As the static potential function, used for $DRiP$ and $G+ManualPBRS$, we simply direct the agents to move away from the center point to encourage exploration:

$$\Phi(s) = dist(agent, center) \quad (23)$$

where *center* is the point at the center of the grid world near which the agents started.

Note that this potential function offers no operational knowledge about the location or value of the POIs, and merely serves to encourage the agents to explore away from the center. This is a contrasting type of potential function to that employed in the BPD, which expressed significant domain knowledge.

5.2 Experimental Setup

We present two studies in the GWD; the first is on a 10x10 grid with deterministic actions whilst the second is on a 100x100 grid approximated by the agents by a single 10x10 tiling with 5% chance of action failure. The first setting ran 2500 episodes of 50 time steps. The second ran 2500 episodes of 250 time steps.

The experimental parameters were as follows: $\alpha = 0.1$, $alpha\_decay\_rate = 0.9999$, $\epsilon = 0.2$, $epsilon\_decay\_rate = 0.9999$, $\gamma = 0.9$, $num\_agents = 4$, and $num\_POIs = 4$. All POIs have a value of 1, except one that is chosen at random at the beginning of a run to have value 5. By assigning the value of POIs randomly, designing a suitable heuristic becomes significantly harder.

In all cases, regardless of the learning signal, we report on the summation of system-level performance G over all time steps and include error bars representative of the standard error of the mean based on 30 statistical runs. Specifically, we calculate the error as $\sigma/\sqrt{n}$ where σ is the standard deviation and n is the number of statistical runs. For some reward functions, in some figures the error bars are smaller than the symbols used to plot the result. However, they are present on all graphs for all reward functions. Please note, given the longer episodes, the agents in the second setting can receive significantly higher rewards.

5.3 Experimental Results

In the first instance of the problem domain, the results presented in Figure 5 show that $G+CaP$ again significantly outperforms G alone and $DRiP$ again significantly outperforms all other solutions.

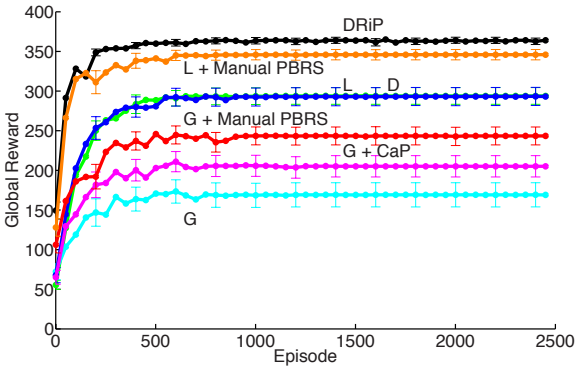

Figure 5: 10x10 GridWorld Domain

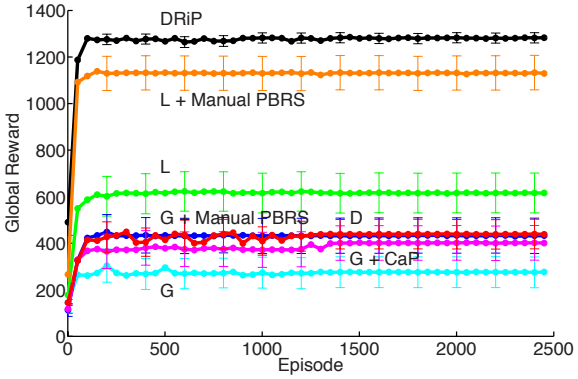

Figure 6: 100x100 Stochastic GridWorld Domain

The largest difference in this result compared to those in the BPD is that agents learning by the local reward can now match those using difference rewards. This occurs because agents acting in their own self interest have a very high signal-to-noise ratio on their local learning signal, allowing them to quickly discern which of their actions increased their reward. In this domain agents are less likely to work at cross-purposes than in the BPD, where the congestion creates a situation where doing so is more likely. This observation further emphasises the difference between the two domains presented in this paper, supporting our argument that CaP and $DRiP$ are generally applicable.

Given the high performance of agents using local rewards, we performed additional experiments with $L+ManualPBRS$. As with global rewards, the manual heuristic used is suitable in this instance of the problem domain and, therefore, $L+ManualPBRS$ significantly outperformed L. However, despite being equal to difference rewards when neither received potential-based reward shaping, $DRiP$ still significantly outperformed $L+ManualPBRS$.

This result becomes even more impressive when we move on to the larger instance of the problem domain, results from which are presented in Figure 6. In this example, agents learning by local rewards alone now significantly outperform those learning with difference rewards alone.

However, agents learning from $DRiP$ are still the best performing of all agents. The effect of adding potential-based reward shaping to difference rewards (a 196% increase in performance) is significantly larger than adding it to local rewards. By exploiting domain specific knowledge, $DRiP$ is able to learn a suitable policy in a problem domain where difference rewards alone cannot.

6. DISCUSSION

To conclude this paper we collate our experiences of using $DRiP$ and CaP in all domains studied. We have seen repeatedly throughout that potential-based reward shaping and difference rewards can be used together to leverage the benefits of both.

$DRiP$ consistently outperforms any other reward function combining L, G, D and $PBRS$. If suitable domain specific knowledge is available, exploiting it by potential-based reward shaping in addition to difference rewards significantly improves agents' learning performance over using difference rewards alone.

CaP provides a method of automatically generating a dy-

namic potential function for multiagent problem domains. CaP will be useful in future multiagent applications as it removes the need to implement a domain-specific potential function. Furthermore, CaP maintains the theoretical guarantee of consistent Nash equilibria provided by the proofs for dynamic and multiagent potential-based reward shaping.

As CaP captures the same knowledge represented by difference rewards, using CaP with $DRiP$ provides no further advantages. In our experiments this combination typically performed worse than difference rewards alone. This occurs because these two signals represent the same knowledge by different methods. Combining them increases the noise of the signal whilst not increasing how informative it is.

The computational cost of CaP is equivalent to difference rewards alone. However, dependent on the complexity of the potential function used, $DRiP$ may be more computationally expensive. In all examples studied the calculation of D is more complex than the potential functions used and, therefore, the theoretical bound of complexity is unaffected.

Furthermore, it is known that difference rewards work best if the original reward function has a gradient throughout [1]. Therefore, given the close links, so do CaP and $DRiP$. Problem domains with a piecewise reward function may still benefit from CaP and $DRiP$, however, if the reward function can be gradual the positive effect is likely to be larger.

Finally, according to the proof of necessity for potential-based reward shaping [10], there must exist a problem domain for which difference rewards alter the Nash equilibria of the system. Therefore, if an application specifically requires the theoretical guarantees of potential-based reward shaping, we would recommend CaP as it benefits from the theoretical properties of both difference rewards and potential-based reward shaping whilst significantly improving performance over G alone. Alternatively, if theoretical guarantees are not a concern, we recommend $DRiP$ for its significantly better performance.

7. REFERENCES

[1] A. Agogino and K. Tumer. Unifying temporal and structural credit assignment problems. *Proc. of the 3rd Intl. Jt. Conf. on Autonomous Agents and Multiagent Systems-Volume 2*, pages 980–987, 2004.

[2] 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.

[3] M. Babes, E. de Cote, and M. Littman. Social reward shaping in the prisoner's dilemma. In *Proceedings of The 7th Annual International Conference on Autonomous Agents and Multiagent Systems*, volume 3, pages 1389–1392, 2008.

[4] L. Busoniu, R. Babuska, and B. De Schutter. A Comprehensive Survey of MultiAgent Reinforcement Learning. *IEEE Transactions on Systems Man & Cybernetics Part C Applications and Reviews*, 38(2):156, 2008.

[5] S. Devlin, M. Grześ, and D. Kudenko. An empirical study of potential-based reward shaping and advice in complex, multi-agent systems. *Advances in Complex Systems*, 2011.

[6] S. Devlin and D. Kudenko. Theoretical considerations of potential-based reward shaping for multi-agent systems. In *Proceedings of The 10th Annual International Conference on Autonomous Agents and Multiagent Systems*, 2011.

[7] S. Devlin and D. Kudenko. Dynamic potential-based reward shaping. In *Proceedings of The 11th Annual International Conference on Autonomous Agents and Multiagent Systems*, 2012.

[8] M. Grzes and D. Kudenko. Plan-based reward shaping for reinforcement learning. *4th International IEEE Conference on Intelligent Systems*, 2:10–22, 2008.

[9] M. Knudson and K. Tumer. Coevolution of heterogeneous multi-robot teams. *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, 2010.

[10] A. Y. Ng, D. Harada, and S. J. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. *Proceedings of the 16th International Conference on Machine Learning*, pages 278–287, 1999.

[11] S. Proper and K. Tumer. Coordinating actions in congestion problems: Impact of top-down and bottom-up utilities. *Autonomous Agents and MultiAgent Systems*, 27(3):419–443, 2013.

[12] J. Randløv and P. Alstrom. Learning to drive a bicycle using reinforcement learning and shaping. *Proceedings of the 16th International Conference on Machine Learning*, pages 463–471, 1998.

[13] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[14] K. Tumer and A. Agogino. Distributed agent-based air traffic flow management. *Proceedings of the 6th International Conference on Autonomous Agents and Multiagent Systems*, pages 330–337, 2007.

[15] K. Tumer and A. Agogino. Multiagent learning for black box system reward functions. *Advances in Complex Systems*, 12:493–512, 2009.

[16] M. Vasirani and S. Ossowski. A market-inspired approach to reservation-based urban road traffic management. *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, 2009.

[17] C. Watkins and P. Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

[18] E. Wiewiora. Potential-based shaping and q-value initialization are equivalent. *Journal of Artificial Intelligence Research*, 19(1):205–208, 2003.

[19] E. Wiewiora, G. Cottrell, and C. Elkan. Principled methods for advising reinforcement learning agents. *Proceedings of the 20th International Conference on Machine Learning*, pages 792–799, 2003.

[20] D. H. Wolpert, J. Sill, and K. Tumer. Reinforcement learning in distributed domains: Beyondteam games. In *Proc. of the 17th Int. Jt. Conf. on Artificial Intelligence*, pages 819–824, Seattle, WA, 2001.

[21] D. H. Wolpert and K. Tumer. Collective intelligence, data routing and Braess' paradox. *Journal of Artificial Intelligence Research*, 16:359–387, 2002.

[22] M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley and Sons, 2002.