there is no choice function (other than one corresponding strictly to one of the agents' preferences), that motivates all participating agents to reveal their true desires. However, this result pertains only to interactions where there is no transferable utility. The Clarke Tax Mechanism depends precisely on such a transfer of utility.

## B.2 Optimality and Social Welfare

Consider the designers of a multiagent environment, who are charged with establishing the rules by which agents in an encounter will interact. Once the rules have been determined, each builder of each agent is free to design his own machine any way that he wants. However, the rules that were established will certainly affect the choices he makes in building his own agent.

Under many circumstances, the designers of multiagent environments will be able to formalize some notion of global utility, and want agents' activity to maximize that global utility. While this point of view may not be universally accepted (for example, designers may be unable to come to agreement about any definition of global utility) it provides a useful starting point for our design considerations. Therefore, in this research we are interested in consensus states that maximize the welfare of the entire society.

However, there are many ways to measure global utility, and it is not obvious how environment designers will decide on one or another. Considerations other than pure utility values (such as income and fairness) might need to be taken into account. For example, it might be desirable in some scenarios to look for a state that maximizes the median of the utilities or some weighted sum of these utilities, or, following an egalitarian approach we might want to maximize the minimal utility ($\max \min_i u_i$) or minimize the differences in utility gain ($\min \max_{i,j} \mid u_i - u_j \mid$).

One simple common approach (due to Nash [65, 66]) is to choose the outcome that maximizes the product of the individual utilities ($\prod_i u_i$). This approach guarantees a relatively fair distribution of the mutually earned utility, but narrows the space of feasible consensus states. It also assumes a positive utility gain for each participating agent. A negotiation protocol for autonomous agents that follows this approach may be found in [71].

In our approach, on the contrary, all agents share the cost of achieving the consensus state, and thus may indeed have negative utility. For these agents, the rationale for participation is to possibly reduce their loss. Taking a global perspective on the agents' activity also ignores the issue of individual fairness (Section 5.6 discusses this problem).

In this research we therefore follow the pure utilitarian approach, and prefer consensus states that maximize the sum of the individual agents' utilities minus the cost of the final state's achievement. In contrast to the approach that maximizes the product, we would rather have, for example, a state that gives two agents a utility of 0 and 11 respectively, over a state that gives each of them a utility of 1. We would also prefer a state that gives a total utility of 11 and costs 1 to achieve, to a state with total utility of 22 and a cost of 13. A further discussion of this approach may be found in [44].

Assuming that each agent assigns a worth to each alternative (see Section 3.1), we define the following criteria for global maximization of utility:

- Given a set of worth functions $\{w_i\}$, of the agents in the decision group, we define the Social Welfare/Utility of each state to be the summation of its worths for all members of the society, minus the cost of achieving the state from $s_0$; $U(s) = -C(s_0 \leadsto s) + \sum_{i=1}^{n} w_i(s)$. If we were to normalize the worth functions of each agent such that $\forall i w_i(s_0) = 0$, we could define the normalized global utility of a state ($U^0(s)$) to be the sum of the normalized worth functions.

- We say that a voting procedure satisfies the Social Welfare Criterion if it ends up choosing a nonempty set of states $SW(\mathcal{S}) \subseteq \mathcal{S}$ such that $\forall s[s \in SW(\mathcal{S}) \Rightarrow U(s) \geq max_{k \in \mathcal{S}} U(k)]$ (i.e., SW(S) contains the states with maximal social utility). Notice that since these states are "maximal," they are also Pareto Optimal (no one can benefit without someone else losing; if this were not true, the state could not be maximal).

## B.3 Game Theoretic Concepts of Solution

The general kind of interaction that we consider here may be viewed as a game. Each agent $i$ chooses a strategy $s_i$ (within a computer, of course, the strategy is simply the program controlling the computer's choices). The strategy tells it which action (declaration of preferences in our case) to choose at each instant of the interaction. The combination of strategies played by the entire group ($S = (s_i, s_{-i})$) determines the outcome of the interaction, and in particular determines the resulting payoff for each agent ($\pi_i$).

Game theory has addressed many interactions similar to the one considered here. Such interactions have been analyzed so as to determine what an agent's chosen strategies would be, given the rules of the interaction. Our aim is complementary; it is to *design* rules that would induce the agents to adopt some specific strategy that we consider to be desirable (similar in spirit to *mechanism design* or the so-called implementation problem, in game theory [4, 27]).

All possible developments of the interaction may be represented by a game tree. Each node represents a decision choice of some player; each different choice is represented by a different branch. Given the history of the interaction, an agent might not be able to distinguish, among a set of possible nodes, which one is the actual node. This set is called the information set at that particular point. Each path on the tree describes one possible interaction. The end nodes of the tree describe each agent's resulting payoff from that path.

To be motivated to adopt a particular strategy, a rational selfish agent should be convinced that that strategy is superior in some sense to his other alternative strategies. The most common solution in game theory derives cooperation as the best response to the other agents' cooperative behavior:

**Definition 1** *The strategy combination* $s^*$ *is a* Nash equilibrium *if no agent has an incentive to deviate from his strategy given that the other agents do not deviate. Formally* $\forall i$, $\pi_i(s_i^*, s_{-i}^*) \geq \pi_i(s_i', s_{-i}^*)$, $\forall s_i'$.

This concept of solution was used for example (within the Distributed Artificial Intelligence literature) in [71]. Although this concept of equilibrium is satisfying in many cases, it suffers several drawbacks. First, it embeds the implicit assumption that each agent can monitor and verify that the other indeed cooperated. Such an assumption is especially problematic when a society of computationally-bounded artificial agents is being considered. Second, in general there might be multiple equilibrium points for the same game, and it thus might be difficult to have the group of users converge to a specific equilibrium point.[22] Third, the desirability of a strategy is considered only from a player's viewpoint at the beginning of the interaction (not taking into consideration *all* possible paths of the game), causing the equilibrium point to be sensitive to the dynamics of the interaction.

A much stronger concept of solution (the second one in the hierarchy of solutions) derives the desired strategy (cooperation in our case) to be the unique equilibrium along any development (path) of the interaction.

**Definition 2** *A* subgame *is a game consisting of an information set which is a singleton in every player's information partition, that node's successors, and the payoffs at the associated end nodes, such that all information sets that include any of that node's successors, does not include a node which is not one its successor nodes.*

*A strategy combination is a* subgame perfect Nash equilibrium *if (a) it is a Nash equilibrium for the entire game; and (b) its relevant action rules are a Nash equilibrium for any subgame.*

This approach was taken, for example, in [1, 51]. However, the strongest concept of solution within the hierarchy is to motivate the agent to follow the desirable behavior regardless of the others. Such motivation would be achieved if that strategy would be proven to be (under the rules of encounter) the best one given *any* strategy of the other agents.

**Definition 3** *The strategy* $s_i^*$ *is a* dominant strategy *if it is an agent's strictly best response to any strategies that the other players might pick, in the sense that whatever strategies they pick, his payoff is highest with* $s_i^*$. *Formally* $\pi_i(s_i^*, s_{-i}) \geq \pi_i(s_i', s_{-i})$, $\forall s_i \forall s_i' \neq s_i^*$.

*A* dominant strategy equilibrium *is a strategy combination of each player's dominant strategy.*

Thus, in our scenario, the most attractive solution would be to have some particular behavior with certain desirable properties as each agent's dominant strategy. Having a mechanism that induces an equilibrium point which is the result of a dominant strategy is

[22] On the other hand, a single equilibrium point among equivalent ones can be specified and agreed upon ahead of time by agent designers. This allows engineers to have machines that converge to a solution.

very desirable, because it simplifies the reasoning required of an agent. The fact that there is no importance to the other agents' behavior does away with the need to reason about the other agents' strategies, knowledge, or even computational capabilities. The behavior of an agent depends solely on its own characteristics (in Economics [45] this attribute is known as "informational decentralization"). Thus, in comparison to other approaches, the individual complexity of decision-making is reduced significantly. This concept of solution was used in [20, 18, 19]. In this article we have focused on the design of rules of encounter that induce a solution in dominant strategy equilibrium.

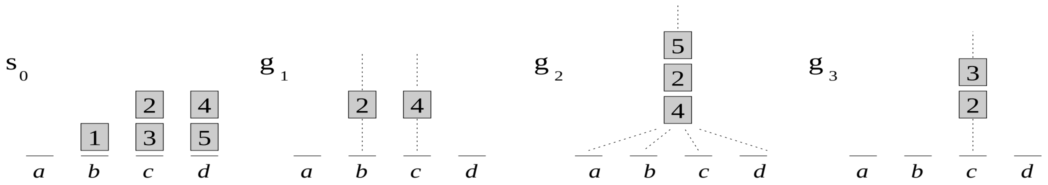


Figure 1: A Blocks World Example of Individual Goals

| | True worth of each state | | | Sum for each state without $i$ | | | Tax for $i$ |
|---|---|---|---|---|---|---|---|
| | $s_1$ | $s_2$ | $s_3$ | $s_1$ | $s_2$ | $s_3$ | |
| $a_1$ | 27 | −33 | 6 | −46 | *23 | *23 | 0 |
| $a_2$ | −36 | 12 | 24 | *17 | −22 | 5 | 12 |
| $a_3$ | −9 | 24 | −15 | −10 | −34 | *44 | 0 |
| $a_4$ | −18 | −15 | 33 | −1 | *5 | −4 | 9 |
| $a_5$ | 17 | 2 | −19 | −36 | −12 | *48 | 0 |
| Sum | −19 | −10 | *29 | | | | |

Figure 2: Calculating the Clarke Tax

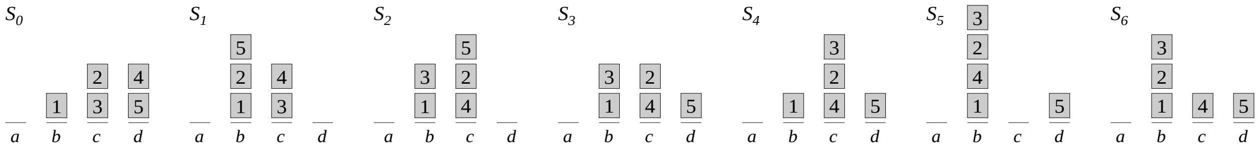


Figure 3: Alternative Final States

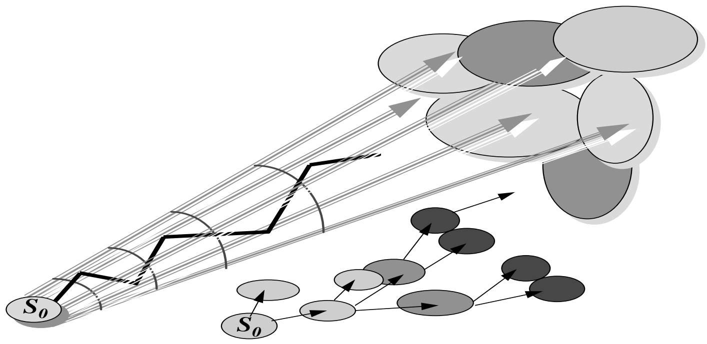


Figure 4: Overview of the Process

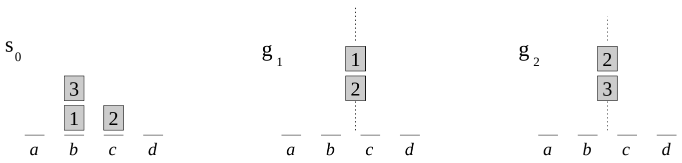


Figure 5: A Simpler Blocks World Example

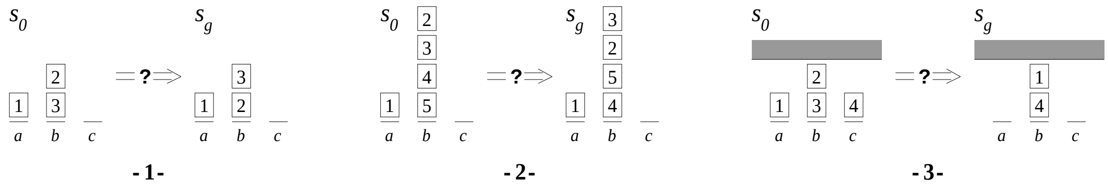


Figure 6: Three scenarios in the slotted blocks world

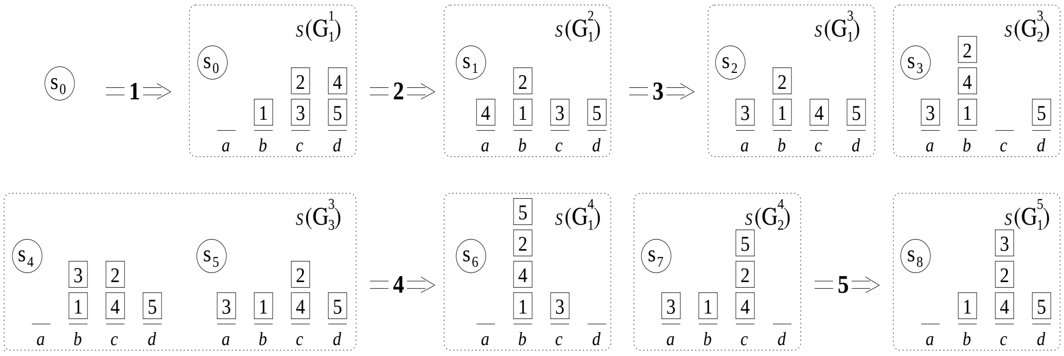

Figure 7: Induced States of the Five Steps

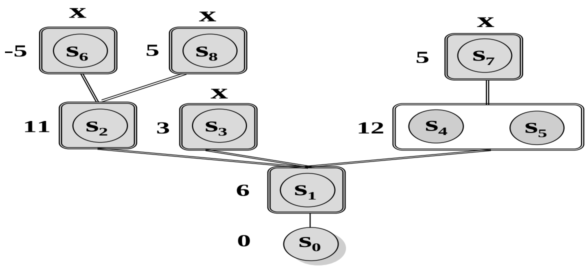

Figure 8: The Entire Search Tree

| | Worth of each state | | | | | | | | Score of each state without $i$ | | | | | | | | Tax |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_5$ | $s_6$ | $s_7$ | $s_8$ | |
| $a_1$ | 2 | 5 | −7 | 0 | 0 | −11 | −5 | −5 | 4 | 6 | 10 | ∗12 | ∗12 | 6 | 10 | 10 | 0 |
| $a_2$ | 2 | 1 | 5 | 4 | 4 | 9 | 7 | −1 | 4 | ∗10 | −2 | 8 | 8 | −14 | −2 | 6 | 2 |
| $a_3$ | 2 | 5 | 5 | 8 | 8 | −3 | 3 | 11 | 4 | ∗6 | −2 | 4 | 4 | −2 | 2 | −6 | 2 |
| *sum* | 6 | 11 | 3 | ∗12 | ∗12 | −5 | 5 | 5 | | | | | | | | | 4 |

Figure 9: Cumulative Voting using the Clarke Tax Mechanism

| | True worth of each schedule | | | | Sum for each schedule without $i$ | | | | Tax for $i$ | Utility for $i$ |
|---|---|---|---|---|---|---|---|---|---|---|
| | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | | |
| $a_1$ | 3 | 3 | −5 | 11 | 14 | ∗41 | −2 | 22 | 0 | 3 |
| $a_2$ | 10 | 16 | 0 | 2 | 7 | 28 | −7 | ∗31 | 3 | 13 |
| $a_3$ | −4 | 20 | −8 | 16 | 21 | ∗24 | 1 | 17 | 0 | 20 |
| $a_4$ | 8 | 5 | 6 | 4 | 9 | ∗39 | −13 | 29 | 0 | 5 |
| *sum* | *17* | **44* | *-7* | *33* | | | | | *3* | *41* |

Figure 10: The Standard Clarke Tax Mechanism

| $r$ | $i$ $(\Sigma^r(s_k))$ | $w(s_k)$ | | | | $\Sigma^r(s_k) - d_i^r(s_k)$ | | | | $t_i^r$ | $u_i^r$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $s_1$ | $s_2$ | $s_3$ | $s_4$ | $s_1$ | $s_2$ | $s_3$ | $s_4$ | | |
| $r = \mathbf{0}$ | $a_1$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 |
| | $a_2$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 |
| | $a_3$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | −4 |
| | $a_4$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 |
| | *sum* | **0* | *0* | *0* | *0* | | | | | *0* | *17* |
| $r = \mathbf{1}$ | $a_1$ | 0 | 0 | 0 | 2 | 0 | ∗4 | 0 | 0 | 0 | 3 |
| | $a_2$ | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 2 | 0 | 16 |
| | $a_3$ | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 2 | 0 | 20 |
| | $a_4$ | 0 | 0 | 0 | 0 | 0 | ∗4 | 0 | 2 | 0 | 5 |
| | *sum* | *0* | **4* | *0* | *2* | | | | | *0* | *44* |
| $r = \mathbf{2}$ | $a_1$ | 0 | 0 | 0 | 6 | 4 | ∗5 | 2 | 0 | 5 | 6 |
| | $a_2$ | 0 | 2 | 0 | 0 | 4 | 3 | 2 | ∗6 | 0 | 2 |
| | $a_3$ | 0 | 2 | 0 | 0 | 4 | 3 | 2 | ∗6 | 0 | 16 |
| | $a_4$ | 4 | 1 | 2 | 0 | 0 | 4 | 0 | ∗6 | 0 | 4 |
| | *sum* | *4* | *5* | *2* | **6* | | | | | *5* | *28* |
| $r = \mathbf{3\ (4)}$ | $a_1$ | 0 | 0 | 0 | 6 | 4 | ∗11 | 2 | 2 | 0 | 3 |
| | $a_2$ | 0 | 5 | 0 | 0 | 4 | 6 | 2 | ∗7 | 1 | 15 |
| | $a_3$ | 0 | 5 | 0 | 1 | 4 | 6 | 2 | 6 | 0 | 20 |
| | $a_4$ | 4 | 1 | 2 | 0 | 0 | ∗10 | 0 | 7 | 0 | 5 |
| | *sum* | *4* | **11* | *2* | *7* | | | | | *1* | *43* |

Figure 11: Iterative Steps Towards a Consensus