Abstract: This report draws on previous work by Wolpert et al. which applies the COllective INtelligence (COIN) framework to Multi-Agent Systems with the aim to effectively reward agents for their actions so that the system as a collective is optimised. A novel extension to the problems studied so far is presented whereby agents can have differing preferences. Two case studies are used to investigate the performance of various payoff utility functions: The El Farol Bar Problem and a multi-agent version of the Grid World Problem. In particular, the Wonderful Life Utility versus a Team-Game approach is investigated and tested on increasingly complex and diverse problem instances so as to address the question of scalability.
Acknowledgements

Thanks to Dr. Gillian Hayes for her fantastic supervision and Jay Bradley for moral support when debugging my code.
# Contents

1 Introduction 1

2 COIN Framework 3

3 El Farol Bar Problem 7
   3.1 Description ........................................... 7
   3.2 Implementation ........................................... 9
   3.3 Experimental Methodology ................................. 13
      3.3.1 General ........................................... 13
      3.3.2 Problem Instances ................................... 13
      3.3.3 Temperature Survey ..................................... 14
   3.4 Results ................................................ 15
      3.4.1 Temperature Survey ................................... 15
      3.4.2 Measure of performance .................................. 17
      3.4.3 Differing preferences ................................... 20
      3.4.4 Extent of differing preferences .......................... 21
      3.4.5 Different proportions of preferences ........................ 23
      3.4.6 Overview ........................................... 24

4 Multiple Autonomous Agents in a Grid World 25
   4.1 Description ........................................... 25
   4.2 Implementation ........................................... 27
   4.3 Experimental Methodology ................................... 30
      4.3.1 General ........................................... 30
      4.3.2 Problem Instances ................................... 30
      4.3.3 ε Survey ........................................... 33
   4.4 Results ................................................ 34
      4.4.1 ε Survey ........................................... 34
      4.4.2 Greater COllective INtelligence .......................... 35
      4.4.3 Split types of rewards ................................... 36
      4.4.4 Different proportions of types of agents ...................... 38
      4.4.5 Extent of Specialisation ................................. 39
      4.4.6 Hidden bonuses ....................................... 40
      4.4.7 Overview ........................................... 41

5 Conclusion 43

Bibliography 45
1. Introduction

This report draws on work by Wolpert and Tumer [5, 7, 4] which deals with the problem of how to best design a decentralised multi-agent system (MAS). Each of the agents in the system learns from their past actions through Reinforcement Learning (RL) and the main problem is how best to reward each of the agents to ensure that the utility of the entire world is maximised. The work presented here adds another dimension to the surveys of methods conducted by Wolpert et al. [5]: It seeks to investigate what effect having agents with differing preferences has on the system in terms of performance of the system, measured by its utility.

Through creating the Collective Intelligence (COIN) framework, Wolpert and Tumer derived a payoff function which proved to be far more effective than conventional approaches such as a ‘team game’ - the Wonderful Life Utility (WLU). Their recent research focusses on two particular problems viewed as multi-agent systems. The first is a famous congestion problem, the El Farol Bar Problem [5, 7], and the other deals with agents each learning an optimal sequence of actions in a grid world[6, 3]. Both of these problems are dealt with as case studies in this paper and their implementations are extended to incorporate the notion of differing preferences between agents.

Motivation for extending work by Wolpert et al. comes from investigating the effectiveness of the Wonderful Life Utility in more complex situations than those previously studied.
1. INTRODUCTION
2. COIN Framework

This chapter aims to briefly describe the COIN framework created by Wolpert and Tumer. Only necessary mathematical detail is discussed but further information can be found in [4]. The aim of the COIN framework is to address the problem of how best to design a ‘COllective INtelligence’ (COIN). A COIN is defined as a large MAS where “there is little or no centralized, personalized communication and/or control” [4]. In addition, the goal of the system is to maximise the world utility function which is based on the actions of all the members of the system as a collective. Due to the lack of central control or communication the agents are essentially lone reinforcement learners, each striving to maximise their own private payoff utility functions. Despite this individuality in agents and associated greed in an effective COIN, one must ensure that the agents do not work at cross-purposes. It may be that individual greed effectively reduces the world utility (Tragedy of the Commons) or that an action which benefits the whole system when applied by a certain subset of agents becomes detrimental when applied by all the agents (Liquidity Trap). Solutions avoiding these phenomena and other adverse conditions of the system involve altering the payoff utility functions for each of the agents. Therefore, what needs to be addressed in the design of a COIN is the inverse problem of how exactly to both initialise and update the payoff utility function of each of the agents so that the entire system of agents achieves an optimal value of the world utility.

It is clear that a requirement of any effective payoff utility function must ensure that an action which improves an agent’s payoff utility also improves the world utility. When the function satisfies this condition it is said to be aligned\(^1\). Much of the past work dealing with RL and MAS has made the payoff utility function for each agent equal to the world utility function, thereby implementing a team game. Although a team-game utility is an example of an aligned function, it suffers from a low signal-to-noise ratio which results in an agent not being able to learn clearly what impact making an action has on the system. Intuitively, the more agents there are in a system, the more a team-game reward signal for a particular agent would be affected by the fact that the reward given would be influenced (made more noisy) by other agents’ actions. In addition the payoff utility function should have a high learnability\(^2\) in that an RL algorithm can effectively learn to optimise it.

Through using the COIN framework Wolpert et al. derived such an individual

\(^1\)In some literature the term factored is used interchangeably with aligned
\(^2\)Learnability is a rather complex mathematical characteristic related to the intelligence of the COIN system. Please refer to [4] for further detail.
2. COIN FRAMEWORK

payoff utility function that is both learnable and aligned: the Wonderful Life Utility (WLU). This was proven to be a vast improvement on a team-game payoff utility in that the system has the potential to converge to optimal solutions far quicker. The WLU addresses the signal-to-noise ratio problem by giving each agent an insight into what the system would have been like had they not existed or if they had chosen a different action. In effect, each agent gains a better understanding of the impact of their actions on the system and only receives positive reinforcement (reward) if their actions have a positive effect on the system.

Let us now define formally what a COIN is in as much detail as is necessary to explain the WLU further. Let $\zeta$ be the joint actions of all the agents in a system and let $G(\zeta)$ be the function which calculates the world utility of the entire system. The aim of the system is to therefore find a $\zeta$ such that it maximises $G(\zeta)$. Let $\eta$ denote each individual agent of the system and $g_\eta$ be agent $\eta$’s payoff utility function. Each agent will therefore be learning its optimal action to perform, $\zeta_\eta$, by getting rewarded according to its payoff utility function $g_\eta$.

The WLU is an example of a difference utility, which have the form:

$$U(\zeta) = G(\zeta) - \Gamma(f(\zeta)).$$  \hfill (2.1)$$

$U(\zeta)$ represents an individual payoff utility function for an agent and defines the reward to be given. As the payoff utility function is related to $G(\zeta)$, the necessary alignment to the world utility is achieved. In addition, because $\Gamma(f)$ is independent of $\zeta_\eta$ there may be some information contained in the reward signal about what a different world would have been like. This different world, $f(\zeta)$, may be one in which only a subset of the other agents were present (not $\eta$), or where $\eta$ carried out different actions to those actually carried out in $\zeta$. By imagining a fictional world each agent is able to gain a greater insight into how the system as a whole is working and more specifically what impact its actions have on the world.

To formalise the fictional worlds supposed, the WLU is parameterised by a prefixed clamping parameter $CL_\eta$ defined as follows:

$$WLU_\eta \equiv G(\zeta) - G(\zeta_\eta, CL_\eta).$$  \hfill (2.2)$$

Where $\hat{\eta}$ is all agents in the system other than $\eta$.

The clamping parameters used for the studies in this paper match those investigated by Wolpert et al. [5, 6] and create three derivative WLUs, each considering a different fictional world:

1. $CL_\eta = 0$
2. $CL_{\eta} = 1$

3. $CL_{\eta} = Average$

The derivative WLUs are differentiated by a superscript representing the clamping parameters: $WL{\overrightarrow{\Omega}}$, $WL{\overrightarrow{I}}$, and $WL{\overrightarrow{\sigma}}$.

The first derivative considers a world in which agent $\eta$ does not exist and lends itself to the naming of the Wonderful Life Utility after the Frank Capra film where the main character is shown what the world would have been like had he not existed. The utility of such a world, is thus defined by $G(\zeta_{\eta}, \overrightarrow{\Omega})$ and is equivalent to setting all of $\eta$’s actions to null.

Imagine a single stage of a multi-agent system consisting of four agents where each agent has the opportunity to make three different actions.\(^3\) Let the following $4 \times 3$ matrix represent the joint actions of all the agents in a system ($\zeta$) where the rows represent the action profile for each agent and the columns represent the different actions. A 1 at $(m,n)$ represents action $n$ being chosen by agent $\eta_m$.

$$
\zeta = \\
\begin{bmatrix}
\eta_1 & 1 & 0 & 0 \\
\eta_2 & 0 & 0 & 1 \\
\eta_3 & 1 & 0 & 0 \\
\eta_4 & 0 & 1 & 0
\end{bmatrix}
$$

Let us suppose that we are in the process of rewarding $\eta_2$ for its actions. $\eta_2$ therefore considers a world without it by applying the clamping parameter $CL_{\eta_2} = 0$ to the world:

$$(\zeta_{\eta_2}, \overrightarrow{\Omega}) =$$

$$
\Rightarrow \\
\begin{bmatrix}
\eta_1 & 1 & 0 & 0 \\
\eta_2 & 0 & 0 & 0 \\
\eta_3 & 1 & 0 & 0 \\
\eta_4 & 0 & 1 & 0
\end{bmatrix}
$$

By evaluating the function $WL{\overrightarrow{\Omega}}_{\eta_2} \equiv G(\zeta) - G(\zeta_{\eta_2}, \overrightarrow{\Omega})$, agent $\eta_2$ gains the reward of the difference that it made to the whole system (world). This may be negative if the agent’s actions had an adverse effect on the system as a whole.

The second derivative means that instead of clamping agent $\eta$’s actions to null, the fictional world considered is one in which $\eta$ makes all possible actions. If the

\(^3\)This example is identical to that used by Wolpert et al. in [5, 6]
system described above, consisting of four agents, limits the amount of actions that any one agent can choose to one then considering a world where \( \eta \) chooses all of the actions simultaneously may seem wrong. The WLU in this form does indeed consider illegal worlds. This may be beneficial, however, as it allows the agent some insight into what the world would have been like had it chosen alternative actions.

\[ CL_{\eta_2} = 1 \]

\[
\begin{bmatrix}
\eta_1 \\
\eta_2 \\
\eta_3 \\
\eta_4 \\
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & 0 \\
1 & 1 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
\end{bmatrix}
\]

The third case considers a world where agent \( \eta \) spreads itself out over all of the possible actions it can choose from assigning an equal ‘amount’ of itself to each action. This is of course an impossibility but as long as the world utility function is continuous then it should be possible for a utility to be calculated for such a case. This third case is called the Average clamping parameter as it replaces \( \eta \)’s action vector with the average action vector. In the on-going example, \( \eta_2 \)’s action vector is originally \( \vec{a} = \{0, 0, 1\} \) which, when clamped to the average makes \( \vec{a} = \{.33, .33, .33\} \).

\[ CL_{\eta_2} = \text{Average} \]

\[
\begin{bmatrix}
\eta_1 \\
\eta_2 \\
\eta_3 \\
\eta_4 \\
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
\end{bmatrix}
= 
\begin{bmatrix}
1 & 0 & 0 \\
.33 & .33 & .33 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
\end{bmatrix}
\]

Research by Wolpert et al. [5] has shown that most of the time the average clamping parameter derivative of the WLU yields better results than the other two derivatives. Nevertheless, all three types are investigated in the first case study of this report.
3. El Farol Bar Problem

3.1 Description

The bar problem is made up of a single-move game whereby each agent in a MAS chooses a single night of the week to attend a bar.¹ Each agent therefore has one of seven actions to choose from and learns which action is best using reinforcement learning. How good a particular action is depends on the attendance of all the agents in the system for that night. Through successive plays of the game, the agents are able to learn iteratively which actions yield the highest reward for them choosing between the seven actions using a soft-policy.

A preference function is used to calculate how good particular action is for the agent and is such that there is a peak on a certain attendance, marking the agent’s optimal preference. The action resulting in attending a day with too few or too many attendees is therefore considered less good. An agent is happier the closer it gets to maximising its preference function.

The world utility is defined as the overall happiness of the system measured by the sum of the cumulative happiness of the agents attending each night of the week. The problem is how to reward each of the agents for their actions so that they learn the best actions to perform ensuring that the world utility is optimised. We shall see that team-game payoff utility approaches, which assign a reward equal to the happiness of the whole world, result in poor performance whereas the Wonderful Life Utility proves an elegant solution to the problem of reward assignment.

Wolpert et al. dealt with cases of the El Farol Bar Problem in which there was clearly a congestion problem: the number of agents greatly exceeds the sum of optimal preferences for each day. In [5] Wolpert and Tumer deal with a single preference function in which the optimal attendance for each night is three. The problems they dealt with consist of 60 agents which is far greater than optimal attendance times the number of days (3 × 7). In this study’s extension of the bar problem, some agents may prefer busier nights than others. This is accomplished by having different preference functions for different subsets of agents.

Clusters of agents are defined as those which share a common preference function. For example, one cluster of the agents may prefer an attendance of three on each night, whilst another cluster may prefer nights with an attendance of twelve.

¹In fact in Wolpert and Tumer’s paper they consider the slightly more complex variation whereby each agent can choose any of ℓ nights where ℓ is in {1,2,3,4,5,6}.
3. EL FAROL BAR PROBLEM

Generally the population of agents is split evenly into clusters but further cases where a certain cluster may be proportionally more dominant are investigated. The number of clusters can range from 1 to $n$ where $n$ is the number of agents in the system. A single cluster represents a system of agents with no differing preferences. A system with $n$ clusters means that every agent has a different preference from every other agent. This study sets out to prove that despite agents having different aims in the system creating an increase in frustration, an optimal world utility is achievable and that the Wonderful Life Utility is an effective payoff utility vastly improving on a team game.

Formally, the world utility for any particular time step (week) is:

$$G(\zeta) \equiv \sum_{j=1}^{\text{no. clusters}} \sum_{k=1}^{7} \phi_j(x_k(\zeta_j))$$

(3.1)

where $\zeta_j$ is the joint moves of all the agents in cluster $j$; $x_k(\zeta_j)$ is the total attendance of agents belonging to cluster $j$ on night $k$; and $\phi_j$ is the preference function for cluster $j$ parameterised by a real value $c_j$:

$$\phi_j(y) \equiv \frac{c_{\text{max}}}{c_j} \times y \times e^{(-y/c_j)}$$

(3.2)

where $c_{\text{max}}$ is the maximum $c$ parameter in the whole system of agents and $c_j$ is the $c$ parameter for the cluster $j$. The fraction acts as a normalisation so that despite different curves being produced by changing $c$, the value of the optimal remains constant.²

Let us formalise the example mentioned above, with a system of two clusters of agents: one cluster preferring an attendance of three and other other preferring an attendance of twelve. All agents belonging to the first cluster would hence share a common preference function with a $c$ value of three and those belonging to the latter cluster would have a similar function but with the $c$ value equal to twelve. $c_{\text{max}}$ is equal to twelve and therefore the preference functions for the two clusters would be as follows:

$$\phi_1(y) \equiv \frac{12}{3} \times y \times e^{(-y/3)}$$

$$\phi_2(y) \equiv \frac{12}{12} \times y \times e^{(-y/12)}$$

These two functions are plotted in Figure 3.1 below.

²A better method of normalisation might be to ensure that both the optimal and the area under the graph is kept the same for all values of $c$. 
3.2. Implementation

The Bar Problem was initialised with a number of agents, each belonging to a particular cluster with an associated preference function. For all of the experiments conducted the single-move game was repeated 500 times, each one consisting of every agent choosing a night of the week to attend. This was implemented serially in that the agents make this choice sequentially, however, the order in which agents attended was randomised at the beginning of each time step.

A single time step of the Bar Problem was implemented algorithmically as follows:

1. choose an agent that has not yet attended the bar randomly
2. choose a night to attend the bar: randomly for the first \( r \) runs and afterwards based on its previous reward history
3. repeat until all agents have attended
4. calculate the world utility
5. assign rewards to each of the agents according to the payoff utility function

The second step involves, in its random stage, each of the agents selecting a night uniformly randomly. This is to allow each agent to build up a reward history at the start, exploring the action-space for a finite amount of time without exploiting knowledge gained. Once \( r \) runs have taken place, the first step enters its learning stage where reinforcement learning is used. For all the experiments in this report, \( r \) was always set to 100.

Each agent constructs an Estimated Utility Vector (EUV) for each night based
on the previous rewards it received for choosing that night (action) in the past. This is done by taking a weighted average of all of the past rewards received for each night. The weights are such that they decrease exponentially according to how far back in the history the reward data is retrieved from. This ensures that the most recently received rewards are given (a slight) priority.

Formally, for every action $a$, i.e. each of the nights of the week, the expected utility ($Q$) of choosing this action at time $t$ is given by the following equation:

$$Q_t(a) = \frac{\alpha^0 Q_{t-1}^*(a) + \alpha^1 Q_{t-2}^*(a) + \alpha^2 Q_{t-3}^*(a) + \ldots}{\alpha^0 + \alpha^1 + \alpha^2 + \ldots}$$

(3.3)

where $Q_{t-1}^*(a)$ is the actual reward received as a result of choosing action $a$ at the time $t-1$ (the last time step); and $\alpha$ is a real value less than 1. In this paper $\alpha$ is kept constant at 0.999. The EUV of every agent would therefore consist of the expected rewards of attending each of the nights:

$$\mathbf{EUV} = \begin{pmatrix}
Q_t(a_1) \\
Q_t(a_2) \\
Q_t(a_3) \\
Q_t(a_4) \\
Q_t(a_5) \\
Q_t(a_6) \\
Q_t(a_7)
\end{pmatrix}$$

(3.4)

The simple action-selection technique of Softmax [2] is utilised to choose an action based on a Boltzmann distribution over the expected utilities for the seven nights of the week. The following Softmax formula is applied the EUV to calculate the probabilities $p_t(a)$ of choosing each action $a$ based on the expected outcomes:

$$p_t(a) = \frac{e^{Q_t(a)/\tau}}{\sum_{b=1}^{n} e^{Q_t(b)/\tau}}$$

(3.5)

where $Q_t(a)$ is the estimated value of taking action $a$ (choosing one of seven nights) at time $t$; and $\tau$ is a positive parameter called the temperature. As the temperature approaches zero the selection process becomes increasingly greedy; as it approaches infinity it becomes increasingly random.

Once Softmax has calculated the probabilities of choosing each action, roulette-wheel selection [1] is performed to finally select a night based on the probabilities of being chosen.

After all the agents in the system have each attended a night of the bar for the week (chosen their actions for the particular time step), the world utility can be
3.2. IMPLEMENTATION

calculated as per Equation 3.1. For each night, the attendance of each cluster of agents is calculated and the value is fed into the corresponding preference function for the particular cluster. It is necessary to calculate the world utility at this stage as it is used to calculate the reward given to each of the agents.

Rewards given to the agents depend on the payoff utility function used. Four such payoff utility functions are investigated in this report which mirror those investigated by Wolpert et al. [5] in their work on the Bar Problem.

Firstly a Team-Game approach assigns simply the world utility as a reward to each of the agents so all the agents receive the same reward:

$$Team(\zeta) \equiv \sum_{j=1}^{\text{no. clusters}} \sum_{k=1}^{7} \phi_j(x_k(\zeta_j)).$$

(3.6)

The further three payoff utility functions investigated are the three derivatives of WLU mentioned in Chapter 2. Each agent receives a customised reward based on the world utility as well as the impact its actions had on the utility of the world. In the following equations, let $\eta$ represent the agent being rewarded.

$WLU^{\eta}$ considers what the world utility would have been had the agent $\eta$ not attended any of the nights:

$$WLU^{\eta}(\zeta) \equiv G(\zeta) - G(\zeta_{\eta}, \emptyset)$$

$$\equiv G_{\eta}^{k_{\eta}}(\zeta) - G_{\eta}^{k_{\eta}}(\zeta_{\eta}, \emptyset)$$

$$= \sum_{j=1}^{\text{no. clusters}} \phi_j(x_{k_{\eta}}(\zeta_j)) - \left( \sum_{j \neq j_{\eta}}^{\text{no. clusters}} \phi_j(x_{k_{\eta}}(\zeta_j)) + \phi_{j_{\eta}}(x_{k_{\eta}}(\zeta_{j_{\eta}}) - 1) \right),$$

(3.7)

where $k_{\eta}$ is the night chosen by $\eta; G_{k_{\eta}}$ is the world utility of the night chosen by $\eta; j_{\eta}$ is the cluster which $\eta$ is a member of; and so, $x_{k_{\eta}}(\zeta_{j_{\eta}})$ is the number of members of the cluster $j$ that $\eta$ is a member of that also chose the night $k$ that $\eta$ chose. The simplification from line 1 of Equation 3.7 to line 2 removes the effect of the nights which $\eta$ did not choose as the attendance of these nights and subsequent world utilities is independent of $\eta$’s actions. Line 3 may be simplified further by removing all clusters apart from that which the agent is a member of, however, the above version matches precisely how the algorithm was implemented namely that the first expression represents the world utility for the day that the agent chose and the second expression represents the world utility for the same day but with the attendance of the cluster that the agent is a member of decreased by one.
3. EL FAROL BAR PROBLEM

\( WLU^{\uparrow} \) considers what the world utility would have been had the agent \( \eta \) not attended any of the nights:

\[
WLU^{\uparrow}(\zeta) = G(\zeta) - G(\zeta_{\eta}, \overline{\eta}) \\
= \sum_{k \neq k_{\eta}} \left( G_k(\zeta) - G_k(\zeta_{\eta}, \overline{\eta}) \right) \\
= \sum_{k \neq k_{\eta}} \left( \frac{\text{no.clusters}}{7} \sum_{j=1}^{7} \phi_j(x_k(\zeta_j)) + \frac{1}{t} \right) + \phi_j(x_k(\zeta_{j_{\eta}}) - 1 + \frac{1}{t}).
\]

In some ways \( WLU^{\uparrow} \) is the inverse of \( WLU^{\uparrow} \) as instead of dealing with only the night that \( \eta \) attended, it deals with all the nights except that which \( \eta \) attended. It need not deal with the day that \( \eta \) chose as \( G_k(\zeta) - G_k(\zeta, \overline{\eta}) \) is always equal to zero. This is because the fictional world created by the \( k \) clamping parameter is only different from the original world for those days other than \( k_{\eta} \). The third line of Equation 3.8 consists of the original world utility minus the fictional world utility. In order to create this fictional world, \( WLU^{\uparrow} \) essentially increases the attendance of all the nights that \( \eta \) didn’t choose by one but only for the cluster for which \( \eta \) belongs to.

\( WLU^{\downarrow} \) considers what the world utility would have been had the agent \( \eta \) contributed an equal fraction of one to each of the nights for the cluster which it belongs to:

\[
WLU^{\downarrow}(\zeta) = G(\zeta) - G(\zeta_{\eta}, \overline{\eta}) \\
= \sum_{j=1}^{\text{no.clusters}} \sum_{k=1}^{7} \phi_j(x_k(\zeta_j)) - \left( \sum_{j \neq j_{\eta}} \sum_{k=1}^{7} \phi_j(x_k(\zeta_j)) \right) \\
+ \sum_{k \neq k_{\eta}} \phi_j(x_k(\zeta_{j_{\eta}}) + \frac{1}{t} + \phi_j(x_k(\zeta_{j_{\eta}}) - 1 + \frac{1}{t}).
\]

Whereas \( WLU^{\uparrow} \) and \( WLU^{\downarrow} \) could be simplified somewhat, \( WLU^{\downarrow} \) needs to deal with every agent’s move in the whole system for that time step (complete \( \zeta \)) in order to calculate the reward for agent \( \eta \). The fictional world is created as follows: for the clusters which \( \eta \) does not belong to the attendance on each night is left the same; for the cluster which \( \eta \) does belong to, the attendance on the nights not chosen by \( \eta \) are incremented by \( \frac{1}{t} \) and the attendance of the night that \( \eta \) did choose is first decremented by one (to simulate the removal of \( \eta \)) and then incremented by \( \frac{1}{t} \).
3.3 Experimental Methodology

3.3.1 General

In each experiment the number of agents in the system is such that the congestion present is kept constant. As mentioned in Section 3.1, the subset of system setups which are interesting are those in which the number of agents exceeds the optimal attendance on each of the seven nights by many (four) times (and hence the agents are congested). Wolpert et al’s paper [7] suggests using this method to keep the congestion levels constant throughout problem instances and this is extended here to incorporate the notion of clusters and different optimal attendances for agents.

\[
No.Aagents = \sum_{j=1}^{No.Clusters} \frac{4 \times c_j \times 7}{No.Clusters}
\] (3.10)

Three features of the Bar Problem system with differing preferences were investigated:

1. what effect splitting the population of agents into clusters had on the convergence of the system to an optimal world utility;

2. if the degree to which the clusters differed in preference had any effect;

3. if changing the proportions of the population in each cluster had any effect.

Various instances of the Bar Problem were setup and run to address the features above. The ones yielding the most interesting results are discussed in the subsequent sections of this chapter. Each instance consists of 500 plays of the Bar Problem single-move game and the world utility at each of these plays is averaged over 20 runs.

3.3.2 Problem Instances

The features described above were investigated by setting up five distinct problem instances. The first is the same as that discussed by Wolpert et al. in [5] with a single cluster preferring an attendance of three on each night. This is used as the base problem from which further instances with multiple clusters are compared to. The second problem splits the population into two clusters with one half preferring an attendance of one on each night whilst the other half prefers an attendance of three. By comparing the results from problem 1 with problem 2, feature 1 could be observed. The third problem is similar to the second but instead involved clusters with preference values of two and twelve. By comparing
problems 2 and 3, feature 2 could be observed. To observe feature 3 two more complex instances were designed (problems 4 and 5) each consisting of three clusters with preference values 2, 4 and 12 but the proportions of the population belonging to each cluster differs between the two problems. In problem 4 the population is split evenly in three between the clusters, whereas in problem 5 a quarter of the population have preference value 2, a quarter value 4 and the remaining half of the population have preference value 12. The five problems are described in details in Table 3.1.

In summary the five problems may be described as follows:

1. Single preference
2. Concentrated preferences
3. Dispersed preferences
4. Equal proportions of differing preferences
5. Unequal proportions of differing preferences

### 3.3.3 Temperature Survey

Clearly, in an instance of the El Farol Bar Problem, there are many variables which alter the behaviour of the system. The temperature variable of the Softmax formula (Equation 3.5) has the most profound effect on the world utility over time as it determines the balance between agents selecting the greedy action (exploitation) or an action with a less than maximal expected reward (exploration). Mirroring Wolpert and Tumer’s work in [5], a survey of different temperature values for each of the payoff utility functions has been conducted. This involved running various instances of the Bar Problem with different temperature values whilst monitoring the performance of the system measured by the world utility.
3.4. RESULTS

<table>
<thead>
<tr>
<th>Payoff Utility</th>
<th>Optimal Temperature</th>
</tr>
</thead>
<tbody>
<tr>
<td>$WLU^0$</td>
<td>0.085</td>
</tr>
<tr>
<td>$WLU^1$</td>
<td>0.08</td>
</tr>
<tr>
<td>$WLU^2$</td>
<td>0.1</td>
</tr>
<tr>
<td>Team-game</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Table 3.2: Optimal temperature values for Problem 1

after 500 time steps. For each value of temperature, the simulation was run 20 times and the average world utility at each time step was used in results. Initially the temperature values investigated were: 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 0.9 but further values were later chosen to get a finer estimation of the optimal value. Different instances of the Bar Problem were used to investigate if the same temperature value could be used regardless of the problem setup. Problems 1, 2 and 4 (as in Table 3.3.2) were used for the survey.

3.4 Results

3.4.1 Temperature Survey

The temperature survey conducted on Problem 1 is equivalent to Wolpert et al.’s investigation in [5]. Graph 3.2 shows the world utility of the system after 500 time steps plotted against different values of the temperature variable. The graph is similar to that in [5] and shows peaks in world utility for each payoff utility for certain values of temperature:

When the same survey was conducted on Problem 2, the world utility for each payoff utility peaked at the same temperature values as the former problem. This is shown by comparing Graphs 3.2 and 3.3.

However, when the survey was conducted on Problem 4 a slight transposition of the peaks in world utility can be observed (Graph 3.4 implying that the optimal temperature values are slightly larger for all payoff utilities. As a higher temperature value means an increase in the frequency that the agents choose a non-greedy action, this means that, compared to the first two problems, Problem 4 benefits from an increase in exploration. This may be explained by the fact that Problem 4 consists of double the number of agents in Problem 1 and three times the number of agents in Problem 2. It is reasonable to assume that a system consisting of more agents requires each agent to explore their actions more due to the increased noise created by the other agents.
3. **EL FAROL BAR PROBLEM**

![Graph](image1)

**Figure 3.2**: Problem 1: Sensitivity of payoff utility functions to changes in temperature

![Graph](image2)

**Figure 3.3**: Problem 2: Sensitivity of payoff utility functions to changes in temperature

![Graph](image3)

**Figure 3.4**: Problem 4: Sensitivity of payoff utility functions to changes in temperature
3.4. RESULTS

Although a shift in optimal temperature values was observed, as it was for all payoff utilities, it was assumed that by keeping the temperature values as in Table 3.4.1, no bias would be created between the payoff utilities which is what was trying to be avoided. Hence, Table 3.4.1 represents the temperature values used for all problem instances discussed in the following sections.

3.4.2 Measure of performance

Before presenting the results from each problem instance, as measured by the world utility over time, let us first define what an optimal solution to the Bar Problem consists of. By observing the low-level behaviour of the learning agents we are able to work out how the system converges to an optimal solution.

In the single-cluster case presented by Wolpert et al. in [5], matching Problem 1 in this report, an optimal solution can be hand-worked without much effort. It is clear that in order to gain maximal world utility the attendance on each of the nights must be equal to the preference value, $c$. In Problem 1, $c = 3$ and the number of agents is 84 (as is a requirement Equation 3.10) thus it is clear that not all the nights can have an attendance of three. An optimal solution consists of an attendance of three agents on 6 out of the 7 nights, with the remaining agents attending the seventh night. We refer to each of the six nights to be good nights whereas the seventh night which has a high-attendance and subsequently very low happiness is a sacrifice night. Those agents which attend the seventh night are as such sacrificing their own desires for the good of other agents which ‘got to the bar on a good night first’. As convergence of the system approaches, each of the agents either wallows in success being able to be part of the optimal days where the attendance equals $c$, or accepts that they can’t do any better than sacrificing themselves to an extremely low-reward-yielding day in which the bar is extremely full; either way, both classes of agent commit themselves eventually to a particular behaviour. The crux of the problem is in getting the agents to ‘agree’ on a day of the week to sacrifice whilst abiding to the rules of the system (i.e. no communication). The structure of an optimal solution to the Bar Problem as described thus far is relevant to all single cluster instances.

In the case where there are multiple preference values (clusters) to deal with, however, the optimal attendance of the high-reward yielding nights becomes less obvious. It is clear that, as a result of Equation 3.10, six nights with relatively low attendance and one with high attendance will make up an optimal solution; yet, the exact attendance on each of the six low-attendance good nights remains unknown. The attendance on each day may or may not be equal to one of the preference values which parameterise the particular problem instance as, due to the overlapping of the function curves, an intermediate value of attendance may in fact result in a higher value of world utility. This is due to the possibility
that several sub-optimal rewards from multiple clusters may be greater than an optimal reward from a single cluster. Looking back at Figure 3.2, we see that clearly an attendance of either 3 or 12 for the six good nights represents a sub-optimal solution as the world utility is the cumulative value, summing the utility of each curve.

Let us now formalise the optimal solution of any multi-clustered instance of the bar problem. Let $\Phi$ be the cumulative function of the world preference (i.e. sum of all the individual cluster preference functions):

$$
\Phi(x(\zeta)) = \sum_{j=1}^{\text{no.\,clusters}} \phi_j(x(\zeta)).
$$

(3.11)

The optimal attendance of the good nights is therefore the closest integer value to the peak of $\Phi$.

Using Problem 4 as an example, the cumulative world utility function is made up of the functions $\phi_1(x(\zeta)) = 6x \times e^{-x/2}$, $\phi_2(x(\zeta)) = 3x \times e^{-x/4}$, $\phi_3(x(\zeta)) = x \times e^{-x/12}$ as follows:

$$
\Phi(x(\zeta)) = \phi_1(x(\zeta)) + \phi_2(x(\zeta)) + \phi_3(x(\zeta))
$$

Graph 3.5 shows the resultant curves and as the peak of the cumulative curve falls closest to 3 on the x-axis and implies that the optimal attendance for the good nights is three.

![Graph 3.5: Cumulative world preference function: the sum of the cluster-specific functions](image)

An optimal solution for Problem 4 is represented by 3.6 or any alternative permutations of attendances where 6 days have 3 agents and the remaining seventh
3.4. RESULTS

day has 150 agents.

![Optimal Attendance Profile for Problem 4]

Figure 3.6: Optimal attendance profile for Problem 4

In order to later normalise the performance of the various problem instances, the optimal world utilities possible for each instance were calculated. These are displayed in the table below. It is clear that such normalisation is necessary to compare the performance of problems between one another as the world utilities possible vary a great deal depending on both the number of clusters in the problem as well as the number of agents.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Optimal World Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( \approx 6.6 )</td>
</tr>
<tr>
<td>2</td>
<td>( \approx 10.9 )</td>
</tr>
<tr>
<td>3</td>
<td>( \approx 38.2 )</td>
</tr>
<tr>
<td>4</td>
<td>( \approx 63.7 )</td>
</tr>
<tr>
<td>5</td>
<td>( \approx 80.4 )</td>
</tr>
</tbody>
</table>

The performance of each problem instance was normalised by dividing the world utility achieved in a particular time step (on average) by the optimal world utility possible for the particular instance. The resultant performance value \( p \) is a value between 0 and 1 (inclusive) where a value 1 indicates that the optimal world utility has been achieved.

At least for the problems studied in this paper, the system rarely converges to the optimal solution. Through inspection of incomplete solutions, though, various stages in system utility refinement can be identified based on the attendance in general on each night. During the initial stage where each agent chooses a night at random, the attendance on each night is generally equal. Once learning is used, if an effective payoff utility is used to reward the agents, the system usually enters the second stage quite quickly which is distinguished by 5 of the nights having a relatively low attendance and 2 nights having a relatively high attendance. This is referred to as the dual-sacrifice night stage. The third stage is when a single sacrifice night has emerged and the majority of the agents have learnt to sacrifice themselves for the good of the collective. This stage does not represent an optimal solution, however, as the remaining six nights tend not to
have an equal nor optimal attendance. The fourth stage represents the optimal attendance profile such as that in Figure 3.6. The more complex problems studied in this study tend to prematurely converge to stages 2 and 3 as the differences in reward given to the agents choosing sub-optimal nights becomes increasingly similar to that given if they were to choose the optimal night as the system gets closer to the optimal attendance profile.

![Figure 3.7: Stages of system utility refinement. Stage 1: Uniform Random (all nights equal); Stage 2: Dual-sacrifice night; Stage 3: Single-sacrifice night, ‘good’ nights have non-equal attendance; Stage 4: Shuffling of ‘good’ nights until all equal and optimal](image)

### 3.4.3 Differing preferences

Graphs 3.8 and 3.9 represent the normalised performance of Problems 1 and 2 respectively. All graphs in the remainder of this chapter plot the normalised world utility (as described in Section 3.4.2) against time and show that for the first 100 time steps the performance for all payoff utilities is roughly equal. This is because of the initial random period in which each agent chooses a night randomly - the net result is that the world utility stays at a constant, poor level.

For both Problem 1 and 2, and indeed for all problems instances in this report, $WLU\overrightarrow{Q}$ and $WLU\overrightarrow{a}$ showed very similar performance both showing a sharp period of learning, after the initial random period, and demonstrate very similar rates slowing in learning shown by the curve (Figures 3.8 and 3.9) becoming shallower after about 110 time steps. Both $WLU\overrightarrow{P}$ and the team-game payoff utility have a performance far less than the other two payoff methods. $WLU\overrightarrow{P}$, although better than the team-game utility, shows rather unpredictable behaviour in that it’s learning curve does not consist of a steady increase but instead fluctuates around the 200 time step mark (Figure 3.8).
3.4. RESULTS

If we compare Figures 3.8 and 3.9 we see that the overall performance is considerably less in the latter case where there are multiple clusters of agents with differing preferences. $WL U^{\theta}$ and $WL U^{\alpha}$ resulted in the system reaching roughly 90% of the optimal utility on average for Problem 1, whilst this performance was reduced to about 70% for Problem 2. $WL U^{I}$ showed a 10% decreased performance when applied to Problem 2 compared to that in Problem 1. The team-game utility, however, had a similar performance for both problems. This is not to say that the team-game is in any way better though as in both cases its performance was severely sub-optimal.

Figure 3.8: Performance of Problem 1 - Single preference ($c = 3$)

Figure 3.9: Performance of Problem 2 - (Concentrated) differing preferences ($c = 1, 3$)

3.4.4 Extent of differing preferences

Figures 3.9 and 3.10 represent problems 2 and 3 respectively consisting each of two clusters with the first with a small extent of differing preferences (concentrated) and the latter a greater extent (dispersed). Once again, $WL U^{\theta}$ and $WL U^{\alpha}$
show extremely similar performance both between one another and also between the two problem instances. Each converging on average to a solution about 70% of the optimal. What is interesting is that in Problem 3 the performances of $WLU\overrightarrow{T}$ and the team-game are less than they were in Problem 2. We see in Figure 3.10 the learning curve for $WLU\overrightarrow{T}$ is not as smooth as it is in Figure 3.9 showing fluctuations around the 250 time step mark and also the end performance after 500 time steps is slightly less in the second case. The team-game shows the most profound performance difference between the two problems having on average allowed a solution 55% of optimal to be learn in Problem 2 but resulting in little or no learning for Problem 3. This demonstrates that the extent to which the clusters of agents have different preferences does affect the difficulty of the problem, at least for some payoff utility functions such as the team-game. It also proves that at least $WLU\overrightarrow{O}$ and $WLU\overrightarrow{d}$ are independent of this change and are able to perform consistently better than alternative payoff utilities studied. What is also interesting with the results from Problem 3 (Figure 3.10) is that the $WLU\overrightarrow{O}$ performed quite considerably better than $WLU\overrightarrow{d}$. This contradicts Wolpert et al.’s findings in [5] which found that $WLU\overrightarrow{O}$ always performed worse than $WLU\overrightarrow{d}$.

One would imagine that if the disparity between preferences were increased then the differences in performance between the two problem instances (concentrated and dispersed preferences) would be more extreme. Perhaps $WLU\overrightarrow{O}$ could be verified as increasingly better than $WLU\overrightarrow{d}$ as the degree of difference was increased.

Figure 3.10: Performance of Problem 3 - Dispersed differing preferences ($c = 2, 12$)
3.4. RESULTS

3.4.5 Different proportions of preferences

The graph in Figure 3.11 shows that in a more complex setup with three clusters of agents with different preferences (Problem 4), $WLU^T$ and the team-game payoff utilities are severely sub-optimal. The performance of both of these, however, is greater for Problem 5 where more of the agents in the population belong to one of the clusters than the other two. Figure 3.12 shows a slightly deeper learning curve for $WLU^T$ and team-game compared to those in Figure 3.11. Both $WLU^T$ and $WLU^d$ demonstrate similar learning rates for both Problem 4 and 5 but their performance reached is higher for Problem 5. This demonstrates that a system in which there is a dominant cluster of preference presents an easier problem to solve than if the preference profile is spread uniformly throughout the population of agents.

![Equal Proportions of Differing Preferences](image1)

Figure 3.11: Performance of Problem 4 - Equal proportions of differing preferences ($c = 2, 4, 12$)

![Unequal proportions of differing preferences](image2)

Figure 3.12: Performance of Problem 5 - Unequal proportions of differing preferences: ($c = 2 [25\%), 4 [25\%], 12 [50\%]$)
3.4.6 Overview

All of the above results are consistent with the formal definition of optimality given by Equation 3.4.2 yet often fail to reach an optimal solution due to premature convergence to a sub-optimal solution. The Wonderful Life Utility does however cope well with the added level of complexity on which this report is based. Whereas Wolpert et al. investigated the Bar Problem with the added option of agents selecting more than one night, this study has added a different novel layer of complexity which can be solved to at a near-optimal degree reliably using WLU in a reasonable amount of time\(^3\).

\(^3\)at least the null and average clamped versions of the WLU
4. Multiple Autonomous Agents in a Grid World

4.1 Description

This problem focuses on multiple autonomous agents navigating a two-dimensional grid world consisting of $n \times n$ squares. This is an extension of the single-agent problems discussed by Sutton and Barto [2] and draws on work done by Wolpert et al. [6] and ’t Hoen [3].

The problem is episodic, consisting of a fixed number of time steps for each problem instance. At each time step each agent moves either up, down, left, or right. All agents start from the same central square in the gridworld 4.1. This presents a more complex situation than the previously discussed Bar Problem as there is notion of an optimal sequence of actions which the agents are to learn as opposed to a single action in the case of choosing a night of the bar to attend.

Tokens are located in various locations throughout the grid world and it is the aim of the system to maximise the value of tokens picked up in a single episode. When an agent enters a square containing a token, the token is picked up and removed from the world.

![Grid World Example](image)

Figure 4.1: Example 5x5 grid world. The X marks where all agents originate and the tokens are represented by coloured circles.

\(^1\)In order for there to be a central square all problem instances in this paper consist of an $n \times n$ grid where $n$ is an odd integer.
When the number of time steps (moves) is reached, an episode comes to an end and the grid world is reset to the state that it started in. This involves resetting the placement of the tokens and the starting positions of the agents. In order to be able to maximise a sum of rewards, the agents are Q-learners (see following section) and through successive episodes they are able to learn incrementally what their optimal sequence of actions should be.

This report adds a further extension to the grid world problem by introducing the notion of agents having differing preferences. A problem instance therefore consists of different types of agent and different types of reward. To reflect empirically that certain agents prefer certain tokens over other ones, a token yields a different value depending on what type of agent picked it up. The problems implemented here investigate the simplest case where there are just two different types of agent and two different types of token: red agents prefer red tokens, whilst blue agents prefer blue tokens. Therefore if it is possible for an agent to collect a token of the same colour then it would be most beneficial to do so rather than for an agent of a different colour to collect it. The fact that there is still some reward given for collecting tokens of a different colour adds complexity to the system. The value a token yields to an agent of the same colour is referred to as the specialised value and the amount yielded to a different coloured agent is the non-specialised value. For example, the specialised value for all tokens might be 100 whilst the non-specialised value is 25. The value yielded to each agent in this case is presented in Table 4.1. Hence, the optimal sequence of actions would involve agents picking up their own colour of tokens if the time steps in which they have to move allows for this. Figure 4.2 shows such a case with an agent of each colour starting from the centre square. The coloured arrows represent a sequence of three moves for each agent of the same colour.

![Figure 4.2: Example of an optimal sequence of actions (Number of time steps = 3). All agents start at X.](image)
4.2. IMPLEMENTATION

<table>
<thead>
<tr>
<th>Token</th>
<th>Value for Red Agent</th>
<th>Value for Blue Agent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>100</td>
<td>25</td>
</tr>
<tr>
<td>Blue</td>
<td>25</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 4.1: Example specialised and non-specialised values of tokens for each agent in a 2-cluster system

As in the bar problem, a suitable payoff utility to reward the agents for their actions needs to be chosen. At each time step, each of the agents receives a reward according to their action chosen and the result thereof. Phenomena such as tragedy of the commons, become increasingly more possible with the added complexity of differing preferences. In Figure 4.2 it is most beneficial in terms of the world utility for the blue agent to collect the blue token and the red agent to collect the red. Imagine, however, that during the first episode the red agent moved to the square with the blue token. This would result in a sub-optimal world utility and if a team game payoff utility were employed then the red agent would get rewarded nonetheless. Due to the nature of the Q-learners, higher rewards would result in that action or sequence of actions being repeated more often and red agent may never explore the world enough to realise that there is an alternative red token elsewhere and that it would be better leaving the blue token for the blue agent. The Wonderful Life Utility addresses this problem elegantly by penalising an agent (giving a negative reward) if it picks up a token that could better be picked up by an alternative agent.

Wolpert et al. suggest that the multi-agent grid world problem as described may be applied to a real-world problem such as autonomous rovers exploring for rock samples to analyse. To extend this analogy to incorporate the idea of differing preference one can imagine different types of rover, specialised for different tasks. All rovers are able to collect all types of rock sample but only those specialised for a particular type of rock are able to analyse it hence gain a higher scientific reward. A mere collection of a rock sample yields a sub-optimal reward and hence samples that are able be picked up by agents specialised in dealing with them should be left uncollected by other non-specialised agents.

4.2 Implementation

The grid world is initialised with the tokens placed in predefined locations with no more than one token on a single square. All the agents are placed in the grid world on the centre square. Each of the agents is a Q-learner (Equation: 4.1) and so each has an \( n \times n \times 4 \) state-action pair matrix of values denoting the expected reward of carrying out each of the actions in each of the possible states. The state is represented by the grid coordinate of the agent in the world. Non-optimistic
Q-values were implemented and so during initialisation all the state-action value pairs are set to zero. Q-values for a particular state-action are updated using the one-step Q-learning equation as stated in Sutton and Barto [2]:

\[ Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right], \]  

(4.1)

where \( \alpha \) is the learning-rate: a constant step-size parameter kept at 0.1 throughout the experiments in this report. Likewise, \( \gamma \), the discount-rate, is kept at a constant value of 0.95 as used by ’t Hoen et al. [3] in their experiments. The discount-rate determines how much an agent considers future actions in a sequence. As the experimental setup has the closest token at most five time steps away from the agents’ starting position, this value is adequate for the Q-value to always propagate from those squares close to the token to the starting square.

At each time step the following procedure is used:

1. choose an agent, that has not yet moved this time step, randomly
2. choose an action using an \( \varepsilon \)-greedy policy
3. update the agent’s location
4. update the agent’s Q-value for the action-state pair chosen
5. repeat until all agents have moved

If an agent chooses an action which would result in it leaving the grid world (i.e. if the agent is at the edge of the map), then, rather than limiting the action-space at these squares, the resultant state is merely the same as the original (the agent is replaced on the map in the same position).

In order to formalise the definition of a single episode of the multi-agent grid world problem, let us define the following:

- \( T \) : The initial locations and types of tokens.
- \( L \) : The location matrix of all agents for all time
- \( L_\eta \) : The location of agent \( \eta \) for all time
- \( L_{\eta,t} \) : The location of \( \eta \) at time \( t \)
- \( V(L_{\eta,t}, L, T) \) : Function that returns the value of the token picked up by agent \( \eta \) when it moves into location \( L_{\eta,t} \)

It is important to keep a static matrix of the token locations and types as, (a) this is used to reset the grid world in a successive episode, and (b) it allows the function \( V \) to calculate the outcome of agents’ moves given a fictional move history. We shall see shortly that this is paramount if one wants to incorporate
the Wonderful Life Utility and investigate what the world would have been like had the agent not existed.

Although the notation of $V(L_{\eta,t}, L, T)$ is identical to that used by Wolpert et al. [6], their version of this function retrieved the value of the tokens from $T$ whereas here, in order to implement differing preferences, the value of the token depends on both the type of token (e.g. red or blue) as specified in $T$ and what type of agent $\eta$ is.

The World Utility $G(\zeta)$ is defined as the sum of the value of the tokens collected by all agents over all time (an episode):

$$G(\zeta) = \sum_{\eta,t} V(L_{\eta,t}, L, T).$$  \hspace{1cm} (4.2)

Let us define the World Reward to be the sum of the value of the tokens collected by all the agents in a particular time step, $t$:

$$R_t(\zeta) = \sum_{\eta} V(L_{\eta,t}, L, T).$$  \hspace{1cm} (4.3)

We are now in a position to define the payoff utilities used to reward each of the agents for their actions at each time step. Using a similar method as that employed when investigating the Bar Problem, three payoff utility functions will be used and performance of each compared.

The first payoff utility function is the Team Game (TG) utility which merely gives the reward that the entire world received in that time step to each of the agents:

$$TG_{\eta,t}(\zeta) = R_t(\zeta) = \sum_{\eta} V(L_{\eta,t}, L, T).$$  \hspace{1cm} (4.4)

Secondly, the Selfish Utility (SU) was implemented which assigns the value of the tokens that each agent picked up itself during a particular time step:

$$SU_{\eta,t}(\zeta) = V(L_{\eta,t}, L, T).$$  \hspace{1cm} (4.5)

Lastly, the Wonderful Life Utility (WLU) implemented for this grid world application dealt only with the null clamping parameter\(^2\) (Section 2) where an agent

\(^2\)The superscript $\theta$ is dropped from this chapter as there is no need to differentiate between multiple WLU derivatives.
is rewarded by being assigned the difference in world reward that it made in a particular time step:

$$WLU_{\eta,t}(\zeta) = R_t(\zeta) - \sum_{\eta' \neq \eta} V(L_{\eta',t}, L_{\eta}, T),$$  \hspace{1cm} (4.6)

where $L_{\eta}$ is the location of all agents other than $\eta$ for all time. This is equivalent to the World Reward minus what the World Reward would be had the agent not existed. By using the framework defined above we are able to simulate what such a fictional world would be like by passing in the locations of all the agents apart from the one being rewarded into the function $V$.

4.3 Experimental Methodology

4.3.1 General

Many problem instances were setup in the testing stages of the implementation. Only those which yielded interesting results are discussed in this report but generally the problems were created in such a way that the optimal world utility was calculable so that the performance of the system could be monitored in terms of how close to an optimal solution it achieved. All problem instances were run 50 times and the average world utility at the end of each of 1000 episodes was recorded.

4.3.2 Problem Instances

Experiment Set 1: 5 × 5 Grid

The first setup consisted of a 5 × 5 grid with just two agents and tokens placed as depicted in Figure 4.3. In all problem instances, unless otherwise stated, the specialised value for tokens is 100 whilst the non-specialised value is 10.

With just two agents, this is not an example of a congestion problem. On the contrary, there is no competition between agents of the same colour at least as long as there is only one of each colour. With the number of time steps fixed at five there are just enough moves for the agents to collect all of the tokens in the optimal way (i.e. the red agent collects all of the red tokens and blue agent collects the blue token). The exact setup of this instance (Problem 1) is described in Table 4.2.
4.3. EXPERIMENTAL METHODOLOGY

This problem was then altered by using eight agents instead (Problem 2) to observe the differences in results as increased congestion is introduced. One can imagine that with multiple agents of each colour, by constructing an optimal collective policy, agents of the same colour compete with one another. As the order in which the agents move is randomised at each time step, a move for an agent which resulted in a high reward in a prior episode might yield far less of a reward in a subsequent episode as a competing agent may have picked up the token from the square first.

**Experiment Set 2: 9 × 9 Grid**

The grid world in Figure 4.4 was used as a basis for several derivative problem instance sets where various changes would be made to investigate the effects of: differing proportions of types of agents, difference in value of tokens for non-specialised agents, and placement of high-value tokens behind low-value tokens. For all setups of the world shown in Figure 4.4, however, the number of agents is kept constant at 20 (10 red and 10 blue agents) and the number of time steps per episode is always 5. This is to ensure that it is possible for all the tokens to be picked up and makes it easier to calculate the value of the optimal world utility.

Table 4.2 represents the problem described above before any alterations have been made (Problem 3). Subsequent rows respresent slight deviations from this problem as described below.

**Derivative 1:** The setup initially had 10 red agents and 10 blue (Problem 3) agents but in order to see the effect of having different proportions of types of agents, the case of having, instead, 5 red agents and 15 blue agents was also investigated (Problem 4) \(^3\).

**Derivative 2:** In a separate series of runs, instead of altering the number of agents, the value that tokens yield if picked up by a non-specialised agent was

\(^3\)this case is analogous to 15 red agents and 5 blue agents
altered. This is the value a red agent receives if it collects a blue token (and vice versa). The different values chosen are as follows and are represented in Table 4.2 as Problems 3, 5, 6, 7 respectively:

<table>
<thead>
<tr>
<th>Specialised Value</th>
<th>Non-specialised Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>10</td>
</tr>
<tr>
<td>100</td>
<td>25</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
</tr>
<tr>
<td>100</td>
<td>75</td>
</tr>
</tbody>
</table>

It was hypothesised that as the non-specialised value increased with respect to the specialised value then the difficulty the system had in reaching collective optimal sequences of moves would increase.

In both of the above derivatives the optimal world utility that can be achieved is constant and equal to 1000. This is because each setup still allows for all tokens to be picked up by specialised agents so the optimal world utility is merely a sum of the specialised values for each token.

**Derivative 3:** To make the problem in Figure 4.4 more complex still, additional tokens were added to the grid world and the number of time steps was increased to 10 to allow for the extra steps necessary to make collection of these additional tokens possible (Problem 8). In order to achieve an optimal world utility each type of agent must learn as a collective to explore both corners of the grid world in order to collect the tokens for which they are specialised. The learning process is hampered due to tokens of the opposite colour ‘hiding’ the higher-value tokens behind.
## 4.3. Experimental Methodology

![9x9 Grid - Split Types of Rewards + Hidden Bonuses](image)

**Figure 4.5:** 9x9 grid - split types of rewards + hidden bonuses

<table>
<thead>
<tr>
<th>Problem</th>
<th>n</th>
<th># Red Agents</th>
<th># Blue Agents</th>
<th># Red Tokens</th>
<th># Blue Tokens</th>
<th>Specialised Value</th>
<th>Non Specialised Value</th>
<th>Time Steps</th>
<th>Optimal World Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>100</td>
<td>10</td>
<td>5</td>
<td>600</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>1</td>
<td>100</td>
<td>10</td>
<td>5</td>
<td>600</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>100</td>
<td>10</td>
<td>4</td>
<td>1000</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>5</td>
<td>15</td>
<td>5</td>
<td>5</td>
<td>100</td>
<td>25</td>
<td>4</td>
<td>1000</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>100</td>
<td>50</td>
<td>4</td>
<td>1000</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>5</td>
<td>100</td>
<td>75</td>
<td>4</td>
<td>1000</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>15</td>
<td>15</td>
<td>100</td>
<td>10</td>
<td>10</td>
<td>3000</td>
</tr>
</tbody>
</table>

**Table 4.2:** Grid World Problem Instances

### 4.3.3 $\varepsilon$ Survey

Analogous to the temperature survey conducted for the Bar Problem, a survey of different values of $\varepsilon$ was also conducted for each of the payoff utility functions. The grid world instance Problem 3 (Figure 4.4) was used for this purpose and the values of $\varepsilon$ which resulted in the highest world utility after 1000 episodes (averaged over 50 runs) was chosen for each of the payoff utility functions in all subsequent experiments.
4.4 Results

This section presents the findings of the experiments defined in the previous section. In most cases the World Utility plotted against successive episodes of a particular problem is used to compare the three payoff utility functions. Where world utility is shown it is normalised\(^4\) to incorporate the notion of optimality where a value of one represents an optimal solution to the problem instance. Normalisation is achieved by dividing the world utility at the end of each episode by the optimal world utility possible (see Table 4.2). All experiments were run 50 times and the average resultant world utility at each episode makes up the data plotted. The experiments are split up into sections below defined by the feature of the system which they are intended to investigate.

4.4.1 \(\varepsilon\) Survey

The results in this section are of running Problem 3 (Table 4.2) with various values of \(\varepsilon\) for each payoff utility. Graph 4.6 shows that altering the \(\varepsilon\) in fact made very little difference to the resultant world utility each system achieved after 1000 episodes. Nevertheless, the value for of \(\varepsilon\) for which the world utility was highest was chosen for each payoff utility for subsequent experimentation. These optimal values are presented in the following table.

<table>
<thead>
<tr>
<th>Payoff Utility Function</th>
<th>Optimal (\varepsilon) value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TG</td>
<td>0.01</td>
</tr>
<tr>
<td>SU</td>
<td>0</td>
</tr>
<tr>
<td>WLU</td>
<td>0.01</td>
</tr>
</tbody>
</table>

![Figure 4.6: Average world utility after 1000 episodes for different values of \(\varepsilon\) for each of the payoff utility functions](image)

\(^4\)Except for the \(\varepsilon\) Survey where the world utility is not normalised
4.4. RESULTS

4.4.2 Greater COllective INtelligence

This section compares the results of Problems 1 and 2 (see Table 4.2) the performance of the system in each problem is shown in Figures 4.7 and 4.9 respectively.

![2-Agent 1 Optimal Solution](image)

Figure 4.7: Simple problem: 2 Agents, 1 Optimal solution

The performance of Problem 1 is poor for all payoff utilities with the system achieving a world utility not higher than 35% of the optimal on average after 1000 episodes (Figure 4.7). Studying the low-level behaviour of the agents shows that instead an optimal sequence of actions being followed such as that in Figure 4.8, the system only manages to achieve a sub-optimal sequence in the amount of learning time given (1000 episodes). The optimal solution as shown in Figure 4.8 demonstrates that in order to achieve optimality the blue agent must learn specifically to avoid the red tokens. This is the trait which is not learned in Problem 1 in the time given and a common end solution after 1000 episodes consists of the blue agent moving directly upwards from its starting position, thereby collecting a red token (for a sub-optimal reward) and the blue token. This leaves less red tokens for the red agent to collect. The graph in Figure 4.7 shows WLU to have a slightly higher performance and this is due to the fact that the blue agent can be penalised (given a negative reward) for taking a red token when the red agent could have picked it up. The reason why learning is extremely slow for the WLU especially is that there are exactly two optimal sequences of actions to be found: that shown in Figure 4.8 and if the red agent were to reverse its sequence in a clockwise direction.

Looking at the results of Problem 2 (Figure 4.9) which has 4 red agents and 4 blue agents, a huge increase in performance is clear when compared with Problem 1. This shows that that ability for the agents to learn as collective outweighs any disadvantage produced by added competition between agents. It is this competition, however, which causes problems for the Selfish Utility and the Team-Game utility. In order to reach an optimal solution in Problem 2, or indeed all instances of the grid world problem where there is congestion (too many
Figure 4.8: Optimal sequence of actions for Problem 1. Agents start at centre square.

Figure 4.9: More complex problem: 8 Agents, multiple optimal solutions

agents/too few tokens for all agents to pick up a token), certain agents must learn to avoid the tokens altogether so that other agents can have priority. The WLU achieves the necessary role assignment to each of the agents and consistently converges to the optimal solution for Problem 2 where a varying number of the agents don’t collect any tokens.

4.4.3 Split types of rewards

Let us now deal with the performance of Problem 3 (Table 4.2) as it is the basis of all the subsequent experiments. Compared to Problem 2, it is clear from Figure 4.10 that Problem 3 represents a more difficult problem to solve. The reduced values of the curves show that the world performance achieved for Problem 3 was less than Problem 2 for all three payoff utilities.

Figure 4.11 shows an optimal sequence of actions for Problem 3. By studying the low-level behaviour of the system in terms of the sequences actually learnt by the agents through successive episodes, we gain a better understanding about how the actual solutions are sub-optimal and can speculate reasons for this. After
4.4. RESULTS

1000 episodes of learning, however, even a system using WLU rarely manages to pick up all the tokens successfully. Figure 4.10 clearly shows that WLU is far superior to the two alternative payoff utilities but that it takes longer than 1000 episodes to reach a fully optimal solution on average.

![Split types of rewards](image1)

Figure 4.10: Split types of rewards

![Optimal sequence of actions for Problem 1. Agents start at centre square.](image2)

Figure 4.11: Optimal sequence of actions for Problem 1. Agents start at centre square.

The degrees to which each payoff utility manages to reach an optimal solution such as that in figure 4.11 differs greatly. Inspection showed that certain tokens in the Problem 3 setup are more difficult for the agents to learn to collect. This is because for the tokens closest the edge of the world have fewer different paths from the agents’ starting position to them. Figure 4.12 shows the number of possible paths to each token. The difficulty is inversely proportional to the number of possible paths.

Most solutions found by the team-game utility only manage to collect the two tokens which have five possible paths to them and sometimes a few of the tokens
valued '4' on Figure 4.12. The selfish utility usually found a solution which collected at least the tokens with either 4 or 5 possible paths to them but rarely any of the tokens at the edge of the world. The WLU, on the other hand, usually converged to a solution collecting all but one or two of the tokens at the edge of the world. This shows that agents being rewarded with WLU are able to learn far quicker. All three payoff utilities showed evidence of agents picking up tokens for which they were non-specialised and hence contributing to the sub-optimality of the world but the WLU showed this to happen far less than the SU or team-game.

4.4.4 Different proportions of types of agents

This section presents the result for the Derivative 1 problem instance (Problem 4) as described in Section 4.3.2. The aim of the system in Problem 4 is the same as that of Problem 3 as both the optimal world utility possible and how this is achieved is common for both instances. It was designed to see if having less of a certain type of agents and more of the other had an effect on the performance of the system. Results showed that the effect of this was very small indeed and that the performance of Problem 3 and 4 were very similar for all three payoff utilities. Figure 4.13 shows the performance of problems 3 and 4. The curve labelled 'even distribution' is the performance of Problem 3 and the 'skewed distribution' curve is the performance of Problem 4. The latter curve shows a slightly slower learning rate.

Figure 4.13 shows that having a skewed proportion of agents made a slight but definite negative impact on the performance of the system. This is because although there were more blue agents to work collectively and find the blue
tokens; with fewer red agents, the task of finding red tokens was made more difficult. This demonstrates how types of agents work collectively as a group as well as showing how the whole system of agents contribute towards an optimal world utility.

4.4.5 Extent of Specialisation

Let us now look at the results for Derivative set 2 (see Section 4.3.2), which set out to see what effect changing the values that tokens yield to agents of a different colour (non-specialised value) has on the performance of the system. Figure 4.14 shows the performance of the WLU with four different values that non-specialised tokens yield. Converse to what was hypothesised, the highest non-specialised value resulted in the best performance.

This is misleading however as the normalised world utility is not an appropriate performance measure in this case. As the non-specialised value increases then the world utility will increase if there are agents in the system choosing non-specialised tokens. This increase in ‘performance’ is independent of better
solutions actually being found. A better measure of performance for this case is
to consider the percentage of time an optimal sequence of actions was found after
1000 episodes. Figure 4.15 shows this new performance measure.

![Graph showing percentage of optimal policy at various non-specialised token values]

Figure 4.15: Extent of specialisation - Percentage of time optimal policy is
achieved

Interestingly, the largest difference between specialised and non-specialised values
of the tokens did not yield the best results in terms of percentage of times an
optimal policy was learned by the agent collective. Nevertheless, as the non-
specialised value is considerably increased, relative to the specialised value, the
frequency of the optimal policy drops off considerably.

4.4.6 Hidden bonuses

Dealing with Derivative 3 (see Section 4.3.2) now, we are presented with a far
more complex solution where an optimal sequence of actions can not easily be
worked out. It is possible for all of the tokens to be picked up by agents of
the same colour but the agents must learn to 'wait' until the opposite colour
tokens have been removed before moving to collect the tokens, for which they are
specialised, behind. Figure 4.16 shows the performance of the system described
by Problem 8.

If we compare the graph in Figure 4.10 with that in Figure 4.16 we can see that in
the latter, TG and SU were able to get slightly closer to the optimal world utility
given more tokens to choose from and despite the optimal world utility being far
higher. WLU on the other hand shows a steady learning curve as before, however
a little slowed on this more complex case compared with the former.

Looking at the low-level behaviour shows that the team-game utility results in
very few tokens being picked up at all: usually only the same ones as those
mentioned for Problem 8. The selfish utility results in the agents learning to pick
up nearly all of the tokens (the tokens towards the edge are more difficult due to
the reason mentioned above) but the performance of such a system is far from optimal as many of the tokens picked up are by agents of the opposite colour. WLU results in most tokens being picked up but the number often being less than the SU achieved. WLU results in a far higher performance than SU, however, as nearly all of the tokens picked up were collected by agents of the same colour.

4.4.7 Overview

The Wonderful Life Utility has proved to result in consistently better performance than the other two payoff utilities studied. The Selfish Utility, although able to collect tokens in a grid world successfully if there are many agents (performing considerably better than a team-game approach), suffers from not be able to communicate knowledge to the agents through its reward signal about which tokens in particular are better for each agent. In more complicated instances of the grid world with differing preferences, SU and team-game payoff utility either learn extremely slowly or suffer from premature convergence to sub-optimal solutions. WLU, on the other hand, consistently demonstrated that it could result in a system in which the agents showed a steady learning curve.
4. MULTIPLE AUTONOMOUS AGENTS IN A GRID WORLD
5. Conclusion

This study has extended two commonly studied theoretical multi-agent systems with a novel layer of complexity by introducing the notion of differing preferences between the agents. The COllective INtelligence framework has been used to incorporate the Wonderful Life Utility which has been proven to work well with such problems in previous work. The investigation here has shown that the WLU can indeed be applied to system with differing preferences proving that there is scope for scaling use of the WLU to increasingly complex problems even whilst using relatively simple reinforcement learning techniques.

Although the El Farol Bar Problem is computationally a simple congestion problem, the conceptualisation of performance of solutions is more difficult than understanding various hand-coded instances of a Grid World Problem. Although it was possible to incorporate the notion of differing preferences into the bar problem, the structure of the grid world problem lends itself better to this task: visualising the problem is aided by the fact that the grid world problem can be represented pictorially far easier and by using the analogy of different colours for specialisation, differing preferences seem to be a natural extension of the problem. Another strength of the grid world problem is that competition between agents can be increased by adding more to an instance without altering the other properties of the system, e.g. optimal world utility as would be the case if the number of agents was changed in an instance of the bar problem.

There is scope in the bar problem, however, to alter the structure of the preference functions, which may improve the understanding of the problem. Various simplifications may also be made such as decreasing the number of days (actions) the problem deals with. Another alteration which may be interesting to investigate would be to make the bar problem into a sequential problem, much like the grid world problem: instead of a single-move game, the agents would be required to learn a sequence of actions.

An interesting extension to the grid world problem would be to investigate the performance if the agents has the ability to sense tokens. A sight or smell analogy could be applied meaning that agents have the ability to sense a token without being in the same square as it. The token might emit a ‘scent’ for, say, a 2-square radius around it or an agent might be able to ‘see’ a token in the distance if it has line of sight. With such extensions, instead of the agents learning a set sequence of actions with tokens at fixed positions, they may be able to learn a policy based on environment conditions which they sense. How COIN could be incorporated with such a system may be interesting to investigate.

In general, the way in which the agents work as a collective could be extended.
Adding communication between agents of a same cluster would allow knowledge of the system to be transferred explicitly between agents. The effect this has on the performance of the system versus a system without communication might be interesting to investigate. This approach does, however, lead away from the COIN way of thinking dealing with systems where there must be "little or no... communication" [4]. Nevertheless, comparing the performance of such a system with the original investigation presented here may prove that the system without direct communication performs as well as a system with communication. A situation where inter-cluster communication is possible could also be implemented which would give clusters of agents the knowledge about the preferences of other clusters in the population. Whether or not this facilitates group learning would be a useful result.

The results of the experiments described in this report generally agree with what has been done in previous work with COIN and both the Bar Problem and the Grid World Problem. The Wonderful Life Utility proved to be an effective way of ensuring that the agents learnt in a way that the entire collective benefitted. There remain some scalability issues, however, as performance decreases as the complexity of the problem increases. Although the Wonderful Life Utility theoretically converges to an optimal solution, the rate at which learning takes place depends on the many other factors that make up a problem. In terms of real-life applications the COIN framework, which one must adhere to in order to ensure the WLU works, is limited. To deal with more complex problems, research into using more sophisticated reinforcement learning techniques would be useful such as that conducted by ’t Hoen et al. [3].

Nevertheless, it remains impressive that the systems used as case studies in this report, conforming to the COIN framework specification in that there is no centralised control or communication between agents, still manage to facilitate effective individual agent learning towards an unknown collective goal.
Bibliography


