Social Dilemmas in Computational Ecosystems

Tad Hogg
Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304, U.S.A.
hogg@parc.xerox.com

Abstract

Computational ecosystems are large distributed systems in which autonomous agents make choices asynchronously based on locally available information which can be uncertain and delayed. They share these characteristics with biological ecosystems, human societies and market economies. We show that, even when designed with a single overall goal in mind as in the case of distributed problem solving, computational ecosystems can face well-known social dilemmas of sustaining cooperative behavior among selfish agents. Specifically, public-goods problems, where a common good is available to all regardless of individual contribution, can arise due to information limitations as well as the commonly recognized incentive conflicts. Some techniques for mitigating the impact of these problems are also presented.

1. Introduction

Effective use of distributed computation is challenging since the individual processes or agents must obtain resources in a dynamically changing environment and collaborate despite a variety of asynchronous and unpredictable changes. Because centralized control is often unable to respond rapidly to local changes in these systems, the agents are often designed to act autonomously using locally available information. Such decentralized systems are often more robust and simpler to design and incrementally modify than those using a central controller. On the other hand, these agents face the difficult task of performing well in spite of incomplete, imperfect and changing information. Because these characteristics are shared with biological ecosystems and human societies, we refer to these systems as *computational ecosystems* [Hogg, 1994; Huberman and Hogg, 1988; Gasser and Huhns, 1989; Miller and Drexler, 1988; Waldspurger et al., 1992; Wellman, 1993; White, 1994].

This analogy between computational and human societies offers suggestions for the appropriate design of agents as well as new potential problems not seen with centralized control or with a small number of agents. In this paper we show that one such problem, namely the social dilemma involved in maintaining cooperation among a large group of self-interested agents, can readily arise even when the agents are designed with a single overall goal. This situation is unlike most studies of distributed artificial intelligence (DAI) that consider agents with possibly conflicting goals and focus on designing incentives to encourage cooperation [Gasser and Huhns, 1989]. In the situation reported here, social dilemmas can still arise even with full control of the agents' overall incentives. Thus issues raised by DAI have a broader range of applicability than might initially appear to be the case. While social dilemmas can adversely affect a group's performance, we also show how they can be addressed to some extent with methods used in human societies.

This paper is organized as follows. The next two sections summarize the social dilemmas for cooperation and a model for the dynamical behavior of computational ecologies. Section 4 gives the main new result, showing how the dilemmas can readily arise in computational ecosystems, even those with a single overall goal, and how this lowers performance. In some cases, this can lead to the paradoxical situation in which giving more resources to the system results in lowered overall performance, as shown in section 5. Some possible ways to address this problem, and concluding remarks are given in the remaining two sections.

2. Social Dilemmas

One of the most challenging problems for societies of autonomous agents is providing for public goods [Hardin, 1968], i.e., benefits produced by the society and available to all of its members regardless of individual contribution. Examples of public goods in human societies include provision of parks, roads, a clean environment and national defense. When there is a cost involved in contributing, there is the temptation for agents to free ride on the efforts of others; but if everyone reasons this way, no public good is produced, lb the extent that the individual costs are less than the public good's benefit to the whole group, individually rational behavior leads to an overall suboptimal result This is the social dilemma involved in maintaining cooperation for the production of such goods.

While it is easy to see how such problems arise computationally when agents have different goals [Rosenschein and Zlotkin, 1994], when there is a single overall goal for the system (e.g., in distributed problem solving) one might hope to avoid social dilemmas entirely by explicitly programming desired cooperative behaviors into the agents. Recently, however, it has been recognized that such dilemmas can occur in contexts such as coevolutionary genetic algorithms with a single overall goal [Glance and Hogg, 1995]. This is an example of the general mathematical fact that locally optimal actions can result in suboptimal global performance. At the same time, determining the globally superior choice by central fiat may be computationally intractable. In some cases, the optimal state may actually be provably impossible to find within a distributed setting since individual choices will take the system away from the optimum instead of towards it This is the computational analogue of a social dilemma. In this case the dilemma is due to information limits rather than incentive problems, but has the same detrimental effect Social dilemmas stand in contrast to other causes of difficulty in computational problems. For instance, hilly landscapes cause methods such as neural nets [Rumelhart et al., 1986], simulated annealing [Kirkpatrick et al., 1983] and greedy heuristics for NP-hard search problems [Garey and Johnson, 1979] to become trapped in local minima, making the overall minimum hard, but not impossible, to find. Despite these observations, one might hope computational social dilemmas are exceptional and so rarely seen in practice. The major new finding of this paper, given in Section 4, is that these dilemmas are indeed readily found in computational ecosystems.

3. Dynamics of Computational Ecosystems

Computational ecosystems consist of agents, with computational tasks to perform, and various resources with which to accomplish their tasks. These resources can include both hardware and software (e.g., information from sensors, databases and the use of specialized algorithms). A simple model of these societies supposes each agent uses one resource at a time, evaluates its choice at a rate a and selects that resource it perceives to be best based on locally available information. The suitability of a resource can depend on how many other agents are using it, leading to a range of dynamical behaviors including simple equilibria, continual oscillations and chaos [Kephart et al., 1989].

Specifically, the state of a system with *r* resources at a given time can be characterized by the fraction of agents fi using resource *i* with

$$\sum_{1=1}^{r} f_i = 1 \quad (1)$$

Let c, be an agent's cost of using the resource, which can depend on the state of the system. The overall average cost per agent is then given by $C_{global} = \sum f_i c_i$., This provides one simple measure of global performance for the system and is appropriate for situations in which the agents are each contributing to the result More complex global measures can arise in other situations such as cooperative problem solving in which a single solution is required and so the global performance is determined by the time for the first agent to finish [Clearwater et al., 1991].

The dynamical behavior, on average, is given by [Huberman and Hogg, 1988]

$$\frac{df_i}{dt} = -\alpha(f_i - \rho_i) \quad (2)$$

where pi, is the probability that resource i will be perceived to be the best one, i.e., have lowest cost, when an agent makes a choice. Note that pi = 1 and that the p- depend on the /, because the resource costs do. The two terms on the right side of Eq. 2 are readily understood as 1) a decrease in use of resource i due to agents already using it making choices at a rate a, and 2) an increase in the use of resource *i* due to agents perceiving it to be the best one when they make their choices.

The simplest case is when the agents have full information on their individual costs and there are no delays in the information. In that case they will always select that resource with the lowest actual cost giving *pi* = 1 when ci is the lowest cost, and zero otherwise. Uncertainty and delays in the information give well-studied mechanisms that degrade performance [Kephart et al., 1989]. Here we focus instead on the perfect information case to highlight the suboptimal behavior due explicitly to the social dilemmas.

4. How Dilemmas Arise

Having introduced the possibility of social dilemmas and the dynamical model for computational ecosystems, we now turn to the question of how often these dilemmas can be expected to occur. Although ideally approached empirically once many such systems become available for study, we can gain considerable insight from examining the kinds of resource costs that will give rise to a social dilemma. Specifically, this will happen when the local dynamics does not lead to the globally optimal state even when the agents have perfect local information. In such cases, the performance of the group as a whole would benefit if some agents "cooperated" by making different resource choices even though they would incur higher costs.

4.1 Theory for Linear Costs

Consider the situation with *r* resources and with costs depending linearly on their utilization:

$$\mathbf{c} = M\mathbf{f} + \mathbf{v} \quad (3)$$

with the vectors c and f having components ci and fi, giving the cost and utilization of the resources. The matrix *M* specifies how the costs depend on the utilization, and the vector v is a usage-independent contribution to the costs. Note that because of Eq. 1, the choices for A/ and v are not unique, e.g., this relation can be used to eliminate f_r from the costs so the last column of *M* will be zero. Furthermore, because agents make choices based on the minimum cost option (when they have perfect information), the results presented here will not be affected by rescaling the unit of cost, i.e., multiplying each component of M and v by a constant, or by shifting the origin of the costs, i.e., adding a constant to each component of v.

The global average cost is given by

$$C_{global} = \sum f_i c_i = \mathbf{f}^T M \mathbf{f} + \mathbf{f} \cdot \mathbf{v} \tag{4}$$

From this we see that the linear cost example considered here is particularly useful for investigating social dilemmas: except in degenerate special cases, C_{global} has only one minimum and so avoids the complication of local minima. The resource usage that minimizes this cost is found by minimizing $C_{global} - \lambda(\sum f_i - 1)$, where the Lagrange multiplier λ is used to enforce Eq. 1. Setting all partial derivatives to zero gives the extremum as the solution to

$$A_G \begin{pmatrix} \mathbf{f} \\ -\lambda \end{pmatrix} = \begin{pmatrix} -\mathbf{v} \\ 1 \end{pmatrix} \tag{5}$$

with

$$A_G = \begin{pmatrix} M^T + M & \mathbf{1} \\ \mathbf{1} & 0 \end{pmatrix} \tag{6}$$

where 1 represents the vector all of whose components equal 1. There are two degenerate cases in which this extremum will not in fact be the global minimum cost. First, while this is the only point where the partial derivatives vanish, it could be a maximum or saddle point instead of a minimum. Second, the extremum may not have each $f_i > 0$. In either case, the true minimum cost will be found at the boundary of the allowed set of values, i.e., at least one resource is not used at all. In effect, such systems can be viewed as having fewer resources.

By contrast, the equilibrium found by local choices will be where all resources have the same cost so agents will remain where they are, on average. That is $c = \mu\mathbf{1}$ where μ is a constant, which gives $C_{global} = \mu$. This can be combined with Eq. 1 to obtain

$$A_L \begin{pmatrix} \mathbf{f} \\ -\mu \end{pmatrix} = \begin{pmatrix} -\mathbf{v} \\ 1 \end{pmatrix} \tag{7}$$

with

$$A_L = \begin{pmatrix} M & \mathbf{1} \\ \mathbf{1} & 0 \end{pmatrix} \tag{8}$$

whose solution gives the equilibrium state found by local choices, again provided $f_i \geq 0$ (otherwise, some resources, with relatively high costs, will not be used in equilibrium).

4.2 The Existence of Social Dilemmas

To see that social dilemmas can be expected quite frequently in computational ecosystems, consider a typical situation in which the cost to use a resource increases with its use, with some fixed overhead, but is not directly dependent on the

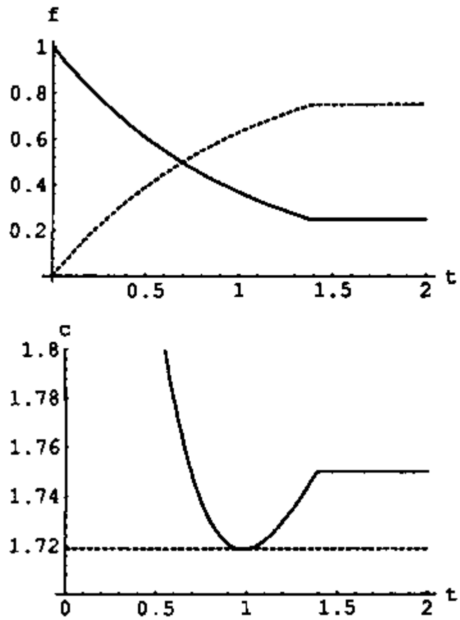

Fig. 1. Behavior as a function of time for a 2–resource computational ecosystem with a social dilemma. We use $f_1(0) = 1$ and $\alpha = 1$. The costs are $c_1 = f_1 + \frac{3}{2}$ and $c_2 = f_2 + 1$. The top figure shows the fraction of agents using resources 1 (solid) and 2 (dashed). On the bottom, the solid curve is the global average cost C_{global} while the dashed one is the minimum cost, achieved when $f_1 = \frac{3}{8}$.

use of other resources. That is $c_i = f_i/a_i + v_i$ with $a_i > 0$ so that M is a diagonal matrix with $m_{ii} = 1/a_i$. The global minimum, from Eq. 5, is at $f_i/a_i = s + \frac{1}{2}(\bar{v} - v_i)$, with costs $c_i = s + \frac{1}{2}(\bar{v} + v_i)$. In these expressions we defined $s = 1/\sum a_k$ and $\bar{v} = s \sum a_k v_k$ as a weighted average of the components of v. This gives

$$C_{global} = s + \bar{v} - \frac{1}{4} \sum a_i (\bar{v} - v_i)^2 \tag{9}$$

By contrast, the local equilibrium from Eq. 7 is $f_i/a_i = s + \bar{v} - v_i$, with cost $c_i = s + \bar{v}$, which is also the value of C_{global} at this equilibrium In both cases, a negative resulting value for f_i indicates resource i isn't used and the calculation should be repeated without it to find the correct resource use.

We thus see that the local equilibrium is globally optimal only if all v_i are the same, as was the case in a limited previous study of the 3-resource case [Kephart et al., 1989]. Otherwise, the local equilibrium point will have wider range of resource use than is optimal. Thus we see that generally, even with resources whose costs depend only on their own load, the public goods social dilemma arises.

A simple example, with two resources, is shown in Fig. 1 with the dynamics given by Eq. 2, $M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and

$\mathbf{v} = \begin{pmatrix} 3/2 \\ 1 \end{pmatrix}$. The global minimum cost is at f_1 = | but the local equilibrium is at the suboptimal value f_i = 1/4. Even though, in this case, the system passes through the optimal state during its evolution, more agents continue selecting to use resource 2. This results in the final state having a higher cost, as shown in the figure.

The basic cause of the problem is agents choose based only on their own costs, not on how their actions affect others. For example, if there are 1000 agents at the global minimum (i.e., 375 agents using resource 1), then the costs are $c_1 = \frac{15}{8}$, $c_2 = \frac{13}{8}$ and agents using resource 1 are tempted to switch. Suppose one agent moves from resource 1 to 2, reducing its own cost by about 0.25. This move also affects the remaining agents: the cost of resource 1 drops slightly, thereby benefiting the 374 agents that remain there, but the cost of resource 2 increases slightly, harming the 625 already there. The net result in this case is that the increased cost to the agents already using resource 2 outweighs the benefit both to the agent that moved and those remaining on resource 1: the global cost increases slightly.

43 Generalizations

In the general case with nondiagonal matrix M, we can see that dilemmas will also often arise as follows. In order that Eqs. 5 and 7 give the same value for f, we must have

$$(A_G^{-1} - A_L^{-1})\begin{pmatrix} -\mathbf{v} \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -\lambda + \mu \end{pmatrix} \tag{10}$$

which is a linear equation for v. In general, most vectors v will not satisfy this equation, and hence the global minimum and the local equilibrium will be at different points. Thus on very general grounds we can expect the results seen explicitly above for simple diagonal matrices to apply to a wide range of linear cost functions.

An example is where resource 1 uses a communication channel controlled by resource 2, so agents using resource 1 incur higher cost when resource 2 is busy with its own tasks and unable to service the requests. Such an instance is

$$M = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \mathbf{v} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{11}$$

which has global minimum at / = (1/2,1/2) with cost 3/4, but local equilibrium at f = (0,1) with cost 1. To avoid the dilemma, Eq. 10 requires that $v_1 = v_{2-1}$.

Finally, for the more realistic case of nonlinear cost functions, the analysis cannot give a complete characterization of the dynamics (e.g., there may be local minima), but can provide some insight. The dilemma will exist (in addition to any problems due to local minima) when the local dynamics drives the system away from the global minimum, even if it is initially quite close. Whether this is the case can be determined by expanding Eq. 2 around the global minimum, giving a linearized system to which the above analysis applies. Because we saw that dilemmas exist quite commonly with linearized costs, this argument shows they will correspondingly exist for many nonlinear problems. How important the dilemmas will be in such cases will depend on how easily the system is stuck in local minima. That is, if there are many local minima, they can be expected to prevent the system from getting near the global minimum at all and hence the existence of a dilemma that prevents convergence to the global minimum will rarely affect the dynamics. On the other hand, systems with fewer local minima, or that have techniques to avoid or overcome them, would be limited instead by a social dilemma. This suggests that an interesting question for future work is to determine the interplay between these effects, and whether strategies that alleviate one problem increase the other.

5. Braess' Paradox

A particularly subtle version of the dilemma is given by Braess' Paradox [Cohen and Horowitz, 1991; Irvine, 1993] in which adding resources to a system can actually lower performance. While usually presented in the context of traffic or electric current flow, it has a natural interpretation as selection among resources. A specific example, which corresponds to that considered for traffic and genetic algorithms [Glance and Hogg, 1995] is given by the three resource case with the costs

$$\begin{aligned} c_1 &= 3 - f_2 \\ c_2 &= 3 - f_1 \\ c_3 &= 3 - f_1 - f_2 \end{aligned} \tag{12}$$

When only the first two resources are available, *p1* = 1 when $c_1 < c_2$, corresponding to $f_1 < f_2$, and similarly *p2* — 1 when $f_1 > f_2$. Thus, starting from an initial condition of all agents using resource 1, i.e., $f_1(0) = 1$, Eq. 2 gives $f_1(t) = e^{-\alpha t}$ and $f_2(t) = 1 - f_1(t)$ until these values reach 1/2, at which point they remain at that value.

When the third resource is added, each agent finds the new resource has a lower cost and selects it, i.e., *p3* = 1. Thus f_1 and f_2 decay exponentially toward zero increasing the overall cost, as illustrated in Fig. 2.

In our previous notation, this case corresponds to

$$M = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & 0 \\ -1 & -1 & 0 \end{pmatrix}, \mathbf{v} = \begin{pmatrix} 3 \\ 3 \\ 3 \end{pmatrix} \tag{13}$$

Eq. 5 gives the global minimum at $f_1 = f_2 = 1/2$, $f_3 = 0$ with a cost of $C_{global} = 5/2$, while Eq. 7 gives the local equilibrium at $f_1 = f_2 = 0$, $f_3 = 1$ with cost $C_{global} = 3$.

More generally, when a social dilemma exists, the local equilibrium has a higher cost than the global minimum. Hence some additional constraints on the resource use, e.g., eliminating any use of the third resource in the example of Eq. 13, can in fact change the local equilibrium to be closer to, or even reach, the global minimum. Thus social dilemmas will always be associated with a seemingly paradoxical

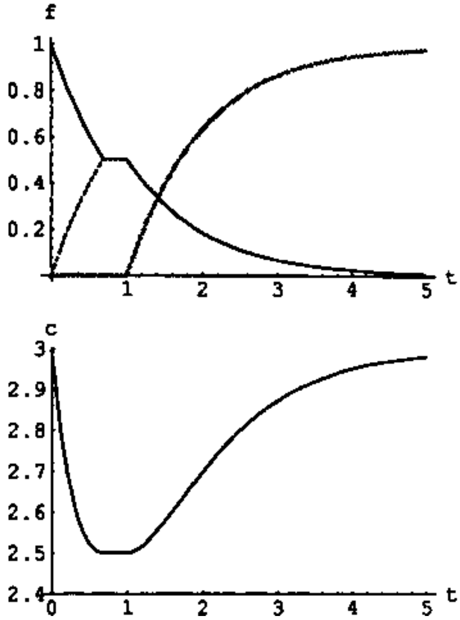

Fig. 2. Behavior as a function of time for a computational ecosystem facing resource costs that include Braess' Paradox. We use $f_1(0) = 1$ and $a = 1$. Initially there are two resources, but at t = 1 the third resource is added The top figure shows the fraction of agents using resources 1, 2 and 3 (solid, dashed and gray, respectively). On the bottom is the global average cost C_{global} for the system. Note that the addition of the third resource causes the cost to rise from its minimum value of 2.5 up to 3.

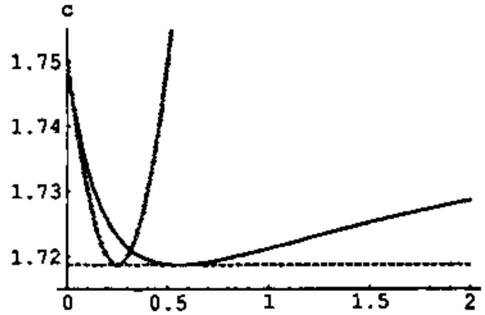

Fig. 3. Removing the social dilemma for the example of Fig. 1. This shows C_{global} at the local equilibrium point selected by the agents as a function of the adjustment added to the system. The black curve is for added uncertainty σ and the gray curve is for a cost x added to the evaluation of resource 2. The horizontal axis for the two curves gives the value of σ and x respectively. With no adjustment, the cost is 1.75, as shown in Fig. 1. Both methods are able to reach the minimum cost (at $\sigma = 0.555$ and $x = 0.25$ respectively) where they meet the dashed line showing the global minimum cost.

situation in which additional restrictions can result in lower costs. This contrasts with the usual intuition that eliminating some available choices in a minimization problem gives a cost at least as large as that for the unrestricted problem. Braess' paradox is particularly surprising in the form given here where an additional constraint of simply eliminating one of the resources is sufficient to lower the cost.

6. Addressing Social Dilemmas

Having seen that social dilemmas can arise for a variety of resource costs, a natural question is how their effect can be alleviated while still retaining the autonomous local decision-making of the computational ecosystem.

A dynamical approach relies on using uncertainty in the agents' evaluation of the costs, which tends to move the system toward equal use of the available resources [Kephart et al., 1989]. When the dilemma gives an equilibrium with more variation in resource use than is optimal the addition of some uncertainty (either deliberately added to the agents or from external causes) can help to improve the performance. This benefit of uncertainty is in addition to its previously recognized use to improve the stability of computational ecosystems. Specifically, uncertainty is modeled by adding a normally distributed random variable with zero mean and standard deviation σ to the costs as evaluated by each agent. For the case of two resources, this changes the probability that an agent will choose resource 1, used in Eq. 2, to be $p_1 = \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{c_2 - c_1}{2\sigma}\right)\right)$ where erf(x) denotes the error function. When is small agents usually choose the resource with the lower cost, but for large σ each is chosen with nearly equal probability. An example of the improvement this can give is shown in Fig. 3.

A second approach to public goods dilemmas, commonly used in human societies, is to introduce mechanisms that enforce contribution, such as taxation. This can be used to artificially change the relative costs perceived by the agents when making choices. An example, shown in Fig. 3, is to add an extra cost x to the overused resource.

While they can be effective, these methods require finding the correct value for the adjustment which can be hard to determine, especially in a complex, changing situation. Hence one would like to rely on the agents themselves to make the necessary adjustments. One such method is suggested by a different approach to the dilemma in human societies: create new markets that, at least partially, restrict benefits to those that contribute. This amounts to privatizing the public good using computational pricing and accounting mechanisms [Miller and Drexler, 1988] within computational economies [Waldspurger et al., 1992]. The resulting prices also provide information on competing uses for resources in a manner similar to that in market economies [Hayek, 1978]. For example, in Fig. 1 agents could be required to pay for the use of resource 2 thus encouraging more of them to use resource 1 instead.

Finally, cooperation can be maintained if the benefits are concentrated in small groups with long-range plans [Glance and Huberman, 1994] or if interactions are repeated [Axelrod and Hamilton, 1981; Bendor and Mookherjee, 1987]. There are also a variety of mechanisms to evolve cooperation in such cases [Simon, 1990]. These observations can suggest additional mechanisms for computational ecosystems as well.

7. Conclusion

We have seen that social dilemmas readily arise in computational ecosystems even when all the agents are designed with a single overall goal. This is a new mechanism that lowers performance, in addition to the previously recognized oscillations and chaotic behavior caused by dynamical instabilities [Kephart et al., 1989]. Since the existence of a dilemma will not always be readily apparent from the cost functions, especially in cases that include nonlinearities, it may be easy to confuse the lowered performance due to a social dilemma with that due to uncertainty or delays in the available information. Such confusion could lead to inappropriate attempts to address the problem.

In dynamic environments with changing resource availability (e.g., a new machine added to a network) or cost functions (e.g., due to a better implementation of a database search) the system could have public goods problems at some times but not others. This is another reason to investigate analogies with human social institutions such as markets that allow local decisions by many agents to respond to these problems in a timely way.

Finally, social dilemmas can also occur in (coevolutionary genetic algorithms and cooperative problem solving where the usefulness of particular methods depends on choices of other agents [Glance and Hogg, 1995]. The results presented here, showing that many cost functions can give rise to dilemmas, suggest they may be common in these situations as well despite their very different dynamical behaviors. Thus methods for recognizing and alleviating social dilemmas are likely to prove useful in a variety of multiagent computational contexts.

Acknowledgments

I have benefited from discussions with N. Glance.

References

Axelrod, R. and Hamilton, W. (1981). The evolution of cooperation. Science, 211:1390-1396.

Bendor, J. and Mookherjee, D. (1987). Institutional structure and the logic of ongoing collective action. American Political Science Review, 81(1):129-154.

Clearwater, S. H., Huberman, B. A., and Hogg, T. (1991). Cooperative solution of constraint satisfaction problems. Science, 254:1181-1183.

Cohen, J. E. and Horowitz, P. (1991). Paradoxical behavior of mechanical and electrical networks. Nature, 352:699-701.

Garey, M. R. and Johnson, D. S. (1979). A Guide to the Theory of NP-Completeness. W. H. Freeman, San Francisco.

Gasser, L. and Huhns, M. N., editors (1989). Distributed Artificial Intelligence, volume 2. Morgan Kaufmann, Menlo Park, CA.

Glance, N. S. and Hogg, T. (1995). Dilemmas in computational societies. In Proc. of Intl. Conf. on Multiagent Systems (ICMAS95). to appear.

Glance, N. S. and Huberman, B. A. (1994). Dynamics of social dilemmas. Scientific American, 270(3):76-81.

Hardin, G. (1968). The tragedy of the commons. Science, 162:1243.

Hayek, F. A. (1978). Competition as a discovery procedure. In New Studies in Philosophy, Politics, Economics and the History of Ideas, pages 179-190. University of Chicago Press, Chicago.

Hogg, T. (1994). Dynamics of multiagent . systems. A World Wide Web page with URL ftp://parcftp.xerox.com/pub/dynamicsAnultiagent.html.

Huberman, B. A. and Hogg, T. (1988). The behavior of computational ecologies. In Huberman, B. A., editor, The Ecology of Computation, pages 77-115. North-Holland, Amsterdam.

Irvine, A. D. (1993). How Braess' paradox solves Newcomb's problem. International Studies in the Philosophy of Science, 7(2): 141-160.

Kephart, J. O., Hogg, T., and Huberman, B. A. (1989). Dynamics of computational ecosystems. Physical Review A, 40:404-421.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. Science, 220:671-680.

Miller, M. S. and Drexler, K. E. (1988). Markets and computation: Agoric open systems. In Huberman, B. A., editor, The Ecology of Computation, pages 133-176. North-Holland, Amsterdam.

Rosenschein, J. S. and Zlotkin, G. (1994). Designing conventions for automated negotiation. AI Magazine, 15(3):29-46.

Rumelhart, D. E., McClelland, J. L., and the PDP Research Group, editors (1986). Parallel Distributed Processing: Explorations in the Microstructure of Cognition. MIT Press, Cambridge, Mass.

Simon, H. A. (1990). A mechanism for social selection and successful altruism. Science, 250(4988): 1665-1668.

Waldspurger, C. A., Hogg, T., Huberman, B. A., Kephart, J. O., and Stometta, W. S. (1992). Spawn: A distributed computational economy. IEEE Trans, on Software Engineering, 18(2):103-117.

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

White, J. E. (1994). Telescript technology: The foundation for the electronic marketplace. Technical report, General Magic, Inc.