Social Pathologies of Adaptive Agents

David Jensen and Victor Lesser

Department of Computer Science
University of Massachusetts
Amherst, MA 01003-4610
{jensen|lesser}@cs.umass.edu

From: AAAI Technical Report SS-02-07. Compilation copyright © 2002, AAAI (www.aaai.org). All rights reserved.

Abstract

We describe, briefly characterize, and give examples of social pathologies of multiagent systems. Social pathologies are system behaviors in which two or more agents interact such that improvements in local performance do not improve system performance. They are widely observed in both natural and artificial systems.

Introduction

Multiagent systems do not always behave as their designers intend. In some cases, a simple flaw in design or implementation causes sub-optimal behavior, and such flaws are often easy to remedy. In other cases, sub-optimal behavior is caused by fundamental characteristics of system structure or problem representation, and these flaws can be difficult to understand and avoid. Such cases are often called *pathologies*.

Several pathologies of single-agent systems have been studied extensively. For example, game-tree search algorithms exhibit pathological behaviors in certain, limited classes of games (Nau 1983, Kaindl 1988). Similarly, many induction algorithms exhibit overfitting, a tendency to produce models that contain excessive structure (Jensen and Cohen 2001, Oates and Jensen 1997).

Our research focuses on a particular class of pathologies of multiagent systems — *social pathologies of adaptive agents*. Such pathologies are adaptations of individual agents that are locally beneficial but that degrade the performance of the overall system. Related phenomena have also been called *social dilemmas* (Glance 1994, Glance and Hogg 1995) or *social traps* (Etcheson 1989). Social pathologies are particularly troublesome because they contradict a central assumption of multiagent systems — that local performance improvements by one or more agents will lead to system-wide performance improvements.

Below, we define social pathologies and analyze the conditions under which they arise. We provide several examples of social pathologies and draw examples from diverse multiagent systems. Finally, we briefly discuss how understanding social pathologies can improve our knowledge of other aspects of multiagent system behavior.

A Framework for Adaptive Multiagent Systems

Consider a multiagent system $\mathcal{S}$ composed of n agents $\mathcal{A} = a_1, a_2, \ldots, a_n$ and an environment $\mathcal{E}$. An agent a_i is defined by a set of state variables. Subsets of those variables define different aspects of an agent, including its local resources, knowledge, and inference mechanisms. At most, an agent's actions depend on its own state, the environment $\mathcal{E}$, and the states of other agents in $\mathcal{A}$. In general, however, an agent's reasoning is bounded in one or more ways: it may have access to only partial information about other agents and the environment or it may have only limited representational or reasoning abilities. Like the agents, the environment $\mathcal{E}$ is defined as a set of state variables. Subsets of those variables define different aspects of the environment, such as the shared resources available to one or more agents.

Any individual agent a_i, as well as the entire system $\mathcal{S}$, can be characterized by its performance P. For a single agent, $P_i = f(\mathcal{A}, \mathcal{E})$. That is, local performance is a function of the state of the agent, *all other agents*,[1] and the environment. Similarly, for the system, $P_{\mathcal{S}} = f(\mathcal{A}, \mathcal{E})$.

Agents can contain learning mechanisms by which they alter internal state variables in order to maximize local performance. There are many pathologies of learning in individual agents, but we do not consider them here. Instead, we focus on situations where agents succeed in maximizing their local performance through learning.

In many systems, system performance is some additive or multiplicative function of agent performance. In general, the measure of system performance is often assumed to obey a fundamental assumption:

> *The local improvement assumption:* If the performance of all other agents remains unchanged, then improvements in local performance by one or more agents will improve system performance.

For example, perhaps the simplest system performance function is a simple additive function:

[1] Note that $P_i = f(\mathcal{A}, \mathcal{E})$, rather than $f(a_i, \mathcal{E})$. That is, the states of all other agents in the system partially determine the performance of the agent a_i. These agents help form the environment within which a_i operates.

$$P_s = \sum_i P_i$$

Designing effective multiagent systems is largely about upholding the local improvement assumption. In well-designed systems, changes in information and decisionmaking that improves the performance of a single agent also improves the performance of the entire system. There is often no guarantee that system performance will reach a global maximum, but many multiagent systems are based on the assumption that local performance improvements will, at a minimum, improve system-wide performance.

Clearly, there are many relatively simple violations of the local improvement assumption. For example, agents can have characteristics that interfere with the local improvement assumption either unintentionally (e.g., a poorly-designed measure of local performance) or intentionally (e.g., computer viruses).

This paper deals with *social pathologies* — more complex behaviors that violate the local improvement assumption.

> *Social pathology:* a system behavior in which two or more agents interact such that improvements in local performance do not improve system performance.

Social pathologies are *emergent* properties of mulitagent systems — system behaviors that cannot be examined at the agent level, but can only be examined at the system level.

Types of Social Pathologies

How do social pathologies arise? Under what conditions can interactions change an agent's state, improve its local performance, but degrade system performance? Recall that $P_i=f(\mathcal{A},\mathcal{E})$ and $P_s=f(\mathcal{A},\mathcal{E})$. For simultaneous improvement in one or more agent's performance and degradation in system performance, either the agents $\mathcal{A}$ or the environment $\mathcal{E}$ must change. If we assume the environment is free from external fluctuations, then either the agents themselves must change or they must change environment.

In the remainder of this section, we discuss four pathologies in terms of how a agent interactions can adversely affect system performance. First, we discuss a pathology where a change by one agent affects the environment $\mathcal{E}$ of all agents (*tragedy of the commons*). Next, we discuss a pathology where a change by one agent affects the aggregate characteristics of all agents $\mathcal{A}$ (*lock-in*). Finally, we discuss two pathologies where a change by one agent affects the behavior of one or more other agents (*cycling* and *blocking*).

Tragedy of the Commons

The environment $\mathcal{E}$ is one determinant of both agent and system performance. A social pathology will result if an agent improves its own performance by adversely affecting the environment for other agents. This is the basis of the Tragedy of the Commons (*TOC*).

In *TOC*, an agent improves its own performance by making greater use of one part of the environment — typically a shared resource. The shared resource responds to the increased demand by *more than* proportionately decreasing the value it provides to all other agents. Typically, this behavior results from the lack of excess capacity in the resource, and transaction costs associated with increased use.

Consider the hypothetical example of a processor shared by ten agents. The processor has a capacity of 100 units (measured on some consistent scale of computational resources), with 2 units/agent devoted to the costs of context switching among the different processes. As a result, 80 units are spent on useful computation. Because each agent requires 8 units, the processor is used at precisely its current capacity. If an eleventh agent begins using the processor, a total of 100-(11*2) or 78 units will be spent on useful computation. Because the processor is being used at capacity and transaction costs exist, an additional agent reduces the total useful computation the processor can perform (despite a local improvement in the new agent's performance due to its use of the processor).

TOC is probably the best-know pathology of multiagent systems. It is most commonly understood in the context of natural resource systems. For example, Hardin's classic paper (Hardin 1968) uses the example of a common grazing land for cattle that is shared by several herdsmen. Each herdsman makes choices about how many cattle to graze on the common land. In such cases, the common land is almost always overused, resulting in lowered overall yields. Another example of *TOC* has been labeled "the diner's dilemma" (Glance 1994). Each of several dining companions orders a meal from the menu and then the bill is split evenly among all parties. An extravagant diner receives all the benefits of ordering an expensive meal, but bears only a small portion of the costs. In such cases, individuals are well served by ordering an expensive meal, yet the costs skyrocket if all diners follow this policy.

Analogous situations are common in computational multiagent systems. Systems can have many shared resources, such as processors, memory, long-term storage, external sensors, and communications channels. For example, Sugawara and Lesser (1994) describe a shared communication channel that quickly becomes overloaded because of the uncoordinated actions of several agents. Turner (1993) provides a number of other examples.

Counterexamples also exist. One can easily imagine the inverse of *TOC* — what might be called the "blessing of the commons" — a situation in which an agent's actions improve both local and system performance. For example, blackboard systems demonstrate this behavior. The data collection and reasoning efforts of a single agent are posted to a central repository, potentially increasing the performance of all other agents.

One common solution to *TOC* is to communicate the true cost of using shared resources via a system of prices. Wellman (1993) explores the use of an artificial economy to derive the activities and resource allocations for a set of computation agents. This approach has also been employed in human societies when shared resources have been overexploited. Pollution taxes, license fees, and auctions of public goods have all been employed to allocate resources that are both scarce and shared to their most useful purpose and communicate the true cost of resources to the agents that use them.

Lock-in

Some system properties that determine performance may not be derivable from individual agents. One example of such a system property is agent diversity. Diversity can be an important determinant of long-term system performance by increasing a system's ability to adapt to sudden environmental changes or external attack. For example, biodiversity is widely regarded by biologists as a necessary condition for ecosystem health. Similarly, recent work in computer science has examined the importance of diversity in operating systems and security measures to combat viruses and direct attack. However, some systems reward individual agents for conforming to a prevailing standard, reducing diversity and degrading system performance. This is the essence of *lock-in*.

Lock-in occurs when individual agents can improve their performance by conforming to some aspects of another agent's state. One agent may adopt another agent's beliefs or reasoning techniques. Sociologists refer to the incentive to conform as "social pressure" and to the outcome as "groupthink." Economists sometimes call the pressure to conform a "network externality" and sometimes refer to the outcome as a "monopoly."

In addition, small perturbations in the initial states of agents can lead to selecting suboptimal beliefs and reasoning techniques. If the local performance associated with a set of beliefs depends partially on the extent to which they are shared by other agents, then the initial state of all agents can determine which belief system prevails. This can lead to a suboptimal state if initial conditions favor a belief system that, when adopted by all agents, has lower performance than another. This sort of *lock-in* has been recently identified in economic systems (Arthur 1990).

Cycling

Changing state can require significant computational resources from an agent, disrupt previously stable agent behaviors, and require other agents to change state as well. This can degrade system performance until a new equilibrium is reached. *Cycling* is a pathology where no new equilibrium is reached.

In *cycling*, a system's agents engage in a series of ultimately circular state transitions. This can proceed indefinitely, or end in a state with the same performance as the initial state. *Cycling* occurs in systems where one agent's state affects the relative performance of other agents' states. An initial state change by one agent increases the desirability of a state change by another agent. *Cycling* consumes computational resources without producing any corresponding benefit.

Arms races are a frequently-studied form of *cycling* in international politics (Etcheson 1989, Gleditsch and Njolstad 1990). Nations often measure the performance of their national defense policy in terms of their security with respect to a potential aggressor. Thus, security depends not only on a nation's environment (e.g., its geographic position) but also the state $\mathcal{A}$ of other agents in the system. When one nation changes the state of its military forces (e.g., by developing a new weapons system), other nations may perceive their security to be reduced, and change state as well (by developing a similar or countervailing weapons system). This, in turn, changes the security of the first nation. Such arms races can proceed for years or decades, with both nations consuming large amounts of resources but attaining no greater security.

Similarly, arms races have been observed in biological systems on evolutionary time scales (Dawkins 1986, Dawkins and Krebs 1979, Van Valen 1973). The performance of predators depends on their prey and vice versa. This can lead to a string of evolutionary adaptations in predators (e.g., faster running speed, sharper teeth, keener eyesight) and in prey (e.g., faster running speed, harder shells, and better camouflage, respectively). While these changes in individual species could be termed "improvements," they often produce no net change in the relative advantage among predators and prey.

Cycling is related to, but distinct from, two well-known computational phenomena. Early networks sometimes reached a state where messages endlessly cycled among a finite set of routers, never reaching their final destination. This phenomenon, sometimes called "ping-ponging," produces a cycle, but not a cycle with respect to agent state; the messages cycle, not the routers' behavior. Another phenomenon similar to *cycling*, often called "thrashing," occurs in time-sharing systems that accomplish little useful computation because the system spends nearly all it cycles shifting among multiple processes. Thrashing captures the aspect of looping behavior and poor performance, but the looping is an intentional part of a time-sharing system and the behavior associated with each process does not change.

Blocking

Environments can impose limitations on agent states and their relative performance. For example, only a fixed number of agents may be able to fulfill a particular role in a system, or the performance associated with a particular agents state may depend on the states of other agents in the system. This characteristic of environments and agents can lead to *blocking*.

In *blocking*, one or more agents occupy a particular role in a system. Because they occupy that role, other agents cannot, even though system performance would improve under a different arrangement of agents and roles. *Blocking* occurs because of how the environment structures performance, and because of the current states of agents. Given other states, or a more flexible environment, system performance would improve.

For example, biological systems can exhibit *blocking* due to the presence of ecological niches. In the short term, a species may not be able to extend its geographic range into a new region because its particular niche is filled by another species in that region. In the long term, a particular evolutionary path may be prevented because another species fills the niche into which another species might move. In human societies, *blocking* can occur in terms of positions within an organization (two employees often cannot fill the same position) and in terms of marital pairings (bigamy is outlawed). A computational model that exhibits *blocking* has been created by Numaoka (1995).

Discussion

We have described only a few of a potentially large number of social pathologies associated with multiagent systems. The examples of *tragedy of the commons*, *lock-in*, *cycling*, and *blocking* illustrate the ways in which the local improvement assumption can be false. They highlight this assumption as an important property of multiagent systems. Systems which guarantee the validity of the local improvement assumption are "safe" — they cannot exhibit pathological behaviors.

Unfortunately, it is not clear how the local improvement assumption can be guaranteed without sacrificing the promise of multi-agent systems. For example, one obvious way of guaranteeing that local improvement leads to global improvement is for the actions of each agent to be independent. That is, the actions of each agent cannot affect the performance of other agents. However, this also removes the possibility of beneficial interactions among agents, the *sine qua non* of multiagent systems.

Instead, designers of multiagent systems should be alert to the possibility of social pathologies in adaptive agents, and researchers should continue to identify, investigate, and catalog them. We hope this paper provides a first step in this direction.

Acknowledgments

This research is supported by The Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratoy, Air Force Material Command, USAF, under agreement number F30602-00-2-0597. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of the The Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory or the U.S. Government.

References

Arthur, B. 1990. Positive Feedbacks in the Economy, *Scientific American.* February.

Cross, J. and Guyer, M. 1980. *Social Traps.* Ann Arbor: University of Michigan Press.

Dawkins, R. 1986. The Blind Watchmaker. New York: W.W. Norton.

Dawkins, R. and Krebs, J. 1979. Arms races between and within species. *Proceedings of the Royal Society of London B* 205:489-511.

Etcheson, C. 1989. *Arms race theory: Strategy and structure of behavior.* New York: Greenwood.

Gleditsch, N. and Njolstad, O. eds. 1990. *Arms races: Technological and political dynamics.* London: Sage Publications.

Glance, N. and Huberman, B. 1994. The dynamics of social dilemmas. *Scientific American.* March. 76-81.

Glance, N. and Hogg, T. 1995. Dilemmas in computational societies. In Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95). Menlo Park, California: AAAI Press. 117-124.

Hardin, G. 1968. The tragedy of the commons. *Science* 162:1243-1248.

Jensen, D. and Cohen, P.R. 2001. Multiple comparisons in induction algorithms. *Machine Learning* 38(3):309-338..

Kaindl, H. 1988. Minimaxing: Theory and practice. *AI Magazine* 9(3): 69-76.

Nau, D. 1983. Pathology on game trees revisited, and an alternative to minimaxing. *Artificial Intelligence* 21:221-244.

Numaoka, C. 1995. Introducing the blind hunger dilemma: Agents' properties and performance. In Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95). Menlo Park, California: AAAI Press. 290-296.

Oates, T. and Jensen, D. 1997. The effects of training set size on tree size. Proceedings of the Fourteenth International Conference on Machine Learning. 254-262.

Sugawara, T. and Lesser, V. 1998. Learning to improve coordinated actions in cooperative distributed problem-solving environments. *Machine Learning* 33:129-153.

Turner, R. 1993. The tragedy of the commons and distributed AI systems. In Twelfth International Workshop on Distributed Artificial Intelligence, Hidden Valley, PA, 1993. 379-390.

Van Valen, L. 1973. A new evolutionary law. *Evolutionary Theory* 1:1-30.

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