Collective Intelligence and Braess' Paradox

Kagan Tumer and David Wolpert
NASA Ames Research Center
Moffett Field, CA 94035
{kagan,dhw}@ptolemy.arc.nasa.gov

Abstract

We consider the use of multi-agent systems to control network routing. Conventional approaches to this task are based on Ideal Shortest Path routing Algorithm (ISPA), under which at each moment each agent in the network sends all of its traffic down the path that will incur the lowest cost to that traffic. We demonstrate in computer experiments that due to the side-effects of one agent's actions on another agent's traffic, use of ISPA's can result in large global cost. In particular, in a simulation of Braess' paradox we see that adding new capacity to a network with ISPA agents can *decrease* overall throughput. The theory of COllective INtelligence (COIN) design concerns precisely the issue of avoiding such side-effects. We use that theory to derive an idealized routing algorithm and show that a practical machine-learning-based version of this algorithm, in which costs are only imprecisely estimated substantially outperforms the ISPA, despite having access to less information than does the ISPA. In particular, this practical COIN algorithm avoids Braess' paradox.

INTRODUCTION

There is a long history of AI research on the design of distributed computational systems, stretching at least from the days of Distributed AI through current work on Multi-Agent Systems (Huhns 1987; Sandholm & Lesser 1995). One particularly important version of such design problems, exhibiting many of the characteristics of the more general problem, involves a set of agents connected across a network that route some form of traffic (here enumerated in "packets") among themselves, and must do so without any centralized control and/or communication. The goal of the system designer is to have the agents act in a way that optimizes some performance measure associated with that traffic, like overall throughput (Bertsekas & Gallager 1992).

Currently, many real-world solutions to this problem use Shortest Path Algorithms (SPA), in which each agent estimates the "shortest path" (i.e., path minimizing total cost accrued by the traffic it is routing) to each of its destinations, and at each moment sends all of its traffic with a particular destination down the associated (estimated) shortest path. Unfortunately, even in the limit of infinitesimally little traffic, performance with SPA's can be badly suboptimal, since each agent's routing decisions ignore side-effects on the traffic of other agents (Korilis, Lazar, & Orda 1997; Wolpert, Tumer, & Frank 1999). Indeed, in the famous case of Braess' paradox (Bass 1992), not only does this scheme result in suboptimal global cost, it causes *every* agent's traffic individually to have higher cost than at optimum. This even holds when each agent's estimated costs are (unrealistically) taken as perfectly accurate, so that those agents are all using Ideal SPA's (ISPA's). This is an instance of the famous Tragedy Of the Commons (TOC) (Hardin 1968).

As an alternative to ISPA's we present a solution to the Braess' paradox bases on the concept of COllective INtelligence (COIN). A COIN is a multi-agent system where there is little to no centralized communication or control among the agents and where there is a well-specified world utility function that rates the possible dynamic histories of the collection (Wolpert, Tumer, & Frank 1999; Wolpert & Tumer 2000b; 2000a; Wolpert, Wheeler, & Tumer 2000). In particular, we are concerned with agents that each use reinforcement learning (Kaelbing, Littman, & Moore 1996; Sutton & Barto 1998; Sutton 1988; Watkins & Dayan 1992) to try to achieve their individual goal. We consider the central COIN design problem: *How, without any detailed modeling of the overall system, can one set utility functions for the individual agents in a COIN to have the overall dynamics reliably and robustly achieve large values of the provided world utility?* In other words, how can we leverage an assumption that our learners are individually fairly good at what they do so as to induce good collective behavior? For reasons given above, we know that in routing the answer to this question is not provided by SPA's goals — some new set of goals is needed.

In this article, we illustrate the Braess' paradox in the network domain, and present a COIN based algorithm for network routing. We present simulations demonstrating that in networks running ISPAs, the per packet costs can be as much as 23 % higher than in networks running COIN algorithms. In particular, even though it only has access to imprecise estimates of costs (a handicap not affecting the ISPA), the COIN algorithm almost always avoids Braess' paradox, in stark contrast to the ISPA. In that the cost incurred with ISPA's is presumably a lower bound on that of a real-world SPA not privy to instantaneous communication, the implication is that COINs can outperform such real-world SPA's. A much more detailed investigation of the issues addressed here can be found in (Wolpert & Tumer 2000a).

Braess' Paradox

Braess' paradox (Bass 1992; Cohen & Kelly 1990; Cohen & Jeffries 1997; Korilis, Lazar, & Orda 1997) dramatically underscores the inefficiency of the ISPA. This "paradox" is perhaps best illustrated through a highway traffic example given in (Bass 1992): There are two highways connecting towns S and D. The cost accrued by a traveler along either highway when x travelers in total traverse that highway (in terms of tolls, delays, or the like) is $V_1(x) + V_2(x)$, as illustrated in Net A of Figure 1. So when $x = 1$ (a single traveler), for either path total accrued cost is 61 units. If on the other hand six travelers are split equally among the two paths, they will each incur a cost of 83 units to get to their destinations. Now suppose a new highway is built connecting the two paths, as shown in Net B in Figure 1. Note that the cost associated with taking this highway is not particularly high (in fact for any load higher than 1, this highway has a lower cost than any other highway in the system). The benefit of this highway is illustrated by the dramatically reduced cost incurred by the single traveler: by taking the short-cut, one traveler can traverse the network at a cost of 31 units ($2\ V_1 + V_3$). Adding a new road has seemingly reduced the traversal cost dramatically.

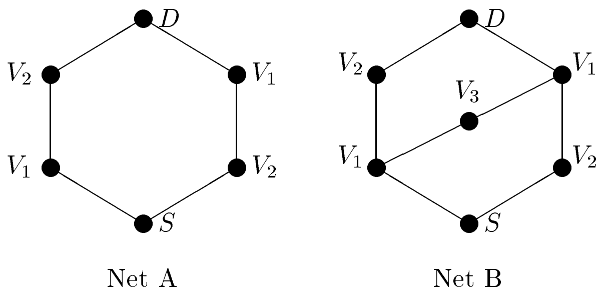

Figure 1: Hex network with $V_1 = 10x$; $V_2 = 50 + x$; $V_3 = 10 + x$

However consider what happens when six travelers are on the highways in net B. If an ISPA is used to make each routing decision, then at equilibrium each of the three possible paths contains two travelers.[1] Due to overlaps in the paths however, this results in each traveler accruing a cost of 92 units, which is higher than than what they accrued *before* the new highway was built. The net effect of adding a new road is to increase the cost incurred by *every* traveler.

[1]We have in mind here the Nash equilibrium, where no traveler can gain by changing strategies (Fudenberg & Tirole 1991).

The COIN Formalism

One common solution to side-effect problems is to have certain components of the network (e.g., a "network manager" (Korilis, Lazar, & Orda 1995)) dictate actions to other routers. This solution can incur major brittleness and scaling problems however. Another kind of approach, which avoids the problems of a centralized manager, is to provide the routers with extra incentives that can induce them to take actions that are undesirable to them from a strict SPA sense. Such incentive can be in the form of "taxes" or "tolls" added to the costs associated with traversing particular links to discourage the use of those links. Such schemes in which tolls are superimposed on the routers' goals are a special case of the more general COIN-based approach of replacing the goal of each router with a new goal. In the COIN approach the new goals are specifically tailored so that if they are collectively met the system maximizes throughput. *A priori*, a router's goal need have no particular relation with the cost accrued by that router's packets. Intuitively, in a COIN approach, we provide each router with a goal that is "aligned" with the global objective, with no separate concern for that goal's relation to the cost accrued by the traffic routed by that router. To see how this can be done, in the remainder of this section we summarize salient aspects of the theory of COIN's.

In this paper we consider systems that consist of a set of agents, connected in a network, evolving across a set of discrete time steps, $t \in \{0, 1, ...\}$. Without loss of generality, all relevant characteristics of an agent η at time t — including its internal parameters at that time as well as its externally visible actions — are encapsulated by a Euclidean vector $\underline{\zeta}_{\eta,t}$ with components $\underline{\zeta}_{\eta,t;i}$. We call this the "state" of agent η at time t, with $\underline{\zeta}_{,t}$ the state of all agents at time t, while $\underline{\zeta}$ is the state of all agents across all time.

In this paper, we restrict attention to utilities of the form $\sum_{t \geq \tau} R_t(\underline{\zeta}_{,t})$ for **reward functions** R_t (simply $\sum_t R_t(\underline{\zeta}_{,t})$ for non-time-varying utilities). **World utility**, $G(\underline{\zeta})$, is an arbitrary function of the state of all agents across all time. (Note that that state is a Euclidean vector.) When η is an agent that uses a machine learning algorithm to "try to increase" its **private utility**, we write that private utility as $g_\eta(\underline{\zeta})$, or more generally, to allow that utility to vary in time, $g_{\eta,\tau}(\underline{\zeta})$.

Here we focus on the case where our goal, as COIN designers, is to maximize world utility through the proper selection of private utility functions. Intuitively, the idea is to choose private utilities that are aligned with world utility, and that also have the property that it is relatively easy for us to configure each agent so that the associated private utility achieves a large value.

We need a formal definition of the concept of having private utilities be "aligned" with G. Constructing such a formalization is a subtle exercise. For example, consider systems where the world utility is the sum of the private utilities of the individual nodes. This might seem a reasonable candidate for an example of "aligned" utilities. However such systems are examples of the more general class of systems that are "weakly trivial". It is well-known that in weakly trivial systems each individual agent greedily trying to maximize its own utility can lead to the tragedy of the commons (Hardin 1968) and actually *minimize* G. In particular, this can be the case when private utilities are independent of time and $G = \sum_\eta g_\eta$. Evidently, at a minimum, having $G = \sum_\eta g_\eta$ is not sufficient to ensure that we have "aligned" utilities; some alternative formalization of the concept is needed.

A more careful formalization of the notion of aligned utilities is the concept of "factored" systems. A system is **factored** at time τ when the following holds for each agent η individually: A change at time τ to the state of η alone, when propagated across time, will result in an increased value of $g_{\eta,\tau}(\underline{\zeta})$ if and only if it results in an increase for $G(\underline{\zeta})$ (Wolpert & Tumer 2000b).

For a factored system, the side-effects of any change to η's $t = \tau$ state that increases its private utility cannot decrease world utility. There are no restrictions though on the effects of that change on the private utilities of other agents and/or times. In particular, we don't preclude an agent's algorithm at two different times from "working at cross-purposes" to each other, so long as at both moments the agent is working to improve G. In game-theoretic terms, in factored systems optimal global behavior corresponds to the agents' always being in a private utility Nash equilibrium (Fudenberg & Tirole 1991). In this sense, there can be no TOC for a factored system. As a trivial example, a system is factored for $g_{\eta,\tau} = G \; \forall \eta$.

Define the **effect set** of the agent-time pair (η, τ) at $\underline{\zeta}$, $C^{eff}_{(\eta,\tau)}(\underline{\zeta})$, as the set of all components $\underline{\zeta}_{\eta',t}$ which under the forward dynamics of the system have non-zero partial derivative with respect to the state of agent η at $t = \tau$. Intuitively, (η,τ)'s effect set is the set of all components $\underline{\zeta}_{\eta',t\geq\tau}$ which would be affected by a change in the state of agent η at time τ. (They may or may not be affected by changes in the $t = \tau$ states of the other agents.)

Next, for any set σ of components (η', t), define $\mathrm{CL}_\sigma(\underline{\zeta})$ as the "virtual" vector formed by clamping the components of the vector $\underline{\zeta}$ delineated in σ to an arbitrary fixed value. (In this paper, we take that fixed value to be 0 for all components listed in σ.) The value of the effect set **wonderful life utility** (WLU for short) for σ is defined as:

$$WLU_\sigma(\underline{\zeta}) \equiv G(\underline{\zeta}) - G(\mathrm{CL}_\sigma(\underline{\zeta})). \tag{1}$$

In particular, we are interested in the WLU for the effect set of agent-time pair (η, τ). This WLU is the difference between the actual world utility and the virtual world utility where all agent-time pairs that are affected by (η, τ) have been clamped to a zero state while the rest of $\underline{\zeta}$ is left unchanged.

Since we are clamping to $\vec{0}$, we can loosely view (η, τ)'s effect set WLU as analogous to the change in world utility that would have arisen if (η, τ) "had never existed". (Hence the name of this utility - cf. the Frank Capra movie.) Note however, that CL is a purely "fictional", counter-factual operator, in that it produces a new $\underline{\zeta}$ without taking into account the system's dynamics. The sequence of states the agent-time pairs in σ are clamped to in constructing the WLU need not be consistent with the dynamical laws of the system. This dynamics-independence is a crucial strength of the WLU. It means that to evaluate the WLU we do *not* try to infer how the system would have evolved if agent η's state were set to $\vec{0}$ at time τ and the system evolved from there. So long as we know $\underline{\zeta}$ extending over all time, σ, and the function G, we know the value of WLU.

If our system is factored with respect to private utilities $\{g_{\eta,\tau}\}$, we want each agent to be in a state at time τ that induces as high a value of the associated private utility as possible (given the initial states of the other agents). Regardless of the system dynamics, having $g_{\eta,\tau} = G \; \forall\eta$ means the system is factored at time τ. It is also true that regardless of the dynamics, $g_{\eta,\tau} = WLU_{C^{eff}_{(\eta,\tau)}} \; \forall\eta$ is a factored system at time τ (proof in (Wolpert & Tumer 2000b)). However, note that since each agent is operating in a large system, it may experience difficulty discerning the effects of its actions on G when G sensitively depends on all the myriad components of the system. Therefore each η may have difficulty learning from past experience what to do to achieve high $g_{\eta,\tau}$ when $g_{\eta,\tau} = G$.[2]

This problem can be mitigated by using effect set WLU as the private utility, since the subtraction of the clamped term removes much of the "noise" of the activity of other agents, leaving only the underlying "signal" of how the agent in question affects the utility. (This reasoning is formalized as the concept of "learnability" in (Wolpert & Tumer 2000b).) Accordingly, one would expect that setting private utilities to

[2]In particular, in routing in large networks, having private rewards given by the world reward functions means that to provide each router with its reward at each time step we need to provide it the full throughput of the entire network at that step. This is usually infeasible in practice. Even if it weren't though, using these private utilities would mean that the routers face a very difficult task in trying to discern the effect of their actions on their rewards, and therefore would likely be unable to learn their best routing strategies.

WLU's ought to result in better performance than having $g_{\eta,\tau} = G \; \forall \eta, \tau$.

Simulation Overview

In this section we describe the model used in our simulations. We then present the ISPA in terms of that model, and apply the concepts of COIN theory to that model to derive private utilities for each agent. Because these utilities are "factored" we expect that agents acting to improve their own utilities will also improve the global utility (overall throughput of the network). We end by describing a Memory Based (MB) machine learning algorithm that each agent uses to estimate the value that its private utility would have under the different candidate routing decisions. In the MB COIN algorithm, each agent uses this algorithm to make routing decisions aimed at maximizing its estimated utility.

Simulation Model

As in much of network analysis, in the model used in this paper, at any time step all traffic at a router is a set of pairs of integer-valued traffic amounts and associated ultimate destination tags (Bertsekas & Gallager 1992). At each such time step t, each router r sums the integer-valued components of its current traffic at that time step to get its **instantaneous load**. We write that load as $z_r(t) \equiv \sum_d x_{r,d}(t)$, where the index d runs over ultimate destinations, and $x_{r,d}(t)$ is the total traffic at time t going from r towards d. After its instantaneous load at time t is evaluated, the router sends all its traffic to the next downstream routers, according to its routing algorithm. After all such routed traffic goes to those next downstream routers, the cycle repeats itself, until all traffic reaches its destination. In our simulations, for simplicity, traffic was only introduced into the system (at the **source routers**) at the beginning of successive disjoint **waves** of L consecutive time steps.

In a real network, the cost of traversing a router depends on "after-effects" of recent instantaneous loads, as well as the current instantaneous load. To simulate this effect, we use time-averaged values of the load at a router rather than instantaneous load to determine the cost a packet incurs in traversing that router. More formally, we define the router's **windowed load**, $Z_r(t)$, as the running average of that router's load value over a window of the previous W timesteps: $Z_r(t) \equiv \frac{1}{W} \sum_{t'=t-W+1}^{t} z_r(t') = \sum_{d'} X_{r,d'}(t)$, where the value of $X_{r,d}(t)$ is set by the dynamical law $X_{r,d}(t) = \frac{1}{W} \sum_{t'=t-W+1}^{t} x_{r,d}(t')$). ($W$ is always set to an integer multiple of L.) The windowed load is the argument to a **load-to-cost** function, $V(\cdot)$, which provides the **cost** accrued at time t by each packet traversing the router at this timestep. That is, at time t, the cost for each packet to traverse router r is given by $V(Z_r(t))$. Different routers have different $V(\cdot)$, to reflect the fact that real networks have differences in router software and hardware (response time, queue length, processing speed etc). For simplicity, W is the same for all routers however. With these definitions, world utility is

$$G(\underline{\zeta}) = \sum_{t,r} z_r(t) \; V_r(Z_r(t)) \qquad (2)$$

Our equation for G explicitly demonstrates that, as claimed above, in our representation we can express $G(\underline{\zeta})$ as a sum of rewards, $\sum_t R_t(\underline{\zeta}_{,t})$, where $R(\underline{\zeta}_{,t})$ can be written as function of a pair of (r,d)-indexed vectors: $R_t(x_{r,d}(t), X_{r,d}(t)) = \sum_{r,d} x_{r,d}(t) V_r(\sum_{d'} X_{r,d'}(t))$.

Routing Algorithms

At time step t, ISPA has access to all the windowed loads at time step $t-1$ (i.e., it has access to $Z_r(t-1) \; \forall r$), and assumes that those values will remain the same at all times $\geq t$. (Note that for large window sizes and times close to t, this assumption is arbitrarily accurate.) Using this assumption, in ISPA, each router sends packets along the path that it calculates will minimize the costs accumulated by its packets.

We now apply the COIN formalism to the model described above to derive the idealized version of our COIN routing algorithm. First let us identify the agents η as individual pairs of routers and ultimate destinations. So $\underline{\zeta}_{\eta,t}$ is the vector of traffic sent along all links exiting η's router, tagged for η's ultimate destination, at time t. Next, in order to compute WLUs we must estimate the associated effect sets.

In the results presented here, the effect set of an agent is estimated as all agents that share the same destination as that agent.[3] Based on this effect set, the WLU for an agent η is given by the difference between the total cost accrued by all agents in the network and the cost accrued by agents when all agents sharing the same destination as η are "erased." More precisely, using Eq. 2, one can show that each agent η that shares a destination d, will have the following effect set WLU:

$$\begin{aligned} g_d(\underline{\zeta}) &= G(\underline{\zeta}) - G(\mathrm{CL}_{C_\eta^{eff}}(\underline{\zeta})) \\ &= \sum_t \sum_r \left[z_r(t) \; V_r(Z_r(t)) \; - \right. \\ & \qquad \left. \sum_{d' \neq d} x_{r,d'}(t) \;\; V_r\Big(\sum_{d'' \neq d} X_{r,d''}(t)\Big) \right] \qquad (3) \end{aligned}$$

Notice that the summand in Eq. 3 is computed at each router separately from information available to that router. Subsequently those summands can be propagated across the network and the associated g_d's "rolled up" in much the same way as routing tables updates are propagated in current routing algorithms.

Unlike the ISPA, the MB COIN has only limited knowledge, and therefore must *predict* the WLU value that would result from each potential routing decision. More precisely, for each router-ultimate-destination pair, the associated agent estimates the map from windowed loads on all outgoing links (the inputs) to WLU-based reward (the outputs). This is done with a single-nearest-neighbor algorithm. Next, each router could

[3] Exact factoredness obtains so long as our estimated effect set contains the true effect set; set equality is not necessary.

send the packets along the path that results in outbound traffic with the best (estimated) reward. However to be conservative, in these experiments we instead had the router randomly select between that path and the path selected by the ISPA (described below).

SIMULATION RESULTS

Based on the model and routing algorithms discussed above, we have performed simulations to compare the performance of ISPA and MB COIN. In all cases traffic was inserted into the network in a regular, non-stochastic manner at the sources. The results we report are averaged over 20 runs. We do not report error bars as they are all lower than 0.05. In both networks we present[4], ISPA suffers from the Braess' paradox, whereas the MB COIN almost never falls prey to the paradox for those networks. For no networks we have investigated is the MB COIN significantly susceptible to Braess' paradox.

Hex Network

In Table 1 we give full results for the network in Fig. 1. In Table 2 we report results for the same network but with load-to-cost functions which incorporate non-linearities that better represent real router characteristics. (Instances of Braess' paradox are shown in **bold**.)

For ISPA, although the per packet cost for loads of 1 and 2 drop drastically when the new link is added, the per packet cost increases for higher loads. The MB COIN on the other hand uses the new link efficiently. Notice that the MB COIN's performance is slightly worse than that of the ISPA in the absence of the additional link. This is caused by the MB COIN having to use an (extremely unsophisticated) learner to estimate the WLU values for potential actions whereas the ISPA has direct access to all the information it needs.

For this particular network, the equilibrium solution for the MB-COIN consists of ignoring the newly added middle link. This solution is "unstable" for the ISPA, since any packet routed along the middle path will provide a smaller cost to the router from which it was routed than would otherwise be the case, so that the system settles on the the suboptimal Nash Equilibrium solution discussed above. However, by changing the utilities of the agents (from a shortest path to the WLU), the COIN approach moves the Nash equilibrium to a more desirable location in the solution space.

Butterfly Network

The next network we investigate is shown in Figure 2. We now have three sources that have to route their packets to two destinations (packets originating at S_1 go to D_1, and packets originating at S_2 or S_3 go to D_2). Initially the two halves of the network have minimal contact, but with the addition of the extra link

[4]See (Wolpert & Tumer 2000a) for additional experiments.

Table 1: Average Per Packet Cost for HEX network for $V_1 = 50 + x$; $V_2 = 10x$; $V_3 = 10 + x$.

Load	Net	ISPA	MB COIN
1	A	55.50	55.56
	B	31.00	31.00
2	A	61.00	61.10
	B	52.00	51.69
3	A	66.50	66.65
	B	**73.00**	64.45
4	A	72.00	72.25
	B	**87.37**	73.41

Table 2: Average Per Packet Cost for HEX network for $V_1 = 50 + log(1 + x)$; $V_2 = 10x$; $V_3 = log(1 + x)$.

Load	Net	ISPA	MB COIN
1	A	55.41	55.44
	B	20.69	20.69
2	A	60.69	60.80
	B	41.10	41.10
3	A	65.92	66.10
	B	61.39	59.19
4	A	71.10	71.41
	B	**81.61**	69.88

two sources from the two halves of the network share a common router on their potential shortest path.

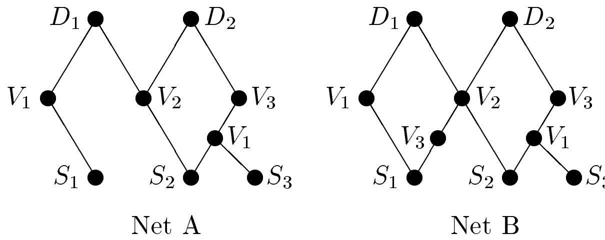

Figure 2: Butterfly Network

Table 3 presents results for uniform traffic through all three sources, and then results for asymmetric traffic. For the first case, the Braess' paradox is apparent in the ISPA: adding the new link is beneficial for the network at low load levels where the average per packet cost is reduced by nearly 20%, but deleterious at higher levels. The MB COIN, on the other hand, provides the benefits of the added link for the low traffic levels, without suffering from deleterious effects at higher load levels.

For the asymmetric traffic patterns, the added link causes a drop in performance for the ISPA, especially for low overall traffic levels. This is not true for the MB COIN. Notice also that in the high, asymmetric traffic regime, the ISPA performs significantly worse than the MB COIN even without the added link, showing that a bottleneck occurs on the right side of network alone.

Table 3: Average Per Packet Cost for BUTTERFLY network for $V_1 = 50 + log(1+x)$; $V_2 = 10x$; $V_3 = log(1+x)$.

Loads (S_1, S_2, S_3)	Net	ISPA	MB COIN
1,1,1	A	112.1	112.7
	B	92.1	92.3
2,2,2	A	123.3	124.0
	B	**133.3**	122.5
4,4,4	A	144.8	142.6
	B	**156.5**	142.3
3,2,1	A	81.8	82.5
	B	99.5	81.0
6,4,2	A	96.0	94.1
	B	**105.3**	94.0
9,6,3	A	105.5	98.2
	B	**106.7**	98.8

CONCLUSION

Collective Intelligence design is a framework for controlling decentralized multi-agents systems so as to achieve a global goal. In designing a COIN, the central issue is determining the private goals to be assigned to the individual agents. One wants to choose those goals so that the greedy pursuit of them by the associated agents leads to a globally desirable solution. We have summarized some of the theory of COIN design and derived a routing algorithm based on application of that theory to our simulation scenario. In our simulations, the COIN algorithm induced costs up to 23 % lower than the idealized version of conventional algorithms, the ISPA. This was despite the ISPA's having access to more information than the MB COIN. Furthermore the COIN-based algorithm avoided the Braess' paradoxes that seriously diminished the performance of the ISPA.

In the work presented here, the COIN-based algorithm had to overcome severe limitations. The estimation of the effect sets, used for determining the private goals of the agents was exceedingly coarse. In addition, the learning algorithms used by the agents to pursue those goals were particularly simple-minded. That a COIN-based router with such serious limitations consistently outperformed an ideal shortest path algorithm demonstrates the strength of the proposed method.

References

Bass, T. 1992. Road to ruin. *Discover* 56–61.

Bertsekas, D., and Gallager, R. 1992. *Data Networks.* Englewood Cliffs, NJ: Prentice Hall.

Boyan, J., and Littman, M. 1994. Packet routing in dynamically changing networks: A reinforcement learning approach. In *Advances in Neural Information Processing Systems - 6*, 671–678. Morgan Kaufmann.

Cohen, J. E., and Jeffries, C. 1997. Congestion resulting from increased capacity in single-server queueing networks. *IEEE/ACM Tran. on Net.* 5(2):305–310.

Cohen, J. E., and Kelly, F. P. 1990. A paradox of congestion in a queuing network. *Journal of Applied Probability* 27:730–734.

Fudenberg, D., and Tirole, J. 1991. *Game Theory.* Cambridge, MA: MIT Press.

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

Heusse, M.; Snyers, D.; Guerin, S.; and Kuntz, P. 1998. Adaptive agent-driven routing and load balancing in communication networks. *Advances in Complex Systems* 1:237–254.

Huhns, M. E., ed. 1987. *Distributed Artificial Intelligence.* London: Pittman.

Kaelbing, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4:237–285.

Korilis, Y. A.; Lazar, A. A.; and Orda, A. 1995. Architechting noncooperative networks. *IEEE Journal on Selected Areas in Communications* 13(8).

Korilis, Y. A.; Lazar, A. A.; and Orda, A. 1997. Achieving network optima using Stackelberg routing strategies. *IEEE/ACM Transactions on Networking* 5(1):161–173.

Sandholm, T., and Lesser, V. R. 1995. Issues in automated negotiations and electronic commerce: extending the contract net protocol. In *Proc of the 2nd Intl. Conf. on Multi-Agent Systems*, 328–335. AAAI Press.

Subramanian, D.; Druschel, P.; and Chen, J. 1997. Ants and reinforcement learning: A case study in routing in dynamic networks. In *Proc. of the 15th Intl Conf. on Artificial Intelligence*, 832–838.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press.

Sutton, R. S. 1988. Learning to predict by the methods of temporal differences. *Machine Learning* 3:9–44.

Watkins, C., and Dayan, P. 1992. Q-learning. *Machine Learning* 8(3/4):279–292.

Wolpert, D. H., and Tumer, K. 2000a. Avoiding Braess' paradox through collective intelligence. Available as tech. rep. NASA-ARC-IC-99-124 from `http://ic.arc.nasa.gov/ic/projects/coin_pubs.html`.

Wolpert, D. H., and Tumer, K. 2000b. An Introduction to Collective Intelligence. In Bradshaw, J. M., ed., *Handbook of Agent technology.* AAAI Press/MIT Press. Available as tech. rep. NASA-ARC-IC-99-63 from `http://ic.arc.nasa.gov/ic/projects/coin_pubs.html`.

Wolpert, D. H.; Tumer, K.; and Frank, J. 1999. Using collective intelligence to route internet traffic. In *Advances in Neural Information Processing Systems - 11*, 952–958. MIT Press.

Wolpert, D. H.; Wheeler, K.; and Tumer, K. 2000. Collective intelligence for control of distributed dynamical systems. *Europhysics Letters* 49(6).