Collective Coordination and Agent Symmetry in Multi-Agent Dispersion Games

Pieter Jan 't Hoen and Sander M. Bohte

CWI, Centrum Wiskunde & Informatica, Kruislaan 413, NL-1097 CA Amsterdam, The Netherlands
{hoen,sbohte}@cwi.nl

Summary. The design of a Multi-Agent System (MAS) to perform well on a collective task is non-trivial. Straightforward application of Reinforcement Learning techniques in a MAS can lead to sub optimal solutions as agents compete or interfere. The COllective INtelligence (COIN) framework of Wolpert et al. proposes an engineering solution for MASs where agents learn to focus on actions which support a common task. Here, we study various dispersion games where fine-grained coordination between the agents is required. Although we show that the COIN framework can successfully solve reasonably simple versions of these retrieval problems, the performance for more complex games – more representative of real-life scenarios – is less than optimal. We show how the individual agent utility functions can be shaped to exploit the particular structure of dispersion-type games. These advances to the COIN framework dramatically improve convergence results for MAS with a large number of agents. The increased convergence properties for the dispersion games are competitive with especially tailored strategies for solving dispersion games. The enhancements to the COIN framework proved to be essential to solve the more complex variants of token retrieval dispersion games, and they point the way to how a MAS can be applied in real life coordination problems.

1 Introduction

In our increasingly connected world, solving complex computational problems that involve many parties and many resources is quickly becoming essential for daily life. Finding conceptual approaches to these problems is both paramount and difficult. As any computational problem can be considered as a resource allocation problem (Wellman, 1996a, 1996b), the insight from economics is that few concepts for resource allocation scale well with increasing complexity of the problem domain. Centralized allocation planning in particular can quickly reach a point where the design of satisfying solutions becomes complex and intractable. Distributed systems may be one of the few concepts that do allow for scaling, both of resource allocation, and analogously for solving complex computational problems.

Conceptually, an attractive option for resource allocation is to devise a distributed system where different parts of the system each contribute to the solution for the allocation problem. Embodied in a so-called distributed Multi-Agent System (MAS), the aim is to elicit "emergent" behavior in the form of overall efficient resoure allocation from a collection of individual agents that each solve a part of the problem. In typical problem settings, individual agents in the MAS contribute to some part of the collective through their private actions. The joint actions of all agents derive some reward from the outside world. To enable local learning, this reward has to be divided amongst the individual agents where each agent aims to increase its received reward by some form of learning. However, unless special care is taken as to how this reward is shared, there is a risk that agents in the collective work at cross-purposes. For example, agents can reach sub-optimal solutions by competing for scarce resources or by inefficient task distribution among the agents as they each only consider their own goals. A prime example of this potential conflict is embodied in the well known "Tragedy of the Commons" (Hardin, 1968).

A weak point of distributed Multi-Agent systems has however long been the typical bottom-up type of approach: researchers first build an intuitively reasonable system of agents and then use heuristics and tuned system parameters such that – hopefully – the desired type of behavior emerges from running the system. Only recently has there been work on more top-down type of approaches to establish the conditions for MASs such that they are most likely to exhibit good emergent behavior (Barto & Mahadevan, 2003; Lauer & Riedmiller, 2000; Guestrin, Lagoudakis, & Parr, 2002), see also the survey in (Panait & Luke, 2005).

The COllective INtelligence (COIN) framework, as introduced by Wolpert et al., suggests how to engineer (or *modify*) the rewards an agents receives for its actions (and to which it adapts to optimize) in *private utility functions*. Optimization of each agent's private utility here leads to increasingly effective emergent behavior of the collective, while discouraging agents from working at cross-purposes.

In particular, the work by Wolpert et al. explores the conditions sufficient for effective emergent behavior for a collective of independent agents, each employing Reinforcement Learning (RL) for optimizing their private utility. These conditions relate to (i) the learnability of the problem each agent faces, as obtained through each individual agent's private utility function, (ii) the relative "alignment" of the agents' private utility functions with the utility function of the collective (the *world utility*), and lastly (iii) the learnability of the problem. Whereas the latter factor depends on the considered problem, the first two in COIN are translated into conditions on how to shape the private utility functions of the agents such that the world utility is increased when the agents improve their private utility. This allows an agent to optimize its reward, without decreasing the utility of the collective.

Wolpert et al. have derived private utility functions that perform well on the above first two conditions listed above. The effectiveness of this top-down approach and their developed utilities are demonstrated by applying the COIN framework to a number of example problems: network routing (D. Wolpert, Turner, & Frank, 1998), the El Ferrol Bar problem (D. Wolpert & Tumer, 1999), Braess' paradox (Tumer & Wolpert, 2000), influencing Google pageranks (Agogino & Ghosh, 2002), and air traffic control (Tumer & Agogino, 2007). The COIN approach proved to be very effective for learning these problems in a distributed system.

The COIN work has been expanded upon to include distributed function optimization with bounded rational agents, in the form of Probability Collectives (D. Wolpert & Bieniawski, 2004; Lee & Wolpert, 2004; D. Wolpert, Strauss, & Rajnayaran, 2006). In this formulation, "agents" each optimize a single variable in a function optimization problem. Such an approach handles constraints well, and has been shown to outperform Genetic Algorithms for a number of challenging optimization problems (Huang, Bieniawski, Wolpert, & Strauss, 2005).

The work on COIN and recently on probability collectives by Wolpert et al. suggests that this framework may be a concept that does scale well, and they demonstrated this in the case where the problem complexity is increased by adding more agents into the system. This still leaves open the question of how COIN scales with other problem properties, in particular related to the amount of cooperation needed between the agents

Here, we study the application of the COIN framework to games where different agents choose distinct actions, so called *anti-coordination* or *dispersion* games. Such problems are typical for a growing class of large-scale distributed applications such as load balancing (e.g. (Azar, Broder, Karlin, & Upfal, 2000)), division of roles within robotics, or application in logistics. Many *niche selection* problems studied in economics and evolutionary biology are also natural applications of dispersion games (Grenager, Powers, & Shoham, 2002). Games like minority games (Challet & Zhang, n.d.) or variants of the El Farol Bar problem (Arthur, 1994) can be modeled as dispersion games in a straightforward way.

We explore scaling in dispersion games: (Grenager et al., 2002) have shown near exponential complexity of the empirical performance of many algorithms for the convergence of the system versus the number of participating agents. We focus in particular on games where there is *agent and action symmetry*: an agent's preference over outcomes depends only on the overall configuration of actions and agents, but not on particular identities of the agents or actions (confer also (Grenager et al., 2002)). We investigate coordination in the form of task allocation: the allocation of n agents to k tasks. Agents acting in parallel and using local feedback with no central control must learn to arrive at an optimal distribution over the available tasks.

Using the standard COIN framework, we find increasingly slow convergence with increasing size of the MAS, and rapidly decreasing performance for more

complex dispersion games. However, with agent and action symmetry, it is easy to see that the standard COIN utility function – the Wonderful Life Utility (WLU) – has agents *learning* at cross-purpose. As a number of agents select a particular task, *all* agents effectively receive a penalty for doing so. We show that taking this into account, and enhancing the WLU function, vastly speeds up the convergence of the symmetric dispersion problems, and also allows for better solutions to be found for more complex instantiation of the problem, where the traditional WLU exhibits highly unsatisfactory performance. As we remarked, dispersion games model many important natural problems. By enhancing the COIN framework to exploit the inherent symmetry in many of instantiations of these games, we are able to materially improve convergence speed and performance for distributed means of solving these problems.

2 COllective INtelligence

Here, we briefly outline the theory of COIN as developed by Wolpert et al., e.g. (D. H. Wolpert, Wheeler, & Tumer, 1999; D. Wolpert & Tumer, 1999, 2001). Broadly speaking, COIN defines the conditions that an agent's private utility function has to meet to increase the probability that learning to optimize this function leads to increased performance of the collective of agents. Thus, the challenge is to define a suitable private utility function for the individual agents, given the performance of the collective.

In particular, the work by Wolpert et al. explores the conditions sufficient for effective emergent behavior for a collective of independent agents, each employing, for example, Reinforcement Learning (RL) for optimizing their private utility. These conditions relate to (i) the learnability of the problem each agent faces, as obtained through each individual agent's private utility function, (ii) the relative "alignment" of the agents' private utility functions with the utility function of the collective (the *world utility*), and lastly (iii) the learnability of the problem. Whereas the latter factor depends on the considered problem, the first two in COIN are translated into conditions on how to shape the private utility functions of the agents such that the world utility is increased when the agents improve their private utility.

Formally, let ζ be the joint moves of all agents. A function $G(\zeta)$ provides the utility of the collective system, the *world utility*, for a given ζ. The goal is to find a ζ that maximizes $G(\zeta)$. Each individual agent η has a private utility function g_η that relates the reward obtained by the collective to the reward that the individual agent collects. Each agent will act such as to improve its own reward. The challenge of designing the collective system is to find private utility functions such that when individual agents optimize their payoff, this leads to increasing world utility G, while the private function of each agent is at the same time also easily learnable (i.e. has a high *signal-to-noise* ratio, an issue usually not considered in traditional mechanism design). In this paper, ζ represents the choice of which of the k tasks each of the n agent chooses

to execute and the challenge is to find a private function for each agent such that optimizing the local payoffs optimizes the total task execution.

Following a mathematical description of this issue, Wolpert et al. propose the **Wonderful Life Utility** (WLU) as a private utility function that is both *learnable* and *aligned* with G, and that can also be easily calculated.

$$WLU_\eta(\zeta) = G(\zeta) - G(CL_{S_\eta^{eff}}(\zeta)) \tag{1}$$

The function $CL_{S_\eta^{eff}}(\zeta)$ as classically applied "clamps" or suspends the choice of task by agent η and returns the utility of the system without the effect of agent η on the remaining agents $\hat{\eta}$ with which it possibly interacts. For our problem domain, the clamped effect set are those agents $\hat{\eta}$ that are influenced in their utility by the choice of task of agent η. Hence $WLU_\eta(\zeta)$ for agent η is equal to the value of all the tasks executed by all the agents minus the value of the tasks executed by the other agents $\hat{\eta}$. If agent η picks a task τ, which is not chosen by the other agents, then η receives a reward of $V(\tau)$, where V assigns a value to a task τ. If this task is however also chosen by any of the other agents, then the first term $G(\zeta)$ of Equation 1 is unchanged while the second term increases with the value of $V(\tau)$ as agent η no longer competes for completion of the task as η is clamped. Agent η then receives a penalty $-V(\tau)$ for competing for a task targeted by one of the other agents $\hat{\eta}$. The WLU hence has a built in incentive for agents to find an unfulfilled task and hence for each agent to strive for a high global utility in its search for maximizing its own rewards.

Compared to the WLU function, other payoff functions have been considered in the literature for distributed Multi-Agent Systems: the Team Game utility function (TG), where the world-utility is equally divided over all participating agents, or the Selfish Utility (SU), where each agent only considers the reward that it itself collects through its actions. The TG utility can suffer from poor learnability, as for larger collectives it becomes very difficult for each agent to discern what contribution is made (low signal-to-noise ration), and the SU suffers from – potentially – poor alignment with the world-utility, i.e. agents can work at cross purposes. TG and SU are representative for types of utility often found in the literature. We compare the performance of the SU and TG relative to the variants of the WLU.

We use Q-learning (Sutton & Barto, 1998) as RL algorithm for each of the n agents in the MAS. A learner's input space consists of the available k tasks. Q-learning that proved to work well for dispersion games; COIN learners have also been implemented with other RL methods, such as e.g. Q(λ) (Hoen & Bohte, 2003), as the COIN principles are generically implementable by any sufficiently powerful RL method. In our Q-learning implementation, the policy π is stochastic according to a softmax function; in the policy, a random task k_i is chosen for state s and constant c (set at 50) with normalized chance in $[0, 1]$ of $\frac{c^{Q(s,k_i)}}{\sum_j c^{Q(s,k_j)}}$. As each agent only must choose one task/action, we use a single state per agent. The discount factor γ is set to 0.95. The learning rate

α, unless specified otherwise, is set at 1 as this produced best results for all the utility functions considered. The next section presents application of the RL learners for a MAS task assignment problem.

2.1 Dispersion Games

Dispersion games (Alpern, 2001; Grenager et al., 2002) are a general class of problems where n agents each have to decide which of the k tasks they are to undertake. We follow the formal definition of dispersion games of (Grenager et al., 2002), where a dispersion game is defined as a subclass of *normal form games*. We investigate the case where $n = k$ and the agents are fully symmetric (i.e. it does not matter which agents executes which task). This type of dispersion game is called the *full dispersion game*. Full utility is achieved only when all k tasks are chosen by exactly one of the n agents.

2.2 Exploiting Symmetry in Dispersion Games

We observe that the WLU is symmetric when there is action and agent symmetry in the game, as in the dispersal games we study. If, for example, two agents a_1 and a_2 both choose task k_i, then both agents, according to equation 1, receive a penalty when calculating $WLU_{a_1}(\zeta)$ and $WLU_{a_2}(\zeta)$ respectively. This however can lead to slower convergence as *both* agents then may be forced to target different tasks while only *one* of the agents need choose a different task. This slower convergence becomes more dramatic as more than one agent, say $l > 2$ agents, focuses on the same task and $l - 1$ agents need to "switch".

We break the symmetry in the penalties of the WLU in two ways. First of all, we consider the case where one of the agents η targeting a task k is $r$andomly chosen as the winner and is awarded the positive reward while the other $\hat{\eta}$ agents choosing the same task k are penalized. We name this the $WLUr$ as we consider a $r$andom winner in which of the agents happen to arrive at a specific task. Secondly, as a more refined variant of the WLUr, we consider the case where the positive reward is assigned to the agent that is most likely to choose action k. We reward agent η with the highest Q-value for this task. We name this the $WLUm$ from **m**ost likely. Conceptually, no internal knowledge of the agent is needed with the WLUr, whereas the WLUm requires access to the actual RL strategy of the individual agents to compute the respective agent utilities.

3 COIN for Dispersion Games

We first discuss some results from (Grenager et al., 2002) for the $n = k$ dispersion game setting where agents use different strategies for choosing their tasks[1]. As analyzed in (Grenager et al., 2002), for $n = k$, the expected time

[1] See (Grenager et al., 2002) for details and references.

to successful allocation for a naive strategy with random choices by agents is $n^n/n!$. This is exponential in n. Similar long time to convergence results where found for Fictitious play, even with slight modifications to the updates of beliefs to avoid oscillatory behavior within sets of suboptimal outcomes. Better results where found using RL with a Q-learning Algorithm with a Boltzmann exploration policy. The agents learned the expected reward for choosing a specific task. The (selfish) reward for each of the agents is a function of the number of agents that use the same action. For this setting, with a well chosen temperature decay trajectory, a polynomial time to convergence was found for convergence to the optimal solution. Similar convergence results are found for the *Freeze* strategy where an action is chosen randomly by an agent until the first time it is alone in choosing an action, at which point the agent replays that action indefinitely. Best results where found for the *Basic Simple Strategy* (BS) and the *Extended Simple Strategy* (ES) where agents quickly focus on a task when they are the only candidate and otherwise stochastically choose from the *remaining* tasks that are still under contention. See Figure 1 for an overview of the results.

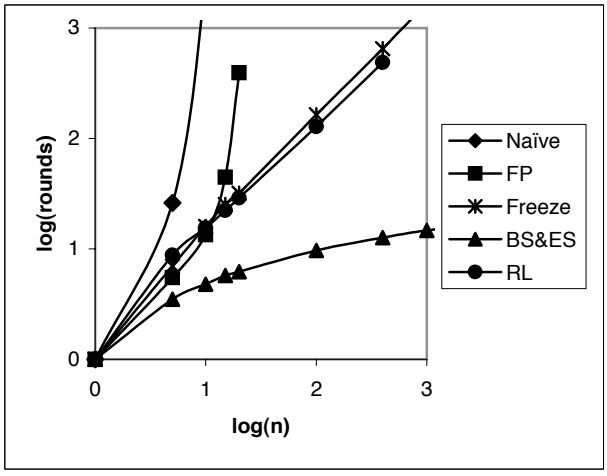

Fig. 1. Log-log plot of the emperical performance of different strategies in symmetric dispersion games. Results reprinted from (Grenager et al., 2002), with permission.

Figure 2 shows the results for the WLU for increasing number of agents and corresponding number of tasks ($n = k$). The reward for executing a task by an agent is 1. The convergence results improve on the used reinforcement learning algorithm of (Grenager et al., 2002) and are competitive with the BS en ES strategies, with however a much more local signal as tasks that still need to be resolved are not communicated to the agents and an agent will have to explore for its "own" task. The RL signal for agent η is purely based upon how many agents $\hat{\eta}$ choose the same task. Agents using the SU

quickly reach a maximum fitness of ≈ 0.8 (figure 3a). The agents using the SU however have difficulty in targeting the last 20% of the tasks as they continue to compete for tasks. The TG utility (figure 3b) performs even worse as a maximum utility of 0.7 is reached for 10 agents and a utility of ≈ 0.65 for a larger number of agents as the signal-to-noise ratio decreases. In contrast, the penalties imposed by the WLU successfully drive agents to efficiently disperse.

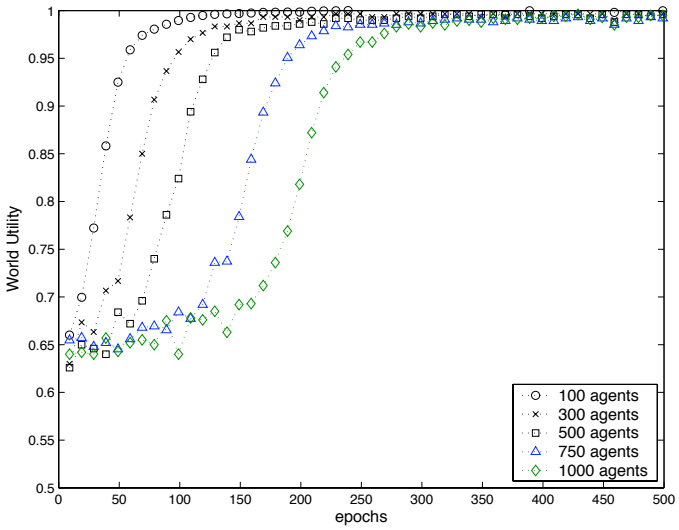

Fig. 2. WLU for dispersion games

3.1 Dispersal WLU

From figure 2, we can see that using the standard COIN framework, as the number of agents increases, the point at which individual agents choose a task is delayed. Agents compete for tasks for longer periods in their early exploratory behavior and the penalties incurred cannot yet push unsuccessful agents to unfulfilled tasks, while this incentive for correct dispersion is necessary for the system to quickly converge. This phenomenon partially explains the trend in slower convergence of the WLU for an increasing number of agents in Figure 2. To improve on convergence of the COIN framework, we investigate application of the enhanced WLUm and WLUr utility functions.

In Figure 3.1 we show typical results for the new utility functions, in this case for 2500 agents.[2] The $WLUr$ and $WLUm$ converge dramatically faster than the classic WLU, even for a large number of agents. The $WLUm$

[2] We did not explore settings with more agents due to memory restrictions with the current implementation.

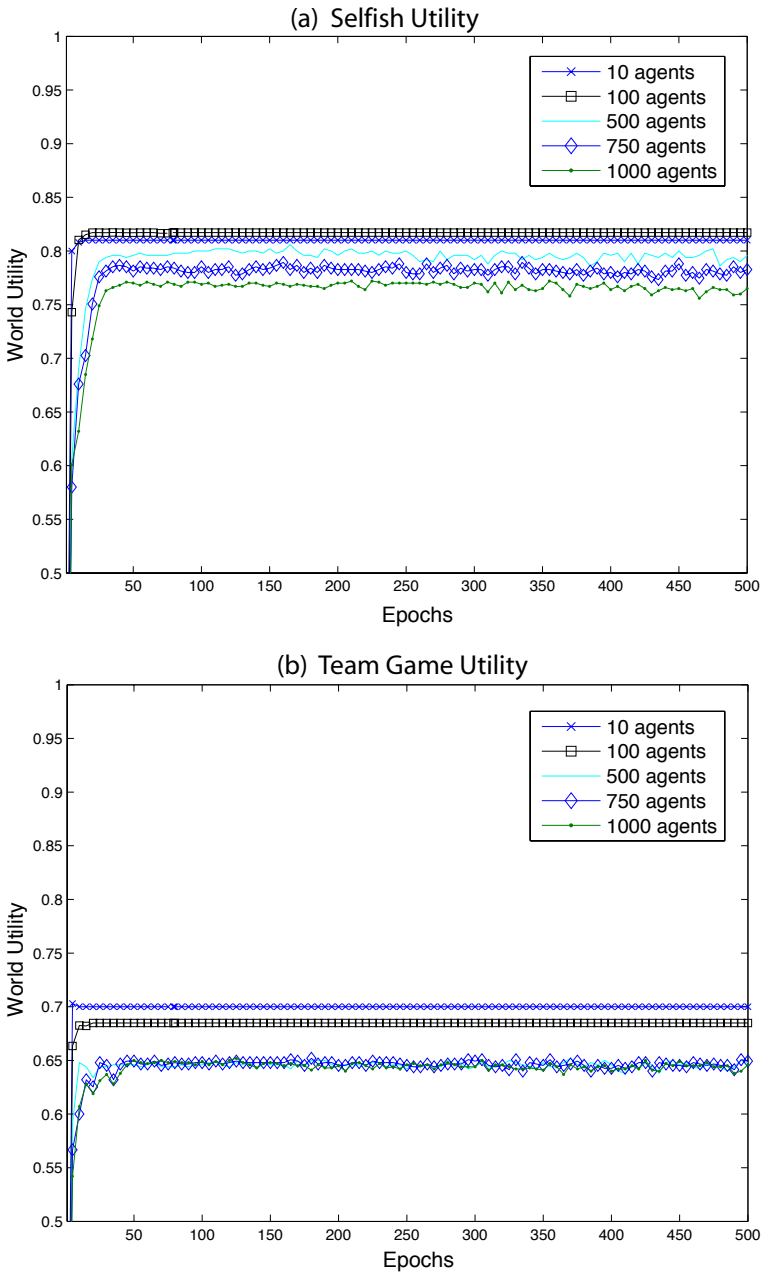

Fig. 3. Dispersion games: (a) agents using Selfish Utility utility (b) agents using Team Game.

outperforms the $WLUr$ similarly in all experiments for the range of agents studied in Figure 2. Agents using the WLUm can most quickly converge to a task and drive other agents to choose another task. Note that the adaptions of the WLU's still only involve local use of information per task in the problem domain and no global information is used while the WLUr and WLUm are competitive with the ES and BS strategies of (Grenager et al., 2002).

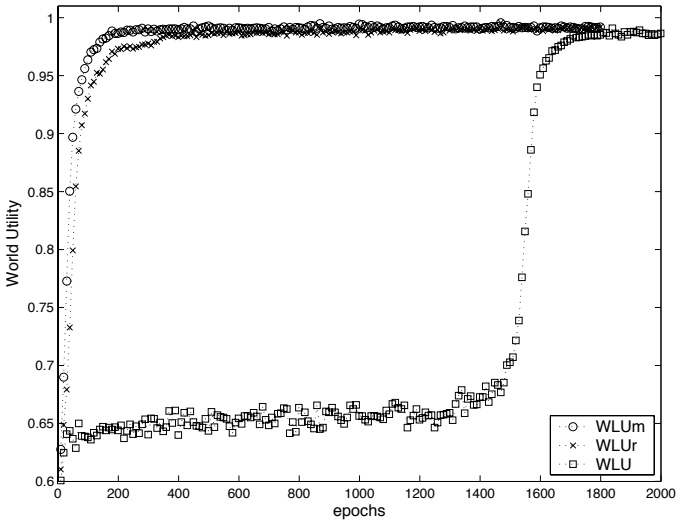

Fig. 4. Improved convergence for the full dispersion games for n=k=2500.

We investigate the influence of the adaption for the WLUm to the SU, which we name SUm. Like for the $WLUm$, the agent **m**ost likely to choose a task is given the reward. Penalties to contenders for the same task are however not given. In Figure 3.1 we show typical results for a 100 agents (similar results held for 10, 1000, 1500, and 2500 agents). The performance of the SUm is inbetween that of the WLUr and WLUm while all learning methods converge in the limit to optimal results. In Section 3.2 we show that this property does however not hold for the SUm in the more difficult task choice problems. As we will show, COIN-like utilities are needed to solve these type of problems.

In the above experiments we found best convergence results for all RL algorithms while using a large learning rate α for the individual Q learners. Increasing α from 0.1 to 1 with increments of 0.1 led to continuous increased performance as agents then most quickly choose an individual task to execute. Such large values of α are less suitable for applications where the agent receives reward from a sequence of actions (e.g. (Sutton & Barto, 1998),(Tumer, Agogino, & Wolpert, 2002)). For dispersal games, which have no such sequential property, this is not an issue.

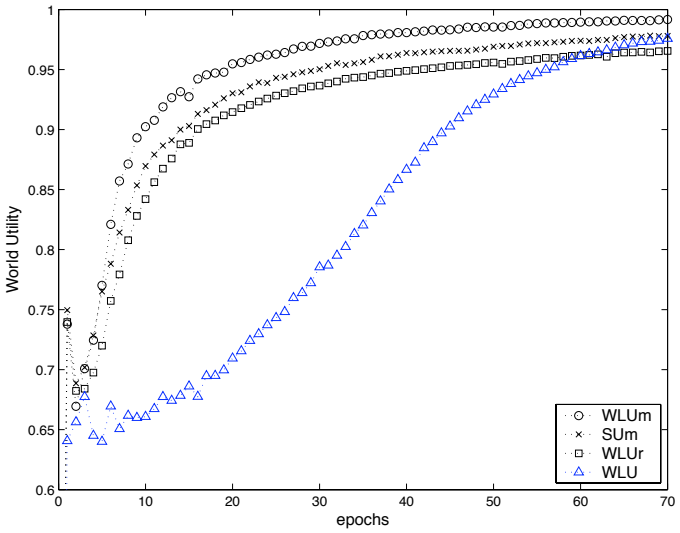

Fig. 5. Performance of SUm for $n = k = 100$, as compared to WLU(m/r).

3.2 El Farol Dispersion

In the original El Farol Bar problem (Arthur, 1994), agents have to decide on what day week they will visit one of a given set of bars. Good solutions can be hard to reach in a distributed setting as agents oscillate in their choice of attendance:

> No one goes there nowadays, it's too crowded. (Yogi Berra)

.

Inspired by this problem, we here devise a dispersion problem that is harder than the previously discussed example. In our "El Farol Dispersion game", we have n agents that have to choose between 7 tasks that each give a reward of 1 to the first $n/7$ agents that choose the task. Reward for attendance is however only given if at least $n/7$ agents choose a task. Compared to the previously studied dispersion game, this game is harder in that the agents have a less gradual, more discontinuous reward signal to learn from. In terms of dispersion games, we study $k = 7$ tasks that require 7^{c-1} agents to fulfill for a total of $n = 7^c$ agents, for some constant $c \geq 1$.

Figure 6 shows the results for the various learning algorithms for 49 agents ($c = 2$). Each bar is interpreted as a task that requires 7 agents for a total reward of 7, or 1 per agent helping to accomplish the "task". The SU and TG perform badly as both cannot locally interpret the RL signal to optimize their actions. The original WLU performs well, though it takes considerable time to converge. The WLUm converges significantly faster than the WLU, at the same performance; the WLUr also converges significantly faster, but

seems to achieve a lower level of world utility as compared to the WLUm. We remark that this is in fact an artifact of scale, as the WLUr also achieves full world utility, but only very slowly, after $> 20,000$ epochs. The SUm, which showed comparable performance for the previous dispersion tasks, only slightly improves on the admittedly poor performance of the original SU. It converges to a maximum utility of 0.8 after $30,000$ epochs. The issuing of penalties as defined for the WLU's seems fundamental for convergence to a high world utility.

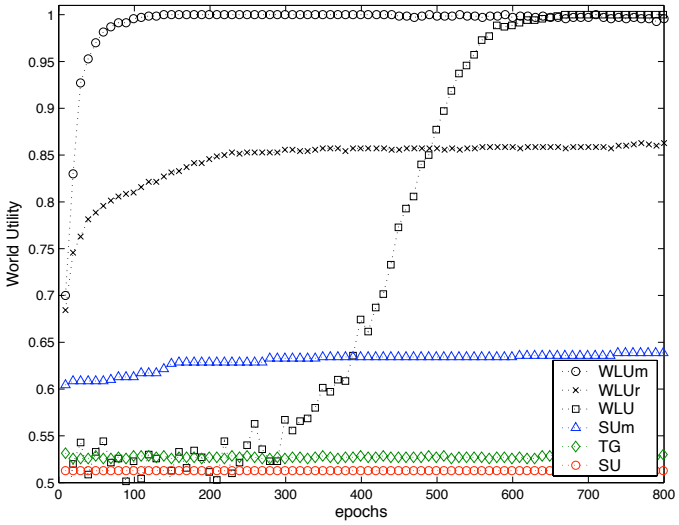

Fig. 6. Bar attendance for 49 (7^2) agents

In Figure 7, we show results for 343 (7^3) agents. The WLUr and WLUm as exceptions are both able to achieve good results. We however only achieved these best convergence results for all RL utilities by changing the used learning rate α to an unconventional high level of 10. Both optimizations resulted in stronger convergence by forcing an agent to choose a task. The original WLU, however did not improve beyond its shown level even after $150,000$ epochs. The WLUr for this problem shows surprising results in performance in that as in the smaller setting, it first seems to converge quickly to a lower performance level than the WLUm. After many many epochs though, the WLUr manages to converge to maximal world utility.

As the WLUr exhibits more exploration as compared to the WLUm, just by the mere fact that the winning agent is selected randomly, this successful but slow learning demonstrates the difficulty of the setting. It also demonstrates the trade-off between quicker learning, as the WLUm does, compared to more exploration, as per the WLUr utility.

For this interpretation of the El Farol dispersion problem with such a large number of agents we are reaching the limit of the straightforward application of the WLU and even of the proposed enhancements and we had to resort to modifications in the parameters of the learning algorithm We are hence reaching a point where we are moving beyond straightforward application of the COIN framework as an engineering approach. This problem hence merits further study to arrive at more fundamental solutions and insights.

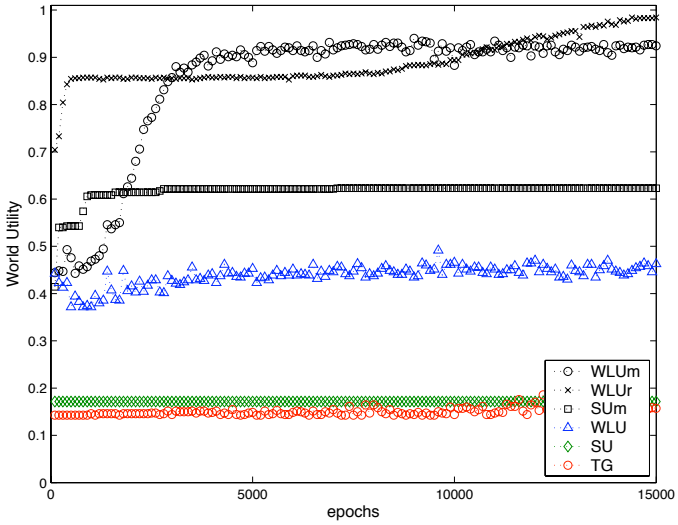

Fig. 7. El Farol dispersion game world utility for 343 (7^3) agents

4 Discussion

We studied the COllective INtelligence (COIN) framework of Wolpert et al. for a standard full dispersion game (Grenager et al., 2002) and a harder dispersion problem based on the EL Farol Bar problem. The essence of dispersion problems is that agents have to learn to choose which individual tasks to execute. We observed that for complex problems the COIN framework is able to solve fairly difficult MAS problems where fine-grained coordination between the agents is required, in contrast to multi-agent systems that use more common decentralized coordination.

We enhanced the COIN framework to dramatically improve both convergence speed and performance for difficult dispersion problems by taking advantage of the symmetry in both the standard WLU in COIN, and the corresponding agent and action symmetry in the dispersion games we consider. The increased convergence properties for the dispersion games are competitive

with especially tailored strategies for solving these task assignment problems. The generic enhancements to the COIN framework proved to be essential to solve the more complex variant of the El Farol Bar-like problem using the COIN framework.

We believe the dispersion games of (Grenager et al., 2002) form an important testbed for learning methods applied to Multi-Agent Systems (MASs), with application for many natural and important problem domains. The task assignment for n agents to k tasks is straightforward to implement, yet can quickly become difficult for distributed approaches due to parallel, asynchronous learning by the agents and the lack of global information. A fundamental question for learning methods is at what point to they begin to fail as the problems are scaled (increasing n or more difficult tasks). Can this point be delayed by increasing communication between the agents and at what cost? MAS learning is a growing research area. Dispersion games can form an interesting benchmark problem to research the limits and possibilities of this new field.

As future work we consider boot-strapping techniques for single agent RL to the COIN framework. RL in general can significantly benefit from directed exploration ((Mitchell, 1997; Thrun, 1992) and (Wiering, 1999)). Sub-goal detection as in (Menache, Mannor, & Shimkin, 2002) can also greatly speed up the learning of complex tasks. For example, in (Menache et al., 2002) an agent learns to focus in learning on critical points in the task which form bottlenecks for good overall performance. An open question is how the above work can be integrated in the (extended) COIN Framework for task with bottlenecks occurring due to dynamic interactions in a MAS.

Acknowledgement

This work has been carried out under theme SEN4 "Evolutionary Systems and Applied Algorithmics". Part of this research has been performed within the framework of the project "Distributed Engine for Advanced Logistics (DEAL)" funded by the E.E.T. program in the Netherlands, and PJH was supported by a partial travel grant by the NASA-Ames Research Center Moffett Field, California. Work of SMB is supported by NWO VENI grant 639.021.203. We thank Stefan Blom for letting us use the STW cluster at CWI.

References

Agogino, A., & Ghosh, J. (2002). Increasing pagerank through reinforcement learning. In *Proceedings of intelligent engineering systems through artificial neural networks* (Vol. 12, pp. 27–32).

Alpern, S. (2001). *Spatial dispersion as a dynamic coordination problem.* (Tech. Rep.). London School of Economics.

Arthur, B. (1994). Inductive reasoning and bounded rationality. *American Economic Association Papers*, *84*, 406–411.

Azar, Y., Broder, A., Karlin, A., & Upfal, E. (2000). Balanced allocations. *SIAM Journal on Computing*, *29*(1), 180–200.

Barto, A. G., & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems*, *13*(4), 341–379.

Challet, D., & Zhang, Y. (n.d.). Emergence of cooperation and organization in an evolutionary game. *Physica A*, *246*(3–4), 407–418.

Grenager, T., Powers, R., & Shoham, Y. (2002). Dispersion games: general definitions and some specific learning results. In *Eighteenth national conference on artificial intelligence* (pp. 398–403). Menlo Park, CA, USA: American Association for Artificial Intelligence.

Guestrin, C., Lagoudakis, M. G., & Parr, R. (2002). Coordinated reinforcement learning. In *Icml '02: Proceedings of the nineteenth international conference on machine learning* (pp. 227–234). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Hardin, G. (1968). The tragedy of the commons. *Science*, *162*, 1243-1248.

Hoen, P. J., & Bohte, S. M. (2003). Collective intelligence with sequences of actions - coordinating actions in multi-agent systems. In N. Lavrac, D. Gamberger, L. Todorovski, & H. Blockeel (Eds.), *Proceedings of the 14th european conference on machine learning, ecml'03* (Vol. 2837, p. 181-192). Springer.

Huang, C.-F., Bieniawski, S., Wolpert, D. H., & Strauss, C. E. M. (2005). A comparative study of probability collectives based multi-agent systems and genetic algorithms. In H.-G. Beyer & U.-M. O'Reilly (Eds.), *Proceedings of the genetic and evolutionary computation conference, gecco 2005* (p. 751-752). ACM.

Lauer, M., & Riedmiller, M. (2000). An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proc. 17th international conf. on machine learning* (pp. 535–542). Morgan Kaufmann, San Francisco, CA.

Lee, C. F., & Wolpert, D. H. (2004). Product distribution theory for control of multi-agent systems. In *Proceedings of the 3rd international joint conference on autonomous agents and multiagent systems (aamas 2004)* (p. 522-529). IEEE Computer Society.

Menache shai, Mannor, S., & Shimkin, N. (2002). Q-cut - dynamic discovery of sub-goals in Reinforcement Learning. In T. Elomaa, H. Mannila, & H. Toivonen (Eds.), *Machine learning: Ecml 2002, 13th european conference on machine learning* (Vol. 2430, p. 295-306). Springer.

Mitchell, T. (1997). *Machine learning.* McGraw-Hill.

Panait, L., & Luke, S. (2005). Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, *11*(3), 387–434.

Sutton, R., & Barto, A. (1998). *Reinforcement learning: An introduction*. Cambridge, MA: MIT-press.

Thrun, S. B. (1992). *Efficient exploration in reinforcement learning* (Tech. Rep. No. CMU-CS-92-102). Pittsburgh, Pennsylvania: Carnegie Mellon University.

Tumer, K., Agogino, A., & Wolpert, D. (2002, July). Learning sequences of actions in collectives of automomous agents. In *Proceedings of the autonomous agents and multi agents systems conference* (pp. 378–385). Bologna, Italy.

Tumer, K., & Agogino, A. K. (2007). Distributed agent-based air traffic flow management. In E. H. Durfee, M. Yokoo, M. N. Huhns, & O. Shehory (Eds.), *Proceedings of the 6th international joint conference on autonomous agents and multiagent systems (aamas 2007)* (p. 255). IFAAMAS.

Tumer, K., & Wolpert, D. (2000). Collective intelligence and braess' paradox. In *Aaai/iaai* (p. 104-109). AAAI Press / The MIT Press.

Wellman, M. P. (1996a). The economic approach to artificial intelligence. *ACM Computing Surveys*, *28*(4es), 14–15.

Wellman, M. P. (1996b). Market-oriented programming: Some early lessons. In S. Clearwater (Ed.), *Market-based control: A paradigm for distributed resource allocation*. River Edge, New Jersey: World Scientific.

Wiering, M. (1999). *Explorations in efficient reinforcement learning*. Unpublished doctoral dissertation, University of Amsterdam.

Wolpert, D., & Bieniawski, S. (2004). Distributed control by lagrangian steepest descent. *CoRR*, *cs.MA/0403012*.

Wolpert, D., Strauss, C., & Rajnayaran, D. (2006). Advances in distributed optimization using probability collectives. *Advances in Complex Systems*, *9*(4), 383–436.

Wolpert, D., & Tumer, K. (1999). *An introduction to COllective INtelligence* (Tech. Rep. No. NASA-ARC-IC-99-63). NASA Ames Research Center. (A shorter version of this paper is to appear in: Jeffrey M. Bradshaw, editor, Handbook of Agent Technology, AAAI Press/MIT Press, 1999)

Wolpert, D., & Tumer, K. (2001). Optimal payoff functions for members of collectives. *Advances in Complex Systems*.

Wolpert, D., Turner, K., & Frank, J. (1998). Using collective intelligence to route internet traffic. In M. J. Kearns, S. A. Solla, & D. A. Cohn (Eds.), *Advances in neural information processing systems 11* (p. 952-960). The MIT Press.

Wolpert, D. H., Wheeler, K. R., & Tumer, K. (1999, May 1–5). General principles of learning-based multi-agent systems. In O. Etzioni, J. P. Müller, & J. M. Bradshaw (Eds.), *Proceedings of the third annual conference on autonomous agents (AGENTS-99)* (pp. 77–83). New York: ACM Press.