UNIVERSITY OF SOUTHAMPTON

FACULTY OF LAW, ARTS & SOCIAL SCIENCES

School of Management

Evaluating Reinforcement Learning for Game Theory Application

Learning to Price Airline Seats Under Competition

by

Andrew Collins

Thesis for the degree of Doctor of Philosophy

January 2009

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF LAW, ARTS & SOCIAL SCIENCES
SCHOOL OF MANAGEMENT

Doctor of Philosophy

EVALUATING REINFORCEMENT LEARNING FOR GAME THEORY APPLICATION: LEARNING TO PRICE AIRLINE SEATS UNDER COMPETITION

by **Andrew Collins**

Applied Game Theory has been criticised for not being able to model real decision making situations. A game's sensitive nature and the difficultly in determining the utility payoff functions make it hard for a decision maker to rely upon any game theoretic results. Therefore the models tend to be simple due to the complexity of solving them (i.e. finding the equilibrium).

In recent years, due to the increases of computing power, different computer modelling techniques have been applied in Game Theory. A major example is Artificial Intelligence methods e.g. Genetic Algorithms, Neural Networks and Reinforcement Learning (RL). These techniques allow the modeller to incorporate Game Theory within their models (or simulation) without necessarily knowing the optimal solution. After a warm up period of repeated episodes is run, the model *learns* to play the game well (though not necessarily optimally). This is a form of simulation-optimization.

The objective of the research is to investigate the practical usage of RL within a simple sequential stochastic airline seat pricing game. Different forms of RL are considered and compared to the optimal policy, which is found using standard dynamic programming techniques. The airline game and RL methods displays various interesting phenomena, which are also discussed. For completeness, convergence proofs for the RL algorithms were constructed.

Contents

List of Figures

List of Tables

DECLARATION OF AUTHORSHIP

I, Andrew James Collins, declare that the thesis entitled *Evaluating Reinforcement Learning for Game Theory Application: Learning to Price Airline Seats Under Competition* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- none of this work has been published before submission

Signed:

Date:

ACKNOWLEDGEMENTS

For his patience, calm and above all encouragement, my supervisor Professor Lyn Thomas has been a source of inspiration to me throughout the research and deserves special thanks. I have failed to count the number of times I have asked Professor Thomas for advice on a problem I have been struggling with for weeks, only to receive the correct answer within a matter of minutes. From discussion with my contemporaries, I realise that I am unique in my continual enjoyment of the PhD process and I know that I have Prof. Thomas to thank for that.

I am also grateful to the EPSRC who funded me throughout this research.

I would also like to thank Dr David Baker for helping me compile my programs on the University of Southampton's super-computer. Without Dr Baker's expert knowledge this would never have been achieved and I would have never have completed the number of runs (and re-runs) required for this research.

There are several other people I would like to thank, namely: Prof Robin Mason (for his expert advice on Game Theory and for stopping me trying to read all of von Neumann and Morgenstern's book), Dr Christine Currie (for introducing me to airline pricing and for good advice about conducting a PhD, which I passed on but never followed), Dr Wahib Arroum (for the frequent mathematical discussions which would occur on social occasions *with probability one*), Ms Jayne Cook and the School of Management.

Finally, I would like to thank my wife, Mrs Sue Collins for her support and for proofreading my thesis (even though she disagrees with my use of terms like *interestingly* and *surprisingly*). She also deserves thanks for marrying a penniless student.

Definitions

Notation	Definition
e	Episode
n	Round
i	Player index
P1	Player One
P2	Player Two
λ	Learning Rate or Step-size parameter
τ	Temperature parameter
β	Scale parameter

Chapter 1

Introduction

Game Theory is the main analytical method that is used to find the *optimal* solution[1] when a theoretical model involves more than one decision maker. Finding a game's solution can be intelligently and computationally hard, which leads to a tendency for over-simplistic games. In subject areas like Economics, these simplistic games are acceptable due to their requirements. However, the validation rigour and direct *real-world* application of Operational Research (OR) means that these simplistic games are avoided.

OR modellers tend to use scripted behaviour especially within simulation environments and when dealing with multiple agents within a model (see Chen and Zhan, 2008; Bailey, 2003, for examples). Though this application can be acceptable to the decision maker, it removes a level of sophistication and assumes that the modeller knows and can anticipate what actions the agents will take. An inductive approach is commonly used within simulation to determine the scripted behaviour (i.e. the simulation is run, agents' behaviour is observed and then modified to achieve *desired* results). This approach to determining an agent's behaviour (or policy) can lead the simulation to produce self-fulfilling results due to the assumption made by the modeller.

[1] There are various debates about what an optimal solution of a game means (see Binmore, 1990). There are different solution concepts available to an analyst i.e. Maximin, Nash Equilibrium, Stackelberg Equilibrium, Core, Shapley Value, etc. See Thomas (1984) for more details.

Using scripted behaviour, as an alternative to Game Theory, is not always ideal and this implies that there is benefit in developing Game Theoretic techniques that can be applied in a *real-world* context.

1.1 Objective

The thesis presents research into possible means to overcome one of the difficulties of applying Game Theory within a practical context. There are several problems with applying Game Theory (e.g. deriving the payoffs from play (Collins et al., 2003; Barzilai, 2007)) but the focus of this thesis is on the difficulty of finding a solution to complex games (i.e. games that are likely to be encountered in practice). Reinforcement Learning (RL) has been suggested as one method to overcome this difficulty (Ravulapati et al., 2004) and this thesis aims to give a detailed evaluation of several different RL techniques.

The evaluation of the RL techniques was applied to a single case study. The application of Game Theory to pricing models with complex customer models is a current issue in Revenue Management (see Boyd, 2007). An airline pricing game was chosen as the case study model. The airline pricing game remains simple enough to be solved in the traditional sense (i.e *backward induction*, see Fudenberg and Tirole (1991)) so that the RL results can be compared to a game's solution. A new game has been developed for the research and its solutions analysed and explained before the Reinforcement Learning results were analysed.

Ideally, the learning players (under the RL technique) would learn to play like the *Nash Equilibrium* policy (see Nash, 1951), which is considered the standard solution to a game. The Nash Equilibrium policy is not necessarily unique, so it was useful to find out which policies the learning players had learnt to play. If the learning players did not play like any Nash Equilibrium policy, what kind of policy did they play like? For instance, they might have played randomly or myopically. Answering these type of questions was the second aim of this thesis.

This thesis is about practical application of Game Theory and it is important to consider the practical limitations of the RL technique. These are also presented here.

Though the RL techniques do not always reach the Nash Equilibrium policy[2] within the time-frame available to run the method, it was important to know whether the techniques would reach this policy theoretically. An example proof of convergence is offered within this thesis.

The RL techniques were studied within the framework of a simple pricing game. They were then applied to more complex games (i.e. with advanced customer behaviour (see Talluri and van Ryzin, 2004, for details)) to understand the impact of this complexity.

Objectives

The objectives of the research were three-fold. Firstly, they were to see which RL techniques (SARSA, Q-Learning or Monte Carlo learning) produced the *best* results when applied to a simple airline price game. To determine these *best* results, each technique results were compared to those generated by the Nash Equilibrium (or a variation of it).

Secondly, what other results could be drawn from these experiments? These were explored by comparing the learnt results to those of myopic and random play. Theoretical convergence results for the SARSA method were found.

The final objective was to find the limitations of using RL to solve a simple airline pricing problem, both computational limitations and limitations to the complexity of the model. Computational limitations were discovered from experimental runs of the model and complexity issues were address for varying parts of the model (i.e. the customer model).

Benefit of this Work

It is not being suggested that Game Theory should become an all encompassing technique for solving OR problems. A problem should be solved by the appropriate technique and model-fitting (Pidd, 1996) should be avoided. Game Theory as a modelling technique should be part of a coherent OR practice framework (see Murphy, 2005). However, the research presented within this thesis gives an insight into the possible

[2]Actually, this thesis is concerned with a variation on the Nash Distribution policy, this is discussed within the literature review.

ways and benefits of applying Game Theory within a practical context. The research also gives a presentation of the limitations.

Another benefit of the research is the results obtained from the airline pricing game used as the case study. The game solutions are interesting in their own right and give insight into ways that airlines can dynamically price their seats.

1.2 Overview of Thesis

The thesis has been divided into eight chapters and four supporting appendices. The appendices display a selection of tables which represent the important empirical and theoretical results. A brief summary of each chapter (excluding this one) is given below.

Chapter Two of this thesis highlights the current relevant literature and the associated issues. There are three research areas considered in this thesis, namely: Game Theory, Reinforcement Learning and Revenue Management. Each of these areas is considered in turn and important terms and methods are defined. Where necessary, further reading is suggested within the chapter.

In Chapter Three, the research methodology is introduced. This includes the formalisation of the problem and the technical approach to the analysis. The chapter discusses the choosing of a case-study to investigate the different Reinforcement Learning techniques and what assumptions were required for this to be implemented. An airline pricing game was selected as the case-study game and is introduced in this chapter. An assessment criteria mechanism for comparisons was also required and this is discussed.

In Chapter Four, the airline pricing game is constructed using the framework outlined in Chapter Three. This game is *solved* by finding the Nash Equilibrium and Nash Distributions. A discussion about the sophistication and implications of these solutions follows. In Reinforcement Learning methods are implemented via an overarching learning model, which was constructed using the C++ programming language. A description of how this was done and the verification and validation of the model are also included in this chapter.

In Chapter Five the empirical results from the learning model are presented. An assessment of the different comparison techniques is given and a comparison method is chosen. The learnt policies are compared to various standard policies and a detailed description is given of how learning occurs. The physical limitations and scalability of the model is discussed.

Chapter Six contains a proof of convergence for the SARSA method within the context of the learning model. This chapter can be considered independently of the other chapters and its own notation is given. An inductive proof is used and is built up, using stochastic approximation methods, around a basic conceptual framework of the game.

In Chapter Seven, two variations of the game are considered. The first variation considers the results from a *meta-game* where the airlines are allowed to vary the number of seats that they have available. The second variation looks at using a more sophisticated model of customer demand, arrival and acceptance. The chapter also briefly discusses previous versions of the game and possible further research.

A summary of the research and conclusions are given in Chapter Eight.

Chapter 2

Literature Review

2.1 Introduction

The research presented in this thesis touches on many different academic fields, for instance Operational Research (OR), Game Theory and Artificial Intelligence (AI). The casual reader is not expected to be well-versed in the theory and developments of these fields so an introductory overview of them is presented in this chapter .

The scope of this literature review will mainly be confined to the academic publications (i.e. journals, conference proceedings and books). The intention is to present the research from an academic perspective and therefore have been ignored possible anecdotal information sources (i.e. commercial airline databases and media reports). This limitation of possible sources could have lead to bias within the research and the thesis. To counteract any bias present, any criticism of the proposed techniques found was included.

Aim

In this chapter it is intended to introduce some of the fields, theory and ideas that were used within the research for this thesis. As this research presented here is original, it is important to establish the relevant research which has preceded it. Each field is introduced in turn, giving a brief description, history and recent developments within the field. Where appropriate, mathematical formulae have been included that highlight the analytical techniques that were employed later in the thesis.

Multiple Discipline

The work undertaken covers many different fields, which can be both an advantage and a disadvantage. Using a multi-discipline approach allows us to draw on several different research resources; it also means that problems are encountered like differences in the paradigms and terminology. There can also be difference in presentation style of the work.

As this thesis has been conducted as part of an examination of a PhD in Management Science / Operational Research, the style and terminology of the thesis is as expected for a piece of OR literature. Where ambiguity from the different fields arises, it is intended that this will be explicitly made clear.

Overview of Literature Review

Each of the fields described above are not independent and there exists a large body of multi-disciplinary literature already. This means each field cannot be considered individually. To give the literature review the coherent flow, the different fields are combined and presented in the following order:

- Operational Research and Game Theory
- Learning in Games
- Reinforcement Learning
- Revenue Management

The remainder of this chapter is divided up in to sections determined by the above headings. The sections themselves are not independent either and the later sections do refer to earlier ones.

2.2 OR and Game Theory

Operational Research (OR) [1] (see Winston, 1993) is defined by the Operational Research Society as looking at an organisation's operations and uses mathematical or computer models, or other analytical approaches, to find better ways of doing them

[1] Also known as Operations Research, Operational Analysis or Management Science (MS)

(see Quinton, 2007). To be able to use the analytical approaches, operational researchers must first develop them. Some of these techniques are widely used and others not so much. One technique requiring this development is Game Theory.

Game Theory

Game Theory (GT) is the study of multi-agent decision problems. Game Theory is not exclusive to OR, in fact its home is in micro-economics (see Fudenberg and Tirole, 1991). There have also been several successful applications of the technique in areas as diverse as computer science (Dash et al., 2003), evolutionary biology (Maynard Smith, 1982, 1974) and many others (see Fudenberg and Tirole, 1991, for details).

Modern GT can split into three basic types: Zero-Sum games, Non-Zero-Sum games and cooperative games. A new type of game has arisen in recent years, these are called *soft* games, which are related to *soft* OR (see Howard, 2001; Bryant, 2007). The research presented within this thesis is concerned with extensive form (as opposed to normal form) Non-Zero-Sum games.

History

Game Theory was started when, in the nineteenth century, Antione Cournot proposed an idea that economist should look at situations where there are only a *few* competitors (Cournot, 1838). Until that point, economists had only looked at markets without competition (called Crusoe on his island) or when there was infinite competition (called Multeity of atoms), see Eatwell et al. (1987) for details. The work was virtually ignored until John Von Neumann and Oskar Morgenstern wrote their ground-breaking work *Theory of Games and Economic behaviour* during the Second World War (von Neumann and Morgenstern, 1944). Their work became the bedrock of modern Game Theory.

Seven years later, John Nash develop his Nash Equilibrium concept (Nash, 1951)which allowed Game Theory to become the useful technique which it has become today. This development won Nash, with John Harsanyi and Reinhard Selten, a Nobel prize in 1994 (Kuhn and Nasar, 2002; Harsanyi and Selten, 1988). Over the preceding years, Game Theory has been developed and adapted further via tens of thousands

of academic publications. However, these developments did not leave Game Theory without controversy, one of which is the interpretation of its results.

Interpretation

There are generally two ways within which Game Theoretic results can be interpreted (see Hume, 1740; Binmore, 1990): normative and descriptive[2]. The descriptive interpretation tries to explain real-world phenomenon where multiple agents interact. The normative interpretation is that a model shows the decision-maker how they should '*play the game*'. The interpretation depends on the level of abstraction.

Though the descriptive interruptive has been successfully used within positive economics (Friedman, 1953), a much more specific problem is being addressed here and weakness in this paradigm begin to creep through. One such weakness is the use of *Homo Economicus* [3] (see Persky, 1995).

Homo Economicus is the ultimate competitive player of a game. Homo Economicus has infinite intelligence, rationality and knowledge. Homo Economicus will always play a Nash Equilibrium and will always find the weakness in an opponent's play. Though Homo Economicus is the underlying player used within Game Theoretic modelling, they do not exist. The kind of character that *Homo Economicus* represents can be compared to the political doctrine in Machiavelli's famous work *The Prince* (Machiavelli, 1532).

Learning players are being used and it might be hoped that they learn to play like Homo Economicus eventually but it is not assumed that this is how the real world system works. Therefore a normative view of Game Theory is being used, which seems appropriate in this context.

By being normative (or saying a game should be played) does lead to some interesting problems with validation, as perfect play is unlikely to be performed in practice. However the intention is not to move away from Homo Economicus and concepts like Nash Equilibrium will be used to act as the underlying paradigm of the games.

[2]Also know as constative or positive view

[3]Also know as Economic Man

Nash Equilibrium

Nash Equilibrium was introduced by John Nash during the fifties (see Nash, 1951). A Nash Equilibrium of a game is a special set of policies used by the players. The *policy* (also known as *strategy*), is the mechanism that the players use to choose their actions within a game. If all players are using their respective Nash Equilibrium policy, then no player can gain a higher expected reward by changing to any other policy. This does not mean that players get the maximum reward obtainable within the game.

In many games both players could do better with a different set of policies than a Nash Equilibrium one. Both players could agree to undertake their respective policies to achieve this higher reward. However, Homo Economicus would deviate from policy to gain a greater reward at the expense of the other player. When both players' policy is a Nash Equilibrium, Homo Economicus does not have any incentive to change and the outcome of the game remains stable.

There are other solution concepts, like minimax (see von Neumann and Morgenstern, 1944), but a Nash Equilibrium is a generally accepted concept within the Game Theory community. Mathematically, the Nash Equilibrium can be represented as best-response function to an opponent's policy.

For a player's action $a \in A$ the expected reward , under the opponent's current policy, is $Q(a)$. A player's policy $\pi \in \Delta$ is considered to be a probability measure[4] over finite set A. Then the best-response policy is:

$$\beta(Q) = argmax_{\pi \in \Delta} \left\{ \sum_{a \in A} \pi(a) Q(a) \right\}$$

When both players are using a best-response policy to each other's policy, then a Nash Equilibrium is achieved. This pair of policys are not necessarily unique and selection of a Nash Equilibrium pair has been the focus of much research (i.e. Harsanyi and Selten (1988); Herings et al. (2003)).

[4] A probability measure is not defined here. Please see Williams (1991); Durrett (2004) and chapter six for details on measure theory

If a policy is deterministic (i.e. the player has one one possible action response to any situation presented to them) then this is called a *pure* strategy. If, however, the policy allocates a probability to selecting certain actions in response to a certain situation then this policy is called a *mixed* strategy. The concept of a mixed strategy can be difficult to interrupt, especially in the one-off games (see Binmore, 1990, for more details).

For example, consider advising someone that their Nash Equilibrium policy is to play one action 99% of the time and another only 1% of the time. If they were only going to play the game once, you might expected them to just play the first action without bothering to randomise their choice between the two, hence they would be playing a *pure* strategy and not the *mixed* strategy suggested. This could result in them not gaining the best response benefit that the Nash Equilibrium offers (i.e. their opponent is likely to realise that they will only play the pure strategy and will change their strategy accordingly). Repeatedly played games are the only ones that are considered within this thesis so this dilemma is of no consequence to this research and mixed strategies can be used without fear[5].

If all the possible actions $a \in A$ have a positive probability of occurring then the policy is called a *totally mixed* strategy. A totally mixed Nash Equilibrium strategy can have good stability properties and sometimes game theorists insist that the players only use totally mixed strategies (this version of a game is called the *perturbed* game). Perturbed games are behind the trembling hand perfect equilibrium which was part of the Nobel prize winning work of John Harsanyi and Reinhard Selten (see Harsanyi and Selten, 1988). A variation on the Nash Equilibrium concept which always considers perturbed games is discussed below.

Nash Distribution

This variation on the Nash Equilibrium is called the *Nash Distribution* (Fudenberg and Kreps, 1993; Fudenberg and Levine, 1995, 1998, 1999). Unlike the Nash Equilibrium, a Nash Distribution policy always gives a positive probability of selecting every action available. This perturbed policy is a very useful property when a player is unsure of the rewards 'Q' they obtain from each action. The Nash Distribution ensures

[5]In fact, it was required to use them for the modelling method to work.

it is a perturbed policy by using a *smooth best response* function, which incorporates *smoothing function* 'υ' and a *temperature* parameter $\tau > 0$.

υ is a smooth strictly differentiably concave function on a policy and the temperature parameter is fixed. For the smoothing function, a variation called *Logistic Fictitious Play* is used which was introduced by Fudenberg and Levine (1995) [6]:

$$\upsilon(\pi) = -\sum_{a \in A} \pi(a) ln(\pi(a))$$

This makes the our *smooth best response* function for a given player:

$$\begin{aligned}
\beta(Q) &= argmax_{\pi \in \Delta} \left\{ \left(\sum_{a \in A} \pi(a) Q(a) \right) + \tau . \upsilon(\pi) \right\} \\
&= argmax_{\pi \in \Delta} \left\{ \left(\sum_{a \in A} \pi(a) Q(a) \right) - \left(\sum_{a \in A} \pi(a) ln(\pi(a)) \right) \right\} \\
&= argmax_{\pi \in \Delta} \left\{ \sum_{a \in A} \pi(a) . \left(Q(a) - \tau . ln(\pi(a)) \right) \right\}
\end{aligned}$$

From Fudenberg and Levine (1999), it is seen that:

$$\beta(Q)(a) = \frac{e^{Q(a)/\tau}}{\sum_{b \in A} e^{Q(b)/\tau}} \tag{2.1}$$

Using this υ, the smoothed best response function has been transformed into *Boltzmann*[7] action selection. This method of action selection was first proposed by Luce (1959) though it has been compared to Thurstone's *Law of Comparative Judgment* (see Thurstone, 1927a; Fudenberg and Levine, 1998) and to the *multinomial-logit* model of customer-behaviour (see Talluri and van Ryzin, 2004).

The Boltzmann action selection weights the different actions available by their expected return, as τ decreases the bias is towards the action which yields the largest return. Thus, in the limit of τ decreasing to zero; the player will select an action greedily. This greedy action selection corresponds to a Nash Equilibrium. This leads to the important property of Nash Distributions, which is that they will converge to a unique Nash Equilibrium as τ is decreased to zero (see Fudenberg and Levine,

[6] They originally called it 'κ-exponential fictitious play'

[7] also known as *Gibbs* action selection or *Softmax* action selection, see Bridle (1990) for details

1998). Thus for small values of τ it is expected that players will use a policy similar to a Nash Equilibrium policy, only slightly perturbed.

Within the research presented in this thesis, multi-round sequential games are considered. To apply the Nash Distribution within this context can be problematic. Unlike the single stage version, randomization occurs over pure policies (i.e. instructions on which action to take at each stage/state of game) instead of single actions. Finding all these pure policies alone can be computationally intensive.

To overcome this problem, a variation on the Nash Distribution was used. This variation uses multiple randomizations, at each stage of the game, instead of a single Boltzmann randomization at the start. Thus only the current possible actions need be considered by the players in each randomization, as opposed to the complete policy.

Like the original Nash Distribution, this variation has been shown to converge to a Nash Equilibrium policy (see Appendix D for details). Because of the similarity to the original Nash Distribution policy, this variation is referred to as the Nash Distribution policy throughout the remainder of this thesis.

This discussion now moves onto the more practical side on Game Theory. There were several Game Theory terms which have not been explicitly defined here (i.e. Stackelberg leader, Extensive-form, etc.). These terms are briefly mentioned throughout the thesis and can be ignored by a non-expert without loss of comprehension. However, if the reader would like a further introduction to Game Theory please see Thomas (1984) or Fudenberg and Tirole (1991).

Applying Game Theory

Game Theory has been applied to most situations where there are multiple interacting agents be this negotiations (Goodwin, 2005), business (Chatterjee and Samuelson, 2001), social situations (Glance and Huberman, 1994), games (Thomas, 2003) or war (Collins et al., 2003). However, there are several weaknesses of using Game Theory within a practical context. It has already been seen the effect of *Homo Economicus* as Game Theory's paradigm player, now some other limitations are looked at.

In the practical application of a game, the modeller will have to make judgements on the *returns* (also called *payoffs*) received by the players for the different policies that can be played. Deciding what these rewards (or even the player's preference to different possible rewards) will be is non-trivial and the academic field called *Utility Theory* is dedicated to understanding this task (see von Neumann and Morgenstern, 1944; Winston, 1993, for more details). One simple method, which is used by naive practitioners, is to set the return to its expected value (i.e. if gaining money is the objective of the game, then a policy would be determined by the expected amount of money that the player receives). However, another Nobel laureate Maurice Allias showed that humans display paradoxical behaviour toward expected values (which was called *Allias' paradox*, see Allais (1953))[8].

The payoffs used within a game usually need to be accurate because the Nash Equilibrium solution is non-linearly dependent on them. Therefore, a practical game theorist needs to take this on board otherwise they will face the *garbage in, garbage out* maxim[9]. Even if a game has been well constructed, there is no guarantee that a solution (a Nash Equilibrium) can be found.

Finding a Nash Equilibrium of a game can be very difficult especially when dealing with a large or complex game. In a complex game, it may be difficult to explicitly work out all the players' different actions payoffs (and thus a Nash Equilibrium). This can occur with *stochastic* games[10] especially when complex methods are used to determine the next stage of the game. A sequential stochastic game was the type of game which is considered within the research conducted for this thesis.

[8]Another paradox related to expected values is *St Peterburg's paradox* (see Bernoulli, 1738). In this game, a coin is tossed repeatly until a *tail* has been seen, the player then receives 2^n, were n is the number of *heads* seen. The paradox is that under expected value, this game is worth infinite pounds, so the player should be prepared to pay all their wealth to play the game. Would you be prepared to do this?

[9]The phase was derived in the fifties as a teaching mantra by George Fuechsel, an IBM 305 RAMAC technician/instructor in New York. This has been placed within an Game Theory and Operational Research context by Barzilai (2007)

[10]Stochastic games are multi-staged games where there is uncertainty about what the next stage will be. Stochastic games are the multi-player equivalent to Markov Decision Process (MDP) (seee Bellman, 1957; Winston, 1993).

Even if the game is not complex and can be solved using standard approaches[11] problems of complexity can occur. Finding a Nash Equilibrium of a normal-form game has been shown to be PPAD-Hard (see Chen and Deng, 2005). PPAD-Hardness is subclass of NP-Hardness[12] problems. Though solving mechanisms exist for PPAD-Hard problems (i.e. ways to calculate a Nash Equilibrium), the computational time required can be unreasonable for large games.

Both of the problems (finding realistic payoffs and computing a Nash Equilibrium) above have had an impact on the practical application of game theory (Collins et al., 2003). The research presented in this thesis has focused on the second of these problems.

Though all these problems may seem very depressing for anyone wishing to practically apply Game Theory, there are several good books that talk about practical implementation see Chatterjee and Samuelson (2001); Kott and McEneaney (2007) for more details. There is even a Game Theory freeware available called GAMBIT (McKelvey et al., 2007). GAMBIT is a library of game theory software and tools for the construction and analysis of finite normal-form and extensive-form games. A discussion on some of this implimentation is given in the next section.

Games in Operational Research

As mentioned above, Game Theory is one of the many techniques that can be applied within an OR context. A basic introduction to this application of found in Winston (1993) or Thomas (1984). Though this thesis has given a lot criticisms of the applicability of Game Theory, its application does exist. A survey of OR games can be found in Borm et al. (2001) but this is mainly confined to cooperative games[13].

Recent practical advances in the area include Combinatorial Auctions (de Vries and Vohra, 2003) and Congestion games (Roughgarden and Tardos, 2002). This includes the famous Braess' Paradox, which shows that adding more transport links can lead to more congestion (Braess, 1968; Braess et al., 2005). Other examples include ren-

[11] Dynamic programming was used to solve the simple games (see Methodology chapter for details).
[12] PPAD stands for Polynomial Parity Argument, Directed. NP stands for Non-polynomial
[13] Sometimes called *n-person* Game Theory

dezvous search (Thomas and Hulme, 1997) and using game theoretic approaches to bargain over long-term contracts (Kim and Kwak, 2007).

All these approaches suffer from some (or all) the problems highlighted above. One approach to deal with some of these problems has been to apply Artificial Intelligence methods to them.

2.3 Learning in Games

Given the complexity of solving games, it is unsurprising that many people have turned to computers and use them to *learn* the game's solution. Thus the using Artificial Intelligence (AI) in games begun. Using AI to solve games gives us two distinct advantages. Firstly it gives an ability to solve games that were otherwise too complex to handle. Secondly, solutions that are reached under learning conditions can be found (which might indicate which are going to be opposed in the *real world*). These advantages also come with disadvantages. There is no guarantee that the learning dynamic will converge, and even if it does there is no guarantee that it will converge within a reasonable length of time. Once a solution is arrived at, this solution might be far from the stable Nash Equilibrium that is required. Understanding these possible outcomes forms the basis of this research into learning. Now a brief history of learning in games is looked at and then different AI approaches are considered.

History

The first known attempt at using learning to solve games was Brown's fictitious play (see Brown, 1951). Brown described a method of action select that was based on an opponent's previous play. A player would assume that an opponent would choose an action with the same probability as the normalised frequency that that action had been played in the past. The player then simply choices the best-response to this assumed opponents policy. Though this sounds like both players might converge to a common solution, this method is notorious for the player ending up swapping policies in a cyclic fashion, thus no convergence is reached.

Though Brown's method does not guarantee convergence[14] it has inspired academics

[14]The cyclic behaviour can be observed in some games thus convergence is not guaranteed, see Fudenberg and Levine (1995)

to continue applying learning to games. The most successful work on the subject was *The Theory of Learning in Games* by Fudenberg and Levine (1998). The book is based around their work of the previous ten years (for example see Fudenberg and Levine (1995, 1999)) and probably the most citied work on the subject. Though the work was focussed on normal-form games, Fudenberg also considers extensive-form games [15] (see Fudenberg and Kreps, 1993, 1994, 1995). Fudenberg and Levine come from an economics perspective and their book is written for this field, thus it only covers two types of dynamics (which are appropriate for that field): Best-response dynamics and Replicator dynamics. While the economists were concerned with the theoretical side of learning, other AI was being applied to the practical side.

Artificial Intelligence (AI)

Learning in games is not only found in the economic literature but also the computer science literature, especially within the Artificial Intelligence subfield of *Machine Learning*. Learning in games has been part of AI for many years, with the computer scientists looking mainly at the standard games of Chess (Fogel et al., 2005) and Go (Műller, 2002). However, the shift towards using Game Theory came after Robert Axelrod's famous experiment involving a prisoner's dilemma (see Axelrod, 1984, 1997). At present, computer scientists are concerned with how to use multiple agents to achieve specific tasks (called *Mechanism Design*) (Dash et al., 2003). For a good introduction to AI see Russell and Norvig (1995) and for examples of learning in games, from a computer scientist prospective, see (see Littman, 1994; Bowling and Veloso, 2002).

There are many different Artificial Intelligence methods that have been developed over the years and many have been applied to Game Theory. No attempt has been made to cover every possible subject and only a brief review is given here. The major AI techniques that have been applied to games are:

- Evolutionary methods (Weibull, 1995; Maynard Smith, 1982, 1974)
- Neural Nets (Gosavi, 2003; Zizzo and Sgroi, 2007; Neal, 1996)

[15] Extensive-form games are used within the research, in extensive-form a game can be represented as a tree-like structure with the arc representing the different actions

- Reinforcement Learning (Fudenberg and Levine, 1998)

There have also been attempts to bring the learning and games into the commercial world of OR. These attempts include *Linguistic Geometry* by Stilman (2000). Here Howard uses game learning to determine how agents in a mutli-agent simulation will react without using scripted behaviour. The method has had its successes (see Kott and McEneaney, 2007) but remains a heuristic method. For an overview of OR and AI see Kobbary et al. (2007) [16].

Within this research, Reinforcement Learning (RL) has been the exclusive focus. This method was chosen for study for three reasons. Firstly, it is related to a way in which the psychologists believe that humans learn (Leslie, 2001). Secondly, it can be shown to converge unlike some other methods (i.e. Generic Algorithms (Russell and Norvig, 1995)). Finally Reinforcement Learning is not just approximation of another method (i.e. Neural Networks are related to multivariate regression see Neal (1996)). Thus further discussion on Reinforcement Learning is required.

2.4 Reinforcement Learning

Reinforcement Learning (RL), like most techniques, goes by other names (including Neuro-dynamic programming, see Bertsekas and Tsitsiklis (1996)). Within a gaming context, Reinforcement Learning assumes that the players have an approximate knowledge of what rewards (or expected rewards) are associated with the actions available to them, and update this knowledge based on the observations of the outcomes from repeated play of the game. A good introduction to the subject can be found in Sutton and Barto (1998) and a survey of the techniques can be found in Kaelbling et al. (1996). Kaelbling et al does not consider the multi-agent case but a survey of multi-agent Reinforcement Learning is found in Shoham et al. (2004).

The history of Reinforcement Learning comes from two separate strands. One was psychologist attempts to explain animal learning and the other was computer scientists trying to achieve machine learning through trial and error. The psychologist

[16] However, Kobbary et al. (2007) makes no mention of Reinforcement Learning in their paper, for details on this see Gosavi (2003)

strand started with the *law of effect* by Thorndike (1911). The first suggested computational investigation of Reinforcement Learning is found in Minsky (1954). Over the years, the two strands merged and split at various points. Only in recent years has Reinforcement Learning been used as a practical analytical technique, such as *Simulation-Optimisation* (see Gosavi, 2003), because of the advancement of computer technology. For more details on the history of Reinforcement Learning see Sutton and Barto (1998).

There are various different aspects to Reinforcement Learning, and these aspects determine the different types of RL that are currently being used within the literature. The research presented in this thesis considers some of these types, which are defined below.

Aspects of Reinforcement Learning

The framework that a Reinforcement Learning technique can be directly applied to has certain limitations. For the technique to be of any practical use, the *agent* (or *player*) must have a finite set of actions which to choose from and which results in two things. The first is change in the *state* (i.e. the environment). These states can be terminal (i.e. the game finishes) or non-terminal (i.e. the agent must choose another action). The other effect from choosing an action is a *reward* obtained by the agent. The sum of all the rewards obtained after choosing an action and before reaching a terminal state is called the *return*.

If either the choice of action by an agent or reward obtained from an action is non-deterministic then the process is a *Markov Decision Process*(MDP) (Bellman, 1957) (or a *stochastic* games in the multi-agent case). The MDP forms the underlying framework for which Reinforcement Learning can be applied.

The RL mechanisms work by updating the way that an agent selects their actions (called their *policy*) by considering what returns were obtained from the different action selections. The main way that this is done is by updating various values with the rewards observed in a single play of the game (called an *episode*). These values are either associated with each state (value-based updating) or each action pair (action-based updating) that was visited/chosen within an episode. The research

has been based around the updating of values associated with each action, which are called *Q-values*[17].

The Q-values are an estimate of the expected return that a player will receive from choosing a particular action. Thus a return-maximising player (which are players are assumed to be) would choose the action with the highest expected return at each state. However, the Q-values are only estimates[18] of this expected return, which is where RL comes in. Each time an episode is completed the player has new information about the possible returns from the actions that were chosen in that episode, hence can use that information to update the action's Q-values. Using the mechanism of Reinforcement Learning repeatedly, the player gains better estimates of the Q-values. This is called the *prediction* problem.

The initial Q-values that player has can be worked out through a number of ways. This could be that they are assigned values randomly or use some form of prior or heuristic knowledge.

Once the player has better estimates of their Q-values, they are likely to want to change their policy to reflect this[19]. This updating in policy is known as the *Control* problem. From repeatedly updating their policy from observation, a player could eventually find the *optimal*[20] policy for the game. π is used to represent a generic policy and π^* represents the optimal policy.

However, the Q-values are not static for different policies because an action's expected return is dependent on the actions selected later on within a episode which, in turn, is dependent on the current policy. Thus the expected returns associated with on policy will be different to the expected returns of another policy. The expected returns from using the optimal policy is represented as Q^*-values. Thus when updating the Q-values of an action, the player is trying to converge on the Q^*-values.

Figure 2.1 shows this interaction of the two problems. By continually updating the Q-values and thus the policy (which is known as policy iteration), the player hopes

[17] Originally from Shannon (1950) from their work on chess but was not called Q-value until Watkin's Q-learning algorithm (Watkins, 1989).

[18] If they were not estimates then there is nothing for the player to learn.

[19] A player does not have to update their policy after every Q-value update though.

[20] Or Nash Equilibrium policy when there are multiple agents.

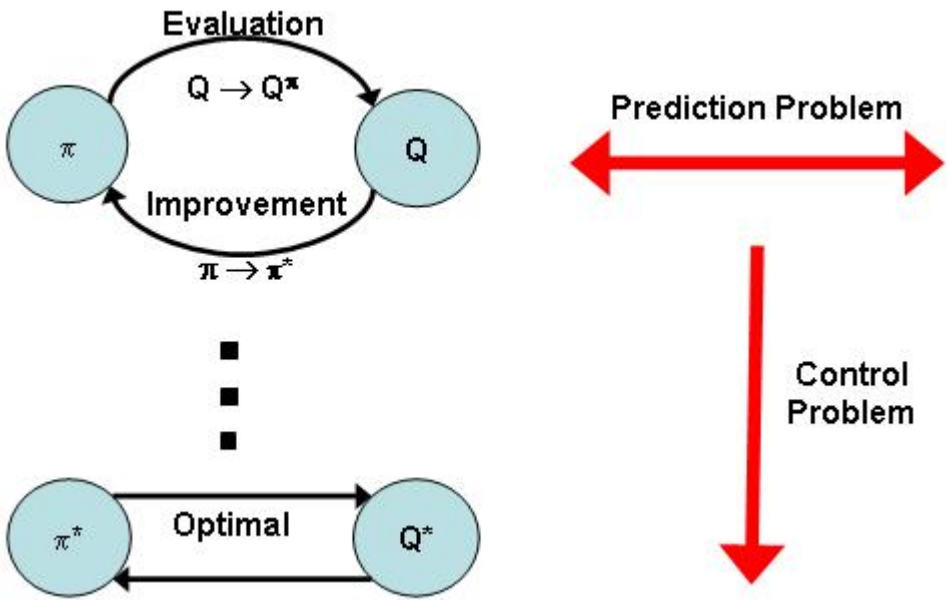

Figure 2.1: Diagram illustrating problems associated with policy iteration

to converge on the optimal policy and learning is complete. Though the problems of evaluating the Q-values and improving the policy may seem complex, there is still yet another problem to deal with.

Action Selection

An optimal policy will tell the player to play *greedily* that is to always play the action which has the highest return at any state (thus the player is *exploiting* their current knowledge). However, the player's estimate of the highest return is based on the Q-values of the actions. Thus as none of the non-greedy actions are selected (and thus their Q-values are not updated), the player's policy could get stuck in a local maximum. Therefore, there is a need for the player to *explore* the returns gained from non-greedy action to gain an better estimate of their Q-values even though they would expect to receive a lower return. This is known as the *exploitation/exploration* problem.

The method by which the actions are selected determines the exploration. The Q-values and the action selection method uniquely determine the policy of a player. Under greedy action selection, there is no exploration. Another method would be to select a non-greedy action for a small fraction of episodes (say ϵ of the time), this is

known as ϵ-greedy action selection. The method that was used within this thesis is the Boltzmann action selection, which was discussed in section 2.2. This method was chosen because of its relationship with the Nash Distribution. An advantage of Boltzmann action selection is that every action as a non-zero probability of being played and when the Q-values are bounded then every action is chosen infinitely offered. This means that Boltzmann Action selection will lead to a complete exploration of the state space (eventually).

The effects from exploration can have an impact on the learnt policy and there are two ways that this can be dealt with: *on-policy* and *off-policy* control. On-policy control is when the learnt policy takes into account the effects from exploration; examples include the Monte Carlo and SARSA RL methods (which is considered below). In off-policy control, the learnt policy ignores the effects from exploration, examples include Q-learning.

The amount of exploration that occurs is controlled by the *temperature* parameter (i.e. Tau or τ). For high temperatures there is a lot of exploration and for low temperatures there is little. For reasons of convergence, the temperature parameter remains fixed over all the episodes (which is called a *run*). However, to ensure that policy begins to converge, the rate at which the player *learns* (i.e. the amount of possible change to a player's Q-values) decreases as the number of episodes increases. This is controlled by the changing learning parameter called the *step-size* parameter (represented by λ_e were e is the number of episodes that has been played).

If step-size parameters decrease too quickly, the player's policy can become stuck in a local maximum. However, if it does not decrease fast enough, the policy can fail to converge at all. The standard restrictions placed on the step-size parameter are (see Sutton and Barto, 1998)[21]:

$$\sum_{e=1}^{\infty} \lambda_e = \infty \quad \sum_{e=1}^{\infty} \lambda_e^2 < \infty$$

A Reinforcement Learning mechanism is determined by these different aspects (i.e. action selection method, temperature, etc.) and the means by which the Q-values are updated (examples are given below). A final point about the whole process is that

[21]Notice that $\lambda_e = \frac{1}{e+C}$, where C ¿ 0, satisfies this condition.

when there is more than one player, the rewards observed will be constantly changing and thus a policy's Q-values will also be changing. This dynamic environment can make learning quite difficult (i.e lots of episodes are required in a run) and can lead to complications in the way that RL is applied. A brief discussion about previous applications of RL is given next.

Examples of RL Usage

RL has had different applications in various contexts, some have been highlighted by Sutton and Barto (1998); Kaelbling et al. (1996). There has been a limited use of RL within an Operational Research context. Examples include Ravulapati et al. (2004) application to business games and Das et al. (1999) application to decision problems. Gosavi (2003) has made attempts to standardise the use of RL within an OR context (called *simulation-optimisation*). Another attempt at giving practical advice can be found towards the end of Kott and McEneaney (2007).

The reason that RL has not been taken up within the OR community is not to do with a limited scope of its application, in fact, there seems to be an abundance of possible applications (i.e. within agent-based simulation as seen in (Hill et al., 2006)). The reason that RL has not been taken up as a main-stream technique is due to its limitations. These limitations are discussed throughout this thesis and several limitations are addressed as part of the conclusions of the research.

Types of Reinforcement Learning

There are various different forms of Reinforcement Learning within the literature (see Kaelbling et al., 1996). Sutton and Barto (1998) presents three basic types of RL. Within the research presented here, the focus is on these types, they are: Monte Carlo method, Q-learning and SARSA method. Before an explanation about the difference of these methods is given, some terms must be introduced.

A player within a game will, at any time, have a set of possible actions that they can undertake. This set of actions is defined as A with an action $a \in A$. In this implementation of RL, it is required that $|A| < \infty$. A player will have an estimate of what each action is worth to them, which is called *Q-value*, represented by $Q(a) \in \mathbb{R}$ for a particular action $a \in A$. Updating these Q-values, from the observed rewards from

repeated play of the game, forms the basis of RL. The rate at which this updating occurs depends on the current number of plays of the game (called *episodes*, represented by $e \in \mathbb{N}$) and is determined by the step-size parameter, which is represented by $\lambda_e \in (0, 1)$. The immediate reward observed, as a consequence of the action selection, can be stochastic or deterministic and is represented by $r_e(a)$. The return observed from an action (i.e. the summation of all rewards gained *after* the action was selected) is given by $R_e(a)$.

The simplest form of RL is when the player has to only make one action selection and can been represent as follows [22]. Given that action $a \in A$ was selected in the episode $e + 1$, its Q-value is updated as follows:

$$Q_{e+1}(a) = (1 - \lambda_{e+1}).Q_e(a) + \lambda_{e+1}.r_{e+1}(a) \tag{2.2}$$

$Q_0(a)$ is defined as the player's initial expected-reward estimation from playing action a. This basic equation looks similar to the *exponential smoothing* forecasting technique (see Brown and Meyer, 1961). In more sophisticated games, a player would have to select actions at different points within the game.

When a player has to select actions at different points within a game, these points are called states s, which belong to a state space S (i.e. $s \in S$). Thus *given that state s is visited* in episode e, the actions available are $A(s)$, the action chosen was $a_e(s) \in A(s)$. Lambda is also dependent on the state and is represented by $\lambda_e(s)$.

Monte Carlo Method

Monte carlo learning is the simplest form of learning that is considered within this research and is one of the earliest forms of RL to be used (see Michie and Chambers, 1968). Its learning mechanism relates to the one in equation (2.2) but return is considered instead of reward (as multiple stages have to be considered). Given all the actions that were visited in episode $e + 1$, their Q-values are updated as follows:

$$Q_{e+1}(a_{e+1}(s)) = (1 - \lambda_{e+1}(s)).Q_e(a_{e+1}(s)) + \lambda_{e+1}(s).R_{e+1}(a_{e+1}(s))$$

As the updating only considers the actual observed returns it would, at first glance, seem a reasonable way to proceed. However, due to the stochastic nature of a game

[22] This representation uses Q-values, which is consistent with this implementation of RL. Other representations do exist (i.e. value based approach) and can be found in Sutton and Barto (1998).

(both action selection and rewards observed), using the actual observed returns can lead to random anomalies having a great impact on the updated Q-values.

Q-learning

To get around the problem of random anomalies having a large impact in the updating of the Q-values, the complete return can be ignored. Instead the immediate reward and next states Q-values are used in the updating. Of course, the next states Q-value is an estimate itself, hence estimates are being used to update estimates (this is called *bootstrapping*). Q-learning is an example of this and was first introduced by Watkins (1989). If s' is the next state that is visited by the player, and its action set is $A(s')$ then the updating formula for Q-learning is:

$$Q_{e+1}(a_{e+1}(s)) = (1 - \lambda_{e+1}(s)).Q_e(a_{e+1}(s)) + \lambda_{e+1}(s).\,(r_{e+1}(a_{e+1}(s)) + max_{b(s')\in A(s')}\left(Q_{e+1}(b(s'))\right))$$

If there is no next state then no extra Q-value is used in the updating. Q-learning is an example of off-policy updating since it ignores the subsequent action that was played and updates using the Q-value of the greedy action instead. As Boltzmann Action Selection will select the greedy action for the majority of the time anyway, the subsequent action is usually the greedy action anyway. By updating this way, the effects from exploration are ignored.

SARSA

The use of bootstrapping in Q-learning makes it a form of *Temporal-Difference* learning. An on-policy example of Temporal-Difference learning is the SARSA method. SARSA method was originally proposed by Rummery and Niranjan (1994), who called it *modified Q-learning*. Unlike Q-learning, SARSA does not use the greedy action in its updating but the actual action observed. If s is the next state that is visited by the player, and its action $a(s') \in A(s')$ is selected, then updating is done by:

$$Q_{e+1}(a_{e+1}(s)) = (1-\lambda_{e+1}(s)).Q_e(a_{e+1}(s))+\lambda_{e+1}(s).\left(r_{e+1}(a_{e+1}(s')) + Q_{e+1}(a_{e+1}(s'))\right)$$

The name SARSA is derived from the sequence of events that are used in the updating namely: State, Action, Reward, next State, next Action. The SARSA method

has not been used much within the RL literature and there are very few papers which consider multi-agent SARSA learning (i.e. Banerjee et al. (2004)). Multi-agent learning is the context of this research.

There have been some attempts to give guidelines of both Q-learning and SARSA multi-agent learning, which can be found in Takadama and Fujita (2005). In this paper, Takadama and Fujita suggest that both techniques should be used within any gaming context to validate the results. The paper suggests that the SARSA method is more risk-adverse than Q-learning (which makes sense since Q-learning ignores all non-greedy Q-values). Finally, their paper points out a single agent learning is faster than multiple agents learn (which is unsurprising since less has to be learnt).

Other RL techniques do exist which are briefly discussed now. This is no means a complete list of possible techniques and the literature on new techniques is continually expanding. For example, R-learning is a variation on Q-learning but has an extra quantity ρ within its updating mechanism, were ρ is also updated like the Q-values. It has been compared to the SARSA and Q-learning method in Ishikawa et al. (2007), though the results were not conclusive.

There have been attempts to use Boltzmann Action Selection within other RL mechanisms. This is seen in Camerer and Ho (1999), where they use a simple form of Reinforcement Learning called *Experience-Weighted Attraction* learning. This method is designed to ensure that the player's initial attractions are always considered within their choice of an action.

Eligibility Traces is an extension to the Temporal-Difference learning, which takes into account of more than just the next state. An example of an Eligibility Trace variation is SARSA(λ) which was first explored in Rummery and Niranjan (1994); Rummery (1995). Though there are benefits to using Eligibility Traces, this research has focussed only on the simple cases.

Another form of Reinforcement Learning is *Cumulative Proportional Reinforcement* (CPR) which can be found in Laslier et al. (2001). An extensive form version was adapted in Laslier and Walliser (2005). Extensive-form games form the focus of the research presented in this thesis and an example can be seen in figure 4.2. The ground-

breaking research in learning in extensive-form games can be found in Fudenberg and Kreps (1994, 1995). Experiments of learning in extensive form games have been conducted by Roth and Erev (1995) which is an example of the connection of RL within psychology.

Psychology

One of the founding strands of RL was within psychology which started with the work on animal intelligence by Thorndike (1911). However, Reinforcement Learning was grounded within psychology ever since Pavlov's famous experiment with dogs (Pavlov, 1927). At present Reinforcement Learning is still considered one of the two main possibilities for how animals learn behaviours, the other being associative learning (Leslie, 2001).

The psychologist use Reinforcement Learning in a more sophisticated way to that seen within this thesis. Instead of dealing with millions of repeated plays of the game (i.e. episodes), the psychological literature deals only with a few within their experiments. The reason for this difference is twofold. Firstly, psychologist experiments only involve actual players and not simulated ones so time to play a game becomes an issue. Secondly, the psychological experiments consider, in depth, the impact of each play experience.

Within the Rescorla-Wagner model (see Rescorla and Wagner, 1972) of psychological RL various advanced aspects are considered. For example, problems like *backward blocking* occur when actual animals are learning [23]. Within the learning results considered for the research presented in this thesis, this level of detail is not explored.

Experiments comparing the results from thousands of episodes have been conducted. For example, Erev and Roth (1998) constructed an RL method and compared their results to human players (their mechanism was shown to converge by Beggs (2005)). Both Valluri (2006) and Prasnikar and Roth (1992) did experiments using sequential games. However, the experiments of Chen and Khoroshilov (2003) indicated that

[23] Backward blocking is where an extreme results is experienced early on in the learning process which has a huge impact on all subsequent actions. See Kruschke and Blair (2000) for more details.

RL does not explain human behaviour [24]. This is not surprising, within a game context, given our previous discussion on *Homo Economicus*. However, whether RL explains human behaviour is only one aspect for consideration, another is whether the RL techniques actually learn.

Convergence Proofs

There is a vast quantity of literature on convergence of Reinforcement Learning techniques, here a small sample is presented. The focus of this sample is on convergence of temporal difference mechanisms within games.

Convergence results have been shown for the single player case of Temporal differences learning see (Dayan and Sejnowski, 1994), though this only consider case when fixed transition probabilities (which cannot be translated into the multi-player case). Singh et al. (2000) proved convergence for the single player SARSA method and Banerjee et al. (2004) proved convergence for the multiple player case but both proofs require certain restriction on the players.

Several different proofs are available for non-general versions of single player Q-learning, the first being Watkins and Dayan (1992). In recent years, Leslie and Collins (2003, 2005, 2006) have looked at convergence for the multi-player case, with special interest of when difference learning rates have been used by the different players.

An issue that has arisen within multi-agent games is when *uncoupled* learning dynamics [25] are used. Hart and Mas-Colell (2006, 2003) have shown that uncoupled learning dynamics cannot be guaranteed to converge. However, this does not stop any investigation into explaining how people play when they are unaware they are in a game (Leslie and Collins, 2005). Further discussion about this issue is found in Chapter Six.

Stochastic Approximation

The main mathematical method that all these convergence proofs have used is *Stochastic Approximation*, which was introduced by Robbins and Monro (1951). The sim-

[24] This work was not conclusive and RL to achieved mixed results in Feltovich (2000). Feltovich found that RL was good at predicting how human players learn if good *heuristic knowledge* was used.

[25] An uncoupled learning mechanism is where players do not take account of the opponent's reward function or policies.

plest stochastic reinforcement learning (see equation 2.2) model was solved for the one player case using the Robbins-Monro method. A good introduction to the subject can be found in Kushner and Yin (2003)[26]

A more advanced version of the Robbins-Monro method can be found in Dvoretzky (1956). A simplified version of the Dvoretzky proof is found in Wolfowitz (1956). New ideas on Stochastic approximation can be found in Benaim (1996, 1999). Benaim has done some work on games and learning in (Benaim and Hirsch, 1999), though the focus is on fictitious play.

This end discussion on the first two academic fields considered in the research within this thesis, namely: Game Theory and Reinforcement Learning. The third, and final, field is Revenue Management and is discussed briefly in the next section.

2.5 Revenue Management

Revenue Management (also known as Yield management[27]) is claimed to be one of the most successful application areas of Operational Research (see Talluri and van Ryzin, 2004). Revenue Management (RM) is concerned with demand-management decisions especially when in relation to *pricing*. Talluri and van Ryzin's book *The Theory and Practice of Revenue Management* forms a comprehensive introduction to the subject.

RM is a relatively modern subject and has its roots in the airline industry. A need for RM was found and addressed in 1978 when the United States of America allowed airlines more flexibility with their airline seat prices. Though there are other applications of RM, this literature focuses on this, airline seat pricing. A brief overview of OR and airline industry is given in Ahmed and Poojari (2008)[28].
There are two aspects to airline seat pricing: dealing with competition and properly forecasting demand. Modelling competition is done via Game Theory, where as forecast demand is done via a variety of techniques (see Luce, 1959; Talluri and van

[26]Though the original work by Robbins and Monro (1951) is surprisingly easy to read and also a good introduction.

[27]Yield Management term is used in airline industry, where as Revenue Management is used elsewhere.

[28]The focus of Ahmed and Poojari (2008) is on optimization techniques

Ryzin, 2004). Though there has been much research into each aspect individually, as noted by (Boyd, 2007), there has been difficulty marrying the techniques. Dealing with this issue is part of the research that is presented in this thesis. The remainder of this review looks at the current use of pricing, learning and games.

Pricing, Learning and Games

A recent example of modelling airline pricing is given in Anjos et al. (2004, 2005), which was a simple continuous one and solved using Nelder-Mead method. This model was then extended to include competition in Currie et al. (2006). Another example of games and Airlines within a OR context can be found in Schipper et al. (2007).
In terms of pricing and learning, Gosavi et al. (2002, 2007) have applied Reinforcement Learning to airline pricing, though they are look at the single agent case. Their work, however, does include complications of over-booking and cancellations. Stochastic approximation has been directly applied to the airline industry in van Ryzin and McGill (2000) . In their paper, van Ryzin and McGill apply a simple stochastic approximation model to determine the seat protection levels of an airline. They even show optimality within the constraints of their simple model.

For work which includes all three aspects (pricing, learning and games) there is limited literature available. Q-learning has been applied to a pricing environment in Sridharan and Tesauro (2000), which was extended in Tesauro and Kephart (2002). Other examples include Chinthalapati et al. (2006)for electronic retail markets and Kőnőnen (2006), within an asymmetric learning environment. No literature can be found on pricing within a learning sequential game context, which forms the basis of the research within this thesis.

2.6 Summary

This chapter has discussed the use of Game Theory within an Operational Research context and the limitations associated with it. Important concepts like the Nash Equilibrium and Nash Distribution were discussed. This lead onto the use of Artifical Intelligence as a means to solve games with a special interest in Reinforcement Learning. Of Reinforcement Learning, the different aspects of the technique were discussed. Three different examples of Reinforcement Learning were introduced, namely:

Monte Carlo method, Q-learning, and SARSA method. Finally this chapter gave a brief introduction into Revenue Management and examples of its use.

Chapter 3

Methodology

3.1 Introduction

As outlined in Chapter One, the thesis intends to investigate the practical use of Reinforcement Learning to solve Game Theoretic models within an OR context. An OR context means that the models would be used to underpin the way decision makers operate in a real-life contest. To undertake this task a clear plan or methodology is required. This chapter considers the methodology that was formulated before any empirical or theoretical results were found.

Aim

Within this chapter, it is intended that a coherent methodology is presented for the reader to follow. This methodology was designed to research the question posed within chapter one. As the question is quite general, outlined here is how this was narrowed down. Any underlying assumptions that have been made within the research are also highlighted.

Analysing quantitative results is the main focus of this research. This does mean that some important considerations of using Reinforcement Learning within a Game Theoretic context are ignored. For example, the ease with which an OR practitioner can implement the techniques or how easy they are to validate the model has not been considered.

Though this does seem like a major loss when considering implication of a technique, it does mean that the technical detail of the implementation can be focused on (i.e. method of Reinforcement Learning used, which parameter should be used, etc.). Also, there is a large body of literature available which covers the practical aspects of technique implementation (Ward, 1989; Pidd, 1996; Bryman and Bell, 2003; Chick, 2006).

There are several examples within the literature of implementing Reinforcement Learning within a pricing games context (i.e. Sridharan and Tesauro, 2000; Tesauro and Kephart, 2002; Chinthalapati et al., 2006; Kőnőnen, 2006). However, all the examples are from a computer science theoretic perspective and are not concerned with the implementation of their methods or models within a practical context. Therefore, without prior research within this context it was difficult to hypothesis what the outcomes would be. Thus the research was conducted in a inductive way, using the following steps:

1. Data collection starts with no initial theory
2. Tentative theory developed from early data and then tested against later data
3. Aim is to identify core concept explaining behaviour

This seems likes a reason way to proceed with the methodology and will form the basis of the research. Firstly data needed to be collected. As the intention is to look at the practical usage of RL within a GT context, collecting data from the modelling of practical problems is required.

Though it would be possible to construct a qualitative model of this situation [1] this research focused on quantitative modelling and only one model output was investigated. This assumption does undermine the generalisation of the results but it does give richness to the analysis which would otherwise be impossible to achieve.

Deciding on which problem to focus on is challenging. However, it is apparent that there is a current need to marry the techniques of customer behaviour modelling and

[1] A 'soft' GT technique could be used (i.e. Drama Theory Howard (2001)) with psychological RL experiments to achieve this.

a competitive revenue management (see Boyd, 2007; Currie et al., 2006). This problem has been highlighted in the literature review chapter. Therefore, this seems to be a problem that can be tackled within this context.

Therefore, a simple airline pricing model is constructed and the optimal solution is found to this simple model using traditional techniques. Then these results are compared to the results obtained using Reinforcement Learning methods. The model was made more complex so that the traditional techniques cannot reasonably be used and interpret the RL results from this.

It has been suggested that factors like seasonality make the airline pricing market to erratic to study. However, it can be argued that price competition always remains a factor in the market and thus can be studied.

Validation and Verification

One criticism of the methodology is that it is impossible to collect data without having an initial theory. This introduces bias to the results which might invalid any conclusions. To overcome this, the problem was tackled from a different angle to give a triangulated approach.

As well as considering the empirical results from these models, it would seem correct to consider the theoretical side as well. Therefore, if the empirical question is "What results are observed in practice?", then the theoretical question would be "Does the learning converge in theory?". By solving these two questions, the problem is approached from two different angles. These different angles allow for more valid conclusions about any emergent behaviour that appears from results.

Overview of Methodology

Different aspects of the methodology are brought together here. A summary is given below:

1. Construct airline pricing model
2. Find solutions using Dynamic Programming
3. Run model to generate the empirical results

4. Compare different RL techniques
5. Conclude which RL technique are most effective
6. Prove theoretical convergence for that technique

The remainder of this chapter will consider each of these methodology aspects in turn. Each section gives an introduction to the aspect and an overview of the methodology and assumption required.

More detail about each of these aspects can be found in the remaining chapters of this thesis. Items one and two are discussed in the Model chapter. Items three to five are investigated in the Empirical Results chapter. Finally, item six remains separated from the other aspects and is discussed in the Convergence Proof chapter.

3.2 Constructing the Model

Various different games could have been constructed within a number of problem areas looking at the effects of using Reinforcement Learning within a practical context. However, by considering a number of games only a superficial analysis of each could be given. The implications of using a new technique are considered, therefore this seems inappropriate as a greater understanding of the application is required. Focusing on a single problem allowed an in-depth analysis of the results to be completed. Therefore the focus on a current problem regarding dynamic pricing within the airline industry.

As outlined in the literature review, combining a plausible customer behaviour model within a game is a current problem with Revenue Management (Boyd, 2007). The use of Reinforcement Learning might give insight into the kind of strategies that are used for more complex models, thus going towards a solution to this problem.

The literature review also shows that there are various existing models that are trying to compute the airline-pricing policy under competition. These airline-pricing model frameworks have been developed with a particular solution concept in mind (i.e. calculus of variation in Currie et al. (2006)). Similarly, the constructed model was designed with Reinforcement Learning as the solution concept. This meant that

the model is constrained in several ways (i.e. the need for a finite state space). Throughout this section the various limitations that have to be imposed on the airline-pricing model are considered so that the learning game will work and the hypothesis can be tested.

Before moving onto a discussion about these limitations and choices, the other aspects that are modelled need to be considered. As well as the airline-pricing model there is the learning model, where the Reinforcement Learning takes place. The implementation of the various RL methods was non-trivial and many decisions were needed to be made about its construction. The following subsections consider these two models in turn.

Framework of Airline-Pricing Model

The aim of the research is to show that RL gives the ability gain good approximate solutions to a complex unsolvable problem, however a game framework that is solvable in the classic sense (i.e. Nash Equilibrium) is still needed so that there is something to compare the RL results with. Therefore, an airline-pricing model was required with a game solution simple enough to have a readily available solution for comparison.

There was still a requirement to observe the results from variations on the model when quite complex behaviours of the customers was used. The solution to this problem was to develop the customer-demand model separately from the main airline-pricing model. By not embedding the customer-demand model, a complex or simple model could be produced depending on the requirements. This customer model is discussed below. There is still a requirement to understand the basis of how the airline-pricing model will interact with this customer model. Therefore, what variables the model will require needs to be considered first.

Decision Variables

The decision variable in this airline-pricing problem is how to maximise revenue by changing seat prices in relation to the market. The airline (or players) decision to change the current price of a ticket will depend on various factors, the main factors (see Talluri and van Ryzin, 2004) that will affect their decision are:

- Number of seats left on each aeroplane
- Time left to flight departure
- Competitors' current price
- Historical information
- Own current price
- Market size

There are many other factors that could be considered (i.e. Seasonality, Global Events, etc.). The more factors that are considered, the greater the state-space and the greater the computing requirements (i.e. run-time and memory requirements). Thus it is preferable to limit the information the decision maker uses for their decisions.

Limiting information is good from a computer modelling perspective, however the usage of RL within a *practical* context was being considered thus the model must still contain enough complexity to be validated as an airline-pricing model. The need to only model the factors that are important is a valid modelling approach by the law of Parsimony (or Occam's Razor[2]). This philosophical approach has been incorporated within the OR Literature through works such as Ward (1989).

To decide which factors are most important, the current OR literature on modelling this problem was considered (i.e. Gosavi et al., 2002, 2007; Currie et al., 2006). The factors that were highlighted from the literature were:

- Competitors' current price
- Time left till flight departure
- Number of seats left on each airplane

[2]This has been stated in many forms, the most common being *All things being equal, the simplest solution is the best* or *Entities should not be multiplied beyond necessity.* Though credit is given to English philosopher William of Ockham (1288 - 1347), its first appearance was in the work of the Irish mathematician Sir William Rowan Hamilton (Hamilton, 1852).

These factors determine a state within the airline-pricing model. Not using the historical information implies a 'lack of memory' by the learning players. However, the historical information is already taken account of within the RL model (by the nature of the method). This is because the Reinforcement Learning process uses historical information to update the policy. Historical information can be misleading to a play, as both players are learning within the model; hence the returns from any policy will be constantly changing.

Through most of the research, the number of seats on the plane does not affect any of the results obtained and could be ignored. However, the number of seats remaining has been included in the state space of the model to give the space a realistic size.

Another missing factor is the number of seats left on an opponent's plane. This reflects the reality that the players will not know what their opponent has sold and thus the game is one of *Incomplete Information*.

Modelling Limitations

Again following almost all of the literature on competition, only two players are considered within the model. The final model (see Model or Convergence Proof chapters) is not constrained to look at only two players but this limit has been imposed on this analysis.

The airline-pricing model is embedded within the learning model and there are certain constraints that must be adhered to. For there to be any chance of convergence of the RL algorithms, a finite number of states had to be used. This means that all the decision variables had to be finite. Obviously, this is true for the number of seats and competitor's price [3]. Time is not a discrete dimension but it is reasonable to assume that there are only a finite number of times that a player can change their price before the aeroplane leaves (as it requires time to process the information about a player's current state) and therefore time can be seen as discrete.

Given that all the decision variables are finite and discrete, there are a finite number of states. The exact range that each decision variables takes are outlined in the

[3] Prices can only go up in discrete steps (i.e. 1p) and are limited to the world wealth ($37.1 trillion according to the Merrill Lynch/Capgemini World Wealth Report 2007 at www.capgemini.com/industries/financial/solutions/wealth/worldwealthreport/).

Model chapter. Reinforcement Learning not only requires that there are a finite number of states for convergence to occur but also that there are a finite number of actions. As the actions are represented by a choice of price, from the arguments above, this is also finite choice of discrete actions.

A consequence of having a finite set of prices is that there is a finite set of payoffs (as there is a finite number of customers at any one point). There were no constraints on whether the Airline model was stochastic or deterministic, this feature is decided within the customer model. These two features mean the game is stochastic, which is the two-player version of a Markov Decision Process (MDP) (see Bellman, 1957).

There are several modelling variations in the literature that are not included for brevity. For example, overbooking or cancellation are not explicitly modeled (see Gosavi et al., 2002, for an example of this). It would be possible to included these factors within the framework, however they are not explicitly represented [4].

As seen in a lot of the literature, a single-leg flight is being dealt with. This avoids the added complication of dealing with an airline network or having to model return policies. It is appropriate to study only single-leg flights, this is highly unrealistic in a practical context as few flight ticket purchases are one-way (Talluri and van Ryzin, 2004). This does limit the application aspect of this research. Most of the literature on Revenue Management only considers one leg prices.

For similar reasons to above, only one seat-class was considered in the model. This implies that the airlines are only attracting customers of a certain type (i.e. business class). A simple way to allow for seat class distinctions is to assume that the different prices offered are for the different classes. However, this has not been modelled as it is reasonable to expect all available classes to be offered simultaneously.

Sequential Move

Probably the most controversial decision for the airline-pricing model was using sequential move, as opposed to simultaneous moves. In this section so far, the model

[4] Cancellations can be represented by allowing negative figures to be output from the customer model and overbooked seats can be adjusted for, within the reward, at the end of play of the game.

has been based on the current literature. Most of the literature relating to airline-pricing uses simultaneous move games (see Schipper et al., 2007; Currie et al., 2006). The use of simultaneous moves has been criticised however (see Eatwell et al., 1987; Binmore, 1990) as being unrealistic and not necessary the only approach to modelling with Game Theory.

Sequential moves were included for several reasons. Firstly, it was important that the players were able to respond to each other's price. Secondly, a sequential moves equilibrium solution is easier to follow and therefore more transparent to understand. Thirdly, sequential moves would reflect how airlines respond to each other's price in practice as it would be impossible for airlines to simultaneously change their prices without some kind of coordination between the organisations. Finally, the dynamics of the strategy is faster in sequential rather than simultaneous games.

Within the literature there are several examples of sequential games experiments (Prasnikar and Roth, 1992). Some even considered Reinforcement Learning (Valluri, 2006; Erev and Roth, 1998). These experiments are based around actual human players and were not connected to airline pricing. There has been no work on learning within sequential airline games that could be found.

Time-steps

The exact form of sequential move that is being used is given in figure 3.1. As the diagram shows, the customers have the opportunity to arrive between the two players' price changes. Thus there will be at least two opportunities for customers to arrive between a customer's price change. This leads onto the question of which time-frame do these steps represent? The arrival rate of the customers can be changed within the customer model to represent any period, therefore this could represent any time-frame. As airlines are likely to check the prices daily, regular updating is expected. The price changes do not occur simultaneously, there is always a chance that a customer will arrive between the price changes.

Each time-step does not have be the same length and could vary depending on how the model is utilised. Therefore, the break-down of the scenario into time-steps was not considered to be a major limitation. The order of the time-steps was limiting

Figure 3.1: Mechanism of sequential game

however. For instance, it may be appropriate (and even beneficial) for a player to be able to make a limited number of price changes, thus part of their policy could be to choose the number opportunities to update their price. This meta-game [5] and others are considered in the Further Results chapter.

One criticism of the sequential time-step is that the players cannot change the price available every time a seat is bought. Within the framework this could be overcome by allowing a maximum of one customer to arrive per customer time-step and by not letting any customers arrive in a customer time-step if one came the previous time-step. This means that each player would get a chance to change their price after each seat sold. In its present form, each customer's time-steps are independent of each other.

The price changes are sequential, therefore there has to be a player who is first to choose their price. This could result in the situation where one player is a leader and the other a follower[6]. As the Model chapter states, being the leader can be advantageous. This may mean that players would compete to place their price first. This can be considered as another meta-game and is briefly discussed in the Further Results chapter. For the purpose of the framework, player one (P1) will be the first player.

[5] e.g. a game where the payoffs are also a game (thus the payoffs are a solution of these sub-games). This is not the traditional use of meta-game (Thomas, 1984) and is just a sub-form of a stochastic game.

[6] This can lead to a type of *Stackelberg* equilibrium (von Stackelberg, 1934). See Fudenberg and Tirole (1991) for a brief introduction.

Previous Learning Games

As an inductive approach was used for this research, so several different airline-pricing models were considered. These are discussed in detail in the Further Results chapter. The current airline-pricing model was derived from these prototypes. Through the evaluation of these models, it become clear that one factor was most important. This factor was memory requirements of the model. Several seemingly simple models are impossible to implement when translating onto a computer program. This problem is sometime called the *Curse of Modelling* (see Gosavi, 2003). Through the remainder of this thesis, many references to this problem are encountered.

Summary of Airline-Pricing Game

The use of an airline pricing model has been discussed as the underlying model to compare the different Reinforcement Learning methods. An airline pricing game framework has been derived with the following characteristics:

- Two airlines
- Identical single-leg flights
- No overbooking or cancellations
- Attraction of customers through dynamic pricing
- Sequential moves between players and customers
- Strict ordering sequence
- Finite interval prices
- States are determined by current prices, seats remaining and time to departure
- Separate customer arrival and preference model

This section gives an overview of the airline pricing model. Details of the framework used and its characteristics, are discussed in the Model chapter.

This sub-section has discussed several limiting and simplified elements of the model. This has not prevented the model from producing interesting and deep results, which

can be found in the Empirical Results and Model chapters. These rich results demonstrate the need for parsimony that was employed, as complex model results could have been too difficult to interpret.

Framework of Customer Model

As discussed above, the customer arrival and acceptance models are seperate. The model's input is the current state and its outputs is a set of stochastic values. These outputs are how many seats each airline has left (i.e. seats remaining minus the number of customers that accepted their price in that timeframe) and the reward that players get. These rewards are simply the current players' price multiplied by the number of customers that accepted in that time-step.

The total of all these rewards from the customer models (at the different time-steps) gives a sample return (or payoff) from the players' current policy. This reward is used to update the players' current policy (see the learning model below). As the return observed is affected by the other player's policy (who is also constantly changing their policy) and stochastic elements of the customer model, the return is likely to be different each episode. The output of the customer model is not necessarily known by the players and the RL model ensures that they learn to react to it[7].

It is assumed that the players are unaware of the customer model's behaviour, therefore it was important that the customer model was separate from the other interacting models (i.e. learning and airline-pricing models). This also allowed a high level of complexity in the customer model without having to worry about the impact on other interacting models. As noted by Andrew Boyd (Boyd, 2007), there is a tendency in the RM literature to consider simple customer models within games or complex customer models without games. By having a separate customer demand model, the technique marries up the two approaches within Revenue Management.

Initially a simple customer demand is considered within the experiments (see Empirical Results chapter). The requirement to solve and find all the Nash equilibria (and Nash Distributions) for the analysis was the driving force behind this decision. The

[7]The convergence of the player's policy from RL is based around what returns were observed. The players do not attempt to actually learn how the customers demand seats.

customer model can be stochastic but it remains static (i.e. unchanging and unlearning). This is not important for the convergence results as it is accounted for within the mathematical framework shown within the convergence proofs.

Customers do not take into account the previous prices offered by the airline players (within the current and previous games) as they assuming that the customer model is static. To do so would require the state visited by the players to also contain the previous pricing information thus driving up the state space to an unmanageable size.

This *lack of memory* for the customers can be seen as a different pool of customers arriving at each time period (and episode). However, if this information was used it would mean making the customers players in their own right. This would be desirable but it would make the sophistication and complexity of the game unmanageable. It is not possible to model every element of the real world and the focus is on an airlines choice of policy regarding a single competitor.

Summary of Customer Model

The customer models that were used are discussed in the Model and Further Results chapters. The requirements considered here are those required for a complete model framework. A summary of the customer model is as follows:

- Static and Stochastic
- Players are not aware of the customer demand[8]

The following section is concerned with an overarching learning model.

Framework of Learning Model

Arguably the most important part of the complete model is the learning model, which is also the most complex. To derive this framework the following questions must be answered:

- How is information stored by the players?

[8] A demand learning mechanism could have been used here but instead a reward learning one was used. See (Lazear, 1986; Talluri and van Ryzin, 2004) for more details on demand learning.

- How are prices chosen by the players?
- How do the players learn from observed reward?

Information Storage

The reason the player's learning occurs is because they wish to derive the best possible policy for the game. This is achieved through learning about the effectiveness of their current policy and adapting it accordingly. This could be achieved a number of ways (i.e. ranking a policy each episode). However, this research is concerned with using Reinforcement Learning methods and there is a limited number of ways that the information can be stored.

When using Reinforcement Learning methods, the players are updating their estimates on the expected return from a state or on action. These estimates are used to generate the policy of the players using some type of action selection mechanism. As it was intended that certain types of RL be evaluated (i.e. Q-learning), update estimates on the actions (i.e. Q-values) are required. The information stored is already determined, however a decision on which action selection method to use must be made.

Price Selection

At each state, there will be an action performed by the players or the customer. The players have to choose the action that they will be using. The players use the information that they currently know about the action to perform this selection. This state-action information is called the *Q-value*. There are various different ways that this can be done and these are:

- Greedy
- ϵ-greedy
- Boltzmann Action Selection (or Softmax)

Greedy action selection is always choosing the action that the player currently thinks will give the best reward/return. As the Q-value is an estimate, it is not necessarily

the case that this action will give the best return. In learning games, this is especially true as the opponent's policy will be constantly changing (as they learn) and therefore actions that seemed unfruitful in the past may now produce good returns. It is important to keep an eye on the current estimate for all the actions. This leads onto the *exploration-exploitation* trade-off within games of learning.

If a player had perfect information about the customers and the opponents play, then their Q-values would be correct and it would be appropriate to exploit that knowledge. However, the players do not have perfect information about the game and must spend some effort exploring the state space to improve their knowledge (especially in a changing environment). Without this exploration, it is likely that the players get stuck in local maximum policy. By exploring the possible actions the player will not necessarily gain full advantage (return) of the knowledge they have acquired. This is the *exploration-exploitation* problem described in the literature review.

One strategy that could be used is that the player chooses the current best action most of the time and a random other action now and again. This would mean that all actions would be chosen (eventually) and that the state-space would be explored. This could be achieved by using ϵ-greedy action selection (Watkins, 1989). With this mechanism, an action is played which has maximum Q-value with probability of $1 - \epsilon$ (equally divided amongst those with the maximum Q-value) and the rest of the actions are played ϵ of the time (again, equally divided amongst the remaining actions).

The problem with this approach is that it does not take into account the difference in the other Q-values. One way to get round this is to assign a probability to the ordered rank Q-values (Singh et al., 2000). An example with three actions could be assigning a probability of $\frac{4}{7}$ to the action with the highest Q-value, $\frac{2}{7}$ to the next highest and finally $\frac{1}{7}$ to the action with lowest Q-value. However, this method does not take account relative proportion scales of the Q-values.

One method that does take this into account is *Boltzmann action selection*, which was introduced in Chapter One. This method assigns probabilities to the selection of the different actions at a state by their current value (i.e. the higher the value, the more chance they will be selected). The method also gives a positive probability of occurrence to every action, hence a good exploration of the state-space is ensured.

The greedy action selection is the one associated with any Nash Equilibrium policy, however, without exploration it could not be guaranteed that this had been reached (because the Q-value estimates would be incorrect). Given the relationship with Boltzmann action, the variation on the *Nash Distribution* policy (see the Literature Review chapter) was the mechanism that was used.

The major concern with using the Boltzmann action selection method is that too much exploration can lead to a Nash Distribution policy that is dissimilar to the Nash Equilibrium policy. This is discussed is the Model chapter.

Learning Mechanisms

So far the underlying airline pricing game (and the customer model) has been described as well as the means in which the information is stored for learning and the action selection mechanism. This only leaves the way in which Reinforcement Learning is used to learn the policies. These learning methods are the reason for developing the model in the first place and it is intended that an evaluation of their usage is conducted. The learning mechanisms considered for evaluation are: SARSA, Q-learning and Monte Carlo methods.

These (with others) were described in the Literature Review section. They were chosen as they form the basis for most Reinforcement Learning types, as outlined in Richard Sutton and Andrew Barto's book *Reinforcement Learning: An Introduction* (Sutton and Barto, 1998).

To discover what happens when these learning mechanisms are used, the players must be allowed to learn. All the mechanisms learn from the outcomes of a game, hence by repeatly playing the game it is seen how the learning mechanisms have changed the policies. The learning mechanism would be expected to converge onto a single policy, though it is possible that they might diverge. This is discussed later in the chapter.

Summary of Learning Model

The learning model can be summarised as follows:

- Three different RL methods considered: Q-learning, Monte Carlo learning and SARSA

- Use of Q-values to store state-action information
- Boltzmann action selection to ensure players explore the state space

In this section, the framework of the model used for comparing the RL methods has been discussed. Before discussion is moved onto the methodology to compare these methods, there is a requirement to decide which benchmark should be used for comparing the learnt policies. This involved a variety of other policies (including the Nash Distribution) and deriving them was a non-trivial task. They are discussed in the next section.

3.3 Find Solutions using Dynamic Programming

The purpose of this research is to investigate how good the different RL methods are. This could be done simply by comparing the results of each of the RL runs. If only a single-player scenario was considered then this would be sufficent as the goal would only be to find the method that gave the highest return (or reached the highest return the fastest). However, the problem considered deals with a two-player game.

Within a game, the higher the reward observed does not necessarily mean the better the policy. A high reward is dependent not only on the player's policy but also on the player's opponent's policy. This is demonstrated in the extensive form game shown in figure 3.2. The bracket pair at the end of the paths represents the P1 and P2 reward respectively.

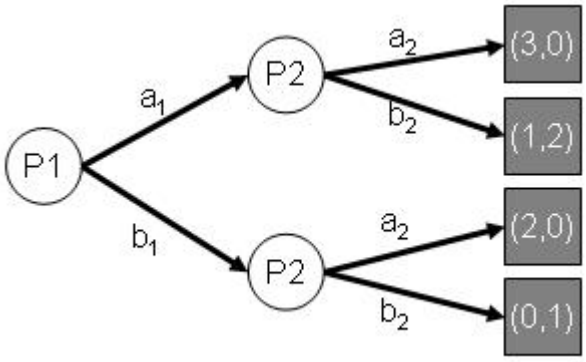

Figure 3.2: Extensive-form game example

Lets consider some possible outcomes from this sequential game. If P1 policy is play action a_1, then the reward they observe will either be *three* or *one*, depending on policy of P2. This means that though P1 has a fixed policy, different rewards are observed depending on the P2 policy. P1's policy could be compared to every possible P2 policy. Not only would this be impossible (when mixed strategies are allowed), it would be meaningless as P1's policy would be compared to a lot of impractical P2 policies (i.e. if P2 always plays a_2). Therefore, it is required that the policy is compared to a 'good' P2 policy.

Nash Equilibria

A Nash Equilibrium policy is a *good* policy for a player to use as described within the Literature Review chapter. When the players are using a Nash Equilibrium policy pair then there will be no reward incentive for either one of the players to change their policy. This means that if P1's policy is compared to a Nash Equilibrium P2 policy, the most the policy can achieve for P1 is the reward that would have been obtained if P1 was using the corresponding Nash Equilibrium policy.

As an example, consider a Nash Equilibrium policy pair for the above game. P2 always plays b_2 and P1 always plays a_1. The observed rewards from these policies are one for P1 and two for P2. This means that no matter which policy P1 uses against the Nash Equilibrium P2 policy, the highest reward P1 will observe is one. Similarly, the most P2 would observe against the Nash Equilibrium P1 policy is two.

This implies that individual player's policies can be now compared i.e. the closer a reward obtained under a policy to the reward obtained under a Nash Equilibrium solution, the better. However, this is not an undisputed claim of *goodness* (Binmore, 1990). It could be argued that being close to a Nash Equilibrium solution is not necessarily a good thing. There are other policy strategies which could be employed, for instance *Co-operative play.*

Co-operative play is where the players agree to perform on polices which allows mutual benefit. However, there is no guarantee that an opponent will follow an agreed policy and may choose a policy which gives increased reward, at the expense of their opponent. Other elements of play must come into force for co-ordinated play to work

(i.e. trust, punishment for non-co-operation, etc.). This would also mean that the airlines are effectively *price-fixing*, which is illegal in most countries.

Using a Nash Equilibrium as the benchmark for the learning policies is one way to compare them, but not the only way. The reason for using Nash Equilibrium policies as the benchmark is that the learning policies are expected to convergence to a Nash Equilibrium[9]. This is discussed further in the Model chapter.

Dynamic Programming (DP)

As the RL results are being compared to the Nash Equilibrium (or variation to the Nash Distribution), these values need to be computed. Within a relatively simple model, it must be possible to compute the values using *Dynamic Programming* (or *backward induction*). When considering a highly complex model (i.e. when using a highly complex customer model), it is impractical to solve using this method. This is one of the reasons for considering a simple model in the first place.

Dynamic Programming was originally presented by Bellman during the 1950s (Bellman, 1952, 1954). It is the main method to solve Markov Decision Problems (MDP) and more importantly, stochastic games. The algorithm works by searching backwards through the decision tree (or sequential game), calculating the value of each state in turn. It is assumed that both players are working optimal, hence Bellman's principle optimally equation can be applied. This means that as a sequential game is used, each decision just depends on the future rewards.

The algorithm starts at all possible pre-terminal states and determines what the expected values are for that state (assuming that the players are using some action selection mechanism). If the state under consideration is a customer model step, the dynamic program will need to calculate the transition probabilities from this state to the next one. Once these calculations have been completed, the expected value of the pre-terminal states are known and are able to calculate the the expected values for the states previous to these ones. This process is repeated until the expected value of each state has been calculated. This leaves the correct policies (the determined action-selection policy at each of the different states) and the expected value of the game (this is the expected value of the initial state).

[9] Actually to the *Nash Distribution*, which is closely related to the Nash Equilibrium.

There are two problems with the dynamic programming method (as highlighted in Gosavi (2003, 2004)). Firstly, working out the exact transition probability from one state to another can be difficult, especially if it is a complex customer model. This problem is called the *Curse of Modelling*. Secondly, if the model has a large number of rounds, there will be a large number of states to evaluate and every one must be evaluated. This problem is called the *Curse of Dimensionality*.

The first of these problems is the reason for investigating the use of Reinforcement Learning. A practical modeller may want to construct some complex customer models and cannot determine the transition probabilities at all possible states. Reinforcement Learning is a heuristic method and Dynamic Programming is needed to calculate the actual Nash Equilibrium so there is something to compare the experimental results to. Thus a simple customer model is required for the experimentation.

As well as finding various Nash Equilibrium policies, dynamic programming was also used to find some of the Nash Distributions ones. To solve that traditional Nash Distribution method would have been difficult within a sequential game, however, this research was concerned with the variation on the Nash Distribution (VND) policy (described in section 2.2. The VND can be solved in a similar way to the Nash Equilibrium, by using backward induction. However, unlike the Nash Equilibrium policy, the solution does not maximise over returns, but instead maximises over expected return under the Boltzmann action selection method.

The VND randomizes the action selection at each state within the game, and this randomization only affects the current state, hence why dynamic programming can be employed. Thus at each state, dynamic programming can determine the expected return from subsequent states, given that Boltzmann action selection is used, and hence determine the expected value for each action at that state.

As mentioned in the Literature Review chapter, there might be several Nash Equilibria for any given game and Dynamic Programming is needed to find all of these. As the different Nash Equilibria relate to different action-selection methods, the dynamic programming algorithm can be tweaked to calculate all of the possible equilibria. In the Model chapter, the implications of the different Nash Equilibria are discussed.

Summary of Dynamic Programming

No other method has been considered, apart from Dynamic Programming, to calculate the Nash Equilibrium solutions because there was no awareness that any exist for the form of sequential game. A summary of this section and Dynamic Programming is as follows:

- Results from learning methods compared to Nash Equilibrium and Nash Distribution policies
- Dynamic programming methods to find the Nash Equilibrium and Nash Distribution of the simple game
- Dynamic Programming is limited by the Curse of Dimensionality and Modelling

So far in this chapter the research aims, the model design to run experiments on and what the experiment results are to be compared to have been described. Now the methodology for running the experiments will be discussed and how the comparisons are to be conducted.

3.4 Empirical Results

The actual experimenting for comparison and the different outputs required are now considered. From previous discussions, the requirement is to compare the learnt polices to the Nash policies but how do you compare a policy? This is a non-trivial question and several different approaches are considered within this section. A stochastic model is used (i.e. both the Customer Model and Boltzman action selection are stochastic), so the outputs will have to be repeated for statistical significance. Finally, as learning from the RL methods can be *improved* upon indefinitely, there is a need to consider how many plays of the game (called *episodes*) to run.

As part of trying to prove the effectiveness of RL, the effectiveness of individual techniques needs to be considered. The three different techniques that are considered are SARSA, Q-learning and Monte-Carlo Learning and are all discussed within the Literature Review chapter. All of these methods are dependent on input parameters (i.e.

the temperature parameter τ and the step-wise parameter λ). Therefore, there is a need to compare the techniques simultaneously while using different input parameters. All three techniques use the same parameter set so presenting how each of the techniques varies over the same set of parameters is possible.

Parameters

There are two variables that are used in the different techniques, namely the temperature parameter 'τ' and step-wise parameter 'λ'. The temperature parameter controls the amount of exploration (i.e. non-greedy play) and the step-wise parameter is the learning rate of updating algorithms. Simply put: lambda determines whether the algorithm will converge and Tau determines what it converges to.

There are a lot of restrictions on the step-wise parameter to ensure convergence, however the convergence proofs presented in Chapter Six allow some flexibility with the values used. After initial experimentation it was deemed appropriate to hold the value. These experiments are discussed in the Previous Research section in Chapter Seven. Another reason for not varying the parameter was that lambda is a function of a number of episodes and it is not clear how this should change.

The temperature parameter was considered more important to vary as it could have an impact on what the learnt policies converge to. The temperature parameter was also a constant value for each run, so was far easier to vary in different runs. It was also possible to make tau a function of the episodes '$\tau(e)$', however this would impact on the convergence of the methods. Therefore tau was kept constant over the number of episodes.

Obviously a requirement is to find the tau value that gives the *best* results. By *best* it is meant that the tau value that produces the most meaningful results after a fixed number of episodes. This could be in producing the policy which is closest to the Nash Equilibrium (or Nash Distribution), it could also be in the producing unexpected but useful results (i.e. co-operative play between players).

There have been several suggested methods on how this to find the *best* tau. For example, some experiments have been conducted to look at getting round the *black art* of choosing the parameters (see Sikora, 2006). However these methods require a

lot of extra computer memory, something that cannot be spared within the model. Therefore, different tau values to find those that produced the best results were sampled. Both tau and lambda are discussed in more detail in the next chapter.

Outputs

There are several things the empirical results are trying to determine. First, while looking at the simplest game, it is asked:

- Which of the techniques produces the *best* results?
- Do the techniques converge to the Nash Distribution policy?
- If so, how many episodes does it take to converge?
- Once the technique has converged, is it stable?
- What other characteristics do the run policies exhibit (i.e. myopic or random play)?

These are all valid questions and runs were conducted to answer them. However, before the model was run to investigate these ideas measures of effectiveness were set and how to determine if a policy has converged.

Measures of Effectiveness

It was required that a *good* learnt policy was found. It is not necessarily clear what makes a policy *good* and whether this *goodness* can be represented in a single value. The assumption is made that being *good* is being like the Nash Distribution policy. How can it be determined that a policy is like a Nash Distribution policy? There are two main ways that this can be approached: comparing probabilities (equivalent to comparing Q-values under Boltzmann action selection) or by comparing returns obtained.

The first way would be to compare the Q-values (and therefore action selection probabilities) at each state. The absolute differences from each action could be summed to give a value for that state. The values for each state could be summed to give a

difference value. Therefore, under this method the measure of *goodness* is how small the this value is.

This assumes that all states are of equal worth. Due to the probability distributions of the actions, some states may be difficult to reach within the game so are not as important to the policy as much as states on well-trodden paths. This means that large differences on uncommonly visited states could lead to the rejection of a policy, even though it may produce the same results as the Nash Distribution. This implies that to use this method of comparison, the states would need to be weighted somehow.

A state could be weighted by the probability of arrival at that state. Within the game a state can be visited only once per play (as a state is dependent on the round within the game, which only occurs once). This leads to several questions about which probability is used. The probability of arriving at a state will depend on whether the Nash Distribution or the learnt policy is considered as the underlying probability distribution[10]. It will also depend on the opponent's policy (and the customer model).

The biggest problem with this method is that there might be several different ways to get to the same return pair. For example, if the Nash Distribution policy results in P1 selling one seat in the *second* round but another policy sells once seat at exactly the same price but in the *first* round, then both policies give the same return (against this fixed opponent) but the compared Q-values would be different. Therefore, it seems reasonable to compare the actual return gained than the difference in Q-values.

The second approach was taken and the returns obtained from playing the policies were compared (against a standard opponent's policy). Given a policy, it can be played against a standard opponent's policy and the return distribution is generated. Standard measures can then be used to compare the return distributions.

The immediate question that arises from this is which policy will be the standard policy that the opponent uses? In theory, the two policies should be played for comparison against every type of opponent's policy. However, as there is an infinite num-

[10]One needs to be used as the differences are weighted between them

ber of policies to choose from, this would be an impossible task. A sample of policies must be selected to be played against.

The decision was made to use an opponent's policies which display behaviour that was of most interest. These policies are the Nash Equilibrium, Nash Distribution (for the appropriate temperature parameter), a completely myopic playing opponent and one which was random (i.e. all actions are chosen randomly).

There are two learning players so this process of comparison was repeated for both. To check for learnt co-operation between the learning player's policies, the return distribution of the learning players was determined and compared it to the return distribution of the corresponding Nash Distribution policy pair.

When looking for co-operation, a higher average reward than experienced in other policy pairings is expected. This is an exception because usually the highest average return as measure of *goodness* was of no concern. If the highest average return was a measure of *goodness*, then the policies that just play the highest prices all the time would be considered a *good* policy pair as a high average return would be achieved. However, if either player's policy were to play against another more sophisticated policy, then it likely to achieve a very low return. Therefore, the higher the average returns does not mean the better the policy.

What is the concern here is *how close* a policy pair is to another policy pair, thus giving an indication that the different policies are similar. Using only the expected return to compare would lose a lot of the information that is shown in a reward distribution and may lead to incorrect conclusions. Hence to use all of the return information from a policy pair's return distribution, the distributions and not the averages need to be compared.

Even if a learnt policy seems similar to another policy, there is no guarantee that this will remain the case. If more episodes are played, more learning is achieved. Therefore, to have convergence of a learnt policy to another policy, there must also be stability.

Measures of Stability

One of the problems with a multi-agent system is that it may look stable but sudden changes (or jumps) can occur. Within a learning game this happens when a player favours one action over another but through learning changes to the policy, changes to the other action. The consequences of one action change can completely affect the returns that are observed from the game and a jump in expected return from the policy can occur[11].

The reason that a single change can have a dramatic effect is that though the player that changed their policy would not notice much difference in their return, their opponent will be facing a completely different policy and hence a completely different outcome. However, though the learning system is sensitive, these jumps become less frequent over time (lambda, the step-wise parameter is getting smaller as the number of episodes increases, so the learning rate decreases).

For a learnt policy to be stable there should not be return jumps (or changes) expected as follow-on episodes are run. There are two factors that determine the stability of the system, namely: unlikely random occurrences and the number of episodes played.

Unlikely random occurrences can result in players observing returns that are uncommon but will still adjust their policies to them. This could happen in any random system and can be dealt with by running the model many times (i.e. 100) and taking the average. This is a standard statistical sampling size[12] and it is employed within the results.

The second factor that affects stability is the number of episodes within a run. The learning parameter decreases with episodes, therefore it becomes less likely that the policy will change after a large number of episodes have been run. However, this is

[11]This does not happen when the Boltzmann action selection method is used as all changes are smooth.

[12]Though sampling the runs 100 times seems like an arbitrary quantity, sensible confidence intervals can be determined from it. However, to use confidence intervals assumptions are made about the underlying distribution of the run's results. This is discussed further in the Empirical Results chapter.

not a certainty and there is always a chance that policies will change. To determine whether a policy will remain stable, special stability runs are conducted where the initial Q-values are those of the policy that the learning mechanism to converge to. By starting with the policy that a run should converge to, it is possible to see if policy remains the same and is stable.

Multiple Nash and Single Learning

This section, so far, has discussed which runs must be conducted and what the output must be. There are details that are discussed now before moving onto the next section about the types of measures to be used.

Multiple Nash Equilibria

As mentioned in the Literature Review chapter, a game can have multiple Nash Equilibria. This raises the question of what to compare against. Boltzmann action selection is being used within the learning model but the learning player's policies are expected to converge to the Nash Distribution policy. The limit of the Nash Distribution policies (as temperature decreases to zero) corresponds to a unique Nash Equilibrium policy (see Fudenberg and Levine, 1998), hence this is the only equilibrium that needs to be considered.

Single Learning Agents

All the runs described so far assume that there are two learning players. It could be possible to only have one learning agent and play them against a static opponent. This is of no interest to the research for two reasons. Firstly, the learning player would only learn to respond to the static player's policy, thus the learnt policy might not be useful against any other policy. Secondly, though it is expected that a single learning player can learn quicker than multiple players learning simultaneously (see Takadama and Fujita, 2005), initial knowledge is required of what policy to learn against.

The obvious policy for a single learning player to play against would be a Nash Equilibrium one. As the underlying purposes of these experiments is to assume that the game cannot be *solved* using conventional means, this is an unreasonable assumption. Therefore, the learning player would have to play against some other policy (i.e. the

myopic policy) and it is unclear what the benefit of doing this is. Hence the learning player plays learning player in these experiments.

Summary of Empirical Results

This section has discussed how the empirical runs were conducted and the decisions that were made in which should be run. The summary of runs that were conducted is as follows:

- A series of runs was conducted for each of the different Reinforcement Learning methods
- The runs were varied by the temperature parameter 'tau' but not the learning parameter 'lambda'
- A *good* learnt policy is one which similar to the Nash Equilibrium and is stable
- Each run outputs a return distribution from the learnt policies for comparisons

The method of comparing these reward distributions is considered in the next section.

3.5 Comparisons

Part of the research is to compare the effectiveness of the different learning models. To make any type of assessment, data must be collected and measured. The data collected from the different runs are the *reward distribution*. This section focuses on the methodology of determining the measurements that are required to assess the reward distribution.

The *Return Distribution* is a bivariate probability distribution of the possible returns when two policies are played against each other. The intention of deriving the return distributions was so that the learnt policies can be compared to other policies and to determine if it has converged or the nature of its behaviour. Given that an infinite number of polices exist for any of the games, it would be impossible to compare against all of them. A selection of comparison policies had to be made.

The policies were selected for comparison by what observations were expected from the learnt policies. There were four different *standard* policies for comparison. The first two standard policies were the Nash Equilibrium policy (NE) and the Nash Distribution (ND) policy. These policies were chosen because it was expected that the players would learn to play like either of these policies after enough episodes were played.

The other two standard policies were the myopic policy (MY) and the completely random policy (RN). The myopic policy is the policy where the players try to maximise their immediate reward (thus playing myopically). Myopic play by the learning players was expected to occur during the early episodes. The completely random policy is where all actions are equally likely to be selected. It was expected that completely random play would occur during the early episodes and when there was too much exploration occurring by the learning players. Ideally, after sufficient episodes neither learning player's policies are like the myopic or completely random policy.

The Nash Distribution policy is the policy that it was hoped the learning player's policies would converge to, given the fixed temperature parameter. As the ND policy is dependent on the temperature parameter, when the return distribution was calculated it was assumed that the same temperature parameter was used. When the learnt policies' return distribution (called RL) was compared to this ND return distribution, it was also calculated using the same fixed temperature parameter.

The Nash Equilibrium relates to the Nash Distribution with a temperature parameter of zero. Hence when the NE reward distribution was calculated it was assumed that the temperature parameter was zero (this refers to the greedy actions selection needed for the Nash Equilibrium policy). It was also possible to calculate a version of the RL reward distribution with a temperature parameter of zero. This gives a deterministic reward distribution to compare the NE reward distribution too. This second version of the RL reward distribution is the *off-policy* version, where exploration is no longer necessary and the best actions are selected.

The other two standard polices' reward were calculated with no need for the temperature parameter.

		P2				
		RL	NE	ND	MY	RN
	RL	X	X	X	X	X
	NE	X	O			
P1	ND	X		O		
	MY	X			O	
	RN	X				O

Table 3.1: An indication of which bivariate reward distributions were calculated for which pairs of players' policy.

Not only were reward distributions created for each of the types of learning, they were created where the single learning player's policy was played against them. Table 3.1 shows the policy pairs considered within the analysis. The 'O' in the table indicates that the bivariate reward distribution of these pair of policies was computed offline from the main collection of runs. The 'X' indicates that the reward distribution was calculated for each learning run (the learning run is repeated a hundred times for statistical significance).

The purpose of calculating these *mixed* distributions was to see how the learnt policies react when not playing their learning partner. This is especially true for a Nash Equilibrium policy as the learnt player's policy could only expect to achieve a return less than if the equivalent NE policy had been played. This upper limit can act as a benchmark for the learnt policies.

This property is useful when there are multiple best responses to a Nash Equilibrium policy (i.e. the corresponding Nash Equilibrium policies) as each possibility will still generate the same expected return (otherwise is would not be a best response). Therefore, if a learnt policy is observed that was not expected but the expected return reaches this bound then this implies that a Nash Equilibrium policy has been found.

If a learning player policy gets an average reward greater than the Nash Equilibrium policy (when playing another Nash Equilibrium policy) then there is a bug within the comparison method and the results would need to be re-verified.

Distribution Comparisons

Now it has been determined which reward distributions were calculated, comparison can begin. As mentioned before, if two policies are the same, they would perform exactly the same way when played against any opponent. This means that the reward distributions would be identical. There are an infinite number of possible opponent's polices so exhaustive comparison is impossible. This means only a finite number of return distribution can be compared.

These comparisons will need some type of measure to indicate how far apart the return distributions are. It is not immediately obvious which measures should be used (e.g. L^2-norm, Chi-squared statistics, etc.). The first part of the empirical analysis is to determine which metric to use. Like possible opponent's policies, there are a lot of metrics to choose from. The list can be shortened with the work of Alison Gibbs and Francis Su (Gibbs and Su, 2002). In their paper *On Choosing and Bounding Probability Metrics*, a list of measures is derived for comparing probability distributions (which are bivariate return distributions). From this list the following measures are chosen to be evaluated: Kolmogorov-Smirnov (KS) statistic, Total Variation (TV) distance, Hellinger (H) distance, Average-KS (AKS), Information Value (IV), Separation Distance over Theoretical Distribution (SD1), Separation Distance over Empirical Distribution (SD2), Chi-squared Distance over Theoretical Distribution (CHI1), Chi-squared Distance over Empirical Distribution (CHI2), Expected reward for P1 (E1) and Expected reward for P2 (E2).

By performing the comparison for each of these measures, it can be concluded which perform well and which do not. From analysing this behaviour it can be determined which measures are appropriate for the comparison. There was no intention to only use one measure for all the comparisons and where appropriate, multiple measures are discussed.

As previously mentioned, unless a measure gives a definite result of zero then it is difficult to conclude that the policy has converged to the policy it is compared to. The measures do give a means of comparison with which to judge the different learnt policies with. These measures form the basis of the results and the conclusions about them.

The benchmark for the measures will be the ND return distribution compared to the NE return distribution. As both these distributions are well understood, that knowledge can be used to make judgments about the different measures. For example, the distributions are expected to diverge as the temperature parameter increases; therefore the measures are expected to increase as well.

There are problems in measuring the different policies (i.e. not being able to compare to all policies, etc). A problem with the measures is that they are condensing two bivariate distributions into one number. Whenever this *dimension crashing* occurs within data, information is lost. It is difficult to determine whether this information is important or not. The alternative of presenting all the reward distributions is impractical from both a analytical and a presentational point of view. This loss of information is therefore accepted.

As mentioned previously, each run was repeated 100 times so that statistical inference can be made. By having a collection of sampled measures, more anomalies should be picked.

It is possible that two different policies produce the same reward distributions when played against a small selection of opponents. Therefore, it can be concluded that only certain observed properties were observed.

The use of reward distributions to compare policies and the use of measures to compare reward distributions is not ideal and prone to several possible errors. However, without a better alternative to use for comparison this method was used. Though it cannot conclude that a policy converges completely in practice, it can theoretically be shown.

3.6 Convergence Proof

The measures used to compare the learning policies to the standard policies will not be adequate for showing complete convergence of the model, therefore it is important that theoretical results of convergence are shown.

The SARSA method was chosen, this was due to the limited academic literature on the subject (see Banerjee et al., 2004). By proving convergence of the learning

game, something original is added to the literature. Chapter six is devoted to this convergence. The proof for convergence for the SARSA method must work within the framework of the model. This is another justification for a simple airline pricing model.

In a stochastic environment, the definition of convergence is debatable as well. Therefore, the standard paradigm from measure theory (see Williams, 1991; Durrett, 2004) has be used for the convergence proofs, which is in accordance with the current literature.

Proving that the SARSA method convergence in theory does not necessarily mean that convergence will be seen in practice. The number of episodes required to show convergence in practice may be well beyond the limits to generate results. However it does give an indicator of where the learning policies are heading.

3.7 Conclusions of Methodology Chapter

In this chapter the means to analyse use of Reinforcement Learning within a game theoretic context has been discussed. The following methodology was derived:

- Construct a simple airline pricing game
- Solve the game using the standard method of dynamic programming
- Allow different Reinforcement Learning method to generate the possible policies for the games
- Use the these policies to generate return distributions
- Compare these return distributions to those generated by the Nash Equilibrium policy using different measures
- Make conclusions about the different measures and thus conclusions about the different RL techniques

The rest of this thesis is devoted to this task. In the next chapter the simple airline pricing games and its properties are discussed.

Chapter 4

Model

4.1 Introduction

In this chapter the framework described within the Methodology chapter is made into an implementable model. Once the model was constructed, it was possible to find various properties about it. These properties are also described in this chapter. The methodology framework did not cover all aspects of the model and where necessary explanation is given about any decisions that were made to complete the model construction. The airline-pricing model and the learning model have been split into separate sections, as within the methodology chapter.

The major property that was considered was the Nash Equilibrium and its variants. Not only was it intended to find the technical details of mathematics of the Nash Equilibrium but also to demonstrate a Nash Equilibrium within a *real world* context. A large proportion of this chapter is therefore devoted to the Nash Equilibrium.

The model is required to provide empirical as well as theoretical results, therefore a mechanism was needed to generate them. A computer-simulation was constructed for this purpose [1]. The final section of this chapter is devoted to this computer-simulation and its verification.

[1] It would have been impractical to use manual or physical modeling methods to generate the numerous runs needed to achieve a sensible number of results.

Summary of Chapter

The chapter considers each of these sections in turn:

- Construction of the airline pricing model
- Nash Equilibrium solution to the airline pricing model
- Construction of the learning model
- Programming code considerations
- Verification and validation

4.2 Construction of the Airline Pricing Model

The airline-pricing model described in the methodology chapter gives the basis for constructing a mathematical version. By constructing the model in mathematical terms it is possible to gauge actual results from it. Any mathematical model is constructed using algebraic notation and the notation that is required is now considered. The model considers the selling of seats of two competing airlines (P1 and P2) over a fixed finite number of discrete time-steps (or round) $n \in \{1, 2, ..., N\}$. It was assumed that when the time-steps reach N then the flights depart and no more seats can be sold. This process is a game because the airlines are able to compete by changing their current prices p^i (where $i \in \{1, 2\}$ indicates the appropriate airline) at fixed intervals within a round[2].

The airlines (which are the players) make decisions about their prices based on the current state of the system. This state is defined as simply as possible within the methodology chapter and each players state only takes into account three variables: current round, opponent's current price and seats remaining on aircraft.

[2]For all purposes, the airlines' single-leg flights are considered to be homogenous. In advanced cases, however, the flights are considered to have different numbers of seats. This means that the players only way to attract customers is their price.

Prices

It was mentioned in the Methodology chapter that prices can only take a finite number of discrete values. This has been interpreted to mean that any finite arbitrary set of ratio data is sufficient. The natural numbers from zero to ten were used for the set of possible prices. Here *ten* represents the maximum amount for which there exists a customer who is willing to pay that price. A price of zero represents the minimum price at which the airline would be better off selling the seat than leaving it empty [3]. Though this set of possible prices will be adequate for this experiment there could be potential problems associated with it.

The inclusion of a price of zero could be considered controversial, however there were two reason for its inclusion. Firstly, a player would not be expected to choose a price of zero voluntarily, hence observing this price could imply that the learning players are still playing randomly (and expected to be observed at the earlier episodes). Secondly, as this chapter will show, the *optimal* policies of the players are not immediately obvious and thus it seems inappropriate to exclude this price without fully understanding the game dynamics. For instance, a player might wish to use a price of zero to *punish* their opponent for previous price choices (see Axelrod, 1997).

Within a real airlines' dynamic pricing model, a wider range of prices is likely to be available with multiple prices being offered at the same time. These multiple prices are related to *fare classes* (or booking classes), and relate to different constraints on the ticket (Talluri and van Ryzin, 2004). These constraints might include factors such as child ticket, return restrictions, etc. This complication has been eliminated from the model by assuming that:

- Single-leg tickets only
- Homogeneous seats available
- Only one price offered by an airline at any one time

[3] This minimum cost is not necessarily zero pounds as there is a marginal cost associated with every customer on a flight (i.e. there is a fuel cost associated with transporting the weight of a customer and their luggage)

- There are only 11 possible prices, which are evenly spaced apart

These assumptions are quite limiting on the model but as the model results show, complicated results are still seen. As mentioned in the literature review chapter, there is some difficultly in determining the utility functions of the airlines (players). Thus as the same prices (and therefore, the same rewards) are available for each airline, there is an assumption that there are *Homogeneous players*.

A lot of time could be spent discussing the possible disadvantages of using a limiting number of available prices, however *Occam's razor* is applied here (see section 3.2 for more details) and it is argued that the extra detail (from using a more complex pricing structure and utility model) is unlikely to add anything new to the model's results given the high level of abstraction already employed.

There is another impact of allowing only 11 prices within the model. The number of prices available will have a direct impact on the number of states required to be stored. As every state needs to be repeatedly visited for complete convergence of policy to occur, the number of prices available will have an impact on any convergence results (as well as the memory requirements). Using only two or three prices would mean even less states to deal with but limited number of price options would be too unrealistic.

In conclusion, the prices available within the model can freely be determined, however the number, scale and ratio of them will have a direct impact on any results. Therefore, the choice of prices is a limitation of the model.

Time-steps and Seats

As already mentioned, the game is sequenced by a number of time-steps. What unit of time these steps represents could be as little as one second[4]. This would mean that in the extreme of modelling, the sale of seats on the flight six months in advance would require 15,778,800 rounds. However, as a customer is unlikely to arrive every second (and therefore, the only change to the state is an increase in rounds) this is a waste of rounds and digital memory.

[4]It is reasonable to assume that even with automatic updating of the state variables within an airline's revenue management system, the process will take at least a second

As defined within the methodology chapter, this is a sequential game thus there is limiting consequence that the players take turns to change their prices[5]. This means that a round can be interpreted to mean that a set number of customers come along and both players get a chance (within that round) to change their price. Thus a round is not a fixed time-step but when a certain number of actions have occurred (i.e. some customers have arrived and the players have had a opportunity to change their price).

A strict sequence of events is imposed onto a round: namely player one (P1) changes their price, customers arrive, player two (P2) changes their price and then more customers come along. This set sequence of events is shown in figure 3.1. The first round is different from the rest as no prices have been set. During this round the sequence is that P1 sets their price, then P2 and then some customers arrive.

Both airlines are limited by seat capacity, which could range from one seat to approximately 850[6]. An airline could allocate more than one plane (or change planes from within their fleet if necessary) to a single-leg journey but this is not considered here. The methodology chapter states that the airlines will not overbook nor will cancellations occur so once the airplane's capacity is reached an airline is unable to sell more seats. Thus once capacity is reached, it is assumed that all customers will purchase seats from the other airline (assuming there are some available otherwise the game has reached a terminal state).

It is important to note that the policy of the players is not necessarily to fill their plane before departure but to achieve the maximum revenue from selling their seats. For example, when demand is much greater than total seats available, a good strategy is for a player to encourage their opponent to sell all their seats so that the player can sell the remaining seats at a high price without fear of competition.

As with the prices available and number of rounds, the number of seats available has an impact on the number of states available and thus on the expected convergence

[5]Though this is also liberating as, the alternative, simultaneous moves are hard to jusfity within the real-world

[6]Using the reference data for the Airbus A380 (see the Airbus website http://www.airbus.com for details(accessed on 1st March 2008)).

results. The game is viewed with as few seats as possible to encourage convergence. All the variables for the game can been seen in table 4.1.

Notation	Meaning
$P1$	Airline player *one*
$P2$	Airline player *two*
n	$\in 1, 2, ..., N$. Current round within a game
i	$\in \{1, 2\}$. Player index for player's P1 and P2
p^i	$\in 0, 1, ..., 10$. Current price offered by player i
S^i	$\in \mathbb{N}$. Number of seats available on player i's plane
s^i	$\in 0, 1, ..., S^i$. Number of seats remaining on player i's plane
r_n^i	$\in \mathbb{R}$. Reward observed by player i during round n

Table 4.1: Notation of Airline-pricing model

Simple 233 games

The primary focus is to determine whether the airlines' policies will converge to the Nash Distribution policy, therefore it would be reasonable to look at the simplest game first. The game with only one round has not been considered as the simplest version as it does not have any dynamical aspects to it (and has a trival result of both players setting the smallest non-zero price). The game with two rounds is therefore considered to be the minimum.

The simplest customer model available can now be defined, where a single customer arrives in each of the customer phases of the game and chooses the airline with the lowest price (or randomly if both have the same price). This is called the *Simple Customer Model*. This means that when there are two rounds three customers are seen. An explicit mathematical representation of the simple customer model is given in equation (4.1). The inputs to the model are: P1 (Player one's current price), P2 (Player two's current price) and ϵ. ϵ is a uniformly random variable on the interval $[0, 1]$ and used to decide between the players when their prices are the same. The

outputs for the model are the rewards gained by the players.

$$f(P1, P2, \epsilon) = \begin{cases} (P1, 0) & if\ P1 < P2 \\ (0, P2) & if\ P1 > P2 \\ (P1, 0) & if\ P1 = P2, \epsilon \leq 0.5 \\ (0, P2) & if\ P1 = P2, \epsilon > 0.5 \end{cases} \tag{4.1}$$

By allowing both airplanes to take all the possible demand (i.e. three seats in the two round case) the effects of the airplanes running out of seats do not need to be considered. Even though the numbers of seats available on the planes will have no impact on the results for this simple game, the number of seats on each plane has been included in the description (i.e. '3' in 233 game) to remind the reader of the possible total number of customers available to the players. The number of seats also remains a factor in determining the state, giving a good framework to use with other customer models (see Chapter Seven), where the number of seats have an impact on the policies.

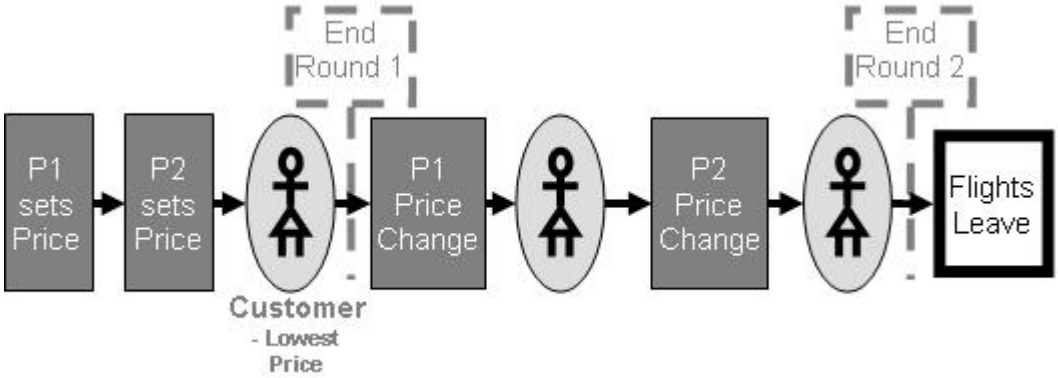

Figure 4.1: Flow chart of Simple 233 game

The game described here has been called the *Simple 233 Game* and produces some surprising results (which will be described later in this chapter). The simple 233 game is represented in figure 4.1. The game can be extended to include more rounds by increasing the number of seats available so that both players continue to satisfy all the demand (i.e. two extra seats per extra round are added to match the number of customers). This way the simple 355 game, simple 477 game, etc. can be derrived.

Variations on this simple game are considered in chapter seven. There the effects of

decreasing the number of seats available are a considered, also the use of more sophisticated customer models. Now that this game is available for the players to play, possible policies are considered, especially the Nash Equilibria.

4.3 Nash Equilibrium

The choices that the players make within the model are called their *policy*. A policy can take several different forms, from the player *always choosing seven as their price* to a player *choosing a price which will minimize their opponent's return* [7]. One policy that you might expect the airlines to play is *choosing prices which maximize their return*. It is not trivial to find this policy and the policy itself is dependent on several factors (e.g. the opponent's policy). By changing the policy to *choosing prices which maximize my return, given the opponent's current policy* then a Nash Equilibrium is found (as defined in the literature review) if both players use this policy. This section therefore deals with finding the Nash Equilibrium policies.

There are a few important non-Nash policies that are defined here; namely *Completely Random* policy and *Myopic* policy. The *Completely Random* policy is when a player always chooses their price at random (i.e. P(choose certain action) = 1/11, as there are 11 possible prices). The *Myopic* policy is when the player is only concerned with obtaining the next reward and does not take into account of any future action (i.e. a myopic player always plays the highest price that allows them to undercut their opponent's current price, this ensures they receive the next reward). These polices are important because an inexperienced player might be expected to play in a similar manner.

There are other important policies that have not been considered because they might be accounted for in another policy. For example, a *reactive* policy (i.e. where the player policy is just to react to an opponent's play) is similar to the myopic policy. Other policies, like a *tic-for-tac* shown in Robert Axelrod's famous prisoner dilemma experiments (see Axelrod, 1984, 1997), are based around the players playing repeated

[7] Within the simple 233 game, this policy can easily be achieved by the player always using a price of zero (hence all customers will buy their seats at a price of zero so the player's opponents observes a zero return).

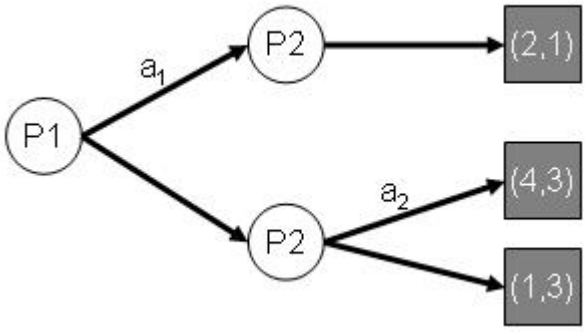

Figure 4.2: Example sequential game with multiple equilibria

plays (or episodes) of the game. Though it is strictly true that the learning players are playing repeated plays, it is assumed that each episode is independent [8].

Multiple Nash Equilibria

As mentioned, it is intended to find the Nash Equilibrium to the sequential game. This can be done using *reverse induction* using dynamic programming as highlighted in the Methodology chapter. The method works by starting at a pre-terminal state and working out what P2 policy will be at that state (since P2 will be the last to select a new price). Once the expected returns have been determined for all states of this kind, the states where the last action for P1 was chosen are next considered. The expected returns can then be used from the pre-terminal states to determine the policy of P1 and therefore, the expected return. This backward induction is repeated until the initial state is reached. P1 and P2 then have a complete policy. The policy was determined by the players selecting the prices which will give them maximum future return. This implies a Nash Equilibrium policy was found for both players. However, there is one consideration that needs to be taken into account: what if two actions have the same return?

From an individual player's point of view at that stage of the game, it does not matter which action they take when the expected return is equal for both. However, it

[8] To observe a learnt policy which takes into account the repeated play aspect of the learning model would be remarkable but highly unlikely as the policies are updated by the disjoint return of each episodes. However, this does not mean that this phenomenon cannot occur as an on-policy updating method is used thus the updating of one episode will affect the actions taken in the next.

could have a impact on the other player's return. To demonstrate this, consider the extensive-form sequential game given in figure 4.2. Within this game P1 has a choice of choosing either action a_1 or action $\neg a_1$. If action a_1 is chosen then P2 only has one available action and the return for the game is (2, 1); P1 gets a return of two and P2 gets a return of one. If P1 were to choose action $\neg a_1$ then P2 now has a choice of possible actions a_2 or $\neg a_2$. From P2 perspective, it does not matter which action is selected as they will observe a reward of three. However, it does have a big impact on P1 reward so much so that they might not have chosen $\neg a_1$ in the first place. Before what response P1 should take is discussed, let's consider possible policies that P2 could employ to deal with this situation. To distinguish from the player's actual policies, these policies are called *tie-breaker* policies. Here is a selection of a few tie-breaker policies:

- **RANDOM**: Player randomly chooses between alternatives
- **HIGH**: Player chooses the price with the highest value
  - (or action a_2 in the case of example 4.2)
- **LOW**: Player chooses the price with the lowest value
  - (or action $\neg a_2$ in the case of example 4.2)

These are just a few of the possibilities; there are an infinite amount of tie-breaker policies that could be employed[9]. The impact of each tie-breaker policy can now be considered in turn. When the RANDOM tie-breaker policy is employed then P2 will choose between a_2 and $\neg a_2$ giving a expected return pair of (2.5, 3), when P1 plays $\neg a_1$. If P1 knows that P2 is using the RANDOM tie-breaker policy, then P1 would choose $\neg a_1$ as a return of 2.5 is greater than the fixed return observed by playing a_1 (which is two). When the HIGH tie-breaker policy is used then P2 would choose a_2 over $\neg a_2$ giving a expected return of (4, 3), when P1 plays $\neg a_1$. Again, P1 would choose to play $\neg a_1$. Finally, if the LOW tie-breaker policy is employed then P2 will

[9]Consider the variation on the RANDOM tie-breaker policy, where each action is given an arbitrary weighting of being selected. In a larger game, a player might wish to employ different tie-breaker policies at different stages

choose between $\neg a_2$ over a_2 giving a expected return pair of (1, 3), when P1 plays $\neg a_1$. In this case P1 would prefer to play a_2, hence obtain a reward of two. To summarise, the expected return from the example for the different tie-breaker policies are:

- **RANDOM**: (2.5, 3)
- **HIGH**: (4, 3)
- **LOW**: (2, 1)

All three of these tie-breaker policies lead to a Nash Equilibrium. This may seem surprising as the return pair obtained from the LOW tie-breaker policy is (Pareto) dominated by the other two pairs. However, the definition of the Nash Equilibrium is refered to to explain why it is a Nash Equilibrium. The solution obtained above for when P2 is using the LOW tie-breaker policy is derived from a policy of P1 which *assumes that P2 will use the LOW tie-breaker policy in future rounds*. So if P2 were to change to the HIGH tie-breaker policy, it would have no impact on the returns obtained because current P1 policy *assumes that P2 will use the LOW tie-breaker policy in future rounds* and therefore will play accordingly. This means that P2 will still reach the same tie-breaker positions and, by definition of a tie-breaker, will obtain the same return. The important point here is that the players have a Nash Equilibrium policy if neither can gain any benefit from changing their current policy assuming that their opponent's policy will stay *completely* the same[10].

Given that multiple Nash Equilibria from the game are faced and that some will give a better return than others (for the players) begs the question: Which Nash Equilibrium should the players choose? There are various different methods of selecting a Nash Equilibrium when more than one is available (see Harsanyi and Selten, 1988; Herings et al., 2003). However, given that players learning to play the game are being dealt with, it assumes that they have a choice of which Nash Equilibrium they learn.

[10]This is not the same as an opponent's policy staying *blindly* the same. A player will recognise when they have moved into a different state to what they might expect and their policy will react to this state accordingly, only it might be based on wrong assumptions about the future rounds

The question is answered as a single Nash Equilibrium which relates to the Nash Distribution policies using Boltzmann action selection. This is the Nash Equilibrium obtained from both players using the RANDOM tie-breaker policy[11].

The concept of multiple Nash equilibria is very important as the size of the game grows, as there is likely to be more of them. It has been shown that the expected number of equilibria increases exponentially in normal-form games as the number of strategies increase (see Mclennan, 2005). Though the results have not been determined for sequential-form games, it can be assumed that similar results might exist.

Nash Equilibria are now considered for the simple 233 game. Other types of Nash Equilibria than the one related to the RANDOM tie-breaker policy, have been included for comparison purposes.

Simple 233 game - Nash Equilibrium

The game framework is one of sequential moves, as opposed to a simultaneous move game. This means that each player takes it in turn to decide their price. Therefore, a 'rational' player (or Homo Economicus as described in the literature review chapter) will choose their current price so that it maximizes their expected return. Using this knowledge the players expected return can be calculated via backward induction (as mentioned above). Let's first consider the following example of the simple 233 game.

Let's pretend P2 is about to make the price choice at the end of the second (and last) round. This means that there is only one customer left to arrive and they will choose the airline that has the lowest price fare (or will choose randomly if the prices are the same for both players). Let's say that P1's current price is *nine*. It is pointless P2 selecting a price of *ten* as this means that P1 will attract the customer. If P2 chooses a price of *nine* as well then they will have an expected return of 4.5, as they will only attract the customer half of the time. However, if they choose a price of *eight* then they will attract the customer and observe a return of eight. Similarly P2 will attract the customer for all lower prices. Therefore, the logical thing for P2 to do is to choose a price of *eight*. This would mean that observed reward from this would be $(0, 8)$, where numbers represent the reward for P1 and P2 respectively.

[11] When two actions have equal expected reward, Boltzmann action selection will assign equal probabilities to each. This relates to the definition of the RANDOM tie-breaker policy.

Now let's take a step back and consider P1 strategy in the last round if P2 has a current price of *ten*. There are now two customers up for grabs here and P1 has to decide whether to attempt to attract one or both of the customers. However, P2 will have a chance to change their price before the second customer comes along and will try and undercut P1's chosen price. Unless P1 chooses a price of *one*, P2 will be able to undercut them but if P1 does this then they will, at most, observe a return of two (assuming the customer's random selection goes in their favour). If P1 only attempts to attract the next customer, by playing *nine*, they will lose the last customer by observe an overall return of nine. Therefore, it is logical for P1 to try and attract only one of the remaining customers and gain a return of nine (with P2 observing a return of eight) for the last round.

A step further back is taken, to P2 action selection in the first round assuming that P1 current price is *five*. All three customers are up for grabs if P2 can select the right strategy. P1 has an opportunity however to take at least one of those players during its price change at the start of round two. Thus it would be expected that P2 tries and take the remaining two customers. To do this P2 will need to undercut P1 current price of *five*, and set their price to *four*. From the arguments above this would result in P1 changing their price to *three* and the P2 would follow suit with the logical choice of *two* for their final price change. So the results from P2 setting a price of *four* are:

- The first customer (from round one) choices P2's lower price of *four* (as opposed to P1's price of *five*)
- The second customer (the first one from round two) chooses P1's price of *three*
- The third customer (the second one from round two) chooses P2's price of *two*

This means that P2 would receive an overall return of six (four plus two). Now consider if P2 chooses a price of *ten* in response to P1's price *five*. From arguments above, the logicial price for P1 would be to choose *nine* and P2's last price change would be *eight*. So the results from P2 setting a price of *ten* are:

- The first customer (from round one) choices P1's lower price of *five* (as oppose to P2's price of *ten*)
- The second customer (the first one from round two) chooses P1's price of *nine*
- The third customer (the second one from round two) chooses P2's price of *eight*

This means that P2 would receive an overall return of eight (from only the last customer). By P2 losing a customer they received more return. By P2 trying to undercut P1 in the first round the only response that P1 has is to continue the price war (as P2 will undercut them for the last price change). This price war results in both players receiving little revenue for their seats sold. However, if P2 does not engage in a price war and increases its price, this encourages P1 to increase their price too and hence both players receive a much higher revenue for the seats sold. This occurs because the airlines are concerned with revenue and not number of seats.

As P1 receives a return of 14 from having a starting price of *five*, it is unsurprising that they choose this price. Hence one of the Nash Equilibria for the game is pure strategy set {*five*, *ten*, *nine* ,*eight*}, where the numbers correspond to the prices chosen by the players throughout the game (i.e. in the first round P1 chooses *five* and P2 chooses *ten*, etc.). A detailed account of expected returns for all the possible price choices is given in table A.6 in Appendix A, the introduction to the tables explains their layout. Though hard to validate, this sudden jump on price to stop a pricing war is seen in real-world airline pricing strategies i.e. British Airways adding an extra fuel surcharge to their prices in 2006.

This relates to Nash Equilibrium associated with the LOW and RANDOM tie-breaker policies. To understand what the Nash Equilibrium associated with the HIGH tie-breaker policy, the scenario when P1 chooses *six* as their initial price must be considered. If P2 chooses to follow this with a price of *ten* they will, again, receive a return of eight (but P1 will now receive a return of 15). If P2 chooses to follow this with a price of *five*, a pricing war results and P2 receives a return of eight again (but P1 will now only receive a return of four)[12]. Thus it does not matter to P2 if they choose a

[12]P2 choosing an initial price of *six* as well will result in them receiving a return of seven (which is less then eight they receive from choosing *five* or *ten*).

five or *ten* after P1 has chosen an initial price of *six* in the simple 233 game. Hence if the tie-breaker policy is HIGH then P1 will select *six* otherwise they are forced to choose a price one lower of *five* to force P2 to play the high *ten* (thus gain the benefit from selling two seats at a reasonable price).

These are surprising results for such a simple game and show the complexity that such a simple framework can bring[13]. This complexity is a reason why such a simple model for the experiment was chosen; otherwise it would be difficult to distingish between complex solutions and random effects within the learning model.

Simple 355+ games

Now that some of the Nash Equilibria for the simple 233 game[14] have been considered, simple games of more than two rounds can be considered (the shorthand 355+ is used for simple games with more than two rounds). The impact of using the different tie-breaker policies and the effect that has on the Nash Equilibrium again has to be considered. From this investigation some conclusions about the simple games can be drawn.

The 355 game has the same results to the 233 game for the last two rounds of the game. This might be expected to be the case, as backward induction is used to solve the game, however this is somewhat surprising. For the solution to repeat the last three actions, the same conditions are needed after the action selection of P1 in the second round (which corresponds to the first action selection in 233 game), which is observed (see table A.3 for details of the policy for the HIGH Nash Equilibrium). This selection guarantees that P1 does get the maximum reward for the remainder of the 355 game but at the sacrifice of the first customer (because P1's initial price is *ten*, the maximum possible). There are several phenomenons like this that appear

[13]This sophistication comes from an odd number of customers, an even number of players and from the low values obtained under a strictly cut-throat policy. Thus acting in a cut-throat way does not benefit the either player and one players will sell one more seat than the other introducing a bias into the game.

[14]As mentioned before, there are infinite possible tie-breaker policies and not all could be covered here. For example, there is another Nash Equilibrium where P1 is indecisive about whether their initial price should be *five* or *six*.

within the policies, however, for brevity, not all of them will be discussed here. The focus will be on the general properties of the different Nash Equilibrium policies.

When using the HIGH and LOW tie-breaker policy, both player's Nash Equilibrium policy will be a pure strategy (as a mixed strategy can only occur when there are two prices which offer the same return and the HIGH and LOW tie-breaker policies uniquely choose between them[15]). These pure strategies are shown for the different games in the tables A.1 and A.2 respectively, which can be found in appendix A. The RANDOM tie-breaker policy does not necessarily result in pure strategy hence a slighty more complex table A.6 gives the results. An explanation for all the tables is also given in the appendix.

There are several important phenomenon of the Nash Equilibrium policies of the simple games that occur as the number of rounds increases. A summary is given here and each in phenomenon is discussed in turn:

- The policy does not change for the end rounds for the different games
- The prices selected within a round start to cycle (period three rounds) for games of five rounds or greater.
- The expected returns for each player can be represented as a simple formula for games of five rounds or greater.

As the tables in appendix A indicate, the prices selected by the player's Nash Equilibrium remains the same for all but the first round of a game, for all games with at least that many rounds for each of the different tie-breaker policies. The policies represent the best the players can do in those later rounds. What is surprising is that neither player attempts to unhinge the trail of price selections which lead to these later rounds. For example, it might be expected that P2 plays prices such that it draws P1 away from playing the low *five* price at the start of the penultimate round hence allowing P2 to gain more reward. However, P2 is unable or unwillingly to do

[15]If this were not true then a player could increase their expected return by playing the price with the highest expected return all the time and therefore increase their overall expected return. Hence the original mixed strategy was not a Nash Equilibrium.

just that (probably because the cost out-weighs the benefit). Hence a regimented set of price selections is obtained as the game size increases.

This regimented price selection transforms into a cycle when the number of rounds is increased past five. Apart from the initial round, the players price selections cycle around three numbers (i.e. Five, nine and ten for HIGH Nash Equilibrium policy). This cycle is the same for both players and results in both players receiving the same return for those three rounds where the cycle takes place (this can be seen in appendix A tables).

It is shown in the tables in appendix A why this cycling occurs. Taking, for example, the HIGH Nash Equilibrium policy for a large enough game. Table A.4 shows that expected returns from P2's play in the ninth from last round are exactly 19 better than the expected returns from P2's play in the sixth from last round. This means that the best response choices of P1 in the ninth from last round will be exactly the same as the choices for P1 in the sixth from last round, therefore the expect returns will be the same. This then repeats for P2's best response choices in the seventh and tenth round respectively, then for the eighth and eleventh rounds and then for the ninth and twelth rounds. Hence it is derived that rewards obtained in the twelth round are exactly 38 (19 + 19) different from the rewards obtained in the sixth round. The best-response prices keep cycling in this fashion and thus the policy keeps cycling. Similiar results are obtained for the LOW and RANDOM tie-breaker policies.

The extra reward obtained by both players over the three cycle rounds is exactly the same (19 for the HIGH Nash Equilibrium, 25 for the LOW Nash Equilibrium, and 18 for the RANDOM Nash Equilibrium). This implies symmetry between the players' policies. If the HIGH Nash Equilibrium is considered again, the players can be observed taking turns in *jumping* out of the price war (i.e. by choosing a price of *ten*). On closer observation, the selection cycle is *five*, *nine*, then *ten* for both players. This allows both players to obtain rewards of five, five, then nine (total 19) in a cycle. This can be explained.

Consider a round in the middle of a very large game. The player that started the game is going to have very little effect on that round (as there will have been many

customers to attract on the way) and who ends the games (again, there are more customers to attract than just the last few). This means who ever starts the game (i.e. P1) will have little influence on the policy for the middle rounds, hence both players are likely to adopt the same strategy. Thus the stable state for the pricing strategies of the players has been found.

It is noticeable that the initial policy of P1 does not necessarily follow this cycle in policy. This is due to the different structure of the first round. However, as the rewards obtained from the cyclic policy also cycle (but with a fixed step increase), the observed variation in the initial policy of P1 is always the same. For example, in the HIGH Nash Equilibrium, the abnormal initial policy is always three, see table A.4 for details.

One consequence of players following a cyclic policy before the last five rounds is that a formula for the expected returns obtained for any sized simple game can be derived. Though these formulae look complex they simply represent the cycle of rewards obtained as the rounds increase.

Given the game has $n \geq 5$ rounds and set $x = (n \bmod 3)$, so x is the remainder of $n \div 3$. x is needed within the formula because there is not a fixed step change in return as n increases but the same step increase occurs every three rounds. Let's look at the formulas for the HIGH Nash Equilibrium policy's returns. For P1, there arise step increases of 3, 11, and 5. The formula for P1 is:

$$3n + 10\lfloor\frac{n+1}{3}\rfloor - 8x(x-2) + 3$$

For P2, the step increases are 5, 5, 9. The formula for P2 is:

$$5n + 4\lfloor\frac{n+1}{3}\rfloor - 2$$

The values that are generated by this formula (for $n \geq 5$) are in the top rows of A.1. Now consider the formulae for the LOW Nash Equilibrium policy's returns. For P1, there are step increases of 8, 4, and 13. The formula for P1 is:

$$4n + 13\lfloor\frac{n+1}{3}\rfloor - 4x(x-2) + 2(x-1)(x-2) - 10.5$$

For P2, the step increases are 9, 8, 8. The formula for P2 is:

$$8n + \lfloor\frac{n+1}{3}\rfloor - 9.5$$

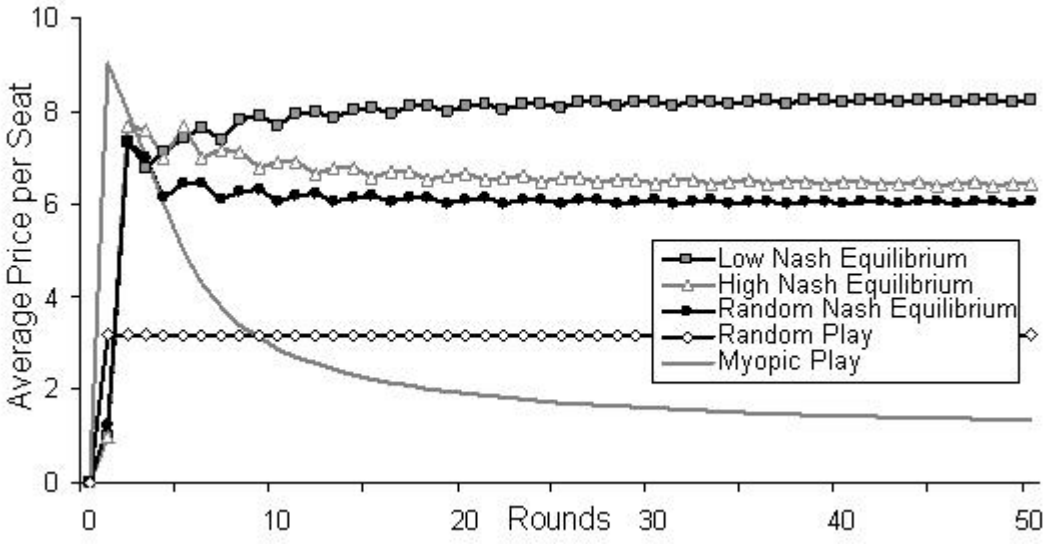

Figure 4.3: Average price of seats sold under different policies

The values that are generated by this formula (for $n \geq 5$) are in the top rows of A.2. Finally consider the formulas for the RANDOM Nash Equilibrium policy's returns. For P1, there are step increases of 5, 3, and 10. The formula for P1 is:

$$3n + 9\lfloor\frac{n}{3}\rfloor + 3.5x(x-1) + 1$$

For P2, the step increases are 8, 5, 5. The formula for P2 is:

$$5n + 3\lfloor\frac{n}{3}\rfloor - 2$$

These formulae have been included for interest only as they are not required for the learning experiment. However, they have allowed generation of the data required for figure 4.3 and calculation of the expected rewards of simple game with a million rounds is possible if required[16].

Average Seat Price

The different Nash Equilibria that are possible also have an impact on the prices which the seats are sold for. By looking at the average price for which the seats are sold, the *social benefit* of the different equilibria can be determined. Figure 4.3 shows the average price of the seats sold under the different Nash Equilibrium policies against the number of rounds within the game. The results when the players use a myopic

[16]Which are 6,333,341 for P1 and 6,333,330 for P2 when using the HIGH Nash Equilibrium

policy (see table A.3) and the completely random policy (i.e. players randomly choose their prices) are also included.

Myopic play is short-sighted play or excessive greed play. A myopic player will only consider their next reward opportunity thus will attempt to snatch the next customer from their opponent. This leads both players to conduct an iterative price war down to the minimum price of *one* (hence the average price of one). In completely random play, the players ignore all information and assign equal probability to each action (including zero). Consequence of random play is that both players observe the same return. This occurs because each customer always faces a random selection of prices so there is no bias in the system.

As figure 4.3 indicates all the average seat prices converge to fixed values, the converged values are as follows: the HIGH Nash Equilibrium value is $6\frac{1}{3}$, the LOW Nash Equilibrium value is $8\frac{1}{3}$, the RANDOM Nash Equilibrium value is 6, the completely random policy is $3\frac{2}{11}$, and the myopic value is 1. This shows that the airline players would be better off playing randomly then aggressively (i.e. myopic). All the Nash Equilibria do better than the two standard policies but some do better than others. Surprisingly the RANDOM Nash Equilibrium does the worst. This is due to the uncertainty that RANDOM tie-breaker policy brings to the opponent, hence they tend to play conservatively (see table A.6 for details). It might also be surprising at first that the LOW Nash Equilibrium does so well. This is due to the threat from both players to continue the pricing war so the prices remain high.

It is difficult to decide which phenomena are due to the models' setup and which are real *truths*, without conducting a large degree of sensitivity analysis. However, all three Nash Equilibria display similar characteristics (i.e. cyclic patterns, price choices that are consistent as the number of rounds increases, etc.), therefore it can be concluded that the Nash Equilibrum are not due to some complexity effect within the model framework[17]. Thus the model framework and Nash Equilibrium are adquete for experimental purposes.

[17]This complexity could be explain as just a consequence of the customers always choosing the lowest price.

Nash Distribution

Throughout the current section different Nash Equilibria for the simple games have been discussed. As mentioned in the methodology chapter, the learning players are expected to find the *Nash Distribution* policy. The Nash Distribution policies have similiar properties to the Nash Equilibrium policies but are *perturbed*[18]. The temperature parameter τ determines how perturbed the policies are. As $\tau \rightarrow 0$ the Nash Distribution policies tend towards the RANDOM Nash Equilibrium. As $\tau \rightarrow \infty$ the Nash Distribution policies tend towards the completely random policy. To investigate the effect of the temperature parameter on the Nash Distribution policy, the returns obtained by both players can be looked at. This effect can be seen on the simple 233 game's expected reward in figure 4.4.

The figure shows a series of graphs depicting probability distribution of the player's expected returns under different policies. The first graph shows the reward obtained when the players are using the RANDOM Nash Equilibrium policy. As discovered, this policy gives a return of 14 for P1 and eight for P2. No other returns are possible hence why both returns have a probability of one. If the Nash Distribution with any value of $\tau < 0.002$ was looked at then the graph would look exactly the same to the human eye. The reward distribution is not exactly the same, as the Nash Distribution policy is perturbed, however the probability of observing an outcome of any value other than 14 (for P1) or 8 (for P2) is so small that it cannot be picked up by the human eye on a graph. It is forgivable to think of the policy generated at these low temperatures to be the same as the RANDOM Nash Distribution policy.

The final graph in figure 4.4 depicts the policy under random play. This is not an even distribution because the customers still get to choose the lowest price offered by the players, thus there is a bias towards the lower end of the return spectrum. The peak at the zero return is due to the high chance that all customers will be able to purchase their seats at the *zero* price (this happens about a quarter of the time with the random policy). All possible returns have a chance of happening under the com-

[18]Perturbed means that the Nash Distribution policy assigns a posistive probability to every possible action.

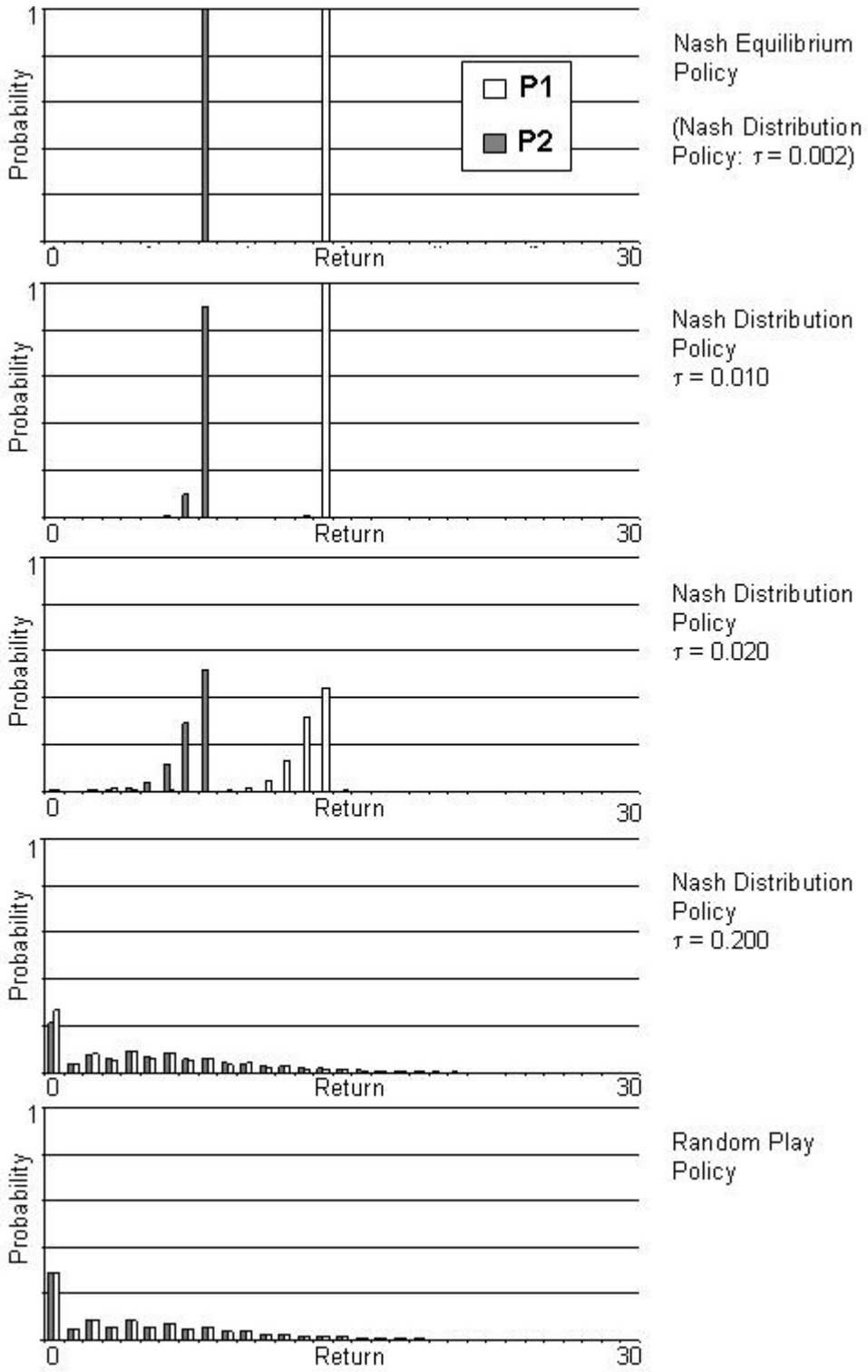

Figure 4.4: Graphs depicting the change in expected returns as temperature parameter varies for the standard 233 game

pletely random policy, even a player receiving a maximum return of thirty[19]. Both distributions are exactly the same for both players. This is because neither player takes advantage of choosing the price first (or choosing the price last) and there is no bias by the customers towards the randomly chosen prices.

The remaining three graphs in figure 4.4 shows the steady transition of the Nash Distribution policy from RANDOM Nash Equilibrium policy to completely random policy as tau increases. It is surprising how quickly this transition occurs, it is seen that the Nash Distribution policy relating to $\tau = 0.20$ already looks very similiar to the completely random policy. Next is a more detailed look at the changes in expected reward as tau increases.

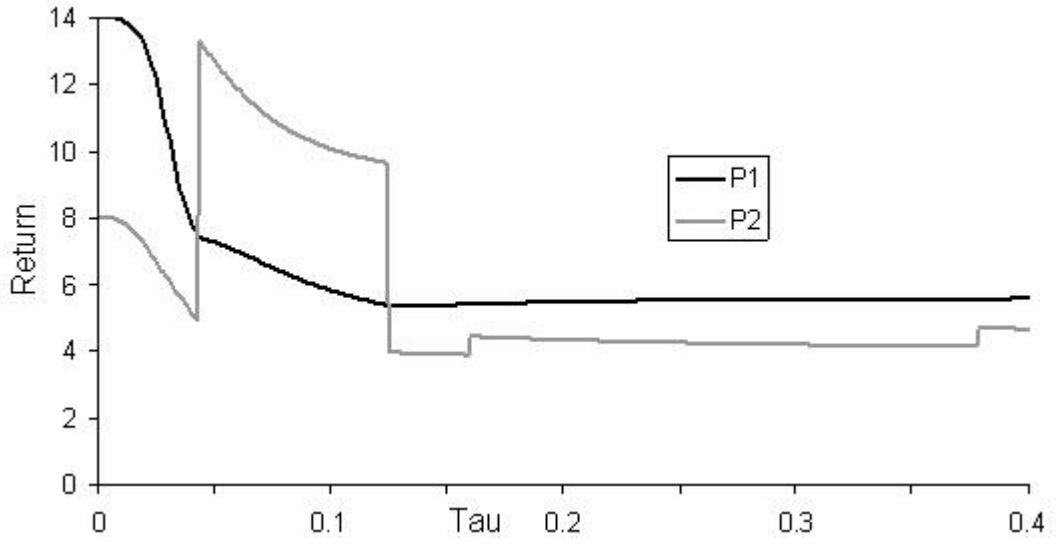

Figure 4.5: Returns obtained under from the Nash Distribution policies for the standard 233 game assuming that the players play their best responses in the first round

Figure 4.5 gives more of an indication of what is happening as tau is changing [20]. Instead of looking at the expected return obtained under the policy (which will decrease with an increase in the temperature parameter, meaning that sub-optimal actions will be chosen more frequently), figure 4.5 looks at the expected reward under

[19]This would occur when the players choose *ten* for all their price choices and all three customers happen to all go to one of the players. The chance of this happening is about 1 in 120,000.

[20]In chapter five, figure 5.1 contains the degradation of the player's return as the temperature parameter 'τ' is increased. However, this graph does not give a clear indication of what is happening to the policy as the temperature parameter is increased.

the current best response pair from the first round. That is, it is assumed that the action that will give P1 the highest expected return is played in round one and that P2 chooses the action which is the best response to this action (for their round one choice). Assuming this fixed choice in round one, an understanding can be developed of how the policy changes as the temperature parameter changes.

The noticable features of figure 4.5 are as follows: firstly, even though the graph shows P1's best initial action, a general decrease in P1's return is expected because as tau increases the players' remaining choices are more perturbed. P1's changes are smoother than than P2 because as the lead player P1 determines the game (in a Stackelberg type way). The graph shows when P1 decides to play a different initial price and P2 responding. This leads to major jumps in return occurring for P2 only. Details of turning points are in the table in appendix A.

The first major jump (at $\tau = 0.043$) occurs when P1 can no longer justify forcing P2 to play high in the first round, so switches to the unsophisticated myopic policy. The actual change in initial rounds policy goes from (x = 2, y = 10) to (10,9), where x is round one's best response policy from P1 and y is round one's best response policy from P2 (assuming that the remaining actions are chosen using Boltzmann Action selection)). The second jump (at $\tau = 0.125$) is more complex, it is a policy change from (10, 9) to (3,2). At such large tau values, P1 will expect P2 to be playing almost randomly (see graph 4.4 for details). P1's best response price of *three* is taking this random play into account. The remaining jumps are quite small and can be ignored.

The last phenomenon in figure 4.5 is in the high values of the temperature parameter and is typical in Game Theory. As the randomness of selection increases (as tau is increasing), there is a slight increase in reward. This is because the players decisions (i.e. their underhandedness) start to have less effect on what actions are actually observed in the game (because of the randomness caused by large tau), so the return converges to the random policy for both players (this phenomenon is seen in figure 4.4)

Summary of Policies

Within this section the properties of different *Nash* policies to the simple versions of the game (primarily the simple 233 game) have been considered. From these simple games complex behaviour has been observed. Now the mechanics of the learning model which will be used to generate learnt policies and that also display this complex behaviour are considered.

4.4 Learning Model

In the methodology chapter, the framework of the learning model was discussed. Within this section, the necessary steps that are required to implement this framework are considered. Though most of the learning model has already been defined, this section looks at two aspects that require further attention: the starting values and the learning parameter.

Apart from the *physical* parameters of the model (i.e. number of rounds, seats, etc.) the only other parameters are the temperature parameter (tau) and the learning parameter (λ). The temperature parameter remains fixed within the model[21]. Different values of the temperature parameter were assigned to different runs of the games and the impact this has within the empirical results was observed. The learning parameter, however, varies within each run of the learning model and has to take a certain form (see the convergence proof chapter and section 2.4 for more details). There are multiple learning parameters to consider as well as each Q-value within the game having its own learning parameter. This means that the temperature parameter here must be defined.

The form the learning parameter needs to take is from the convergence proofs. So if n is the n-th time that a Q-value has been updated and $B > 0$ then the lambda for the Q-value is as follows:

$$\lambda = \frac{1}{n + B}$$

[21]The temperature parameter could be setup to vary over an episode (or episodes). However, as the temperature parameter determines the Nash Distribution the learning policies should converge to, changing the temperature could result in divergence.

Throughout the runs a value of $B = 1$ was used, this allowed the Q-values to gain as much influence from the initial rounds of play as possible with lambda still remaining well defined. By allowing the Q-values to be more affected by the returns observed in the early rounds, the ways the policies were changing could be seen. Other versions of lambda were experimented with, this is discussed in the previous experience section (section 7.4). This version of lambda fulfilled all the criteria required and seemly did what was required.

Another reason for wanting the Q-values to be changed by the initial episodes was so that they move away from their arbitrary initial values. These initial values of the Q-values (and hence the policies) could have been set in several ways. The most common approaches are: *randomly, heuristic knowledge, and optimistic starts.* No matter which approach is used, it will leave a permanent bias on the Q-values and hence the policy (though this decreases with time). This bias is unavoidable but can be soften by using vaguely realistic initial values.

A random approach ensures that the learning players are unaffected by any bias by the modeller, however this method can slow the rate of convergence. The heuristic knowledge approach allows the modeller to influence the initial policy of the learning player by inserting their own understanding of the Q-values (and policy) into the game. This approach would seem useful for increasing the rate of convergence assuming that the modeller's knowledge is correct. As discussed earlier in the chapter, the policy of a game is not necessarily simple (even for simple games) so this approach could hinder convergence if incorrect ideas about the policy are used.

The final method (and the one used here) is called *optimistic starts* (see Sutton and Barto, 1998). Using the method, every Q-value is set to the largest that could possibly be observed from its corresponding state (i.e. if the player has only one seat left, then it was assumed that the seat is sold for the maximum price of *ten* and set the corresponding Q-values to that value). The optimistic starts method encourages exploration of the state space as actions that have not been selected will appear to be worth more (as their Q-values are so high) and hence be more likely selected (by the Boltzmann Action selection method).

This chapter has so far been concerned with an abstract model. In the next section

the issues relating to the physical implementation of the model are considered.

4.5 Programming Code

Though it has not been explicitly said within the thesis so far, a computer was used to calculate almost all of the empirical results. Given the model setup, it could have been possible to work out the results by hand, though given the quantity of results that required this was infeasible. By working the results out through mental arithmetic, it would have subjected the results to errors through both calculation mistakes and rounding errors (i.e. the time required to generate a pseudo-random number by hand to the same number of digits as a computer generated one would be impractical).

The model was constructed using a programming language (i.e. Visual Basic for Applications (VBA) and C++) as opposed to a standard simulation package (e.g. Simul8). There are a number of reasons for this choice but the main ones were flexibility and speed considerations. It was also not possible to construct the model within an existing simulation package because of the complications that the Reinforcement Learning would have caused. This would have meant that the Reinforcement Learning calculations would have had to be run in a separate programme and the results fed into the chosen simulation package. This communication between programs would have been slow and difficult to implement. Using a programming language meant that all necessary calculations could be directly embedded within the program.

Using a programming language to implement the model was not without its draw-backs. The main draw-backs from using a programming language are the lack of visualisation and uncertainty that the model has been verified. Verification is discussed later in the chapter. In this section the focus is on the elements relating to the programming language and source code. All computer storage (memory) and time issues are discussed in the empirical results chapter

Selection of Programming Language

The learning model might be too complex for standard simulation packages like Simul8 and Oracle, however the *Python* simulation language could have been used. Though

Python has been designed for constructing simulations like this model its compiler was not available on the University of Southampton's computational facilities[22]. The technical skill required to install the Python compiler onto the computational facilities mainframe was far beyond anyone involved in the research and the suite's manager, hence a programming language with an existing compiler on the suite was chosen.

The feasible possibilities were C++, Visual C++ and Visual Basic. The model was originally constructed using Visual Basic for Applications (VBA) combined Microsoft Excel application. Though this did give input-output benefits for running and exploring the results, the application was very slow and there was also a tendency for the program to crash. From this prototype, it became clear that speed and memory management were going to be an issue. As the C++ language has excellent memory management features, this programming language was used. The model runs using C++ were at least three times faster than identical runs using the VBA code.

The University of Southampton's computational facilities had several C++ compilers available and a GNU compiler (called *G++*[23]) was used for the modelling. This allowed the program to be written in the standard Microsoft Windows platforms using the Dev-C++ Integrated Development Environment (IDE) freeware [24] from Bloodshed Software (see Bloodshed Software, 2005), which was derived from the G++ compiler (hence compatibility problems were avoided). For transportability between the platforms (i.e. Microsoft Windows and Linux) the International Organisation for Standardization (ISO) standard C++ language was used for all the C++ programs. This use of standards was important to avoid compiler errors that can occur due to slight differences between the platforms compilers.

Pseudo Code

The several thousand lines of code that were written to implement the model have not been included for conciseness and clarity reasons. Including a description of all

[22] which is called *IRIDIS cluster*.

[23] Version: 3.2.3 (released April 2003). Compiler from the GNU Compiler Collection (GCC) freeware ((GNU Project, 2007)).

[24] Version: 4.9.9.2 (released February 2005). This software uses the Minimalist GNU for Windows (MinGW) port of the GNU Compiler Collection (GCC) freeware (see GNU Project, 2007).

State $\langle n, s^1, p^2 \rangle_1$ and $\langle n, s^2, p^1 \rangle_2$

Initialize $Q^1(\langle n, s^1, p^2 \rangle_1, p^1)$, $Q^2(\langle n, s^2, p^1 \rangle_2, p^2)$

Repeat (for each episode):

Select state $\langle 1, S^1, 0 \rangle_1$

Choose p^1 for $\langle 1, S^1, 0 \rangle_1$ using policy derived from Q^1

Select state $\langle 1, S^2, p^1 \rangle_2$

Choose p^2 for $\langle 1, S^2, p^1 \rangle_2$ using policy derived from Q^2

Take actions p^1 and p^2, observe $\acute{r}^1, \acute{r}^2, \acute{s}^1$

$\acute{p}^2 \leftarrow 0; n \leftarrow 1; r^1 \leftarrow 0; r^2 \leftarrow 0; s^1 \leftarrow S^1; s^2 \leftarrow S^2$

Repeat (for each round in episode e):

Select state $\langle n+1, \acute{s}^1, \acute{p}^2 \rangle_1$

Choose $\acute{p}^1$ for $\langle n+1, \acute{s}^1, \acute{p}^2 \rangle_1$ using policy derived from Q^1

If $\acute{s}^1 = 0$ then $\acute{p}^1 =$ **void price**

$Q^1(\langle n, s^1, \acute{p}^2 \rangle, p^1) = \left(1 - \lambda_e(\langle n, s^1, \acute{p}^2 \rangle)\right) Q^1(\langle n, s^1, \acute{p}^2 \rangle, p^1)$

$+\lambda_e(\langle n, s^1, \acute{p}^2 \rangle). \left(r^1 + \acute{r}^1 + Q^1(\langle n+1, \acute{s}^1, p^2 \rangle_1, \acute{p}^1)\right)$

$s^1 \leftarrow \acute{s}^1$

Take actions $\acute{p}^1$ and p^2, observe $r^1, r^2, \acute{s}^2$

Select state $\langle n+1, \acute{s}^2, \acute{p}^1 \rangle_2$

Choose $\acute{p}^2$ for $\langle n+1, \acute{s}^2, \acute{p}^1 \rangle_2$ using policy derived from Q^2

If $\acute{s}^2 = 0$ then $\acute{p}^2 =$ **void price**

$Q^2(\langle n, s^2, p^1 \rangle_2, p^2) = \left(1 - \lambda_e(\langle n, s^2, p^1 \rangle_2)\right) Q^2(\langle n, s^2, p^1 \rangle_2, p^2)$

$+\lambda_e(\langle n, s^2, p^1 \rangle_2). \left(r^2 + \acute{r}^2 + Q^2(\langle n+1, \acute{s}^2, \acute{p}^1 \rangle_2, \acute{p}^2)\right)$

$s^2 \leftarrow \acute{s}^2$

Take actions $\acute{p}^1$ and $\acute{p}^2$, observe $\acute{r}^1, \acute{r}^2, \acute{s}^1$

$p^1 \leftarrow \acute{p}^1$

$p^2 \leftrightarrow \acute{p}^2$

$n \leftarrow n + 1$

Until either $n = N + 1$ or both $\{s^1 = 0$ and $s^2 = 0\}$are terminal

Table 4.2: Pseudo code for SARSA reinforcement learning

the elements that went into constructing the model (i.e. file management etc.) is not intended, and will only briefly touch on them here. Pseudo-code has been included to give a flavour of the model, which can be seen in figure 4.2.

The pseudo code in figure 4.2 represents the process of learning within the model using the SARSA method. This psuedo code is based around the code presented in Sutton and Barto's reinforcement learning book (Sutton and Barto, 1998) and is an embodiment of the SARSA method described in the literature review chapter. An explanation is not given here of the pseudo code but of its complexity[25] justifies to the reader why a description of the complete program has been omitted. However, there are some aspects of the program that require justification and cannot be avoided. The first of these is the pseudo-random number generator of the program.

Pseudo-random Number Generator

It is estimated that several trillion random numbers were generated for the empirical results. With this many random numbers required a pseudo-random number generator was needed that would not display any obvious pattern or cycle within the numbers generated, and would fit within the C++ language framework.

A C++ freeware library file called 'mtrand.cpp' written in ISO Standard C++ (Bedaux, 2002) provided the pseudo-number generator for all the computer models. This code provides a Mersenne Twister pseudo-number generator, which has a period of $2^{19937} - 1$. This level of randomness exceeded the requirements (which were less than 2^{70} random number generations). For more information on the Mersenne Twister pseudo-number generator see Matsumoto and Nishimura (1998). Each run had a new seed generated by the computer system's clock and it was assumed that the numbers generated were random enough for the purpose[26].

The Mersenne twister is related to the Mersenne numbers which were found by the French mathematician Marin Mersenne in 1644 (see Jones and Jones, 1998). Mersenne

[25] Actually, even this pseudo code is a simplified version of the final code.

[26] It was not possible to store a large quantity of generated numbers to check the randomness. Even if it was possible to store the numbers, the computational requirements to check the randomness of the numbers would have been excessive.

numbers are those of the form $2^p - 1$, where p is prime. They have great importance within number theory.

Rounding Errors

Another issue faced within the computer code was rounding errors. Throughout the programming code the most accurate data type available was a *long double* [27] . Though the *long double* could store values upto $1.1e + 4932$, cases were found where this was not good enough.

When using the Boltzmann action selection method, $e^{\frac{1}{\tau}}$ is required to be worked out. Given the limitations on *long double* this meant that it was impossible to calculate for $\tau < 0.00001$ (this would have been $\tau < 0.001$ if VBA had been used). This was slightly disappointing because there is a desire to generate results with smaller Tau values (as their corresponding Nash Distributions would be very similar to the Nash Equilibrium). This physical limitation was accepted and did not affect the model (as learning policies with such low temperature parameters rarely converged).

The discussion about computer code is left there, though other subjects could have been discussed (i.e. single array updating). The focus is turned onto the verification of the model. Incorrectly written code (or *bugged* code) will produce incorrect results, hence why verification was of upmost importance. This and validation of the model are discussed in the next section.

4.6 Verification and Validation

There have been several suggested ways, within the literature, of how to validate an OR quantitative model. The validation work of Mike Pidd, found in his book *Tools for thinking: Modelling in Management Science* (Pidd, 1996) was focussed on for this research. There have been other methods suggested over the years (see Yoshizaki and Plonski, 1995; Brooks and Tobias, 1996), however Pidd's work aims to collect these methods into a single body of work. In Pidd's book, he suggests that they are two main types of validation: *black-box* and *open-box*[28].

[27] This data type uses 10-bytes of computer memory. The normal *float* only uses 4-bytes

[28] Sometimes called *White-box*

Black-box validation is concerned with the data input to and output from the model. This input/output data is compared to a set of theoretical results or *real-world* data determining if the model correctly explains the relationship between the data types. This processes of validation is not appropriate for the modelling because neither theoretical results or real-world data are available to compare the model's output to. Thus an open-box validation for the model must be relied upon.

Open-box validation relies on justifying the construction of the model and its inputs. The argument of this type of validation is that if the model and inputs are correct, then the outputs are correct, thus no further justification of any results is required. This method of validation is appropriate for the research.

To justify the model using the open-box method, its construction had to be justified. The whole of this chapter (and the part of the methodology chapter) discusses the issues with the model and therefore, acts as the justification of the model. However, the construction of the model was not without criticism (i.e. only using one game to compare the RL techniques with, the means with which the policies are compared, etc.). Re-addressing these criticisms must be done when drawing any conclusions from the models results.

Some of the validation of the model comes from re-applying methods from previous research. Endeavours to complete this task were conducted and where the existing literature was not followed; justification of these discussions was attempted. However, ultimately the validation of the model (and method) using open-box validation rests with the reader, whom is intended to be satisfied by these arguments.

Verification

Verification tends to be the *ignored little brother* of validation, were most practitioners tend to assume that it was done within any presented results. It is estimated that 90% of programming time is spent verifying (or debugging) the code (Liberty, 1999) and in a large program it is unlikely that every *bug*[29] has been found. This means that considerable effort was placed on verifying the model and thus validating that

[29] A bug is a error within the written code which causes the computer program to not run correctly.

the correct outputs were acquired are described below. Considering the open-box validation approaches to this research, verification is of the up most importance as it is not clear what outputs would be expected. For the rest of this section verification was discussed.

A model run can be verified by outputs that are produced. Ideally, all the data produced by every single action of the executable program would be stored. However, this was impossible as there would have been tetra-bytes worth of information being produced per run. This means that only certain amount of information was stored. A decision was taken that only policies and statistical results would be the output. This lack of detail means that it is difficult to follow an audit trail of runs to verify it.

Given the vast amount of data output, it would have be virtual impossible have tried and verified all of it anyway. Each Q-value, for instance, would have been derived from several thousand random calculations. Therefore, the verification for the results took place in three phases:

- Extensive testing of each component of the C++ code as it was constructed
- Step-by-step checking of the complete programs for a small scale case (i.e. only ten episodes)
- Internal testing of consistency of data values within the executable program (i.e. bullet-proofing the code)

Even with these checks in place, it is unlikely that the executable program was completely bug-free due to the size of the program used. Not all possible checks were conducted within the program as this would have increased the runtime considerably. Therefore, when unexpected or anomalous output were obtained, explaining them with logical argument was relied upon. However, not all anomalies have been explained and where appropriate further rigorous tests of the C++ code have been done. Within any stochastic system these anomalies would be expected and where appropriate it has been highlighted.

Some outputs were easier to verify due to having exact solutions to compare to. The Nash Distribution and Equilibrium policies were verified this way using the solutions

given earlier in this chapter. However, all of the learning output could not be verified this way and other verification measures had to be employed. One method was to construct the model using the VBA programming language as well as C++ programming language. By constructing two models, it was possible to compare results from the different models to see if similar (not exact as the model is stochastic) results were produced. This was done for a variety of different runs and any bugs found were removed. This process was repeated until both models displayed similar results.

One aspect of the model's output that was difficult to verify was the rate at which the policies converged. The theoretical proofs (in chapter six) show that the SARSA learning models will converge eventually but did not consider the rate of convergence. Proving the rate of convergence is an open problem within Reinforcement Learning. Therefore, it was deemed unnecessary to prove (thus verify) the rates of convergence as this can be parodied as using a *sledgehammer to crack a walnut.* As the convergence rates cannot be verified, care must be taken when making any conclusions about them.

This system of verification (by considering the outputs) has left the results open to type II errors (i.e. when results are accepted which should have been rejected). It is possible that bugs within the code could lead to favourable results. However, with the comparisons of the two models (VBA and C++ versions), this is highly unlikely and can be ignored.

These methods of verification were considered adequate for this purpose but it was not *bullet-proof* (i.e. every possible bug was checked for within the code). As mentioned, to bullet-proof the code would have slowed its run-time performance considerably. This trade-off of run-time performance versus verification is common within all programming and an appeal to the results as the justification of this.

Once the model was constructed and verified, empirical results were generated. These can be found in the next chapter.

Chapter 5

Empirical Results

5.1 Introduction

So far the focus has been on the theoretical aspects of the model. This included constructing a model framework and finding the Nash Equilibrium for various versions of the game. In this chapter the practical aspects of the model are considered. This includes looking at the computer constraints of time and memory. Where appropriate, the numerical results obtained are given in the results appendices.

In this chapter, the simplest game is analysed in detail before moving onto a more complex version of the game. Other variations on the game are considered in the variations Chapter Seven. There are three reasons for analysing such a simple game. Firstly, some interesting results occurred even though the game was so simple. Secondly, the small size of game meant a relatively short run-time; thus allowing lots of sensitivity analysis (i.e. variations in the temperature parameter) to be generated. Thirdly, the simplicity of the game allowed observation of very low *measure* values (i.e. policies converging as expected).

The learning model's runs generated a lot of data, especially in relation to the learnt policy. To summarise and compare this data a measure is used. The first part of this chapter is devoted to choosing this measure, using the comparison of the Nash Distribution policies to the Nash Equilibrium policies as a test case.

Summary of Empirical Results

Once a measure has been selected and analysis is completed for the simple model, more complex models can be considered. The complexity of the model is increased by increasing the number of rounds observed within an episode. The results are divided into various sections as follows:

- Comparison of Nash Distribution to Nash Equilibrium
- In-depth analysis of simple 233 game results
- Results from increasing the number of rounds
- Physical limitations of modelling

Due to space limitations, it would be inappropriate to present all the empirical results that were found. Therefore, the results presented here are designed to highlight the key findings. Where further detail is required, the numerical results can be found in the appendices.

5.2 Nash Distribution

This section is about an investigation of the Nash Distribution and also of the measurement methods that could be used in the analysis. The reason for doing this was to put the learning-run results in context. By understanding how the different measures compare the Nash Equilibrium to the Nash Distribution, the learning results measurements had a baseline for comparison. These Nash comparisons were not used to define the cut-off (or bench mark) point for convergence. This would have been inappropriate, as discussed later.

The standard game was the simple 233 game and it seems appropriate to spend some time considering the results from this game. The results were generated using several steps. Firstly, the Nash Equilibrium and Nash Distribution polices were worked out using the dynamic programming method described in Chapter Four. Secondly, the return distribution was evaluated from the policies being played. This meant evaluating the return distributions of when both players used the Nash Equilibrium policy

and the equivalent return distributions for the Nash Distribution policy. It is important to note that when the Nash Distribution policy's return distribution was generated, action selection was allowed to still be perturbed by the temperature parameter (tau). Finally, the distance between the two distributions was calculated using the different measures (described below).

The Nash Distribution generated return distributions were compared to the myopic and random policies generated return distributions as well. For more discussion on these policies, see Chapter Four. This means that the return distribution generated from playing the Nash Distribution policy versus the Nash Distribution policy was measured against the return distribution generated from playing the myopic policy versus the myopic policy and it was also measured against the return distribution generated from playing the random policy versus the random policy. These comparison results are also presented with the Nash Equilibrium comparison results described above and all the results are summarised in figure 5.1.

In figure 5.1, the graphs have an x-axis of *tau* to indicate the different Nash Distributions considered, as all Nash Distributions are determined by the temperature value. The y-axis indicates the different measures. Starting from the top-left and working across then down there is: Kolmogorov-Smirnov (KS) statistic, Total Variation (TV) distance, Hellinger (H) distance, Adjusted-KS (AKS), Information Value (IV), Separation Distance over Theoretical Distribution (SD1), Separation Distance over Empirical Distribution (SD2), Chi-squared Distance over Theoretical Distribution (CHI1), Chi-squared Distance over Empirical Distribution (CHI2), Expected reward for P1 (E1) and Expected reward for P2 (E2).

Only shown are the results for the temperature parameter between zero and 0.2. Results for higher values of *tau* were collected (until *tau* equalled one hundred). All the measures' results continued to change smoothly and asymptotically, as would be expected from the graphs. For this reason and so that the interesting variation in the measures could be observed, the x-axis is truncated at the 0.2 point.

The graphs are made up of discrete points though they have been presented as a continuous line. The measures were calculated at 0.0001 intervals, with zero tau indicating the Nash Equilibrium policy (thus the Nash Equilibrium policy was compared

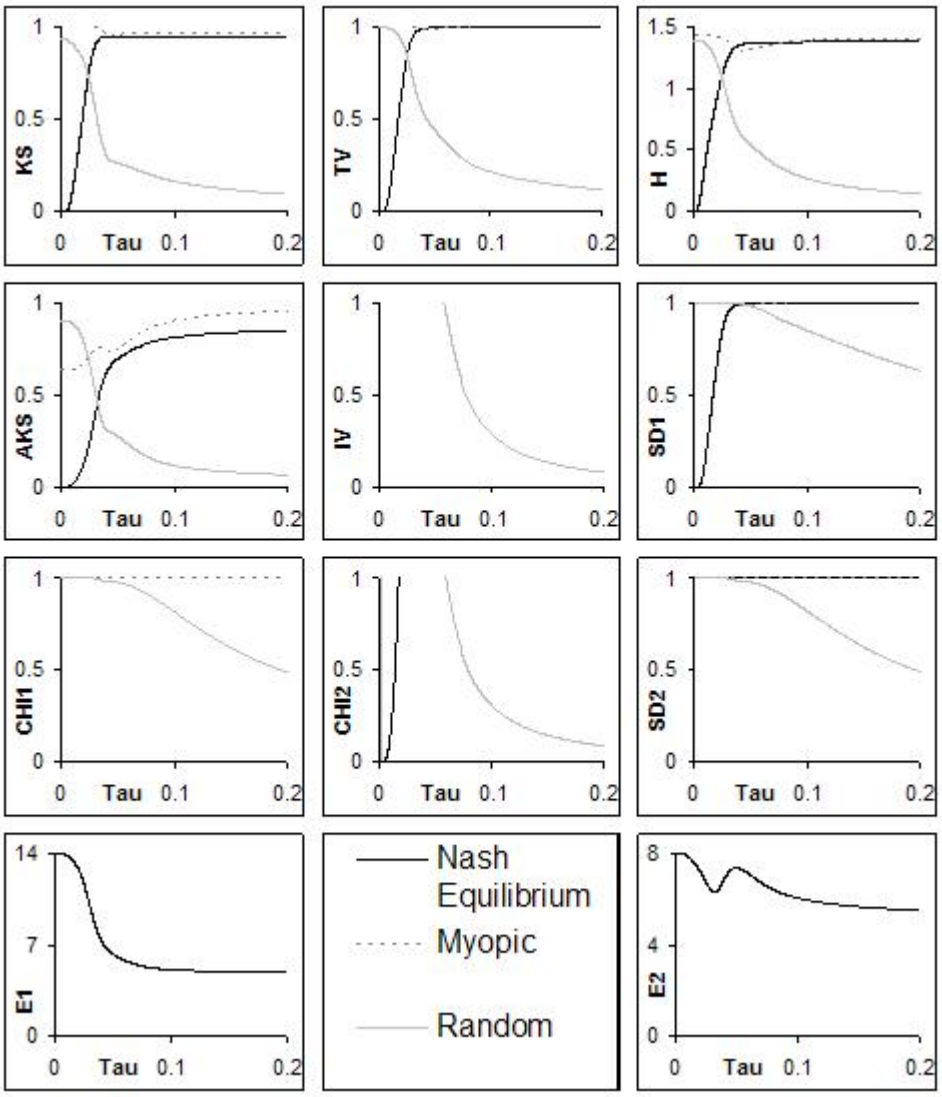

Figure 5.1: Graphs depicting the various return distribution measures of a Nash Distribution policy compared to various other policies, for the standard 233 game.

with itself). These small intervals were deemed adequate as rapid fluctuations within the results are not expected.

Not all of the three lines (they are the measures of the return distribution, generated by playing the Nash Distribution policy against the Nash Distribution policy, compared to the three return distributions generated by playing Nash Equilibrium policy versus Nash Equilibrium policy, by playing the myopic policy versus the myopic policy and by playing the random policy versus the random policy) are presented on every graph. This is due to the line's values being greater than the y-axis scale maximum of one (or the 1.5 in the case of the Hellinger distance). The exception of this is the expected value graphs, where only the expected value of the Nash Distribution is considered.

Before further discussion about the graphs can be done, the measures that were used must be defined.

Distance Measures

An explanation of the different measures, presented in figure 5.1, is given below. For each measure, an indication on how it was calculated and how the results may be interrupted from that measure is given. Greater discussion has been given to the measures that were deemed more important. Finally, an indication is given to which measure will be used of the remainder of the results.

Most of the measures calculated were taken from the review of probability metrics by Gibbs and Su (2002). Not all of the measures stated in the review were used because the reward distributions are discrete. Some of the measures suggested in the paper could only be used with continuous distribution (e.g. Levy metric) or were equivalent to other metrics for discrete distributions (i.e. the Discrepancy metric and the Total Variation distance). Other measures not included in the review were included (i.e. Expected value and Adjusted KS) here, the reason for their inclusion is given with their descriptions below.

The various different measures use different terms in their names like *distance*, *value* and *statistic*. Before the individual measures are discussed, the terminology should be defined.

The term *measure* means something that gives a size or quantity for comparison (which is called the *distance*). It does not mean a mathematical measure as defined within probability theory (see measure theory in Williams (1991)). Some of the measures considered do satisfy the requirements of a *metric* function, where this is the case it is stated within the measures definition. The definition of a metric m (Borowski and Borwein, 1989) is a bivariate function such that:

$$\begin{aligned} & m(x,y)0 \\ & m(x,y) = 0 \Leftrightarrow x = y \\ & m(x,y) = m(y,x) \\ m(x,z) + m(z,y) & \geq m(x,y) \end{aligned}$$

Another term that is used is *statistic*. A *statistic* is simply defined as just quantitative data on any subject (Borowski and Borwein, 1989) and was introduced by Fisher (1925).

For reference purposes, $u_{ab}(x,y) = P_{ab}(X = x, Y = y)$ is the return distribution function under P1 using policy a and P2 using policy b, where (X,Y) is the return pair observed from play. The marginal distributions are simply $u_{ab}(x)$ and $u_{ab}(y)$. $v_{ab}(x,y)$ is the return distribution of the policy pair being tested against. This notation is simplified to $u(x,y)$ and $v(x,y)$.

The cumulative distributions are defined as $U_{ab}(x,y) := P_{ab}(X \leq x, Y \leq y)$ and $V_{ab}(x,y)$ respectively.

The marginal cumulative distributions are as expected. There is no complete ordering of a bivariate pair, therefore it is hard to take into account when two pairs are close to each other (i.e. is (3, 3) closer to (1, 1) than (1, 5)? Different metrics will say different things). This is something to bear in mind during the rest of this section.

Outlined below are the metrics and statistics used to compare the return distributions.

Kolmogorov-Smirnov Statistic (KS)

Kolmogorov-Smirnov statistic is a metric and it looks for the maximum discrepancy between two cumulative distribution functions. It was originally proposed by Andrei

Kolmogorov in 1933 (see Kolmogorov, 1933). The Kolmogorov-Smirnov statistic is one of the few goodness-of-fit tests that is available and has application in areas like credit scoring (see Thomas et al., 2002). It is defined mathematically as follows:

$$d_{KS}(u, v) = sup_{(x,y)}|V(x, y) - U(x, y)|$$

The KS is the standard measure that is used throughout these results. There are three reasons for choosing it as the *flag-ship* measure. Firstly, it is always defined for any comparison and is bounded by zero and one, other measures suffer from not always being defined (due to division by zero errors). Secondly, it produces similar results to several other measures considered (i.e. Hellinger distance and Total Variation distance). This statement has both been observed empirically (see figure 5.1) and theoretically (see Gibbs and Su, 2002). Finally, results from the KS mainly show smooth changes, thus presenting easy-to-follow graphical results. This is shown in figure 5.2.

One criticism of the KS is that it does not take into account (proportionally) major differences within the tails of the return distributions. However, a relative small number of possible return pairs are dealt with and this effect from the return distribution tails is negligible.

As mentioned, the KS metric takes values between zero and one. When a value of one is observed this means that there is no overlap in the return pairs. This means that the two return distributions are unalike, indicating that the underlying policies are dissimilar. When a value of zero is observed, the return distributions are identical; however this does not indicate that the underlying policies are identical. What this does indicate is the return distributions are identical. This problem has already been discussed in the methodology chapter. It is intended that small KS values show closeness of the underlying policies.

Figure 5.2 is the KS section of figure 5.1 with the x-axis stretched out (and not including the comparison to myopic or random policy). The graph shows the KS distance increases as the value of temperature parameter increases. This indicates that polices of Nash distribution move further away from the Nash Equilibrium polices as temperature (or exploration) increases. These results are expected as the Nash Dis-

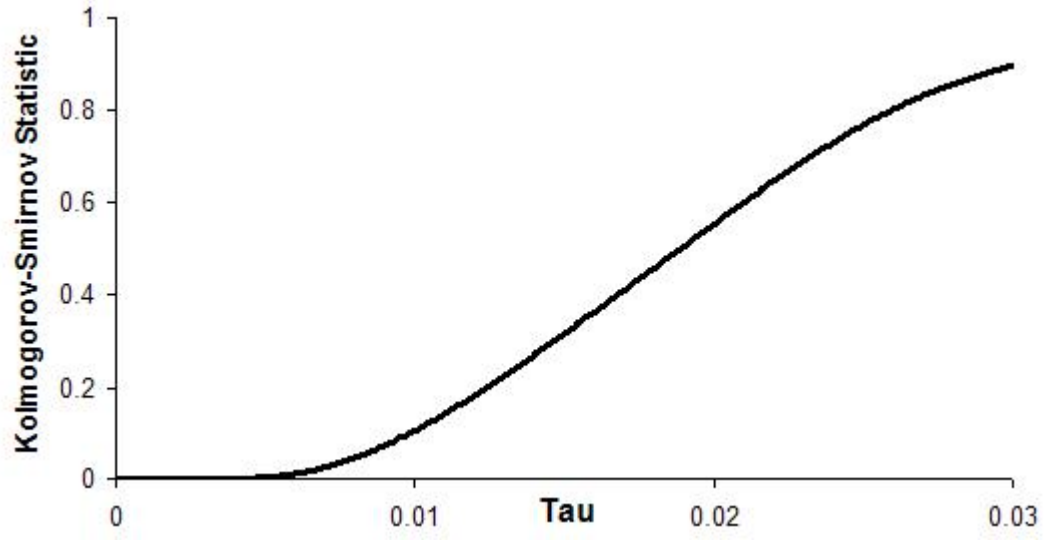

Figure 5.2: Graph depicting the Kolmogorov-Smirnov statistic for the comparison of Nash Equilibrium policies' return distribution to the Nash Distribution policies' return distribution, for the standard 233 game.

tribution policy has to take into account more random behaviour as tau increases, thus reacting to future events differently.

Other features of figure 5.2 are that the KS distance seems to be negligible up to about 0.005 and 0.5 when tau is equal to 0.02. From this, it is tempting to conclude that it is only worth considering tau at less than or equal to 0.005. However, other factors (like convergence rates) have to be taken into account before making such a generalisation.

Other comparisons were conducted against the myopic and random policies, shown in the top-left corner of figure 5.1. As the graph indicates, the Nash Distribution does not share any similarities with the myopic policy. This indicates that the Nash Distribution policy is not a myopic one and this was seen from the results presented in the model chapter.

The Nash Distribution policy becomes similar to the random policy as the temperature parameter is increased. This is indicated by the decrease in the KS distance as tau increases. This is an expected result as a higher temperature parameter means more randomness is present in the Boltzmann action selection hence it is more like

random play. It is interesting that the rate at which the Nash Distribution policies become more like the random policies is roughly the same rate that it becomes less like the Nash Equilibrium policy. This indicates that Nash Distribution policies are simply becoming more random as tau increases.

These phenonemon are seen in the other measure especially the *Total Variation*, which is discussed next.

Total Variation (TV) distance (and Discrepancy metric)

Total Variation (TV) is a simple metric, effectively half the L1-norm, taking values between zero and one. This can seen in the following formula:

$$d_{TV}(u, v) := 0.5 \sum_{(x,y)} |v(x, y) - u(x, y)|$$

The TV gives the same value as the Discrepancy metric for discrete return distributions. The Discrepancy metric is defined as maximum difference in probability achievable for any single closed subset of the return-pairs space. It can easily be shown that the subset $\{(x, y) : v(x, y) > u(x, y)\}$ maximises this difference in probability and gives the same value as the formula above.

This means that the TV gives the worst probability difference between the two return distributions, thus giving an indication of the worst case difference in the return distribution functions. Achieving a TV value of one means the return distributions are mutually exclusive and a value of zero means they are identical.

The TV gives similar results to the KS measure, both empirically and theoretically, therefore the Discrepancy metric definition is used as another interpretation of any KS result. The only difference that can be seen in figure 5.1 is that the TV distance tends to give a slightly smoother rate of change as tau varies. This is due to the TV taking into account the changes within all the individual return pair probability values.

Hellinger Distance (H)

Another metric considered is the Hellinger distance (H), which is similar to the L2-norm but scales down the original probability measures by square-rooting them. This

can be seen in the formula:

$$d_H(u,v) = \left(\sum_{(x,y)} \left(\sqrt{v(x,y)} - \sqrt{u(x,y)} \right)^2 \right)^{\frac{1}{2}}$$

The metric was discovered by polish mathematician Ernest Hellinger (see Borowski and Borwein, 1989) as a means to measure distances of multivariate distributions. As a bi-variate distribution is being considered this distance seems an appropriate one to use.

Again there are both empirical and theoretical (see Gibbs and Su, 2002) similarities between KS and H. The major difference is the H's upper bound is the square-root of two and the KS' upper bound is one. This means that the KS distances are very similar to two standard norm distances: the L1-norm (via TV) and the L2-norm (via H).

Adjusted KS (AKS)

In an attempt to deal with problem of KS ignoring the effects from the return distributions tail, a new metric was derived. This new metric looks at the total absolute difference between the two cumulative return distributions, unlike the KS which only looks for the maximum difference between them. This metric was called the *Adjusted KS* (AKS) and it is defined below:

$$d_{AKS}(u,v) = \sum_{(x,y)} |V(x,y) - U(x,y)|$$

This new metric is related to the Gini coefficient and the Receiver Operator Carrier (ROC) curve, which are used to compare cumulative distributions within signal engineering and credit scoring (see Thomas et al., 2002). The major difference the AKS has to these techniques is the weighting of each term.

There is a problem when trying to calculate this discrete bi-variate metric; there is a bias towards the lower values of the distribution. As mentioned before, the cumulative return distribution U(X, Y) is the summation of all values of the density function u(x, y) up to (X, Y). This means that low values of (x, y) will be included in a lot more cumulative return distributions then high values of (x, y). The problems could be overcome by using weights on the values or by only considering the return

distribution and not the cumulative return distribution (which is effectively the TV distance). However, it was decided that AKS was not a good indicator and to not pursue its development.

From the results obtained, it was observed that the AKS was less sensitive to change and tended to produce overly smooth graphs. This can be seen in figure 5.1, the values have been normalised to fit within the zero-to-one scale. Though a certain amount of smoothness in the results was desirable, the smoothness observed seemed excessive. Also the results did not seem necessarily strongly correlated to the other measures. These reasons meant the metrics development was abandoned.

Information Value (IV)

The Information Value (IV) is used within Communication theory (see Welsh, 1988) and has been applied in practical areas (e.g. Credit Scoring, see Thomas et al. (2002)). The metric is the difference between the Relative entropy (or Kullback-leibler divergence) statistics. The IV is calculated as follows:

$$d_{IS}(u,v) = \sum_{(x,y)} (v(x,y) - u(x,y))(log(v(x,y)) - log(u(x,y)))$$

Though this is a standard metric, it becomes undefined if either u or v is equal to zero. When both are equal to zero, their input into the metric is ignored. This means that to compare two different return distributions, it was required that they had the same return support (i.e. all return pairs that have a positive probability of occurring).

Figure 5.1 indicates that the comparisons that are considered do not contain the same support. This is not surprising as the Nash Equilibrium policies' return distribution only has one return pair in its support (i.e. (14, 8)) and even for very small temperature parameters, the exploration nature of Boltzmann action selection means that every return pair is in the support (even if the probability of occurrence is very small).

The comparison of Nash Distribution policies' return distribution to the random policies' return distribution is defined. This is because the random policies gives a positive probability of occurrence to every single feasible return pair. Again, the Nash

Distribution policy seems to become more like the random policy with an increase in the temperature parameter.

As there is no guarantee a well-defined value from the comparison would result from this metric, it was ignored.

Separation Distance (SD1 and SD2)

Separation distance is the first measure which is not a metric. It is not a metric because it is not commutative; therefore both versions of this distance are considered. Separation distance has been advocated by Aldous and Diaconis (1987), for use with a Markov process due to the special properties it has. The general form of the Separation distance is as follows:

$$d_{SD}(u, v) = \max_{(x,y)}(1 - \frac{u(x, y)}{v(x, y)})$$

The variations of the distance depend on which distribution is the denominator. Both Separation distances over Nash Equilibrium's reward distribution (SD1) and Separation Distance over Nash Distribution's reward distribution (SD2) were considered. The distance takes values of between zero and one.

A distance of zero can only be observed if the distributions are identical. If they are not identical this implies that there exist a (x, y) such that $u(x, y) < v(x, y)$ (as the sum of both return distributions must total to one). A distance of one implies that there exist a return pair (x, y) that is in the support of nominator return distribution and not the denominator return distribution. From figure 5.1, it is seen that only one is observed for the SD2 graph when comparing the Nash Equilibrium over the Nash Distribution (the black line). This happens because as soon as the Nash Distribution is slightly perturbed, there will be return pairs observed which are not (14, 8), which is the only return pair observed for the Nash Equilibrium.

Another feature of the graphs is how the Nash Distribution appears to become more like the random policy as tau increases (the graph line on both SD1 and SD2). This is indicated by the decrease in the distance as tau increases. Notice how the change in distance is not as rapid as for most of the other statistics. This is a very common occurrence of Separation distance and, from the results, there is a tendency for it to

be the last to converge (i.e. move to zero) out of all the distances. This is due to the measure looking for existence differences in the return distributions. By existence difference it is meant that a payoff pair occurs in one of the distributions and not the other. Unfortunately, this convergence criterion was too strong for the limited number of episodes used within the runs and was, therefore, ignored.

Chi-Squared Distance (CHI1 and CHI2)

The Chi-squared distance is non-metric because it is not symmetric. This means the statistics are calculated both ways. The Chi-squared distance is one of the standard *goodness-of-fit* tests for comparing two distributions.

$$d_{CS}(u,v) = \sum_{(x,y)} \frac{(v(x,y) - u(x,y))^2}{v(x,y)}$$

The statistic does suffer from giving not well-defined values because, as with the Information Value, some of the return pairs are regularly zero. This is seen in figure 5.1 and the measure was ignored because of it.

Expected Value (E1 and E2)

The final set of statistics considered were the expected values from the return distributions. Expected values do not give an indication of similarities between the return distributions but they do give an indication of what is happening within the policies. The expected values are used and discussed later in this chapter.

Levels of acceptance

From previous discussion, the KS and expected value will form the statistics used to analyse the learning policies. There has been discussion on what the levels of acceptance are to confirm that two policies are similar and no level has been defined. It would be inappropriate to define some arbitrary value as the level of acceptance, hence all discussion about the KS distance is only concerned with whether it is *good* (i.e when it is close to zero) and when it is *bad* (i.e. close to one). It can be concluded that two policies are similar only if the KS statistic is zero, everything else is just comparing two results.

In the next section, the KS distance is applied to the learnt results of the simple 233 game.

5.3 233 Game

In this section, the simple 233 game is used to demonstrate the effects of the temperature parameter on the three learning methods. Also, an investigation into what policy changes occur during learning has been conducted in this section as well. To allow for a consistent presentation of these policy changes and a clearer representation of what is occurring, only one example of learning was focused on (i.e. using the SARSA method with a temperature value of 0.02).

Though only a simple game was used for this investigation some surprising and interesting results are seen. By using a simple game, the run-time was reduced and more runs could be completed (giving a richer variety of results). Also, as the simple 233 game had less possible states that could be visited (than a more complex model would), less episodes were needed to reach convergence[1]. When convergence is not found in the simple game, it is unlikely to be found in a more complex game.

A variety of graphs are used within this section to visualise the results. Where necessary, a detailed explanation is given for the graphs. Appendix B contains some of the data used to generate these graphs. As such a large quantity of data was required to generate the graphs, not all have been included in this thesis, though are available on request[2].

Variation in the Temperature Parameter

The graph in figure 5.3 shows the results from varying the temperature parameter *tau* within the different reinforcement learning techniques that were considered. The x-axis shows the varying *tau* and the y-axis shows the mean of Kolmgorov-Smirnov (KS) statistic (over 100 runs). Each KS statistic has been calculated by comparing the reward distribution generated by the learning players with that of the corresponding Nash Distribution players. The learning player's policy considered was the one learnt after the ten million episodes.

The graph in figure 5.4 is an enlargement of the area of interest in figure 5.3.

[1] Convergence, in this context, means that the learnt policy becomes virtually identical to the Nash Distribution policy

[2] Email: a.j.collins@soton.ac.uk for details

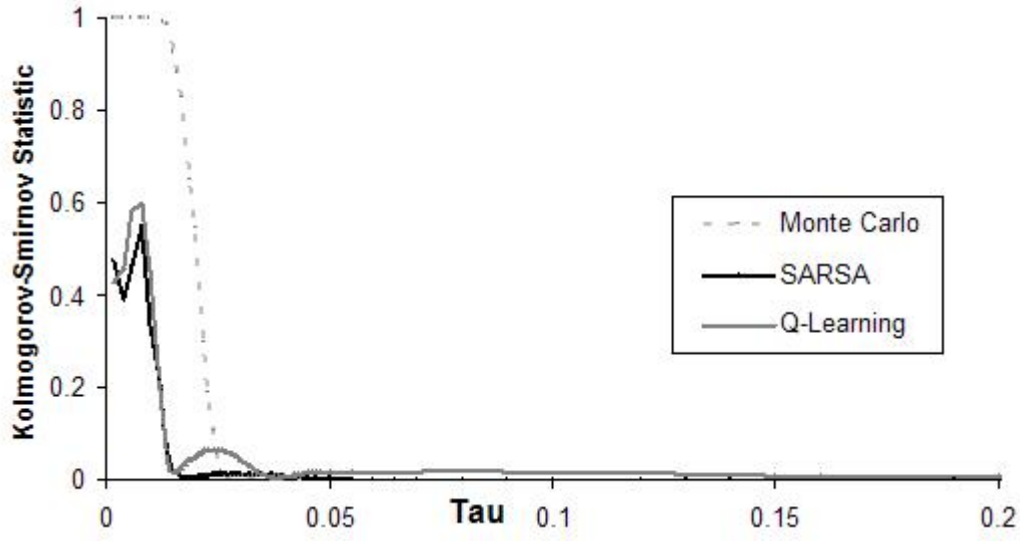

Figure 5.3: Graph showing the mean Kolmogrov-Smirnov statistic against tau, for the simple 233 game.

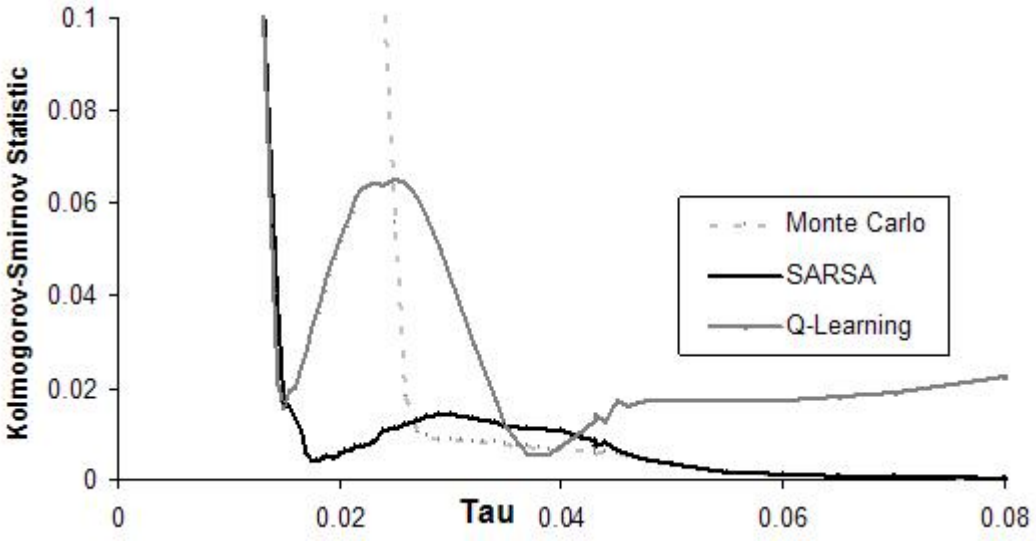

Figure 5.4: Graph showing the mean Kolmogrov-Smirnov statistic against small values of tau, for the standard 233 game.

Before any features are discussed about the graph, it is important to establish exactly how the graph was constructed and why it was done that way. The areas intended for discussion are:

- Why a comparison between the learnt policies and the Nash Distribution policies?
- Why Boltzmann Action selection was used in generating the reward-distributions?
- What are the bounds of the 100 runs' results?
- How many points were used to generate the results?

The reason that the learnt policy was compared to the Nash Distribution policy was because it was expected (from the convergence proofs given in chapter six) the learnt policies to converge to them. As seen from the comparison of the Nash Distribution policy to the Nash Equilibrium policy in figure 5.2, the two policy types are not necessarily the same thing. The Nash Distribution policy (and the effect of tau on this) is the focus.

Both reward distributions were calculated using Boltzmann Action selection and the corresponding *tau* parameter. As shown in chapter four, using Boltzmann Action selection means a smooth change in action selection as Q-values vary. Hence jerky changes are not seen in any of the three lines in figure 5.3. If Boltzmann Action selection was removed (and replaced with greedy action selection) jerky changes within the graph would have been seen. This happens because once an action's Q-value becomes the largest of the available actions there is a sudden change to that action and thus a sudden change in the reward distributions. This does not happen with Boltzmann Action selection because as an action's Q-value increases (or decreases) it begins to have more (or less) influence on the shape of the reward distribution.

As the KS represents the average of 100 different runs, it is appropriate to look at the bounds of these results. This is quite an important issue and is discussed in depth later in this section.

It would have been incorrect to compare the three learning methods for a fixed temperature parameter as it was unclear what effect the temperature parameter has on

methods learning. Therefore, the different learning models were run with different temperature parameters[3]. There are infinite different values to use for the temperature parameter. An idea of an upper limit of the temperature parameter is given in previous section. For high values of the temperature parameter ($\tau > 0.3$), the associated Nash Distribution policies are very dissimilar to the Nash Equilibrium policies (which is intended to be reached through learning) as seen in figure 5.2. Temperatures that were too high were not considered.

Figure 5.3 implies a continuous set of results for $0 \leq \tau \leq 0.2$, however this was not the case. A discrete number of different temperatures (approx. 60) were considered and their results were joined up to make the graph clearer. The points that were considered can be found in the tables in Appendix B. In an ideal world, all the points could have been generated at standard intervals of say 0.0001. As each point generated took approximately a day to run, this would have been impractical. Issues of computer run-time are discussed later in this chapter. This lack of data is not a problem because of the expected smooth nature of the results.

Even though the results were expected to produce a smooth graph, the runs were concentrated on values of tau at areas of interest. One exception to this was around tau near zero. As discussed in the model chapter, it would be impossible to gain accurate results for very small values of tau because of approximation problems with Boltzmann Action selection (see Chapter Four for more details). However, this did not seem to be a problem as the results achieved there did not yield good KS statistics (i.e. seemed to have diverged from the Nash Distribution).

Now that the discussion of the construction of these results is complete, a discussion of the results themselves can now be moved onto.

There are three different lines in figure 5.3, each with their own distinct shape. Each of these distinct graphs belongs to a separate Reinforcement Learning method (i.e. SARSA. Q-Learning, and Monte Carlo learning). The techniques are similar but they do produce some quite different results. This phenomenon was noticed in Takadama and Fujita (2005). It is intended to discuss here these shapes and reasons for their

[3]The variation of tau could be called the *sensitivity* of a reinforcement learning technique.

occurrence, where the reasons are known. The features of the graphs intended for discussion are:

- The low KS statistics observed at high values of tau
- The high KS statistics observed at low values of tau
- The local dip observed by both Q-learning and SARSA methods
- Difference between each technique's graph

The first noticeable feature of all three graphs is that they all tend to do badly (have a high KS value) at low tau values. By low, it is meant between zero and 0.015. From the discussion about the relationship between the Nash Distribution and the Nash Equilibrium, it would have been preferred if there had been better results at low tau values. The reason for these bad results is simple; a low tau value means less exploration and less exploration means a larger number of episodes to converge to the Nash Distribution. Therefore, even though 10 million episodes were played, there is less chance that the game will take a non-greedy path as the size of tau determines this (hence will not explore the state space).

As mentioned in the model chapter, it was not possible to produce results for very low values of tau. However, it is speculated from the arguments above that the situation would get worse for lower tau and higher values of the KS statistic would be observed. Though this has not been proved, it is believed that for a low enough tau, the highest value of the KS statistic (which is one) would be observed. As results with high values of KS statistic are of no use to us, no effort was made to investigate this further.

With high values of tau (defined as 0.06 to infinity), the exact opposite to what happens with the low tau values is observed. Instead of observing KS statistics of around one, values of around zero are observed, as shown in figure 5.4. This shows that if there is adequate exploration the policy will convergence to the Nash Distribution policy. However, from figure 5.2, these Nash Distribution policies (from high tau values) are highly dissimilar to the Nash Equilibrium policy (which would have been desirable) and therefore of no interest.

An interesting feature of this high tau value is that the KS statistic does achieve zero. As this statistic is an average of 100 runs, it is surprising to get a zero result come about. This is discussed further later in this chapter.

Figure 5.4 shows that both the Q-learning and SARSA results have local minimum within the KS statistic[4]. The SARSA method's dip is lower (approx. 0.004) and occurs at a lower tau value (approx. 0.02). There is no certainty about the exact location of the minimum because of the stochastic nature of the results, however it is fair to conclude that they do occur.

From previous discussion, faster convergence, to their appropriate Nash Distributions, should occur for higher values of tau. It is not obvious why there is an increase after the minimum. One suggestion for this occurrence is that this is not a linear system and linear changes within the Nash Distribution as tau varies are unlikely. This can be seen in figure 4.5, where there is clearly not a linear change in expected reward gained with Nash Distribution polices as tau changes.

If non-linearity was the reason for this occurrence then it would be experienced within the Monte Carlo method results as well. This would not be the case if the Monte Carlo method was more sensitive to a factor than the two other methods. That factor is the tau value. The Monte Carlo method seems more affected by the tau value than the other two methods. This is seen from the extremes in values that are observed as tau varies (i.e. a sudden drop in KS distance at tau equals 0.025).

The main difference between the Monte Carlo method and the other methods is that is does not bootstrap (i.e. update estimates based on estimates). By not bootstrapping, the Monte Carlo learning is more sensitive to the rewards observed from non-greedy action selection as actual rewards are used in its Q-value updating. This means that when the Monte Carlo learning policy stumbles onto the Nash Distribution policy it quickly reinforces that policy, however, quite a bit of exploration must occur before this can happen.

For all the methods, if the number of episodes per run was increased, it would be expected that the sudden shift from high expected KS statistic to a low value would

[4]The experiment was repeated and still a local minimum was observed by both techniques.

occur for smaller tau values. As the number of episodes increases, so should this shift until a completely flat-line of zero KS statistics is observed. Only 10m episodes were run and thus the analysis is based around this assumption.

A feature that differentiates the Q-learning results from the other techniques is that the results seem to follow a dampening oscillation in the KS statistics as tau increases. There are local peaks as tau equals approximately 0.025 and 0.08; and there are local minimums at approximately 0.015 and 0.035. Why these oscillations occur and why they seem to only affect Q-learning is not clear. Understanding this phenomenon has been left to further study.

For further study it is required to choose a learning method and temperature to use for examples within this section. Ideally the *best* combination is required. By *best* it is meant to have a small average KS statistic (hence the learnt policy is close the Nash Distribution policy) and the smallest tau value (hence the associated Nash Distribution policy is close to the Nash Equilibrium policy). From visual inspection of the results, the SARSA method with a temperature of 0.02 was considered the *best* for this purpose.

Episodes

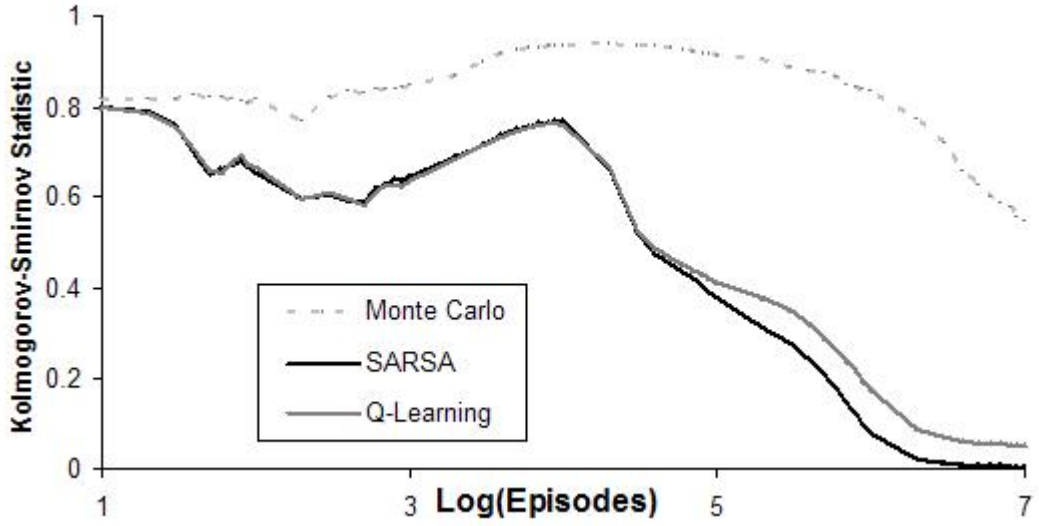

Figure 5.5: Graph showing mean Kolmogorov-Smirnow statistic against variation in number of episodes play (tau = 0.02), for standard 233 game

Figure 5.3 gives an insight on how convergence is affected by the temperature. The graph does not display how the learnt policies change over the episodes. Figure 5.5 takes a fixed temperature (tau = 0.02)[5] and plots the change in policy as the number of episodes increases in each of the three methods. A logarithmic (base ten) axis was used for the episodes. Using a standard logarithmic, the graphs have a smoother look (as there tended to be rapid change at the beginning of the run and less change at the end of a run). However, it is important to remember this logarithmic scale when comparing different rates of change within the graph.

A discrete number of data points (approx. 60) were used to construct each of the different method's graphs in figure 5.5. As before, these points have been joined up on the graphs for ease of reading. Each data point forms the average from 100 runs. Each run was paused at certain points[6] so that the learnt policies' return distribution could be compared to the Nash Distribution policies' return distribution and the KS statistics recorded.

It is immediately seen from the graphs, that the Monte Carlo method is out-performed by the the other methods and thus a higher KS statistic is observed. The other two methods produce similar results, with only a slight variation when the SARSA method out-performs the Q-learning method at the higher number of episodes. This similarity implies that the same phases are being passed through while the methods are learning.

In this section, an explanation is given for these phases. First the phases are categorised by splitting up the episodes into four sections. The first phase occurs between 0 and $10^{1.5}$ episodes and could be considered to be the *warm-up* phase[7]. The second phase occurs between $10^{1.5}$ and 10^3 episodes, where a slight dip in the KS statistic is seen (called the *dip* phase). The third phase relates to the peak in the KS statis-

[5]When using greedy action selection, the Nash Distribution polices behave exactly the same as the Nash Equilibrium policies up until a temperature of 0.0275.

[6]The pause points were 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1K, 2K, 3K, 4K, 5K, 6K, 7K, 8K, 9K, 10K, 20K, 30K, 40K, 50K, 60K, 70K, 80K, 90K, 100K, 200K, 300K, 400K, 500K, 600K, 700K, 800K, 900K, 1M, 2M, 3M, 4M, 5M, 6M, 7M, 8M, 9M, 10M.

[7]A discussion on simulation *warm-up* periods has not be conducted here. An interested reader can find more information about this in Robinson (2007)

tic at 10^3 and $10^{4.5}$ episodes (called the *peak* phase). The *final* phase represents the remaining episodes.

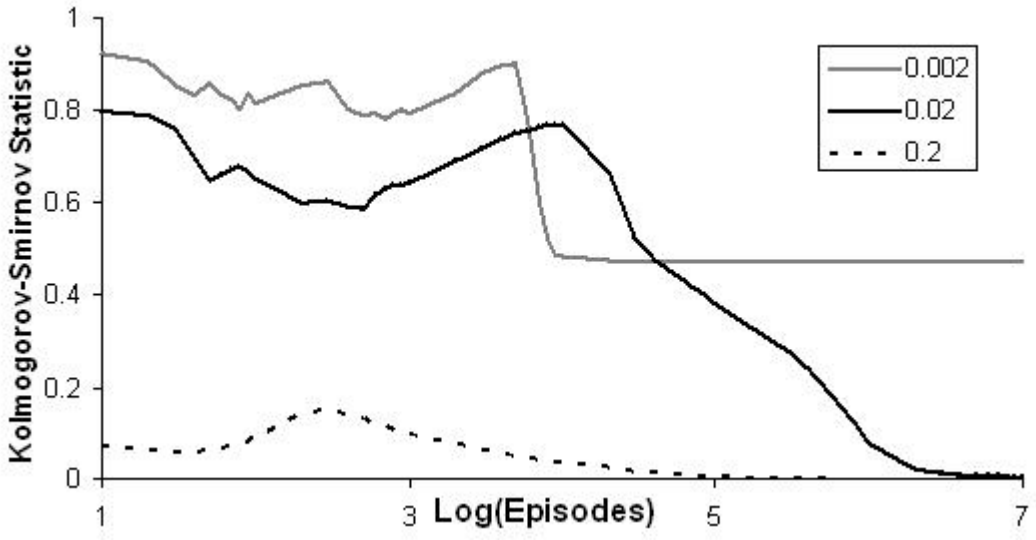

Figure 5.6: Graph showing mean Kolmogorov-Smirnow statistic against episodes for three different SARSA learning temperatures, for standard 233 game

This split of episodes relates to when the temperature is 0.02, however the same phases can be seen for different temperatures (as shown in figure 5.6). These graphs were constructed in the same manner as in figure 5.5 but with a *fixed* policy (SARSA) and variation within the temperature. All three graphs go through the same phases (*warm-up, dip, peak, and decline*) though these phases all happen at different rates and at different points.

The graphs confirm what was said before, that high temperature parameters encourage convergence. Other than greater exploration, another reason for speed of convergence can now be seen. This is that the higher the temperature, the closer the initial policy starts to the corresponding Nash Distribution policy in the first place. This occurs because each of the learning players starts with the completely random policy and the higher the temperature, the more like the completely random policy the Nash Distribution is (this was discussed in the model chapter).

In the quest to explain the phases it is required that a comparison of the learnt policies to other possible policies including the Nash Equilibrium so that more about

them can be explained.

Nash Equilibrium

The main concern of these experiments is whether the learnt policies are close to the Nash Equilibrium policy. Figure 5.7 shows these comparisons for the learnt policies with fixed temperature (Tau = 0.02). The results were constructed in a similar way to figure 5.5 but with two major difference. The first difference was that the reward distributions were calculated assuming greedy action selection is used(i.e. off-policy), which meant that the return distributions represent the outcomes from the best-response action selection. This was done so both reward distributions are using the same action selection method and thus removing any bias that would be present because of the different action selection methods. The Nash Equilibrium's return distribution is P(return is (14, 8)) = 1.

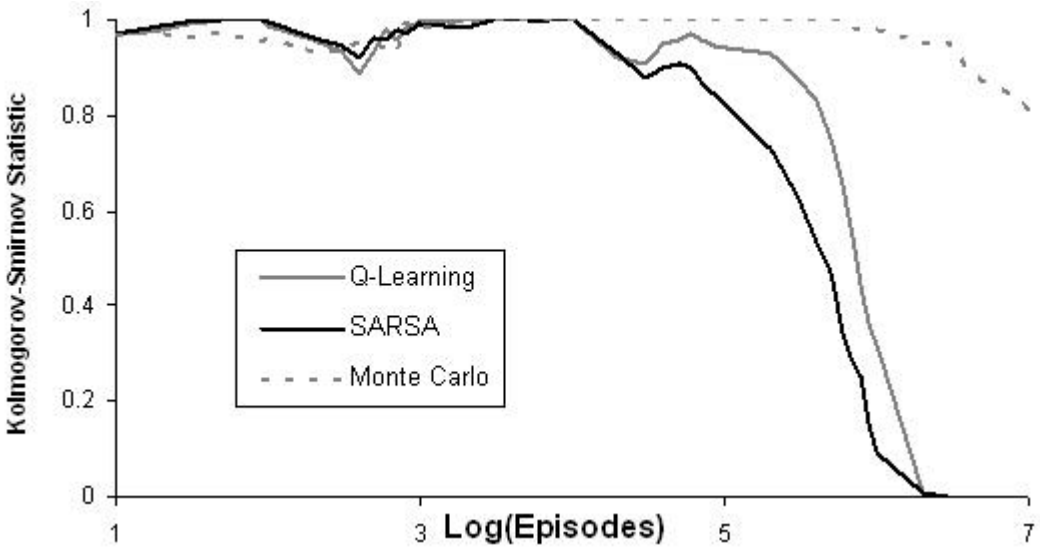

Figure 5.7: Graph showing the mean Kolmogorov-Smirnov statistic against episodes for comparing different RL techniques (Tau = 0.02) to the Nash Equilibrium, for standard 233 game

The second difference from the previous graphs is that the learnt return distribution was calculated using only one learnt policy. P1's policy was the learnt policy but P2's policy was the Nash Equilibrium one. Thus the return distribution of play from the learnt policy versus the Nash Equilibrium was measured against the return distribution of the Nash Equilibrium policy versus the Nash Equilibrium policy. The reasons

for only using one learnt policy are twofold. Firstly P1's Nash Equilibrium policy is of more interest than P2's because it is more sophisticated (as the player must learn to play *five* in the first round). Secondly, the two learning players might have developed a cooperative pair of strategies but how the players react to an aggressive policy (i.e. Nash Equilibrium one) was of interest. Bearing in mind all these construction consideration, the graphs can now be analysed.

For all three learning methods, figure 5.7 indicates that P1's learnt policy is completely dissimilar to the Nash Equilibrium until around 10^5 episodes have been played. After that point, all three learning method's derived policies become closer to the Nash Equilibrium policies (when playing against P2's Nash Equilibrium policy). The SARSA method converges the fastest, followed by Q-learning and then the Monte Carlo method.

When the Nash Distribution policy (Tau = 0.02) is used instead of the learnt P1 policy, the KS statistic is zero. Convergence to the Nash Distribution would imply convergence (i.e. low KS statistics) of the KS statistic to zero (as it is expected that learnt policy would behave like the Nash Distribution policy).

When different temperatures are used, different results occur. When too high temperatures are used then the KS statistic remains at one as the Nash Distribution policy (which the learnt policy is seen to converge to) is too unlike the Nash Equilibrium policy. When too low temperatures are used then the KS statistic also remains low because convergence has not been reached due to lack of exploration within the ten million episodes. Thus the balance between exploration and adequate Nash Distribution must be maintained to achieve the *good* results seen in figure 5.7.

Another feature worth mentioning here about figure 5.7 is the slight *wiggle* in the results at about $10^{2.5}$. This implies a slight move towards the Nash Equilibrium policy which is quickly corrected. These changes to the learnt policies are the focus of the next discussion.

Myopic and Random Comparison

Other policies that the learnt policy was compared to were the completely random policy and the myopic policy. The return distribution generated by playing the learnt

policy versus the myopic policy was measured against the return distribution generated by playing the myopic policy versus the myopic policy (and similarly for the random case). As with the comparison with the Nash Equilibrium policy, only the P1's learnt policy is considered for comparison. For the completely random policy, Boltzmann Action selection is still used when generating the return distribution. The greedy action selection method is used when calculating the reward distribution for comparison with myopic reward distribution.

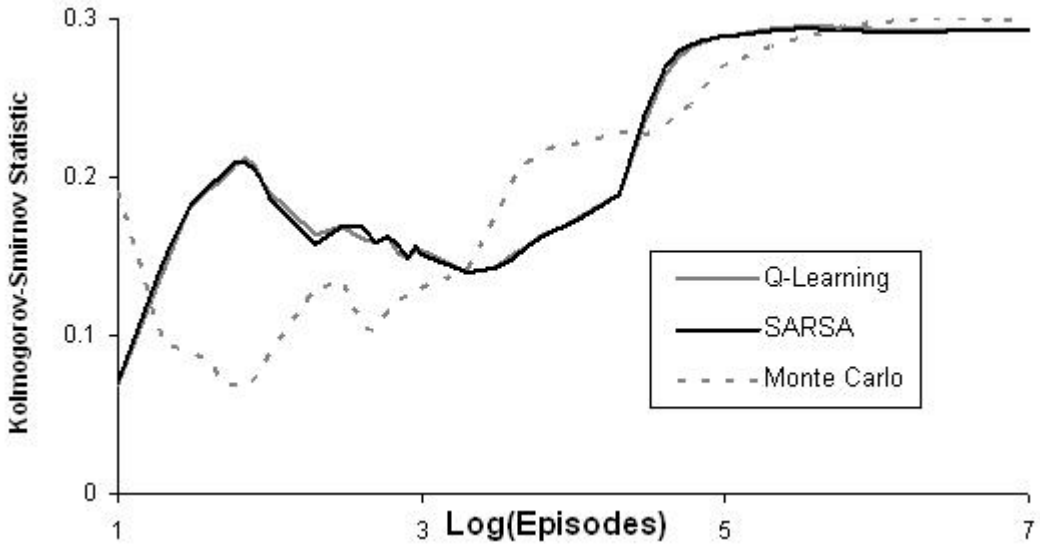

Figure 5.8: Graph showing the mean Kolmogorov-Smirnow statistic against episodes for comparing the different learning runs (Tau = 0.02) to the random policy, for standard 233 game.

Figure 5.8 shows the results for the comparison with the random policy for the three different Reinforcement Learning methods. As would be expected, there is a general increase in the KS statistics for all three methods. This implies the learning player stops play randomly as more episodes are experienced (and starts behaving as a developed policy). This increase is not universal however and these anomalies are explained here.

The Monte Carlo learnt policy becomes closer to the random policies after only 10 episodes but soon becomes less random as the number of episodes increases. This occurs for other values of tau and also when the runs were repeated. This is due to

the updating extremes that both players experience within the first 10 rounds [8] and can be ignored as a *warm-up* anomaly.

The other anomaly that is observed in figure 5.8 is the dip in the SARSA and Q-learning graphs around 10^3 episodes. This implies that P1 is reverting to a more random policy at this point and a second phase of random play by P1.

There is an increase in the average KS statistic but the learnt policies do not exceed a value of 0.3. However, the return distribution of the Nash Distribution policy versus the random policy compared to the return distribution of the random policy versus the random policy gives a KS statistic of approximately 0.2938, thus the learnt policies would be expected to reach such a value. This low KS statistic of the Nash Distribution policy is due to a temperature parameter of 0.02, which implies some exploration (i.e. randomness within the policy). The next policy to compare the learnt policies to is the myopic one.

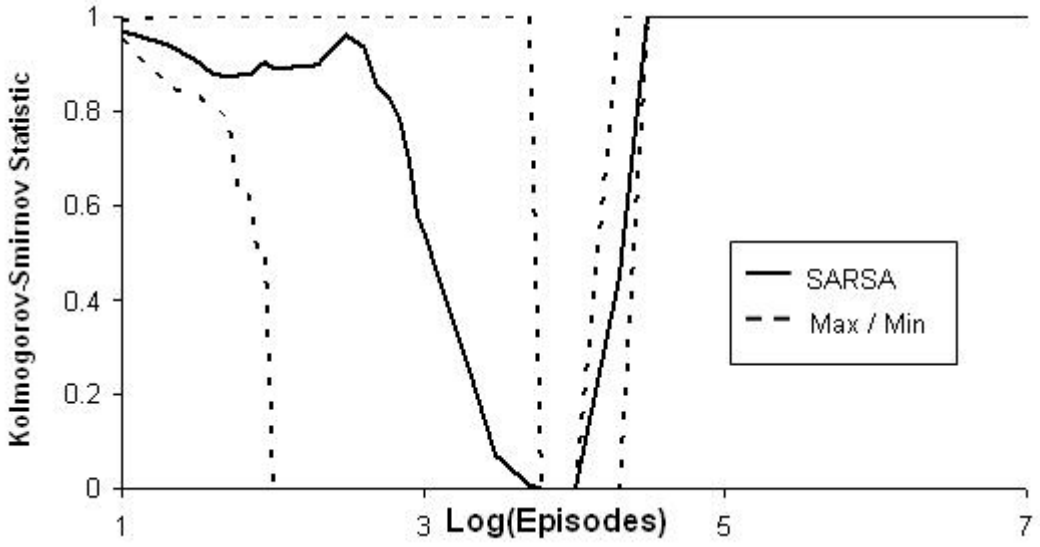

Figure 5.9: Graph depicting the mean Kolmogrov-Smirnov statistic and bounds of the SARSA learnt reward distribution compared to the myopic one (where Tau equals 0.02), against standard logarithm of number of episodes played.

For comparison to the myopic policy, the focus was the SARSA method (with a tem-

[8] The optimistic starts used will make both players play *ten* for the all actions to start with hence in different runs, the values used for updating will be between zero and thirty. As the other two methods use *boot-strapping* for updating, the values used for updating will be more consistant.

perature of 0.02). Figure 5.9 shows the comparison of the return distributions (using the same method as the Nash Equilibrium comparison) with the maximum and minimum observed KS statistics included (shown by dotted lines). The comparison is uneventful, with learnt P1 policy being dissimilar to the myopic policy except for when approximately 10^4 episodes have been played.

At 10^4 episodes, the learning player plays a myopic policy (i.e. plays *ten* as their initial price). This occurs for all 100 runs (as the maximum KS statistic is also zero). Thus the *peak* phase (described previously) corresponds to when the players have learnt to play myopically. Why the players play myopically is unclear, however learning a myopic policy is a process that must be gone through before the players can learn to play the Nash Distribution policy (in the final phase)[9].

Similar results persist for the Q-learning method and other tau value. One suggested theory is that the *obvious* myopic policy is easier to learn than the complex Nash Distribution policy. However, this does not explain why there is a phase of *myopic* play. In an effort to explain this, the variation of expected return over the episodes is shown in figure 5.10.

Comparing the learnt policy to other policies as the episodes increase, has given insight into how the learning occurs within the different Reinforcement Learning methods. As the learning players are trying to maximise their return, the returns gained under the different learnt policies are important (as well as the play observed). These average (over 100 runs) expected returns are shown in figure 5.10 for the learning runs under the SARSA method (using a temperature of 0.02).

As with figure 5.5,the four phases (*warm-up, dip, peak and final*) can be seen in figure 5.10. Using the graph, an explanation for the different phases is given here. The *warm-up* phase relates to when the players have not had a chance to learn anything and hence play randomly (hence their policies produce rewards similar to the completely random play[10]).

[9]When a Nash Distribution policy is substituted into the comparison with the myopic policies, a KS value of approximately one is observed, demonstrating the dissimilarity between the two types of policy.

[10]Completely random play has an expected reward of approximately 4.77 for both players

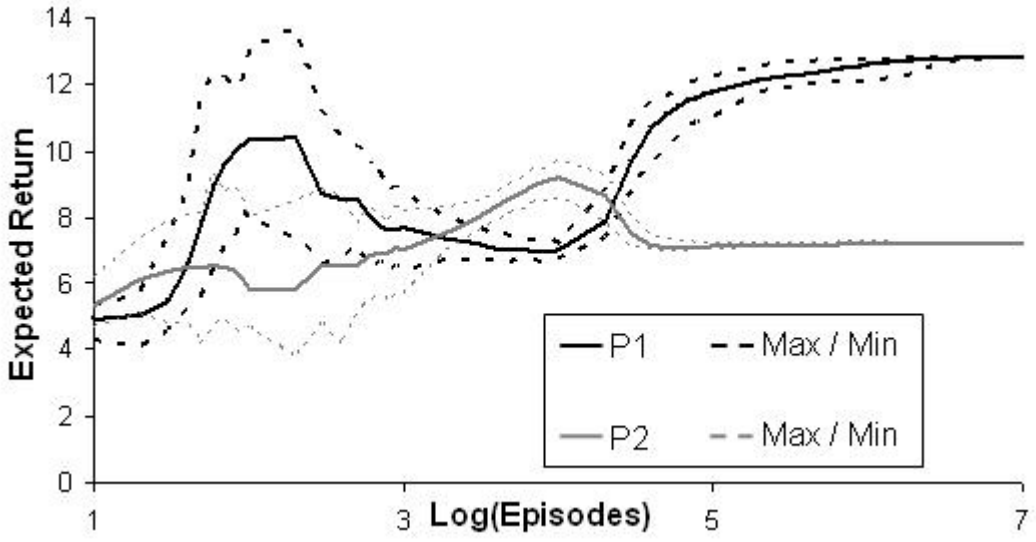

Figure 5.10: Graph depicting the expected values under the SARSA learnt reward distribution (where Tau equals 0.02), against standard logarithm of number of episodes played.

The *dip* relates to P1 learning to play low prices and P2 learning to responsed with a price of *ten*, which is similar to the Nash Distribution policy. However, both players are still learning and do not achieve the best expected reward possible. This policy maintains for some time but as P1 does not chose a low enough initial policy (usually *six* or *seven*), P2 soon learns to undercut this policy with myopic play. This undercutting moves the learning into the second random play phase (or dip) as the players are readjusting to each other, before moving onto the *peak* phase.

The *peak* phase sees P1 responding to P2's myopic policy with its own myopic policy. However, this is disastrous for P1 as they are only able to purchase one seat and receive a lower reward than in the initial *dip* phase. As greed drives the players, P1 learns to place a less greedy initial policy (of *three*) to encourage P2 out of the pricing war.

In the *final* phase and with P2 responding to P1's very low initial price correctly, P1 begins to slowly increase their initial price to the expected limit (of *five*). Hence the Nash Distribution policy has been learnt and is ingrained over the remaining episodes. Under a temperature of 0.02, the expected returns obtained by Nash Dis-

tribution players are (12.846, 7.223).

As this example shows, learning with multiple agents is not a simple reinforcement process. Hence why such a large number of episodes is required for convergence, even in such a simple game. Before discussion moves onto larger games, there are a couple of issues that need to be briefly discussed, namely: confidence bounds and stability.

Confidence Bounds

The smooth graphs in figure 5.3 were generated by Boltzmann action selection. This means that there can be confidence that the results generated by varying the temperature reflect reality, even though they were generated using a finite number of points (approx. 60 for each graph). Average KS statistics over 100 runs have been used but the system is still a stochastic one and therefore there is a margin for error within the results presented in the graph. A demonstration of this with only the SARSA method and its bounds (i.e. the minimum and maximum of the 100 different runs) are shown in figure 5.11.

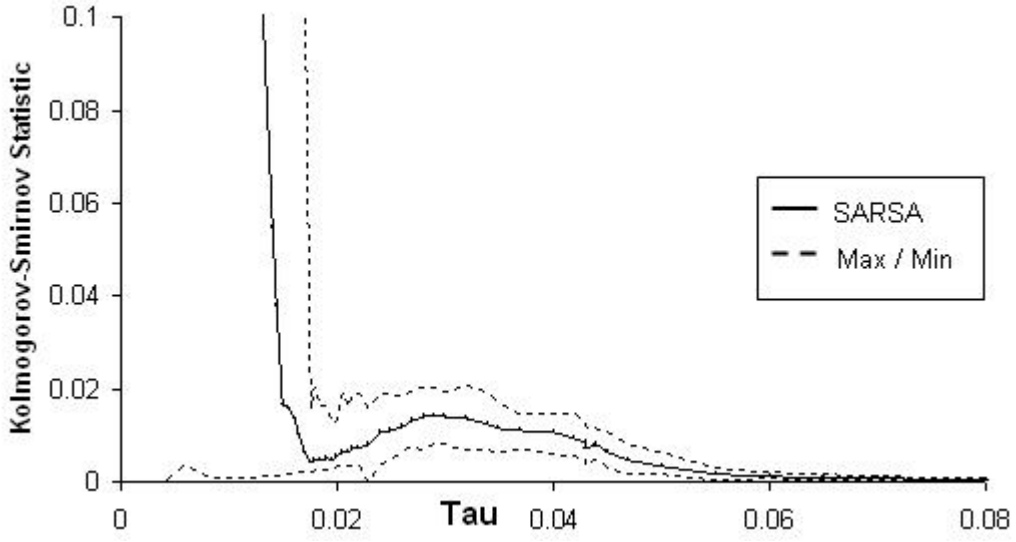

Figure 5.11: Graph showing mean Kolmogorov-Smirnow statistic and its bounds against variation in the temperature parameter, for standard 233 game using SARSA updating

There are several features about the percentiles in figure 5.11. Firstly, there is less variation in the results as the temperature increases. This is due to more exploration

occurring with higher temperature parameters. This increased exploration means that different policies are explored more quickly and thus convergence, by all 100 runs, is reached at a faster rate. It is noticeable that for tau $>$ 0.06 these bounds of the results are quite small. Another reason for this is that there is less variation as tau increases. This is because a player's opponent will also have an increased tau and is more like random play (because of the increased tau). This happens because the individual Q-value has less influence on the action selection as tau increases, the change is less noticeable and therefore it would seem like the opponent was using a more static policy. Having a more static policy would mean fewer variations across the different runs and therefore, similar results being observed.

The second feature is the extremes the bounds take for low tau. This is due to the stochastic nature of the model, hence learning players in some runs might stumble onto the Nash Distribution policy quickly (hence have a KS statistic of zero), while others are still stuck in a policy due to lack of exploration.

The figure represents a bounds result which is very common in all the other run results that were reproduced, however it was decided not to include the bounds on all the other results graphs. The reason for this decision was for visual simplicity. The result's data, given in Appendix B, does include these bounds.

For the simple 233 game, it can be confidently said that the learnt policies converge to the corresponding Nash Distribution policy for high temperatures. However, this is not of much use as at high temperatures the Nash Distribution is different to the Nash Equilibrium (the ideal goal). Even once the players have learnt to play the Nash Distribution policy there is no guarantee that they will stay there.

Stability

Once a learnt policy has reached the Nash Distribution policy, this does not mean that learning finishes and there is a chance that the learnt policy will deviate from this. An investigation into the stability of the Nash Distribution policy needs to be conducted. The only means with which the policy can become unstable is through learning. Thus the learning *step-wise* parameter (*lambda*) must be considered.

By increasing the *step-wise* parameter, the effects from learning is exacerbated. An

extreme of this would be to increase each states lambda to its largest possible value (i.e. the values of lambda before any episodes are run, as lambda decreases with episodes). One of the experiments conducted was to see the effect of replacing the initial policy of a run with the Nash Distribution policy. The results from doing this to the simple 233 game with SARSA learning (Tau = 0.02) are seen in figure 5.12.

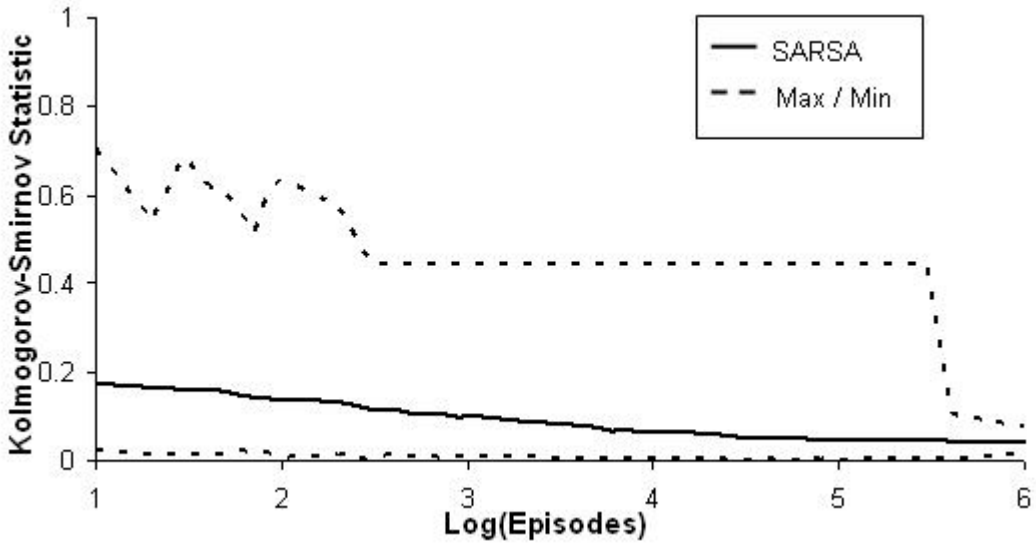

Figure 5.12: Graph showing mean Kolmogorov-Smirnov statistic for evaluting the Nash Distribution policy stability under further learning using the SARSA method (Tau = 0.02), for standard 233 game

If the Nash Distribution policies were stable then a *flat-line* of zeros would be expected in figure 5.12. This is not the case however and there is some deviation from the Nash Distribution policy (in some cases a KS statistic as high as 0.7). The initial starting step-size values seen are large (i.e. $\lambda = 0.5$, which means that half the Q-value is changed by the observed reward) and can explain the correspondingly large KS statistics. Even with these large *knocks* to the policies, the average KS statistic is only 0.2 and does not seem to diverge away from the Nash Distribution policy as more episodes are played. The Nash Distribution seems stable within this example.

Other experiments were conducted for less extreme changes and different temperatures. Most of the experiments produced *flat-line* results for temperatures less than 0.02. However this was not the case when $tau > 0.02$. With larger temperatures, more exploration is likely to happen and there is more chance that the policy will be

knocked by unusual rewards. With higher temperatures, higher average KS statistics were observed. However, like before, they did not all diverge from the Nash Distribution policy (as the maximum KS statistic decreased linearly with the number of episodes).

In reality, the step-size parameter is quite small by the time the learning players have reached the Nash Distribution policy and hence the results are stable (i.e. consistent low KS statistics).

5.4 355 Game +

After the in-depth analysis from the simple 233 game, similarities are looked for within the simple 355 game. As with the simple 233 game, an investigation into the effects of the temperature, learning mechanism and episodes was conducted. As the simple 355 game is larger (i.e. more possible states) than the simple 233 game, it was expected that the convergence results could not be as good as the simple 233 game (because more states have to be explored). Another effect that should slow convergence was the complexity increase of the Nash Equilibrium policies for the simple 355 game (see table A.6 in Appendix A for details). However, the results obtained were surprisingly good.

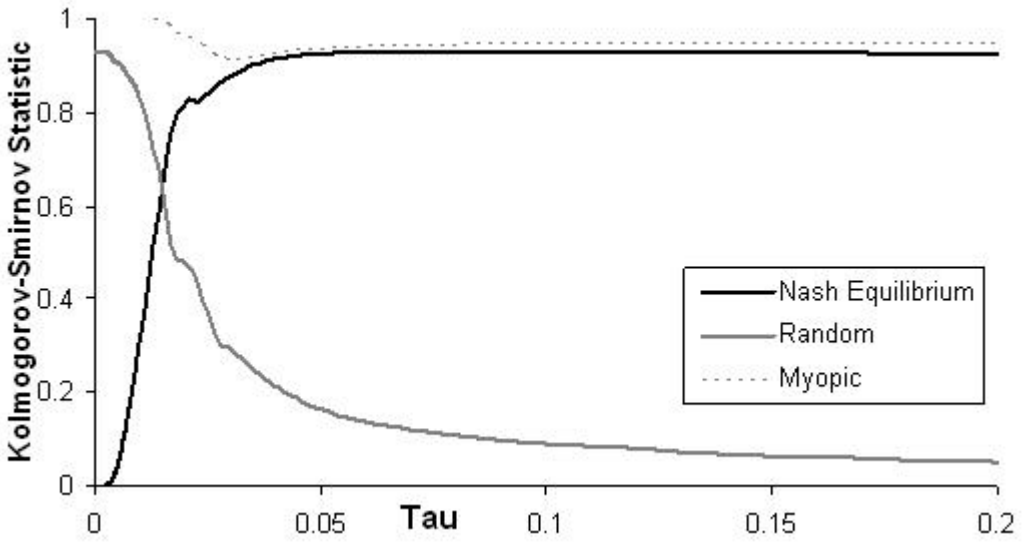

Figure 5.13: Graph showing mean Kolmogorov-Smirnov statistic against variation in the temperature parameter, for standard 355 game

Figure 5.13 shows the comparison of the different policies to the Nash Distribution policy. This graph was constructed in the same way as the KS statistic graph in figure 5.1 and bears a surprising resemblance to it. As with the 233 game, the Nash Equilibrium return distribution starts off the same as the Nash Distribution one but then rapidly distances itself from it (reaching a stable distance at about a temperature of 0.05). The myopic policies reward distributions are unlike the Nash Distribution policies reward distribution, as with the simple 233 game. The random policies reward distributions becomes similar to the Nash Distribution reward distribution as the temperature is increased. Apart from a few minor changes, the results for the simple 355 game are very similar to the simple 233 game results[11].

With more rounds in the game and more possible customers, there is a much larger number of possible return pairs for the simple 355 game than the simple 233 game (i.e. 2601 as opposed to 961). There is more opportunity for variation within the simple 355 game. However, there are a number of different action selections which gives rise to the Nash Equilibrium policies (i.e. P1's initial price could be either *eight*, *nine*, or *ten*) thus fluctuations from exploration will have less impact. This choice of Nash Equilibrium action within the 355 game is one of the possible reasons for similar results as the 233 game. This similarity between the games is also reflected in the learning mechanisms.

Figure 5.14 shows the comparisons of the learnt policies to the different Nash Distribution policies over the temperature parameter. The data is in Appendix B. Figure 5.14 was constructed in a similar way to figure 5.3, however, the 355 game is considered instead of the 233 game. As with the comparisons in figure 5.13, there are similarities between the two games and the graphs could be mistaken for each other. There are a few difference between the graphs; figure 5.15 focuses in on the area of interest to show these differences.

A feature of figure 5.14 is that it requires a larger tau, than in figure 5.3, to converge. By this it is meant that the high KS value observed at low tau, from the lack of exploration at these low values, seems to be more prominent then at that which occurred during the 233 game. For instance, with the SARSA method, a KS value of

[11]These results are deterministic as there is only one Nash Distribution for each temperature value.

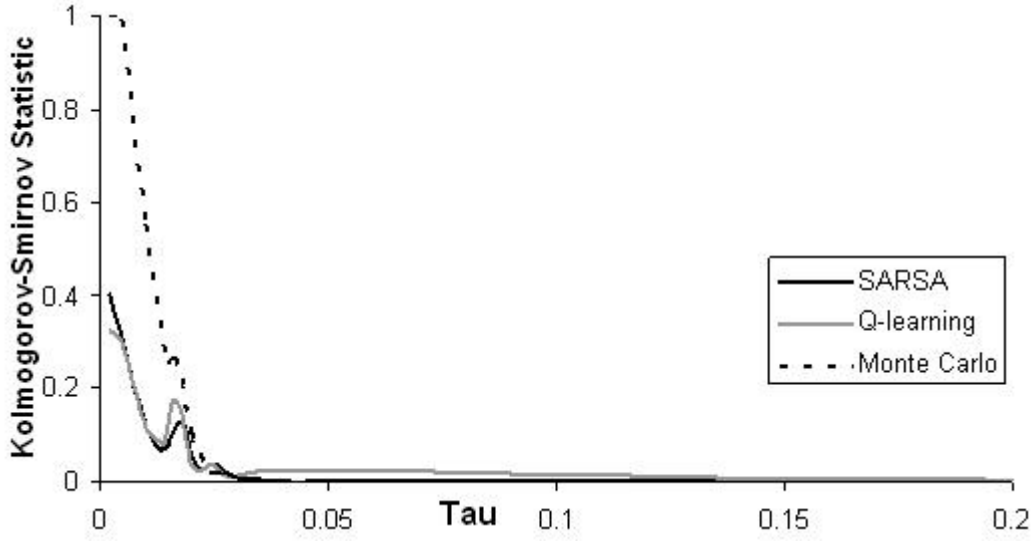

Figure 5.14: Graph showing the mean Kolmogrov-Smirnov statistic against tau, for the standard 355 game.

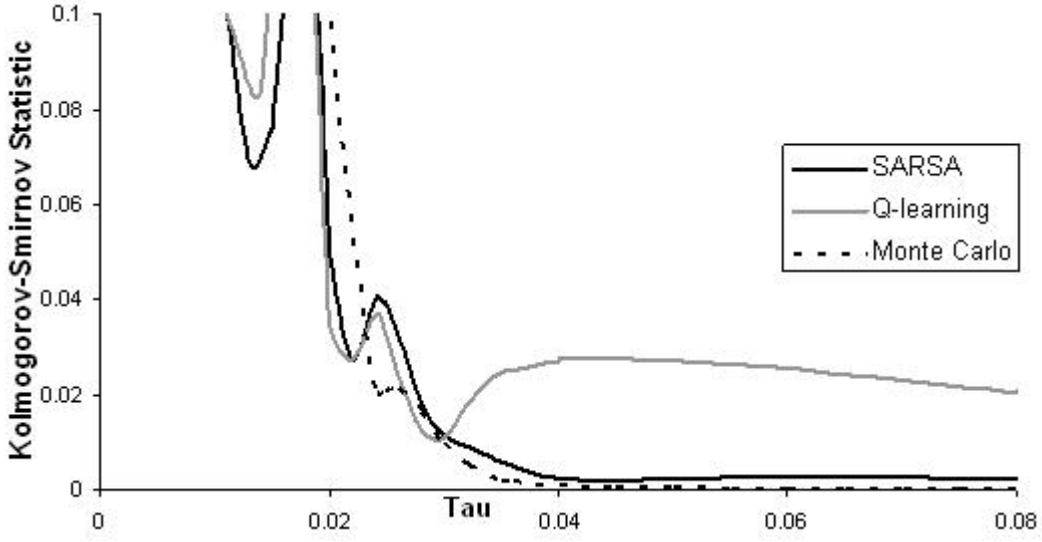

Figure 5.15: Graph showing the mean Kolmogrov-Smirnov statistic against tau, for the standard 355 game.

less than 0.01 is observed when tau is equal to approximately 0.02 for the 233 game but this value is 0.03 for the 355 game.

There are several explanations for a higher tau required to see low KS statistic values. Firstly, the simple 355 game has more states for the players to explore then in the 233 game. There are 67 states within the 233 game and 166 within the 355 game, each with their own Q-values for each of the possible actions. This number increases to 309 for the 477 game. For complete convergence to occur all paths have to be explored and only one path can be explored per episode. If both players were playing completely randomly and there are 10 million episodes, it is expected that the number of times each Q-value updated to be roughly 55000, 33000 and 22000 for the 233, 355 and 477 games respectively.

The number of updates per Q-value seems a lot for all three games however the Q-values will be updated in a changing environment (as the opponent's policy is changing) also this is not a linear or monotone process. With low temperature values, some Q-values will be updated far more often than others (as a lack of exploration means the high Q-valued prices will be selected more often). The rarely visited ones will not have reached their *thresh hold* for convergence to the optimal value[12].

The second reason for a higher tau required for lower KS statistic values is the sequential nature of the game. In the simple 233 game, round one Q-values have only to wait until round two Q-values are (vaguely) correct before the correct values begin to be updated. In the simple 355 game, round two Q-values have to wait until round three Q-values are (vaguely) correct before round one Q-values are updating correctly. This process forms the induction step within the proof chapter.

From these arguments, it is expected that a higher level of exploration is required for a larger game to converge within 10 million episodes. This means that for larger games, a very low (< 0.01) KS statistic is expected to be observed for larger temperatures. This implies that to observe a very low KS statistic, a larger number of episodes is required. It took 5.5 days to run each of the 355 games 100 times for 10

[12]There has been no experimentation into finding how many updates are required to ensure that an individual Q-value converges.

million episodes. If the number of episodes was multiplied by a factor of ten, this would take 55 days using the same computer.

Another feature in figure 5.15 is the oscillation of the Q-learning and SARSA method's KS statistic for low temperature values. This oscillation also occurs within the simple 233 game but is more pronounced here. As with the simple 233 game, it is assumed that the non-linearity of the system is the cause for these oscillations and with a more complex system, these non-linear effects are pronounced. As mentioned before, the effects of the non-linearity are less pronounced for the Monte Carlo method because of its dependency on the temperature parameter.

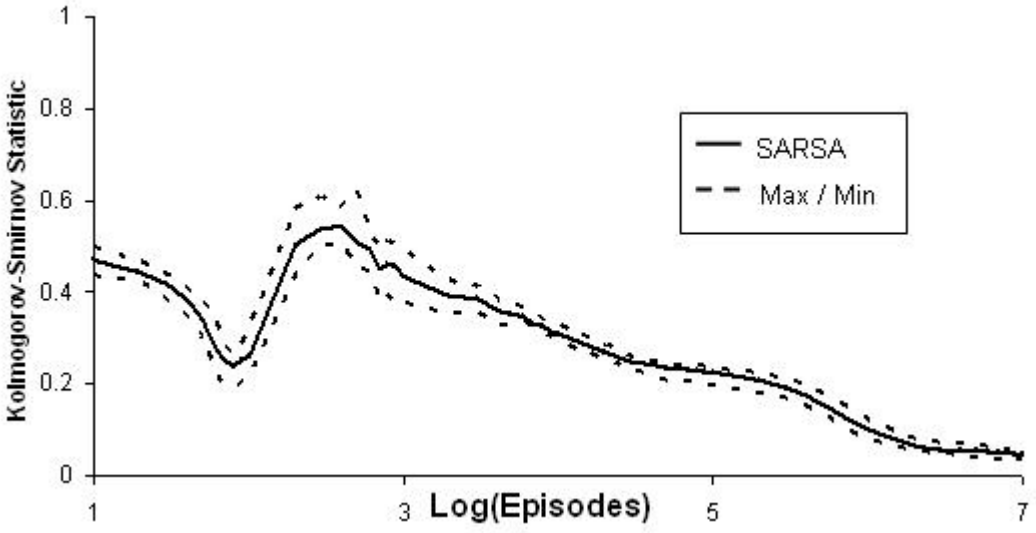

Figure 5.16: Graph showing mean Kolmogorov-Smirnov statistic and its bounds against episodes, for standard 355 game using SARSA updating (Tau = 0.02)

Discussion now moves on to looking at an actual run of the simple 355 game. Figure 5.16 shows the average KS statistic and its bounds for 100 runs of the game from comparing the return distribution, generated by playing SARSA learnt policy versus the SARSA learnt policy, to the return distribution, generated by playing the Nash Distribution policy versus the Nash Distribution policy, using a temperature of 0.02. What is noticeable about this graph is the closeness of the bounds for all episodes. The implication is that this example learning game almost always follows a standard learning routine (i.e. each learning run goes through the same phases) and at the same pace.

As with the simple 233 game, these phases can be labelled as *warm-up, dip, peak, and decline*. They also result for the same reasons as for the simple 233 game (i.e. the *peak* phase occurs due to the players moving towards a more myopic strategy). As with the simple 233 game, it is not obvious why this occurs and requires further research. As a random action selection is used as well as the semi-random customer model, a far greater variation of results is expected.

The implications from this are that a single run can be conducted to find out the convergence results from any temperature. However, without an adequate explanation of the phenomenon, there is a reluctance to conclude this.

477

To build on these results, the simple 477 game was considered. A comparison was done of the Nash Equilibrium and Nash Distribution results, which is shown in figure 5.17. The comparison produced similar results to both of the previous games. There was one noticeable difference; instead of the steady increase of the KS statistics when comparing the Nash Equilibrium to the Nash Distribution, a fluctuation was discovered (with a peak at 0.0088 and a minimum at 0.013). As with all oscillations in the results, this was due to the non-linearity of the game.

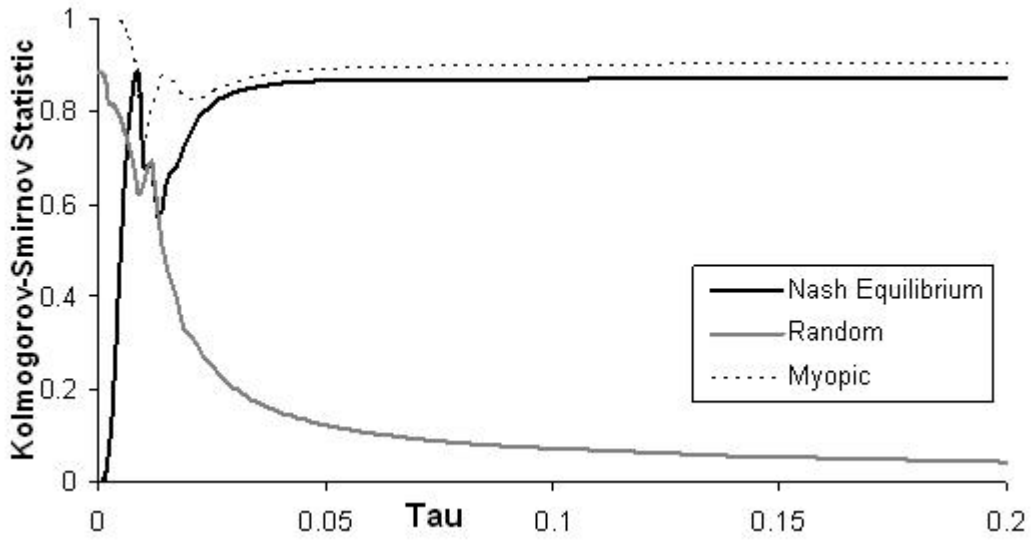

Figure 5.17: Graph showing mean Kolmogorov-Smirnov statistic against variation in the temperature parameter, for standard 477 game

Game	Minimum	Expected	Maximum
233	0.00335	0.00553	0.01334
355	0.03898	0.04745	0.05852
477	0.02044	0.02484	0.03436

Table 5.1: Table showing the average KS statistic and bounds for different sized simple games (using SARSA method and tau = 0.02)

Without plausible explanation for the fluctuation, there was a temptation to move away from any investigation of the larger games. However, this decision was already made because the run-times for simple 477 game. 100 runs of the SARSA method (with tau = 0.02) took 12 days and 4 hours[13] for ten million episodes. This produced an average KS statistic for comparison of the learnt policies to the Nash Distribution policies of 0.02484 (with a minimum of 0.02044 and a maximum of 0.03436). It was deemed impractical to continue the investigation of the simple 477 game.

Scalability

The runs of the simple 477 game were time consuming so no results were obtained for larger versions of the simple game. It would have been possible to use fewer episodes within the runs but this was deemed pointless, as the smaller games required many episodes to show convergence and it was not expected that fewer would be required for the larger games. This could not be verified as the results do not show this, as shown in table 5.1. Less than 100 repeated runs could have been used but this would not have given statistical validation to any results presented.

5.5 Physical Limitations

Discussion so far within the chapter has been about the performance of the learning mechanisms to produce policies similar to the Nash Distribution ones. Now the focus is moved onto the physical aspects of the model, namely: memory requirements and run-time. Both of these physical quantities are dependent on the type of computer used to run the model, which is discussed first.

[13]Of these 12 days, only 3 days were used to run the model. The remaining time was used to analyse the results.

Computer Specification

The decision to use the C++ programming language was discussed in-depth on section 4.5. One of the main reasons for using C++ was so that the runs could be done on the University of Southampton's super-computer. Without this facility, it was estimated that all the runs conducted for the research would have take a single computer three years to complete, assuming that it could be continuously run for that period of time. The super-computer used was an IBM e-server 325 and the runs were submitted to its nodes. Each node had a dual core AMD type 248 processor rated at 2.193 GHz with 2GB RAM. For compatibility with desktop computers all programs are compiled in 32-bit mode. Some simple tests indicated there was not any difference in speed of the runs between 64-bit and 32-bit.

This facility is a world class computing suite and no consideration was given to using any alternatives.

Memory

As would be expected, the memory requirements increased as the size of the game increased. The Q-values for each of the games were stored in binary files. The size of these files had an impact on the time required to complete a run and several tests were conducted for this purpose. These tests computed the time required to run the learning model with ten million episodes for different file sizes (these tests did not include the time to analyse the learnt policies produced. Figure 5.18 shows the results from these tests.

The memory results, in figure 5.18, shows the file-size which produced the fastest results for that particular game (in the graph N = 2 refers to the simple 233 game, N = 3 refers to the simple 355 game, etc.). As the game size increases, so does the size of the file that produces the best time results. This occurs because when small storage files are used, the Q-value is split over multiple files and the program continually switches between them. However, if too large a file size is used the program run takes time in handling the larger files storage.

The largest game considered was the simple 19-37-37 game, and the file-required for

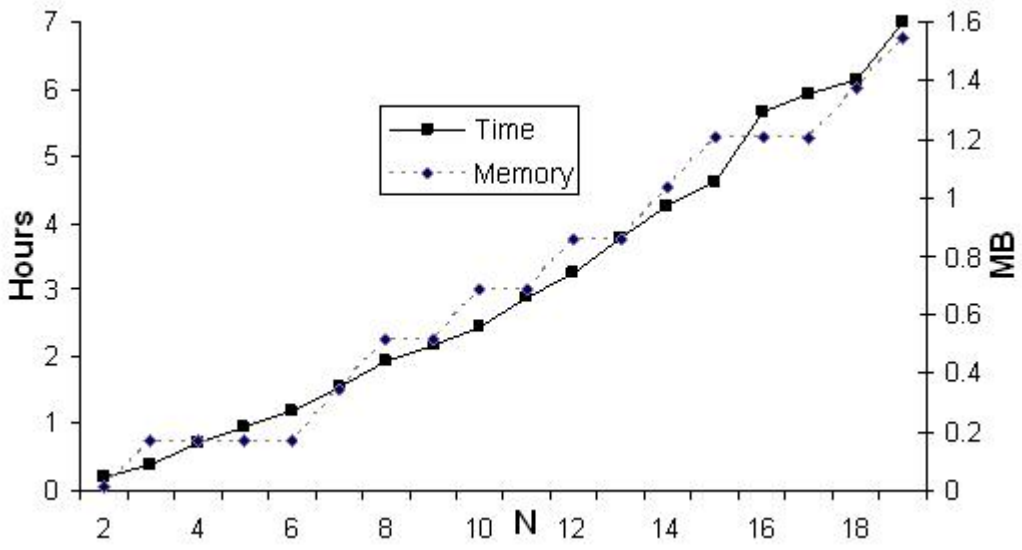

Figure 5.18: Graph depicting time and memory requirements as the game size increases

this was 1.6MB[14]. Given that runs were done on machines with 2GB RAM, this was not a problem. However, the same is not true of run-time.

Time

There are two aspects that need to be considered when talking about the run-time of the model: the time that was taken to run the learning model and the time taken to analyse the policy results. The analysis of the policy results took a significant amount of time. For a simple 477 game run, it took 9 out of the 12 days for analysis. For games larger than the simple 477 game, no analysis was completed in the time-frame[15] available. The time taken to analyse the results increases exponentially with time, this was due to the Q-values.

Visiting every Q-value is both the learning model's saviour and it's curse. It allows convergence (see the convergence proof chapter) but hinders the practical calculations of comparison. Under the current analytical mechanism, the payoff distributions are looked at. This required following every path possible through the game[16]. As the

[14]Each Q-value was stored as a *long double*, which requires 12 bytes of memory. Thus the simple 233 game has 737 Q-values, which means 0.008MB were required. However, the simple 233 game ran faster on files of 0.016MB due the file configuration that could be used.

[15]Even when only one episode was used.

[16]there are 19008 possible paths for the simple 233 game.

number of rounds increase, this number of paths increases exponentially.

This exponential increase in paths corresponds to an exponential increase in the time that is required to calculate the return distributions. This means that even though learnt policy can be found after ten million episodes and the related Nash policies are also found, it was impossible to calculate the related measures (i.e. the KS statistic). This affects games larger than the simple 477 game.

The problem could have been resolved by finding an approximate reward distribution, instead of the actual reward distribution. This would have been done by observing the results from, say, ten thousand plays of the game. However, this approach was ignored for two reasons. Firstly, the distance measures (i.e. KS statistic) would have been measuring approximations to approximations, which (given the approximations already used within the analysis) would make any results difficult to believe. Secondly, the complexity of the games, of size greater than the simple 477 game, increased and became more difficult to interpret into meaningful results.

The actual time taken to complete the single ten million episode learning model run does not seem to increase exponentially, as shown in figure 5.18. This implies that if the only concern was finding a policy (without analysing it), the time taken would be reasonable. This is assuming that the useful policy is learnt after ten million episodes, which might not be the case for the larger games.

5.6 Summary

This chapter has discussed many results (and limitations) from the empirical results obtained by the simple games. These simple games have produced some interesting results, which was analysed using the return distribution. A quick summary of the chapter is given in this section.

The methodology dictated that the way to compare the different policies was to look at the reward distributions that they generated. To compare two reward distributions, a measure was needed. The measure was chosen by using the comparison of the different Nash Distribution policies to the Nash Equilibrium policies, of the simple 233 game, as a baseline. The Kolmogorov-Smirnov statistic was chosen because

it displayed the most useful qualities and was similar to some of the other distances (i.e. Total Variation and Hellinger distance). Another *good* distance measure was the Separation distance; however, this required a greater level of convergence by the learning policies that was currently being displayed within ten million episodes.

Within the comparison of the Nash Equilibrium to the Nash Distribution, it was clear that they produced different policies when *high* temperature values were used. This was confirmed by the results in chapter four.

Though *low* temperature parameters were required, the lack of exploration associated with these temperatures meant that the learnt policies had not converged. A temperature of around 0.02 was the lowest temperature that produced sensible results for the simple 233 game. This value increased to 0.03 for the simple 355 game.

From the learning results, it was observed that the SARSA method slightly out performed the Q-learning method, though this difference was not significant. The Monte Carlo method was out performed by both the other methods for *low* temperature values.

Studying the learnt policies when the SARSA method (with Tau = 0.02) was used, showed that the learning players went through fixed phases as learnt from each episode. These phases related to random and myopic play but eventually converged to the Nash Distribution policy, which was desired. The existence of these phases was confirmed by the maximum and minimum results observed from a hundred runs.

Results from larger games were hard to obtain due the time-requirement of the analytical process used. Running the models without the analyser showed that there seemed to be a linear increase in memory requirements and run-time as the game size increased. However, due to not having analysed the learnt polices from these larger games, no conclusions could be drawn to whether any useful policies were learnt after ten million episodes.

Now the more theoretical aspects of the learning model are looked at by considering the convergence of the SARSA method.

Chapter 6

Convergence Proofs

6.1 Introduction

In this chapter a convergence proof for a specific version of the learning game is constructed. The RL method under consideration is the SARSA method and it will be shown that this method converges to the variation on the Nash Distribution (VND) policy [1]. As the variation on the Nash Distribution policy is determined uniquely by the Q-values of the model, it has been sufficient to show that the Q-values converge correctly under the SARSA method.

There has not been much work on convergence of the SARSA method and there is no guarantee that a Reinforcement Learning technique will converge (chaotic behaviour, not convergence, is observed in Sato et al. (2002)). Currently, the only papers that look specifically at SARSA convergence are Singh et al. (2000); Banerjee et al. (2004). Singh's paper only looks at a single agent and single-step framework. Banerjee's work extends this to the multi-agent case. However, Banerjee's learner must have knowledge of their opponent's current policy for updating to occur.

This requirement on the knowledge of the players is in-line with the work of Hart and Mas-colell (see Hart and Mas-Colell, 2006, 2003). In these papers Hart and Mas-

[1] Throughout the thesis, whenever Nash Distribution is mentioned this means the variation on the Nash Distribution.

colell argue that uncoupled learning dynamics [2] cannot converge for all possibilities. There are two reasons this issue does not affect this proof. Firstly, this proof is concerned with convergence to the variation on the Nash Distribution and not the Nash Equilibrium.

Secondly, a sequential game is used, which means that the player's policy at a pre-terminal state is independent of their opponent's reward function anyway and thus convergence can be shown. Assuming that the pre-terminal state is visited infinitely often, this convergence will occur, thus the preceding state to the pre-terminal state (where the player's opponent makes their action selection) will only have to consider a *fixed* expected reward from the pre-terminal state so will also converge. By induction, convergence is shown to occur for all steps in the multi-step game.

These properties of the learning dynamic combined with a few others ensure convergence results for the SARSA method. Another property is discrete finite variables (i.e. price, customers, rounds, etc.). This allows bounds to be put on the Q-values, which helps enforce convergence. Finally, there are only two players, which again decreases the complexity.

To prove convergence of our SARSA method the model must be explained in the required mathematic notation. Probability / measure theory was used as the underlining framework for convergence. A representation of the model framework has been given in the next section. An attempt has been made to keep everything as generic as possible in the framework, thus allowing the proof to be applied to other models.

Overview of Proof

Due to the obvious sequential nature of the model, the proof was split into the separate processes that occur within the model (i.e. customer model, action selection, etc.). The main body of the proof looks at showing convergence for a single round within the game. This can then be applied to the rest of game by induction. The next section looks at the framework that was used.

[2] A learning mechanism where players do not take account of the opponent's reward function or policies

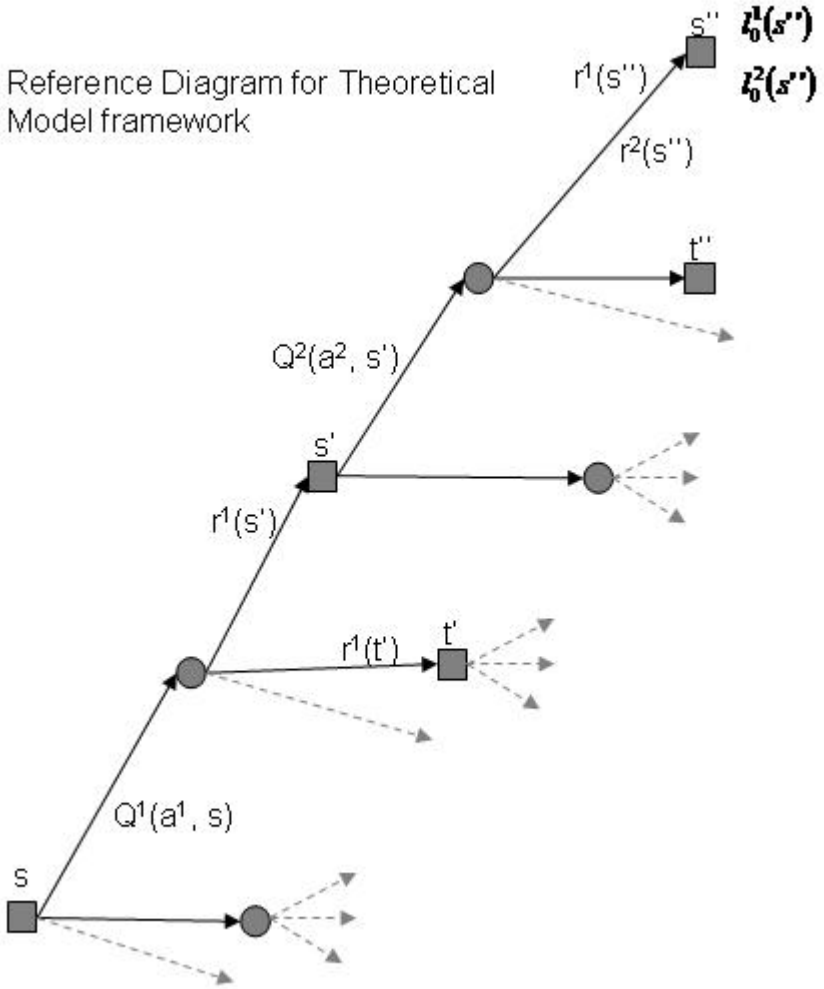

Figure 6.1: Reference diagram for notation within proofs

6.2 Conceptual Framework

To complete the proofs a conceptual framework was constructed, which is shown in figure 6.1. The figure represents the choices and outcomes from a single round of the game. The figure interpretion is now explained.

Player one selects an action $a^1 \in A^1$ (the actual models uses $A^1 = \{0, 1, , 10\}$) using its current Q-values $Q_n^1(a^1)$ (which are dependent on previous n episodes on the run, called the *history*) and the Boltzmann Action selection mechanism. This is passed to the customer model, which reacts to the current state and the game moves into a new state. The outcome from the customer model will be dependent on current

prices offered by the players and the current round. As there are only a finite number of outcomes (as only a finite number of seats to be sold at a finite number of prices), the possible new states are $s'(a^1) \in S'(a^1)$, where $|S| < \infty$. This outcome will give player one a reward of $r^1(s')$ (and a reward of $r^2(s')$ for player two). An important feature here is that for each action a^1, the customer model outcome is episode independent (i.e. is fixed over the episodes).

As the first round of the game is slightly different from the other rounds, it must be ensured that it is represented here. This can be done if you consider $|A^1| = 1$ and $r^1(s') = r^2(s') = 0 \quad \forall a^1 \in A^1$.

The next stage within the round is player two's action selection. Player two now chooses action $a^2(s') \in A^2(s')$ using Q-values '$Q_n^2(a^2)$' and the Boltzmann Action selection mechanism. Again, the customer model will generate a response based on the current state s' and action a^2. This will lead onto either another action selection by player one or a terminal state. The rewards gained from these are represented by L^1 and L^2 for players one and two respectively, where $L^1 = L^2 = 0$ in the case of a terminal state.

The run's history is referred to above, now it is intended to make clear what is meant by history and state. Previously, a state was defined as being a combination of player's prices and seats remaining, which is true here. A history, within the context of this framework, just means a combination of all the action selection and customer model outcomes that have occurred previously, in this episode and previous ones (depicted by h_n). This uniquely identifies a location within the game-tree (though there might be overlap with the use of Q-values etc.). Technically, all the variables and distributions should be written as a function of the previous states (i.e. $s'(s, a^1(s))$, however it is tended that the notation is abused within the proof and they are abbreviated to use only the currently considered state variables (i.e. s'). These abbreviations are used to make the proof easier to read.

Another abbreviation that is commonly used within the proofs is to only consider the times a state was visited. Therefore, even though there have been $n \in \mathbb{N}$ episodes, the state would have only visited $k \in \mathbb{N}$ times. Therefore, given a state, it is defined as

$k := \sum_{i=0}^{n} I_{\text{state visited in episode 'e'}}$, where $I \in \{0, 1\}$ is the standard indicator function. Now other variables' notation is considered.

Notation

In this section all the algebraic notations for the proofs and any special properties of the variables are discussed. State and action variables are considered first, shown in table 6.1. The table describes each notation in turn, stating what the variable would look like if full notation was being used (and not just our abbreviated versions).

All the parents sets have a finite non-zero rank (i.e. $1 \leq |S| < \infty$). Now the variables which depend on the state variables and the current history can be talked about. The realised values for the states and actions in a particular episode n are denoted with a subscript n. Many of these realised variables are dependent on the previous $n-1$ episodes, hence dependent on history h_{n-1}. These realised variables are given in table 6.2.

$Q_n^i(a^i) \in \mathbb{R}$ (or just Q_n^i) is the current Q-value of player i for action a^i. It is assumed that $Q_0^i(a^i) < \infty$ for $\forall a^i \in A^i$. The initial values are generic, within the empirical model *optimistic starts* were used to encourage exploration.

There are two more parameters which are not shown within the figure but are used throughout the proofs. $\tau > 0$, the temperature parameter which is a constant. $\lambda_k \in (0, 1)$ is the step-wise parameter (see chapter four). Each Q-value update will have its own lambda value, hence the usage of the 'k' variable. As before, $\lambda_k \to 0$ as $k \to \infty$.

With and in Probability

In this section the convergence concepts that are used within the proofs are introduced. As a stochastic environment is being dealt with, it is important that a robust way of representing this is used. For this purpose measure theory is used. Good introductionary books are Durrett (2004) and Williams (1991). The probability triple is defined as (Ω, $\mathcal{F}$, P), where P is probability measure on measure space (Ω,$\mathcal{F}$). Ω, is the sample space and $\mathcal{F}$ is a σ-algebra on the subsets of Ω (its elements are called *events*). Events are occasionally refered to throughout the proofs, in which case they

Notation	Meaning
S	Set of possible states that could enter the system
$s \in S$	State which enter the system in or generic term for state
$t \in S$	Alternative state which could have entered the system
A^i	$A^i(s)$. Actions available to player i when at state s. $\lvert A^i \rvert < \infty$. $A^i = \emptyset$ if it is not players turn to select an action. $P(a^i(s)\lvert s) > 0$
$a^i \in A^i$	$a^i(s)$. Action selected at state s by player $i \in \{0, 1\}$ (if allowed)
$b^i \in A^i$	$b^i(s)$. Alternative possible action selected at state s by player i (if allowed)
S'	$S'\left(s, a^1(s)\right)$. Set of possible states that can be entered by from state s after P1 has selected action a^1
$s' \in S'$	$s'\left(s, a^1(s)\right)$. State entered after P1 action selection has occurred and customer model has outputed any changes
$t' \in S'$	$t'\left(s, a^1(s)\right)$. Alternative state which could have been entered after P1 action selection has occurred and customer model has outputed any changes that affect the state
S''	$S'\left(s', a^2(s')\right)$. Set of possible states that can be entered by from state s' after P2 has selected action a^2
$s'' \in S''$	$s''\left(s', a^2(s')\right)$. State entered after P2 action selection has occurred and customer model has outputed any changes
$t'' \in S''$	$t''\left(s', a^2(s')\right)$. Alternative state which could have been entered after P2 action selection has occurred and customer model has outputed any changes that affect the state

Table 6.1: State and action notation for convergence proofs

Notation	Meaning
$n \in \mathbb{N}$	Number of episodes that have occurred
$k \in \mathbb{N}$	Used to depict the n value which indicates the k occurance of some event
h_n	Indicates the history of the run up until the n-th episode
$s_n \in S$	Realised generic state selected in n-th episode
$a_n^i \in A^i(s)$	$a_n^i(s_n, h_{n-1})$. Realised action selected at state s by player i in n-th episode, assuming state is visited and player can select an action
$r^i(s) \in \mathbb{R}$	Reward realised at by player i on entering state s. $0 < r^i(s) < \infty$. Notice that this in independent of the history
$Q_n^i\left(a^i\right) \in \mathbb{R}$	$Q_n^i\left(a^i(s), h_{n-1}\right)$. Q-value for player i's action a^i at state s in the n-th episode (assuming this player i's turn to choice an action)
$L_{0,n}^i\left(s''\right) \subset \mathbb{R}$	$L_{0,n}^i\left(s'', h_{n-1}\right), h_{n-1})$. Realised rewards and Q-values which occur after s'' for player i, used in the updating of the Q-values in n-th episode

Table 6.2: Notation for realised variables in convergence proofs

are represented as $\omega \in \Omega$. Now the different types of convergence that can happen are considered [3].

In probability

A sequence X_n converges towards X in probability if:

$$\lim_{n\to\infty} P\left(|X_n - X| > \epsilon\right) = 0$$

This is usually represented as:

$$X_n \xrightarrow{P} X \text{ as } n \to \infty$$

Almost Surely (a.s.)

A sequence X_n converges almost surely (or with probability 1) towards X if:

$$P\left(\lim_{n\to\infty} X_n = X\right) = 1$$

[3]There are other types of convergence that are not mentioned here (e.g. *in distribution* and *sure* convergence). These have not been mentioned purely because they are not considered anywhere within the proofs.

Within a probability space (Ω, $\mathcal{F}$, P), this means:

$$\{\omega \in \Omega | \text{ Given } \epsilon > 0 \quad \exists N(\omega) \text{ s.t. } \forall n \geq N(\omega) \quad |X_n - X| < \epsilon\} = \Theta \subseteq \mathcal{F}$$

$$P(\Theta) = 1$$

Notice that 'N' is allowed to be dependent on 'ω'.

Whereas '*in probability*' convergence the limits are on the outside of the probability measure, with '*almost surely*' convergence the limit is within the probability measure. Almost surely convergence implies convergence in probability (Grimmett and Stirzaker, 1992). Within practical simulation sense, an iterative process that only converges '*in probability*' is likely to not display as good as results as a process that converges '*almost surely*'. Hence an attempt to prove almost surely is given were possible within the proofs.

Infinitely Often and Eventually

The following are important concepts because they allow convergence of the iterative process to be shown. Given a countable sequence of events E_n, it occurs *infinitely often* (i.o.):

$$(E_n, \text{i.o.}) := \{\omega \in \Omega | \text{ Given } m \in \mathbb{N}, \quad \exists n(\omega) \geq m \text{ s.t. } \omega \in E_n\}$$

Similiarly, an sequence of events occur *eventually* (ev):

$$(E_n, \text{ev}) := \{\omega \in \Omega | \, \exists m(\omega) \text{ s.t. } \forall n \geq m(\omega), \quad \omega \in E_n\}$$

Both (E_n, i.o.) and (E_n, ev) are events and are linked by the following property:

$$(E_n, ev)^c = (E_n^c, i.o.)$$

Though it was not possible to prove almost surely convergence for the whole proof, it was possible to show this type of convergence in some parts. Therefore, the remaing proofs can be split into two groups. One group are the proofs which show that actions are selected infinitely often; these proofs were proved *with probability one.* The other group are those that show that the q-values converge to the correct values; these were proved *in probability.*

Assumptions on L

$L^1_{0,n}(s'') \in \mathbb{R}$ (or just $L^1_{0,n}$) is the reward observed after this round for player one. As SARSA updating is being used, this is effectively the next Q-value observed for player one. There are certain conditions that are placed on $L^1_{0,n}$. The assumptions were placed on $L^1_{0,n}$ are very important as it is intended that, by induction, the current round also has these properties.

$$\exists B > 0 \text{ s.t. } L^1_{0,n}(s^{''}) \in [0, B) \quad \forall n \in \mathbb{N} \text{ and } \forall s^{''} \in S^{''}$$

$$E(L_{0,n}(s^{''})) \to E(L^1_{0,*}(s^{''})) \text{ as } n \to \infty$$

Where $L^1_{0,*}(s^{''}) \in \mathbb{R}$ is a episode-invariant distribution, similar to $L^1_{0,n}(s^{''})$, which show the reward gained from the Nash Distribution policy.

For $j > i$, $cov(L^1_{0,i}(s^{''}), L^1_{0,j}(t^{''})) \to 0$ as $i \to \infty$ for $\forall s^{''}, t^{''} \in S^{''}$.

$L^2_{0,n}(s'') \in \mathbb{R}$ (or just $L^2_{0,n}$) is the reward observed after this round for player two. As SARSA updating is being used, this is effectively the next reward plus the following Q-value observed for player two. All the conditions placed on $L^1_{0,n}$ are placed on $L^2_{0,n}$ too. The rest of this chapter now concerns itself with the proofs.

6.3 Infinitely Often

This section is concerned with the infinite occurance of the of certain actions and events. The need for the infinite occurrance is important as convergence cannot be reached without it.

LEMMA 1. *If action a^2 is selected infinitely often (i.o.) with Probability 1 (a.s.) then state $s''(a^2) \in S^{''}(a^2)$ is visited i.o. a.s. for $\forall s^{''}(a^2) \in S^{''}(a^2)$*

Proof. We abbreviate to $s^{''} := s^{''}(a^2)$ for ease of reading. By definition of $s^{''}$ we know that $P(s^{''}) > 0 \quad \forall s^{''} \in S^{''} \quad \therefore \exists \check{P} > 0 \quad s.t. \quad P(s^{''}) > \check{P} \quad \forall s^{''} \in S^{''}$.

Let $n \in \mathbb{N}$ be the n-th occurance of a^1. Given $m_1, m_2 \in \mathbb{N}$ we define (m_1, m_2) as the event:

$$(m_1, m_2) = \{\omega \in \Omega : m_1 \leq n \leq m_1 + m_2 : s^{''}_n(\omega) \neq s^{''}\}$$

$$P(m_1, m_2) = P(m_1 \leq n \leq m_1 + m_2 : s_n^{''} \neq s^{''})$$

If $m_2 = 0$

$$\begin{aligned} P(m_1, m_2) =& P(s_{m_1}^{''} \neq s^{''}) \\ =& 1 - P(s_{m_1}^{''} = s^{''}) \\ =& 1 - P(s^{''}) && \text{since } s^{''} \text{ is episode invariant} \\ \leq& 1 - \check{P} \end{aligned}$$

If $m_2 > 0$

As a static customer distribution is being dealt with, the history does not affect the probabilities observed. $\therefore P(s_{m_1+1}^{''} = s^{''} | s_{m_1}^{''} \neq s^{''}) > \check{P}$

$$\begin{aligned} P(m_1, m_2) &= P(m_1 + 1 \leq n \leq m_1 + m_2 : s_n^{''} \neq s^{''} | s_{m_1}^{''} \neq s^{''}) P(s_{m_1}^{''} \neq s^{''}) \\ &\leq P(m_1 + 1 \leq n \leq m_1 + m_2 : s_n^{''} \neq s^{''} | s_{m_1}^{''} \neq s^{''})(1 - \check{P}) \\ &= P(m_1 + 2 \leq n \leq m_1 + m_2 : s_n^{''} \neq s^{''} | s_{m_1}^{''} \neq s^{''} \text{ and } s_{m_1+1}^{''} \neq s^{''}) \\ &\quad P(s_{m_1+1}^{''} \neq s^{''} | s_{m_1}^{''} \neq s^{''})(1 - \check{P}) \\ &\leq P(m_1 + 2 \leq n \leq m_1 + m_2 : s_n^{''} \neq s^{''} | s_{m_1}^{''} \neq s^{''} \text{ and } s_{m_1+1}^{''} \neq s^{''})(1 - \check{P})^2 \\ &\leq (1 - \check{P})^{m_2+1} \end{aligned}$$

Now consider the summation of these events for all m_2:

$$\begin{aligned} \sum_{m_2=0}^{\infty} P(m_1, m_2) &\leq \sum_{m_2=0}^{\infty} (1 - \check{P})^{m_2+1} \\ &= \frac{(1 - \check{P})}{1 - (1 - \check{P})} \\ &= \frac{1}{\check{P}} - 1 < \infty && \text{since } \check{P} \neq 0 \end{aligned}$$

$\therefore$ Since the sum of the probability sequence of events is finite, by the first Borel-Cantelli lemma (see Cantelli (1917) for details), the probability of occurring i.o. is zero.

$$P(((m_1, m_2), \text{i.o.})_{m_2}) = 0$$

Where $((m_1, m_2), \text{i.o.})_{m_2}$ is the event that the events (m_1, m_2) occurs i.o. when index by m_2. It is an event as well as it has a countable index (see Williams, 1991). Now

consider the summation of these events when over a different m_1:

$$\sum_{m_1=0}^{\infty} P(((m_1, m_2),\ \text{i.o.})_{m_2}) = 0 \neq \infty$$

$\therefore$ by the first Borel-Cantelli lemma:

$$P((((m_1, m_2),\ \text{i.o.})_{m_2},\ \text{i.o.})_{m_1}) = 0$$

Where $(((m_1, m_2),\ \text{i.o.})_{m_2},\ \text{i.o.})_{m_1})$ uses the countable sequence of events $((m_1, m_2),\ \text{i.o.})_{m_2}$ with m_1 as the indexed. It is difficult to determine what this event means so simplification is tried using $\{\text{Event}^c,\ \text{ev}\}^c = \{\text{Event, i.o.}\}$ (see Williams, 1991).

$$\begin{aligned}
1 =& P((((m_1, m_2),\ \text{i.o.})_{m_2},\ \text{i.o.})^c_{m_1}) \\
=& P((((m_1, m_2),\ \text{i.o.})^c_{m_2},\ \text{ev})_{m_1}) \\
=& P((((m_1, m_2)^c,\ \text{ev})_{m_2},\ \text{ev})_{m_1})
\end{aligned}$$

Consider

$$\begin{aligned}
((m_1, m_2)^c,\ ev)_{m_2} =& \{\omega \in \Omega : \exists m_2(\omega) \in \mathbb{N} \text{ s.t. } \quad \forall m_2 \geq m_2(\omega) \\
& \exists n \in [m_1, m_1 + m_2] \text{ s.t. } s''_n(\omega) = s''\}
\end{aligned}$$

$$\begin{aligned}
&\therefore (((m_1, m_2)^c,\ \text{ev})_{m_2},\ \text{ev})_{m_1} \\
&= \{\omega \in \Omega : \exists m_1(\omega) \in \mathbb{N} \text{ s.t. } \forall m_1 > m_1(\omega) \quad \exists m_2(m_1) \in \mathbb{N} \\
&\quad \text{s.t. } \forall m_2 \geq m_2(m_1) \quad \exists n \in [m_1, m_1 + m_2] \text{ s.t. } s''_n(\omega) = s''\} \\
&= \{\omega \in \Omega :\ \text{Given any } m_1 \in \mathbb{N} \quad \exists n \geq m_1 \text{ s.t. } s''_n(\omega) = s''\} \\
&= \{\omega \in \Omega : s''_n(\omega) = s'' \text{ i.o.}\}
\end{aligned}$$

□

It has been shown that each state s'' is visited infinitely often a.s. as long as the preceding action a^2 is visited infinitely often. Now the bounds on the Q-values are considered first before showing a^2 is visited infinitely often.

LEMMA 2. *Let $n \in \mathbb{N}$ be the n-th occurrance of an action a^2.*
Then for some $B > 0$

$$Q^2_n(a^2) \in [0, B)$$

Proof. For simplicity we abbreviate $Q_n := Q_n^2(a^2)$ and drop the P2 notation. By definition $Q_0(a^2) \in [0, B)$ for some $B > 0$. This will be proved by induction.
$Q_0 < B$
Now there is a need to show that if $Q_k < B$ then $Q_{k+1} < B$.
$Q_{k+1} = Q_k + \lambda_k(r(s_{k+1}^{''}) + L_{0,k+1}(s_{k+1}^{''}) - Q_k)$
Where $s_k^{''}$ is the state entered after a^2 has been selected. By definition both $r(s_{k+1}^{''})$ and $l_{0,k+1}(s_{k+1}^{''})$ are bounded by fixed number above. Therefore, B is chosen s.t. $r(s_{k+1}^{''})$, $l_{0,k+1}(s_{k+1}^{''}) < \frac{1}{2}B$

$$\begin{aligned} Q_{k+1} &< Q_k + \lambda_k(B - Q_k) \\ &= (1 - \lambda_k)Q_k + \lambda_n B \end{aligned}$$

Since $0 < \lambda_k < 1$ (by definition).

$$\begin{aligned} Q_{k+1} &\leq (1 - \lambda_k)B + \lambda_n B \\ &\leq B \end{aligned}$$

Similarly for $Q_0 \geq 0$ □

To make the proofs easier to follow, the following variables are defined assuming that a^2 has been visited n times before. F signifies the reward obtained by both players after action a^2 has been selected.

$$F_n^i(s^{''}(a^2)) = r^i(s^{''}(a^2)) + L_{0,n}^i(s^{''}(a^2))$$
$$F_*^i(s^{''}(a^2)) = r^i(s^{''}(a^2)) + L_{0,*}^i(s^{''}(a^2))$$

Since the customer model is invariant of episodes, this implies that its probability can be represented by a constant value, namely: $P(s^{''}) := P(s_n^{''}(a^{''}) = s^{''}(a^{''})|a^{''})$
Expected return is defined as:

$$E(F_n^i(s^{''}(a^2))) = \sum_{s^{''}(a^2) \in S^{''}(a^2)} E\left(r^i(s^{''}(a^2)) + L_{0,n}^i(s^{''}(a^2))\right) P(s^{''}(a^2))$$

Since $E(X+Y) = E(X) + E(Y)$

$$= \sum_{s''(a^2)\in S''(a^2)} E\left(r^i(s''(a^2)))\right) P(s''(a^2))$$
$$+ \sum_{s''(a^2)\in S''(a^2)} E\left(L^i_{0,n}(s''(a^2))\right) P(s''(a^2))$$

$$E(F^i_*(s''(a^2))) = \sum_{s''(a^2)\in S''(a^2)} E\left(r^i(s''(a^2)))\right) P(s''(a^2))$$
$$+ \sum_{s''(a^2)\in S''(a^2)} E\left(L^i_{0,*}(s''(a^2))\right) P(s''(a^2))$$

Now bounds can be put on F.

COROLLARY 3. *For $i \in \{0,1\}$, $\exists B > 0$ s.t.*

$$F^i_n, F^i_* \in [0, B)$$

Proof. This is shown in the proof of lemma 2 (with slight consideration for P1). □

Now it can be shown that each action a^2 has a posistive probability of occurring, no matter the history.

LEMMA 4. *Let $n \in \mathbb{N}$ be the n-th occurance of s'.*
$\exists \check{P} > 0$ s.t. given any history:

$$P(a^2_n(s') = a^2(s')) > \check{P} \text{ for } \forall n \in \mathbb{N}\backslash\{0\} \quad \forall a^2(s') \in A^2(s')$$

Proof. This is to say that $\exists \omega \in \Omega$ *s.t.* $h_{n-1}(\omega) = h_{n-1}$ *and* $a^2_n(s')(\omega) = a^2(s')$. However, measure theory is not needed here. Consider a fixed n and abbreviate notation by removing player and state identifiers. The trivial case $|A^2(s')| = 1$ is ignored. For fixed n, $\therefore Q(a) := Q^2_n(a(s'))$. Using Boltzmann action selection, it is known that:

$$P(a_n = a) = \frac{e^{Q(a)/\tau}}{e^{Q(a)/\tau} + \sum_{\substack{b\in A \\ b\neq a}} e^{Q(b)/\tau}}$$

Since $0 \leq Q(b) < B \quad \forall b \in A$ (from Lemma 2) and e^x is an increasing function.

$$> \frac{e^{Q(a)/\tau}}{e^{Q(a)/\tau} + \sum_{\substack{b\in A \\ b\neq a}} e^{B/\tau}}$$

$D := (|A| - 1)e^{B/\tau}$, $D \in (0, \infty)$ since $0 < B < \infty$ and $0 < \tau < \infty$

$$\begin{aligned} &= \frac{e^{Q(a)/\tau}}{e^{Q(a)/\tau} + D} \\ &= 1 - \frac{D}{e^{Q(a)/\tau} + D} \\ &\geq 1 - \frac{D}{1+D} = \frac{1}{1+D} \in (0, 1] \end{aligned}$$

$\therefore \check{P} := \frac{1}{1+D}$ □

It has been shown that each action has a positive chance of being selected so it can also be shown to be selected infinitely often too.

LEMMA 5. *If the state $s^{'}$ is visited infinitely often (i.o.) with probability 1 (a.s.) then action $a^2(s^{'})$ is visited i.o. a.s. for $\forall a^2(s^{'}) \in A^2(s^{'})$*

Proof. Let $n \in \mathbb{N}$ be the n-th occurance of $s^{'}$. By lemma 4, $P(a^2(s^{'})) > 0 \quad \forall a^2(s^{'}) \in A^2(s^{'})$. This means that the same arguments as in lemma 1 can be followed but with one exception. Unlike lemma 1, the probability of action selection is not independent over the episodes. However, from lemma 4, it was seen that the mininum value of selection is independent, i.e.

$$\begin{aligned} P(a^2_{n+1}(s^{'}) = a^2(s^{'})) &> \hat{P} > 0 \\ P(a^2_{n+1}(s^{'}) = a^2(s^{'})|a^2_n) &> \hat{P} > 0 \\ P(a^2_{n+1}(s^{'}) = a^2(s^{'})|h_n) &> \hat{P} > 0 \end{aligned}$$

Thus can follow similiar arguments as before. □

6.4 Properties of F

Now some simple properties of F are considered, these properties will be similiar to the properties that were imposed on $L^i_{0,n}$.

LEMMA 6. *Let $n \in \mathbb{N}$ be the n-th occurance of a^2.*

$$E(F^i_n(s^{''}_n(a^2))) \to E(F^i_*(s^{''}(a^2))) \text{ as } n \to \infty$$

Proof. For convenience all reference to the players is removed e.g. a^i. Thus $F_n := F_n^i(s_n^{''}(a^2))$ is used. By the definitions of $E(F_*)$ and $E(F_n)$, we see that:

$$|E(F_*) - E(F_n)| = \left| \sum_{s^{''} \in S^{''}} E\left(L_{0,*}(s^{''})\right) P(s^{''}) - \sum_{s^{''} \in S^{''}} E\left(L_{0,n}(s^{''})\right) P(s^{''}) \right|$$

By the triangle inequality

$$\leq \sum_{s^{''} \in S^{''}} \left| E\left(L_{0,*}(s^{''})\right) - E\left(L_{0,n}(s^{''})\right) \right| P(s^{''})$$

By the definition of L_0, it is known that given $\epsilon > 0 \quad \exists N_{s^{''}} \in \mathbb{N}$ *s.t.* $\forall n \geq N_{s^{''}}$

$$\left| E\left(L_{0,*}(s^{''})\right) - E\left(L_{0,n}(s^{''})\right) \right| < \epsilon$$

Since $|S^{''}| < \infty$, we choose $N = \max_{s^{''} \in S^{''}} \left(N_{s^{''}}\right) \quad \therefore \forall n \geq N$

$$\sum_{s^{''} \in S^{''}} \left| E\left(L_{0,*}(s^{''})\right) - E\left(L_{0,n}(s^{''})\right) \right| P(s^{''})$$
$$< \sum_{s^{''} \in S^{''}} \epsilon P(s^{''}) = \epsilon$$

□

Now it is shown that expected value of the average of observed F converges to the expectation of F_*.

LEMMA 7. *Let $n \in \mathbb{N}$ be the n-th occurance of a^2. $i \in \{0, 1\}$.*

$$E\left(\frac{\sum_{j=1}^{n} F_j^i(s_j^{''}(a^2))}{n}\right) \to E\left(F_*^i(s^{''}(a^2))\right) \text{ as } n \to \infty$$

Proof. For convenience, any reference in the notation to the player and $s^{''}$ are dropped. We set $F_j := F_j^i\left(s_j^{''}(a^2)\right)$. From lemma 6, we know that given $\epsilon_1 > 0$
$\exists N > 1$ *s.t.* $\forall n \geq N$

$$|E(F_*) - E(F_n)| \leq \epsilon_1$$

Now consider corollary 3, for some $\epsilon > 0$, $M > N$ s.t.

$$M > \frac{4(N-1)|E(F_*)|}{\epsilon}$$
$$\frac{\epsilon}{2} > \frac{\sum_{j=1}^{N-1} E(F_j)}{m} \qquad \forall m > M$$

Consider:

$$\left| E(\frac{\sum_{j=1}^{n} F_j}{n}) - E(F_*) \right|$$

Since $E(X+Y) = E(X) + E(Y)$ even if dependent

$$= \left| \frac{\sum_{j=1}^{m} E(F_j)}{m} - E(F_*) \right|$$
$$= \left| \frac{\sum_{j=1}^{N-1} E(F_j)}{m} + \frac{\sum_{j=N}^{m} E(F_j)}{m} - E(F_*) \right|$$

By the triangle inequality

$$\leq \left| \frac{\sum_{j=1}^{N-1} E(F_j)}{m} \right| + \left| \frac{\sum_{j=N}^{m} E(F_j)}{m} - E(F_*) \right|$$
$$\leq \frac{\epsilon}{2} + \frac{\sum_{j=N}^{m} |E(F_j) - E(F_*)|}{m} + \frac{N-1}{m} |E(F_*)|$$
$$< \frac{\epsilon}{2} + \frac{\sum_{j=N}^{m} |E(F_j) - E(F_*)|}{m} + \frac{\epsilon}{4(N-1)|E(F_*)|}(N-1)|E(F_*)|$$
$$< \frac{3\epsilon}{4} + \frac{\sum_{j=N}^{m} \epsilon_1}{m}$$
$$< \frac{3\epsilon}{4} + \epsilon_1$$

Set $\epsilon_1 < \frac{\epsilon}{4}$ □

Now lets consider the convergence of the covariance of F_n.

LEMMA 8. *Let $n, k, j \in \mathbb{N}$ be the n-th k-th and j-th occurances of a^2. $i \in \{1, 2\}$ For $k > j > n$*

$$cov(F_j^i(s_j^{''}(a^2)), F_k^i(s_k^{''}(a^2))) \to 0 \text{ as } n \to \infty$$

Proof. As before $F_j := F_j^i\left(s_j^{''}(a^2)\right)$. Also $s_j := s_j^{''}(a^2)$ and $t_k := s_k^{''}(a^2)$. Note that due to independence and fix nature of the customer model, $P(s_j \text{ and } t_k) =$

$P(s_j)P(t_k)$. Consider:

$$\begin{aligned}
&cov(F_j, F_k)\\
=&E(F_j, F_k) - E(F_j)E(F_k)\\
=&\sum_{s\in S}\sum_{t\in S} E\left((r(s) + L_{0,j}(s))\,(r(t) + L_{0,k}(t))\right) P(s)P(t)\\
&-\left(\sum_{s\in S} E\left(r(s) + L_{0,j}(s)\right) P(s)\right)\left(\sum_{t\in S} E\left(r(t) + L_{0,k}(t)\right) P(t)\right)
\end{aligned}$$

Since r is invariant of episodes, we have:

$$\begin{aligned}
=&\sum_{s\in S}\sum_{t\in S} E\left(L_{0,j}(s).l_{0,k}(t)\right) P(s)P(t)\\
&-\left(\sum_{s\in S} E\left(L_{0,j}(s)\right) P(s)\right)\left(\sum_{t\in S} E\left(L_{0,k}(t)\right) P(t)\right)\\
=&\sum_{s\in S}\sum_{t\in S}\Big(E\left(L_{0,j}(s).l_{0,k}(t)\right) - E\left(L_{0,j}(s)\right) E\left(L_{0,k}(t)\right)\Big)P(s)P(t)\\
=&\sum_{s\in S}\sum_{t\in S} cov\left(L_{0,j}(s), L_{0,k}(t)\right) P(s)P(t)
\end{aligned}$$

By definition of L_0 (see page 149), given any $\epsilon > 0$

$$\exists N > 0 \ s.t. \ \forall j,k > N \quad \left|cov\left(L_{0,j}(s), L_{0,k}(t)\right)\right| < \epsilon$$

$$\begin{aligned}
<&\sum_{s\in S}\sum_{t\in S} \epsilon P(s)P(t)\\
=&\epsilon
\end{aligned}$$

Similiarly, $cov(F_j, F_k) > -\epsilon$ □

LEMMA 9. *Let $n \in \mathbb{N}$ be the n-th occurance of a^2. $i \in \{0, 1\}$.*

$$var\left(\frac{\sum_n^{j=1} F_j^i(s_j''(a^2))}{n}\right) \xrightarrow{P} 0 \ as \ n \to \infty$$

Proof. Set $F_j := F_j^i\left(s_j''(a^2)\right)$.

From corollary 3, $F_j \in [0, B)$. This implies that:

$$\begin{aligned}
var(F_j) &< B^2\\
cov(F_j, F_k) &< B^2
\end{aligned}$$

Using standard formula (see Winston, 1993), it is known that

$$\left| var\left(\frac{\sum_{j=1}^{n} F_j}{n}\right)\right|$$
$$= \left| \frac{1}{n^2}\left(\sum_{j=1}^{n} var(F_j) + 2\sum_{j=1}^{n-1}\sum_{k>j}^{n} cov(F_j, F_k)\right)\right|$$

By bounds and the triangle inequality

$$\leq \left|\frac{1}{n^2} n.B^2\right| + \frac{2}{n^2}\sum_{j=1}^{n-1}\sum_{k>j}^{n} |cov(F_j, F_k)|$$

From lemma 8, it is known that given $\epsilon > 0 \quad \exists N > 0 \ s.t. \ \forall j, k > N$

$$|cov(F_j, F_k)| < \frac{\epsilon}{5}$$

$\therefore$ Choose $n > \max\{N, \frac{5B^2}{\epsilon}, \frac{5(N-1)B^2}{\epsilon}\}$

$$\left| var\left(\frac{\sum_{j=1}^{n} F_j}{n}\right)\right|$$
$$< \frac{\epsilon}{5} + \frac{2}{n^2}\sum_{j=N}^{n-1}\sum_{k>j}^{n} |cov(F_j, F_k)| + \frac{2}{n^2}\sum_{j=1}^{N-1}\sum_{k>j}^{n} |cov(F_j, F_k)|$$
$$< \frac{\epsilon}{5} + \frac{2}{n^2}\sum_{j=N}^{n-1}\sum_{k>j}^{n} \frac{\epsilon}{5} + \frac{2}{n^2}\sum_{j=1}^{N-1}(n-j)B^2$$
$$< \frac{\epsilon}{5} + \frac{2}{n^2}\sum_{j=N}^{n-1}(n-j)\frac{\epsilon}{5} + \frac{2}{n^2}\sum_{j=1}^{N-1} n.B^2$$
$$< \frac{\epsilon}{5} + \frac{2}{n^2}\sum_{j=N}^{n-1} n\frac{\epsilon}{5} + \frac{2}{n}(N-1).B^2$$
$$< \frac{\epsilon}{5} + \frac{2\epsilon}{5} + \frac{2\epsilon}{5}$$
$$= \epsilon$$

□

By showing that the variance of the summation of F converges to zero, is effectively the same as showing that it will always converge to a fixed value, regardless of variation. This is confirmed in the following lemma. This is the first major proof within

the framework. This proof is broadly based on the *stochastic approximation* work of Robbins and Monro (see Robbins and Monro, 1951). It is similiar to the weak law of large numbers but it must be remembered that the distribution of the F's is changing and that they are correlated. It is this correlation that prevents the proving these results using *strong* (or a.s.) convergence. Also, it is noted that the expected value keeps changing with n, however this has been fixed into position with lemma 7.

LEMMA 10. *Let $n \in \mathbb{N}$ be the n-th occurance of a^2. $i \in \{0, 1\}$*

$$\frac{\sum_{j=1}^{n} F_j^i(s_j''(a^2))}{n} \xrightarrow{P} E(\frac{\sum_{j=1}^{n} F_j^i(s_j''(a^2))}{n}) \text{ as } n \to \infty$$

Proof. Set $F_j := F_j^i\left(s_j''(a^2)\right)$.
By Chebyshev Inequality (see Chebyshev (1867)), given $\epsilon > 0$:

$$P(|\frac{\sum_{j=1}^{n} F_j}{n} - E(\frac{\sum_{j=1}^{n} F_j}{n})| > \epsilon) \leq \frac{var(\frac{\sum_{j=1}^{n} F_j}{n})}{\epsilon}$$

By lemma 9, $\exists N$ s.t. $\forall n \geq N$:

$$var(\frac{\sum_{j=1}^{n} F_j}{n}) < \delta.\epsilon$$

$\therefore \forall n > N$

$$P(|\frac{\sum_{j=1}^{n} F_j}{n} - E(\frac{\sum_{j=1}^{n} F_j}{n})| > \epsilon) < \delta$$

□

LEMMA 11. *Let $n \in \mathbb{N}$ be the n-th occurance of a^2. $i \in \{0, 1\}$*

$$\frac{\sum_{j=1}^{n} F_j^i(s_j''(a^2))}{n} \xrightarrow{P} E(F_*^i(s''(a^2))) \text{ as } n \to \infty$$

Proof. Set $F_j := F_j^i\left(s_j''(a^2)\right)$.
By lemma 10, Consider $\epsilon_1, \delta_1 > 0 \quad N_1 > 0$ s.t. $\forall n \geq N_1$:

$$P\left(\left|\frac{\sum_{j=1}^{n} F_j}{n} - E(\frac{\sum_{j=1}^{n} F_j}{n})\right| > \epsilon_1\right) < \delta_1$$

By lemma 7, consider $\epsilon_2 > 0 \quad N_2 > 0$ s.t. $\forall n \geq N_2$:

$$\left|E\left(\frac{\sum_{j=1}^{n} F_j}{n}\right) - E(F_*)\right| < \epsilon_2$$

$$\therefore P\left(\left|\frac{\sum_{j=1}^{n} F_j}{n} - E(F_*)\right| > \epsilon\right) \quad \text{By triangle inequality}$$
$$\leq P\left(\left|\frac{\sum_{j=1}^{n} F_j}{n} - E\left(\frac{\sum_{j=1}^{n} F_j}{n}\right)\right| + \left|E\left(\frac{\sum_{j=1}^{n} F_j}{n}\right) - E(F_*)\right| > \epsilon\right)$$
$$\leq P\left(\left|\frac{\sum_{j=1}^{n} F_j}{n} - E\left(\frac{\sum_{j=1}^{n} F_j}{n}\right)\right| + \epsilon_2 > \epsilon\right)$$

Set $\epsilon_1 = \epsilon_2 < \frac{\epsilon}{2}$ *and* $\delta_1 = \delta$

$$\leq P\left(\left|\frac{\sum_{j=1}^{n} F_j}{n} - E\left(\frac{\sum_{j=1}^{n} F_j}{n}\right)\right| > \frac{\epsilon}{2}\right)$$
$$< \delta$$

□

6.5 Properties of Q

Now that the properties of F have been looked at, it is possible to use these properties to find out if they hold for the Q-values as well. Before applying this, a number of generic proofs were derived to help simplify the notation for the Q-values updating formulas.

LEMMA 12. *If* $j, n \in \mathbb{N}$ *and* $C \in \mathbb{R}$ *then*

$$\frac{1}{j+C}\prod_{i=j+1}^{n}(1 - \frac{1}{i+C}) = \frac{1}{n+C}$$

Proof.

$$\frac{1}{j+C}\prod_{i=j+1}^{n}(1 - \frac{1}{i+C})$$
$$= \frac{1}{j+C}\prod_{i=j+1}^{n}\frac{i+C-1}{i+C}$$
$$= \frac{1}{j+C}(\frac{j+C}{j+C+1})(\frac{j+C+1}{j+C+2})\cdots(\frac{n+C-1}{n+C})$$
$$= \frac{1}{n+C}$$

□

LEMMA 13. *Consider a generic Q-value with* $\lambda_n = \frac{1}{n+C}$ *for some* $C \in \mathbb{R}$ *then the iterative sequence:*

$$Q_{n+1} = (1 - \lambda_{n+1})\, Q_n + \lambda_{n+1} r_{n+1}$$

Is equivlent to (for $n > 0$*):*

$$Q_n = Q_0 \prod_{i=1}^{n} (1 - \lambda_i) + \frac{n}{n + C} \sum_{i=1}^{n} \frac{r_i}{n}$$

Proof. Consider the iterative process:

$$\begin{aligned}
Q_1 &= (1 - \lambda_1) Q_0 + \lambda_1 r_1 \\
Q_2 &= (1 - \lambda_2) Q_1 + \lambda_2 r_2 \\
&= (1 - \lambda_2)((1 - \lambda_1) Q_0 + \lambda_1 r_1) + \lambda_2 r_2 \\
&= (1 - \lambda_1)(1 - \lambda_2) Q_0 + (1 - \lambda_2)\lambda_1 r_1 + \lambda_2 r_2 \\
Q_3 &= Q_0 \prod_{i=1}^{3} (1 - \lambda_i) + r_1 \lambda_1 \prod_{i=2}^{3} (1 - \lambda_i) + r_2 \lambda_2 \prod_{i=3}^{3} (1 - \lambda_i) + \lambda_3 r_3 \\
Q_4 &= Q_0 \prod_{i=1}^{4} (1 - \lambda_i) + r_1 \lambda_1 \prod_{i=2}^{4} (1 - \lambda_i) + r_2 \lambda_2 \prod_{i=3}^{4} (1 - \lambda_i) \\
&\quad + r_3 \lambda_3 \prod_{i=4}^{4} (1 - \lambda_i) + \lambda_4 r_4 \\
&\vdots \\
Q_n &= Q_0 \prod_{i=1}^{n} (1 - \lambda_i) + \sum_{j=1}^{n-1} r_j \lambda_j \prod_{i=j+1}^{n} (1 - \lambda_i) + \lambda_n r_n
\end{aligned}$$

Given $\lambda_n = \frac{1}{n+C}$, by Lemma 12:

$$\begin{aligned}
Q_n &= Q_0 \prod_{i=1}^{n} (1 - \lambda_i) + \sum_{j=1}^{n} \frac{r_j}{n + C} \\
Q_n &= Q_0 \prod_{i=1}^{n} (1 - \lambda_i) + \frac{n}{n + C} \sum_{i=1}^{n} \frac{r_i}{n}
\end{aligned}$$

□

LEMMA 14. *If* $\lambda_n = \frac{1}{n+C}$ *for some* $C \in \mathbb{N} \backslash \{0\}$ *then*

$$\prod_{i=1}^{n} (1 - \lambda_i) \longrightarrow 0 \text{ as } n \longrightarrow \infty$$

Proof.

$$\begin{aligned}
\prod_{i=1}^{2}(1-\lambda_i) &= (1-\frac{1}{C+1})(1-\frac{1}{C+2}) \\
&= (\frac{C}{C+1})(\frac{C+1}{C+2}) \\
&= (\frac{C}{C+2}) \\
\prod_{i=1}^{3}(1-\lambda_i) &= (\frac{C}{C+2})(1-\frac{1}{C+3}) \\
&= (\frac{C}{C+3}) \\
&\vdots \\
\prod_{i=1}^{n}(1-\lambda_i) &= \frac{C}{C+n} \longrightarrow 0 \text{ as } n \longrightarrow \infty
\end{aligned}$$

□

Now that the identities have found using the above three proofs, it is possible to prove convergence for the Q-values.

LEMMA 15. *Let $n \in \mathbb{N}$ be the n-th occurance of a^2.*

$$Q_n^2(a^2) \xrightarrow{P} Q_*^2(a^2) \text{ as } n \to \infty$$

Where:

$$Q_*^2(a^2) := E(r^2(s''(a^2)) + L_{0,*}^2(s''(a^2))) = E(F(s''(a^2)))$$

Proof. First abbreviate $Q_n := Q_n^2(a^2)$, etc. From lemma 13 is the following:

$$Q_n = Q_0 \prod_{i=1}^{n}(1-\lambda_i) + \frac{n}{n+C}\sum_{i=1}^{n}\frac{F_i}{n}$$

From lemma 11, given $\delta_1, \epsilon, \epsilon_1 > 0 \quad \exists N_1 > 0$ s.t. $\forall n \geq N_1$:

$$P\left(\left|\frac{\sum_{j=1}^{n} F_j}{n} - E(F_*)\right| > \epsilon_1\right) < \delta_1$$

$$\therefore P\left(|Q_n - E(F_*)| > \epsilon\right)$$
$$= P\left(\left|Q_0 \prod_{i=1}^{n}(1-\lambda_i) + \frac{n}{n+C}\sum_{j=1}^{n}\frac{F_j}{n} - E(F_*)\right| > \epsilon\right)$$
$$\leq P\left(\left|Q_0 \prod_{j=1}^{n}(1-\lambda_j)\right| + \left|\frac{n}{n+C}\right|\left|\sum_{j=1}^{n}\frac{F_i}{n} - E(F_*)\right| + \left|\frac{C}{n+C}\right||E(F_*)| > \epsilon\right)$$

From lemma 14, $\exists N_2 > 0$ *s.t.* $\forall n \geq N_2$:

$$\left|\hat{Q}_0 \prod_{i=1}^{n}(1-\lambda_i)\right| < \frac{\epsilon}{4}$$

From corollary 3, it is known that $\exists N_3 > 0$ *s.t.* $\forall n \geq N_3$:

$$\left|\frac{C}{n+C}\right|\left|E(\hat{F}^*)\right| < \frac{\epsilon}{4}$$

We choose $N \in \mathbb{N}$ s.t. $N > max\{N_1, N_2, N_3\}$ and $\forall n > N$

$$\therefore P\left(|Q_n - E(F_*)| > \epsilon\right)$$
$$\leq P\left(\left|\frac{n}{n+C}\right|\left|\sum_{j=1}^{n}\frac{F_j}{n} - E(F_*)\right| > \frac{\epsilon}{2}\right)$$
$$\leq P\left(\left|\sum_{j=1}^{n}\frac{F_j}{n} - E(F_*)\right| > \frac{\epsilon}{2}\right)$$

$\therefore$ Set $\epsilon_1 := \frac{\epsilon}{2}$ and $\delta_1 := \delta$ □

So it has been shown that player two's Q-values converge, in probability, to the expected values of F^2. This is a great starting point for showing that they converge to the expected values under the variation to the Nash Distribution policy but the complete proof will have to wait until the inductive step later in the chapter. The next three proofs are concerned with generic random processes which converge to a fixed value (like our F and Q values). These results can then be used to show convergence of the next level of L values.

LEMMA 16. *Let $X_n \in [0, B)$ be a sequence of random variables, $x \in \mathbb{R}$ s.t.*

$$X_n \xrightarrow{P} x \text{ as } n \to \infty$$

Then

$$E(X_n) \to x \text{ as } n \to \infty$$

Proof. Given $\epsilon_1, \delta_1 \quad \exists N$ s.t. $\forall n \geq N$

$$P(|X_n - x| > \epsilon_1) < \delta_1$$

$$\begin{aligned}\therefore E(X_n) &< (x + \epsilon_1)P(|X_n - x| \leq \epsilon_1) + B.P(|X_n - x| > \epsilon_1) \\ &< (x + \epsilon_1) + B.\delta_1\end{aligned}$$

Similiarly, $E(X_n) > x - \epsilon_1$

Set $\epsilon_1 + B.\delta_1 < \epsilon$

$$|E(X_n) - x| < \epsilon$$

□

LEMMA 17. *Let $X_n \in [0, B)$ and $Y_n \in [0, B)$ be sequences of random variables, $x, y \in \mathbb{R}$ s.t.*

$$X_n \xrightarrow{P} x \text{ as } n \to \infty$$
$$Y_n \xrightarrow{P} y \text{ as } n \to \infty$$

Then

$$E(X_i Y_j) \to x.y \text{ as } i, j > n \to \infty$$

X_i and Y_j are not necessary independent [4]

Proof. Given $\epsilon_1, \delta_1, \epsilon_2, \delta_2 > 0 \quad \exists N$ s.t. $\forall i, j \geq N$

$$P(|X_i - x| > \epsilon_1) < \delta_1$$
$$P(|Y_j - y| > \epsilon_2) < \delta_2$$

$$\begin{aligned}&E(X_i.Y_j) \\ \leq&(x + \epsilon_1)(y + \epsilon_2)P(\{|X_i - x| \leq \epsilon_1\} \cap \{|Y_j - y| \leq \epsilon_2\}) \\ &+ B^2.P(\{|X_i - x| > \epsilon_1\} \cap \{|Y_j - y| > \epsilon_2\}) \\ &+ B^2.P(|Y_j - y| > \epsilon_2) \\ \leq&(x + \epsilon_1)(y + \epsilon_2) + B^2.P(|X_i - x| > \epsilon_1) + B^2.P(|Y_j - y| > \epsilon_2) \\ \leq&x.y + x.\epsilon_2 + y.\epsilon_1 + B^2.(\delta_1 + \delta_2) + \epsilon_1.\epsilon_2\end{aligned}$$

[4] Could even be the same random variable.

Similiarly:

$$E(X_i.Y_j) \geq x.y - x.\epsilon_2 - y.\epsilon_1$$

Set $x.y + x.\epsilon_2 + y.\epsilon_1 + B^2.(\delta_1 + \delta_2) + \epsilon_1.\epsilon_2 < \epsilon$

$$|E(X_i.Y_j) - x.y| < \epsilon$$

□

LEMMA 18. *Let $X_n \in [0, B)$ and $Y_n \in [0, B)$ be sequences of random variables, $x, y \in \mathbb{R}$ s.t.*

$$X_n \xrightarrow{P} x \text{ as } n \to \infty$$
$$Y_n \xrightarrow{P} y \text{ as } n \to \infty$$

Then

$$cov\,(X_i, Y_j) \to 0 \text{ as } i, j > n \to \infty$$

X_i and Y_j are not necessary independent.

Proof.

$$cov\,(X_i, Y_j) = E\,(X_i, Y_j) - E(X_i)E(Y_j)$$

Given $\epsilon_1, \epsilon_2, \epsilon_3 > 0 \quad \exists N$ s.t. $\forall i, j \geq N$:

$$|E(X_i) - x| < \epsilon_1$$
$$|E(Y_j) - y| < \epsilon_2 \quad \text{From lemma 16}$$
$$|E(X_i.Y_j) - x.y| < \epsilon_3 \quad \text{From lemma 17}$$

As an arbitrarily large number could be added to each of each sequence, w.l.o.g. that $x, y >> 0$. $\therefore$ From arguments in lemma 16:

$$E(X_i)E(Y_j) \geq (x - \epsilon_1)(y - \epsilon_2)$$
$$\geq x.y - (x.\epsilon_2 + y.\epsilon_1 - \epsilon_1.\epsilon_2)$$
$$E(X_i)E(Y_j) \leq (x + \epsilon_1 + B.\delta_1)(y + \epsilon_2 + B.\delta_2)$$
$$\leq x.y + (x.\epsilon_2 + y.\epsilon_1 + \epsilon_1.\epsilon_2 + y.B.\delta_1 + x.B.\delta_2 + B^2.\delta_1.\delta_2)$$

$\therefore$ Choose $\epsilon_1, \delta_1, \epsilon_2, \delta_2$ s.t.

$$\max\left(x.\epsilon_2 + y.\epsilon_1 - \epsilon_1.\epsilon_2, \quad x.\epsilon_2 + y.\epsilon_1 + \epsilon_1.\epsilon_2 + y.B.\delta_1 + x.B.\delta_2 + B^2.\delta_1.\delta_2\right) < \frac{\epsilon}{2}$$

$$\begin{aligned} &|E(X_i.Y_j) - E(X_i).E(Y_j)| \\ \leq &|E(X_i.Y_j) - x.y| + |x.y - E(X_i).E(Y_j)| \end{aligned}$$

From the choice of ϵ_1 and ϵ_2 and choosing $\epsilon_3 < \frac{\epsilon}{2}$:

$$< \epsilon$$

□

Now these general lemmas can now be applied to our Q-values to gain the following two results:

COROLLARY 19. *Let $n \in \mathbb{N}$ some episode. a^2 and b^2 are fixed actions.*

$$cov\left(Q_i^2(a^2), Q_j^2(b^2)\right) \xrightarrow{P} 0 \text{ as } j \geq i > n \to \infty \quad (i \neq j \text{ if } h = g)$$

Proof. As fixed a^2 and b^2 are considered, it is important to note that each Q-value remains the same until that action is selected again. Since from lemma 2 all $Q_i \in [0, B)$ and by lemma 15 there is convergence to a fixed value so can directly apply lemma 18. □

The following lemma shows that the probability of action selection for player Two convergences towards $Q_* := E(F_*)$, thus it is expected that the probabilities of the action selection to reflect the expected values. This means that if F_* are the values observed under a Nash Distribution policy then the action selection for player Two is also a Nash Distribution policy (for this part of the game tree).

LEMMA 20. *Let $n \in \mathbb{N}$ be the n-th occurance of s'.*

$$P(a_n^2(s^{'}) = a^2(s^{'})) \xrightarrow{P} P(a^2(s^{'})) \text{ as } n \to \infty$$

Where, from boltzmann action selection:

$$P(a^2(s^{'})) = \frac{e^{Q_*^2(a^2(s^{'}))/\tau}}{e^{Q_*^2(a^2(s^{'}))/\tau} + \sum_{\substack{b^2(s^{'}) \in A^2(s^{'}) \\ b^2(s^{'}) \neq a^2(s^{'})}} e^{Q_*^2(b^2(s^{'}))/\tau}}$$

Proof. All unnecessary notation has been removed for claification reasons.

By lemma 15, given $\delta_1, \epsilon_1 > 0 \quad \exists n \geq N \quad$ s.t. $\forall n \geq N$

$$P\left(|Q_n(a) - Q_*(a)| > \epsilon_1\right) < \delta_1 \qquad \forall a \in A$$

$$\begin{aligned}
&\therefore P\left(\exists a \in A \text{ s.t. } |Q_n(a) - Q_*(a)| > \epsilon_1\right) \\
&\leq P\left(\{|Q_n(a_1) - Q_*(a_1)| > \epsilon_1\} \cup \ldots \cup \left\{\left|Q_n(a_{|A|}) - Q_*(a_{|A|})\right| > \epsilon_1\right\}\right)
\end{aligned}$$

By Boole's inequality

$$\begin{aligned}
&\leq \sum_{a \in A} P\left(|Q_n(a) - Q_*(a)| > \epsilon_1\right) \\
&< |A|\delta_1
\end{aligned}$$

$$\begin{aligned}
&\therefore P\left(\exists a \in A \text{ s.t. } |Q_n(a) - Q_*(a)| > \epsilon_1\right) < |A|\delta_1 \\
&\quad P\left(\forall a \in A \text{ s.t. } |Q_n(a) - Q_*(a)| \leq \epsilon_1\right) > 1 - |A|\delta_1
\end{aligned}$$

$$\epsilon_1 := \frac{\tau}{2} \text{ In}\left(1 + \epsilon \frac{\sum_{b \in A} e^{Q_*(b)}}{\max_{c \in A} e^{Q_*(c)/\tau}}\right)$$

Now consider when $|Q_n(a) - Q_*(a)| \leq \epsilon_1 \quad \forall a \in A$

$$\begin{aligned}
P(a_n = a) &= \frac{e^{Q_n(a)/\tau}}{e^{Q_n(a)/\tau} + \sum_{\substack{b \in A \\ b \neq a}} e^{Q_n(b)/\tau}} \\
&\leq \frac{e^{Q_n(a)/\tau}}{e^{Q_n(a)/\tau} + \sum_{\substack{b \in A \\ b \neq a}} e^{\frac{Q_*(b) - \epsilon_1}{\tau}}} \\
&= \frac{e^{\frac{Q_n(a) + \epsilon_1}{\tau}}}{e^{\frac{Q_n(a) + \epsilon_1}{\tau}} + \sum_{\substack{b \in A \\ b \neq a}} e^{\frac{Q_*(b)}{\tau}}} \\
&\leq \frac{e^{\frac{Q_n(a) + \epsilon_1}{\tau}}}{e^{Q_*(a)/\tau} + \sum_{\substack{b \in A \\ b \neq a}} e^{\frac{Q_*(b)}{\tau}}} \\
&\leq \frac{e^{\frac{Q_*(a) + 2\epsilon_1}{\tau}}}{\sum_{b \in A} e^{\frac{Q_*(b)}{\tau}}} \\
&= \frac{e^{Q_*(a)/\tau}}{\sum_{b \in A} e^{\frac{Q_*(b)}{\tau}}} e^{2\epsilon_1/\tau} \\
&= P(a) e^{In\left(1 + \epsilon \frac{\sum_{b \in A} e^{Q_*(b)/\tau}}{\max_{i \in A} e^{Q_*(i)/\tau}}\right)}
\end{aligned}$$

$$
\begin{aligned}
&= P(a)\left(1+\epsilon\frac{\sum_{b\in A}e^{Q_*(b)/\tau}}{\max_{i\in A}e^{Q_*(i)/\tau}}\right)\\
&= P(a)+\epsilon\frac{Q_*(a)/\tau}{\max_{i\in A}e^{Q_*(i)/\tau}}\frac{\sum_{b\in A}e^{Q_*(b)/\tau}}{\sum_{b\in A}e^{Q_*(b)/\tau}}\\
&\leq P(a)+\epsilon
\end{aligned}
$$

Similiarly, the other way round:

$$
\begin{aligned}
P(a_n=a) &= \frac{e^{Q_n(a)/\tau}}{e^{Q_n(a)/\tau}+\sum_{\substack{b\in A\\ b\neq a}}e^{Q_n(b)/\tau}}\\
&\geq \frac{e^{Q_n(a)/\tau}}{e^{Q_n(a)/\tau}+\sum_{\substack{b\in A\\ b\neq a}}e^{\frac{Q_*(b)+\epsilon_1}{\tau}}}\\
&= \frac{e^{\frac{Q_n(a)-\epsilon_1}{\tau}}}{e^{\frac{Q_n(a)-\epsilon_1}{\tau}}+\sum_{\substack{b\in A\\ b\neq a}}e^{\frac{Q_*(b)}{\tau}}}\\
&\geq \frac{e^{\frac{Q_*(a)-2\epsilon_1}{\tau}}}{\sum_{b\in A}e^{\frac{Q_*(b)}{\tau}}}\\
&= \frac{e^{Q_*(a)/\tau}}{\sum_{b\in A}e^{\frac{Q_*(b)}{\tau}}}e^{-2\epsilon_1/\tau}\\
&= P(a)\frac{1}{1+\epsilon\frac{\sum_{b\in A}e^{Q_*(b)/\tau}}{max_{i\in A}(e^{Q_*(i)/\tau})}}\\
&\geq P(a)\frac{1}{1+\epsilon\frac{1}{P(a)}}\\
&= P(a)\frac{P(a)}{P(a)+\epsilon}\\
&= P(a)\left(1-\frac{\epsilon}{P(a)+\epsilon}\right)\\
&= P(a)-\epsilon\frac{P(a)}{P(a)+\epsilon}\\
&\geq P(a)-\epsilon
\end{aligned}
$$

This means that if $|Q_n(a)-Q_*(a)|\leq\epsilon_1 \quad \forall a\in A$ then $|P(a)-P(a)|<\epsilon \quad \therefore$ given $a\in A$:

$$
\begin{aligned}
&P\left(|P(a_n=a)-P(a)|\leq\epsilon\right)\\
\geq &P\left(|Q_n(a)-Q_*(a)|\leq\epsilon_1 \quad \forall a\in A\right)\\
> &1-|A|\delta_1
\end{aligned}
$$

Taking $\delta_1 < \frac{\delta}{|A|}$

$$>1-\delta$$

$\therefore P\left(|P(a_n)-P(a)|>\epsilon\right)<\delta \quad \forall a \in A$

Thus all actions have been solved for at once. □

6.6 Properties of L

Now a new random variable $L^i_{1,n}$ is introduced, where $i \in \{1,2\}$ is the player indicator and n is the n-th time that a^1 is selected (where $_*$ is reference to the ideal distribution). For player Two, this is defined as follows:

$$L^2_{1,n}(s^{'}) = Q^2_n(a^2_n(s^{'}))$$
$$L^2_{1,*}(s^{'}) = Q^2_*(a^2_*(s^{'}))$$

Where $a^2_*(s^{'}) \in A^2_*(s^{'})$ has the distribution:

$$P(a^2_*(s^{'}) = a^2(s^{'})) = P(a^2(s^{'}))$$

The expected value is defined as follows:

$$E\left(L^2_{1,n}(s^{'})\right) = \sum_{\substack{s^{'} \\ \in S^{'}}} \sum_{\substack{a^2(s^{'}) \\ \in A^2(s^{'})}} E\left(Q^2_n(a^2(s^{'}))\right) P(a^2_n(s^{'}) = a^2(s^{'})|s^{'})P(s^{'})$$
$$E\left(L^2_{1,*}(s^{'})\right) = \sum_{\substack{s^{'} \\ \in S^{'}}} \sum_{\substack{a^2(s^{'}) \\ \in A^2(s^{'})}} E\left(Q^2_*(a^2(s^{'}))\right) P(a^2(s^{'})|s^{'})P(s^{'})$$

However, this is defined differently for player One, to reflect that they have not had a chance to update there Q-values.

$$L^1_{1,n}(s^{'}) = F^1_n(s^{''}(a^2_n(s^{'})))$$
$$L^1_{1,*}(s^{'}) = F^1_*(s^{''}(a^2_*(s^{'})))$$

Where $s^{''}$ has the standard fixed distribution from the customer model. The expected values are as follows:

$$E\left(L^1_{1,n}(s^{'})\right) = \sum_{\substack{s^{'} \\ \in S^{'}}} \sum_{\substack{a^2(s^{'}) \\ \in A^2(s^{'})}} \sum_{\substack{s^{''}(a^2(s^{'})) \\ \in S^{''}(a^2(s^{'}))}} \begin{matrix} E\left(F^1_n(s^{''}(a^2(s^{'})))\right) P(s^{''}_n(a^2(s^{'})) = s^{''}(a^2(s^{'}))|a^2(s^{'})) \\ P(a^2_n(s^{'}) = a^2(s^{'})|s^{'})P(s^{'}) \end{matrix}$$

Also, a very long formula for the optimal version

$$E\left(L^1_{1,*}(s^{'})\right) = \sum_{\substack{s^{'} \\ \in S^{'}}} \sum_{\substack{a^2(s^{'}) \\ \in A^2(s^{'})}} \sum_{\substack{s^{''}(a^2(s^{'})) \\ \in S^{''}(a^2(s^{'}))}} \frac{E\left(F^1_*(s^{''}(a^2(s^{'})))\right) P(s^{''}(a^2(s^{'}))|a^2(s^{'}))}{P(a^2(s^{'})|s^{'})P(s^{'})}$$

The equations above are difficult to read, hence why there is a tendency to abbreviate the notation where ever possible. Now if it can be shown that $L^i_{1,n}$ has the same properties as $L^i_{0,n}$, it would be possible to use induction to complete the proof. Therefore, is needs to be shown that if n is the n-th selection of a^1 then:

1. $\exists B > 0 \ s.t. \ L^i_{1,n}(s^{'}) \in [0, B) \quad \forall n \in \mathbb{N} \quad \forall s^{'} \in S^{'}$
2. $E(L^i_{1,n}(s^{'})) \rightarrow E(L^i_{1,*}(s^{'})) \ as \ n \rightarrow \infty \quad \forall s^{'} \in S^{'}$
3. $For \ k > j > n, \quad cov(L^i_{1,j}(s^{'}), L^i_{1,k}(t^{'})) \rightarrow 0 \quad n \rightarrow \infty \quad \forall s^{'}, t^{'} \in S^{'}$

Now proving these statements can begin, starting with showing the bounds work.

COROLLARY 21. *Let n be the n-th occurance of action a^1*
$\exists B > 0 \ s.t.$

$$L^i_{1,n}(s^{'}) \in [0, B)$$
$$L^i_{1,*}(s^{'}) \in [0, B)$$

Proof. Case i = 2
Since $L^2_{1,n}(s^{'}) = Q^2_n(a^2_n(s^{'}))$ the same bounds hold as they do for the Q-values. Therefore, this hold directly from lemma 2.
Case i = 1
Since $L^1_{1,n}(s^{'}) = F^1_n(s^{''}(a^2_n(s^{'})))$ is bounded by corollary 3. □

Now the remaining properties for player Two's L_1 values can be shown.

LEMMA 22. *Let n be the n-th occurance of action a^1*

$$E(L^2_{1,n}(s^{'})) \rightarrow E(L^2_{1,*}(s^{'}))$$

Proof. The notation is abbreviated where possible. Given $\epsilon_1, \delta_1 > 0$ by lemma 20 $\exists N_1 > 0 \ s.t \ \forall n > N_1, \forall a(s^{'}) \in A(s^{'}) \ and \ fixed \ s^{'} \in S^{'}$

$$P\left(\left|P(a_n(s^{'}) = a(s^{'})) - P(a(s^{'}))\right| > \epsilon_1\right) < \delta_1$$

By lemma 15 and lemma 16, given $\delta_2 > 0, \quad \exists N_2 > 0$ *s.t.* $\forall n > N_2$, *and* $\forall$ *fixed* $a(s^{'}) \in A(s^{'})$

$$\left|E(Q_n(a(s^{'}))) - E(Q_*(a(s^{'})))\right| < \delta_2$$

Consider $n \geq \max\{N_1, N_2\}$

$$\begin{aligned}
&E\left(L_{1,n}(s^{'})\right) \\
=&\sum_{\substack{s^{'} \\ \in S^{'}}} \sum_{\substack{a(s^{'}) \\ \in A(s^{'})}} E\left(Q_n(a(s^{'}))\right) P(a_n(s^{'}) = a(s^{'})|s^{'}) P(s^{'}) \\
\leq&\sum_{\substack{s^{'} \\ \in S^{'}}} \sum_{\substack{a(s^{'}) \\ \in A(s^{'})}} \left(E\left(Q_*(a(s^{'}))\right) + \delta_2\right) P(a_n(s^{'}) = a(s^{'})|s^{'}) P(s^{'}) \\
\leq&\sum_{\substack{s^{'} \\ \in S^{'}}} \sum_{\substack{a(s^{'}) \\ \in A(s^{'})}} E\left(Q_*(a(s^{'}))\right) P(a_n(s^{'}) = a(s^{'})|s^{'}) P(s^{'}) + \delta_2 \\
\leq&\sum_{\substack{s^{'} \\ \in S^{'}}} \sum_{\substack{a(s^{'}) \\ \in A(s^{'})}} E\left(Q_*(a(s^{'}))\right) P(a_n(s^{'}) = a(s^{'})|s^{'}, |P(a_n(s^{'}) = a(s^{'})|s^{'}) \\
&- P(a(s^{'})|s^{'})| \leq \epsilon_1) P\left(\left|P(a_n(s^{'}) = a(s^{'})|s^{'}) - P(a(s^{'})|s^{'})\right| \leq \epsilon_1\right) P(s^{'}) \\
&+ \sum_{s^{'} \in S^{'}} \sum_{\substack{a(s^{'}) \\ \in A(s^{'})}} E\left(Q_*(a(s^{'}))\right) P(a_n(s^{'}) = a(s^{'})|s^{'}, |P(a_n(s^{'}) = a(s^{'})|s^{'}) \\
&- P(a(s^{'})|s^{'})| > \epsilon_1) P\left(\left|P(a_n(s^{'}) = a(s^{'})|s^{'}) - P(a(s^{'})|s^{'})\right| > \epsilon_1\right) P(s^{'}) \\
&+ \delta_2
\end{aligned}$$

As all the current theorems relate to fixed $s^{'}$ and by lemma 2

$$\begin{aligned}
\leq&\sum_{\substack{s^{'} \\ \in S^{'}}} \sum_{\substack{a(s^{'}) \\ \in A(s^{'})}} E\left(Q_*(a(s^{'}))\right) \left(P(a(s^{'})|s^{'}) + \delta_2\right) P(s^{'}) \\
&+ \sum_{\substack{s^{'} \\ \in S^{'}}} \sum_{\substack{a(s^{'}) \\ \in A(s^{'})}} B.\delta_1.P(a_n(s^{'}) = a(s^{'})|s^{'}) P(s^{'}) + \delta_2 \\
\leq&E\left(l_{1,*}(s^{'})\right) + \sum_{\substack{s^{'} \\ \in S^{'}}} \sum_{\substack{a(s^{'}) \\ \in A(s^{'})}} B.(\delta_1 + \epsilon_1) P(s^{'}) + \delta_2
\end{aligned}$$

Set $|A| = \max_{s' \in S'} |A(s^{'})|$

$$\leq E\left(L_{1,*}(s^{'})\right) + B.|A|.(\delta_1 + \epsilon_1) + \delta_2$$

Similarly

$$E\left(L_{1,n}(s^{'})\right) \geq E\left(L_{1,n}(s^{'})\right) - B.|A|.(\delta_1 + \epsilon_1) - \delta_2$$

$\therefore$ *Set* ϵ_1, δ_1 *and* δ_2 *s.t.* $\epsilon > B.|A|.(\delta_1 + \epsilon_1) + \delta_2$ □

LEMMA 23. *Let* n, j, k *be the n-th j-th and k-th occurances of action* a^1
Let $j > n$ *be s.t.* $s^{'}_j(a^1) = s^{'}(a^1)$
Let $k > j$ *be s.t.* $s^{'}_k(a^1) = t^{'}(a^1)$

$$cov(L^2_{1,j}(s^{'}), L^2_{1,k}(t^{'})) \to 0 \text{ as } n \to 0$$

Proof. This proof abbreviates for clarity, $a_j := a^2_j(s^{'}_j(a^1))$ and $b_k := b^2_k(t^{'}_k(a^1))$.

$$\begin{aligned}
&cov(L^2_{1,j}(s^{'}), L^2_{1,k}(t^{'})) \\
=&E(L^2_{1,j}(s^{'}).L^2_{1,k}(t^{'})) - E(L^2_{1,j}(s^{'}))E(L^2_{1,k}(t^{'})) \\
=&\sum_{a \in A^2(s^{'})} \sum_{b \in A^2(t^{'})} (E(Q_j(a).Q_k(b)))\, P(a_j = a, b_k = b) \\
&\qquad - E(Q_j(a))E(Q_k(b))P(a_j = a)P(b_k = b)
\end{aligned}$$

From lemma 19, given $\epsilon_1 > 0 \qquad \exists N_1 \in \mathbb{N}$ s.t. $\forall j, k > N_1 \quad \forall a \in A^2(s^{'})$
$\forall b \in A^2(t^{'}) \quad \forall s^{'}, t^{'} \in S^{'}$

$$|cov(Q_j(a), Q_k(b))| < \epsilon_1$$

From lemma 20, given $\epsilon_2, \delta_2 > 0 \quad \exists N_2 \in \mathbb{N}$ s.t. $\forall n > N_2 \quad \forall a \in A^2(s^{'}) \quad \forall s^{'} \in S^{'}$

$$P\left(|P(a_n = a) - P(a)| > \epsilon_2\right) < \delta_2$$

$\therefore$ Given some event X:

$$\begin{aligned}
\Rightarrow &P\left(|P(a_n = a) - P(a)| > \epsilon_2 | X\right) P(X) + \\
&P\left(|P(a_n = a) - P(a)| > \epsilon_2 | \neg X\right) P(\neg X) < \delta_2 \\
\Rightarrow &P\left(|P(a_n = a) - P(a)| > \epsilon_2 | X\right) P(X) < \delta_2
\end{aligned}$$

Consider

$$
\begin{aligned}
& P(a_j = a, b_k = b) \\
=& P(a_j = a | b_k = b) P(b_k = b) \\
=& P\left(a_j = a || P(a_j = a) - P(a)| > \epsilon_2, b_k = b\right) P(|P(a_j = a) - P(a)| > \epsilon_2) P(b_k = b) \\
& + P\left(a_j = a || P(a_j = a) - P(a)| \leq \epsilon_2, b_k = b\right) \\
& P(|P(a_j = a) - P(a)| \leq \epsilon_2) P(b_k = b) \\
\leq& \delta_2 + (P(a) + \epsilon_2) P(b_k = b) \\
\leq& \delta_2 + (P(a) + \epsilon_2)(P(b_k = b || P(b_k = b) - P(b)| \leq \epsilon_2) P(|P(b_k = b) - P(b)| \leq \epsilon_2) \\
& + P(b_k = b || P(b_k = b) - P(b)| > \epsilon_2) P(|P(b_k = b) - P(b)| > \epsilon_2)) \\
<& \delta_2 + (P(a) + \epsilon_2)(P(b) + \epsilon_2 + \delta_2) \\
=& P(a).P(b) + \epsilon_2(P(a) + P(b)) + \epsilon_2^2 + \delta_2.\epsilon_2 + P(a).\delta_2 + \delta_2 \\
=& P(a).P(b) + 2.\epsilon_2 + \epsilon_2^2 + \delta_2.\epsilon_2 + 2.\delta_2 + \delta_2
\end{aligned}
$$

$$
\begin{aligned}
& P(a_j = a).P(b_k = b) \\
>& (P(a) - \epsilon_2).(P(b) - \epsilon_2) \\
>& P(a).P(b) - 2.\epsilon_2
\end{aligned}
$$

From Lemma 2, $E(Q_j(a).Q_k(b)) < B^2$

$$
\begin{aligned}
& E(Q_j(a).Q_k(b)) P(a_j = a, b_k = b) - E(Q_j(a)) E(Q_k(b)) P(a_j = a) P(b_k = b) \\
<& E(Q_j(a).Q_k(b))(P(a).P(b) + 2.\epsilon_2 + \epsilon_2^2 + \delta_2.\epsilon_2 + 2.\delta_2 + \delta_2) \\
& - E(Q_j(a)) E(Q_k(b))(P(a).P(b) - 2.\epsilon_2) \\
\leq& \left(E(Q_j(a).Q_k(b)) - E(Q_j(a)) E(Q_k(b))\right) P(a).P(b) + B^2 \left(4.\epsilon_2 + \epsilon_2^2 + \delta_2.\epsilon_2 + 2.\delta_2\right) \\
\leq& \epsilon_1 P(a).P(b) + B^2 \left(4.\epsilon_2 + \epsilon_2^2 + \delta_2.\epsilon_2 + 2.\delta_2\right)
\end{aligned}
$$

$$
\begin{aligned}
& cov(L_{1,j}^2(s'), L_{1,k}^2(t')) \\
<& \sum_{a \in A^2(s')} \sum_{b \in A^2(t')} \epsilon_1 P(a).P(b) + B^2 \left(4.\epsilon_2 + \epsilon_2^2 + \delta_2.\epsilon_2 + 2.\delta_2\right)
\end{aligned}
$$

Set $|A| = \max\{|A^2(s')|, |A^2(t')|\}$

$$
\begin{aligned}
=& \epsilon_1 P(a).P(b) + B^2 |A|^2 \left(4.\epsilon_2 + \epsilon_2^2 + \delta_2.\epsilon_2 + 2.\delta_2\right)
\end{aligned}
$$

Choice ϵ_1, ϵ_2 and δ_2 s.t.

$$<\epsilon$$

Similiarly $cov(l^2_{1,j}(s^{'}), l^2_{1,k}(t^{'})) > -\epsilon$ □

LEMMA 24. *Let n, j, k be the n-th j-th and k-th occurances of action a^1*
Let $k > j > n$

$$cov(L^2_{1,j}(s^{'}), L^2_{1,k}(t^{'})) \rightarrow 0 \ as \ n \rightarrow 0$$

Proof. For clarity the following are abbreviated, $a_j(s^{'}) := a^2_j(s^{'}_j(a^1))$ and $b_k(t^{'}) := b^2_k(t^{'}_k(a^1))$. a^1 is ignored. By the independence of the customer model it is known that $P(s^{'}_j = s^{'}, t^{'}_k = t^{'}) = P(s^{'})P(t^{'})$

$$\begin{aligned}
&cov(L^2_{1,j}(s^{'}), L^2_{1,k}(t^{'})) \\
=&E(L^2_{1,j}(s^{'}).L^2_{1,k}(t^{'})) - E(L^2_{1,j}(s^{'}))E(L^2_{1,k}(t^{'})) \\
=&\sum_{s^{'} \in S^{'}} \sum_{t^{'} \in S^{'}} \sum_{a(s^{'}) \in A^2(s^{'})} \sum_{b(t^{'}) \in A^2(t^{'})} \\
&(E(Q_j(a).Q_k(b)))\, P(a_j(s^{'}) = a(s^{'}), b_k(t^{'}) = b(t^{'}))P(s^{'})P(t^{'}) \\
&- E(Q_j(a))E(Q_k(b))P(a_j(s^{'}) = a(s^{'}))P(b_k(t^{'}) = b(t^{'}))P(s^{'})P(t^{'})
\end{aligned}$$

From lemma 23 and that $|A^2(s^{'})| < \infty$, it is known that given sufficiently large n:

$$\begin{aligned}
&< \sum_{s^{'} \in S^{'})} \sum_{t^{'} \in S^{'}} \sum_{a(s^{'} \in A^2(s^{'})} \epsilon P(s^{'})P(t^{'}) \\
&<\epsilon
\end{aligned}$$

Similiarly $cov(L^2_{1,j}(s^{'}), L^2_{1,k}(t^{'})) > -\epsilon$ □

It has been shown that all the properties hold for $L^2_{1,n}$. Now it needs to be shown that they work for $L^1_{1,n}$. Property one has already been shown so it is just needed that the other two are shown as well.

COROLLARY 25. *Let n be the n-th occurance of action a^1*

$$E(L^1_{1,n}(s^{'})) \rightarrow E(L^1_{1,*}(s^{'}))$$

Proof. The notation is abbreviated where needed. Since
$P(s_n^{''}(a(s^{'})) = s^{''}(a(s^{'}))|a(s^{'})) = P(s^{''}(a(s^{'})))$, as from a fixed distribution.

$$E\left(L_{1,n}(s^{'})\right) = \sum_{\substack{s^{'} \\ \in S^{'}}} \sum_{\substack{a(s^{'}) \\ \in A^2(s^{'})}} \sum_{\substack{s^{''}(a(s^{'})) \\ \in S^{''}(a^2(s^{'}))}} \begin{matrix} E\left(F_n(s^{''}(a(s^{'})))\right) P(s^{''}(a(s^{'}))) \\ P(a_n(s^{'})=a(s^{'})|s^{'}) P(s^{'}) \end{matrix}$$

By lemma 6, it is known that $E(F_n(s_n^{''}(a(s^{'}))))$ converges to $E(F_*(s^{''}(a(s^{'}))))$ for each $s^{''}(a(s^{'})) \in S^{''}(a(s^{'}))$ and that there is a finite number of them (for fixed $a^{'}$. By similiar arguments to lemma 22:

$$\left|E(L_{1,n}^1(s^{'})) - E(L_{1,*}^1(s^{'}))\right| < \epsilon$$

□

COROLLARY 26. *Let n, j, k be the n-th j-th and k-th occurances of action a^1*
Let $j > n$ be s.t. $a_j^2(s_j^{'}(a^1)) = a^2$
Let $k > j$ be s.t. $b_k^2(t_k^{'}(a^1)) = b^2$

$$cov(F_j^i(s_j^{''}(a^2)), F_k^i(t_k^{''}(b^2))) \rightarrow 0 \text{ as } n \rightarrow 0$$

Proof. Since $P(s^{''}(a^2))$ and $P(s^{''}(b^2))$ are independent, the same arguments as in lemma 8 can be followed. □

LEMMA 27. *Let n, j, k be the n-th j-th and k-th occurances of action a^1*
Let $j > n$ be s.t. $s_j^{'}(a^1) = s^{'}$
Let $k > j$ be s.t. $t_k^{'}(a^1) = t^{'}$

$$cov(L_{1,j}^1(s^{'}), L_{1,k}^1(t^{'})) \rightarrow 0 \text{ as } n \rightarrow 0$$

Proof. The notation is abbreviated where possible.

$$
\begin{aligned}
& cov(L^1_{1,j}(s^{'}), L^1_{1,k}(t^{'})) \\
=& E(L^1_{1,j}(s^{'}).L^1_{1,k}(t^{'})) - E(L^1_{1,j}(s^{'}))E(L^1_{1,k}(t^{'})) \\
=& \sum_{a\in A^2(s^{'})} \sum_{b\in A^2(t^{'})} \sum_{s^{''}(a)\in S^{''}(a))} \sum_{t^{''}(b)\in S^{''}(b)} \\
& E(F_j(s^{''}(a)).F_k(t^{''}(b)))P(s^{''}(a))P(t^{''}(b))P(a_j = a, b_k = b) \\
& - E(F_j(s^{''}(a)))E(F_k(t^{''}(b)))P(s^{''}(a))P(t^{''}(b))P(a_j = a)P(b_k = b) \\
=& \sum_{a\in A^2(s^{'})} \sum_{b\in A^2(t^{'})} \sum_{s^{''}(a)\in S^{''}(a))} \sum_{t^{''}(b)\in S^{''}(b)} \\
& \Big(E(F_j(s^{''}(a)).F_k(t^{''}(b)))P(a_j = a, b_k = b) \\
& - E(F_j(s^{''}(a)))E(F_k(t^{''}(b)))P(a_j = a)P(b_k = b))P(s^{''}(a))P(t^{''}(b))\Big)
\end{aligned}
$$

Since $P(s^{''}(a))$ and $P(t^{''}(b))$ can be removed outside the brackets by using corollary 26 it is possible to follow the same argument as lemma 23. □

COROLLARY 28. *Let n, j, k be the n-th j-th and k-th occurances of action a^1 Let $k > j > n$*

$$cov(L^1_{1,j}(s^{'}_j(a^1)), L^1_{1,k}(t^{'}_k(a^1))) \to 0 \text{ as } n \to 0$$

Proof. Directly from lemma 27 and using the same arguments as lemma 24. □

6.7 Inductive Step

It has been shown that all the properties of $L^i_{0,n}$ all hold for $L^i_{1,n}$. This means that as the system repeats itself, it is possilbe to show this is case for all $L^i_{r,n}$ as well.

LEMMA 29. *If the propierties of $L^i_{0,n}$ hold, then they hold for all $L^i_{r,n}$ where $r < R$ for rounds $R < \infty$*

Proof. As action selection alternate between the players, $L^i_{2,n}$ is defined as a function of $s \in S$ with the same form as $L^i_{1,n}$ but with the definitions of each player swap i.e.

$$L^1_{2,n}(s) = Q^1_n(a^1_n(s))$$

This allows the proof, by induction, to follow the exactly the same arguments in corollary 21 to corollary 28 (but with the player references reversed) and show that $L^i_{2,n}$ also has the same properties as $L^i_{0,n}$. This process of proof can be repeated the finitely many times (less than R) to establish that the properties hold for all $L^i_{r,n}$ where $r < R$. □

The case holds for all $L^i_{r,n}$ where $r < R$, given that they hold for $L^i_{0,n}$. Now either $L^i_{0,n}$ refers to another round or a terminal round. If the round it terminal, the rewards observed will be zero.

LEMMA 30. *The function* $f(s) = 0 \quad \forall s \in S$ *has all the properties of* $L^i_{0,n}$

Proof. Trivial since $0 \in \{0, B\}$, $cov(0, 0) = 0$ and $E(0) = 0$. □

LEMMA 31. *Within a system with finite number of rounds, all Q-values converge (in probability) to their respective* Q^**-value.*

Proof. Through lemma 15 it has been shown that if $E(L^2_{0,n}(s^{''}(a(s^{'}))))$ converges, then each $Q^2_n(a^2(s^{'}))$ converges (which we call Q^*). Thus by applying the same arguments it can be concluded that $Q^1_n(a^1(s))$ converges and so does all preceeding Q-values. Notice that since all $L^i_{r,n}(s)$ have the same properties, it does not matter about the different sizes of tree's branches. □

LEMMA 32. Q^**-values represent the value obtained under the Nash Distribution.*

Proof. The $L^i_{o,n}$ values that related to a terminal node return and converge to a value of zero. Thus the preceeding $Q^2_n(a)$ values will converge to the expected values of rewards obtained from the independent customer model when action a is selected (by lemma 6). Thus the policy will converge to using the correct expected values for its Q-values, which means that the Boltzmann Action Selection is Nash Distribution policy for this case.

As player Two's policy convergence to Nash Distribution policy for this node, the $L^i_{1,n}$ will converge to the values that would be observed under a Nash Distribution policy. Hence, player One's Q-values will converge to the actual expected values observed under a Nash Distribution policy (by using the same arguments as lemma 6). This means that both sets of Q-values will converge to the correct values so $L^i_{2,n}$ converges correctly.

By finite induction and that each action in the system in visited i.o. (by extending lemma 5 to cover all actions), thus it is concluded that all Q^*-values are those obtained by under the Nash Distribution policy.

Again, its does not matter that different branches of the game tree have different lengths, as it works from the furtherest branches inwards. □

COROLLARY 33. *Both players policy converges (in probability) to the Nash Distribution policy.*

Proof. Directly from lemma 32 as the Q-values converge. □

THEOREM 34. *The learning model described in Chapter 4 is compatable with framework described here and therefore the players policies converges (in probability) to the Nash Distribution policy.*

Proof. The proof framework can simply be transferred to model framework by translating a few of the variables. The states in the proof framework related to the number of the seats left, round and current prices for the players. Hence the customer model will react accordingly to these inputs. The only special case to consider it reaction to the player One's action selection in the first round. This can be ignore by setting, for the first round, $|S(a)| = 1$ and $r^i(s(a)) = 0$, which relates to no seats being sold.

The learning parameters are directly similiar to the those found in the modelling framework (under the SARSA method), hence everything can be translated. This means the theorem follows from corollary 33. □

6.8 Discussion

Theorem 34 shows that the SARSA learning model will converge (in probability) to the Nash Distribution policy, no matter what customer models are used, number of seats available or the number of rounds (as long as they are finite). The only constraint on the proof is that everything should be finite, which is a fair assumption in any pricing model (as instantaneous changes cannot happen in the real world). This gives confidence to the results that have been obtained under this method (see chapter five for details) and that most anomalies will even out as the number of learning episodes is increased.

The generic nature of the proofs means that they can be applied to all of the SARSA learning models (i.e. the simple 355 game and customer behaviour models presented in the next chapter). However, this does not mean that the learnt policies observed from the learning runs will be the Nash Distribution policy as only a limited number of episodes were played (i.e. ten million).

A reason that obtaining a stronger convergence result (i.e. *with probability one*) would have been beneficial was because of the practical implications. In practice, a stronger convergence results usually means that convergence is reached faster in the runs (see Kushner and Yin, 2003). At present, the proof only offers a strong convergence of the infinite selection of each possible action.

It is not believed that a stronger convergence result can be obtained from the other parts of the proof due to the correlation that is observed between the actions which are selected in each of the different episodes. This correlation opens up bias within the system and therefore put an element of doubt on whether the Q-values will converge correctly. No formal proof that the system does not converge strongly has been offered here and this is an opportunity for further research.

Chapter 7

Variations on the Model

7.1 Introduction

The results so far have been based around the simple 233 game, which has lead to some interesting developments with both the Nash Equilibrium and the learnt policies. A lot of variations could be made to the simple 233 game and in this chapter two are focused on. The first variation is looking at different games where different Nash Equilibria are formed by allowing the players to vary the size of the aeroplanes they are using. The second variation is using more sophisticated customer models and the impact that his has on the learning process.

The simple 233 model and the learning model were developed from experience with earlier prototype models. The lessons learnt from these prototype models is also discussed within this chapter. The focus of these discussions is the impact any variation in the learning mechanism had on observed results, especially on its convergence.

Finally, within this chapter, there is a brief discussion on possible future research areas.

7.2 Metagame

Under the present simple 233 game, both players have an aircraft containing three seats. However, each airline is likely to have a fleet of aircraft available to them so they could vary the number of seats available. More importantly, each airline could

specify how many seats they have available before the game begins. By self-restricting the number of seats available and *telling their opponent* that they have done this will have an impact on the Nash Equilibrium pricing policy used by their opponent. This change in pricing policy by their opponent could be beneficial to a player; hence restricting their seats could be beneficial. Let's consider an example to explain this situation.

In this example player Two dropss their seat capacity from three to one. In the simple 233 game, P1 starting price is *five* when using the RANDOM Nash Equilibrium policy[1]. As mentioned in chapter four, this price was chosen to deter P2 from trying to attract more than one customer. However, now that P2 has only one seat, they can only attract one customer hence there is little reason for P1 to try and deter them from attracting two customers. The impact of this is that P1 starting price becomes *ten* and the overall returns obtained under the new Nash Equilibrium is $(20, 9.75)$. A breakdown of the new Nash Equilibrium policy can be found in appendix C in table C.1.

By P2 restricting the number of seat they have, they would actually gain an increase in return (because under the previous Nash Equilibrium policy, they would only observe a return of 8). This might seem surprising; however it is important to remember that under the previous RANDOM Nash Equilibrium policy they only sold one seat, so the other two seats were empty and these other two empty seats gave P1 something to worry about, hence P1 dropped their price.

By restricting the number of seats available (and telling their opponents they are doing so), different Nash Equilibrium policies can be discovered. Thus different number of seats available can mean different Nash Equilibrium policies and hence different returns observed. If the players continued to play the current Nash Equilibrium policy with the seat restriction in place, then *at least* one player would have an incentive to change policy due to the restricted seats (by definition of a Nash Equilibrium). Hence it would be expected that the players would end up playing the new Nash Equilibrium policies formed from the restrictions.

[1] See chapter four for details on the RANDOM Nash Equilibrium policy.

		P2		
		1	2	3
P1	1	(10, 10)	(10, 20)	(9.5, 20)
	2	(20, 10)	(10.5, 11)	(10, 10)
	3	**(20, 9.75)**	(13, 8)	(14, 8)

Table 7.1: Payoff matrix for meta-game

It is important to note that there is no benefit to either player in increasing the number of seats available to more than three because there are only three possible customers. However, as it has been shown, there can be benefit to a player in restricting the number of seats available. This could be considered a *meta-game*[2], where the players both restrict the number of seats available before any pricing is conducted. This meta-game can be represent as a normal-form game and is shown in figure 7.1.

The actions that are available to the players are the number of seats available on their aeroplane (i.e. one, two or three). Each of the return pairs (or payoff pairs) shown in figure 7.1 are derived from assuming that the players play the equivalent RANDOM Nash Equilibrium policy once pricing has started. A break-down of these policies can be found in table C.1, which is in appendix C.

This meta-game has its own Nash Equilibrium which is when P1 has three seats available and P2 has only one seat available (this was the situation given the example above). Thus even though P2 restricts the number of seats available, P1 gains the most benefit. This happens because of P1's control of the game due to them having first choice of price.

Though the use of a meta-game does produce some interesting results, in reality this behaviour is unlikely to be observed for a number of reasons. Firstly, there is no guarantee that P2 will restrict their aeroplane size to only one seat (unless they only own one-passenger aeroplanes). Secondly, it is unlikely that either player will know that *exactly* three customers will purchase seats on the aeroplanes. In the next

[2]By meta-game it is meant a game who's payoffs are derived from another game (thus the payoffs are from a solution of these sub-games). This is not the traditional use of meta-game (Thomas, 1984) and is just a sub-form of a stochastic game.

section the impact in changing the customer model to reflect this uncertainty is discussed. The use of a meta-game does give a deeper understanding of the underlying model.

7.3 Variation in Customer

The simple 233 game uses a very simplistic customer model, namely: a single customer comes along and chooses the lowest price seat (or randomly chooses when both airlines have the same current price). This customer model was chosen so that the Nash Equilibrium policies could be found (using dynamic programming), which were used for comparisons with the learnt policies (for assessing the successfulness of the learnt policies). By removing this need for comparison, more complex customer models can be considered.

In this section, three new aspects of the customer model are considered, namely: *Customer Choice, Customer Demand* and *Market Size*. Using the SARSA method (with tau = 0.02), learning runs were conducted with the different customer models. The average return values were generated and presented in the graphs below[3]. Changing the customer model within the learning model was simple due to its separate self-contained nature, however, the same is not true for the dynamic programming solver.

The reason for investigating the effect of more sophisticated customer models was to check whether Reinforcement Learning produces reasonable results when handling a complex game. This ability to *solve* complex games is a current issue within Revenue Management (see Boyd, 2007) and the one of the main reasons for conducting this research. As the games have not been *solved* in the traditional way, subjective judgement must be used on the policies that are derived.

Customer Choice

The first aspect of the customer model considered is the customer choice. In the simple customer model it is assumed that the customer always takes the seat with lowest

[3]The KS statistics could not be used as no Nash Distribution policies were found to compare the results to. This does not mean that the Nash Distribution policies do not exist, only that they could not easily be derived.

price. Though this would seem a reasonable assumption, this is not always the case and sometimes a customer will choose the higher price product. This was originally found in the psychology literature and was called the *A law of comparative judgement* (Thurstone, 1927a,b). A mathematical version of this law was called the *multinomial Logit model* (MNL) and was introduced by Luce (1959) and formalised by Manski (1977). In the case of model, the MNL assigns a probability to customer acceptance[4], which looks like:

$$P(\text{customer accepts P1's price}) = \frac{e^{\frac{-p^1}{\beta}}}{e^{\frac{-p^1}{\beta}} + e^{\frac{-p^2}{\beta}}}$$

Where p^i is the current price of player i and β is the *scale* parameter. The MNL looks surprising like the Boltzmann Action selection model used by the learning players. The scale parameter for the MNL works in a similar way to the temperature parameter for the Boltzmann Action selection. The larger the beta value, the more the customers behave in a random way.

The MNL is not the only customer choice model and there are others that can used; a review is found in McFadden (1980). Another popular model is the Probit model, however this is more difficult to handle. There are problems with the MNL (i.e. Independence of Irrelevant Alternatives (IIA) in Debreu (1960) and in the work of Oum (1979)). However, the MNL will suffice for the demonstration into learning with customer choice.

Runs were completed for different values of β (i.e. 0.2, 1, 10 and 10K) and the results are presented in figure 7.1. The results using the customer model seen in the simple games (which has been called the *normal* or greedy customer model) have also been included. The graphs are created in a similar way to those in chapter five (i.e. figure 5.10) and they show the average return obtained from playing the policy learnt by the players after a certain number of episodes.

As the graphs indicate there is very little difference between the returns observed of standard *greedy* customer (i.e. one that always takes the lowest price) and the returns observed when a low beta value is used. This happens because there is very little difference between the customer behaviours. Given a run of ten million episodes,

[4]MNL model uses the Gumbel distribution, see Gumbel (1958).

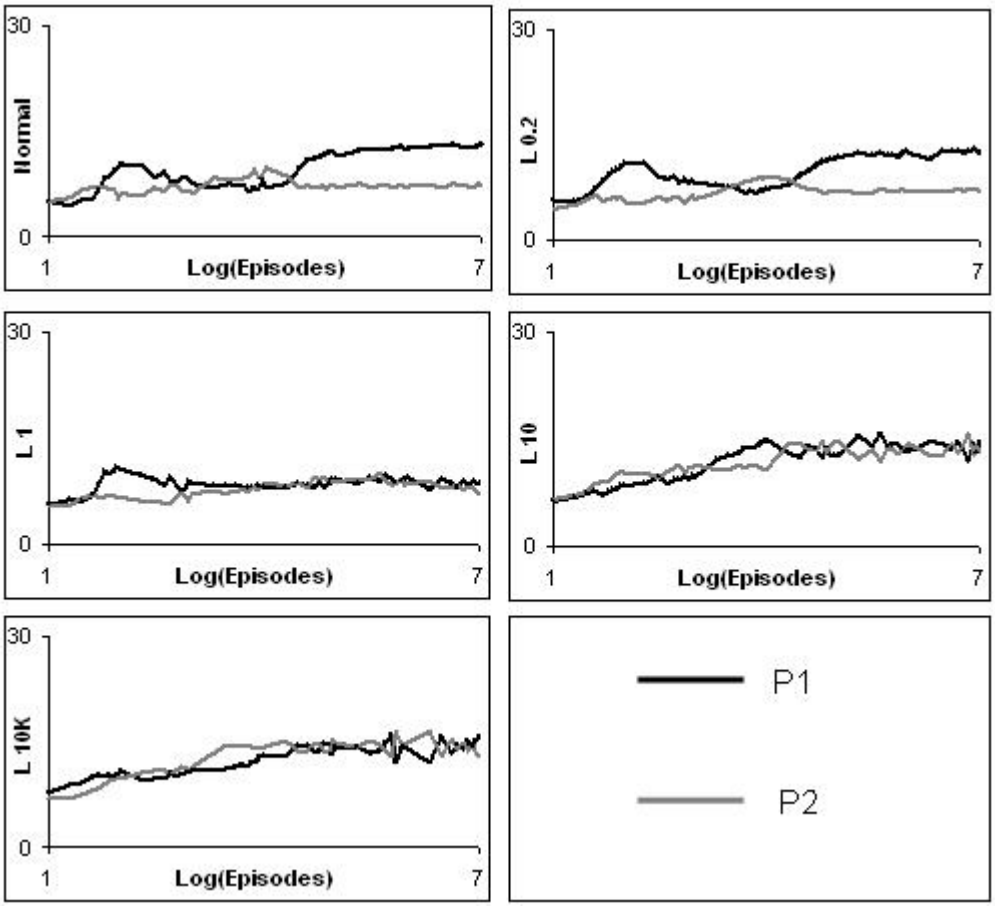

Figure 7.1: Graphs depicting the variation of average return values against episodes for the changes of the Beta value of the Logit customer choice model, using SARSA learning runs (tau = 0.02)

if a beta value of less than 0.02 was used then there would be expected to be no deviation from the greedy customer (as the chance of deviation is so low).

As the beta value increase so does the average returns from the learnt policies. This occurs because of the policies that the learning players achieve. As the beta value increases, the less impact the difference between the player's prices has on the customer's selection. In the extreme situation where the customers choose the airline randomly, it does not matter to each player what their opponent's price is, hence they just choose the price which will give them the highest return, which is *ten*. When both players continually use a price of *ten* at every stage in the game, their average return is 15. For games where there is a high beta value, the players learn to set their price to *ten* and hence obtain an average return of 15.

From this experimentation, it has been shown that for low and high values of beta, the policies have converged as expected. Also, there is a gradual change in the learnt policy as the beta value increases.

Customer Demand

In the present customer model, it is assumed that a customer only cares about the lowest price offered but does not worry about the price itself. Each individual customer will be willing to pay a different amount for airline seat, even if it is the lowest price offered. This willingness to pay forms the basis for the customer model variation. There are various different ways that this has been modelled (see Talluri and van Ryzin, 2004) and the simplest version is considered here, namely a *linear* customer demand model. This linear customer demand model means that the number of customers that are prepared to pay decreases linearly as the price increases. Therefore, given a customer at random, the chance that they are of the type that will accept an offered price will decrease linearly as the price increases. This can be represented as:

$$P(\textit{accepts price}) = A - B.Price$$

Where $A, B \in \mathbb{R}$ are some arbitrary constants. This formula says that the probability that the customer is of the type that would accept the price offered (the lowest price offered by the players) is a linear function of that price. This means that just because

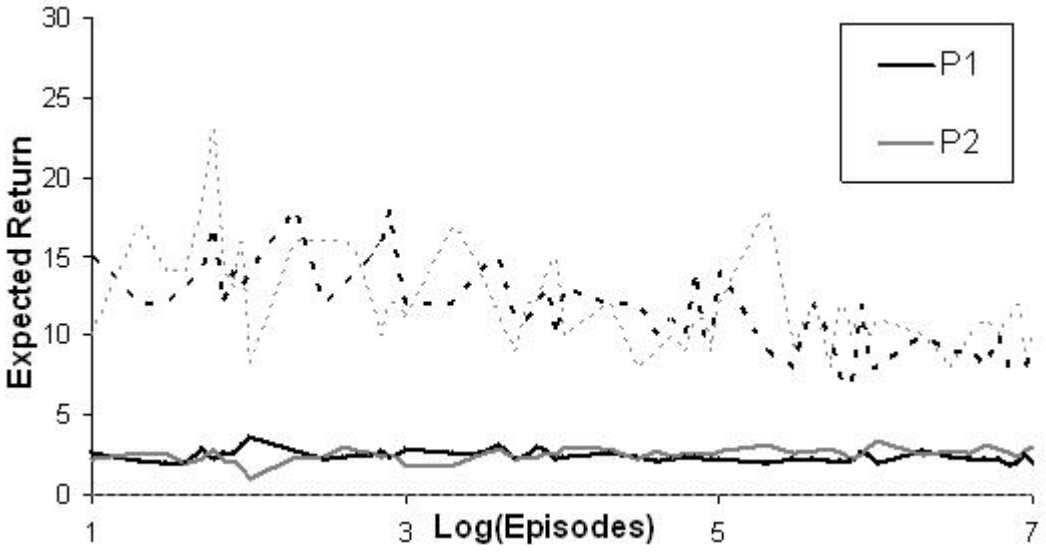

Figure 7.2: Graph depicting the average return values of a SARSA learning run (tau = 0.02) with customer demand

a player has the lowest current price does not mean that they will sell the seat. The values of A and B can be worked out by using the following assumptions:

- Everyone will accept a price of zero.
- No one is prepared to pay ten for a seat.

These assumptions make $A = 1$ and $B = 0.1$. By inserting this formula within the customer model for the simple 233 game, average return from a learnt policy were generated and shown in figure 7.2.

There are three noticeable features of the graph, namely: the low average returns, the high variation in the returns and the stable nature of average returns. The low average returns are now due lower expected returns that will be observed from the customer model. Under the new customer demand scheme, it is impossible to sell a seat at a price of *ten*, the highest expected value that a seat can be sold for is 2.5 (when a price of *five* is chosen, the chances that a customer accepts it is 0.5, hence a expected value of 2.5). This means that, even without an opponent, a player's highest expected return under any policy is 7.5 (as opposed to 30, which was observed under the simple 233 game). This implies that there is less variation in expected return

from any policy.

The second feature of the graph is that there is high fluctuations in the observed return, with the maximum return reaching levels around 10 to 15. This occurs because occasionally both players will play a high price and there happens to be customers that will accept this high price, however, the majority of the time this would not be the case.

These high fluctuations of observed return and the little difference between players' policies combine to make an environment which is very difficult for the players to learn in. This results in a slow rate of learning and hence why there seems to a stable average return (i.e. not much change in the average return for the players as the episodes increase). From investigating the actual policies learnt, there is not variation from just using a random policy (maybe with a slight basis towards the lower prices). This indicates that the players are having difficulty learning and hence not moving away from the initial random policy.

Though the Nash Distribution policy is not known for this game, the myopic one is and it generates a expected return of approximately (2.4, 4.6). There is no indication that the players pass through a *myopic* phase, hence confirming slow learning rate. From the convergence proof in chapter six, the learnt policies will eventually reach the Nash Distribution ones. However, these results indicate that there are situations where this progress is very slow.

Market Size

Using the a linear customer demand model resulted in slow learning by the players. This phenomenon occurred due to the low average returns observed and the high fluctuations in observed return. It is not necessary clear which of these affects slow learning, hence to investigate which of these affects learning the most, one factor is considered without the other. High fluctuations in observed return can be imposed by allowing a changing *market size*.

The original model assumes that there is constant market size at every customer model period (i.e. one customer comes along to choose a product). In reality the

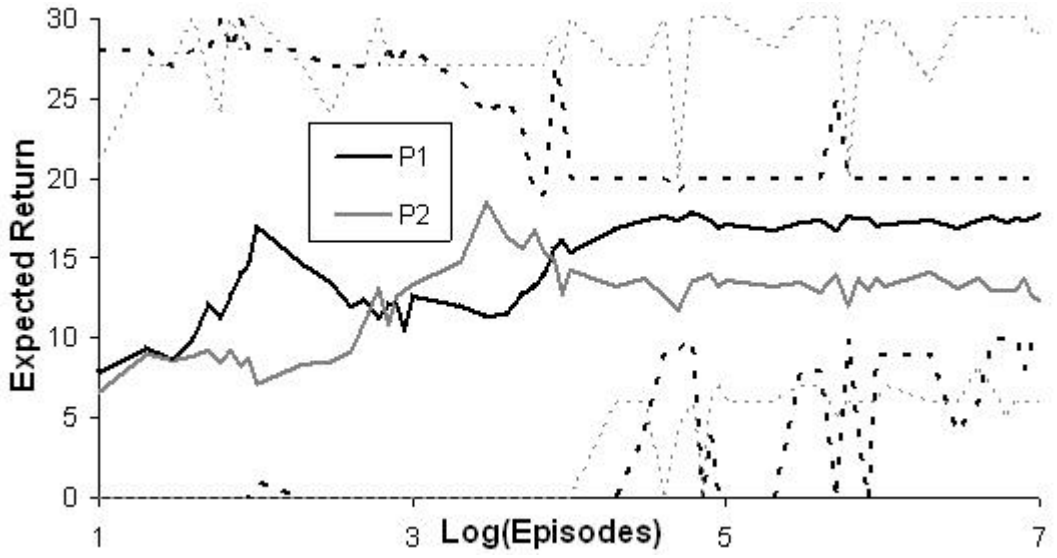

Figure 7.3: Graph depicting the average return values of a SARSA learning run (tau = 0.02) with stochastic market size

demand will fluctuant over the time period. It is common to see an increase in demand at the end of the selling period (this corresponds to last-minute business customers). To model this increase, it is assumed that a either one or two customers come along in each customer model period during the final round (i.e. round two in the 233 game).

A uniformly random distribution was used to determine whether one or two customer arrive in the final round's customer model periods (hence three to five customers can arrive in a single play of the simple 233 game). All other factors about the simple 233 game were kept the same, including that each player had only three seats available. The results from the learning run are given in figure 7.3.

The shape of the graph's results look very similar to the shape of the results from normal game given in figure 7.1. However, as expected, there are high fluctuations within the result. The policy learnt within the game with increased market size is identical to the RANDOM Nash Distribution policy. Thus it can be concluded that learning has occurred, even with these high fluctuations and that a sophisticated policy has been derived (though it is not clear if this is the Nash Distribution policy for this variation on the simple 233 game).

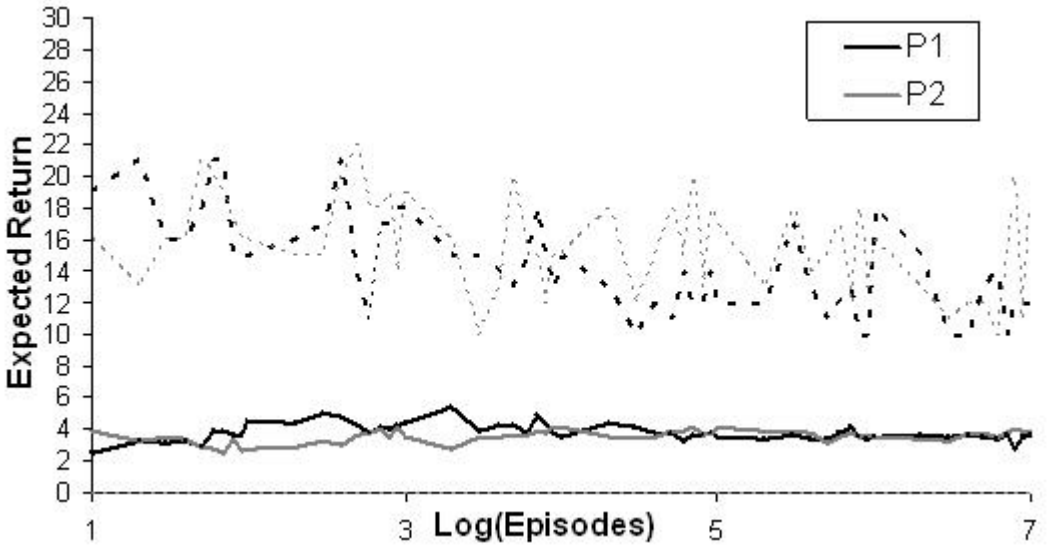

Figure 7.4: Graph depicting the average return values of a SARSA learning run (tau = 0.02) with stochastic market size, demand and customer choice (beta = 0.2)

As there are more customers, the expected return would increase under any policy. This has occurred and the expected return from the learnt policies after ten millions episodes is approximately (17.7, 12.3) (an increase on (14, 8)).

Combining Factors

From increasing the market size, it has been shown that learning can still occur when high fluctuations are present. This implies that observing only a small expected return from a customer is the cause of the customer demand games slow learning. To overcome this slow learning rate all three possible customer factors were combined into a sophisticated customer model (with a beta value of 0.2 for the customer choice aspect). These results are presented in figure 7.4.

As the graph indicates, slow learning was still present even though higher average returns were observed. Thus the customer demand problem dominated the learning. This might have been overcome by using a smaller temperature (i.e. tau < 0.02) to allow the players to distinguish between the returns of their different possible actions (a lower temperature would have amplified the small difference in return during action selection). However, as seen in chapter five, a low temperature means less exploration and hence also a lower rate of learning.

Other aspects of the customer behaviour could have been considered (i.e. overbooking and cancellation) but were not. As seen from these variations, some customer behaviours can be modelled and good policies learnt (i.e. customer choice and market size). However, other customer behaviours have a devastating impact on learning (i.e. varying customer demand).

7.4 Previous Experience

The model presented in chapter four, was not the first model to be used within this research. Various different models were considered and their results were used in generating the next version of the model. This section briefly discusses some of the important factors which were observed.

State Definition

There is a temptation when constructing a model to include lots of variables within a states definition. In previous versions of the model, each state was defined by a lot of variables (i.e. included previous prices used within a episode) and memory requirement for the state space increased to unmanageable proportions. Thus the appeal to Occum's Razor (Hamilton, 1852) in the methodology for a small number of variables within a state was beneficial in more than one way.

Another issue that was opposed the states was initial values given to each state's Q-values. Though optimistic starts were used for the Q-values (to encourage exploration), if the values were too high (and therefore unrealistic) then the learning methods took a lot longer to reach a sensible policy. Thus the initial Q-values used were not just some arbitrary large value.

Lambda - the Step-size Parameter

Various different versions of the step-size parameter where used to try and encourage learning. These were mainly based around trying to slow the rate at which step-size parameter decreased over the episodes. For example, one variation used was:

$$\lambda_e = \frac{1}{\left(\frac{e}{100000}\right) + 5}$$

Though the slower-decreasing step-size parameter did seem to given conformity within the learning over the different runs, it also meant that learnt policies took longer to

stabilize. As the results obtained from under the normal step-size parameter seemed reasonable, a decision was taken to not pursue development of the step-size parameter further (developing a slower-decreasing step-size parameter that was more stable seemed less likely).

Bootstrapping

As ten million episodes were being used per run, several ways to decrease this number were considered. One suggestion was to use a form of *bootstrapping*. The bootstrapping took the form of randomly selecting rewards observed from previous episodes and using them to update the appropriate Q-values. By repeating this process, it was hoped that the Nash Distribution policy would be reached at a faster rate.

However, the method had the effect of causing the policies to diverge in extreme directions (i.e. always playing a price of *ten*, etc). These divergent results were more extreme than any other results obtained. The reason this happened was because of the dynamic nature of a game, thus rewards obtained a million episodes previously would have no impact on the reward observed now (as the opponent's policy would have changed). These same divergent results were obtained when discounting of the previous rewards was also introduced.

The effect wanted from the bootstrapping might have been achieved by increasing the step-size parameter. However, from the discussions above, this was also not appropriate.

The final variation on game that was considered was the use of Bayesian updating within the Reinforcement Learning process. Though it would have interesting to see the effects from this approach, the only means of implementation seemed to involve increasing the state space and hence were impractical. Deployment of this idea was left to further research.

7.5 Future Research

The runs conducted within this research are but a small sample of the possible runs that could have been done. Due to time limitations, other lines of investigation had to be ignored. Outlined in this section are some of these possible lines of enquiry, which have been left for future research.

Static Learning

Another way that learning could occur is by playing the learning player against a static policy. This has been done in various different research papers, an example of this can be found in Takadama and Fujita (2005). From these studies the following tend to be observed:

- Learning is faster when only using one learning player
- The learnt policy converges to the best response to static policy

The use of single-player learning would give another mechanism to compare the different RL techniques with. However, the results from this type of learning were not presented with this thesis.

There are three crucial reasons why results from this type of learning have not been presented in this thesis. Firstly, the results do display the above listed phenomenon that has occurred in other research. Secondly, having only one player learning is no longer a multi-agent system. Single agent RL systems have already been well studied ((see Sutton and Barto, 1998) and it is not believed there is anything to add here. Finally, using a static policy (especially the Nash Equilibrium) goes against part of the purpose of the research. This research is investigating the use of RL in games that are not easily solved and any learning results from ones that use the *optimal* strategy in the learning mechanisms would seem pointless for the research.

There are expectations why this approach would be useful to a practitioner and should not be disregarded. The game involved may have a Nash Equilibrium that is easily found for only one of the players and the other strategy needs to found. It is hard to imagine a non-trivial situation where this might occur but this cannot be ruled out.

Another expectation is if the modeller wishes to find the best-response of a player to a static opponent. As the opponent cannot vary their policy, this goes against the foundations of Game Theory. It can be argued that this is not a game and just the well-studied single player case. However, the practitioner may wish to test a player's policy to see if it is a Nash Equilibrium policy by putting it against a learning player. This would be a reasonable use of this method; however it was not pursued in this research.

Decreasing the Temperature

Though the convergence proofs are dependent on the system having a constant temperature parameter, this does not mean that experimentation into a varying temperature cannot be conducted. If the temperature can be reduced as the number of episodes increases then, hopefully, the learnt policy would converge to the Nash Equilibrium policy instead of the Nash Distribution policy.

However, from initial experimentation this has been shown to be a bad thing, with the policy converging to some non-Nash policy. This could have been because the rate of decrease was too large and thus if rate of decrease was slower (and more episodes could be run), then maybe some interesting results could be observed. However, it would be difficult to say which rates the temperature should decrease at in different games and given the current sensitive nature of the temperature on the convergence results, difficulties would be expected to be encountered.

Different Learning Players

At present, all the results are from learning players which were played against similar learning players (i.e. both using SARSA with a temperature of 0.02). However, it has been suggested that different learning players can aid learning (see Leslie and Collins, 2003) thus it could be interesting to see the effect from allowing different learning to learn against each other (e.g. have a Q-learning learner play against a SARSA learner). In Takadama and Fujita (2005), it was suggested that the different methods could be used to validate the results. Maybe by playing them against each other would be a way of doing this.

Without going into depth, other possible areas for research include:

- Playing learnt policies against previous learnt policies to see if an improvement was made.
- Varying the game to involve three learning players.
- Allow the customers to also learn.

All these suggestion are worthy of further research which may lead to fruitful insights into the use of Reinforcement Learning within a practical gaming context.

Chapter 8

Summary, Conclusion and Recommendations

8.1 Summary

Throughout this thesis, there have been many interesting and surprising results. Before any conclusions are given about the work, a brief summary of the thesis is given here.

The research arose from the need to address some of the practical problems with the implementation of Game Theory as an Operational Research technique. The problem of solving the complex game used by OR practitioners was focussed on and Reinforcement Learning has been suggested as a possible approach to overcome this (Ravulapati et al., 2004). Though traditional techniques (i.e. Dynamic Programming) do exist for solving games, they can require an enormous amount of computational time to find the solution (*Curse of Dimensionality*) or they can be difficult to formalise for complex games (*Curse of Modelling*)[1]. Reinforcement Learning (RL) does not suffer from these faults.

To check whether Reinforcement Learning could be applied in a practical sense, a current problem within OR was needed. It has been highlighted by Boyd (2007)) that there is a difficulty in marrying up games and complex customers models within

[1]This is discussed in Gosavi (2003)

an airline pricing context. Thus an airline pricing game was chosen as the case study for the application of Reinforcement Learning.

The features of the airline pricing game were discussed in Chapters Three and Four. To ensure that Reinforcement Learning methods were working, the model had to be simple enough to be solved by traditional methods (so that the results could be compared). However, the customer model within the airline pricing model was constructed independently of the other parts so that it could be replaced by a more complex customer model.

The model was solved using the traditional methods and the Nash Equilibrium policies were found. The difference between the Nash Equilibrium polices were discussed. The Nash Equilibrium policies were not obvious and indicated a high level of sophistication by the players (i.e. first player to chooses a price, choose a low price to force the other player into choosing a higher price). As the game size was increased it became clear that the Nash Equilibrium policies followed a cyclic pattern, hence generalisations could be made about any sized game.

Once the Nash Equilibrium were found and understood, they were compared to the Nash Distribution policies for varying temperatures. The results indicated that the Nash Distribution policies were very similar to the Nash Equilibrium policies for low temperatures and like the random policy for high temperatures. Thus using the lowest temperature possible would be ideal. However, this was not necessarily possible due to rounding problems within the model.

The model was constructed using the C++ programming language and run on the University of Southampton's super-computer. Various issues were discussed which related to the use of a computer model, including its verification and validation. Validation was conducted using the *open-box* method suggested by Pidd (1996).

Before any results were presented, different measures were considered for comparing the learnt policies to static ones. By comparing the Nash Equilibrium policy to the Nash Distribution policy, it was indicated the Kolmogorov-Smirnov (KS) statistic would be appropriate for the comparison. This statistic was used for all comparisons within the results.

Three different RL techniques were considered, namely: SARSA, Q-learning and the Monte Carlo method. Monte Carlo method was out-performed by the other two methods, which were very similar (though the SARSA method produced slightly better results). All three techniques produced bad convergence results for low temperature parameters; this was due to the lack of exploration that a low temperature would imply. All three techniques produced good convergence results for high temperature; however, the policies learnt were too dissimilar to the Nash Equilibrium to be useful. A temperature of around 0.02 had results which were the best of both worlds (i.e. good convergence and similar to the Nash Equilibrium policy) and was used as a case-study example.

The case-study looked at how the learnt policy changed as more games (episodes) were played. Learning moved through four distinct phases. The first phase was close to a random policy and could be considered to be the method's *warm-up* period. In the second phase, the RL methods moved towards the Nash Distribution policy. In the third phase, the RL methods moved away from the Nash Distribution policy and moved towards the myopic policy. The final phase indicated a convergence toward the Nash Distribution policy. These same phases were seen within other learning games, including those of a larger size.

The method for comparing the policies became unfeasible for large games and further comparisons were abandoned. The time and memory requirement for the RL technique could still be calculated for the larger games. Both requirements increased linearly with a increase in game size for a fixed number of episodes. However, it was estimated that a larger number of episodes would be required for the larger games for convergence of the policy to be achieved.

To check that the RL techniques would converge in theory (if not in practice), a convergence proof was constructed for the SARSA method. This proof used a generic framework for the game so it could be applied to lots of different variations of the game and an inductive approach was used. The proof showed that the method converged *in probability*. Ideally a stronger level convergence would have been preferred; however, this was unlikely due to the dependency between actions selected in different episodes.

Two variations on the original case study game were considered. The first looked at a *meta-* game which allowed the players to select the size of their aircraft for the game. This meta-game had its own Nash Equilibrium which gave a deeper understanding of the underlying airline pricing game.

The second variation looked at the consequences of applying the SARSA learning technique to more complex customer behaviour. Though good results were found when customer choice and variable market sizes were introduced, the same was not true for when varying customer demand was considered. With varying customer demand added to the game, little learning was observed by either player. This lack of learning stemmed from the low expected prices at which a player sold in their seats. Thus it was suggested that this possible lack of learning should be watched out for in any future application of the RL techniques.

Finally, the thesis discussed how the airline pricing game has evolved from other games and recommendations for future research.

8.2 Conclusions

There are several conclusions that can be drawn from this research, some positive and some negative. Each conclusion is considered in turn and are in no particular order.

The use of Reinforcement Learning as a technique in games is both adaptable and easily implemented[2]. As with the airline pricing model, the RL technique can be dealt with by a separate model hence reducing the complexity of the modelling processes. However, the RL technique does require certain conditions on the underlying model (i.e. finite number of possible actions and state-space) though these can be dealt with using heuristic techniques.

An advantage of having a separate model for the RL technique was that the players learn to react to the customer model but do not need to consider an explicit representation of it. This means that the customer model can be designed to be as com-

[2]It is assumed that the practitioner implementing this method has some mathematical and programming knowledge from either a degree or post-graduate qualification in OR.

plex as necessary without having to change the method of Reinforcement Learning. As shown in the variations to the model chapter, some changes to the customer model can have an impact on the on the rate of learning and any modeller must be aware of these pitfalls.

The airline pricing game required that a lot of action exploration was present within the learning run to ensure that the state space was explored. When dealing with a dynamic environment (i.e. when there is more than one player), this exploration of the state space needs to occur repeatedly so that a players' learning takes into account the other players' policy changes. Any extra exploration will have an impact on the expected results from learning (i.e. convergence to a Nash Distribution and not to a Nash Equilibrium). A modeller will need to balance the trade-off these considerations when using a RL technique. This research did not produce any rules on how this should be done for different games[3].

This level of exploration would not be appropriate if the data comes from a *real-world* game as it would require the player to occasionally play non-greedy action, which they might be unwilling to do.

Not only does Reinforcement Learning need enough exploration to reach the desired results, it needs enough repeated plays of the game (episodes). Again, no guidelines are given in this thesis on the number of episodes required. The results showed that that there were several *phases* that the learning players had to pass through to reach these desired results and identifying these stages might be the key to determining the number of episodes required.

The comparison of the RL techniques suggests that the SARSA method should be used in any RL modelling though there is not much different between it and Q-learning. As convergence results were proved for the SARSA method within a generic sequential game framework, the modeller might consider using this framework because they will know that the desired results (i.e. Nash Distribution policies) will be reached eventually (in probability).

[3]As choosing what level of trade-off is needed between exploration and exploitation results is referred to as the *black-art* of modelling (Sutton and Barto, 1998).

From the study of the simple 233 game[4], complex results were observed which were not anticipated before the game was constructed. This level of complexity in such a simple game raises the questions of whether complex games should be used for analytical purposes because they are likely to generate even more complex results. Any RL solutions from an complex model will require a robust analysis before any generalisation can be validated. Advantages of the method are that games with complex customer models can be analysed and that the learning process can add insight into the analysis.

The airline pricing game does demonstrate the practical application of RL to solving a game. As discussed there are several advantages and limitations to this application, which must be considered before any application is made.

The physical limitations of applying the model form the greatest constraint on the applying the RL techniques. Though the RL method allows an easier way to set up a game for analysis (thus addressing the *curse of modelling*), the number of episodes required to solve the game can be excessive and beyond any reasonable run-time length (i.e. *curse of dimensionality*). This slow speed of learning is due to having a multi-player environment which is dynamically changing as the episodes increase.

8.3 Recommendations

The research from this thesis suggests that the RL techniques can produce interesting results, worthy of analysis. It would be recommend that RL was applied with some degree of caution and the time was spent investigating the means with which the models learns (i.e. which phases the learning passes through). A simple game can produce complex results and this complexity would need to be dealt with in any analysis.

If a single RL technique was required for analysis, the SARSA method (with a temperature of 0.02) would be recommended. If a similar framework is used for the game

[4]Though the game has been called simple because of its structure, there still is possible application. Anecdotal evidence indicates that over the last two years, the budget airlines (e.g. Flybe, Ryanair, Easyjet, etc.) are now increasingly selling single-leg flights though there is currently no evidence to support this.

as the case-study used within this thesis, then the modeller can be reassured of eventual convergence of any run of the learning game.

From this case study, Reinforcement Learning does seem a good alternative to dynamic programming for solving complex games. However, as a Nash Equilibrium result cannot be guaranteed, only insights about the game can be drawn from any analysis. It would be inappropriate to use the method described as a normative means of determining policy.

Several future developments of this research are suggested in Chapter Seven. One suggestion would be to develop a means to test whether a learnt policy had actually converged or had reached a local maximum[5]. Comparison of the case study presented in this thesis to another one could bring further insight into the Reinforcement Learning method.

This concludes the thesis and the research, which has produced some interesting and varied results.

[5]This might be achieved by allowing more episodes per run and analysing the results using the separation distance described in Chapter Five.

Appendix A

Nash Equilibrium and Nash Distribution

This appendix deals with several types of Nash Equilibrium policies for the simple airline pricing game, namely HIGH, LOW and RANDOM. As both HIGH and LOW Nash Equilibria were pure policies, the prices chosen and returns observed can be summaried and are given in tables A.1 and A.2 respectively. The tables show the prices chosen by the players, under the respective policies, for different game sizes. Where blanks are present indicates that the round does not exist in that size game (i.e. a game of size one has only one round). A similar table is also shown for the myopic policy.

Tables A.4, A.5 and A.6 indicate how the Nash Equilibrium policies were derrived by showing the best response policies for the different types of Nash Equilibria. Each column considers a different competitor's price and the rows show the players best response price and the resultant returns for different stages in the game. As the games are solved using backwards induction thus the policies of any sized game can be worked out from these tables.

Each of the policies begins to repeat a sequence of prices after a certain number of rounds. The table indicates this by showing that the returns obtained after from the best responses virtually repeat (i.e. are the same as those in a previous round minus a constant value) after a certain number of rounds.

Game	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
P1 Return	0.5	15	21	25	38	41	52	57	60	71	76
P2 Return	0.5	8	17	24	31	36	41	50	55	60	69
11-P1											10
11-P2											9
10-P1										5	5
10-P2										10	10
9-P1									3	9	9
9-P2									5	5	5
8-P1								10	10	10	10
8-P2								9	9	9	9
7-P1							5	5	5	5	5
7-P2							10	10	10	10	10
6-P1						3	9	9	9	9	9
6-P2						5	5	5	5	5	5
5-P1					8	8	8	8	8	8	8
5-P2					10	10	10	10	10	10	10
4-P1				4	9	9	9	9	9	9	9
4-P2				7	7	7	7	7	7	7	7
3-P1			10	10	10	10	10	10	10	10	10
3-P2			9	9	9	9	9	9	9	9	9
2-P1		6	6	6	6	6	6	6	6	6	6
2-P2		10	10	10	10	10	10	10	10	10	10
1-P1	1	9	9	9	9	9	9	9	9	9	9
1-P2	1	8	8	8	8	8	8	8	8	8	8

Table A.1: Deterministic action selection for various games under the *High* Nash Equilibrium policy

Game	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
P1 Return	0.5	14	19	23	35.5	43.5	47.5	60.5	68.5	72.5	85.5
P2 Return	0.5	8	15	27	31.5	40.5	48.5	56.5	65.5	73.3	81.5
11-P1											8
11-P2											10
10-P1										4	9
10-P2										8	8
9-P1									10	10	10
9-P2									9	9	9
8-P1								8	8	8	8
8-P2								10	10	10	10
7-P1							4	9	9	9	9
7-P2							8	8	8	8	8
6-P1						10	10	10	10	10	10
6-P2						9	9	9	9	9	9
5-P1					8	8	8	8	8	8	8
5-P2					9	9	9	9	9	9	9
4-P1				9	9	9	9	9	9	9	9
4-P2				8	8	8	8	8	8	8	8
3-P1			8	8	8	8	8	8	8	8	8
3-P2			7	7	7	7	7	7	7	7	7
2-P1		5	5	5	5	5	5	5	5	5	5
2-P2		10	10	10	10	10	10	10	10	10	10
1-P1	1	9	9	9	9	9	9	9	9	9	9
1-P2	1	8	8	8	8	8	8	8	8	8	8

Table A.2: Deterministic action selection for various games under the *Low* Nash Equilibrium policy

Game	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
P1 Return	0	8	14	18	20	21	22	23	24	25	26
P2 Return	9	16	21	24	25	26	27	28	29	30	31
11-P1											10
11-P2											9
10-P1										10	8
10-P2										9	7
9-P1									10	8	6
9-P2									9	7	5
8-P1								10	8	6	4
8-P2								9	7	5	3
7-P1							10	8	6	4	2
7-P2							9	7	5	3	1
6-P1						10	8	6	4	2	1
6-P2						9	7	5	3	1	1
5-P1					10	8	6	4	2	1	1
5-P2					9	7	5	3	1	1	1
4-P1				10	8	6	4	2	1	1	1
4-P2				9	7	5	3	1	1	1	1
3-P1			10	8	6	4	2	1	1	1	1
3-P2			9	7	5	3	1	1	1	1	1
2-P1		10	8	6	4	2	1	1	1	1	1
2-P2		9	7	5	3	1	1	1	1	1	1
1-P1	10	8	6	4	2	1	1	1	1	1	1
1-P2	9	7	5	3	1	1	1	1	1	1	1

Table A.3: Deterministic action selection for various games under the myopic policy

Round	Opponent's Price	0	1	2	3	4	5	6	7	8	9	10
8-P2	Best Response	10	1	2	2	3	4	5	6	7	**8**	9
	P1 Return	0	0.5	1	0	0	0	0	0	0	0	0
	P2 Return	0	0.5	1	2	3	4	5	6	7	8	9
8-P1	Best Response	10	2	2	2	3	4	5	6	7	8	9
	P1 Return	0	1	2	3	3	4	5	6	7	8	**9**
	P2 Return	0	1.5	2	1	2	3	4	5	6	7	8
7-P2	Best Response	10	10	10	10	10	10	**10**	6	7	8	9
	P1 Return	9	10	11	12	13	14	15	5	6	7	8
	P2 Return	8	8	8	8	8	8	8	10	12	14	16
7-P1	Best Response	6	6	6	6	6	4	5	6	6	**6**	6
	P1 Return	15	15	15	15	15	17	19	21	21	21	21
	P2 Return	8	9	10	11	12	8	8	8	8	8	8
6-P2	Best Response	4	4	4	4	3	4	4	4	4	8	**9**
	P1 Return	15	16	17	18	15	15	15	15	15	21	21
	P2 Return	12	12	12	12	14	16	16	16	16	16	17
6-P1	Best Response	10	10	10	10	10	10	10	**10**	7	8	9
	P1 Return	21	21	21	21	21	21	21	21	22	23	30
	P2 Return	17	18	19	20	21	22	23	24	16	16	17
5-P2	Best Response	7	7	7	7	7	4	5	6	7	**7**	7
	P1 Return	21	22	23	24	25	21	21	21	21	21	21
	P2 Return	24	24	24	24	24	25	27	29	31	31	31
5-P1	Best Response	4	4	4	4	3	4	4	4	4	8	**9**
	P1 Return	25	25	25	25	27	29	29	29	29	29	30
	P2 Return	24	25	26	27	24	24	24	24	24	31	31
4-P2	Best Response	10	10	10	10	10	10	10	10	**10**	8	9
	P1 Return	30	31	32	33	34	35	36	37	38	29	30
	P2 Return	31	31	31	31	31	31	31	31	31	32	40
	Best Response	8	8	8	8	8	**8**	5	6	7	8	8

Continued on next page

Round	Opponent's Price	0	1	2	3	4	5	6	7	8	9	10
4-P1	P1 Return	38	38	38	38	38	38	40	42	44	46	46
	P2 Return	31	32	33	34	35	36	31	31	31	31	31
	Best Response	5	5	5	5	4	4	5	5	5	**5**	5
3-P2	P1 Return	38	39	40	41	40	38	38	38	38	38	38
	P2 Return	36	36	36	36	37	39	41	41	41	41	41
	Best Response	3	3	3	2	3	4	4	6	7	8	**9**
3-P1	P1 Return	41	41	41	42	44	44	44	44	45	46	47
	P2 Return	36	37	38	36	36	37	37	41	41	41	41
	Best Response	10	10	10	10	10	**10**	5	6	7	8	9
2-P2	P1 Return	47	48	49	50	51	52	44	44	44	45	46
	P2 Return	41	41	41	41	41	41	42	43	48	49	50
	Best Response	5	5	5	5	3	4	5	5	5	**5**	5
2-P1	P1 Return	52	52	52	52	53	55	57	57	57	57	57
	P2 Return	41	42	43	44	41	41	41	41	41	41	41
	Best Response	3	3	3	2	3	3	5	6	7	8	9
1-P2	P1 Return	52	53	54	52	52	52	55	57	57	57	57
	P2 Return	44	44	44	45	46	46	46	47	48	49	50
	Best Response	10	10	10	10	10	10	6	6	7	8	**9**
1-P1	P1 Return	57	57	57	57	57	57	58	61	64	65	66
	P2 Return	50	51	52	53	54	55	49	46	47	48	49
	Best Response	5	5	5	5	4	4	5	5	5	**5**	5
0-P2	P1 Return	57	58	59	60	59	57	57	57	57	57	57
	P2 Return	55	55	55	55	56	58	60	60	60	60	60
	minus 19	38	39	40	41	40	38	38	38	38	38	38
	minus 19	36	36	36	36	37	39	41	41	41	41	41

Continued on next page

Round	Opponent's Price	0	1	2	3	4	5	6	7	8	9	10

Table A.4: Best response actions of opponent's current price and expected returns obtained while using the *High* Nash Equilibrium policy

Round	Opponent's Price	0	1	2	3	4	5	6	7	8	9	10
	Best Response	0	1	1	2	3	4	5	6	7	8	9
8-P2	P1 Return	0	0.5	0	0	0	0	0	0	0	0	0
	P2 Return	0	0.5	1	2	3	4	5	6	7	8	9
	Best Response	1	1	1	2	3	4	5	6	7	8	9
8-P1	P1 Return	0.5	1	1.5	2	3	4	5	6	7	8	9
	P2 Return	0.5	1	0.5	1	2	3	4	5	6	7	8
	Best Response	10	10	10	10	10	10	5	6	7	8	9
7-P2	P1 Return	9	10	11	12	13	14	4	5	6	7	8
	P2 Return	8	8	8	8	8	8	8	10	12	14	16
	Best Response	5	5	5	5	3	4	5	5	5	5	5
7-P1	P1 Return	14	14	14	14	15	17	19	19	19	19	19
	P2 Return	8	9	10	11	8	8	8	8	8	8	8
	Best Response	3	3	2	3	3	3	3	3	7	8	9
6-P2	P1 Return	14	15	15	15.5	14	14	14	14	19	19	19
	P2 Return	11	11	11	12.5	14	14	14	14	15	16	17
	Best Response	8	8	8	8	8	8	5	6	8	8	9
6-P1	P1 Return	19	19	19	19	19	19	19	20	23	27	28
	P2 Return	15	16	17	18	19	20	14	14	19	15	16
	Best Response	5	5	5	5	3	4	5	5	5	8	8
5-P2	P1 Return	19	20	21	22	19	19	19	19	19	23	23
	P2 Return	20	20	20	20	21	23	25	25	25	27	27
	Best Response	9	9	9	3	3	3	3	3	7	9	9

Continued on next page

Round	Opponent's Price	0	1	2	3	4	5	6	7	8	9	10
5-P1	P1 Return	23	23	23	23.5	25	25	25	25	26	27.5	32
	P2 Return	27	28	29	21.5	20	20	20	20	25	31.5	27
	Best Response	9	9	9	9	9	9	9	9	9	9	9
4-P2	P1 Return	27.5	28.5	29.5	30.5	31.5	32.5	33.5	34.5	35.5	32	27.5
	P2 Return	31.5	31.5	31.5	31.5	31.5	31.5	31.5	31.5	31.5	36	40.5
	Best Response	8	8	8	8	8	4	5	6	7	8	8
4-P1	P1 Return	35.5	35.5	35.5	35.5	35.5	35.5	37.5	39.5	41.5	43.5	43.5
	P2 Return	31.5	32.5	33.5	34.5	35.5	31.5	31.5	31.5	31.5	31.5	31.5
	Best Response	4	4	4	3	3	4	4	4	4	4	9
3-P2	P1 Return	35.5	36.5	37.5	37	35.5	35.5	35.5	35.5	35.5	35.5	43.5
	P2 Return	35.5	35.5	35.5	36	37.5	39.5	39.5	39.5	39.5	39.5	40.5
	Best Response	10	10	10	10	10	10	10	10	10	8	10
3-P1	P1 Return	43.5	43.5	43.5	43.5	43.5	43.5	43.5	43.5	43.5	43.5	48.5
	P2 Return	40.5	41.5	42.5	43.5	44.5	45.5	46.5	47.5	48.5	39.5	45.5
	Best Response	8	8	8	8	8	4	5	6	7	8	8
2-P2	P1 Return	43.5	44.5	45.5	46.5	47.5	43.5	43.5	43.5	43.5	43.5	43.5
	P2 Return	48.5	48.5	48.5	48.5	48.5	48.5	50.5	52.5	54.5	56.5	56.5
	Best Response	4	4	4	3	3	4	4	4	4	4	9
2-P1	P1 Return	47.5	47.5	47.5	48	49.5	51.5	51.5	51.5	51.5	51.5	52.5
	P2 Return	48.5	49.5	50.5	50	48.5	48.5	48.5	48.5	48.5	48.5	56.5
	Best Response	10	10	10	10	10	10	10	10	10	8	10
1-P2	P1 Return	52.5	53.5	54.5	55.5	56.5	57.5	58.5	59.5	60.5	51.5	57.5
	P2 Return	56.5	56.5	56.5	56.5	56.5	56.5	56.5	56.5	56.5	56.5	61.5
	Best Response	8	8	8	8	8	4	5	6	7	8	8
1-P1	P1 Return	60.5	60.5	60.5	60.5	60.5	60.5	62.5	64.5	66.5	68.5	68.5
	P2 Return	56.5	57.5	58.5	59.5	60.5	56.5	56.5	56.5	56.5	56.5	56.5
	Best Response	4	4	4	3	3	4	4	4	4	4	9
0-P2	P1 Return	60.5	61.5	62.5	62	60.5	60.5	60.5	60.5	60.5	60.5	68.5

Continued on next page

Round	Opponent's Price	0	1	2	3	4	5	6	7	8	9	10
	P2 Return	60.5	60.5	60.5	61.5	62.5	64.5	64.5	64.5	64.5	64.5	65.5
	minus 25	35.5	36.5	37.5	37	35.5	35.5	35.5	35.5	35.5	35.5	43.5
	minus 25	35.5	35.5	35.5	36	37.5	39.5	39.5	39.5	39.5	39.5	40.5

Table A.5: Best response actions of opponent's current price and expected returns obtained while using the *Low* Nash Equilibrium policy

Round	Opponent's Price	0	1	2	3	4	5	6	7	8	9	10
	Best Response	All	1, 2	2	2	3	4	5	6	7	8	9
5-P2	P1 Return	0	0.5	0.5	0	0	0	0	0	0	0	0
	P2 Return	0	0.5	1	2	3	4	5	6	7	8	9
	Best Response	1, 2	1	1, 2	2	3	4	5	6	7	8	9
5-P1	P1 Return	0.5	1	1.5	2.5	3	4	5	6	7	8	9
	P2 Return	0.75	1	1.25	1	2	3	4	5	6	7	8
	Best Response	10	10	10	10	10	10	5, 10	6	7	8	9
4-P2	P1 Return	9	10	11	12	13	14	9.5	5	6	7	8
	P2 Return	8	8	8	8	8	8	8	10	12	14	16
	Best Response	5	5	5	5	3, 4	4	5	5	5	5	5
4-P1	P1 Return	14	14	14	14	15	17	19	19	19	19	19
	P2 Return	8	9	10	11	9	8	8	8	8	8	8
	Best Response	3	3	2, 3	3	3	3	3	6	7	8	9
3-P2	P1 Return	14	15	15.5	15.5	14	14	14	16.5	19	19	19
	P2 Return	11	11	11	12.5	14	14	14	14	15	16	17
	Best Response	8, 9, 10	8, 9, 10	8, 9, 10	8, 9, 10	8, 9, 10	8, 9, 10	5, 8, 9, 10	6, 7	7	8	9
3-P1	P1 Return	19	19	19	19	19	19	19	20	23.5	27	28

Continued on next page

Round	Opponent's Price	0	1	2	3	4	5	6	7	8	9	10
	P2 Return	16	17	18	19	20	21	20	15.75	14	15	16
	Best Response	5	5	5	5	4	4	5	5, 6	5, 6	5, 6	5, 6
2-P2	P1 Return	19	20	21	22	20	19	19	19	19	19	19
	P2 Return	21	21	21	21	22	24	26	26	26	26	26
	Best Response	3	3	2, 3	3	3	3	3	3, 6	7	8	9
2-P1	P1 Return	22	22	22	23.5	25	25	25	25	26	27	28
	P2 Return	21	22	22.5	22.5	21	21	21	23.5	26	26	26
	Best Response	8, 9 10	8, 9 10	8, 9 10	8, 9 10	8, 9 10	8, 9 10	5, 8 9, 10	6, 7	7	8	9
1-P2	P1 Return	27	28	29	30	31	32	31	26.75	25	26	27
	P2 Return	26	26	26	26	26	26	26	27	30.5	34	35
	Best Response	5	5	5	5	3, 4	4	5	5, 6	5, 6	5, 6	5, 6
1-P1	P1 Return	32	32	32	32	33	35	37	37	37	37	37
	P2 Return	26	27	28	29	27	26	26	26	26	26	26
	minus 18	14	14	14	14	15	17	19	19	19	19	19
	minus 18	8	9	10	11	9	8	8	8	8	8	8

Table A.6: Best response actions of opponent's current price and expected returns obtained while using the *Random* Nash Equilibrium policy

A.1 Key Points in the Best Response Pairs of the Nash Distribution Policies

Table A.7 indicates the where there are changings in the best response pair of the Nash Distribution policy as the temperature parameter increases. The expected returns are also given.

τ	Best Response Pair	Return P1 (to 5 d.p)	Return P2 (to 5 d.p)
Nash Equilibrium	(5, 10)	14.0	8.0
0.028	(4, 10)	11.08287	6.40343
0.035	(3, 10)	8.85125	5.66836
0.037	(2, 10)	8.38225	5.47384
0.043	(10, 9)	7.41163	13.30719
0.125	(3, 2)	5.38968	3.98981
0.160	(4, 3)	5.41751	3.91389
0.378	(5, 4)	5.58706	4.16188
Random Policy	(5, 4)	5.68182	4.31818

Table A.7: Tau values were change in Best Response (for round one) policy has changed as the temperature parameter is increased.

Appendix B

Results from varying the Temperature Parameter

The tables in this appendices show the Kolmogorov-Smirnov (KS) statistics from the different learning runs of the simple 233 game and the simple 355 game after ten million episodes for varying temperature parameters (tau). Each learning run was repeated a hundred times for statistical signifiance. The results shown are the minimum, average (mean) and maximum KS statistics from these hundred runs. All values have been rounded to 7 deciminal places.

Simple 233 game

Tau	Minimum	Average	Maximum
0.002	0.0000106	0.4700079	1.0000000
0.004	0.0001274	0.3898494	0.9999994
0.006	0.0037129	0.4570226	0.9998762
0.008	0.0013231	0.5502184	0.9539115
0.01	0.0006876	0.3390407	0.8948072
0.012	0.0008700	0.2152772	0.8179054
0.013	0.0011200	0.1116025	0.7750743
0.014	0.0013535	0.0548438	0.7302506

Continued on next page

Tau	Minimum	Average	Maximum
0.0145	0.0013638	0.0379627	0.7072575
0.015	0.0014832	0.0165169	0.1035060
0.0155	0.0017837	0.0159924	0.1099583
0.016	0.0019955	0.0135390	0.1176160
0.0165	0.0020974	0.0104274	0.1186212
0.017	0.0022675	0.0062133	0.1160050
0.0175	0.0023099	0.0042900	0.0156234
0.018	0.0025727	0.0045189	0.0203473
0.0185	0.0028713	0.0048611	0.0163859
0.019	0.0027772	0.0052506	0.0163427
0.0195	0.0029968	0.0047585	0.0126216
0.02	0.0033478	0.0055296	0.0133445
0.0205	0.0033510	0.0062676	0.0192988
0.021	0.0032896	0.0065377	0.0161931
0.0215	0.0035344	0.0073449	0.0188463
0.022	0.0037370	0.0074749	0.0193613
0.023	0.0000000	0.0082693	0.0156728
0.024	0.0039261	0.0108766	0.0192699
0.025	0.0048766	0.0112693	0.0188941
0.026	0.0069744	0.0120048	0.0183980
0.027	0.0076898	0.0130899	0.0198458
0.028	0.0067322	0.0139082	0.0203815
0.029	0.0086110	0.0140291	0.0205508
0.03	0.0079762	0.0142306	0.0194158
0.032	0.0069876	0.0135061	0.0210652
0.034	0.0070333	0.0125219	0.0194022
0.035	0.0062165	0.0115879	0.0172786
0.037	0.0073812	0.0112110	0.0147088
0.04	0.0061943	0.0107879	0.0143825
0.042	0.0055678	0.0092198	0.0151394

Continued on next page

Tau	Minimum	Average	Maximum
0.043	0.0045509	0.0085372	0.0129754
0.044	0.0051207	0.0081591	0.0114728
0.045	0.0029928	0.0062725	0.0106905
0.047	0.0015551	0.0048837	0.0080675
0.05	0.0016096	0.0033577	0.0063223
0.055	0.0004710	0.0017069	0.0029221
0.06	0.0007160	0.0012978	0.0021555
0.065	0.0004251	0.0009758	0.0019211
0.07	0.0003947	0.0008207	0.0014053
0.08	0.0002429	0.0005729	0.0010515
0.09	0.0002294	0.0004559	0.0009244
0.1	0.0001538	0.0004158	0.0008231
0.12	0.0001128	0.0003794	0.0007678
0.13	0.0001021	0.0003458	0.0007420
0.14	0.0001251	0.0003213	0.0006082
0.15	0.0001062	0.0002908	0.0005448
0.16	0.0001261	0.0002662	0.0004969
0.19	0.0000824	0.0002270	0.0003676
0.2	0.0000000	0.0002026	0.0003530
0.21	0.0000892	0.0001968	0.0003345
0.24	0.0000452	0.0001656	0.0002791
0.25	0.0000369	0.0001593	0.0002862
0.3	0.0000429	0.0001203	0.0002225
0.31	0.0000406	0.0001207	0.0002513
0.32	0.0000440	0.0001116	0.0002036
0.33	0.0000427	0.0001136	0.0002059
0.34	0.0000389	0.0001093	0.0001876
0.35	0.0000287	0.0000997	0.0001706
0.36	0.0000378	0.0000983	0.0001727
0.37	0.0000287	0.0001042	0.0001707

Continued on next page

Tau	Minimum	Average	Maximum
0.38	0.0000324	0.0000960	0.0001862
0.39	0.0000299	0.0000946	0.0002038
0.4	0.0000335	0.0000887	0.0001767
0.41	0.0000365	0.0000849	0.0001641
0.42	0.0000382	0.0000857	0.0001730
0.43	0.0000250	0.0000834	0.0001655
0.44	0.0000000	0.0000801	0.0001511
0.45	0.0000302	0.0002045	0.0121250
0.5	0.0000176	0.0000688	0.0002555
0.55	0.0000169	0.0000609	0.0001510
0.6	0.0000217	0.0000562	0.0001016

Table B.1: Kolmogorov-Smirnov results for different tau in the simple 233 games with SARSA learning

Tau	Minimum	Average	Maximum
0.002	0.0000075	0.4300088	1.0000000
0.004	0.0001273	0.4597967	0.9999993
0.006	0.0004143	0.5852718	0.9998970
0.008	0.0008832	0.5973433	0.9983503
0.01	0.0009248	0.4524748	0.8948080
0.012	0.0020237	0.2052510	0.8179173
0.013	0.0029329	0.1104813	0.7750939
0.014	0.0043392	0.0408081	0.7299849
0.0145	0.0054982	0.0212503	0.0910790
0.015	0.0062039	0.0155159	0.0971568
0.0155	0.0075452	0.0193396	0.0909334
0.016	0.0084303	0.0203227	0.0947790
0.0165	0.0094114	0.0246202	0.1114473
0.017	0.0112067	0.0273295	0.0391948

Continued on next page

Tau	Minimum	Average	Maximum
0.0175	0.0196291	0.0325867	0.0426188
0.018	0.0240813	0.0366840	0.0552030
0.0185	0.0320754	0.0413762	0.0546058
0.019	0.0347094	0.0447691	0.0548587
0.0195	0.0400641	0.0483909	0.0572970
0.02	0.0405552	0.0523513	0.0664981
0.0205	0.0444595	0.0550269	0.0662207
0.021	0.0509922	0.0581635	0.0721183
0.0215	0.0499630	0.0611953	0.0762949
0.022	0.0529794	0.0629421	0.0730141
0.023	0.0520388	0.0643085	0.0721332
0.024	0.0572337	0.0639162	0.0728336
0.025	0.0575043	0.0650787	0.0721482
0.026	0.0582048	0.0644104	0.0697374
0.027	0.0545767	0.0611804	0.0675557
0.028	0.0511411	0.0564524	0.0624040
0.029	0.0437808	0.0502243	0.0573408
0.03	0.0357584	0.0430250	0.0504790
0.032	0.0241492	0.0297232	0.0356327
0.034	0.0105558	0.0174661	0.0224396
0.035	0.0085307	0.0118006	0.0163821
0.037	0.0039946	0.0055887	0.0083738
0.039	0.0041126	0.0057864	0.0098594
0.04	0.0050185	0.0071257	0.0116590
0.042	0.0077984	0.0107196	0.0149543
0.043	0.0093464	0.0127292	0.0155815
0.044	0.0076991	0.0123734	0.0164430
0.045	0.0116188	0.0170777	0.2137460
0.046	0.0129479	0.0160043	0.0198500
0.047	0.0128843	0.0169077	0.0204604

Continued on next page

Tau	Minimum	Average	Maximum
0.048	0.0145886	0.0172987	0.0214164
0.05	0.0149750	0.0173748	0.0196250
0.055	0.0162838	0.0173809	0.0186082
0.06	0.0160528	0.0172771	0.0186155
0.065	0.0171192	0.0179519	0.0187546
0.07	0.0184196	0.0189516	0.0196571
0.08	0.0193418	0.0223625	0.2580374
0.09	0.0189061	0.0193421	0.0197843
0.1	0.0174961	0.0179095	0.0182785
0.11	0.0158440	0.0161831	0.0166356
0.15	0.0103416	0.0106055	0.0108751
0.16	0.0094311	0.0096156	0.0098375
0.2	0.0065000	0.0078418	0.1226191
0.21	0.0060262	0.0061572	0.0063431

Table B.2: Kolmogorov-Smirnov results for different tau in the simple 233 games with Q-learning

Tau	Minimum	Average	Maximum
0.002	0.9999998	0.9999999	1.0000000
0.004	0.9999891	0.9999996	1.0000000
0.006	0.9998983	0.9999920	0.9999999
0.008	0.9997732	0.9999225	0.9999975
0.01	0.9916523	0.9989593	0.9999558
0.013	0.7750871	0.9899083	0.9995280
0.014	0.7302386	0.9861694	0.9923649
0.015	0.1022407	0.9261587	0.9885606
0.016	0.1094438	0.8812332	0.9833997
0.017	0.1292992	0.8254245	0.9561543
0.019	0.1209468	0.6477259	0.9207344

Continued on next page

Tau	Minimum	Average	Maximum
0.02	0.0275981	0.5467979	0.8974010
0.022	0.0111138	0.3007775	0.6718300
0.023	0.0041758	0.1896071	0.3070849
0.024	0.0047567	0.1079739	0.2650413
0.025	0.0050543	0.0556034	0.2259097
0.026	0.0047515	0.0181570	0.1908723
0.027	0.0046267	0.0105550	0.0255333
0.028	0.0036999	0.0095413	0.0187244
0.029	0.0038758	0.0087272	0.0174160
0.03	0.0030675	0.0087054	0.0144736
0.031	0.0035474	0.0085659	0.0160322
0.032	0.0028073	0.0083237	0.0171801
0.033	0.0026866	0.0081340	0.0141784
0.034	0.0034140	0.0082745	0.0124175
0.035	0.0032055	0.0076746	0.0128366
0.036	0.0030702	0.0072706	0.0123327
0.037	0.0030250	0.0071135	0.0121970
0.038	0.0029676	0.0068955	0.0115720
0.039	0.0030552	0.0065998	0.0117655
0.04	0.0023472	0.0063417	0.0101638
0.043	0.0015763	0.0056493	0.0087783
0.044	0.0028311	0.0062337	0.0113563
0.045	0.0020172	0.0054181	0.0085408
0.05	0.0006924	0.0034239	0.0072003
0.055	0.0005772	0.0017889	0.0042732
0.06	0.0002658	0.0011371	0.0019879
0.07	0.0002119	0.0006611	0.0013605
0.08	0.0001804	0.0004986	0.0011059
0.09	0.0001575	0.0004383	0.0009096
0.1	0.0001185	0.0003197	0.0006767

Continued on next page

Tau	Minimum	Average	Maximum
0.12	0.0000746	0.0002488	0.0004826
0.13	0.0000797	0.0002079	0.0004388
0.14	0.0000773	0.0001860	0.0004014
0.15	0.0000794	0.0001721	0.0003790
0.16	0.0000517	0.0001549	0.0003775
0.19	0.0000582	0.0001345	0.0003351
0.2	0.0000405	0.0001137	0.0002426
0.21	0.0000525	0.0001132	0.0002329
0.24	0.0000352	0.0000970	0.0002147
0.25	0.0000369	0.0000901	0.0001700
0.3	0.0000371	0.0000745	0.0001452
0.31	0.0000366	0.0000738	0.0001539
0.32	0.0000321	0.0000677	0.0001343
0.33	0.0000275	0.0000716	0.0001518
0.34	0.0000303	0.0000673	0.0001249
0.35	0.0000287	0.0000670	0.0001230
0.36	0.0000174	0.0000627	0.0001179

Table B.3: Kolmogorov-Smirnov results for different tau in the simple 233 games with Monte Carlo learning

Simple 355 game

Tau	Minimum	Average	Maximum
0.002	0.0918716	0.4046581	0.9998481
0.01	0.0826391	0.1137716	0.1315338
0.011	0.0856215	0.1026496	0.1175853
0.013	0.0540044	0.0685828	0.0815645
0.015	0.0495972	0.0759068	0.0990580
0.015	0.0466994	0.0755153	0.0973274
0.018	0.1003806	0.1304822	0.1549255

Continued on next page

Tau	Minimum	Average	Maximum
0.02	0.0331872	0.0496457	0.0619167
0.02	0.0373051	0.0498654	0.0639551
0.022	0.0161947	0.0271757	0.0368239
0.024	0.0214254	0.0400378	0.0564469
0.025	0.0228628	0.0389189	0.0577565
0.025	0.0219130	0.0394277	0.0537849
0.026	0.0181477	0.0331351	0.0561137
0.028	0.0107885	0.0178438	0.0260569
0.03	0.0076001	0.0117261	0.0146747
0.032	0.0000000	0.0093674	0.0126329
0.034	0.0041190	0.0069114	0.0096842
0.035	0.0038732	0.0057451	0.0077405
0.036	0.0030289	0.0048725	0.0068444
0.038	0.0021583	0.0034959	0.0052522
0.04	0.0017259	0.0025169	0.0037010
0.04	0.0017058	0.0025679	0.0039051
0.042	0.0012719	0.0020326	0.0029907
0.045	0.0011514	0.0019133	0.0030560
0.05	0.0015030	0.0023054	0.0032466
0.06	0.0022042	0.0028242	0.0035338
0.06	0.0019446	0.0028705	0.0035530
0.08	0.0019214	0.0026130	0.0032343
0.08	0.0023245	0.0026610	0.0031667
0.1	0.0018835	0.0021539	0.0024330
0.2	0.0008286	0.0009467	0.0011005
0.3	0.0004886	0.0005771	0.0006687

Table B.4: Kolmogorov-Smirnov results for different tau in the simple 355 games with SARSA learning

Tau	Minimum	Average	Maximum
0.002	0.1125300	0.3270694	0.9998534
0.005	0.1123415	0.2986428	0.9946987
0.01	0.0824743	0.1140538	0.1338569
0.012	0.0681539	0.0932908	0.1111077
0.014	0.0627601	0.0830604	0.1092131
0.015	0.0879690	0.1175796	0.1572614
0.015	0.0837689	0.1165599	0.1476798
0.016	0.1270963	0.1755815	0.2237726
0.018	0.1231862	0.1471604	0.1752660
0.02	0.0194608	0.0347315	0.0468672
0.022	0.0190216	0.0272861	0.0407847
0.024	0.0255921	0.0368517	0.0496786
0.025	0.0199910	0.0325392	0.0436481
0.026	0.0137831	0.0242755	0.0364664
0.028	0.0072705	0.0126386	0.0199310
0.03	0.0085678	0.0110669	0.0142741
0.032	0.0153477	0.0179510	0.0207567
0.034	0.0202340	0.0230194	0.0253940
0.035	0.0215808	0.0245131	0.0271927
0.04	0.0254473	0.0274035	0.0292574
0.04	0.0251360	0.0274997	0.0295932
0.045	0.0263078	0.0274343	0.0285694
0.05	0.0260064	0.0272204	0.0282253
0.055	0.0255793	0.0265390	0.0276319
0.06	0.0245816	0.0255098	0.0265237
0.08	0.0199774	0.0207109	0.0212213
0.08	0.0200727	0.0207274	0.0213812
0.1	0.0155210	0.0159871	0.0166551
0.15	0.0083884	0.0086819	0.0090994

Continued on next page

Tau	Minimum	Average	Maximum
0.2	0.0050412	0.0053452	0.0056473
0.25	0.0034078	0.0036284	0.0037888
0.3	0.0024432	0.0026323	0.0028555

Table B.5: Kolmogorov-Smirnov results for different tau in the simple 355 games with Q-learning

Tau	Minimum	Average	Maximum
0.002	0.9999258	0.9999916	0.9999997
0.005	0.6749559	0.9918924	0.9999232
0.01	0.1115308	0.5518629	0.9885875
0.015	0.0670920	0.2562376	0.7778504
0.016	0.1394679	0.2673187	0.6664866
0.018	0.1599594	0.2167413	0.2817599
0.02	0.0797090	0.1032177	0.1387778
0.022	0.0430677	0.0580183	0.0746491
0.024	0.0141312	0.0204504	0.0268592
0.025	0.0107267	0.0203087	0.0351446
0.026	0.0078221	0.0213962	0.0352891
0.028	0.0041207	0.0163202	0.0284937
0.03	0.0038875	0.0096733	0.0164471
0.032	0.0008721	0.0052879	0.0098085
0.034	0.0007716	0.0028771	0.0064557
0.035	0.0007196	0.0022841	0.0061238
0.036	0.0004570	0.0019592	0.0044858
0.038	0.0002861	0.0013522	0.0030138
0.04	0.0003667	0.0011745	0.0028099
0.042	0.0003026	0.0010383	0.0030219
0.044	0.0002885	0.0009375	0.0018434
0.045	0.0002917	0.0008166	0.0019050

Continued on next page

Tau	Minimum	Average	Maximum
0.05	0.0001991	0.0006598	0.0014415
0.055	0.0001832	0.0005779	0.0015119
0.06	0.0001328	0.0004809	0.0011047
0.065	0.0001120	0.0004135	0.0010453
0.07	0.0001041	0.0003824	0.0008525
0.075	0.0001139	0.0003405	0.0007849
0.08	0.0001169	0.0002964	0.0005759
0.085	0.0000709	0.0002766	0.0005449
0.09	0.0000751	0.0002465	0.0005834
0.1	0.0000665	0.0002145	0.0004245
0.12	0.0000619	0.0001599	0.0003658
0.14	0.0000461	0.0001287	0.0002934
0.15	0.0000477	0.0001330	0.0002967
0.16	0.0000328	0.0001089	0.0002706
0.2	0.0000296	0.0000835	0.0001942
0.25	0.0000185	0.0000696	0.0001338
0.3	0.0000198	0.0000526	0.0001330
0.35	0.0000134	0.0000447	0.0001101
0.4	0.0000156	0.0000401	0.0000977

Table B.6: Kolmogorov-Smirnov results for different tau in the simple 355 games with Monte Carlo learning

Physical limitations

The following table shows the memory and run-time requirements of the simple games of increasing rounds. The experimental runs were repeated for several different file-sizes. The second column show the minimum time (hours) required to run the game for ten million episodes under the different file-sizes. The third column shows the file-size required to achieve this minimum time, its units are mega-bytes (MB). The data only consider the time required to run the episodes and the time required to analysis the results has not been included. In all the results cases, each player only used

one memory file (the experiments did included cases were multiple files were allowed, though they were outpreformed by the single file case).

Run	**Hours**	**MB**	**Run**	**Hours**	**MB**
2	0.221	0.016	11	2.891	0.688
3	0.397	0.172	12	3.237	0.86
4	0.708	0.172	13	3.788	0.86
5	0.959	0.172	14	4.260	1.032
6	1.200	0.172	15	4.610	1.204
7	1.551	0.344	16	5.649	1.204
8	1.928	0.516	17	5.933	1.204
9	2.183	0.516	18	6.140	1.375
10	2.450	0.688	19	6.996	1.547

Table B.7: Physical limitation of simple games

Appendix C

Meta-game

The table in this appendix shows the Nash Equilibrium policies for the different versions of the simple game with two rounds. The different versions are based around the number of seats that are available to the players (i.e. 232 stands for the game with two rounds and where P1 has three seats and where P2 has only two).

Each line in the table indicates the game under consideration, what prices where chosen by the players, the customer's reaction (i.e. which airline they chose to buy a seat with) and the overall returns obtained. Sometimes the customer will be indifferent to either player (because they have the same price); when this occurs both possibilities are considered and different outcomes are shown on separate lines in the table.

	Round One			**Round Two**				**Reward**	
Game	**P1**	**P2**	**C**	**P1**	**C**	**P2**	**C**	**R1**	**R2**
211	10	10	P1	-	P2	-	-	10	10
			P2	10	P1	-	-	10	10
212	10	10	P1	-	P2	10	P2	10	20
			P2	10	P1	10	P2	10	20
					P2	-	P1	10	20
221	10	10	P1	10	P1	10	P2	20	10
					P2	-	P1	20	10

Continued on next page

	Round One			**Round Two**				**Reward**	
Game	**P1**	**P2**	**C**	**P1**	**C**	**P2**	**C**	**R1**	**R2**
			P2	10	P1	-	P1	20	10
222	6	6	P1	5	P1	10	P2	11	10
			P2	10	P2	-	P1	10	12
213	10	10	P1	-	P2	10	P2	10	20
			P2	9	P1	10	P2	9	20
231	10	10	P1	10	P1	9	P2	20	9
					P2	-	P1	20	10
			P2	10	P1	-	P1	20	10
223	5	6	P1	5	P1	10	P2	10	10
232	4	10	P1	9	P1	8	P2	13	8
233	5	10	P1	9	P1	8	P2	14	8

Table C.1: Nash Equilibrium play for meta-games

Appendix D

Convergence of the Variation of the Nash Distribution

The Nash Distribution policy has been shown to converge to a unique Nash Equilibrium policy as the temperature parameter is decreased to zero (see Fudenberg and Levine, 1998). However, as mentioned in Section 2.2, a variation on the Nash Distribution (VND) was considered within this research due to ease of implementation. There is a need to prove that this variation also converges to a unique Nash Equilibrium policy as the temperature parameter is decreased. This appendix contains that proof.

The variation of the Nash Distribution involves using Boltzmann Action selection at each stage of the game to select an action (as opposed to only randomizing once at the beginning of the game). Thus an inductive approach can be used to prove convergence to a sub-game perfect Nash Equilibrium policy. The proof given in Chapter Seven, which shows convergence of the SARSA method to the variation of the Nash Distribution policy, also follows an inductive approach and therefore, the terminology of that proof can be employed here.

The Nash Equilibrium policy that the variation of the Nash Distribution is compared to is the one that randomizes uniformly over actions which produce the maximum return. As a sequential game is considered, this Nash Equilibrium policy is known to

exist (see Fudenberg and Tirole, 1991, for details on sub-game perfect Nash Equilibrium). Thus given a state s, the probability that a player will select an action a from action set A under this Nash Equilibrium policy is given by:

$$P_*(a) = \begin{cases} \frac{1}{|A_{max}|} & \text{if } a \in argmax_{a \in A} Q_*(a) := A_{max} \\ 0 & \text{o/w} \end{cases} \tag{D.1}$$

Where $Q_*(s)(a)$ is defined a expected return obtained from selecting that action (under the Nash Equilibrium policy). If s is a pre-terminal state, then $Q_*(s)(a)$ is independent of any policy. This means that the the expected return from selecting action a under the VND policy $Q(s)(a)$ equals $Q_*(s)(a)$ at a pre-terminal states. Thus the probability of selecting a action a at a pre-terminal state using the VND policy is given by:

$$P(a) = \frac{e^{Q_*(s)(a)/\tau}}{\sum_{b \in A} e^{Q_*(s)(b)/\tau}} \tag{D.2}$$

Using this knowledge at any pre-terminal state, it is possible to show that probabilities of selecting action a converge to those of the Nash Equilibrium policy (and hence the policies are the same for a pre-terminal state), when the temperature τ is decreased to zero. Using backward induction, it is shown below that the VND policy converges to the Nash Equilibrium policy for all states.

It is recommended that Chapter Six is read first before continuing with reading of these proofs as it gives the paradigm framework from which they were constructed. As with Chapter Six, it is assumed that there are only a finite number of states, actions and rewards within the game.

The convergence proof takes the following steps, each assuming that temperature is decreasing to zero:

1. Prove that probability of selecting an action converges to the same probability used by a Nash Equilibrium, for pre-terminal states.

2. Prove that value of a pre-terminal state converges to the same value as if under a Nash Equilibrium, for both players.

3. Prove that Q-value of any action converge to the same value as if under a Nash Equilbrium, given that all follow-on states converge to the same value as if under a Nash Equilibrium.

4. Prove that Probability of selecting an action has the same values as if under a Nash Equilibrium, given that the Q-value has the condition shown in the last item.

5. Conclude the above applies to the whole game using an inductive step.

D.1 Terminal States

In this section it is shown that the probabilites generated by the Boltzmann Action Selection method (used within VND) converge to a Nash Equilibriums probabilities, as the temperature drops to zero.

LEMMA 35. *Given pre-terminal state s s.t. $|A| < \infty$ and $Q(a) < \infty \forall a \in A$*

$$P(a) = \frac{e^{Q(a)/\tau}}{\sum_{b \in A} e^{Q(b)/\tau}} \rightarrow P_*(a) \text{ as } \tau \rightarrow 0$$

Proof. As a pre-terminal state is considered $Q(a) = Q_*(a)$. If $|A| := n = 1$ then the result is trival. Elements of A can be arranged so that $a_1, ..., a_m, b_{m+1}, ..., b_n$ s.t.

$$Q(a_i) = Q(a_j) \quad \forall i, j \in 1, ..., m$$
$$Q(b_i) < Q(a_1) \quad \forall i \in m+1, ..., n$$

$$\therefore \exists \delta_i > 0 \text{ s.t. } Q(b_i) = Q(a_1) - \delta_i \quad \forall i \in m+1, ..., n$$

Given $i \in 1, ..., m$

$$
\begin{aligned}
P(a_i) &= \frac{e^{Q_*(a_1)/\tau}}{\sum_{b\in A} e^{Q_*(b)/\tau}} \\
&= \frac{e^{Q_*(a_1)/\tau}}{\sum_{i=1}^{m} e^{Q_*(a_1)/\tau} + \sum_{i=m+1}^{n} e^{Q_*(s)(b_i)/\tau}} \\
&= \frac{e^{Q_*(a_1)/\tau}}{m.e^{Q_*(a_1)/\tau} + \sum_{i=m+1}^{n} e^{Q_*(b_i)/\tau}} \\
&= \frac{e^{Q_*(a_1)/\tau}}{m.e^{Q_*(a_1)/\tau} + \sum_{i=m+1}^{n} e^{(Q(a_1)-\delta_i)/\tau}} \\
&= \frac{e^{Q_*(a_1)/\tau}}{m.e^{Q_*(a_1)/\tau} + e^{Q(a_1)/\tau}.\sum_{i=m+1}^{n} e^{-\delta_i/\tau}} \\
&= \frac{1}{m}.\frac{m}{m + \sum_{i=m+1}^{n} e^{-\delta_i/\tau}}
\end{aligned}
$$

Given any $\epsilon_1 > 0 \quad \exists t > 0$ s.t. $\forall \tau < t \quad \forall i \in m+1, ..., n$

$$e^{-\delta_i/\tau} < \epsilon$$

Therefore, choose τ s.t. $\forall i \in m+1, ..., n$

$$e^{-\delta_i/\tau} < \frac{\epsilon_1}{n-m}$$

$$
\begin{aligned}
P(a_i) &> \frac{1}{m}.\frac{m}{m + \sum_{i=m+1}^{n} \frac{\epsilon_1}{n-m}} \\
&= \frac{1}{m}.\frac{m}{m+\epsilon_1} \\
&= \frac{1}{m} - \frac{1}{m}\frac{\epsilon_1}{m+\epsilon_1}
\end{aligned}
$$

Given any $\epsilon > 0$, choose ϵ_1 s.t. $\epsilon_1 < m$ and $\epsilon > \frac{\epsilon_1}{m^2} > \frac{\epsilon_1}{m^2+m\epsilon_1}$

$$P(a_i) > \frac{1}{m} - \epsilon$$

Similarly $\forall i \in m+1, ..., n$

$$P(b_i) < \epsilon$$

Thus $P(a)$ converges to $P_*(a)$, which is defined above. □

D.2 State Values

Now the expected value for a state L is considered, for the active player, when both the Q-values and probabilities of action selection converge to Nash Equilibrium values (with a decrease of τ to zero).

LEMMA 36. *Given state s, whose active player is i, s.t.* $|A^i| < \infty$ *s.t.* $\forall a \in A^i$, $Q(a) < \infty$ *and*

$$P(a) \to P_*(a) \text{ as } \tau \to 0$$

$$Q(a) \to Q_*(a) \text{ as } \tau \to 0$$

Then

$$L^i(s) \to L^i_*(s) \text{ as } \tau \to 0$$

Proof.

$$L^i(s) := \sum_{a \in A^i} Q(a)P(a)$$

As both $Q(a)$ and $P(a)$ converge to $Q_*(a)$ and $P_*(a)$ respectively, $\exists \delta_1, \delta_2 > 0$ s.t. $\forall a \in A$

$$Q(a) > Q_*(a) - \delta_1$$

$$P(a) > P_*(a) - \delta_2$$

$$\begin{aligned} L^i(s) &> \sum_{a \in A^i} (Q_*(a) - \delta_1)(P_*(a) - \delta_2) \\ &= \sum_{a \in A^i} Q_*(a) \left(P_*(a) - \delta_1 P_*(a) - \delta_2 Q_*(a) + \delta_1 \delta_2\right) \\ &= \sum_{a \in A^i} Q_*(a) \left(P_*(a) - \delta_1 P_*(a) - \delta_2 Q_*(a)\right) \end{aligned}$$

Since $Q_*(a) < \infty$, given $\epsilon > 0 \quad \exists \delta_1$ and $\delta_2 > 0$ s.t.

$$\epsilon > \sum_{a \in A^i} (\delta_1 P_*(a) + \delta_2 Q_*(a))$$

$$\begin{aligned} \therefore L^i(s) &> \sum_{a \in A^i} \delta_1 P_*(a) + \epsilon \\ &= L^i_*(s) + \epsilon \end{aligned}$$

Similiarly, $L^i(s) < L^i_*(s) + \epsilon$ □

To include the pre-terminal state case, the following corollary is included.

COROLLARY 37. *If $Q(a) = Q_*(a) \quad \forall a \in A$ then Lemma 36 holds.*

Proof. As $Q(a) \to Q_*(a)$ as $\tau \to 0 \quad \forall a \in A$ □

The value of a state for the non-active player (i.e. the player that does not get to choose an action) needs to be seen to converge as well.

COROLLARY 38. *For pre-terminal state s and non-active player i*

$$L^i(s) \to L^i_*(s) \text{ as } \tau \to 0$$

Proof. This follows from lemma 36 and corollary 37 by using the observed return for the player i instead of Q-values. As in Chapter Six, this is defined as $L^i_0(s')$, where s' is the terminal state in this case. □

The previous proofs show that expected value observed at a preterminal state converges to the same value as under a Nash Equilibrium policy, for both players. The next stage is to show that observed Q-values from the previous state converge correctly as well.

D.3 Non-Terminal States

Given an state s and action selected a, the states that can follow are $s'(a) \in S'(a)$, each with their own value $L^i(s'(a))$ for player $i \in 1, 2$. The expected return to each player, after an action is selected, is represented by $F^i(a)$. For the player that selected the action, this is just the Q-value of the action.

LEMMA 39. *Given a and all possible follow-on states $s' \in S'$, and associated reward $r(s'(a))$ if*

$$L^i(s') \to L^i_*(s') \text{ as } \tau \to 0 \quad \forall s' \in S'$$

Then

$$F^i(a) \to F^i_*(a) \text{ as } \tau \to 0$$

Proof.

$$F^i(a) = \sum_{s' \in S'} p(s'|a) \left(E(r(s')) + L^i(s') \right)$$

$p(s'|a)$ and $E(r(s'))$ are independent of policy as defined in Chapter Six. Since $L^i(s')$ converges and $|S'| < \infty$, given any $\epsilon > 0 \quad \exists T > 0$ s.t. $\forall \tau < T \quad |L^i(s') - L^i_*(s')| < \epsilon$

$$F^i(a) > \sum_{s' \in S'} p(s'|a) \left(E(r(s')) + L^i_*(s') - \epsilon \right)$$

$$F^i(a) > F^i_*(a) - \epsilon$$

Similiarly, $F^i(a) < F^i_*(a) + \epsilon$ □

Now the probabilities from a state with convergent Q-values can be shown to converge correctly.

LEMMA 40. *Given state s s.t. $|A| < \infty$ and $Q(a) < \infty \forall a \in A$ and*

$$Q(a) \to Q_*(a) \text{ as } \tau \to 0 \quad \forall a \in A$$

then

$$P(a) = \frac{e^{Q(a)/\tau}}{\sum_{b \in A} e^{Q(b)/\tau}} \to P_*(a) \text{ as } \tau \to 0$$

Proof. This proof is similiar to lemma 35, consider the Q-values given by the Nash Equilibrium policy. If $|A| := n = 1$ then the result is trival. Elements of A can be arranged so that $a_1, ..., a_m, b_{m+1}, ..., b_n$ s.t.

$$Q^*(a_i) = Q^*(a_j) \quad \forall i, j \in 1, ..., m$$

$$Q^*(b_i) < Q^*(a_1) \quad \forall i \in m+1, ..., n$$

Since $Q(a)$ converges to $Q^*(a)$ given $\delta > 0 \quad \exists T > 0$ s.t. $\forall \tau < T \quad |Q(a) - Q^*(a)| < \delta$. Set δ s.t. $Q^*(a_1) - Q^*(b_i) > 2.\delta \quad \forall i \in m+1, ..., n$. Now lets consider the probabilities.

$$\begin{aligned}
P(a_i) &= \frac{e^{Q(a_1)/\tau}}{\sum_{b \in A} e^{Q(b)/\tau}} \\
&= \frac{e^{Q(a_1)/\tau}}{\sum_{i=1}^m e^{Q(a_1)/\tau} + \sum_{i=m+1}^n e^{Q(b_i)/\tau}} \\
&\geq \frac{e^{(Q_*(a_1)-\delta)/\tau}}{\sum_{i=1}^m e^{(Q_*(a_1)+\delta)/\tau} + \sum_{i=m+1}^n e^{(Q_*(b_i)+\delta)/\tau}} \\
&= e^{-2\delta/\tau} \frac{e^{Q_*(a_1)/\tau}}{\sum_{i=1}^m e^{Q_*(a_1)/\tau} + \sum_{i=m+1}^n e^{Q_*(b_i)/\tau}}
\end{aligned}$$

Using the arguments in lemma 35

$$\geq e^{-2\delta/\tau}\left(P^*(a) - \delta\right)$$
$$= P^*(a) - \left(1 - e^{-2\delta/\tau}\right)P^*(a) - e^{-2\delta/\tau}\delta$$

Since $e^{-2\delta/\tau} \rightarrow 1$ as $\tau \rightarrow 0$ and $P^*(a) \in [0,1]$, given $\epsilon > 0 \quad \exists T > 0$ s.t. $\tau < T \quad \left(1 - e^{-2\delta/\tau}\right)P^*(a) + e^{-2\delta/\tau}\delta < \epsilon$

$$\therefore P(a) \geq P^*(a) - \epsilon$$

Similiarly, $P(a) \leq P^*(a) + \epsilon$ □

From lemma 39, it is know that the Q-values for a state before a pre-terminal state converges and hence from lemma 40, it is known that this states probabilities converge to a Nash Equilibrium policy's probabilities.

D.4 Inductive Step

All the conditions have been included in the previous lemmas and corollaries, now an inductive step can be taken.

THEOREM 41. *The VND policy to the finite sequential game described in Chapter Six converges toward a Nash Equilibrium policy as the temperature is decreased to zero.*

Proof. The expected values of actions are finite by the definition of the finite sequential game given in Chapter Six. Thus lemma 35 shows that the probabilities of action selection from a VND policy converge a Nash Equilibrium policy as the temperature parameter is decreased to zero. Thus, from corollary 37 and corollary 38, the expected value of a pre-terminal state also converges to the expected value achieved under a Nash Equilibrium policy (as the temperature parameter is decreased to zero).

From lemma 39, the F^i (i.e. the Q-value for the active player) is shown to converge correctly for actions selected before a pre-terminal state and hence the probability of selecting that action is shown to converge correctly (assuming that all preceeding states are terminal) from lemma 40.

By applying lemma 36 it is shown that the states before a pre-terminal state expected values also converge correctly. Thus all conditions have been satisified to apply the above lemmas to the state before this state and hence, by induction, the complete policy is shown to converge. □

Glossary

Acronym	Definition
AI	Artificial Intelligence
a.s.	Almost surely
DP	Dynamic Programming
ev	Eventually
GCC	GNU Compiler Collection
GNU	GNU's Not Unix
MinGW	Minimalist GNU for Windows
IDE	Integrated Development Environment
i.o.	Infinitley often
ISO	International Organisation for Standardization
KS	Kolmogorov-Smirnov
MDP	Markov Decision Process
MS	Management Science
OR	Operational Research
RAM	Random Access Memory
RL	Reinforcement Learning (unofficial acronym)
RM	Revenue Management
ROC	Receiver Operating Characteristics
s.t.	Such that
w.l.o.g.	Without loss of generality
VBA	Visual Basic for Applications
VND	Variation on the Nash Distribution

Notation	Defintion
233	A version of our airline game, with a simple customer model, where there are only two rounds before the flight leaves and each of the players has three seats available on their plane.
355	A version of our airline game, with a simple customer model, where there are only three rounds before the flight leaves and each of the players has five seats available on their plane.
Bullet-proofing	A computer programming term to describe the code-writing practice of ensuring that the program can handle any exceptions that occur during runtime.
Distance	The size or quantity that a measure gives.
Freeware	Software that is available free of charge for personal use.
Measure	Something that gives a size or quantity for comparison.
Metric	A non-negative symmetric binary function with certain properties. See the Dictionary of Mathematics (Borowski and Borwein, 1989) for more details.
Reward	Payoff obtained from a single round (or stage) of the game.
Return	Total reward obtained from a complete game or for the remaining rounds of a game.
Statistic	Quantitative data on any subject.

References

Ahmed, A. H. and C. A. Poojari (2008). An overview of the issues in the airline industry and the role of optimization models and algorithms. *Journal of the Operational Research Society 59*, 267–277.

Aldous, D. and P. Diaconis (1987). Strong uniform times and finite random walks. *Advances in Applied Mathematics 8*, 69–97.

Allais, M. (1953). Le comportement de lhomme rationnel devant le risque: critique des postulats et axiomes de lécole américaine. *Econometrica 21*, 503–546.

Anjos, M. F., R. C. H. Cheng, and C. S. M. Currie (2004). Maximizing revenue in the airline industry under one-way pricing. *Journal of the Operational Research Society 55*, 535–541.

Anjos, M. F., R. C. H. Cheng, and C. S. M. Currie (2005). Optimal pricing polices for perishable product. *European Journal of Operational Research 177*(1), 246–254.

Axelrod, R. (1984). *The evolution of co-operation.* Basic Books. ISBN 0-14-012495-0.

Axelrod, R. (1997). *The complexity of cooperation: Agent-based models of competition and collaboration.* Princeton University Press. ISBN 0-691-01568-6.

Bailey, P. W. (2003). The DIAMOND model of peace support operations. In *Analysis of the Military Effectiveness of Future C2 Concepts and Systems*, Volume RTO-MP-117 AC/323(SAS-039)TP/32. NATO Research and Technology Organisation. ISBN 9-283-70035-X (UNCLASSIFIED UNLIMITED).

Banerjee, B., S. Sen, and J. Peng (2004). On-policy concurrent reinforcement learning. *Journal of Experimental & Theoretical Artifical Intelligence 16*(4), 245–260.

Barzilai, J. (2007). Game theory foundational errors - part i. Department of Industrial Engineering, Dalhouse University, *mimeo*.

Bedaux, J. (2002). C++ mersenne twister pseudo-random number generator. http://www.bedaux.net/mtrand/. (accessed on 27th April 2007).

Beggs, A. W. (2005). On the convergence of reinforcement learning. *Journal of Economic Theory 122*, 1–36.

Bellman, R. (1952). On the theory of dynamic programming. *Proceedings of the National Academy of Sciences of the Unitied States of America 38*, 716–719.

Bellman, R. (1954). The theory of dynamic programming. *Bulletin of American Mathematical Society 60*, 503–516.

Bellman, R. (1957). A markovian decision process. *Journal of Mathematics and Mechanics 6*, 679–684.

Benaim, M. (1996). A dynamical system approach to stochastic approximations. *SIAM Journal on Control and Optimization 34*, 437–472.

Benaim, M. (1999). *Dynamics of Stochastic Approximation Algorithms*. Séminaire de probabilités de Strasbourg. Strasbourg: Springer-Verlag.

Benaim, M. and M. W. Hirsch (1999). Mixed equilibria and dynamical systems arising from fictitious play in perturbed games. *Games and Economic Behavior 29*, 36–72.

Bernoulli, D. (1738). Exposition of a new theory on the measurement of risk. *Commentaries of the Imperial Academy of Science of Saint Petersburg*. translated by Dr. Lousie Sommer (1954) in Econometrica 22(1): 2236.

Bertsekas, D. P. and J. N. Tsitsiklis (1996). *Neuro-dynamic programming*. Athena Scientific. ISBN 1-886-52910-8.

Binmore, K. (1990). *Essays on the foundations of Game Theory*. Basil Blackwell. ISBN 0-631-16866-4.

Bloodshed Software (2005). Bloodshed software - dev-c++. http://www.bloodshed.net/devcpp.html. (accessed on 27th April 2007).

Borm, P., H. Hamers, and R. Hendrickx (2001). Operations research games: A survey. *TOP 9*(2), 139–199.

Borowski, E. J. and J. M. Borwein (1989). *Dictionary of Mathematics*. Harper Collins. ISBN 0-004-34347-6.

Bowling, M. and M. Veloso (2002). Multiagent learning using a variable learning rate. *Artificial Intelligence 136*, 215–250.

Boyd, E. A. (2007). Perspectives on the future of pricing. In *7th Annual INFORMS Pricing and Revenue Management Conference*, Barcelona, Spain.

Braess, D. (1968). Uber ein paradoxon der verkenrsplanning. *Unterehmensforschung 12*(1), 258–268.

Braess, D., A. Nagurney, and T. Wakolbinger (2005). On a paradox of traffic planning. *Transportation Science 39*(4), 446–450.

Bridle, J. S. (1990). Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In D. S. Touretzky (Ed.), *Advances in neural information processing systems*, pp. 211–217. Morgan Kaufmann. ISBN 1-558-60100-7.

Brooks, R. J. and A. M. Tobias (1996). Choosing the best model: level of detail, complexity, and model performance. *Mathematical and Computer Modelling 24*(4), 1–14.

Brown, G. W. (1951). Iterative solution of game by fictitious play. In T. C. Koopmans (Ed.), *Activity Analysis of Production and Allociation*, Chapter 24, pp. 374–376. New York: Wiley.

Brown, R. G. and R. F. Meyer (1961). The fundamental theorem of exponential smoothing. *Operations Research 9*(5), 673–687.

Bryant, J. W. (2007). Drama theory: dispelling the myths. *Journal of the Operational Research Society 58*, 602–613.

Bryman, A. and E. Bell (2003). *Business Research Methods*. Oxford: Oxford University Press.

Camerer, C. and T.-H. Ho (1999). Experienced-weighted attraction learning in normal form games. *Econometrica 67*(4), 827–874.

Cantelli, F. P. (1917). Sulla probabilitá come limite della frequenza. *Rendiconti della R. Accademia dei Lincei: Classe di scienze fisiche matematiche e naturali 5a*(26), 39–45.

Chatterjee, K. and W. Samuelson (Eds.) (2001). *Game Theory and Business Applications*. Kluwer Academic Publishers. ISBN 0-792-37332-4.

Chebyshev, P. L. (1867). Des valeurs moyennes. *Journal de Mathematiques Pures et Appliquees 12*(2), 177–184.

Chen, X. and X. Deng (2005). Settling the complexity of 2-player Nash-Equilibrium. In *Electronic Colloquium on Computational Complexity*, Volume 140.

Chen, X. and F. B. Zhan (2008). Agent-based modelling and simulation of urban evacuation: relative effectiveness of simultaneous and staged evacuation strategies. *Journal of the Operational Research Society 59*(1), 25–33.

Chen, Y. and Y. Khoroshilov (2003). Learning under limited information. *Games and Economic Behavior 44*(1), 1–25.

Chick, S. (2006). Six ways to improve a simulation analysis. *Journal of Simulation 1*, 21–28.

Chinthalapati, V. L. R., N. Yadati, and R. Karumanchi (2006). Learning dynamic prices in multiseller electronic retail markets with price sensitive customers, stochastic demands, and inventory replenishments. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews 36*(1), 92–106.

Collins, A. J., F. Pullum, and L. Kenyon (2003). Applications of game theory in defence project: year one report. Dstl/CR07880, Defence Science and Technology Laboratories, Ministry of Defence, UK. (UNCLASSIFIED).

Cournot, A. A. (1838). *Recherches sur les principes mathématiques de la thorie des richesses (Researches into the Mathematical Principles of the Theory of Wealth).* Paris: Hachette. (1897, Engl. trans. by N.T. Bacon).

Currie, C., R. C. H. Cheng, and H. K. Smith (2006). Dynamic pricing of airline tickets with competition. University of Southampton, *mimeo*.

Das, T. K., A. Gosavi, S. Mahadevan, and N. Marchalleck (1999). Solving semi-markov decision problems using average reward reinforcement learning. *Management Science 45*(4), 560–574.

Dash, R. K., N. R. Jennings, and D. C. Parkes (2003). Computational-mechanism design: A call to arms. *IEEE Intelligent Systems 18*(6), 40–47.

Dayan, P. and T. J. Sejnowski (1994). Td(λ) converges with probability 1. *Machine Learning 14*, 295–301.

de Vries, S. and R. Vohra (2003). Combinatorial auctions: a survey. *INFORMS Journal on computing 15*, 284–309.

Debreu, G. (1960). Review of individual choice behaviour: A theoretical analysis by R. Duncen Luce. *The American Economic Review 50*, 186–188.

Durrett, R. (2004). *Probability: Theory and Examples* (3rd ed.). Duxbury Press. ISBN 0-534-42441-4.

Dvoretzky, A. (1956). On stochastic approximation. In *Third Berkeley Symposium on Mathematical Statistics and Probability*, pp. 39–55.

Eatwell, J., M. Milgate, and P. Newman (Eds.) (1987). *The new palgrave: Game Theory*. Macmillan Press. ISBN 0-333-49537-3.

Erev, I. and A. E. Roth (1998). Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *The American Economic Review 88*(4), 848–881.

Feltovich, N. (2000). Reinforcement-based vs. belief-based learning models in experimental asymmetric-information games. *Econometrica 68*(3), 605–641.

Fisher, R. A. (1925). *Statistical methods of research workers*. Oliver and Boyd Ltd.

Fogel, D. B., T. J. Hays, S. L. Hahn, and J. Quon (2005). Further evolution of a self-learning chess program. In *IEEE 2005 Symposium on Computational Intelligence and Games*, Colchester, Essex, UK, pp. 73–77.

Friedman, M. (1953). *Essays in positive economics*. University of Chicago Press.

Fudenberg, D. and D. M. Kreps (1993). Learning mixed equilibria. *Games and Economic Behavior 5*, 320–367.

Fudenberg, D. and D. M. Kreps (1994). Learning in extensive-form games: II. experimentation and nash equilibrium. Harvard University and Stanford University, *mimeo*.

Fudenberg, D. and D. M. Kreps (1995). Learning in extensive-form games: I. self-confirming equilibria. *Games and Economic Behavior 8*, 20–55.

Fudenberg, D. and D. K. Levine (1995). Universal consistency and cautious fictitious play. *Journal of Economic Dynamics and Control 19*, 1065–1089.

Fudenberg, D. and D. K. Levine (1998). *The theory of learning in games*. MIT Press. ISBN 0-262-06194-5.

Fudenberg, D. and D. K. Levine (1999). Conditional universal consistency. *Games and Economic Behavior 29*(1), 104–130.

Fudenberg, D. and J. Tirole (1991). *Game Theory*. MIT Press. ISBN 0-262-06141-4.

Gibbs, A. L. and F. E. Su (2002). On choosing and bounding probability metrics. *International Statistical Review 70*(3), 419–435.

Glance, N. S. and B. A. Huberman (1994). The dynamics of social dilemmas. *Scientific American March*, 76–81.

GNU Project (2007). Gcc: The gnu compiler collection. http://gcc.gnu.org. (accessed on 27th April 2007).

Goodwin, D. (2005). *The Military And Negotiation: The Role Of The Soldier-diplomat*. Talyor & Francis. ISBN 0-203-01028-0.

Gosavi, A. (2003). *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Operations Research / Computer Science Interfaces Series. London: Kluwer Academic Publishers. ISBN 1-402-07454-9.

Gosavi, A. (2004). Reinforcement learning for long-run average cost. *European Journal of Operational Research 155*, 654–674.

Gosavi, A., N. Bandia, and T. K. Das (2002). A reinforcement learning approach to a single leg airline revenue management problem with multiple fare classes and overbooking. *IIE Transactions 34*(9), 729–742.

Gosavi, A., N. Bandia, and T. K. Das (2007). Simulation optimization for revenue management of airlines with cancellation and overbooking. *OR Spectrum 29*(1), 21–38.

Grimmett, G. R. and D. R. Stirzaker (1992). *Probability and Random Processes* (2nd ed.). Oxford University Press. ISBN 0-198-53665-8.

Gumbel, E. J. (1958). *Statistics of Extremes*. New York: Columbia University Press.

Hamilton, W. R. (1852). *Discussions on philosophy and literature, education and university reform*. London.

Harsanyi, J. C. and R. Selten (1988). *A general theory of equilibrium selection in games*. MIT Press. ISBN 0-262-58238-4.

Hart, S. and A. Mas-Colell (2003). Uncoupled dynamics do not lead to Nash Equilibrium. *The American Economic Review 93*(5), 1830–1836.

Hart, S. and A. Mas-Colell (2006). Stochastic uncoupled dynamics and Nash Equilibrium. *Games and Economic Behavior 57*, 286–303.

Herings, P. J.-J., A. Mauleon, and V. J. Vannetelbosch (2003). Fuzzy play, matching devices and coordination failures. *International Journal of Game Theory 32*, 519–513.

Hill, R., R. Carl, and L. Champagne (2006). Using agent-based simulation to empirically examine search theory using a historical case study. *Journal of Simulation 1*(1), 29–38.

Howard, N. (2001). *Drama without tears*. Dramatec.

Hume, D. (1740). *A treatise of human nature: Being an attempt to introduce the experimental Method of reasoning into moral subjects*, Volume 3. London: Thomas Longman.

Ishikawa, K., A. Sakurai, T. Fujinami, and S. Kunifuji (2007). R-learning with multiple state-action value tables. *Electrical Engineering in Japan 159*(3), 72–82.

Jones, G. A. and J. M. Jones (1998). *Elementary Number Theory*. Springer. ISBN 3-540-76197-7.

Kaelbling, L. P., M. L. Littman, and A. W. Moore (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research 4*, 237–285.

Kim, J. S. and T. C. Kwak (2007). Game theoretic analysis of the bargaining process over long-term replenishment contract. *Journal of the Operational Research Society 58*(6), 769–778.

Kobbary, K. A. H., S. Vadera, and M. H. Rasmy (2007). AI and OR in management of operations: History and trends. *Journal of the Operational Research Society 58*(1), 10–28.

Kolmogorov, A. N. (1933). Sulla determinazione empirica di una legge di distribuzione. *Giornale dell'Istituto Italiano degli Attuari 4*, 83–91.

Kőnőnen, V. (2006). Dynamic pricing based on asymmetric multiagent reinforcement learning. *International Journal of Intelligent Systems 21*, 73–98.

Kott, A. and W. M. McEneaney (Eds.) (2007). *Adversal Reasoning: Computational Approaches to Reading the Opponent's mind*. Chapman & Hall. ISBN 1-584-88588-2.

Kruschke, J. K. and N. J. Blair (2000). Blocking and backward blocking involve learned inattention. *Psychonomic Bulletin & Review 7*(4), 636–645.

Kuhn, H. W. and S. Nasar (Eds.) (2002). *The essential John Nash*. Princeton University Press. ISBN 0-691-09527-2.

Kushner, H. J. and G. G. Yin (2003). *Stochastic Approximation and Recursive Algorithms and Applications*. Springer. ISBN 0-387-00894-2.

Laslier, J.-F., R. Topol, and B. Walliser (2001). A behavioral learning process in games. *Games and Economic Behavior 37*, 340–366.

Laslier, J.-F. and B. Walliser (2005). A reinforcement learning process in extensive form games. *International Journal of Game Theory 33*(2), 219–227.

Lazear, E. P. (1986). Retail pricing and clearence sales. *The American Economic Review 76*, 14–32.

Leslie, A. M. (2001). Learning: Association or computation? introduction to the special section. *Current Directions in Psychological Science 10*(4), 124–127.

Leslie, D. S. and E. J. Collins (2003). Convergent multiple-timescales reinforcement learning algorithms in normal form games. *The Annals of Applied Probability 13*(4), 1231–1251.

Leslie, D. S. and E. J. Collins (2005). Individual q-learning in normal form games. *SIAM Journal on Control and Optimization 44*(2), 495–514.

Leslie, D. S. and E. J. Collins (2006). Generalised weakened fictitious play. *Games and Economic Behavior 56*, 285–298.

Liberty, J. (1999). *Sam's Teach Yourself C++ in 21 Days* (3rd ed.). Sams Publishing. ISBN 0-672-31564-5.

Littman, M. L. (1994). Markov games as a framework for multiagent reinforcement learning. In *Eleveth International Conference on Machine Learning*, San Francisco.

Luce, R. D. (1959). *Individual Choice Behaviour*. Wiley.

Machiavelli, N. (1532). *Il Principe*. Antonio Blado d'Asola.

Manski, C. F. (1977). The structure of random utility models. *Theory and Decision 8*, 229–254.

Matsumoto, M. and T. Nishimura (1998). Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Transactions on Modelling and Computer Simulation 8*(1), 3–30.

Maynard Smith, J. (1974). The theory of games and the evolution of animal conflicts. *Journal of Theoretical Biology 47*, 209–221.

Maynard Smith, J. (1982). *Evolution and the theory of games*. Cambridge University Press. ISBN 0-521-28884-3.

McFadden, D. (1980). Econometric models of probabilistic choice among products. *Journal of Business 53*, S13–S29.

McKelvey, R. D., A. M. McLennan, and T. L. Turocy (2007). Gambit: Software tools for Game Theory - version 0.2007.01.30. http://econweb.tamu.edu/gambit. (accessed on 10th September 2007).

Mclennan, A. (2005). The expected number of Nash equilibria of a normal form game. *Econometrica 73*(1), 141–174.

Michie, D. and R. A. Chambers (1968). BOXES: An experiment in adaptive control. In E. Dale and D. Michie (Eds.), *Machine Intelligence 2*, pp. 137–152. Edinburgh: Oliver and Boyd.

Minsky, M. L. (1954). *Theory of Neural-Analog Reinforcement Systems and Its Application to the Brain-Model Problem*. Ph. D. thesis, Princeton University.

Műller, M. (2002). Computer GO. *Artificial Intelligence 134*, 145–179.

Murphy, F. H. (2005). ASP, the art and science of practice: Elements of a theory of the practice of operations research: A framework. *Interfaces 35*(2), 154–163.

Nash, J. (1951). Non-cooperative games. *The Annals of Mathematics 54*(2), 286–295.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. New York: Springer-Verlag. ISBN 0-387-94724-8.

Oum, T. H. (1979). A warning on the use of linear logit models in transport mode choice studies. *Bell Journal of Economics 10*, 374–388.

Pavlov, I. P. (1927). *Conditioned Reflexes*. Oxford University Press.

Persky, J. (1995). Retrospectives: The ethology of Homo Economicus. *The Journal of Economic Perspectives 9*(2), 221–223.

Pidd, M. (1996). *Tools for Thinking: Modelling in Management Science*. Wiley. ISBN 0-471-96455-7.

Prasnikar, V. and A. E. Roth (1992). Considerations of fairness and strategy: Experimental data from sequential games. *The Quarterly Journal of Economics 107*(3), 865–888.

Quinton, C. (2007). The Operational Research society. http:// www.orsoc.org.uk. (accessed on 10th September 2007).

Ravulapati, K. K., J. Rao, and T. K. Das (2004). A reinforcement learning approach to stochastic business games. *IIE Transactions 36*, 373–385.

Rescorla, R. A. and A. R. Wagner (1972). A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black and W. F. Prokasy (Eds.), *Classical Conditioning II: Current Theory and Research*, Century Psychology Series, Chapter 3. Appleton-Century-Crofts.

Robbins, H. and S. Monro (1951). A stochastic approximation method. *Annals of Mathematical Studies 22*, 400–407.

Robinson, S. (2007). A statistical process control approach to selecting a warm-up period for a discrete-event simulation. *European Journal of Operational Research 176*, 332–346.

Roth, A. E. and I. Erev (1995). Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term. *Games and Economic Behaviour 8*, 164–212.

Roughgarden, T. and E. Tardos (2002). How bad is selfish routing. *Journal of the Association of Computing Machinery 49*(2), 236–259.

Rummery, G. A. (1995). *Problem Solving With Reinforcement Learning*. Ph. D. thesis, Cambridge University.

Rummery, G. A. and M. Niranjan (1994). On-line q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR 166, Engineering Department, Cambridge University.

Russell, S. I. and P. Norvig (1995). *Artificial Intelligence: A Modern Approach*. New Jersey: Prentice Hall. ISBN 0-137-90395-2.

Sato, Y., E. Akiyama, and F. J. Doyne (2002). Chaos in learning a simple two-person game. *Proceedings of the national academy of sciences of the USA 99*(7), 4748–4751.

Schipper, Y., P. Nijkamp, and Rietveld (2007). Deregulation and welfare in airline markets: An analysis of frequency equilibria. *European Journal of Operational Research 178*(1), 194–206.

Shannon, C. E. (1950). Programming a computer for playing chess. *Philosphical Magazine 41*, 256–275.

Shoham, Y., R. Powers, and T. Grenager (2004). Multi-agent reinforcement learning: A critical survey. In *AAAI Fall Symposium on Artificial Multi-Agent Learning*.

Sikora, R. T. (2006). Learning optimal parameter values in a dynamic environment: An experiment with softmax reinforcement learning algorithm. In *INFORMS Annual Meeting: Artificial Intelligence and Data Mining Workshop*, Pittsburgh.

Singh, S., T. Jaakkola, M. L. Littman, and C. Szepesvari (2000). Convergence results for single-step on-policy reinforcement-learning algorithm. *Machine Learning 39*, 287–308.

Sridharan, M. and G. Tesauro (2000). Multi-agent q-learning and regression trees for automated pricing decisions. In *Proceedings 17th International Conference on Machine Learning*, Stanford, USA, pp. 927–934.

Stilman, B. (2000). *Linguistic Geometry: From Search to Construction*. Operations Research / Computer Science Interfaces Series. Springer. ISBN 0-792-37738-9.

Sutton, R. S. and A. G. Barto (1998). *Reinforcement Learning: An Introduction*. MIT Press. ISBN 0-262-19398-1.

Takadama, K. and H. Fujita (2005). Toward guidelines for modeling learning agents in multiagent-based simulation: Implications from Q-learning and SARSA agents. *Lecture Notes in Computer Science 3415*, 159–172.

Talluri, K. T. and G. J. van Ryzin (2004). *Theory and Practice of Revenue Management*. Springer-Verlag. ISBN 1-402-07701-1.

Tesauro, G. and J. O. Kephart (2002). Pricing in agent economies using multi-agent q-learning. *Autonomous Agents and Multi-Agent Systems 5*(3), 289–304.

Thomas, L. C. (1984). *Games, theory and application*. Wiley. ISBN 0-486-43237-8.

Thomas, L. C. (2003). The best banking strategy when playing The Weakest Link. *Journal of the Operational Research Society 54*, 747–750.

Thomas, L. C., D. B. Edelman, and J. N. Crook (2002). *Credit Scoring and Its Applications*. Monographs on mathematical modeling and computation. Society for Industrial and Applied Mathematics (SIAM). ISBN 0-898-71483-4.

Thomas, L. C. and P. B. Hulme (1997). Searching for targets who want to be found. *Journal of the Operational Research Society 48*, 44–50.

Thorndike, E. L. (1911). *Animal Intelligence*. Darien, CT: Hafner.

Thurstone, L. L. (1927a). A law of comparative judgement. *Psychological Review 34*, 273–286.

Thurstone, L. L. (1927b). Psychological analysis. *American Journal of Psychology 38*, 368–389.

Valluri, A. (2006). Learning and cooperation in sequential games. *Adaptive Behaviour 14*(3), 195–209.

van Ryzin, G. and J. McGill (2000). Revenue management without forecasting or optimization: An adaptive algorithm for determining airline seat protection levels. *Management Science 46*(6), 760–775.

von Neumann, J. and O. Morgenstern (1944). *Theory of games and economic behaviour*. Princeton University Press. ISBN 0-691-11993-7.

von Stackelberg, H. F. (1934). *Marktform und Gleichgewicht.* Vienna: Julius Springer.

Ward, S. C. (1989). Arguments for constructively simple models. *Journal of the Operational Research Society 40*(2), 141–153.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards.* Ph. D. thesis, Cambridge University.

Watkins, C. J. C. H. and P. Dayan (1992). Q-learning. *Machine Learning 8*, 279–292.

Weibull, J. W. (1995). *Evolutionary Game Theory.* MIT Press. ISBN 0-262-23181-6.

Welsh, D. (1988). *Codes and Cryptography.* Oxford University Press. ISBN 0-198-53287-3.

Williams, D. (1991). *Probability with Martingales.* Cambridge University Press.

Winston, W. L. (1993). *Operations research: Application and algorithms* (3 ed.). Duxbury Press. ISBN 0-534-20971-8.

Wolfowitz, J. (1956). On stochastic approximation methods. *The Annals of Mathematical Statistics 27*(4), 1151–1156.

Yoshizaki, H. T. Y. and G. A. Plonski (1995). A contextual approach of designing and using OR quantitative models. *International Transactions in Operational Research 2*(4), 309–319.

Zizzo, D. and D. Sgroi (2007). Neural networks and bounded rationality. *Physica A: Statistical Mechanics and its Applications 375*(2), 717–725.