USING COLLECTIVE INTELLIGENCE TO ROUTE INTERNET TRAFFIC

David H. Wolpert
NASA Ames Research Center
Moffett Field, CA 94035
dhw@ptolemy.arc.nasa.gov

Kagan Tumer
NASA Ames Research Center
Moffett Field, CA 94035
kagan@ptolemy.arc.nasa.gov

Jeremy Frank
NASA Ames Research Center
Moffett Field, CA 94035
frank@ptolemy.arc.nasa.gov

Abstract

A COllective INtelligence (COIN) is a set of interacting reinforcement learning (RL) algorithms designed so that their collective behavior optimizes a global utility function. We summarize the theory of COINs, then present experiments using that theory to design COINs to control internet traffic routing. These experiments indicate that COINs outperform all previously investigated RL-based, shortest path routing algorithms.

1 INTRODUCTION

COllective INtelligences (COINs) are large, sparsely connected recurrent neural networks, whose "neurons" are themselves reinforcement learning (RL) algorithms. The distinguishing feature of a COIN is that its dynamics involves no centralized control, but only the collective effects of the individual neurons each modifying their behavior via their (local) RL algorithms. This restriction holds even though the goal of the COIN concerns the system's global behavior.

One naturally-occurring COIN is a human economy, where the "neurons" consist of individual humans trying to maximize their reward, and the "goal", for example, can be viewed as having the overall system achieve high gross domestic product. The robustness and power of naturally-occurring COINs suggests there would be many uses for Artificial COINs (ACOINs). This paper presents a preliminary investigation of one such use of ACOINs, as controllers of distributed systems, in particular as routers of internet traffic.

The design of an ACOIN starts with a provided global utility function concerning the global behavior of the application at hand. Our task is to initialize and then update the neurons' utility functions so that as the neurons improve their utilities, global utility also improves. (We may also wish to update the local topology of the ACOIN to the same end.) In particular, we need to ensure that the neurons do not "frustrate" each other as they attempt to increase their utilities. All of this must be accomplished without centralized control.

We refer to the RL algorithms at each neuron that aim to optimize that neuron's local utility as *microlearners*. The learning algorithms that update the neuron's utility functions and the the overall topology are *macrolearners*. For robustness and breadth of applicability, we assume essentially no knowledge concerning the dynamics of the full system, i.e., the macrolearning and/or microlearning must "learn" that dynamics, implicitly or otherwise. This rules out any approach that models the full system. Moreover, rather than using domain knowledge to handcraft the local utilities as is done in multi-agent systems, in COINs, the local utility functions are automatically initialized and updated using the global utility and locally observed dynamics.

The problem of designing an ACOIN has never previously been addressed in full — hence the need for introducing the new concepts described below. Nonetheless, this problem is related to previous work in many fields: distributed artificial intelligence, multi-agent systems, computational ecologies, adaptive control, game theory, computational markets, Markov decision theory, and ant-based optimization.

For the particular problem of routing, examples of relevant work are [3, 4, 6, 9]. Most of that previous work uses RL to set the internal parameters of conventional shortest path algorithms (SPAs). However the RL is used, the reward signal received by each router is its personal routing performance, so that the problem of routers working at cross purposes is not addressed.

We used simulations to compare three algorithms for internet routing. The first two are an SPA and an ACOIN. Both had "full knowledge" (FK) of the true reward-maximizing path, with reward being routing time for the SPA's, but set by COIN theory for the ACOINs. The third algorithm was an ACOIN using a memory-based (MB) microlearner [1] whose knowledge was limited to local observations.

The performance of the FK ACOIN was the theoretical optimum. The performance of the FK SPA was 12.5 ± 3 % worse than this. Despite limited knowledge, the MB ACOIN outperformed the FK SPA, reducing the deficit in that algorithm's performance by 36 ± 8 %. Note that the performance of the FK SPA is an upper bound on the performance of any RL-based SPA. Accordingly, the performance of the MB ACOIN is superior to that of any RL-based SPA.

Section 2 presents a cursory overview of the mathematics behind ACOINs, and shows how local utility functions are derived from the global utility function. Section 3 discusses how the network routing problem is mapped into the COIN formalism, and introduces our experiments. Section 4 presents results of those experiments, which establish the power of COINs in the context of routing problems. Finally, Section 5 presents conclusions and summarizes future research directions.

2 MATHEMATICS OF COINS

The mathematical framework for COINs in general is quite extensive [10]. This paper concentrates on four of the concepts from that framework: subworlds, factored systems, constraint-alignment, and the wonderful-life utility function.

i) We consider the state of the system across a set of $T+1$ consecutive timesteps, which we write without loss of generality as $t \in [0, ..., T]$. Let $\underline{\zeta}_{\eta,t}$ be the (Euclidean-vector-valued) state of neuron η at time t. World utility is a function of the state of all neurons across all t, which we write as $G(\underline{\zeta})$.

ii) To simplify the design of the ACOIN, we group neurons into *subworlds*, all sharing the same *subworld utility*. In the ACOINs considered here, subworlds have no overlap, and a neuron's microlearner receives the utility of that neuron's subworld as its reward signal. We indicate the state of neuron σ in subworld ω at time t by $\underline{\zeta}_{\omega,\sigma,t}$, and by $g_\omega(\underline{\zeta})$ we indicate the subworld utility of subworld ω.

iii) One of the primary desiderata in designing an ACOIN is to have it be *subworld-factored*. This means that a change at time 0 to the state of the neurons in subworld ω results in an increased value for $g_\omega(\underline{\zeta})$ iff it results in an increased value for $G(\underline{\zeta})$.

For a subworld-factored system, ω's increasing its own utility does not have "side-effects" on the rest of the system that decrease world utility. For these systems, the separate neurons pursuing their separate goals do not frustrate each other, as far as world utility is concerned. In particular, consider a Nash equilibrium of the system, i.e., a state in which no neuron can increase its utility by changing its action, given the actions of the other neurons [7, 8]. The following is proven in [10]:

Theorem 1: If a system is subworld-factored, then any state at time 0 of the system that constitutes a Nash equilibrium for the subworld utilities is a critical point of world utility. Conversely, a maximum of world utility is such a Nash equilibrium.

In other words, assuming we are not at a critical point, for a subworld-factored system world utility is assuredly maximized when each of the neurons' reinforcement learning algorithms are performing as well as they can, given each others' behavior. So we have the correct behavior if and when the learners reach an equilibrium. There can be no tragedy of the commons [5], etc.

iv) A (perfectly) *constraint-aligned* system is one in which any change to the state of the neurons in subworld ω at time 0 will have no effects on the states of neurons outside of ω at times later than 0.

v) Let $Zero_\omega(\underline{\zeta})$ be defined as the vector $\underline{\zeta}$ modified by setting the values of all neurons in subworld ω across all time to $\overline{0}$. The *wonderful life subworld utility* (WLU) is:

$$g_\omega(\underline{\zeta}) \equiv G(\underline{\zeta}) - G(Zero_\omega(\underline{\zeta})). \tag{1}$$

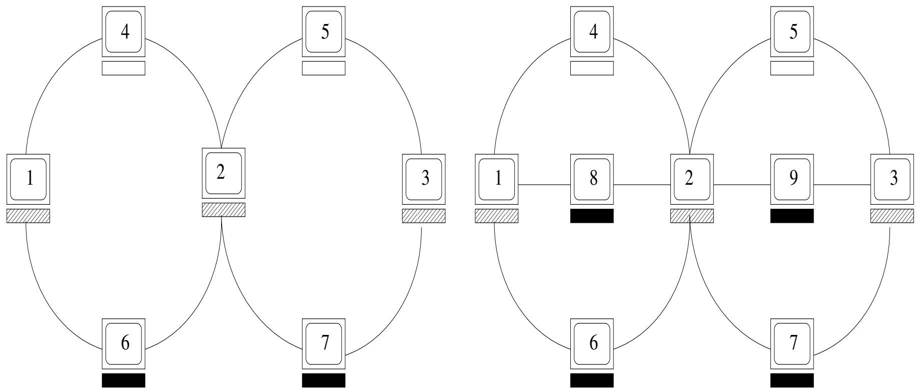
Intuitively, the WLU is analogous to the change in world utility that would have arisen if subworld ω had never existed. (Hence its name - cf. the Frank Capra movie.) Note though that no assumption is made that $Zero_\omega(\underline{\zeta})$ is consistent with the dynamics of the system. In particular, to evaluate the WLU we do *not* have to infer how the system would have evolved if all neurons in ω were set to 0 at time 0. So long as we know $\underline{\zeta}$ (by observation, in the process of macrolearning that occurs at time T), and so long as we know G (set in the problem specification of the ACOIN), we can evaluate the WLU. This is true even if we know nothing of the dynamics of the system. This dynamics-independence is a major strength of the WLU.

In addition to assuring the correct equilibrium behavior, there exist many other theoretical advantages to having a system be subworld-factored [10]. In this paper we use computer experiments to establish the applicability of these advantages to network routing. In particular, the following theorem, proved in [10], serves as the basis for the investigation of this paper:

Theorem 2: A constraint-aligned system with wonderful life subworld utilities is subworld-factored.

3 COINS FOR NETWORK ROUTING

In our experiments we concentrated on the two networks in Figure 1, both slightly larger than those in [6]. To facilitate the analysis, traffic originated only at routers indicated with white boxes and had only the routers indicated by dark boxes as ultimate destinations. Note that in both networks there is a bottleneck at router 2.



(a) Network A **(b)** Network B

Figure 1: Network Architectures.

As is standard in much of traffic network analysis [2], at any time, all *traffic* at a router is a real-valued number together with an ultimate destination tag. At each timestep, each router sums all traffic received in this timestep, to get a total *load*. A running average is kept of each router's load over a window of the previous L timesteps, and that average is run through a *load-to-delay* function, $W(x)$, to get the router's total *delay* at this timestep. To ensure a heterogeneous pool of routers, different routers had different load-to-delay functions, to reflect the fact real networks have differences in router software and hardware (response time, queue length, processing speed etc). In the experiments reported here, the load-to-delay functions were $W(x) = x^3$ for routers 1 and 3, and $W(x) = log(x+1)$ for router 2, for both networks. The load-to-delay functions of the other routers did not affect our simulations.

In terms of the COIN formalism, we identified the neurons η as individual pairs of routers and ultimate destinations. So $\underline{\zeta}_{\eta,t}$ was the vector of traffic sent along all links exiting η's router, tagged for η's ultimate destination at time t. Each subworld consisted of the set all neurons that shared a particular ultimate destination.

An SPA tries to set $\underline{\zeta}_{\eta,t}$ to minimize the sum of the total delays along all possible paths to η's ultimate destination. In contrast, an ACOIN tries to set $\underline{\zeta}_{\eta,t}$ to optimize g_ω for the subworld ω containing ζ. For both algorithms, "full knowledge" means that all of the routers know the values of $\underline{\zeta}$ for the previous time step, and assume that those values will be the same at t. For large enough window sizes L, this assumption will be arbitrarily good in most cases, and therefore will allow the routers to make arbitrarily accurate estimates of how best to route their traffic, according to their respective routing criteria.

In contrast, having limited knowledge, the MB ACOIN could only predict the WLU value resulting from each routing decision. More precisely, for each router-ultimate-destination pair, the associated microlearner tries to form an accurate map from the loads on all outgoing links (inputs) to resultant WLU values (outputs). Then the WLU corresponding to each potential routing decision is predicted by using a single nearest neighbor algorithm. At that point, a natural choice is for each router to send the packets along the path that provided the best (estimated) WLU. However, in order to generate a more complete training set, the router randomly selects between the path that leads to the best WLU and the choice of the FK SPA.

The load at router r at time t is determined by $\underline{\zeta}$. Accordingly, we can encapsulate the load-to-delay functions at the nodes by writing the delay at node r at time t as $W_{r,t}(\underline{\zeta})$. In our experiments world utility was the total-delay, i.e., $G(\underline{\zeta}) = \sum_{r,t} W_{r,t}(\underline{\zeta})$. So using the WLU, $g_{\omega}(\underline{\zeta}) = \sum_{r,t} \Delta_{\omega,r,t}(\underline{\zeta})$, where $\Delta_{\omega,r,t}(\underline{\zeta}) = [W_{r,t}(\underline{\zeta}) - W_{r,t}(Zero_{\omega}(\underline{\zeta}))]$. Now $W_{r,t}(Zero_{\omega}(\underline{\zeta}))$ is the delay that would have existed at router r at time t if all the traffic from r at t that was destined for ω had not existed. So for our G, the WLU is just a sum of differences in delays "caused" by the routing decisions of the neurons in subworld ω. In these experiments, T was the full length of the simulation, and at each time t, the WLU was approximated by ignoring other times, i.e., by $\sum_{r} \Delta_{\omega,r,t}(\underline{\zeta})$.

In the MB ACOIN, evaluating this subworld utility in a decentralized fashion was straight-forward. The essence of the scheme is that all traffic had a header containing a running sum of the Δ's encountered in all the routers it has traversed so far. Then each ultimate destination summed all such headers it received and echoed that sum back to all routers that had routed to it. In this way each neuron is apprised of the subworld utility of its subworld that resulted (in part) from its routing decisions.

4 EXPERIMENTAL RESULTS

The networks discussed above were tested under different traffic patterns. Table 1 shows the destinations for all sources under light, medium and heavy loads.

Table 1: Destinations for Different Traffic Regimes

Network	Source	Light	Medium	Heavy
A	4	6	6,7	6,7
	5	7	7	6,7
B	4	7,8	7,8,9	6,7,8,9
	5	6,9	6,7,9	6,7,8,9

Table 2 reports the average total delay (i.e., average per packet time to traverse the total network) in each of the traffic regimes, for the shortest path algorithm with full knowledge, the ACOIN with full knowledge, and the MB ACOIN. Each table entry is based on 50 runs, with a window size of 50, and traffic for 200 timesteps[1]. All the entries in Table 2 are statistically different, including FK SPA vs. MB ACOIN for Network A under light traffic conditions (i.e., the hypothesis that any two entries are equal is rejected at the $\alpha = 0.01$ level for the t-test.)

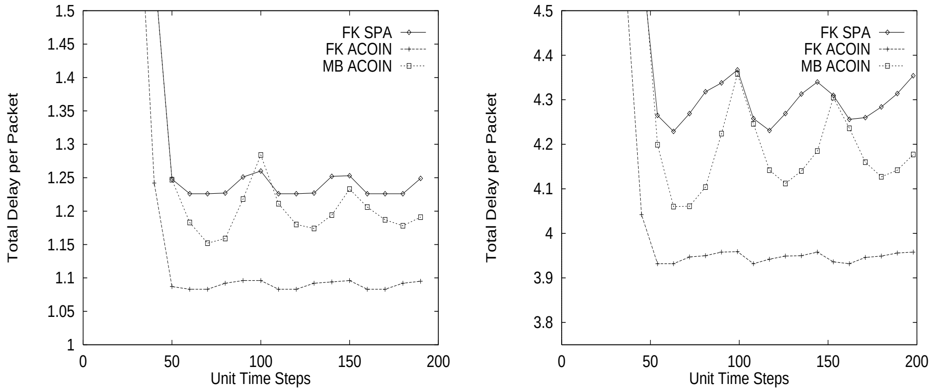
[1]The results are qualitatively identical for window sizes 20 and 100 along with total timesteps of 100 and 500.

Table 2: Average Total Delay

Network	Load	FK SPA	FK ACOIN	MB ACOIN
A	light	0.53	0.45	0.50
	medium	1.26	1.10	1.21
	heavy	2.17	1.93	2.06
B	light	2.13	1.92	2.05
	medium	4.37	3.96	4.19
	heavy	6.94	6.35	6.82

Table 2 provides two important observations: First, the WLU-based ACOIN outperformed the SPA when both have full knowledge, thereby demonstrating the superiority of the new routing strategy. By not greedily pursuing the quickest reward (shortest path), the ACOIN settles into a more desirable state that reduces the average total delay for *all* packets. Second, even when the WLU is estimated through a memory-based learner (using only information available to the local routers), the performance of the ACOIN still surpasses that of the full knowledge shortest path algorithm. This result not only establishes the feasibility of ACOIN-based routers, but also demonstrates that for this task ACOINs will outperform *any* algorithm that attempts to estimate the shortest path. This is because the performance of the FK SPA is necessarily a ceiling on the performance of any RL-based SPA.

The results reported above concern the average behavior of the networks under various regimes and different routing strategies. However, the behavior of how total delays vary with time also sheds light on the differences between the algorithms. As an example, Figure 2 shows how the total delay varies with time for the medium traffic regime (each plot is based on 50 runs). Note that for both networks the FK ACOIN not only provides the shortest delays, but also settles into that solution very rapidly. Both the FK SPA and the MB ACOIN take longer to settle and are more susceptible to perturbations.



(a) Network A

(b) Network B

Figure 2: Total Delay.

5 DISCUSSION

Many distributed computational tasks are naturally addressed as recurrent neural networks of reinforcement learning algorithms (i.e., ACOINs). The difficulty in doing so is ensuring that, despite the absence of centralized communication and control, the reward functions of the separate neurons work in synchrony to achieve the global good, rather than work at cross-purposes.

The mathematical framework synopsized in this paper is a theoretical solution to this difficulty. To assess its real-world applicability, we employed it to design a full-knowledge (i.e., perfect RL) ACOIN as well as a memory-based (imperfect RL) ACOIN, for the task of packet routing on a network. We then compared the performance of those algorithms to that of a full-knowledge shortest-path algorithm. These comparisons conclusively demonstrated that not only did the FK ACOIN beat the FK SPA, but also that the memory-based ACOIN, despite having only limited knowledge, beat the full-knowledge SPA. This latter result is all the more remarkable in that the performance of the FK SPA is an upper bound on the performance of previously investigated RL-based routing schemes, which use the RL to provide accurate knowledge to an SPA.

There are many directions for future work on ACOINs, even restricting attention to their application to packet routing. Currently we are extending our experiments to larger networks, using industrial event-driven network simulators. Concurrently, we are investigating the use of macrolearning for ACOIN-based packet-routing, i.e., the run-time modification of the neurons' utility functions to improve the subworld-factoredness of the ACOIN. We are also investigating some of the other possible mappings of the mathematics of ACOINS onto the problem of packet routing (e.g., different definitions of neurons, of subworlds, etc.).

References

[1] C. G. Atkenson, A. W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, Submitted, 1996.

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

[3] J. Boyan and M. Littman. Packet routing in dynamically changing networks: A reinforcement learning approach. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems - 6*, pages 671–678. Morgan Kaufmann, 1994.

[4] S. P. M. Choi and D. Y. Yeung. Predictive Q-routing: A memory based reinforcement learning approach to adaptive traffic control. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems - 8*, pages 945–951. MIT Press, 1996.

[5] B. L. Crowe. The tragedy of the commons revisited. *Science*, 166:1103–1107, 1969.

[6] P. Marbach, O. Mihatsch, M. Schulte, and J. Tsisiklis. Admission control and routing in integrated service networks. In *Advances in Neural Information Processing Systems - 10*. MIT Press, To Appear, 1998.

[7] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36(48-49), 1950.

[8] M. Osborne and A. Rubenstein. *A Course in Game Theory*. MIT Press, Cambridge, MA, 1994.

[9] D. Subramanian, P. Druschel, and J. Chen. Ants and reinforcement learning: A case study in routing in dynamic networks. In *Proceedings of the 15th International Conference on Artificial Intelligence*, pages 832–838, 1997.

[10] D. Wolpert, J. Frank, K. Tumer, and K. Wheeler. The mathematics of collective intelligence. Technical Report SFI-TR-10-9847, October 1998.