Minimal Social Laws

David Fitoussi and Moshe Tennenholtz
Faculty of Industrial Engineering and Management
Technion–Israel Institute of Technology
Haifa 32000, Israel

Abstract

Research on social laws in computational environments has proved the usefulness of the law-based approach for the coordination of multi-agent systems. Though researchers have noted that the imposition of a specification could be attained by a variety of different laws, there has been no attempt to identify a criterion for selection among alternative useful social laws. We propose such a criterion which is based on the notion of *minimality.* A useful social law puts constraints on the agents' actions in such a way that as a result of these constraints, they are able to achieve their goals. A minimal social law is a useful social law that minimizes the amount of constraints the agents shall obey. Minimal social laws give an agent maximal flexibility in choosing a new behavior as a function of various local changes either in his capabilities or in his objectives, without interfering with the other agents. We show that this concept can be usefully applied to a problem in robotics and present a computational study of minimal social laws.

1. Introduction

The design of an agent which is about to operate in a multi-agent environment is quite different from the design of an agent which performs his activities in isolation from other agents. Typically, a plan that would have allowed an agent to obtain his goals had he operated in isolation might yield unexpected results as a consequence of other agents' activities. Various approaches to multi-agent coordination have been considered in the DAI literature (Bond & Gasser 1988; Durfee 1992; Demazeau & Muller 1990). We could, for instance, subordinate the agents to a central controller. This approach may be useful in various domains (Stuart 1985; Lansky 1988), but might suffer from well-known limitations (e.g. bottleneck at the central site or sensitivity to failure). Another approach is to design rules of encounter, i.e. rules which determine the agent's behavior (and in particular, the structure of negotiation) when his activities interfere with those of another agent (Rosenschein & Zlotkin 1994; Kraus & Wilkenfeld 1991). Rules of encounter may be quite useful for conflict resolution, but might sometimes be inefficient, requiring repeated negotiations to solve on-line conflicts.

In this paper we consider an intermediate approach to coordination, referred to as the *artificial social systems* approach (Moses & Tennenholtz 1990; Shoham & Tennenholtz 1995). An artificial social system institutes a *social law* (Shoham & Tennenholtz 1995; Minsky 1991a; Briggs & Cook 1995) that the agents shall obey. Intuitively, a social law restricts, off-line, the actions legally available to the agents, and thus minimizes the chances of an on-line conflict, and the need to negotiate. Similarly to a code of laws in a human society (Rousseau 1762), an artificial social law regulates the individual behavior of the agents and benefits the community as a whole. Yet, the agents should still be able to achieve their goals, or any other specification of the system, and restricting their legal actions to a too wide extent might leave them with no possible way to do so. Consider for instance a domain consisting of roads on which our agents travel. These roads cross one another at junctions where total freedom on the side of the agents makes an accident a likely event. In order to guarantee accident-free traffic we could set a law that allows an agent to enter an intersection only if the crossing road is free. This law certainly prevents accidents. However, it restricts the agents too much. Although we have guaranteed an accident-free environment, we have also introduced a possibility of a deadlock: when two agents reach the intersection via crossing roads, they might find themselves waiting indefinitely for the crossing road to get free before initiating their move. This example illustrates the fact that we must be careful in designing social laws: Only useful social laws, i.e laws which guarantee that each agent achieves his goals, are to be considered. In the example above, the law could oblige the agents to cross the intersection one at a time, in a round-robbin policy. Roughly speaking, a useful law will set constraints on the set of actions available to the agents, but not so much that they cannot achieve their goals anymore.

The previous discussion emphasizes that meeting the system specification is an essential requirement for accepting a candidate social law. This raises the question of whether this condition is the only requirement one may need to consider. Should we be satisfied once the law we designed guarantees the system specification? Clearly, there might still be quite a diversity of laws with such a characteristic. In this paper, we look for a way to further classify social laws, and define the notion of *minimal social laws* on which we found our analysis. The idea behind minimal social laws is to prefer a useful social law to another useful social law if the former imposes less restrictions than the other. Hence, minimal social laws capture the idea of maximal individual flexibility. We wish to enable the agents to behave arbitrarily, as long as the original specification of the system can be guaranteed. There is no need to disallow particular actions if doing so will not prevent other agents from obtaining their goals. We further discuss the implications of choosing minimal social laws in the following sections.

The study of social laws has concerned itself with various semantic and computational issues, as well as with applications (Onn & Tennenholtz 1997; Ben-Yitzhak & Tennenholtz 1997). A problem that we believe to be essential to the understanding of social laws and their role in multi-agent systems is the search for an optimal law according to some measure of optimality. With this purpose in mind, we shall now investigate the notion of minimal social laws. The remainder of this paper is structured as follows. In Section 2 we introduce and define minimal social laws. In Section 3 we present a study of minimal social laws in the context of mobile robots, or Automated Guided Vehicles (Sinreich 1995; Latombe 1991). In Section 4 we present a computational study of minimal social laws. We show that the problem of deciding whether a law can be minimized by dropping constraints on actions is NP-hard. We then show that an efficient algorithm can be obtained in a restricted setting.

2. Minimal Social Laws

A central issue in the design of social laws, which has not been addressed yet is the design of optimal laws, according to some measure of optimality. Given a system of agents and a specification, the job of the designer is to find an implementation consistent with this specification. A possible way to come up with a suitable implementation will be to identify a strategy profile such that if each agent acts according to the strategy assigned to him in this profile, the specification is satisfied. This behavior will then be enforced and serve as the law of the system (Minsky 1991a; 1991b). Clearly, this law will be consistent with the system specification (according to which it has been designed). Most often, a less restrictive law will exist, and we should be able to compare the two laws. How to do so? When could we say that we have found an optimal law, and under which criterion?

Following the literature on mechanism design in Economics (Kreps 1990) a law could be considered as optimal if it brings maximal utility to the designer of the system. Although legitimate, this definition does not capture the fact that the purpose of social laws is to provide a flexible framework for a system to evolve in. This will therefore motivate another notion of optimality, that we call minimality, and which we would like to relate to the impact social laws have on the dynamics of the system (and its components).

Roughly speaking, given two different useful laws l_1 and l_2, we say that l_2 is smaller than l_1 if the set of behaviors induced by strategies consistent with l_1 is included in the set of behaviors induced by strategies consistent with l_2 (i.e., if we regard a law l as a set of restrictions, l_2 imposes *less* restrictions than l_1). Intuitively, a smaller law is a law that rules out less behaviors consistent with the agents' goals. We then relate the size of a law to optimality in the following way: a useful social law l^* is minimal (and optimal) for some system specification, if-f for any other useful social law l, l is not smaller than l^*.

A minimal law will grant the agents maximal freedom in the process of choosing an appropriate behavior for achieving their goals, while ensuring that they conform to the system specification. Systems with smaller social laws are therefore more robust to changes in the environment specification or in the capabilities of the agents. Notice that minimal laws need not be unique, and choosing among them might require some other exogenous criterion (either quantitative or qualitative).

We now define the notions of useful and minimal social laws in the framework of a general strategic model. For ease of exposition, we present the definitions for environments consisting of two agents. Extension to the case of n agents ($n \geq 2$) follows easily.

Definition 1 *An* environment *is a tuple* $\langle N, S_1, S_2 \rangle$, *where* $N = \{1, 2\}$ *is a set of agents, and* S_i *is a set of strategies available to agent* i.

Given an environment, agents are assigned goals which they must fulfil by selecting an appropriate strategy among their possible strategies. In addition, there might be some safety goals that should always be guaranteed.

Definition 2 *Given an environment $\langle N, S_1, S_2 \rangle$, a goal g is a subset of the Cartesian product over the agents' strategy spaces, i.e. , $g \subseteq S_1 \times S_2$.*

The above definition captures goals in very general terms. Roughly speaking, a goal is associated with the set of joint strategies in which it is indeed obtained.

We distinguish between several sets of goals. Let us denote by G_i the set of *liveness goals for agent i*. These are goals that agent i need to obtain. Naturally, at a given initial state the agent may wish to obtain a particular goal and another goal may be irrelevant.[1] In addition, there is a set G_{safe} of *safety goals*. These are goals that should always be obtained. The formal definition of goal achievement for liveness and safety goals will be given below.

The purpose of (useful) social laws is to set regulations that ensure safety and guarantee that liveness is achievable. When searching for minimal social laws, we seek a minimal set of such regulations. Given an environment and a goal for agent i, it is not certain that the agent has a strategy such that, independently of the strategy profile chosen by his fellow agents, he will achieve his goals. The job of the designer is to devise a social law (if one exists) which makes sure that each agent in the resulting system achieves his goals, regardless of the (legal) behavior chosen by the other agents. Fulfilling this condition will qualify the law as useful.

Definition 3 *Given an environment $< N, S_1, S_2 >$, and given the sets of goals G_1, G_2, and G_{safe}, a social law is a set of restrictions $SL =< \overline{S_1}, \overline{S_2} >$ such that $\overline{S_1} \subseteq S_1$ and $\overline{S_2} \subseteq S_2$. SL is* useful *if :*

1. *for every goal $g_{1_i} \in G_1$ there exists $s_{1_i} \in S_1 \setminus \overline{S_1}$ such that for all $s_2 \in S_2 \setminus \overline{S_2}$ we have $(s_{1_i}, s_2) \in g_{1_i}$.*
2. *for every goal $g_{2_i} \in G_2$ there exists $s_{2_i} \in S_2 \setminus \overline{S_2}$ such that for all $s_1 \in S_1 \setminus \overline{S_1}$ we have $(s_1, s_{2_i}) \in g_{2_i}$.*
3. *for every $g_j \in G_{safe}$ and for all $s_1 \in S_1 \setminus \overline{S_1}$, $s_2 \in S_2 \setminus \overline{S_2}$, we have that $(s_1, s_2) \in g_j$.*

Notice that in general, social laws need not be symmetric and may assign roles to agents.[2] In addition, notice that a social law precisely defines which strategies are allowed and which are not (see the semantics of artificial social systems in (Moses & Tennenholtz 1995)). Hence, a social law is different from a refined specification in which the set of allowed actions for a particular agent is to be determined at a later point (e.g., do not collide will not be a typical social law, but drive in the right lane will be one).

As discussed before, social laws differ in their properties, and some (useful) social laws are more stringent than others. Our approach to selecting among laws is to prefer those that satisfy the safety and liveness conditions with minimal constraints. Formally,

Definition 4 *Consider an environment with a specification of liveness and safety goals. A useful social law $SL =< \overline{S_1}, \overline{S_2} >$ is* minimal *if there is no other useful social law $SL' =< S'_1, S'_2 >$ that satisfies $S'_i \subseteq \overline{S_i}$ for all i.*

In the sequel we will refer, unless stated otherwise, to minimal useful social laws as *minimal social laws*. It is quite clear that if an agent need to modify his behavior (such a change might be motivated e.g., by changes in his capabilities or in his goals), minimal social laws will grant him maximal freedom to do so.

This will be illustrated in the next section and in part of our computational study. The reader should be careful not to confuse minimality and anarchy: minimality is bounded by the need to satisfy the basic specification (including safety and liveness goals). Needless to say, from time to time, changes in the system and new requirements require a re-design of the social law. The role of minimal social laws is to serve as a basic optimization tool in between these transitions, allowing the agents maximal choice of behaviors (in order to adapt to changes) while enabling them to obtain the original specification. It is this delicate tradeoff between free choice and the need to obtain the original specification that minimality attempts to capture.

3. A case study: AGVs in a circular automated assembly line

In this section we study minimal social laws in the domain of Automated Guided Vehicles (AGVs). We wish to emphasize that although we use this study mainly for purposes of illustration, the simple setting discussed below and variants of it are quite popular in the AGVs literature (Sinreich 1995).

In a single-robot automated assembly line, a robot is programmed to perform some activity which will lead it to a goal state, i.e. a situation where its goal has been fulfilled. When several robots are acting together, interactions between the actions of the robots may tamper with normal operation. To make the ideas more concrete, let us consider a simple automated assembly line, where m robots can move between n stations in a circular fashion ($2 \leq m \leq n$). This domain is represented by a connected undirected graph $G = (V, E)$

[1] The initial state is implicit in the agent's strategy in our general model. It will be treated more specifically in our more concrete applications and computational study.

[2] This point is discussed in a more complete version of this paper, and is omitted due to space constraints.

where $|V| = n, |E| = n$ and for all $v \in V, deg(v) = 2$ (where $deg(v)$ is the degree of the vertex v) i.e. the domain is represented by a graph with n vertices and a ring topology. Each node in the ring represents a station. In our simple model, a robot can move at time t (we will assume that time steps are discrete and that time is infinite) from the station it stands at to one of its two neighbors, or stay immobile. All the robots move at the same speed, and a robot which left a station at time t will reach one of the adjacent stations at time $t+1$. A collision occurs when two robots are at the same station. We will assume knowledge of the immediate environment, in the sense that each robot can observe the state (occupied or free) of the two stations following it (in clockwise order), and the two stations preceding it. Initially the system may be in any configuration where no collision occurs.

The specification of the system consists of liveness and safety goals. A liveness goal specifies a particular station to be reached. We will assume that any of the stations can be the target of such a goal. The safety goal prohibits collisions. We now describe a simple social law which obtains this specification:

Traffic Law 1: Each robot is required to move constantly clockwise, from one station to the other along the ring.

It is easy to show that the following holds:

Proposition 1 *Traffic Law 1 guarantees that no collision will occur and that each robot will reach any location it might want to get to in $O(n)$ steps.*

Although very simple (and certainly as a consequence of its simplicity), Traffic Law 1 is also very constraining. It does not leave any choice to the robot when selecting its actions. Notice that from a design perspective it is not enough to require usefulness (that is, meeting the system requirements) from a law, since a useful law might be too tightly related to the specification. There is no need to put a particular restriction on a robot, if this restriction does not interfere with goal achievement by other robots or might lead to unsafe situations. For example, we may wish to allow a robot to move back and forth as it wishes, in order to obtain some temporary new goals, as long as the basic specification can also be fulfilled. Hence, we need to go further and examine the notion of a minimal law.

Note that our study fits nicely in the general model presented in the previous section. Strategies are built using the three basic actions a robot can take at each station (move left, right, or rest). The state of a robot consists of its recent location/observation and its history, and a strategy for a a robot will be a function from its state to action. Goals involve getting from one station to another station on the ring (a liveness goal) as well as avoiding collisions when moving around (a safety goal). We now present a minimal social law for the related setting.

Traffic Law 2:

1. Staying immobile is forbidden if the station which can be reached by a single anti-clockwise movement is occupied.
2. Moving anti-clockwise is allowed only if the two stations which can be reached by moving anti-clockwise *twice* are free.

Proposition 2 *Traffic Law 2 is a minimal (and useful) social law.*

It is easy to check that when the robots follow Traffic Law 2, they can still choose behaviors which are as efficient as the one induced by Traffic Law 1 (according to the basic specification). Still, Traffic Law 2 is to be prefered as it adds maximal flexibility.

4. A computational study of minimal social laws

In this section we initiate a computational study of minimal social laws. We wish to emphasize that we see the design of social laws, and minimal social laws in particular, as an off-line activity. This implies that the automatic synthesis of minimal social laws, its considerable importance notwithstanding, is not the ultimate way to design them. As we showed in the previous section, minimal social laws are central to the design of specific social systems and address issues not necessarily related to computational problems. Nevertheless, a computational study of minimal social laws can shed light both on their design and on their connection to related concepts and issues.

We model a (two-agent) *system* as a tuple $S = (L_1, L_2, c_0, A, A_1, A_2, \tau)$ where L_i is a finite set of states of agent i, $c_0 \in L_1 \times L_2$ is an initial configuration, A is a finite set of actions, A_i is a function from L_i to 2^A that determines the actions that are physically possible for agent i (as a function of its state), and τ is a transition function $\tau : L_1 \times L_2 \times A_1 \times A_2 \rightarrow L_1 \times L_2$.

A plan for agent i is a total function from L_i to A, such that the action prescribed to agent i by the plan at any state $s \in L_i$ is in $A_i(s)$. An execution of a plan $\mathcal{P}$ by agent i is a sequence $s_0, s_1, \ldots, s_k$ of states in L_i, where s_0 is the state of agent i in c_0, and where the s_i's are the states visited by agent i when it follows his plan and the other agent follows one of his possible plans. An execution of a joint plan (consisting of one plan for each agent) is the sequence of configurations reached by following it (by both agents respectively). We assume that each state includes a time stamp, so that an action will lead from a state with time stamp t to a state with time stamp $t+1$.

A liveness goal for agent i is associated with a state $s_{goal} \in L_i$. A *safety* goal is associated with a subset of $L_1 \times L_2$. A plan for agent i is said to *guarantee* a liveness goal s_{goal} if all of its executions include s_{goal}, and the length of the prefix up to the state s_{goal} in each execution is polynomially bounded (in the size of the system, that we take to be $|A| + max_i|L_i|$). A system is said to guarantee a safety goal g_{safe} if there does not exist an execution of a joint plan in the system that includes configurations which are not in g_{safe}.

A social law $\sigma \in \Sigma$ is a set of functions, one for each agent, that restrict the plans available to the agents. Formally, a social law σ consists of functions $\langle A'_1, A'_2 \rangle$, for agents 1 and 2 respectively, where A'_i is a function from L_i to 2^A that defines the subset of actions prohibited for agent i in each state ($A'_i(s) \subseteq A_i(s)$ for every agent i and state $s \in L_i$). A social law σ and a system S induce a *social system* S_σ similar to S, where the A_i functions are altered based on the functions A'_i (i.e., in the state s only actions in $A_i(s) \setminus A'_i(s)$ are allowed). The social law σ is useful if the system guarantees each safety goal (regardless of the law-abiding strategies chosen by the agents), and for every liveness goal s_{goal} of agent i there exists a plan $\mathcal{P}$ in S_σ that guarantees s_{goal}.

Consider the set Σ_S of useful social laws for the system S. We define a partial order $\prec$ on the set Σ_S of useful social laws: given two social laws $\sigma_1 = \langle A_1^{\sigma_1}, A_2^{\sigma_1} \rangle$ and $\sigma_2 = \langle A_1^{\sigma_2}, A_2^{\sigma_2} \rangle$ in Σ_S, we say that $\sigma_1 \prec \sigma_2$ if $A_i^{\sigma_1}(s) \subseteq A_i^{\sigma_2}(s)$ for all i and all $s \in L_i$, with at least one strict inclusion for one s and i. A minimal social law σ_i is a useful social law such that there is no useful social law σ_j in Σ_S, $\sigma_j \neq \sigma_i$, that satisfies $\sigma_j \prec \sigma_i$.

Roughly speaking, the algorithm we have in mind when searching for minimal social laws starts from a useful social law and decrements the set of constraints. We can formulate the decision problem underlying this algorithm as follows: given a system, an appropriate useful social law, and a pair that consists of a state s and an action a prohibited in s for agent i, can we allow i to take action a in s, i.e do we still remain with a useful social law after such an addition? The answer to this question reveals an interesting connection between problems of planning with incomplete information and the design of minimal social laws. We can show:

Theorem 1 *Given a system S, and a useful social law σ that prohibits action a in state s of agent 1, deciding whether by allowing a in s we remain with a useful social law, is NP-hard.*

The above theorem shows that a most basic question in the construction of minimal social laws is NP-hard. The proof of this theorem is not less important than its statement. Although the proof is omitted from this version of the paper, we wish to emphasize that it exposes a connection between the construction of minimal social laws and the problem of Planning while Learning (Safra & Tennenholtz 1994). In Planning while Learning (PWL) we look for a plan which achieves an agent's goal for any environment behavior (while learning on the structure of the environment). As it turns out, by allowing a forbidden action we may move from a situation where goals are guaranteed with simple plans, to a situation where we need to answer the question of whether a goal is achievable regardless of the environment behavior. This situation is captured by a reduction from the PWL-problem (which is known to be NP-complete) to our problem.

The above result leads us to consider a special class of systems. In the sequel, we study the case where an agent's basic goal is to follow a predefined plan $\mathcal{P}_i$. This goal or the agents' capabilities may change over time and therefore, we wish to avoid the much too restricted law which would oblige each agent to follow this and only this plan. We show that an efficient incremental algorithm for the computation of minimal social laws exists for this class of systems.

Given a system, consider a pair of plans $\mathcal{P}_1$ and $\mathcal{P}_2$, where $\mathcal{P}_i$ is a plan for agent i. Let t_i be a bound on the number of steps of $\mathcal{P}_i$. Let $(s_0^i, s_1^i, \ldots, s_{t_i}^i)$ be the execution of the joint plan $(\mathcal{P}_1, \mathcal{P}_2)$ projected on the states of agent i. We will associate this execution with the goal of agent i. In the AGVs example this sequence can be associated with a sequence of stations $l_1, l_2, \ldots, \ldots$ where l_{i+1} and l_i are neighboring stations. Basically, this kind of goals is quite typical in systems where agents must follow given protocols. Such systems also provide an excellent illustration of the role of social systems, namely to allow maximal freedom by relaxing unnecessary constraints.

The Minimal Social Law Algorithm (MSLA)

1. Let s_k^i denote the k-th state to be visited in the execution that corresponds to the achievement of the original goal. Let s_0^i denote the initial state of agent i. Let $(a_0^i, \ldots, a_{t_i}^i)$ be the sequence of actions executed by agent i in the corresponding plan (P_i).

2. Initialization step: $k = 0$ and $A'_i(s) = \emptyset$ for all i and s.

3. Let B_k^j be the set of reachable states at step k for agent j when agent $3-j$ follows his (original/basic) plan. Initially, B_0^1 contains the initial state of agent 1 in c_0, B_0^2 contains the initial state of agent 2 in c_0, and all the other B_k^j's are empty sets.

4. For each state $s \in B_k^1$ and for each action $a \in A_1(s) \setminus A'_1(s)$, if $\tau(s, s_k^2, a, a_k^2) = (s^1, s_{k+1}^2)$, $s^1 \in L_1$, then add s^1 to B_{k+1}^1. Otherwise, add a to $A'_1(s)$.

5. Increment k. While $i \leq t_i$, go to 4.

6. Execute steps 4–5 again, switching the indexes for the agents.

7. Output the social law whose components are the functions A'_1, A'_2.

Proposition 3 *Given a system S, and a goal that is a projection of a given joint plan, MSLA outputs a minimal social law for the system S in polynomial time.*

Notice that the algorithm computes a law in which the original plan of each agent remains available. Hence the law maximizes flexibility in the selection of behaviors while leaving the original behavior intact.

The AGVs case study presented in Section 3 can be represented and solved within our model. We now describe a sketch of how this can be done. Each robot has a finite set of states L_i which is a product of two components. Each state of one of the components refers to a station the AGV may be in, and each state of the other component refers to a station it may be in and observations it may have (i.e., in this second component a state refers both to a station and to the local observations made). There is an initial state where the robot selects once and for all whether it will observe only its station or the neighboring coordinates as well (i.e, a decision about the component of states to be visited). There is also a distinguished state that denotes collision. The goal is to follow a particular path along the ring without (the need to) observing the neighborhood, and without colliding. This can be easily defined by a path of states of the first component. The transition function will capture movements in the ring topology and potential collisions. The MSLA algorithm can be used now in order to build a minimal social law that is similar to the one presented in Section 3.

Further work: In the full paper we discuss minimal social laws in the context of a typical consensus problem. In difference to the AGVs case study, this part of our study deals with non-symmetric laws, which allow the assignment of different roles to the agents. The study illustrates subtle points in the inter-relationships between minimality and usefulness. Notice that although our work has been concerned with specifications of liveness and safety conditions, one can discuss a situation where efficiency stated as a cardinal measure over the goal space (not expressible in the above general form) is taken into account. This calls for work on tradeoffs between efficiency and minimality, which we hope to pursue in the future.

References

Ben-Yitzhak, O., and Tennenholtz, M. 1997. On the Synthesis of Social Laws for Mobile Robots: A Study in Artificial Social Systems (Part I). *Computers and Artificial Intelligence* 14.

Bond, A. H., and Gasser, L. 1988. *Readings in Distributed Artificial Intelligence.* Ablex Publishing Corporation.

Briggs, W., and Cook, D. 1995. Flexible Social Laws. In *Proc. 14th International Joint Conference on Artificial Intelligence*, 688–693.

Demazeau, Y., and Muller, J. 1990. *Decentralized AI.* North Holland/Elsevier.

Durfee, E. 1992. What your computer really needs to know, you learned in kindergarten. In *10th National Conference on Artificial Intelligence*, 858–864.

Kraus, S., and Wilkenfeld, J. 1991. The Function of Time in Cooperative Negotiations. In *Proc. of AAAI-91*, 179–184.

Kreps, D. 1990. *A Course in Microeconomic Theory.* Princeton University Press.

Lansky, A. L. 1988. Localized Event-Based Reasoning for Multiagent Domains. Technical Report 423, SRI International.

Latombe, J. 1991. How to move (physically speaking) in a multi-agent world. In *MAAMAW91.*

Minsky, N. 1991a. The imposition of protocols over open distributed systems. *IEEE Transactions on Software Engineering* 17(2):183–195.

Minsky, N. 1991b. Law-governed systems. *Software Engineering Journal* 285–302.

Moses, Y., and Tennenholtz, M. 1990. Artificial Social Systems Part I: Basic Principles. Technical Report CS90-12, Weizmann Institute.

Moses, Y., and Tennenholtz, M. 1995. Artificial Social Systems. *Computers and Artificial Intelligence* 14(6):533–562.

Onn, S., and Tennenholtz, M. 1997. Determination of Social Laws for Agent Mobilization. *Artificial Intelligence.* 95:155–167.

Rosenschein, J. S., and Zlotkin, G. 1994. *Rules of Encounter.* MIT Press.

Rousseau, J. 1762. *Du Contrat Social.* Translated and introduced by Maurice Cranston (1968). First published in 1762. (also published by Dent Everyman along with the Discourses).

Safra, S., and Tennenholtz, M. 1994. On Planning while Learning. *Journal of Artificial Intelligence Research* 2:111–129.

Shoham, Y., and Tennenholtz, M. 1995. Social Laws for Artificial Agent Societies: Off-line Design. *Artificial Intelligence* 73.

Sinreich, D. 1995. Network design models for discrete material flow systems: A literature review. *International Journal of Advanced Manufacturing Technology* 10:277–291.

Stuart, C. 1985. An Implementation of a Multi-Agent Plan Synchronizer. In *Proc. 9th International Joint Conference on Artificial Intelligence*, 1031–1033.