Cooperative Information Sharing to Improve Distributed Learning

Partha S. Dutta, Srinandan Dasmahapatra, Steve R. Gunn, Nicholas R. Jennings, and Luc Moreau
School of Electronics & Computer Science
University of Southampton
Highfield, Southampton SO 17 1BJ, U.K.
{psa01r,sd,srg,nrj,l.Moreau}@ecs.soton.ac.uk

Abstract

Effective coordination in partially observable MAS requires agent actions to be based on reliable estimates of non-local states. One way of generating such estimates is to allow the agents to share state information that is not directly observable. To this end, we propose a novel strategy of delayed distribution of state estimates. Our empirical studies of this mechanism demonstrate that individual reinforcement-learning agents in a simulated network routing problem achieve a significant improvement in the overall success, robustness, and efficiency of routing compared with the standard Q-routing algorithm.

1. Introduction

In partially observable MAS, the agents need reliable estimates of the world states to effectively coordinate their actions. Moreover, in dynamic systems, these estimates need continuous and effective updates and to this end, reinforcement learning [4] (RL) is often used. However, while RL can generate reliable estimates, it relies on the agents having access to information about unobserved states. One way of achieving this is to have the agents cooperatively share such information. Given this, the focus of this paper is on the strategies that can be used to manage this process.

Previous research on the use of communication to improve learning lacks a sufficiently detailed investigation to understand its benefits and drawbacks. For example, a value search algorithm [4] is used by Boyan and Littman [2] where agents learn the estimates of packet routing delays in a communication network using the information received only from their immediate neighbours. In [1] and [8], on the other hand, the agents use policy search algorithms [3] to learn the optimum policy by globally broadcasting the rewards received by all agents at all times. These represent the extremities of the space of possible communication strategies for multi-agent learning (from very restricted communication in [2] to unrestricted communication in [1, 8]) and both have limitations. In a real-world MAS, it is typically not possible to broadcast individual reward information to all agents (due to excessive bandwidth usage and the communication latency involved). Also, updating estimates using the information of only immediate neighbours can potentially suffer from the count-to-infinity problem [5] and the error in the non-local state estimates can become arbitrarily high in dynamic systems (hence causing sub-optimal action selection).[1]

Therefore, in this paper, an alternative information sharing strategy is proposed to address the above limitations. Specifically, we consider a cooperative MAS in which the actions of the agents should be chained together to successfully complete tasks. Thus, the individual agents should take the appropriate actions to accomplish parts of the overall task and, subsequently, hand over the task to another agent who continues with this process until the task is completed. In this case, the agents learn the estimates of the states of other agents to choose the subsequent agent such that the probability of successful task completion is maximised without any global coordination. We propose that the agents should take their individual actions based on their current estimates of the states of other agents (as opposed to asking the immediate neighbours for their current estimates [2], or waiting for the feedback of all other agents [1, 8]). Then, after completing their current task, they should communicate their state information to one another (thus, *delaying* communication until task completion) to allow them to update their prior estimates of these states. Note that the agents *cooperate* by voluntarily communicating state information to others to mutually expedite the learning process and improve their collective routing decisions.

The motivation for this communication mechanism is the

[1] We are currently working towards a formal proof of this intuition.

following. The states of the agents change as they carry out a given task. Therefore, by delaying the transmission of information until the task is completed, our protocol ensures that all those agents who participated in the task completion process are informed about these state changes. Using this information, these agents subsequently reinforce their prior estimates of these states. This should not be confused with conventional supervised learning [7] where the actual outcome of a multi-stage prediction problem is fed back to the individual learners (predictors). Instead, in our work, we use the standard Q-learning approach [6] and concentrate on developing practical and effective means of distributing the non-local state information among the learners.

The principle of delayed communication is evaluated in a simulated telephone network (TN) domain where the agents work together to allocate bandwidth to connect circuits. The agents use Q-learning to estimate the bandwidth usage at other nodes. Further, the agents distribute information about their individual node bandwidth usage after completing the routing of calls from source to destination nodes. Such information distribution uses two heuristics that we have formulated, based on the above motivation, in the context of the TN domain.[2] The performance of these heuristics is benchmarked against the widely cited Q-routing approach of [2]. Empirical results indicate (statistically) significant improvements in performance over the benchmark by using our communication principle.

The following section formulates the communication heuristics. Section 3 analyses the performance of these heuristics based on empirical studies in the TN domain. Finally, section 4 concludes and identifies future directions.

2. Communication Heuristics

We present three different formulations of cooperative information sharing in the TN domain. Section 2.1 emulates the strategy of [2]. Section 2.2 and 2.3 present two heuristics based on the principle of delayed communication proposed in this paper.

The formulations are based on the following notion. Each agent in our network (with a total of N agents) acts as a routing node with a certain amount of bandwidth units, each of which can be used to route one call. It also maintains estimates (Q values) of the available bandwidth on all routes from each of its neighbouring agents to all other agents. Thus, the Q table entry $Q_i(d, n)$ indicates agent a_i's estimate of the bandwidth availability on all paths from its neighbour a_n to another node a_d. The information received from other agents is used as the "reward" to update these estimates (according to the standard model-free Q-update rule).

While routing a call, an agent probabilistically selects a neighbour for which it has the best (maximum bandwidth availability) estimate [3] and forwards it to that neighbour after reserving a bandwidth unit for the call. This forwarding process is aborted if either a node is reached that has no available bandwidth (hence, it is unable to route any more calls) or the time taken to forward exceeds a certain "setup" time, which is equivalent to the maximum (worst case) delay that a caller can experience between dialling a number and hearing the ring tone. In either case, the call fails to connect and the reserved bandwidth units of all nodes on the partially completed path are deallocated. A call successfully connects when it reaches the destination node. In that case, the bandwidth units of all nodes on the call path remain reserved for the duration of the call.

2.1. Continuous Inform and Update (CIU)

In this model of information sharing, an agent, a_i, while forwarding a call to its destination a_d, asks each of its immediate neighbours, a_j, for their best Q-estimate for that destination. It then uses the estimate $E_k(a_d)$ received from a_k, whom it selects (using Boltzmann exploration) to forward the call, to update its prior estimate $Q_i(d, k)$ as $Q_i(d, k) \leftarrow (1 - \alpha)Q_i(d, k) + \alpha E_k(a_d)$, where α is the learning rate. Note that agent a_i's prior estimates of its other neighbours remain unchanged. Therefore, this model allows information sharing and estimate updates on a continuous basis while the agents keep forwarding the call.

2.2. Delayed Inform Average Capacity (DI-A)

This is the first of two formulations of the delayed communication principle. The agents forward a call from source to destination using their current Q estimates (thus, not requesting neighbours like CIU). When the call reaches its destination a_d, it appends its bandwidth usage $b(d)$ (the fraction of the total bandwidth units available at that node) to a message m_c and sends this to the previous agent (node before the terminal node). This process is repeated by every agent on the path that the call was routed until the source node is reached. Upon receiving the m_c, an agent a_i using the DI-A model computes a reward R_i by *averaging* the $b(k)$ values of all nodes a_k on the route from a_i to a_d as $R_i = \frac{\sum_{k \in \{i+1,\ldots,d\}} b(k)}{L(a_i,a_d)}$, where $L(a_i, a_d)$ is the hop count (or, the number of agents) on this route from a_i to a_d. Then, it updates it prior estimate $Q_i(d, i + 1)$ as, $Q_i(d, i + 1) \leftarrow (1 - \alpha)Q_i(d, i + 1) + \alpha R_i$.

[2] We believe that our approach is generic and communication strategies based on the same principle can be formulated in other cooperative MAS.

[3] Here standard Boltzmann exploration is used.

2.3. Delayed Inform Minimum Capacity (DI-M)

This model is similar to DI-A: but instead of the average bandwidth availability, the *minimum* is used as reward.[4] For example, agent a_i using the DI-M model computes the reward $R_i = \min(b(i+1), ..., b(d))$. This is a more conservative estimate of bandwidth availability than the average capacity model. Thus it has the advantage that the probability of an agent overestimating the bandwidth availability at other nodes (hence, the chance of forwarding to an agent with no bandwidth causing a failure) is reduced.

3. Experimental Evaluation

This section reports the observations from empirical evaluation of the heuristics described in section 2. Their performance is also compared against a theoretical optimum strategy that works by globally searching for a call path and connecting the call instantaneously (the learning agents route calls one hop every simulated time step, thereby, incurring a finite delay) if such a path exists. Experiments are conducted on a number of different network topologies (figures 1, 2, and 3) to verify that the observations hold across a variety of conditions. Also, the impact of increasing network load on the performance is tested by increasing the probability with which calls originate at each simulation time step. Specifically, the performance of the three strategies is compared against the following properties in the context of the TN domain.

Call success rate: The total number of calls successfully routed to their corresponding destinations measures the overall success of the MAS. Specifically, the ratio of the total number of calls generated (NO) to the total number of these calls successfully connected (NC) is measured in steady state to indicate the overall call success rate ($x = \frac{NC}{NO}$) in the system. Also, NC is computed for the theoretical optimum strategy to indicate the optimum success rate, $x_{opt} = \frac{NC_{opt}}{NO}$. The percentage deviation ($\frac{x_{opt}-x}{x_{opt}}$) of the success rates from the theoretical optimum is also measured for comparison.

Success rate of different call lengths: This measures the success rate of calls ($x_d = \frac{NC_d}{NO_d}$) that have their source and destinations at a distance d (in terms of minimum hop count) apart. It indicates how effective a given communication strategy is in successfully routing calls at a given distance from the source node. Because routing a call to destinations further away is more difficult than to route to closer destinations,the communication strategy that achieves a higher value of x_d for larger d would be considered to generate more robust routing policies.

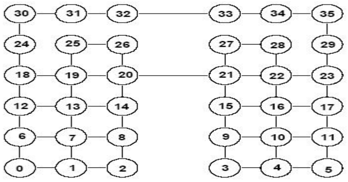

Figure 1. 36 node irregular grid

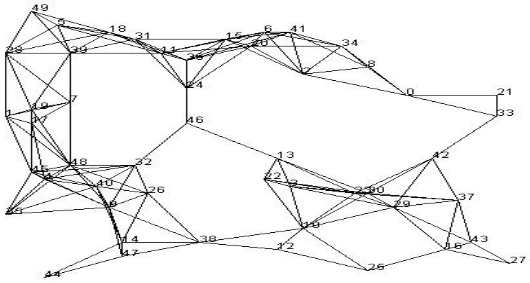

Figure 2. 50 node random graph

Message rate: The communication messages that contain the node state information contribute directly towards the quality of the estimates learned. Nevertheless, such messages pose an overhead to the system. The total number of these messages generated in the system, therefore, determine the “cost” of the corresponding communication strategy. Under identical simulation conditions, the strategy that achieves a better performance (in terms of any of the aforementioned measures) than the others at a lower message rate can, therefore, be considered the most cost efficient.

The following parameter values are chosen: α = 0.03, temperature (in Boltzmann exploration) = 0.1, number of bandwidth units per node = 10 (so, each node can simultaneously handle 10 calls), infinite capacity to handle the reward information messages, call setup time is the number of simulation time steps equal to the number of nodes in the topology used, call duration = 20 times setup time. One simulation run is 500,000 simulation time steps when figure 1 is used; 1,000,000 for figure 2; 2,000,000 for figure 3. Results are averaged over 10 simulation runs which means values are statistically significant at the 95% confidence level.

[4] Minimum is chosen because the maximum number of calls that can be routed along a path depends on the node with the minimum available bandwidth.

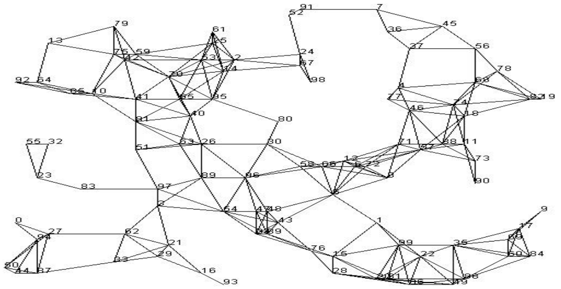

Figure 3. 100 node random graph

Load	CIU		DI-A		DI-M	
	Rate	Deviation	Rate	Deviation	Rate	Deviation
0.1	50.98	29.9	50.94	29.3	51.85	26.2
0.2	32.41	46.6	32.64	46.1	33.02	43.9
0.4	19.99	63.9	20.27	63.3	20.38	61.4
0.6	14.87	72.9	15.07	71.8	15.08	70.2

Table 1. Call success rates for all strategies — topology of figure 1

3.1. Results — Call Success Rate

Tables 1, 2, and 3 show the average steady state call success rate and its deviation from the optimal for all three strategies under various loads for the topologies of figures 1, 2, and 3, respectively. DI-M achieves the highest rate of successful call connections and the least deviation from the optimal — both indicative of better performance than the other strategies. The success rate and the deviation values worsen with increasing network load. This is because the nodes have a fixed number of bandwidth units and, hence, fail to connect the majority of the calls as they increase in number.

We also study the improvement in the deviation values achieved by a DI strategy over the benchmark (for different network loads) in figures 4, 5, and 6. The graphs indicate how much closer is the success rate to the optimal for DI-A and DI-M relative to CIU. These values are measured as $\frac{v^{DI} - v^{CIU}}{v^{CIU}}$, where v^x is the success rate deviation from optimal of strategy x for a given load. Hence, the smaller these values (the more negative), the better is the performance of the corresponding DI strategy relative to CIU. Again, DI-M achieves a significant benefit relative to CIU under all loads and all topologies.

Load	CIU		DI-A		DI-M	
	Rate	Deviation	Rate	Deviation	Rate	Deviation
0.1	58.15	29.9	57.31	28.8	58.49	27.7
0.2	37.32	46.8	37.11	45.7	37.65	44.9
0.4	22.98	62.6	23.35	62.7	23.51	61.1
0.6	17.17	70.8	17.47	71.5	17.49	69.6

Table 2. Call success rates for all strategies — topology of figure 2

Load	CIU		DI-A		DI-M	
	Rate	Deviation	Rate	Deviation	Rate	Deviation
0.1	35.79	46.2	35.86	45.3	37.34	40.4
0.2	24.56	57.9	24.17	58.5	24.8	54.3
0.4	16.16	70.6	15.88	70.9	16.18	68.1

Table 3. Call success rates for all strategies — topology of figure 3

3.2. Results — Success Rate at Different Lengths

The percentage improvement in the values of x_d (defined earlier in this section), $pc = \frac{x_d^{DI-Y} - x_d^{CIU}}{x_d^{CIU}}$ (where, Y is A or M), for increasing values of d, are shown in figures 7, 8, and 9 for the three topologies using different network loads. For calls that need to be routed very short distances (e.g., values of 1 and 2 on the "minimum hop count" axis in figure 7), CIU performs slightly better than the DI strategies (indicated by a small negative pc). Nevertheless, the advantage of the DI strategies is obtained in their significantly higher rate of success for calls at longer distances (indicated by the positive pc). Additionally, the pc values are higher for longer distances as the network load increases. Thus the impact of load on the DI strategies is less (thus, indicating greater robustness) in connecting long-distance calls (although all strategies suffer with increasing load as shown in section 3.1) — further substantiating the usefulness of the principle of delayed communication.

3.3. Results — Information Message Rate

The number of information messages transmitted in the entire system per unit time step is shown in tables 4, 5, and 6 for all three strategies with different loads and topologies. Both DI-M and DI-A achieve significant amount of savings

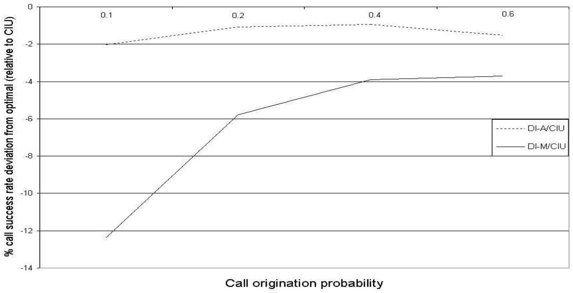

Figure 4. Improvement of success rate deviation over CIU — topology of figure 1

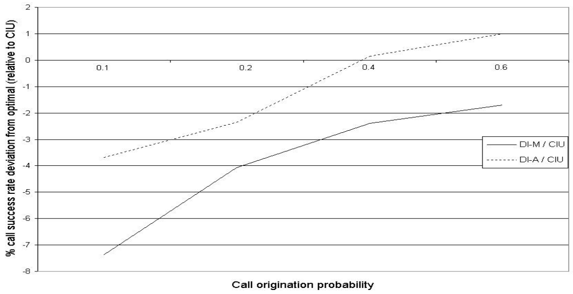

Figure 5. Improvement of success rate deviation over CIU — topology of figure 2

in the number of messages exchanged compared to CIU, indicating a greater efficiency of the delayed information principle in terms of communication cost. Also, the increase in the number of messages exchanged due to increasing load is much less in DI strategies than CIU — further establishing their benefit. The main reason for the difference in the message rate of CIU and DI-M or DI-A is that the agents using the DI strategies transmit their state information only after a call connects. So, for all failed calls (which happen more with increased network load), they do not transmit any messages whereas those using CIU continue to do so.

Load	Strategies			% Saving	
	CIU	DI-A	DI-M	DI-A / CIU	DI-M / CIU
0.1	0.39	0.26	0.251	33.3	35.64
0.2	0.66	0.281	0.275	57.42	58.33
0.4	1.1	0.289	0.284	73.73	7418
0.6	1.45	0.2894	0.285	80.04	80.34

Table 4. Information message rates for all strategies — topology of figure 1

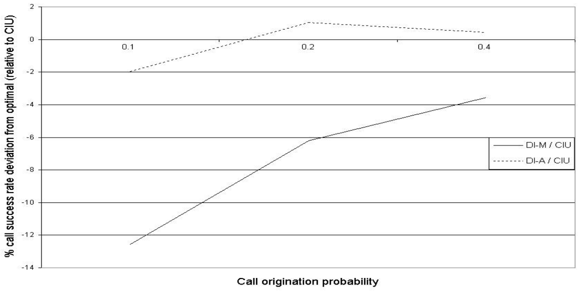

Figure 6. Improvement of success rate deviation over CIU — topology of figure 3

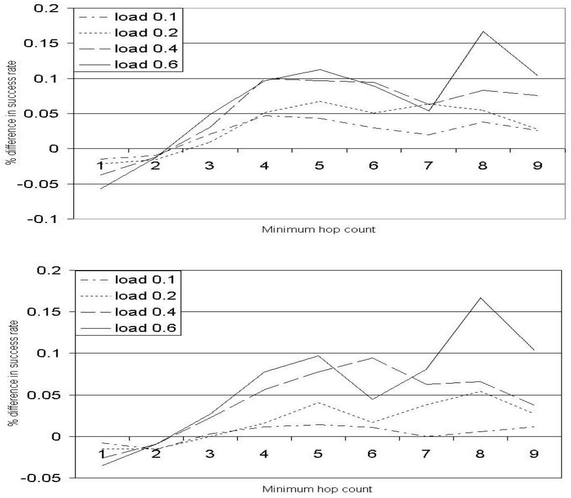

Figure 7. Call success rates at various distances. DI-M vs. CIU (top), and DI-A vs. CIU (bottom) — topology of figure 1

4. Conclusions and Future Work

In this paper, we have proposed the principle of delayed information sharing to improve learning in cooperative MAS. Its advantages are demonstrated using empirical studies on a simulated network routing problem. Two communication heuristics, based on this principle, are designed for this domain and shown to achieve better performance than the widely adopted protocol in [2].

Extending our current work, we aim to theoretically explain the impact of this communication principle on learning quality and, hence, on the overall system performance. Also, further experiments will be conducted to study performance dependence on domain properties such as traffic patterns and network structures.

Acknowledgement: This research is part of the EPSRC-funded project MOHICAN (GR/R32697/01).

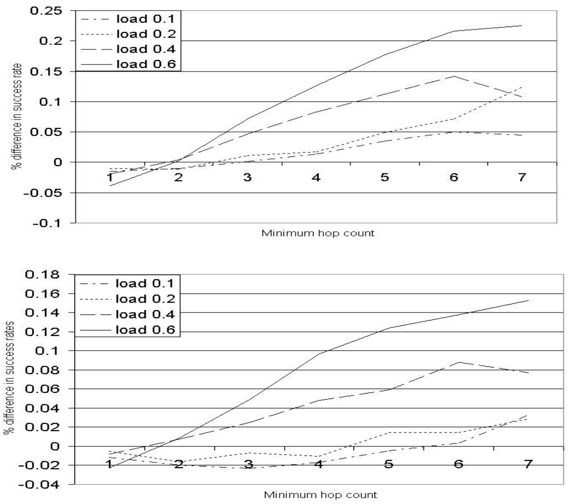

Figure 8. Call success rates at various distances. DI-M vs. CIU (top), and DI-A vs. CIU (bottom) — topology of figure 2

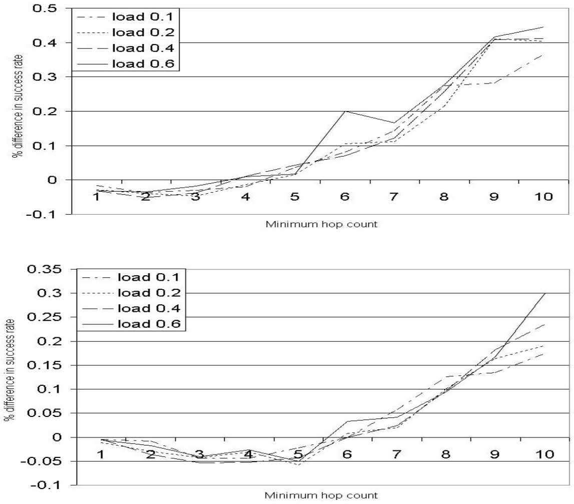

Figure 9. Call success rates at various distances. DI-M vs. CIU (top), and DI-A vs. CIU (bottom) — topology of figure 3

Load	Strategies			% Saving	
	CIU	DI-A	DI-M	DI-A / CIU	DI-M / CIU
0.1	0.33	0.23	0.23	30.3	30.3
0.2	0.55	0.26	0.25	52.73	54.55
0.4	0.9	0.27	0.26	70.0	71.11
0.6	1.2	0.272	0.267	77.33	77.75

Table 5. Information message rates for all strategies — topology of figure 2

Load	Strategies			% Saving	
	CIU	DI-A	DI-M	DI-A / CIU	DI-M / CIU
0.1	0.58	0.214	0.21	63.1	63.79
0.2	0.83	0.235	0.229	71.69	72.41
0.4	1.26	0.249	0.241	80.24	80.87
0.6	1.63	0.253	0.245	84.48	84.97

Table 6. Information message rates for all strategies — topology of figure 3

References

[1] J. Baxter and P. L. Bartlett. Direct gradient-based reinforcement learning: I. Gradient estimation algorithms. Technical report, Research School of Information Science and Engineering, Australian National University, 1999.

[2] J. A. Boyan and M. L. Littman. Packet routing in dynamically changing networks: A reinforcement learning approach. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 671–678, 1993.

[3] R. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy-gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12:1057–1063, 2000.

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

[5] A. S. Tanenbaum. *Computer Networks*, chapter 5: The Network Layer. Prentice Hall PTR, 4th edition, 2003.

[6] C. J. C. H. Watkins and P. Dayan. Technical note: Q-learning. *Machine Learning*, 8:279–292, 1992.

[7] B. Widrow and M. E. Hoff. Adaptive switching circuits. In *WESCON Convention Record Part IV*, pages 96–104. 1960.

[8] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229 – 256, 1992.