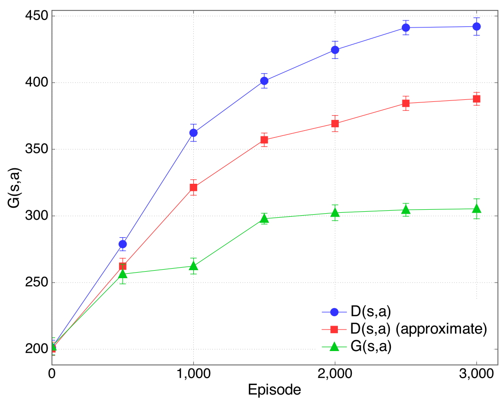

Figure 1: Rover domain results (10 agents). Approximating $D_i(s,a)$ results in 88% of the performance attained when analytically computing $D_i(s,a)$. Approximating the difference evaluation function results in significant performance gains when compared to using the system evaluation function $G(s,a)$.

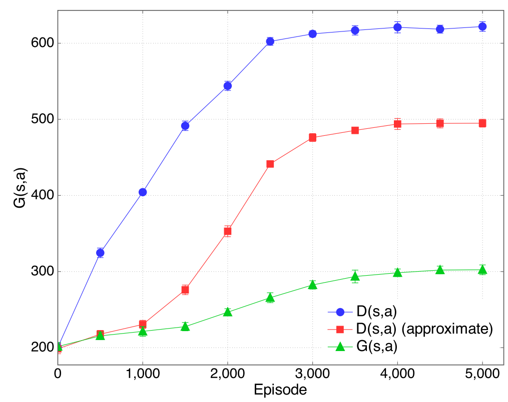

Figure 2: Rover domain results (100 agents). Approximating $D_i(s,a)$ results in 79% of the performance attained when analytically computing $D_i(s,a)$. Approximating the difference evaluation function results in significant performance gains when compared to using the system evaluation function $G(s,a)$.

formation, as well as the broadcast value of the system evaluation function. This information is typically available in any multiagent learning system.

This approximation approach is tested in a multiagent rover domain (See [2] for details on implementation), where a set of rovers move in a planar world in order to observe points of interest. Agents are trained using a cooperative coevolutionary algorithm, and fitness values are assigned with either $G(s,a)$, $D_i(s,a)$, or $\hat{D}_i(s,a)$.

3. RESULTS

The rover domain experiments were initialized as follows. For the first experiment, there are 10 agents and 10 points of interest in a 25 by 25 unit planar world. For coevolution, each agent maintains a population of 25 neural network policies. Each episode lasts 25 timesteps, and the coevolutionary algorithm is allowed to run for 3000 generations. 150 statistical runs were conducted, and reported error bars represent error in the mean. For the second experiment, there are 100 agents and 100 points of interest in a 50 by 50 unit planar world, and learning proceeds for 5000 generations. Other parameters are identical to the first experiment.

Results for the 10 agent rover domain are shown in Figure 1. Approximating $D_i(s,a)$ results in 23% better performance compared to $G(s,a)$, and achieves 88% of the performance when analytically computing $D(s,a)$. Although $\hat{D}_i(s,a)$ results in 12% lower performance than $D(s,a)$, it requires 90% less information to compute, demonstrating the approximation is effectively utilizing locally available information.

Results for the 100 agent rover domain are shown in Figure 2. $\hat{D}_i(s,a)$ results in 49% better performance than $G(s,a)$, and achieves 79% of the performance of analytically computing $D_i(s,a)$. It is of note that in this larger domain, although $\hat{D}_i(s,a)$ performs worse compared to $D_i(s,a)$ (79% vs. 88%), it outperforms $G(s,a)$ by a wider margin (49% vs 23%). Additionally, in this larger domain, $\hat{D}_i(s,a)$ requires even less information than $D_i(s,a)$ compared to the 10 agent domain (99% less vs. 90% less). This demonstrates that $\hat{D}_i(s,a)$ scales well with the number of agents in the system.

4. DISCUSSION

Although difference evaluation functions have produced excellent results in many multiagent settings, their requirement for global state and action information makes them difficult to compute in practice. The contribution of this work is to demonstrate that agents may approximate difference evaluations requiring only local knowledge. Our results demonstrate that the approximation uses far less information than $D_i(s,a)$ (90-99% less), but still achieves comparable performance (up to 88%). The information requirements for this approximation technique are equivalent to traditional multiagent learning techniques, allowing for the implementation of difference evaluations in any multiagent system where the system evaluation function is broadcast.

5. ACKNOWLEDGEMENTS

This work was partially supported by the NETL under grants DE-FE0012302 and DE-FE0011403.

6. REFERENCES

[1] A. K. Agogino and K. Tumer. Analyzing and Visualizing Multiagent Rewards in Dynamic and Stochastic Environments. *Journal of Autonomous Agents and Multi-Agent Systems*, 17(2):320–338, 2008.

[2] M. Colby and K. Tumer. Shaping Fitness Functions for Coevolving Cooperative Multiagent Systems. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Valencia, Spain, June 2012.

[3] S. Proper and K. Tumer. Modeling Difference Rewards for Multiagent Learning (Extended Abstract). In *Proceedings of the Eleventh International Joint Conference on Autonomous Agents and Multiagent Systems*, Valencia, Spain, June 2012.