

A PRINCIPLED METHODOLOGY FOR THE DESIGN
OF AUTONOMOUS TRADING AGENTS WITH
COMBINATORIAL PREFERENCES IN THE PRESENCE
OF TRADEOFFS

A Dissertation

Presented to the Faculty of the Graduate School
of Cornell University

in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by

Ioannis A. Vetsikas

August 2005

A PRINCIPLED METHODOLOGY FOR THE DESIGN OF AUTONOMOUS
TRADING AGENTS WITH COMBINATORIAL PREFERENCES IN THE
PRESENCE OF TRADEOFFS

Ioannis A. Vetsikas, Ph.D.

Cornell University 2005

Online auctions have become a popular method for business transactions. The variety of different auction rules, the restrictions in supply or demand, and the agents' combinatorial preferences for the different commodities, have led to the creation of a very complex multi-agent "environment" and a number of strategic tradeoffs. Designing an agent that deals efficiently with these tradeoffs has been a multi-pronged effort. Using game-theoretic approaches, some equilibria have been computed for relatively simple auctions. However, since these equilibria have limited practical application, due to the significant number of varying auctions that take place simultaneously, empirical approaches and experimental evaluations of various strategies have also been used. Furthermore, progress has been made into designing better agent architectures.

This dissertation presents results in all of these directions (theoretical and empirical). We present a methodology for designing trading agents, and deciding their bidding strategy, when they participate in a large number of simultaneous auctions with a variety of rules. We use a modular, adaptive, scalable and robust agent architecture, combining principled methods and empirical knowledge. We decompose the problem faced by the agent into several components, and restrict communication between them. This allows us to analyze each component individually. The "optimizer" coordinates the effort of the entire system and determines the set of commodities that maximizes the utility of the

agent. We provide a principled way of generating strategies for each bidding module. Then we use rigorous experimentation to explore the strategy space and determine the best combination of strategies. This enables the agent to perform efficiently against any opponent agents. We also expand this methodology to include design decisions based on the equilibria computed for particular auctions. Furthermore, we present several novel Bayes-Nash equilibria for m^{th} price multi-unit auctions with multiple possible closing times, one of which is chosen randomly, in which multiple rounds of bidding can occur.

We applied this methodology when creating WhiteBear, the winner of the 2002 and 2004 Trading Agent Competitions. We also present the “complete” set of experiments for determining an overall best strategy in TAC, which guided the design of WhiteBear. We show that by optimizing over the strategy space one obtains highly effective and flexible agents that outperform other designs.

BIOGRAPHICAL SKETCH

Ioannis A. Vetsikas graduated from the Department of Electrical and Computer Engineering of the National Technical University of Athens, Greece, in 1998 at the top of his class. He followed the program of the Computer Engineering and Informatics (Computer Science) direction of the department. He then entered the Department of Computer Science at Cornell University and graduated in 2005. At Cornell, he worked on the design of trading agents and he is interested in e-Commerce, game theory and AI. He worked alone on the design and implementation of several trading agents, which were entered under the name “WhiteBear” in the Trading Agent Competition, and were among the top scoring in the competition. He worked under the supervision of Professor Bart Selman.

To my family and friends.

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to Professor Bart Selman, my scientific advisor at Cornell University. He was the one who introduced me into the Trading Agent Competition, at a time when I was still uncertain about my research direction, except for the fact that it had to involve AI and probably games or game theory. He never failed to provide his support and his wisdom during the long and solitary journey that was my Ph.D.

My special thanks and acknowledgement go to my committee members at Cornell, Professor Joe Halpern and Professor Steve Vavasis. Steve Vavasis has been helpful in providing guidance with the solution of the differential equations that were part of this thesis. Professor Steven Strogatz helped in that direction as well, by verifying that the differential equations for the multi-round auctions do not have a known analytical solution. Finally, I would like to thank the organizers of the Trading Agent Competition, that is the University of Michigan agent group and SICS, for their assistance in getting our agent running and with our experiments.

I also acknowledge the help of my friends, who not only kept me sane and entertained during my years at Cornell, but also helped me with some of the smaller problems that I encountered, and provided interesting discussions : Cristi, Sasha, Pavel, Dmitriy, Zoya, Liviu, Alin, Alexei, Radu. I would also like to thank anyone else who assisted me in any way.

Finally, I would like to thank my parents and relatives for their constant support, by word and by action.

TABLE OF CONTENTS

1	Introduction	1
1.1	General Problem Setting	4
1.2	Motivating Example: The TAC Game	6
1.2.1	The TAC Game as a Competitive Benchmark	9
1.3	Thesis Statement and Contributions	10
1.4	Guide to the Thesis	14
2	Background and Overview	16
2.1	Game Theory and Solution Concepts	16
2.1.1	Basic Definitions	16
2.1.2	Solution Concepts	18
2.2	Auction Theory	21
2.2.1	First-price Sealed-Bid Auctions	24
2.3	Combinatorial Auctions	25
2.3.1	Formal Definition	26
2.3.2	Solving the Winner Determination Problem	28
2.4	Multiple Auctions	30
2.4.1	Single-Sided Auctions	30
2.4.2	Double-Sided Auctions	32
2.4.3	Sequential Auctions	34
2.4.4	Strategies Used by TAC agents	35
2.5	Experimental Equilibria	39
2.5.1	Hierarchical Reduction of Large Symmetric Games	41
2.6	Other Interesting Auction Related Work	43

3	Methodology and Agent Architecture	46
3.1	Problem Decomposition	47
3.1.1	Optimization Module	47
3.1.2	Bidding Modules	48
3.2	Experimental Setup	50
3.3	Agent Architecture	52
4	Optimizer	55
4.1	Price Estimate Vectors	55
4.1.1	Prediction Improvements	56
4.1.2	Implementation in our TAC Agent	58
4.2	Optimizer	59
5	Generating Strategies and Dealing with Tradeoffs	66
5.1	Paying for Flexibility: The Flight Auctions	66
5.1.1	Tradeoffs in TAC 2001 Game	67
5.1.2	Tradeoffs in TAC 2004 Game	72
5.1.3	Flight Price Models	73
5.2	Bid Aggressiveness: The Hotel Room Auctions	77
5.3	Entertainment	82
6	Using Equilibrium Analysis to Generate Strategies	83
6.1	Equilibria For a Two Round Auction	84
6.1.1	Formal Definition of the Problem	84
6.1.2	Useful Mathematical Knowledge	86
6.1.3	Bayes-Nash Equilibria For a Single Unit Auction	88
6.1.4	Bayes-Nash Equilibria For a Multi-unit Auction	96

6.2	Bayes-Nash Equilibria for Multiple Round Auctions	112
6.2.1	Formal Definition of the Problem	113
6.2.2	Computing the Equilibria	114
6.2.3	Numerical Evaluation Of Equilibrium Strategies	117
7	Experiments	138
7.1	Performance Evaluation Between Competing Agents	138
7.1.1	Reducing the Number of Required Runs By Estimating the Ef- fect of Random Parameters	139
7.2	Experiments for Determining TAC 2001-2003 Agent	141
7.2.1	Modeling the Flight Auctions' Prices	143
7.2.2	Evaluating the Aggressiveness Tradeoff	145
7.2.3	Evaluating the "Paying For Flexibility" Tradeoff	148
7.2.4	Using Historical Averages In Price Predictions	150
7.2.5	Re-evaluating the "Paying For Flexibility" Tradeoff	151
7.2.6	Re-evaluating the Aggressiveness Tradeoff	154
7.2.7	Using Walrasian Equilibrium In Price Predictions	157
7.2.8	Agent Design For TAC Competition	159
7.3	Experiments for Determining TAC 2004-2005 Agent	160
7.3.1	Evaluating the Aggressiveness Tradeoff	160
7.3.2	Agent Design For TAC Competition	166
8	Conclusions	168
	Bibliography	171

LIST OF TABLES

7.1	Average scores of agents WB-N2L, WB-M2L, WB-M2M and WB-M2H. For experiment 1, the scores of the 2 instances of each agent type are also averaged. <i>The number inside the parentheses is the total number of games for each experiment, and this will be the case for every table.</i>	144
7.2	Scores for agents WB-M2L, WB-M2M and WB-M2H as the number of aggressive agents (WB-M2H) participating increases. In each experiment, agents 1 and 2 are instances of WB-M2M. The agents above the stair-step line are WB-M2L, while the ones below are WB-M2H. The averages scores for each agent type are presented in the next rows. In the last rows, \checkmark indicates statistically significant difference in the scores of the corresponding agents, while \times indicates statistically similar scores.	146
7.3	Scores for agents WB-M2M and WB-M0H as the number of early bidding agents (WB-M0H) participating increases. The agents above the stair-step line are WB-M2M, while the ones below are WB-M0H. . . .	148
7.4	The effect of using historical averages in the PEVs. Early bidding agents benefit the most from this.	150
7.5	Scores for agents WB*A2z, WB*ASz and WB*A0z (where z=M or H), as the number of early bidding agents (WB-A0z) participating increases. In each experiment, agent 1 is WB*ASM and 2 is WB*ASH. The agents above the stair-step line are WB*A2z, while the ones below are WB*A0z (the scores when z=M are presented in <i>italic</i>). In the last rows, we compared the scores for the M and H aggressiveness of the two versions; so M \checkmark in the A2z/Asz box means that the difference between A2M and ASM is significant.	152
7.6	Scores for agents WB*AyM and WB*AyH (where y=S or 0) as the number of aggressive agents (WB*AyH) participating increases. The agents above the stair-step line are WB*AyM, while below are WB*AyH (the scores when y=0 are presented in <i>italic</i>).	154
7.7	Scores for agents WB*ASL, WB*ASM, WB*ASP and WB*ASH as the number of aggressive agents (ASH) participating increases. In each experiment, agents 1 and 2 are WB*ASP and 3 and 4 are WB*ASM. The agents above the stair-step line are WB*ASL, while the ones below are WB*ASH.	156
7.8	Evaluating the use of prices derived from a Walrasian equilibrium at the beginning of the game.	158

7.9	Scores for agents of low (L), moderate (M) and high (H), bidding aggressiveness as the number of aggressive agents participating increases. In each experiment, agents 1 and 2 are of moderate (M) aggressiveness. The agents above the stair-step line are of low aggressiveness (L), while the ones below are the most aggressive (H). The average scores for each agent type are presented in the next rows. In the last rows, ✓ indicates statistically significant difference in the scores of the corresponding agents, while × indicates statistically similar scores.	161
7.10	Scores for agents of low (L), moderate (M) and high (H) bidding aggressiveness, as well as the equilibrium (E) agent, as the number of aggressive agents participating increases. In each experiment, agents 1 and 2 are equilibrium (E) agents, and agents 3 and 4 are of moderate (M) aggressiveness. The agents above the stair-step line are of low aggressiveness (L), while the ones below are the most aggressive (H). The average scores for each agent type are presented in the next rows. In the last rows, ✓ indicates statistically significant difference in the scores of the corresponding agents, while × indicates statistically similar scores. (The performance of the equilibrium agent is dominated by the performance of both the H and M agent, thus is not included in the last rows.)	163
7.11	Scores for agents of low (L), moderate (M) and high (H) bidding aggressiveness as well as the P agent, as the number of aggressive agents participating increases. In each experiment, agents 1 and 2 are P agents, and agents 3 and 4 are of M agents. The agents above the stair-step line are of low aggressiveness (L), while the ones below are the most aggressive (H). The averages scores for each agent type are presented in the next rows. In the last rows, ✓ indicates statistically significant difference in the scores of the corresponding agents, while × indicates statistically similar scores. (The performance of the L agent is dominated by the performance of the P and the M agent in most cases, thus is not included in the last rows.)	165

LIST OF FIGURES

1.1	Illustration of the environment a TAC agent operates within. (taken from www.sics.se/tac)	7
3.1	Diagram of the problem decomposition according to our methodology.	47
3.2	The agent architecture used by our methodology.	53
6.1	The graphs from top to bottom represent the solution of Equation 6.11, when $\frac{1-p}{p}$ takes values 1, 2, 5, 10 and 20 respectively.	95
6.2	Algorithm for solving the system of Theorem 9.	118
6.3	The experimental distribution $F_r(u)$ for round r (r takes values between 1 and 8 as shown by the lines from top to bottom respectively) that we used for our experiments. It has been created by sampling the utilities from a large number of real TAC games.	118
6.4	Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 8 in the case of a uniform distribution with $N = 2$ and $m = 1$. The equilibrium from Theorem 10 is the same.	121
6.5	Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 7, in the case of a uniform distribution with $N = 2$ and $m = 1$	122
6.6	Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 7, in the case of a uniform distribution with $N = 2$ and $m = 1$	122
6.7	Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 5, in the case of a uniform distribution with $N = 2$ and $m = 1$	123
6.8	Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 5, in the case of a uniform distribution with $N = 2$ and $m = 1$	123
6.9	Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 1, in the case of a uniform distribution with $N = 2$ and $m = 1$	124
6.10	Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 1, in the case of a uniform distribution with $N = 2$ and $m = 1$	124
6.11	Expected utility $EU_r(u, Q)$ from Theorem 9 for round 8, in the case of a uniform distribution with $N = 2$ and $m = 1$	125
6.12	Expected utility $EU_r(u, Q)$ from Theorem 10 for round 8, in the case of a uniform distribution with $N = 2$ and $m = 1$	125
6.13	Expected utility $EU_r(u, Q)$ from Theorem 9 for round 1, in the case of a uniform distribution with $N = 2$ and $m = 1$	126
6.14	Expected utility $EU_r(u, Q)$ from Theorem 10 for round 1, in the case of a uniform distribution with $N = 2$ and $m = 1$	126
6.15	Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 8, in the case of a uniform distribution with $N = 64$ and $m = 16$. The equilibrium from Theorem 10 is the same.	127
6.16	Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 5, in the case of a uniform distribution with $N = 64$ and $m = 16$	128
6.17	Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 5, in the case of a uniform distribution with $N = 64$ and $m = 16$	128

6.18	Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 1, in the case of a uniform distribution with $N = 64$ and $m = 16$	129
6.19	Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 1, in the case of a uniform distribution with $N = 64$ and $m = 16$	129
6.20	Expected utility $EU_r(u, Q)$ from Theorem 9 for round 8, in the case of a uniform distribution with $N = 64$ and $m = 16$	130
6.21	Expected utility $EU_r(u, Q)$ from Theorem 10 for round 8, in the case of a uniform distribution with $N = 64$ and $m = 16$	130
6.22	Expected utility $EU_r(u, Q)$ from Theorem 9 for round 1, in the case of a uniform distribution with $N = 64$ and $m = 16$	131
6.23	Expected utility $EU_r(u, Q)$ from Theorem 10 for round 1, in the case of a uniform distribution with $N = 64$ and $m = 16$	131
6.24	Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 8, in the case of a uniform distribution with $N = 64$ and $m = 16$. The equilibrium from Theorem 10 is the same.	132
6.25	Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 6, in the case of the experiment derived distribution with $N = 64$ and $m = 16$	133
6.26	Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 6, in the case of the experiment derived distribution with $N = 64$ and $m = 16$	133
6.27	Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 4, in the case of the experiment derived distribution with $N = 64$ and $m = 16$	134
6.28	Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 4, in the case of the experiment derived distribution with $N = 64$ and $m = 16$	134
6.29	Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 1, in the case of the experiment derived distribution with $N = 64$ and $m = 16$	135
6.30	Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 1, in the case of the experiment derived distribution with $N = 64$ and $m = 16$	135
6.31	Expected utility $EU_r(u, Q)$ from Theorem 9 for round 8, in the case of the experiment derived distribution with $N = 64$ and $m = 16$	136
6.32	Expected utility $EU_r(u, Q)$ from Theorem 10 for round 8, in the case of the experiment derived distribution with $N = 64$ and $m = 16$	136
6.33	Expected utility $EU_r(u, Q)$ from Theorem 9 for round 4, in the case of the experiment derived distribution with $N = 64$ and $m = 16$	137
6.34	Expected utility $EU_r(u, Q)$ from Theorem 10 for round 4, in the case of the experiment derived distribution with $N = 64$ and $m = 16$	137
7.1	Average scores for some agents that do, and some that do not, model the flight ticket prices. Two different experiments are presented. For the exact scores, see Table 7.1.	144
7.2	Changes in agents' average scores as the number of aggressive agents participating in the game increases. For the exact scores, see Table 7.2.	145
7.3	Changes in agents' average scores as the number of early bidders participating increases. For the exact scores, see Table 7.3.	149

7.4	The effect of using historical averages in the PEVs. For the exact scores, see Table 7.4.	151
7.5	Changes in agents' average scores as the number of early bidders participating increases. For the exact scores, see Table 7.5.	153
7.6	Changes in agents' average scores as the number of aggressive agents participating in the game increases. For the exact scores, see Table 7.6.	155
7.7	Changes in agents' average scores as the number of aggressive agents participating in the game increases. For the exact scores, see Table 7.7.	156
7.8	Changes in agents' average scores as the number of aggressive agents participating increases. For the exact scores, see Table 7.9.	162
7.9	Changes in agents' average scores as the number of aggressive agents participating increases. For the exact scores, see Table 7.10.	164
7.10	Changes in agents' average scores as the number of aggressive agents participating increases. For the exact scores, see Table 7.11.	166

Chapter 1

Introduction

Auctions are becoming an increasingly popular mechanism for business transactions over the Internet both between individuals using, e.g., eBay, and between businesses and their suppliers. While a good deal of research on auction theory exists, this is mostly from the auction mechanisms point-of-view (for a survey, see [43]). Strategies for bidding in an auction for a single item are also known. However, in practice, agents (or humans) are rarely interested in a single item; and, even when looking for a single item, users on, e.g. eBay, often consider the same item in several auctions. In fact, they may wish to bid in several auctions in parallel, for multiple interacting goods. In this case, they must bid intelligently in order to get exactly what they need. For example, a person may wish to buy a TV and a VCR, but if (s)he does not have a *flexible plan*, (s)he may end up acquiring only the VCR. On the other hand, if (s)he bids for more than one TV, (s)he might end up with 2 different TVs. It is apparent that even in this simple scenario, it is necessary for a human, or an agent who acts on your behalf, to bid intelligently in such auctions, in order to get what they wish and maximize their utility (i.e., happiness or profit). The complexity of the problem in this case comes from the fact that an agent (or human) has a different valuation (or preference) for a combination of items compared to their valuations independently. Therefore, we call these items complementary (or substitutable), depending on whether their combined valuation is higher (or lower). If we set $V(S)$ to be the valuation of a set of items S , then we can give the following two definitions.

Definition 1. Two goods a, b are called *complementary*, if their valuation as a set is higher than the sum of their individual valuations:

$$V(\{a, b\}) > V(\{a\}) + V(\{b\})$$

An example of this is the case of an agent wishing to buy a TV and a VCR; the value of the VCR by itself is very low, and thus, the agent gets the highest utility when it acquires both.

Definition 2. Two goods a, b are called *substitutable*, if their valuation as a set is lower than the sum of their individual valuations:

$$V(\{a, b\}) < V(\{a\}) + V(\{b\})$$

An example of this is the case of an agent wishing to buy a computer and not caring whether it is a Dell or Gateway manufactured unit. If the agent bids and gets both, its utility is only slightly higher than buying just one of the two computers.

Similar problems arise when the agent bids for required resources sequentially. In this case, the agent also needs to be careful in order to get what it needs, however once some auctions have closed, an agents' strategy becomes easier to compute. One of the most convincing approaches found in the literature for this problem is presented in [6]; the authors develop a dynamic programming model for agents to compute bidding policies based on estimated distributions over winning bids, which are represented with a fully observable MDP. In this way, agents limit their exposure from earlier bids that cannot be retracted.

The alternative mechanism for selling resources to agents with combinatorial preferences is through a combinatorial auction. In this case, each agent bids for resource bundles, and therefore is guaranteed to get exactly what it wants if it wins. This reduces the computation needed in order to determine an agent's bid, but increases the computation needed on the part of the auctioneer in order to determine the bids that maximize its profit. Another issue in this case is the problem of preference elicitation, as the agents need to submit their utilities for several sets of items (and the total number of possible sets is exponential in the number of goods).

We concentrate our attention to the problem of several different parallel auctions, rather than the combinatorial auction variant, because these are more readily available to the general public (e.g. eBay), and the design of an agent that would bid intelligently in this case. It should be noted that simple agents are available for bidding on eBay, like the proxy bidder that eBay itself provides, and which is a simple program that bids progressively up to a predetermined value, or the sniping program available from eSnipe.com, which places a proxy bid on eBay a few seconds before an auction would close. However, these types of agents are not the intelligent agents that we are interested in designing. It is prudent to give, at this point, a formal definition for what constitutes an agent. This information is taken primarily from [102].

Definition 3. An agent is a computer system that is situated in some environment, which is capable of autonomous action in this environment in order to meet its design objectives.

This definition does not guarantee, though, that the agent will be intelligent. A simple UNIX demon would also be considered an agent. An *intelligent agent* should be capable of flexible autonomous action in order to meet its design objectives, where flexibility means:

- *reactivity*: an intelligent agent should be able to perceive its environment, and respond in a timely manner to changes that occur in it;
- *pro-activeness*: an intelligent agent should be able to exhibit goal-directed behavior by taking the initiative in order to satisfy its design objectives;
- *social ability*: capable of interaction with other agents (some of which may be humans)

1.1 General Problem Setting

The general problem setting that we deal with in this dissertation involves several autonomous agents, which wish to trade commodities in order to acquire the goods that they need. There is a predefined time window during which the trades can take place (defining the duration of each “game”), after which each agent calculates the payoff to itself. The agents are not allowed to cooperate in any explicit way (even though implicit cooperation might arise from their behavior), and they are also assumed to be self interested. In particular, each agent i is trying to maximize its own utility function

$$U_i(\theta_i, C_i, t_i)$$

where θ_i is the type of the agent, that is a group of parameters which are selected randomly from a given distribution and influence the utility function,

C_i is the set of commodities that the agent owns, and

t_i is the net monetary transfer, that is the algebraic sum of payments for selling goods minus the cost of buying goods.

In most cases we can assume that the utility is *quasilinear*, i.e., linear in the monetary transfers t_i ; thus

$$U_i(\theta_i, C_i, t_i) = u_i(\theta_i, C_i) + t_i$$

This is a reasonable assumption for a trading agent, given that usually they are risk-neutral and their goal is to maximize their profit, hence their utility is linear with respect to monetary gain.

The utility function of each agent should include several complementary and substitutable goods in order for the game to be interesting. Otherwise, one can often find an equilibrium to the game analytically; e.g. in the case that the value of a set of goods is equal to the sum of the values of each good, the optimal strategy can be found by analyzing the procurement (or sale) of each item individually.

The mechanism used in order to exchange the various commodities is a number of different auctions, during which units of certain commodities are traded in exchange for monetary payments. This means that a single good is traded for money; we have not examined, in this dissertation, the possibility of bartering items, i.e., trading one good directly for another. We will assume that there is *no discriminatory pricing* in these auctions, which means that if two agents wish to buy the same good at the same time, they will have to make the same payment. We will also assume that *similar goods are sold in auctions with similar rules*.¹ Other than, that we allow the auctions to have a wide variety of rules, the most important of which are:

1. Agents may act as *buyers only*, *sellers only* or *both* in the auctions, so we can have *single-sided* and *double-sided auctions*. In case e.g. they act as buyers, then an external source would have to provide (input) goods into the system and remove money from it.
2. There can be a *finite* or an *infinite* number of units for each commodity. In the extreme case, only one unit exists.
3. Auctions can clear *continuously*, *several times* or *only once*. The first case means that trades can take place at any time, while the last that trades takes place exactly once, when the auction closes.
4. Each auction can close at *a known preset time*, at *one of several possible closing times*, or at *unspecified times* (e.g. determined by a period of inactivity or some random parameters).

¹We make these assumptions because this usually makes reasoning about bidding strategies easier. However, there are several cases in which our methodology would work even if discriminatory pricing exists or similar goods are sold in auctions with different rules.

5. Clearing prices can be determined *only by the bids of the agents or by external parameters* as well (e.g. set by an external seller). Pricing in the first case, in particular, can follow any pricing scheme (e.g. N^{th} , $(N + 1)^{\text{th}}$ highest bid, where N is the number of identical goods for sale in the auction).

Definition 4. A bid b_i placed at an auction that sells a particular commodity is a tuple (q_i, p_i, T_i) where:

q_i is the quantity of the commodity to be bought (if it's negative, this indicates a desire to sell),

p_i is the price at which to buy (or sell), and

T_i is the time that the bid should be placed.

The agent must determine what set \mathcal{B} of bids to place at each game. Therefore, it must determine all of the bids (quantity, price and time of bid placement) that it wishes to submit for all of the possible auctions in order to maximize its utility at the end.

1.2 Motivating Example: The TAC Game

To make the presentation more concrete, we now discuss an example domain where our methodology is needed. We use this domain throughout the thesis.

In the Trading Agent Competition, an autonomous trading agent competes against seven other agents in each game. Each agent is a travel agent with the goal of arranging a trip to Tampa for eight customers, and the trips can be scheduled over a period of five days. To do so, it must purchase plane tickets, hotel rooms, and entertainment tickets. This is done in simultaneous online auctions. The agents send bids to the central server² and are informed about price quotes and transaction information. Each type of commodity (tickets, rooms) is sold in separate auctions. Each game lasts for a preset

²Originally the game was organized and run by the University of Michigan agent group; currently, it is run by SICS (Swedish Institute of Computer Science).

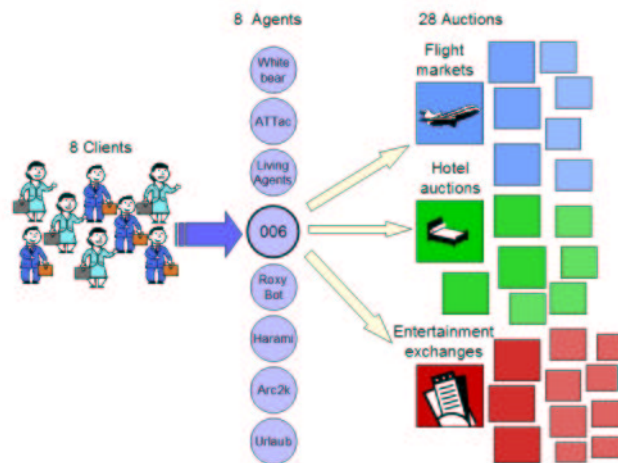


Figure 1.1: Illustration of the environment a TAC agent operates within. (taken from www.sics.se/tac)

amount of time: until the 2003 competition this was 12 minutes (720 seconds), and since 2004 it has been reduced to 9 minutes (540 seconds).³ Figure 1.1 illustrates the tasks of each individual agent.

There is only one flight per day each way with an unbounded number of tickets offered by the airline company, which is represented in the marketplace by an agent that sets prices according to a stochastic function. Tickets for these flights are sold in single seller, continuous one-sided auctions, with one auction for each day and direction (in or out). These auctions close at the end of the game. Since all clients must stay at least one night in Tampa, there will be no inflights on the last day, nor outflights on the first day. The auctions will clear continuously and any buying bids that are at least as high as the current asking price, which is set by the airline agent, will match immediately at the asking price. Once purchased, flight tickets cannot be resold or cancelled.

There are two hotels in Tampa: the Tampa Towers and Shoreline Shanties. The

³The first three minutes of the game were removed, since there was little actual trading that took place during that time.

Tampa Towers is cleaner, more comfortable, more convenient and all-around a nicer place to stay. For this reason, we would expect the Towers to cost more, because of the clients' preference for this hotel and the fact that each client is willing to pay an extra sum (bonus) to the trading agent if her booking is at this hotel. Note that a client cannot move between hotels during their trip and (s)he must be given a hotel room for every night. There are $m = 16$ rooms available each night at each of the two hotels. Since clients need hotels only from the night of their arrival and through the night before their departure, no hotels will be available (or needed) on the last day. Rooms for each of these days are sold by each hotel in separate, ascending, open-cry, multi-unit, 16th-price auctions (standard English auctions), except they close at randomly determined times in the last 8 minutes of each game. More specifically, exactly 8 minutes before the end of the game (that is 4:00 for TAC01-TAC03 and 1:00 for TAC04-TAC05) a randomly selected auction closes, then another one a minute later, etc., until exactly one minute before the end of the game (that is 11:00 for TAC01- TAC03 and 8:00 for TAC04-TAC05), the last one closes. A hotel auction clears and matches bids only once, when it closes. Price quotes are only generated once per minute, on the minute. Agents may submit buy bids, but not sell bids. Only the hotel owners may submit sell bids. The hotel owners submit bids to provide up to 16 rooms at each hotel type on each night, for a minimum price of \$0. No prior knowledge of the closing order exists and agents may not resell rooms they bought, or subtract bids.⁴ In the case of a tie between bids, the earliest bids are awarded the rooms.

There are $ET = 3$ different entertainment events. Tickets for these events are traded (bought and sold) among the agents in continuous double auctions (stock market type auctions) that close when the game ends. The travel agents can act both as buyers and

⁴The agent must always offer to buy in any subsequent bids to the auction at least the number of rooms that it is currently winning (i.e. the rooms that it would win if the auction closed at this point) at a price higher than the current price Q .

sellers in these markets. There is one auction for each event-night combination. Bids match as soon as possible, because the auctions clear continuously; so when a buyer is willing to pay more than the price a seller asks for, a trade takes place. There are a total of 8 tickets available for each event type on each day and each agent starts with an endowment of 12 random tickets from these. The initial tickets are the only available in the game, and therefore, the only way to procure a particular ticket is to trade with an agent who possesses it. Price quotes are issued immediately in response to new bids that are not matched. The price quote is specified as the bid and ask price. The bid price is the price of the highest standing buy bid. The ask price is the price of the lowest standing sell point. As with the hotels, a client cannot use an entertainment ticket on the day of departure.

Each customer i has preferences for specific dates, staying at the good hotel and attending entertainment events. The more accurately the assigned trip matches these preferences, the higher the monetary value of the trip to the customer, and the higher the profit of the agent. The total utility of the agent is in fact, the sum of the money that the trips are worth to their customers. The preferences of the customers are also those of the agent. For more details on the actual formula of this utility, see Chapter 4.

1.2.1 The TAC Game as a Competitive Benchmark

The TAC game encapsulates most of the issues of the general problem presented in section 1.1, and is thus, an appropriate test-bed for evaluating our agent design.

- Each auction has rules which cover the various options discussed in the previous section: some auctions are *single-sided* and others *double-sided*, some offer a *finite* and some an *infinite* number of identical goods, some clear *continuously* and others *only once*, some close at *preset times* and some at *random times*, some auctions' clearing prices are determined *by the agents* and others *by outside sellers*.

- There are 28 auctions running in parallel (and, in fact, our strategies and methodology scales well and would also work for a larger TAC game with many more auctions). This setting is far too complex for current analytical methods used to derive equilibrium (or optimal) strategies.
- A number of different tradeoffs are present in this game, which makes the determination of an appropriate bidding strategy a difficult design problem. For a detailed analysis of these tradeoffs, see Chapter 5.
- The TAC setting is designed to model a realistic marketplace setting, as might be encountered by, for example, a travel agent. It also includes several complementary and substitutable goods and a complex utility function.

It is also the most comprehensive and competitive benchmark available for agents and multi-agents research, in general. In earlier work, e.g. [1, 34, 42, 72], ideas about agent design were tested in smaller market games that the researchers designed themselves. The Trading Agent Competition (TAC) was designed and organized by a group of researchers at the University of Michigan [101] in order to allow researchers to spend their time on research, rather than implementing markets. It also provides a common market scenario to compare strategies and agent designs. It is a challenging benchmark domain which incorporates several elements found in real marketplaces. In [32, 7], it is shown that the utility maximization problem faced by a TAC agent is isomorphic (under certain conditions) to variants of the winner determination problem in combinatorial auctions.

1.3 Thesis Statement and Contributions

In this thesis, we provide a principled design strategy for an agent with combinatorial preferences, participating in a large number of parallel auctions with different rules. The contributions of this thesis are:

- We provide a *novel methodology* accompanied by a corresponding agent architecture for this problem. Our approach relies on decomposing the overall auction problem into a number of manageable sub-problems. The main benefit of such a problem decomposition with a restricted flow of information between the various components is that we can fully analyze each individual component. Furthermore, the overall optimization problem that the agent faces becomes relatively easy to solve. Despite initial skepticism from some of our peers that decomposing the problem to such a degree would severely degrade the overall performance of the agent, due to the obvious inter-dependencies between the various components that our methodology suppresses, it is shown that this methodology in fact leads to extremely effective and robust agents.
- We design all modules of our agent with *scalability, speed, and adaptability* in mind. We show that the use of sub-optimal (approximate) algorithms to various elements of the agent does not affect its performance. We also base a very limited part of the agent on learning the behavior of other agents, and thus the agent should be able to perform competitively no matter what other agents it faces, despite rather inaccurate predictions of closing prices for various auctions.
- We developed a *systematic way for generating candidate strategies for the various bidding components of the agent*. This approach also allows us to optimize component strategies in a principled manner, in contrast to the mostly “heuristic” approaches, prevalent in the literature. We can also generate enough strategies to cover, to a sufficient degree, the entire strategy space. Domain knowledge can still be incorporated to some extent, and in fact, it is desirable in order to concentrate the experimentation on the part of the strategy space where the most efficient candidate strategies lie.

- We provide *novel Bayes-Nash equilibria* for the case of an auction with a set of multiple possible closing times, one of which is selected at random. We extended our methodology to incorporate candidate strategies based on such equilibrium strategies.
- We propose *a systematic method for exploring the entire strategy space* (for all auctions) and determining the best combination of strategies for the agent. We limit, to some extent, the large number of experiments that are needed in order to accomplish this. We also provide *a complete set of experiments* for determining a very competitive agent for both versions of the TAC game.

During the development of our methodology and the experimentation, we learned some important lessons:

- The decomposition of the problem, which is done according to our methodology, is a major benefit of our approach, as it makes it feasible to analyze each component independently and determine possible strategies. Furthermore by rigorously exploring the strategy space through experimentation, we are able to select the best strategy for the agent.⁵
- We have also observed that agents, that depend too much on learning the bidding patterns of other agents, can be quite inefficient, if those patterns change. Our agents, on the other hand, are able to participate successfully in any game, against unknown combinations of other agents. This is because their strategy is selected so as to perform well *no matter what* strategies the other agents use. In fact, we intentionally do not allow the agent to learn information specific to particular

⁵In the current TAC competition (TAC-05), another strong agent, Walverine, now also explores the strategy space in order to find the best strategy.

agents, but we do allow it to learn general information, that is not specific to a particular combination of agents.

- One final lesson that we learned is that the components of the agent need not be optimal as long as they are reasonably efficient. For this reason we design them to be fast, scalable and adaptive. One example of this is that we decided to use a non-optimal solution to the optimization problem, which is solved by the optimizer, in order to make it fast, scalable and non-domain specific. We have observed that the agent is *better than the sum of its individual components*, as long as those components are designed to be efficient; they don't have to be the best components possible. One other design decision that helps in this direction is that our agent is completely modular, so individual components can be replaced by other modules with relative ease. One reason efficiency appears to be more important than optimality — provided one can obtain reasonable approximate solutions fast — is that the continuously changing information available to the agents during the game makes rapid adaptation and flexibility more important than optimality at any given point in time.

If we were to summarize our objective in one statement it would be this:

By fully decomposing the problem faced by an agent participating in several auctions, it becomes feasible to analyze each component individually; generate strategies for it in a systematic way; and through rigorous experimentation determine the design of a competent, flexible and scalable agent, despite any shortcomings of individual agent components. We show that this approach leads to the design and implementation of highly effective agents. Our agent WhiteBear, developed using the design strategy described in this thesis, was the winner of the 2002 and the 2004 Trading Agent Competi-

tion; and, was the most successful agent in the competition overall.

1.4 Guide to the Thesis

The rest of the thesis structure as is follows:

- **Chapter 2 - Background and Overview.** In this chapter, we present the basic knowledge from game theory and auction theory necessary to understand our work. We also discuss work in various areas related to agent design, auctions, and bidding strategies.
- **Chapter 3 - Methodology and Agent Architecture.** In this chapter, we present our methodology for dealing with the general problem, which was presented in section 1.1. We describe our agent architecture, the methodology for generating strategies and the experimental process through which the strategies are evaluated.
- **Chapter 4 - Optimizer.** In this chapter, we formulate the optimization problem faced by our agent and the work we did on the optimizer component of our TAC agent.
- **Chapter 5 - Generating Strategies and Dealing with Tradeoffs.** In this chapter, we elaborate on how strategies for the various strategic tradeoffs that the agent faces may be generated in a systematic way. We present the strategies generated for the various TAC games.
- **Chapter 6 - Using Equilibrium Analysis to Generate Strategies.** In this chapter we present several novel Bayes-Nash equilibria for m^{th} price auctions when the closing time is randomly selected from a set of possible closing times. We compute these equilibria under various assumptions. We then proceed to describe

how this theoretical work is applied, in order to generate a bidding strategy for the hotel room bidding component of our agent.

- **Chapter 7 - Experiments.** In this chapter we apply the experimental setup that is described by our methodology and analyze the performance of each strategic combination in the face of opposition from various agent mixtures. We provide complete sets of experiments on the strategies provided in chapters 5 and 6. We also examine the performance of our agent compared to agents used by several other teams in the TAC competition.
- **Chapter 8 - Conclusions.** In this chapter, we conclude our work with a review of our contributions along with a brief discussion of future work that extends the results of this thesis.

Much of the work in this thesis is presented in [85, 86, 87, 88, 89, 90, 91].

Chapter 2

Background and Overview

2.1 Game Theory and Solution Concepts

Game theory studies the strategic interactions in systems of self-interested agents. This section provides some basic game-theoretic solution concepts from non-cooperative game theory, that we use in our work. For more information on Game Theory there is a large number of books and other literature, e.g. [20] and [5]. [68] also has a very detailed introduction to game theory, as well as a much more detailed one about mechanism design and combinatorial auctions.

2.1.1 Basic Definitions

It is useful to introduce the idea of the type of an agent, which determines the preferences of an agent over different outcomes of a game. Let $\theta_i \in \Theta_i$ denote the type of agent i , from a set of possible types Θ_i . An agent's preferences over various outcomes can be expressed in terms of a utility function, that is parameterized on the type θ of the agent.

The fundamental concept of agent choice in game theory is expressed as a strategy. A strategy can loosely be defined as:

Definition 5. A strategy is a complete contingent plan, or decision rule, that defines the action an agent will select in every distinguishable state of the world.

Let $s_i(\theta_i) \in \Sigma_i$ denote the strategy of agent i given type θ_i , where Σ_i is the set of all possible strategies available to an agent. Sometimes the conditioning on an agent's type is left implicit, and in that case, we simply write s_i for the strategy selected by agent i given its type. In addition to *pure*, or deterministic strategies, agent strategies can also be *mixed*, or stochastic. A mixed strategy, written $\sigma_i \in \Delta(\Sigma_i)$, defines a probability

distribution over pure strategies. For example, if this probability distribution is 40% and 60%, for an agent i with available strategies s_1 and s_2 , then this agent executes strategy s_1 with probability 60% and strategy s_2 with probability 40%. In the case of an auction, the possible strategies define what formula each agent uses in order to calculate its bid.

A game would normally define the set of actions available to an agent (e.g. valid bids, legal moves, etc.), and a mapping from agent strategies to an outcome and a corresponding utility. Therefore, in each game, we can express an agent's utility as a function of the strategies of all the agents to capture the essential concept of strategic interdependence.

Definition 6. Let $u_i(s_1, \dots, s_N, \theta_i)$ denote the utility of agent i at the outcome of the game, given preferences θ_i and strategy profile $s = (s_1, \dots, s_N)$, where N is the total number of agents and s_i is the strategy selected by agent i .

For example, in an auction that sells one item, an agent i with strategy s_i that instructs the agent to place bids at price $v_i = g_i(u'_i)$, where u'_i is the agent's inherent value for the item in question and depends on the type θ_i of the agent. The utility of the agent is

$$u_i(s_i, s_{-i}, \theta_i) = u'_i - g_i(u'_i), \text{ if it wins, that is if its bid is the highest,} \\ \text{otherwise } u_i(s_i, s_{-i}, \theta_i) = 0.$$

Note that we use the notation s_{-i} , to indicate the strategy profile of all the agents except of agent i . Thus $s_{-i} = (s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_N)$. Let us also use the notation θ_{-i} to denote the set of types of all agents except i .

The basic model of agent rationality in game theory is that of an *expected utility maximizer*. An agent will select a strategy that maximizes its expected utility, given its preferences θ_i over outcomes, beliefs about the strategies of other agents, and structure of the game.

2.1.2 Solution Concepts

Game theory provides a number of solution concepts to compute the outcome of a game with self-interested agents, given assumptions about agent preferences, rationality, and information available to agents about each other. We are going to discuss the three most commonly used solution concepts: dominant strategies, Nash equilibrium and Bayes-Nash equilibrium.

The strongest solution concept in game theory is the *dominant strategy* equilibrium. In a dominant strategy equilibrium, every agent has the same utility-maximizing strategy, no matter what strategies the other agents use.

Definition 7. Strategy s_i is a dominant strategy, if it maximizes the agent's expected utility for all possible strategies of other agents. This means that for $\forall s'_i \neq s_i, s_{-i} \in \Sigma_{-i}$ it is:

$$u_i(s_i, s_{-i}, \theta_i) \geq u_i(s'_i, s_{-i}, \theta_i)$$

In other words, a strategy s_i is a dominant strategy for an agent with type (preferences) θ_i , if it maximizes expected utility, whatever the strategies of other agents. If this equation holds with equality, then the strategy is called *weakly dominant*.

For example, in a sealed-bid second-price (Vickrey auction), the item is sold to the highest bidder for the second-highest price. If the item is valued by agent i for u'_i , bidding strategy $g_i(u'_i) = u'_i$ is a dominant strategy for this agent. For more details on this and other auctions see the next section.

This definition can be extended to the case of mixed strategies easily enough:

Definition 8. Strategy s_i is a dominant strategy in mixed strategies, if it maximizes the agent's expected utility for all possible strategies of other agents. This means that for $\forall \sigma'_i \in \Delta(\Sigma_i) : \sigma'_i \neq \sigma_i$, and for $\forall s_{-i} \in \Sigma_{-i}$ it is:

$$u_i(\sigma_i, s_{-i}, \theta_i) \geq u_i(\sigma'_i, s_{-i}, \theta_i)$$

Note that it is sufficient that the other agents' strategies are selected from their set of pure strategies, since, if they use a mixed strategy, the utility of agent i is the expectation of the utility conditional on each of the strategies that make up the mixed strategy of every other agent.

Dominant-strategy equilibrium is a very robust solution concept, because it makes no assumptions about the information available to agents about each other, and does not require an agent to believe that other agents will behave rationally to select its own optimal strategy. However, the drawback is that it is not guaranteed that a game will have a dominant strategy.

Another solution concept, probably the best-known one, is the Nash equilibrium [64, 63], which states that in equilibrium every agent will select a utility-maximizing strategy given the strategy of every other agent.

Definition 9. A strategy profile $s = (s_1, \dots, s_N)$ is in Nash equilibrium, if every agent maximizes its expected utility, assuming that the strategies of the other agents do not change. Thus, for $\forall i$:

$$u_i(s_i(\theta), s_{-i}(\theta_{-i}), \theta_i) \geq u_i(s'_i(\theta), s_{-i}(\theta_{-i}), \theta_i), \forall s'_i \neq s_i, s_{-i} \in \Sigma_{-i}$$

In other words, every agent maximizes its utility with strategy s_i , given its preferences and the strategy of every other agent. This definition can be extended in a straightforward way to include mixed strategies.

Although the Nash solution concept is fundamental to game theory, it makes very strong assumptions; it assumes that the other agents will not deviate from the equilibrium strategy, which is by no means guaranteed. It also loses power in games with multiple equilibria. To play a Nash equilibrium in a game, every agent must have perfect information (and know every other agent has the same information, etc., i.e., common knowledge) about the preferences of every other agent, agent rationality must also be

common knowledge, and agents must all select the same Nash equilibrium.

The last solution concept is the Bayes-Nash equilibrium, which is an extension of the Nash equilibrium. This is mainly due to Harsanyi [36]. In a Bayes-Nash equilibrium, every agent is assumed to share a common prior about the distribution of agent types, $F(\theta)$, such that for any particular game, the agent profiles are distributed according to $F(\theta)$. In equilibrium, every agent selects a strategy to maximize expected utility in equilibrium with expected-utility maximizing strategies of other agents.

Definition 10. A strategy profile $s = (s_1, \dots, s_N)$ constitutes a Bayes-Nash equilibrium, if for every agent i and all preferences $\theta_i \in \Theta_i$

$$E\{u_i(s_i(\theta), s_{-i}(\theta_{-i}), \theta_i) | \theta_{-i}\} \geq E\{u_i(s'_i(\theta), s_{-i}(\theta_{-i}), \theta_i) | \theta_{-i}\}, \forall s'_i \neq s_i, s_{-i} \in \Sigma_{-i}$$

Comparing the Bayes-Nash equilibrium with the Nash equilibrium, the key difference is that agent i does not necessarily play a best-response to the actual strategies of the other agents. Its strategy $s_i(\theta_i)$ must be a best-response to the expected outcome given the other agents' strategy distributions and the distribution $F(\theta)$ from which the preferences of the other agents are drawn.

Bayes-Nash makes more reasonable assumptions about agent information than Nash, but is a weaker solution concept than dominant strategy equilibrium. In fact, any dominant strategy will be a Nash equilibrium by default; the opposite does not hold.

Algorithms for computing Nash Equilibria

Computing Nash equilibria in general is a hard task; in fact, if additional properties are required (e.g. maximal support), this is known to be an NP-hard problem. However, there are some algorithms capable of computing them for 2-player and N -player games. For a survey on the topic see [93]. For 2-player games, the Lemke-Howson algorithm [50] was still the state of the art, until recently. This, and other algorithms,

are implemented in GAMBIT [60], which is a state of the art solver for finding Nash equilibria for games in matrix form. However, some better algorithms have since been proposed. For example in [70], the authors propose a search algorithm for finding Nash equilibria for both the 2-player and the N -player game. They use the fact that computing whether a Nash equilibrium with a particular support (i.e., particular pure strategies that are used in the equilibrium with non-zero probability) exists for each player is a relatively easy feasibility program. They design an algorithm that explores the space of support profiles using a backtracking procedure to instantiate the support for each player separately. After that, they check for actions in a player's support that are strictly dominated, given that the other agents will only play actions in their own supports and delete them. Experimental validation confirms that this algorithm outperforms older algorithms. The experimental data that they use are from GAMUT [67], which is a suite of game generators designed for testing game-theoretic algorithms, and particularly, algorithms for finding Nash equilibria. This suite includes a large number of different games that have been analyzed in the economics and the multi-agent systems literature.

In [57], the authors compute what they call an ϵ -approximate Nash equilibrium. This is an extension of the classical definition of the equilibrium that requires the equilibrium strategy to either be better than every other strategy given the opponent's strategy profile, or to be within ϵ of the best strategy. They show that this ϵ -approximate Nash equilibrium, can be computed in quasi-polynomial time.

2.2 Auction Theory

Auction theory analyzes protocols and agents' strategies in auctions. An auction consists typically of an auctioneer and potential bidders. Since all the agents involved, including the auctioneer, are self-interested, the auctioneer wishes to sell its items at the

highest possible price, while the agents wish to buy them at the lowest possible cost. In this section, we will discuss some of the most basic knowledge that is useful for understanding our own work. For more information on auctions, see [44], or [77]. In addition, there are also several review papers like [59] and [43].

There are different strategies for the various auctions depending on the assumptions made about the private valuations that agents have about the items being sold. In particular, we can have the following cases:

1. In *private value* actions, the value of the good depends only on the agent's own type θ . This is usually the case if the agent is going to keep the items and use them himself, but not resell them.
2. In *common value* actions, the value of the good depends entirely on the other agents' types and valuations. This is usually the case if the agent is going to buy the items and then resell them, in which case its profit depends on the amount it will get from other agents.
3. In *correlated value* actions,¹ the value of the good depends on the types of all agents. This is the intermediate case of the previous two cases. This case would be encountered if the agent is planning on either using the items itself, or reselling if the price is right.

In most of the examples that follow, we will assume private valuations.

There are various auction protocols used in practice even for selling a single item. The most common of these are:

1. *The English (or first-price open-cry) auction*. In this auction, each bidder is free to raise her bid at any point. When no bidder is willing to raise her bid, then the

¹These are sometimes referred to as common value auctions.

auction ends, and the agent who placed the highest bid wins the item at a price equal to her bid. The optimal strategy for each bidder is to keep increasing her bid by a small increment ϵ , until the bid reaches her private valuation, at which point she should stop bidding. For a small ϵ , and under a few other assumptions, this auction is strategically equivalent to the second-price sealed-bid auction.

2. *The Dutch (or descending) auction.* In this auction, the bidder starts at a high price and gradually reduces the bid until a bidder bids at some price and gets the item while paying that particular price. This auction is strategically equivalent to the first-price sealed-bid auction under a few assumptions.²
3. *The first-price sealed-bid auction.* In this auction, each bidder places a bid in secrecy without knowing the other agents' bids. The bids are revealed, and the highest bidder gets the item at a price equal to her bid. The dominant strategy for each bidder, in this case, under any assumptions is to bid less than her true valuation. Given the fact that we deal with extensions of this auction in our work, we give more information on equilibrium strategies under various assumptions in Section 2.2.1.
4. *The second-price sealed-bid (or Vickrey) auction.*[92] In this auction, each bidder places a bid in secrecy without knowing the other agents' bids. The bids are revealed and the highest bidder gets the item at a price equal to the second highest bid. The dominant strategy for each bidder in this case is to bid her true valuation, if this is known. A quick reason for this is that there is no reason to bid more than her true value, since otherwise she might get an item at a price higher than her valuation. In addition, there is no reason to bid below her valuation, because if she wins, the price she pays depends on the price of the next highest bid, and if

²Sometimes people call a multi-unit English auction Dutch erroneously, e.g. at eBay.

she were to do so, she might miss an opportunity to win and make a profit.

All these auctions give the same expected revenue to the auctioneer, despite the different rules and bidding strategies. This is a result of the *revenue equivalence* theorem. We present a brief version of it here that pertains only to the four auction types that we just presented, but for more details one may see [61].

Theorem 1. *Revenue Equivalence*

The four protocols we presented produce the same expected revenue to the auctioneer in private value auctions where the values are independently distributed, and bidders are risk-neutral.

If the bidders are risk-averse, then the first-price sealed-bid auction will generate more income than the second-price auction, because risk averse bidders will bid more in the former case than they would if they were risk-neutral. A bidder is called risk-neutral, if she is indifferent between any two events that have the same expected payoff. A bidder is called risk-averse, if she prefers to get a certain payoff to any lottery that has the same expected payoff; so a risk-averse bidder would prefer to get \$100 to a lottery that would give them a 10% chance at \$1000, whereas a risk-neutral bidder would be indifferent between these two payoffs.

2.2.1 First-price Sealed-Bid Auctions

We assume that N agents wish to buy 1 unit of a certain good each. An independent seller sells 1 unit of the desired good in a first price sealed-bid auction, i.e., the good is sold to the agents which submitted the highest bid at a price equal to its bid price. The agents have valuations (utilities) u_i which are i.i.d. with probability distribution $F(u)$. Each agent knows its own valuation and the distribution $F(u)$. In the case that the agents are risk-neutral (i.e., utility is equal to the monetary profit), then each agent i

with valuation u_i should bid (see for example [44])

$$g(u_i) = u_i - \frac{1}{(F(u_i))^{N-1}} \cdot \int_0^{u_i} (F(\omega))^{N-1} \cdot d\omega$$

and in the case that $F(u)$ is a uniform distribution between 0 and 1, then the equilibrium strategy is to bid

$$g(u_i) = \frac{N-1}{N} \cdot u_i$$

Therefore, the larger the number of bidders that participate in an auction is, the larger the expected revenue of the auctioneer. It should be noted that while in the second price auction, the risk attitude of the bidders does not change the equilibrium strategy in a first price auction it does: a risk-averse agent would wish to bid higher and thus increase the probability that it will get the item.

Collusion between agents

All the equilibrium strategies that were presented for the various auctions in this section, do not account for the possibility that the bidders might wish to collaborate if this will help their utility. In fact, it can be shown that if the bidders are allowed to collaborate, they would not wish to follow the equilibrium strategies presented earlier. One way to do this is to form a cartel, i.e., a group of agents, that would bid for the item at a reduced price and then the members of the cartel would split the items among themselves. In the general problem setting that we discussed, explicit collaboration between agents is strictly forbidden, so we did not have to deal with this problem. However, there is a growing literature on this subject, e.g. [56].

2.3 Combinatorial Auctions

In this section, we discuss Combinatorial Auctions and present the work done on this subject. As it was mentioned in the introduction, the winner determination problem of

combinatorial auctions is isomorphic (under certain assumptions) to the optimization problem in TAC, and in related problems where there are combinatorial preferences on the part of the agents. The combinatorial auction setting shifts the computational cost to the auctioneer, instead of the agents, since there are protocols (e.g. VCG) that allow the agents simply to submit their valuations, without need for any additional computation. For a survey of combinatorial auctions, see [14]. The thesis of Professor Leyton-Brown is mostly dealing with this problem as well, and is thus, a good general reference.[51]

Combinatorial auctions have been applied to certain real-world situations. The best known example of this is the FCC spectrum auctions, where bidders could desire licenses for multiple geographical regions at the same frequency band, while being indifferent to which particular band they receive [62]. Another example is presented in [8]. This is an auction for the right to use railroad segments, and a bidder desires a bundle of segments that connect two particular points; at the same time, there may be alternate paths between these points and the bidder needs only one.

2.3.1 Formal Definition

In a combinatorial auction, a seller is faced with a set of price offers for various bundles of goods, and her aim is to allocate the goods in a way that maximizes her revenue. The winner determination problem (WDP) is choosing the subset of bids that maximizes the seller's revenue, subject to the constraint that each good can be allocated once, at most. In this case, each agent pays the sum of her winning bids.

Definition 11. Let $C = \{c_1, c_2, \dots, c_m\}$ be a set of commodities to be sold, and let $B = \{b_1, b_2, \dots, b_n\}$ be a set of bids. Bid b_i is a pair $(p(b_i), g(b_i))$ where $p(b_i) \in \mathbb{R}^+$ is the price offer of bid b_i and $g(b_i) \subseteq C$ is the set of goods requested by b_i . For each bid b_i define an indicator variable x_i (which takes values 0 or 1) that encodes the inclusion or exclusion of bid b_i from the allocation.

The single-unit winner determination problem can be formulated as:

$$\max_{x_i} \sum_i x_i \cdot p(b_i)$$

with the constraints:

$$\sum_{i:c \in g(b_i)} x_i \leq 1, \forall c \in C$$

$$x_i \in \{0, 1\}, \forall i$$

In the case that there are multiple identical units of each commodity for sale, let us define $q(c_i)$ as the number of units of commodity c_i . The multi-unit winner determination problem can be formulated as:

$$\max_{x_i} \sum_i x_i \cdot p(b_i)$$

with the constraints:

$$\sum_{i:c \in g(b_i)} x_i \leq q(c), \forall c \in C$$

$$x_i \in \{0, 1\}, \forall i$$

It should be noted that the definition only allows bidders to describe complementarities in their valuations. However, it does not explicitly allow the expression of substitutabilities. To alleviate this problem, in [21] the authors propose the use of “dummy goods”. This means that the agents can insert non-existent commodities in their bids; any bids that should be mutually exclusive, because the commodities in them are substitutable, ask for the same dummy commodity from the auctioneer. Since any dummy commodity c_{dummy} is assumed to have a quantity $q(c_{dummy}) = 1$, the auctioneer can only satisfy one of the bids at most. More generally, it is possible to introduce n -unit dummy commodities to enforce the condition that no more than n of a set of bids may be allocated.

2.3.2 Solving the Winner Determination Problem

The WDP is equivalent to weighted set-packing and is, therefore, NP-hard even in its single-unit variant.[75] Furthermore, it is known that the WDP is inapproximable within any constant factor.[76] However, there has been work that identifies subcases of the winner determination problems, which are solvable in polynomial time (e.g. bids contain no more than two goods).[75, 65, 83, 14] Another case solvable in polynomial time, is when the commodities are infinitely divisible; then the problem becomes a linear constraint problem (without the requirement of integrality constraint), which can be solved in polynomial time with any number of available linear programming packages (e.g. CPLEX).

Despite the fact that this problem cannot be approximated with guarantees, there has been work on approximate solutions, most of which use some sort of local search. For some examples see [40] and [105]; these algorithms are shown to perform well in most experiments. There has also been work on alternative economic mechanisms that are built around approximation algorithms to this problem, e.g. [66] and [49].

There has also been a substantial amount of work on optimal solutions to this problem. In most of the early cases, the algorithms involve a search (usually “branch-and-bound” search) with pruning, while branching on the different commodities, e.g. CASS.[55] Later Tuomas Sandholm reported a better performance by branching on the available bids, which is the CABOB algorithm.[79] These algorithms do a number of tricks in order to reduce the solution time, e.g. reducing the number of bids, by removing dominated bids³, or caching parts of the search space. There has also been work on other approaches, e.g. primal-dual methods, which continuously compute tighter bounds until they find the actual solution. For other work on optimal solutions see

³A bid is called dominated, if a combination of other bids offer a better price for a subset of the original bid.

also [21, 76, 28, 65, 78, 29]. Some other interesting work, which has been done by the multi-agent group of Stanford University, explores techniques that have been widely used in Constraint Programming problems. They create portfolios of algorithms and learn through experimentation, and learning, which of the available solvers is likely to perform best in each case. Using this information, they can make an intelligent decision on which algorithm (or algorithms) to use in order to solve each case in the minimum expected time. For more information on this, see [53, 52]. Furthermore, they also provide test cases that can be used for testing of combinatorial auctions winner determination algorithms, which they call CATS [54]. Other interesting combinatorial auctions, which are variations of this are presented in [69, 103].

A problem that exists in this setting is the method in which the agents submit their preferences. This problem is usually called “preference elicitation”. In the worst case, each agent should submit its valuation for all possible subsets of the set of commodities, which is exponential in the number of commodities. For some of the work done to alleviate this problem, see [41, 11].

This mechanism assumes that the payments are equal to the submitted bids. However, this forces agents to compute a bid that would maximize their profit. The most celebrated mechanism in game theory concerning combinatorial auctions is the Vickrey-Clarke-Groves mechanism. For more information see [58] or [84].

Definition 12. Vickrey-Clarke-Groves mechanism

Let B_i be the set of all bids $b_k \in B$, that were submitted by agent i . Let $A = \{i : x_i = 1\}$ be the allocation of the optimal solution in the winner determination problem with input the set of bids B , and let A_{-i} be the optimal allocation in the case that agent i does not participate.

The auctioneer allocates the commodities according to allocation A , and each agent i

must pay:

$$t_i = \sum_{j \neq i} \left(\sum_{k: k \in B_j \wedge x_k \in A_{-i}} p(b_k) \right) - \sum_{j \neq i} \left(\sum_{k: k \in B_j \wedge x_k \in A} p(b_k) \right)$$

This means that each agent pays to the auctioneers the difference between what the auctioneer would make if this agent did not participate, minus what it would make from all the other agents in the case that it does participate.

This mechanism is incentive-compatible, meaning that is a dominant strategy for the agents to submit their true utilities for each bundle of commodities, only if the winner determination problem can be solved to optimality, and can give rise to arbitrarily bad outcomes when agents suspect that there is any possibility that any non-optimal solution to the WDP will be used.[66]

2.4 Multiple Auctions

In this section, we present several attempts to design sophisticated and efficient bidding strategies for agents participating in multiple (online) auctions.

2.4.1 Single-Sided Auctions

In the vast majority of the papers referenced in this section, the analysis is for substitutable goods, and in particular, for the case that the same good may be purchased from several different auctions. In addition, the simulation is based on rather simple cases of multiple auctions with the same rules in most cases.

In [42], the authors propose BiddingBot, which is a multi-agent system that supports users in attending, monitoring, and bidding in multiple auctions through a process called co-operative bidding. The whole system is composed of one leader agent, and several bidder agents, where the leader agent acts as the coordinator of the whole bidding process. Several messages are exchanged between the user, the leader agent, and the bidder

agents. However, this system is not autonomous as the user must make the final decision. It might be appropriate, though, in order to facilitate the introduction of agents to the general public, where there seems to be an amount of discomfort about relocating the complete decision process to an agent, i.e., a program, whose strategy the average user does not understand.

In [72, 71], an algorithm is proposed for the design of agents that participate in multiple, simultaneous English auctions. They propose a coordination algorithm to be used in the environment, where all the auctions terminate simultaneously, and a learning method to tackle auctions that are terminating at different times. This work is not applicable in an environment where multiple auctions with multiple protocols are used, as it is designed for the case of multiple English auctions.

In [33, 34] the authors present and analyze the behavior of several strategies, i.e., myopic, based on Q-learning, based on a simple Nash equilibrium etc., when the shopbot wishes to buy a certain item or certain items, from a number of available online auctions. This is very interesting work from the point-of-view that the different strategies are tested against each other, and conclusions are drawn from their performance. In addition, it shows that equilibrium strategies can do a decent job in practice, even if other strategies outperform them.

In [1], the authors present the design of an autonomous trading agent that participates across multiple online auctions (English, “Dutch”, and Vickrey auctions) in order to get one or more particular items. The decision on how much to bid in the auctions, in which the agent decides to participate, is made based on a series of “heuristic” tactics and strategies. These take into account the number of auctions left, the time left, and other factors. The bidding algorithm that they proposed was evaluated in a simulated marketplace environment. What is rather interesting about this paper is that their heuristic strategy displays a number of clever features; e.g. the prices increase with the

passage of time, and the decrease in the number of auctions remaining, which is a simple strategy that works well even in the more complicated problems presented in this dissertation.

Other approaches are based on beliefs that the agents have about other participating agents. In [25], the authors propose the Recursive Modeling Method. According to this, an agent makes decisions based on what it thinks the other agents are likely to do, and what the other agents think about the agent's beliefs and so on. In practice though, little or no information might be available for the agent, and this limits the application of this method.

In [22, 23] the strategies are considered as a decision made under uncertainty. They model a buyer agent's behavior; they do this by generating a possibility distribution based on previous similar market situations that their agent has observed.

2.4.2 Double-Sided Auctions

In this section, we summarize some of the work done on continuous double-sided auctions, a.k.a. stock-market type auctions. In most cases, the analysis is done on a single auction, but since it is usually assumed that there are no interdependences between goods, the approaches generalize trivially to multiple auctions.

Probably the best-known, and also most simplistic, approach to this problem is the Zero-Intelligence (ZI) strategy.[26] This strategy essentially ignores the state of the market (i.e., the prices it has observed) when forming its bid, both for buying and selling, and submits a random value drawn from a uniform distribution. Despite its simplicity, if all agents participating in the market adopt this strategy, the market efficiency is close to optimal. An improvement of this strategy is the Zero-Intelligence Plus (ZIP) strategy.[10] Bidding slightly more intelligently than the ZI strategy, the ZIP strategy places bids at its valuation for the good, plus or minus a certain margin. This margin represents the

profit of the agent, and is adapted to the prevailing market conditions through a learning mechanism, so that the trader can submit sell and buy bids that remain competitive. In experimentation, this agent outperforms the ZI strategy.

In [94] the authors propose what they call a “risk-based” strategy. Essentially, the agent has an adaptive layer that learns the risk factor of the market, and thus, computes a competitive equilibrium price. This is done by using a set of updated rules, which move the price up or down depending on the bids that are placed in the market, and the transactions that occur. This price is then used in the bidding component of the agent, in order to place bids and accept offers that offer at least a small margin of profit over the computed competitive equilibrium price. They show that their agent outperforms both the ZI and ZIP agents in experimentation. Other approaches involve using a belief function for price formation, called the GD strategy [24], and using fuzzy logic to decide on the buy and sell bids, based on the median of the previous transaction prices, called the FL strategy.[39]

A more real-life scenario is examined in [104, 18]. The authors examine automated stock-trading agents, which they test in the context of a real-data market trading simulator. The first agent they propose devises a market-making strategy exploiting market volatility without predicting the exact direction of the stock price movement. In particular, the agent places a pair of buy and sell bids for each stock it wishes to trade, at prices which are respectively lower and higher than the current price by a certain margin. The second agent uses what the authors call the “reverse” strategy. It might seem natural to buy when the market is on the rise, and sell when it’s on the decline, but the second agent does exactly the opposite: it buys when the prices are dropping and sells when they are rising. Both strategies work well, under the assumption that there are no prolonged periods of rising or falling prices in the stocks. If this is not the case, then the strategies will perform poorly, because both strategies will need to buy or sell at disadvantageous

prices in order to cover their previous trades.

An experiment that is interesting, not so much from the point of agent strategies for the continuous double-sided auction, but rather for validating our belief that agents are necessary in order for humans to be competitive in electronic marketplaces is presented in [13]. A number of controlled experiments, where human traders interact with software agents were performed, and the authors observed that the agents consistently obtained higher profits than their human counterparts.

2.4.3 Sequential Auctions

When the agent bids for required resources sequentially, it needs to be careful in order to get what it needs. However, once some auctions have closed, an agents' strategy becomes easier to compute. One of the most convincing approaches found in the literature for this problem is presented in [6]; the authors develop a dynamic programming model for agents to compute bidding policies based on estimated distributions over winning bids, which are represented with a fully observable MDP. In this way, agents limit their exposure from earlier bids that cannot be retracted.

An alternate approach is available, if one does not wish to analyze the bidding patterns of its competitors; it can approximate it by taking the expected utility from the next auctions, if the opponents bid according to the optimal (equilibrium) strategy. In [17, 15, 16], the authors analyze the case that there are several objects to be sold in sequential auctions. In most cases, they examine the case that similar objects are sold in each of the sequential auctions, and that each bidder is interested in acquiring one of them. They give equilibrium strategies for the bidders under three different auction types, that is first-price sealed-bid, second-price and English auction, and they prove something similar to the revenue equivalence theorem, i.e., that the various auctions give the same expected revenue to the auctioneer under certain assumption (e.g. risk-

neutral agents). In addition, they compute how the auctioneer can increase his revenue, by selecting carefully the order with which items are auctioned off, and other issues, like increasing efficiency by having the auctioneer reveal his private information about the object being sold. This work significantly extends previous work that was done on this setting, e.g. in [27], where the items being sold had either private valuations for each agent, or common valuation, but not both.

2.4.4 Strategies Used by TAC agents

This section describes, very briefly, several different strategies used by various TAC agents. This information is primarily taken from [31], which is a survey paper of the different strategies that agents used in TAC 2002, which were contributed to, in large part, by the respective research teams. For a survey of the first competition see [30]. This information is provided to complement the algorithms and agents that have been presented so far in Section 2.4, and to show the various “heuristic” strategies that agents used. It should be noted that each agent normally implements one, or just a handful, of very similar strategies, which even though they are, in a lot of cases, based on general techniques (e.g. learning, POMDPs, etc.), they only cover a very small part of the strategy space. This is to contrast with our approach, which selects a large number of candidate strategies and experiments with them in order to determine the best. The only other agent that does something similar to this is WALVERINE, which was developed by the University of Michigan multi-agent research group.[99, 100]

This agent aims to bid optimally, based on a competitive equilibrium analysis of the TAC travel economy. WALVERINE predicts hotel prices by calculating the *Walrasian competitive equilibrium* for the game. This is presented further in Chapter 4, as we evaluated this feature in our agent. By sampling outlier values around the predicted prices, the prediction is used to compute an “outlier hedged” set of optimal travel dates

(hence, flight purchases) and marginal values for the available hotel rooms. During each hotel-bidding round, WALVERINE constructs bids that maximize expected value, taking into account its marginal value of each room unit, the probability of winning for a given bid, the probability that this bid will actually be the sixteenth (and, thus, set the price), and the expected price if it does not bid. These probabilities are derived from an expected distribution of other agents' valuations, as well as a competitive analysis of the trading domain.

It should be noted that this approach uses a specific strategy as well. Their approach to selecting strategies is, therefore, limited to selecting appropriate parameters for some of the components in their agent. In this way, they generate a number of candidate strategies which they evaluate experimentally. Since this process is very relevant to our work, it is presented independently in Section 2.5.

Other agents used a variety of other strategies:

- ATTAC [81, 82]

The core of ATTAC's approach is a learning algorithm that builds a model of price dynamics based on empirical data, and utilizes this model to compute bids. ATTAC uses a distribution-based approach, relying on a general boosting-based algorithm for conditional density-estimation problems of this kind, that is, supervised learning problems in which the goal is to estimate the conditional distribution of a real-valued label. Given posted prices for goods, one can compute a set of sales, purchases, and an allocation of goods to clients that maximizes profits. However, the utility of this approach is very sensitive to the accuracy of predictions.

- ROXYBOT [32, 7]

ROXYBOT focused its approach on building an optimal solver to the optimization problem faced by the agent. They do this by incorporating a lot of clever heuristics

into their search algorithm. In order to form their optimization problem, they relied on a set of price point estimates, called “pricelines”. They later extended this by computing policies not simply from price point estimates, but rather from estimated price distributions, and furthermore, by using a Monte Carlo simulator to generate the price point estimates by sampling from estimated price distributions.

- LIVINGAGENTS [19]

LIVINGAGENTS divided their agent into various sub-agents. The main of these are a manager agent, which is responsible for starting and stopping the other agents, and 8 agents representing the clients, which are responsible for the calculation of the best combination of tickets for their respective clients. Then other agents are responsible for getting the information from auctions and placing bids. What was remarkable about their strategy, was that it made most of its decisions at the very beginning of each game, and only decisions to buy or sell entertainment tickets can occur later in the game. Its strategy is reminiscent of one of the strategies we experimented with in Chapter 7.

- PACKATAC [12]

The PACKATAC agent is a hybrid open-loop, closed-loop agent that commits to some aspects of packages at the beginning of the game, and later completes packages by optimizing over current and predicted states of the game. In fact, it is rather similar in some aspects to our own entry in TAC 2001, before we implemented most of our methodology.

- SOUTHAMPTONTAC [37, 38]

SOUTHAMPTONTAC is an adaptive agent that varies its bidding strategy according to the prevailing market conditions. To be more specific, it classifies TAC game instances into noncompetitive, semi-competitive, and competitive, depend-

ing on how much competition it expects, while purchasing the limited resources in the game (i.e., hotel rooms). The agent makes this classification based on the outcomes of the most recent games and the current prices of the hotels in the various auctions. Depending on the classification, the agent uses a different strategy that is most appropriate. SOUTHAMPTONTAC uses fuzzy reasoning techniques to predict hotel clearing prices. It uses several fuzzy rules and the factors considered in the prediction are the prices observed in various auctions and the latest price changes.

- PAININNEC

PAININNEC's strategy incorporates a combination of heuristics, including a genetic algorithm-based optimization technique. This optimization routine takes, as input, the auctions' expected clearing prices, as well as the clients' preferences, and outputs a set of goods to buy and sell. In the genetic algorithm, each client's package is represented by a six-digit string. The various strings are evaluated and are selected for reproduction using the tournament method; that is, two strings are selected at random, and with high probability, the string with the greater objective value reproduces, although errors occur with low probability.

- TOMAHACK

The TOMAHACK team's goal was to model TAC as a sequential decision process, specifically a partially observable Markov decision process (POMDP). Although TAC is a multiagent domain that might more accurately be modeled as a partially observable Markov game, the TOMAHACK team made the simplifying assumption that the other agents are part of the environment. In the TOMAHACK formulation, the state of the game encompassed elapsed time, client preferences, current quotes, outstanding bids, and the agent's holdings. A key difficulty in this

choice of representation was how to best determine the underlying state-transition function. In an attempt to uncover the state-transition function, the TOMAHACK team built models of the bidding behavior of each of its opponents. The models were constructed offline. Each opponent was modeled as a function from observed state to active bids and neural network regression was used to approximate this function.

- **THALIS**

In its initial allocation, an optimizer decides on the best package for each client. They use a very limited form of Strategic Demand Reduction to defer purchases and increase their utility, but their algorithm is nowhere as sophisticated as the related bidding strategies in our agent. The most interesting feature of its bidding component is that it issues its initial bids for the limited resources (i.e., hotel rooms) at relatively high prices to buy some learning time for the optimizer.

A number of other agents (e.g. SICS [3], KAVAYAH, CUHK, TNITAC, UMBC-TAC, ZEPP) used algorithms that were overall similar to the ones that have already been presented here. For this reason, they are not presented here, but one may find further information on the web at TAC related sites:

`www.sics.se/tac`

`tac.eecs.umich.edu`

2.5 Experimental Equilibria

In the previous sections, we presented several approaches to the problem of bidding in several auctions. Common to all these approaches was the use of one or more “heuristic strategies”, which define the actions an agent will take within each auction. Each one of these strategies, although based on principled approaches, only covers a single point

of the space of strategy spaces. Even though in practice they are quite efficient, in most cases, there is no reason to believe that more efficient strategies do not exist. Our methodology presents a more systematic way for selecting strategies across the whole space of possible strategies (see Chapter 5).

However, once these candidate strategies are selected, it is necessary to determine which strategy works best. This is difficult even for a very small number of candidate strategies. For this reason, there is work on what is commonly called an “experimental equilibrium”. A heuristic payoff table is formed, based on the performance of the candidate heuristic strategies when they participate against each other. This table is similar, in essence, to the matrix form of a game. Thus it has been used as the basis for several forms of analysis, including computation of the Nash equilibria with respect to the restricted strategy space, and the market efficiency at those equilibria. Since we allow the players to use only the set of “heuristic strategies” provided, finding the equilibrium is usually easy once the payoff matrix has been computed. There is, in fact, a rich tradition in performing experiments with automated trading agents.[95, 98]

However, creating the payoff table is not a trivial task. The expected payoff to each agent in the auction for all strategy profiles must be measured through usually a very large number of games. To get just one entry of this table, a number of samples from the distribution of agent types is taken, and the game is run with a particular profile of candidate strategies for all these samples. The distribution of scores for each agent gives the average score for each agent under this strategy profile and its variation. Especially in the case of complex real-time games, like TAC, generating the matrix is much more costly than computing the equilibrium.

In [96], the authors propose an information-theoretic methodology to the sampling problem, and thus, address the problem of selecting simulations more intelligently. They describe methods to interleave the sampling of the payoff in the underlying market game

with the calculation of Nash equilibrium. Using the information again, that they expect to get from each new sample based on their current beliefs, they sample the strategy profile that is expected to provide the most value of information.

2.5.1 Hierarchical Reduction of Large Symmetric Games

An approach that is much closer to our own work was proposed by the University of Michigan multi-agent research group, led by Professor Wellman.[99, 73] In fact, the experimental section of our methodology has been influenced by their approach, especially the part where the effect of the random parameters is computed, and an effort is made to adjust the agents' scores appropriately. Furthermore, the latest directions of their work were heavily influenced by our approach, since we were the first research team participating in TAC to perform a comprehensive and exhaustive search over candidate strategies, and also by our decision to run agents in pairs, in order to reduce the number of games needed, which influenced, to some degree, their methodology of hierarchical reduction of large symmetric games.

The idea behind their research is that although an agent's payoff does depend on the play of other agents, it may be relatively insensitive to the exact numbers of other agents playing particular strategies. So in the case of a game where m of the other agents play a particular strategy s , it is expected that the payoff of their agent will vary smoothly with the number m . Therefore, they can restrict their attention to subsets of profiles, for instance those with only even numbers of any particular strategy (like we do in most of our experiments). This transforms the N -player game to an $\frac{N}{2}$ -player game over the same set of heuristic strategies. These two games are assumed to be reasonably similar, and the second one needs a far smaller number of games in order to form the payoff matrix.⁴ They make the following definitions:

⁴The new number of games can be as low as \sqrt{z} , where z is the original number of

Definition 13. $\Gamma = \langle N, \{S_i\}, \{u_i(\cdot)\} \rangle$ is an N -player normal form game, with strategy set S_i the available strategies for player i , and the payoff function $u_i(s_1, \dots, s_N)$ giving the utility accruing to player i when players choose the strategy profile (s_1, \dots, s_N) .

Definition 14. A normal-form game is symmetric if the players have identical strategy spaces ($S_i = S$) and $u_i(s_i, s_{-i}) = u_j(s_j, s_{-j})$, for $s_i = s_j$ and $s_{-i} = s_{-j}$ for all $i, j \in \{1, \dots, N\}$. Thus, we can denote a symmetric game by $\langle N, S, \{u(\cdot)\} \rangle$, with $u(t, s)$ the payoff to any player playing strategy t when the remaining players play profile s .

Definition 15. *Reduced game*

Let $\Gamma = \langle N, S, \{u(\cdot)\} \rangle$ be an N -player symmetric game, with $N = p \cdot q$ for integers p and q . The p -player reduced version of Γ , written $\Gamma \downarrow p$, is given by $\langle p, S, \{\hat{u}(\cdot)\} \rangle$, where

$$\hat{u}_i(s_1, \dots, s_p) = u_{q \cdot i}(\underbrace{s_1, \dots, s_1}_q, \underbrace{s_2, \dots, s_2}_q, \dots, \underbrace{s_p, \dots, s_p}_q)$$

In other words, the payoff function in the reduced game is obtained by playing the specified profile in the original q times.

As an example of this approach, they compute the Bayes-Nash equilibrium for the reduced game of the first-price sealed-bid auction, which they denote $FPSB_N$, where N is the number of bidders. We know that the equilibrium strategy for $FPSB_N$, when the bidders' valuations are drawn from a uniform distribution $U[0, 1]$ is $g(u_i) = \frac{N-1}{N} \cdot u_i$. They compute the unique symmetric equilibrium for the reduced game $FPSB_N \downarrow p$ to be

$$g(u_i) = \frac{N \cdot (p-1)}{p + N \cdot (p-1)} \cdot u_i$$

They apply their technique to the TAC game by starting with game $TAC \downarrow 1$. They have 35 candidate strategies and, therefore, in $TAC \downarrow 1$ they need to examine 35 strategy profiles; note that in each profile all the agents use the same strategy. Then they use this

games needed.

information to restrict the number of strategies at the next level, that is the reduced game $TAC \downarrow 2$, and then at $TAC \downarrow 4$, etc. They do not need to examine strategy profiles that are similar to strategy profiles for which it is known that a certain strategy is performing poorly. They spent almost a year of run time in order to collect enough games in order to analyze the game $TAC \downarrow 4$. They decided not to analyze $TAC = TAC \downarrow 8$, because of the extreme amount of games that they would need. Even this approach, which reduces considerably the number of games required in order to compute an experimental equilibrium, takes an extremely large number of games compared to our methodology, but it has the potential of being a bit more accurate.

2.6 Other Interesting Auction Related Work

Computationally Bounded Agents

The results from game theory and auction theory that we presented earlier in this chapter (see Section 2.2), ignore the fact that in many multi-agent systems, the agents are not fully rational. Since, in reality, agents tend to have restrictions in the amount of time or other resources that they have available, they are computationally bounded; this means that they may be unable to compute the optimal strategies in the case that these are available or even their utilities. In cases like this, the equilibrium for computationally bounded agents are not necessarily the same as the ones computed for fully rational agents. Thus, the computationally bounded agents will not performed in the desired way. Although in this dissertation we do not have to deal with this problem, because in the case that we have to compute an equilibrium (see Chapter 6), we pre-compute the solution at our leisure and then input the equilibrium strategy in the agent in matrix form. We do keep in mind that our agent has limited resources. In fact, we chose scalable and fast methods, even if they are sub-optimal wherever possible.

A considerable amount of work on this subject was done by Professor Larson.[45, 46, 47] In her work, she introduces a model of bounded rationality where agents must compute in order to determine their preferences. She presents a fully normative model of deliberation control, the performance profile tree, and a new game-theoretic solution concept, the deliberation equilibrium. She computes the social welfare that is lost due to the computation limits imposed on the agents and analyzes some negotiation protocols for computationally limited agents. Among other things, she computes the deliberation equilibrium for various auctions and, in particular, she shows that the Vickrey auction is no longer incentive compatible, but rather that the agents need to strategically deliberate, that is use computing resources in order to (partially) determine their competitors' valuations. She also examines what mechanisms would be appropriate for computationally bounded agents.

There is also a considerable amount of work on anytime algorithms, that is algorithms that can be stopped at any time and would provide a solution, whose quality depends on the amount of time the algorithm was allowed to run. For more information see e.g. [35].

Fairness in Assigning Chores

There is a large number of other extremely interesting work on auctions, and its intersection with various other fields. We present a few of these in this section.

In [80], the authors propose an auction that achieves a notion of “fairness” for dividing chores among agents. Each agent provides its cost for performing each chore. Fairness is defined, as assigning the jobs in such a way that no agent gets a much larger share than the others. They give a mechanism that is budget balanced and incentive compatible, and each chore is assigned to the agent with the minimum bid, so this mechanism is efficient. But in order to achieve this, they make each agent pay a sum to the

agents that get the chores that is related to the bids of the agents for the chores that they were assigned. This, in turn, means that if one agent should happen to have the lower cost for doing most (or all) the chores, it is going to be forced to do them all, and all the other agents will be forced to pay large sums of money to that agent.

In [4], a different notion of fairness is presented. The authors define the notion of “min-max fairness”, as minimizing the maximum total cost of any agent. They proceed further to define the notion of “lex min-max fairness”: according to this notion, not only should the maximum total cost of any agent be minimized, but also the second highest total cost, and then the third highest, etc. until the total cost for all agents is as small as it can possibly get. The authors prove that even determining a min-max fair allocation is an NP-hard problem, and give an empirical study of the hardness of the problem; they identify subcases that are tractable and critical parameters for the hardness of the problem. Even though this is a much better notion of fairness than the one presented in [80], this mechanism does not achieve incentive compatibility.

Chapter 3

Methodology and Agent Architecture

An agent facing the general setting described in Section 1.1 receives as input:

1. its type θ , which is determined by the random parameters that affect its utility function, and which is received once (at the beginning of each game); and,
2. the various messages it receives from each auction, concerning quantities bought and sold, current ask and bid prices, closing times, etc.

The output messages of the agent are the set of bids \mathcal{B} that it submits.

The goal of the agent is to maximize its utility from selling the trips to its customers minus expenses.

If the prices at which goods would be bought or sold were known, then the agent would have a rather simple task; it would have to find the set of goods that maximizes its profit, and then it can get them, by bidding at those prices. However, this is not the case, as prices can be quite unpredictable and vary a lot between different instances of the same game. This is the reason why a lot of research has been devoted to learning bidding patterns and predicting auction closing prices. It is also counterintuitive, to some degree, to split the problem into many parts and then to optimize each one independently. However, by splitting the problem, we are able to analyze the various components independently. This approach, which proceeded in the opposite direction from the one taken by most other researchers in the field, is the general idea of our methodology. Figure 3.1 presents the decomposition of the problem (and the agent design), according to our methodology; the type θ of the agent is not presented as input there, because it is assumed to be already incorporated into the utility function of the agent.

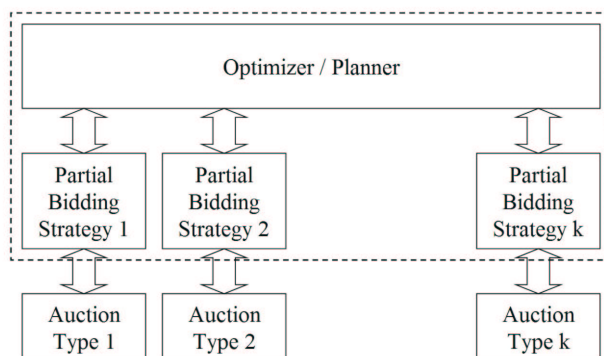


Figure 3.1: Diagram of the problem decomposition according to our methodology.

To elaborate further, there are two main parts in our methodology. Initially, we decompose the problem into sub-problems, identify the tradeoffs present in each component, and decide on the range of possible strategies that can be applied for each component. We then used rigorous experimentation to evaluate all of the strategic combinations, and to determine the “best strategy” overall.

3.1 Problem Decomposition

According to our methodology, we must decompose the problem as much as possible. This involves two different component types: the optimizer and the bidding modules.

3.1.1 Optimization Module

Input The “optimizer” module takes information about the current bid and ask prices, the status of each auction (open or closed etc.), as well as the quantities that the agent currently possesses from the bidding modules. It then proceeds to generate price predictions, based on this information.

Output It determines the quantities to be bought (or sold) for each commodity, but not

the price that the bid should have or the time of the bid placement. The desired quantities to be traded are computed by maximizing the utility of the agent, assuming that all the goods are traded at the predicted prices (even if we know that the predictions are inaccurate), and that every unit will be bought instantly. These quantities are then sent to the corresponding bidding components for acquisition of the commodities.

Optional Output It should be noted that the optimizer may also provide to the bidding components information on the marginal utility of each unit, of each commodity needed by the plan that it has formed.

Definition 16. The marginal utility w_c^i of the i^{th} unit of commodity c is defined as follows:

$$w_c^i = U(\tilde{S}^{-c} \underbrace{\cup \{c, \dots, c\}}_{i \text{ units}}) - U(\tilde{S}^{-c} \underbrace{\cup \{c, \dots, c\}}_{(i-1) \text{ units}}) \quad (3.1)$$

where $U(S)$ is the maximum utility that the agent can get if it acquires the set of goods S ,

\tilde{S} is the set of goods that the optimizer's plan calls for, and

\tilde{S}^{-c} is the set \tilde{S} without any goods of type c .

It should be noted that i can take any value between 1 and the number of units of the commodity c that are part of set \tilde{S} , and that c can be any commodity which is part of \tilde{S} .

For more information on how this module of our TAC agent WhiteBear is implemented, see Chapter 4.

3.1.2 Bidding Modules

Input The quantity of each commodity that is to be bought or sold, which was computed by the optimizer, is input into the corresponding bidding module responsible for the auction in which that commodity is traded.

Output Each bidding module must determine the price and the time that the bid will

be placed. It must do so intelligently, as to maximize the probability that the desired quantity will be acquired, while minimizing the expense. It should also consider the fact that failure to acquire a certain good will impact on the performance of other components and the agent as a whole, and also that purchasing a good that cannot be sold later will commit the optimizer module to using this good, and thus, restrict its flexibility.

Additional Output The bidding modules also provide the optimizer module with the information on current prices, quantities bought (and what would be bought at current prices), and the status of auctions that it needs to formulate the optimization problem.

As we consider each module separately, we compute the space of possible “partial strategies” for each auction type. We called these strategies partial because they only deal with one particular type of auction. Initially, we compute the extent of this space by setting “boundary partial strategies”, that is strategies that place bids at the lowest or highest prices that a rational agent would consider possible for this setting, and at the earliest or latest time possible. This defines the boundaries of the strategy space for each auction type. Having accomplished this, it is necessary to generate “intermediate strategies”, which together with the boundary strategies, will be used in the experiments. If the boundary strategies will place bids at price p_{low} and p_{high} in a certain case, then the intermediate strategy should place its bid at price $p : p_{low} \leq p \leq p_{high}$; the same should be the case for the time of bid placement. Naturally, we cannot consider all possible strategies available, as there is an infinite number of these. We can generate intermediate strategies systematically by

- computing equilibrium strategies for each auction (making some relaxations, if necessary) and then using our observations about the equilibrium to select a strategy,
- combining the boundary strategies, e.g. by bidding as low as possible in some

cases, and the highest possible in others¹, and

- modifying the boundary strategies using empirical knowledge from the domain and strategies that work well in other domains, e.g. using the idea of strategic demand reduction to modify the “buy all tickets at the beginning of the game” strategy in the TAC game.²

It should be noted that in some cases we might wish to change one of the boundary strategies. For example, it could become obvious from experimentation, or from analysis of the game, that a certain boundary strategy is clearly inefficient; in this case we choose to “move” the boundary strategy and change it to something more appropriate. For example in the TAC 2001 game, it is not at all efficient to wait until the last possible minute to buy all the plane tickets, and thus, we chose to use a different boundary strategy in that case. For more details and the specifics of how this part of the methodology is applied to the TAC domain, see Chapter 5.

3.2 Experimental Setup

The number of possible combinations of partial strategies, coupled with the fact that each strategy can behave differently based on the mixture of agents that participate in the game, leads to a prohibitive number of experiments that need to be performed in order to get the “overall best strategy”. We propose one possible way of exploring this space: *we determine how the behavior of the multi-agent system changes when exactly one of the partial strategies is varied (different) between the participating agents*. This means that the agents in each experiment have the same combination of partial strategies for all auctions, except the one we are examining.

¹This is essentially a mixed strategy of the boundary strategies.

²This case can also be considered a mixed strategy of one of the boundary strategies and some other (most likely intermediate) strategy.

To do this in a systematic way, we use several sets of experiments, each of which is designed to evaluate one particular partial strategy in different mixtures of agents. This is important since the performance of each strategy depends on the strategies used by the other agents.

In each experiment set, we keep a fixed number of agents who are using the intermediate strategies, while systematically changing the mixture of agents using the boundary cases. This will sufficiently explore the different multi-agent environments that the agents can participate in, since the behavior caused by the intermediate strategies is within the bounds of the behavior caused by the boundary strategies. Using statistical tests, we can evaluate the performance of the agent and determine whether some agent performs significantly better than the others. Therefore, each experiment set explores the strategy space in one dimension.

There are a couple of details to note concerning this step. In order to guarantee that the strategy space is explored sufficiently, we need to have at least half of the total number of agents to be using the boundary strategies; this means that at most, half of the total number of agents would be the fixed agents using the intermediate strategies, which remain fixed in number in all of the experiments of that particular set. In case we have more intermediate strategies to experiment with than the number of slots available in each experiment set, then we organize a “tournament”; this means that we split the intermediate strategies into groups, experiment on each group, and promote the winners to compete in the next group. The second point of note is that it is probably better to use agents in pairs; the reason for this is because we can compare the performance of more agent pairs, and thus, get statistically significant results faster.

In order to explore the whole strategy space, we run several experiment sets, varying different partial strategies in each set; the partial strategies that are not varied are set to the best strategy (or strategies found so far). In the case that our experiments show that

the fixed strategy we used for one particular sub-problem was not the best one, then we vary this partial strategy in the next experiment. After a sufficient number of experiment sets, we reach an “optimal strategy”; if any one of its partial strategies is changed, the agent using this modified strategy cannot do better than the optimal strategy.

In addition, if we want to explore a feature that would improve the performance of the optimizer, we set half of the agents to use this feature and half not to, while all agents use the best partial strategy combinations that we have found thus far. However, in this case, we can also be a bit less strict about the composition of agents, if we have a reason to examine the possible improvement of a less efficient agent, when it uses this new feature.

An extra benefit of this methodology is that it allows us to derive general observations about the behavior of certain strategies in different domains. For a description of the experiments necessary in order to determine an “overall best strategy” in both instances of the TAC game, see Chapter 7.

3.3 Agent Architecture

This methodology and the desire to have a scalable and general system, imposes some requirements on the agent architecture that we must use. For an agent architecture to be useful in a general setting, it must be *adaptive*, *flexible* and *easily modifiable*, so that it is possible to make changes on the fly, and adapt the agent to the system in which it is operating. In addition, as information is gained by participation in the system, the agent architecture must allow and *facilitate the incorporation of the knowledge obtained*. The architecture should support interchangeable parts, so that different strategies are easy to implement and change, otherwise running experiments with agents using the different strategies would be quite time consuming, and also the incorporation of domain specific

```

while (not end of game) {
    1. Get price quotes and transaction information
    2. Calculate price estimates
    3. Optimizer: Form and solve optimization problem
    4. Bidding Modules: Bid to implement plan
        Determine each bid independently of all other bids
        Use a different “partial strategy” for each different module
}

```

Figure 3.2: The agent architecture used by our methodology.

knowledge would be an arduous task. These were lessons that we incorporated into the design of our architecture. The general architecture that we used follows the “Sense Model Plan Act (SMPA)” architecture (it was presented and so named by Brooks [9],) also known as the “subsumption” architecture. Other trading agents, e.g. [32], have used a similar global design. Including the decomposition in the bidding section of the architecture that we introduced, the overall architecture is summarized in Figure 3.2. After the initial data, such as the type of the agent, are received, then the agent runs the same loop continuously until the end of the game. The first step is to “sense”, that is to get price information, quantities bought, and other information on the auctions (e.g. if some have closed) from the server. Then the agent “models”, which means that it generates price predictions and updates various other statistics and models that it uses. The next step is to “plan” its action, to determine what quantities to buy or sell, which is the optimizer’s job. The last step is to transfer the quantity information to the individual bidding components which will bid (“act” to implement the plan).

This architecture is quite modular, and each component of the agent can be replaced or modified by another. In fact, parts of the components themselves are also substi-

tutable (e.g. the partial strategies). One last highly desirable requirement is to design the modules of the agent to be as fast and adaptive as possible, without sacrificing efficiency. Speed is not an absolutely crucial issue in the current implementation of the TAC game, since each agent can spend several seconds deciding its next bids. But it is important, nonetheless, due to the limited period of time that the game lasts, and the fact that the agent needs to complete each loop faster than price information is updated, in order to be able to keep accurate models of various game elements. But in general, in other instances of games that fall in the general problem setting that we examine, it would be crucial to react fast (within a couple of seconds), to domain information and other agents' actions; therefore, we design the agent with speed as a main objective as well.

Chapter 4

Optimizer

In this chapter, we present the work that was done on the optimizer component of our agent. In Section 4.1, we present the idea of *Price Estimate Vectors*, which are used by the optimizer to form the utility function. In Section 4.2, we present the way that the optimizer module chooses to perform its task, i.e., to maximize the utility function, and generate the marginal utilities.

4.1 Price Estimate Vectors

In order to formulate the optimization problem that the optimizer solves, it is necessary to estimate the prices at which commodities are expected to be bought or sold. We started from the "priceline" idea presented in [32], and we simplified and extended it where appropriate (e.g. we use only one vector per auction). We implemented a module, within the optimizer module, which calculates *price estimate vectors* (PEV). These contain the value (price) of the x^{th} unit for each commodity. For some goods, this price is the same for all units, but for others it is not; e.g. buying more hotel rooms usually increases the price one has to pay.

This vector has price estimates for all units of a particular commodity, both those that are already bought and those that are not. The reason for this is that, if an already owned unit i of a commodity can be sold for a price p_i , then, if the optimizer decides that the agent should use this unit, it loses the sale value of that item. Therefore, the PEVs have the utility loss that would be incurred, if the agent were to use a particular unit of a commodity, regardless of the fact that it is currently owned by the agent or not.

In Section 4.1.2, we discuss the implementation of this idea in our TAC agent, White-Bear. However, prior to that, we need to discuss some methods for improving the price

prediction that is inserted into the PEVs; this is presented in Section 4.1.1.

4.1.1 Prediction Improvements

Initially, we put very little effort into generating the PEVs for the hotel rooms, since we simply used a linear function

$$PEV = (\alpha + i * \beta) * Q + \gamma * i + \delta \quad (4.1)$$

where Q is the current price of the auction, i is the number of the unit of the particular commodity, and $\alpha, \beta, \gamma, \delta$ are constants that were set to certain values so that the prediction would be reasonably (but not extremely) accurate for a few minutes (but not much beyond that); these parameters vary slightly depending on the type of hotel and the day in question.

However, it became obvious as the agent became more competent, that these predictions, which are quite inaccurate over the length of the whole game, were not good enough to guarantee a good performance of the agent, no matter how much the strategies were improved. Therefore, we decided to use the historical averages of the hotel prices over several hundreds (or thousands) of games, in order to bias the price predictions. This was accomplished as follows:

$$PEV^{new} = \begin{cases} w \cdot PEV^{old} + (1 - w) \cdot P_H, & \text{if } PEV^{old} < P_H \\ PEV^{old}, & \text{if } PEV^{old} \geq P_H \end{cases} \quad (4.2)$$

where P_H is the historical average for that particular hotel room type, and w is a weight parameter that begins low (around 1/3 at the very beginning of the game), until the first hotel room auction closes. Then it increases as a linear function of elapsed time, until after half of the hotel auctions have closed.¹ It then becomes 1, and from that point on,

¹At that point, the hotel room prices have increased sufficiently and reached close (in most cases) to their final values, so the PEVs are more accurate. The accuracy issue exists mainly in the beginning of the game.

the PEVs are calculated based solely on the current price Q of the auction. As it will be demonstrated in Chapter 7, this simple change had a very obvious positive effect on the performance of the agent.

Perhaps the most innovative approach we have seen in the literature about price predictions, in this case, is the one presented in [100]. Their agent computes the price predictions based on an equilibrium in prices that is achieved through a process of tatonnement. To this effect, the agent computes the demand that would exist, if the prices were set at a certain level, assuming for the demand of the other agents, which is not known, that it is equal to the expectation of the demand over the distribution of all possible types θ for the other agents, given that we know the distribution from which the agent's types are drawn. If the supply exceeds the demand, then the prices should drop; if there is more demand than supply, then the prices should increase. This process, called tatonnement, is continued until an equilibrium is reached, where demand and supply are equal. At that point, the prices which caused the supply and demand to balance are used as the predicted prices by the agent. The authors prove that this method is at least as accurate, as the more specialized methods of other researchers, which used some kind of learning to predict the prices.

We implemented this algorithm for our agent as well. However, because this algorithm takes a considerable amount of time to run (over 15 seconds most times), we decided to use it at the beginning of the game only, and fall back to our previous approach, the historical averages, for the later stages of the game. In experimentation, we observed that the new agent did not perform better than the unmodified version. For more details and a discussion of possible explanations see Chapter 7.

4.1.2 Implementation in our TAC Agent

Let O_d^{arr} , O_d^{dep} , O_d^{goodh} , O_d^{badh} , $O_{d,t}^{ent}$ and $PEV_d^{arr}(x)$, $PEV_d^{dep}(x)$, $PEV_d^{goodh}(x)$, $PEV_d^{badh}(x)$, $PEV_{d,t}^{ent}(x)$ be, respectively, the quantities owned and the the price estimates of the x^{th} unit for the plane tickets (inbound and outbound flights respectively), the hotel rooms (that is the Tampa Towers, or “good” hotel, and the Shoreline Shanties, or “bad” hotel, respectively) and the entertainment tickets for each event t for each day d . Let us also define the operator σ , which sums the values of a function $f(i)$ for $i = 1, \dots, z$, as follows:

$$\sigma(f(x), z) = \sum_{i=1}^z f(i)$$

The price estimate vectors must take values:

$$PEV_{d,t}^{ent}(x) \begin{cases} \leq \text{current bid price,} & \text{if } x \leq O_{d,t}^{ent} \\ \geq \text{current ask price,} & \text{if } x > O_{d,t}^{ent} \end{cases} \quad (4.3)$$

$$PEV_d^{type}(x) \begin{cases} = 0, & \text{if } x \leq O_d^{type} \\ = \text{current ask price,} & \text{if } x > O_d^{type} \end{cases} \quad (4.4)$$

where $type \in \{arr, dep\}$, and

$$PEV_d^{type}(x) \begin{cases} = 0, & \text{if } x \leq O_d^{type} \\ \geq \text{current ask price,} & \text{if } x > O_d^{type} \end{cases} \quad (4.5)$$

where $type \in \{goodh, badh\}$,

Since, once bought, an agent cannot sell plane tickets nor hotel rooms to anyone else, this means that their value for resale is 0 for the agent; it has to treat the money that it has already paid for them as a “sunk cost”. Therefore, the PEV should have values of 0 for all units of hotel rooms and flight tickets that have been purchased. The agent must buy all hotel rooms that it is currently winning, if the auction closes at the current price, so these are also considered owned by the planner. For the plane tickets, the PEVs are

equal to the *current ask price*, for tickets not yet bought, but for hotel rooms, it is higher than that. The price of each room are calculated with a function of the type presented in Equation 4.1; the parameters are set a bit higher for rooms on days 2 and 3, or in the good hotel, since they are estimated to cost more than rooms on days 1 and 4, or in the bad hotel.

For the entertainment tickets' PEVs, the choice is more clear. Since all we are certain from observing the current ask and bid prices is that there is at least one ticket up for buy and sale respectively at those prices, but we're not sure if there are any more, we set the PEVs as follows:

$$PEV_{d,t}^{ent}(x) = \begin{cases} 0, & \text{if } x < O_{d,t}^{ent} \\ \text{current bid price,} & \text{if } x = O_{d,t}^{ent} \\ \text{current ask price,} & \text{if } x = O_{d,t}^{ent} + 1 \\ 200, & \text{if } x > O_{d,t}^{ent} + 1 \end{cases} \quad (4.6)$$

Note that 200 is the maximum benefit from using any entertainment ticket, so this is an appropriate estimate of cost for any tickets that we are not certain that we can buy. Also note that entertainment tickets owned by the agent do not have a value of 0, since they can be sold. By using them, the agent loses a value equal to the price at which it could have sold them. Therefore, the PEV of owned tickets is equal to the price at which they are expected to be sold.

4.2 Optimizer

The planner uses the *Price Estimate Vectors* to generate the customers itineraries that maximize the agent's utility. In order to do this, the planner must formulate the utility function. The first step in this process is to compute the agent's profit, which is the sum of the money (utilities) that it gets from each of its clients.

Let $DAYS = 5$ be the total number of days and $ET = 3$ the number of different entertainment types. Each customer i has a preferred arrival date PR_i^{arr} , and a preferred departure date PR_i^{dep} ; each arrival and departure combination is equally likely, and therefore, the middle days (days 2 and 3) are the ones during which there is a larger demand for hotel rooms. (S)He also has a preference for staying at the good hotel represented by a utility bonus UH_i , which is drawn from a uniform distribution $[50, 150]$, as well as individual preferences for attending each entertainment event j represented by utility bonuses $UENT_{i,j}$, which are drawn from a uniform distribution $[0, 200]$.

The parameters concerning each customer i 's itinerary, that the optimizer has to decide upon, are the assigned arrival and departure dates, AA_i and AD_i respectively, whether the customer is placed in the good hotel GH_i (which takes value 1 if she is placed in the Towers, and 0 otherwise) and $ENT_{i,j}$, which is the day that a ticket of the event j is assigned to customer i (this is e.g. 0 if no such ticket is assigned).

The utility that the travel plan has for each customer i is:

$$util_i = 1000 - travel_penalty + hotel_bonus + entertainment_bonus$$

The base value of each trip is \$1000 minus \$100 for each day that the arrival and departure dates are different from the preferred dates, plus the bonuses for staying at the good hotel and for attending entertainment events. Thus

$$\begin{aligned} util_i &= 1000 + UH_i \cdot GH_i & (4.7) \\ &+ \sum_{d=AA_i}^{AD_i} \max_j \left\{ UENT_{i,j} \cdot \mathbf{I}(ENT_{i,j} = d) \right\} \\ &- 100 \cdot \left(|PR_i^{arr} - AA_i| + |PR_i^{dep} - AD_i| \right) \end{aligned}$$

if $1 \leq AA_i < AD_i \leq DAYS$, otherwise

$$util_i = 0$$

because the plan is not feasible.

It should be noted that only one entertainment ticket can be assigned each day, and this is modeled by taking the maximum utility from each entertainment type on each day. We assume that an unfeasible plan, means no plan (e.g. $AA_i = AD_i = 0$).

We define the function $\mathbf{I}(bool_expr)$ to be 1, if the $bool_expr = \text{TRUE}$, and 0 otherwise.

The total income for an agent is equal to the sum of its clients' utilities. The optimizer searches for a set of itineraries (represented by the parameters AA_i , AD_i , GH_i and $ENT_{i,j}$) that maximize this profit minus its expenses. Therefore,

$$\max_{AA_c, AD_c, GH_c, ENT_{c,t}} \left\{ \sum_{c=1}^{CUST} util_c - COST \right\} \quad (4.8)$$

where the cost of buying the resources can be computed by summing the PEVs for all items used:

$$\begin{aligned} COST = & \sum_{d=1}^{DAYS} \left[\sigma \left(PEV_d^{arr}(x), \sum_{c=1}^{CUST} \mathbf{I}(AA_c = d) \right) \right. \\ & + \sigma \left(PEV_d^{dep}(x), \sum_{c=1}^{CUST} \mathbf{I}(AD_c = d) \right) \\ & + \sigma \left(PEV_d^{goodh}(x), \sum_{c=1}^{CUST} [GH_c \cdot \mathbf{I}(AA_c \leq d < AD_c)] \right) \\ & + \sigma \left(PEV_d^{badh}(x), \sum_{c=1}^{CUST} [(1 - GH_c) \cdot \mathbf{I}(AA_c \leq d < AD_c)] \right) \\ & \left. + \sum_{t=1}^{ET} \sigma \left(PEV_{d,t}^{ent}(x), \sum_{c=1}^{CUST} \mathbf{I}(ENT_{c,t} = d) \right) \right] \end{aligned} \quad (4.9)$$

Once the problem has been formulated, the next step is solving it. This problem is NP-complete, but for the size of the TAC problem, an optimal solution usually can be produced fast (within 5 seconds in most cases, in a system similar to our test system). In order to create a more general algorithm, we realized that it should scale well with the size of the problem, and should not include elaborate heuristics applicable only to the TAC problem. Thus, we chose to implement a greedy algorithm: the order of customers is randomized and then each customer's utility is optimized separately, using

a Depth First Search algorithm with pruning of paths that are probably not optimal. This is done several hundred times, with a different random client ordering each time, so as to maximize the chances that the solution will be optimal. It should be noted, that even if we took all $8! = 40320$ possible orderings of customers, the solution might still not be optimal. A case in which this happens is when two clients have similar itineraries, and both wish to get the same two entertainment tickets; in each step of the algorithm, whichever customer is examined first would get both tickets, and thus, the overall solution would be to give the tickets to the customer who has the highest sum of valuations for them. However, the optimal solution might be to give one ticket to each customer. Situations like this, however, do not happen too often, and furthermore, we are able to explore a very large number of customer orderings between several runs of the optimizer.

In practice, we have found the following additions to be quite useful:

1. We compute the utility of the plan (or set of itineraries) \mathcal{P}_1 from the previous loop before considering other plans. Thus, the algorithm always finds a plan \mathcal{P}_2 that is at least as good as \mathcal{P}_1 and there are relatively few radical changes in plans between loops. We observed empirically that this prevented some radical bid changes and improved efficiency. The reason for this is that without this improvement, the agent might not consider any customer ordering that provides at least as good an itinerary as the plan \mathcal{P}_1 . In that case, the agent's bidding would tend to be inconsistent, as far as the items that it wishes to acquire. Furthermore, by using this improvement, we have a fairly high chance of finding the ordering of customers that maximizes the utility.
2. We added a constraint that dispersed the bids of the agent for resources in limited quantities (hotel rooms in TAC). Plans, which demanded a single day, more than 4

rooms in the same hotel,² or more than 6 rooms in total, were not considered. This leads to some utility loss, in rare cases. However, bidding heavily for one room type means that overall demand will very likely be high, and therefore, prices will skyrocket. This will, in turn, lower the agent's score significantly. We observed empirically that the utility loss from not obtaining the best plan tends to be quite small, compared to the expected utility loss from rising prices. This improvement was inspired by the idea of Strategic Demand Reduction. [97]

Marginal Utilities

The second output of the optimizer is the marginal utilities of the goods that it wishes the agent to acquire. To compute the marginal utility u_c^i of the i^{th} unit of commodity c , the optimizer must do the following:

1. Set the PEV for the auction in which this commodity c is traded equal to ∞ for any good after the i^{th} ; this essentially means that the optimizer is unable to acquire more than i units of c .
2. Compute the optimal itinerary in that case, that gives utility U_i^c .
3. Set the PEV for the auction for commodity c equal to ∞ for any good after the $(i - 1)^{th}$; this essentially means that the optimizer is unable to acquire more than $(i - 1)$ units of c .
4. Compute the optimal itinerary in that case, that gives utility U_{i-1}^c .
5. The marginal utility is $u_c^i = U_i^c - U_{i-1}^c$.

²To enforce this rule, we only had to change the PEVs so that any unit after the fourth has a PEV equal to ∞ .

However, since this would involve computing the optimal allocation several times in each loop of the agent, we decided that it was more appropriate to compute an approximate value of the marginal utilities that can be computed very fast (in milliseconds). We examine for each client whose itinerary requires a particular resource (hotel room), what is the effect on the utility of that itinerary when that room becomes unavailable. To be more precise, we used the following algorithm to compute the marginal utility u_c^i in the case that a client j is using the i^{th} unit of commodity c :

1. Set the PEV for the auction for commodity c equal to ∞ for any good after the $(i - 1)^{\text{th}}$; this way the client cannot get that unit.
2. Assume that all clients other than j have the same itineraries as in the optimal plan.
3. Compute the optimal itinerary just for client j , that gives total (for all clients) utility U'_j .
4. The marginal utility is $u_c^i = U - U'_j$, where U is the utility of the optimal itinerary.

Although we examine the effect on only one client when resources are denied, this is fairly close to the “true” marginal utility, that we would get from running the complete algorithm twice, since our algorithm for solving the full optimization problem examines a random ordering of clients and optimizes each client independently.

Performance

We have also verified that this randomized greedy algorithm gives solutions, which are often optimal and never far from optimal. To verify this, we compared the itineraries at the end of the game, that were provided by the optimizer in 100 randomly selected runs of the agent, against the optimal allocations that the server generated for our agent. We

observed that over 50% of the itineraries were indeed optimal and on average, the utility loss was about \$15 out of \$9800 to \$10000 usually,³ namely close to 0.15%. Compared to the average utility of about \$3000 that our agents scored in most games, they achieved about 99.5% of optimal. These observations are consistent with the ones about a related greedy strategy in [81]. Considering that at the beginning of the game the optimization problem is based on inaccurate values, and since the closing hotel prices are not known, an 100%-optimal solution is not necessary and can be replaced by our almost optimal approximation. As commodities are bought and the prices approach their closing values, most of the commodities needed are already bought and we have observed empirically that bidding is rarely affected by the generation of approximately optimal solutions, instead of optimal ones.

This algorithm takes approximately *1 second* to run through 500 different randomized customer orderings, and compute an allocation for each. Our test-bed was a cluster of 8 Pentium III 550 MHz CPU's, with each agent using no more than one CPU. This system was used for all our experiments and our participation in the TAC.⁴ Hence, it is verified that our goal to provide an optimizer that is fast, and not domain specific, is accomplished without sacrificing the agent's overall performance.

³These were the scores of the allocations at the end of each game without considering expenses.

⁴During the competition, only one processor was used, but during the experimentations we used all 8, since 8 different instantiations of the agent were running at the same time.

Chapter 5

Generating Strategies and Dealing with

Tradeoffs

In this chapter, we present the work that was done on the bidding modules of our agent. Once the optimizer has generated the desired types and quantities of each good, the bidding modules place separate bids for all these goods; each module must determine the time of bid placement and the price offered in each bid. According to our methodology, we need to find strategies for each different set of auctions, and this procedure is described in this chapter. Our methodology provides a more structured and systematic way of generating strategies than previous other approaches, which were mostly “heuristic”. However, there is still a fair amount of freedom for researchers to incorporate empirical knowledge and observations into the strategies together with principled approaches, where applicable. In fact, every participating team in the TAC competition, including ours, used empirical observations from the games it participated in (several thousands of games between the 2001 and 2005), in order to improve its strategy. In the next sections, we examine the tradeoffs that exist in the various auctions and generate strategies for each bidding component, that will later be evaluated in experiments.

5.1 Paying for Flexibility: The Flight Auctions

The purchase of flight tickets presents an interesting dilemma. Under the rules of the TAC game, the prices will vary according to a random walk. In TAC 2000, the expected increase was 0, so it was a dominant strategy to wait until the end of the game in order to buy the tickets needed.

However, in TAC 2001, the random walk was changed so that the prices would

increase and, in fact, we have calculated (based on the model of price change described in the rules) that ticket prices are expected to increase approximately, in proportion to the square of the time elapsed since the start of the game. This means that the more one waits, the higher the prices will get, and the increase is more dramatic towards the end of the game. From that point-of-view, if an agent knows accurately the plan that it wishes to implement, it should buy the plane tickets immediately. On the other hand, if the plan is not known accurately (which is usually the case), the agent should wait until the prices for hotel rooms have been determined. This is because buying plane tickets early restricts the flexibility (adaptability) that the agent has in forming plans: e.g. if some hotel room that the agent needs becomes too expensive, and if it has already bought the corresponding plane tickets, it must either waste these, or pay a higher price to get the room. *An obvious tradeoff exists in this case, since delaying the purchase of plane tickets increases the flexibility of the agent, and hence, provides the potential for a higher income at the expense of some monetary penalty.* One way to solve this is to use a cost-benefit analysis. In this case, the cost of deferring the purchase can be computed, but in order to estimate the benefit from delayed buying, one must use models for the opponent agents, which are not easy to obtain. It is, however, easy to compute that the cost is prohibitively high if a lot of ticket purchases are deferred until the end of the game.

In TAC 2004, the rules changed once more and the price increase became rather limited (on expectation), so the pressure to buy the flight tickets early was relieved. In 5.1.1 and 5.1.2, we present the strategies for dealing with this tradeoff.

5.1.1 Tradeoffs in TAC 2001 Game

The model of the random walk for the ticket prices in TAC 2001 generated significant increases in prices over time. The prices are perturbed every 24 to 32 seconds by a value

drawn uniformly from $[-10, x(t)]$,

where $x(t) = 10 + (t/720) \times (x - 10)$, and

t is the elapsed time from the beginning of the game.

The only unknown variable is the maximum change x , which is drawn uniformly from $[10, 90]$. But we can compute the most likely values of x , by computing the probability of $x = z$, $z \in [10, 90]$ as is demonstrated in Section 5.1.3. Using these probabilities, we can compute an estimate of the expected price increase at any point.

Boundary Strategies

The first step, according to the methodology, is to decide the boundary strategies. Since the only issue is the time of bid placement, the two obvious strategies are:

- to buy everything at the beginning of the game, or
- to defer all the tickets purchases to the end of the game.

However, after a relatively short number of runs, it became obvious that the second strategy is quite clearly inefficient. A cost analysis reveals that the agent would have to pay at least several thousands of dollars more, if it leaves all the ticket purchases for the end of the game, which is clearly a lot more than any benefit that is derived. To this effect, we changed that boundary strategy to:

- Defer all the tickets purchases to a much later time.

Initially, we set this later time to be right after 2 (out of the 8) hotel auctions have closed. The reason for this is that at that time, the intentions of the other agents can be partially observed by their effect on the auctions' bid prices, and thus after this time, the room prices approximate sufficiently to their potential closing prices. Hence, a plan generated at that time is usually quite similar to the optimal plan, when the closing prices are

known. Another reason is that since ticket prices are expected to increase approximately in proportion to the square of the time elapsed, the price increases after this point tend to be prohibitive.

However, this is still not a very good boundary case. It became apparent quite fast that a limited number of tickets could be bought at the beginning of the game without noticeable loss of flexibility on the part of the optimizer. Therefore, a further improvement to this boundary strategy is to buy some tickets at the start of the game. We buy about 50% of the tickets at the beginning: these are the “almost certain to be used” tickets. These tickets are bought based on the prices and the preferences of the customers (preference is given to tickets on days 1 and 5, to cheaper tickets, and to customers with shorter itineraries). About 50% of the tickets are bought in this way, and we have observed that these tickets are almost never wasted.

Another improvement that we did to this strategy is to bid for 2 less tickets per flight than required by the plan, until the first hotel auction closes and for 1 less until the second hotel auction closes. This provides greater flexibility and it is highly unlikely that the final optimal plan will not make use of these tickets. Although it is rather rare that more than 2 tickets would be needed on any day, in addition to the “almost certain to be used” tickets that are already bought in the beginning, this can happen; having more than 2 unpurchased tickets of one type gives almost no extra flexibility compared to having just two, even when the price estimate (and the plan that the optimizer forms) are highly inaccurate. The reason for leaving only 1 ticket unpurchased between the first two hotel auctions’ closing times, is that the price predictions become a lot more accurate after some bids have been placed, and all agents place bids for everything they need before the first closing time. Therefore, the optimizer requires less flexibility and 1 unpurchased ticket accomplishes just that.

Thus, the boundary strategies we used in the experiments are:

- Buy everything at the beginning of the game, and
- Buy at the beginning of the game only the tickets that are almost certain to be used, and defer all other ticket purchases to a later time; after the first hotel auction closes, buy all except one ticket of each type, and buy everything immediately after 2 hotel room auctions have closed.

Intermediate Strategies

Given these boundary strategies, we first obtained an intermediate strategy by modifying the second boundary strategy as follows:

- Buy at the beginning of the game only the tickets that are almost certain to be used, and then all the tickets except one of each type; buy everything immediately after the first hotel room auctions closes.

Another intermediate strategy is a modification of the first boundary strategy (“buy everything at the beginning”), and comes from the idea of *strategic demand reduction* [97]: we compute the minimum number of tickets which, if left unpurchased, will allow the agent to complete its itineraries, even if it fails to buy a hotel room on days during which it wishes a lot of rooms (a day when such a problem exists is one in which the agent wishes either 4 rooms of one kind, or 6 in total). A “small” optimization problem is solved to determine these tickets. In fact, an optimal solution with the minimal number of unpurchased tickets can be found by using a greedy algorithm; the only condition is that we wish the tickets to be exactly on the date that there is a problem, and not before if possible. We run through the days in increasing order and once a problem day is found, we remove an inbound flight for that day. We then take the minimum of the longest trip starting on that day and the continuous number of days for which there is a problem with the particular hotel of the same type (days for which the problem

is 6 total tickets count as both types), and continue the process from the next day. A similar algorithm is run for the outbound flights, but backwards as far as the days are concerned. 80% to 100% of the tickets are now bought at the beginning and usually it is over 90-95%. Therefore, this boundary strategy can be summarized as:

- Buy at the beginning of the game the maximum number of tickets that allow no resources to be wasted (flight tickets or hotel rooms), even if the agent fails to buy, at most, one hotel room in each day where the demand of hotel rooms for the agent's plan is at its highest limit.

General Improvements

An improvement, especially for agents who defer the purchase of some tickets, was obtained by estimating the likelihood of price increases using the estimates of the hidden parameter x of each flight, that it already computes. In practice, we have observed that just the knowledge of the smallest value of z for which $p_z > 0$ is enough to estimate the price increase, and that we obtain this knowledge quite fast, that is well before the first hotel auction closes. This information is then used to bid earlier for tickets whose price is very likely to increase, and to wait longer, for tickets whose price is expected to increase little or none; these are bought when their usefulness for the agent is almost certain. We calculated that the agent approximately halves the cost it would otherwise pay for the deferred purchases.¹

¹If this procedure is not used, then the agent pays an average extra cost of 240 to 300 because it waits until the 4th or 5th minute (after the first two hotel auctions close), to buy the last tickets and this doubles the cost.

5.1.2 Tradeoffs in TAC 2004 Game

The rule changes in 2004 eliminated the tradeoff in the purchase of plane tickets. In fact, it can be shown that it is a dominant strategy not to buy any tickets whatsoever at the beginning of the game. The reason for this is that until after the middle of the game, there is no increase in the expected price of the plane tickets, so the agent can defer these purchases, and has more flexibility in its planning. The model of the random walk for the ticket changes is known, since the prices are perturbed every 10 seconds by a value drawn uniformly from

- $[-10, x(t)]$, if $x(t) > 0$
- $[x(t), 10]$, if $x(t) < 0$
- $[-10, 10]$, if $x(t) = 0$

where $x(t) = 10 + (t/540) \times (x - 10)$, and t is the elapsed time from the beginning of the game. The only unknown variable is the hidden parameter x , which is drawn uniformly from $[-10, 30]$. But we can compute the probability of $x = z$, $z \in [-10, 30]$ as is demonstrated in Section 5.1.3. Using these estimates, we can compute the expected price increase at any point. Therefore, we modified the strategy used in the agent as follows:

We decide how many plane tickets to buy depending on their expected price increase, and the time elapsed since the beginning of the game; the higher the increase is, the larger the percentage of desired tickets that the agent buys. Our agent is relatively risk averse in the fact that if the expected price increase is definitely positive (i.e., we know that $P[x \leq 10] = 0$), then it buys most of what it needs at an early time and buys every needed unit of that commodity right after the middle of the game. In addition, a case of special interest is when the probability that the hidden parameter $x < 0$ for that flight

becomes high, an event that only occurs in the second half of the game; in this case our agent buys all desired tickets for that particular flight immediately. This is done not only because there is a definite and significant price increase predicted in this case, but also because by that time, the agent has secured most of the needed hotel rooms. Based on the performance of our agent in the competition, it is clear that this strategy works extremely well.

It should be noted that we also tried other variations of this strategy, e.g. buying all desired quantity of a certain plane ticket when its price would reach the minimum price possible. However, based on the experiments the we ran, it did not appear that such changes improved the performance of the agent, therefore, they were not used in our agent that participated in TAC 2004, nor in our experiments.

5.1.3 Flight Price Models

In this section, we compute the hidden parameter of each flight auction in the both the TAC 2001 and the TAC 2004 game.

TAC 2001

Let $\{y_i, t_i\}, \forall i \in \{1, \dots, N\}$ be the pairs of the price changes y_i observed at times t_i , and let N be the number of such pairs.

Let $A = x - 10$. Then $A \in \{0, \dots, 80\}$ and

$$P[A = z] = \frac{1}{81}, \forall z \in \{0, \dots, 80\}$$

and

$$P[A = z] = 0, \forall z \notin \{0, \dots, 80\}$$

since x is uniformly chosen among the integers in $[10, 90]$.

Since A is independent of the times $\{t_i = T_i\}$ when the changes occur, therefore,

$$P[A = z] = P[A = z | \{\bigwedge_{i=1}^N (t_i = T_i)\}]$$

Also y_i is uniformly chosen in $\{-10, \dots, 10 + \lfloor \frac{A \cdot T_i}{720} \rfloor\}$, therefore,

$$P[y_i = Y_i | (A = z) \wedge (t_i = T_i)] = \frac{1}{\lfloor \frac{A \cdot T_i}{720} \rfloor + 21}$$

if Y_i is an integer in $[-10, 10 + \frac{A \cdot t_i}{720}]$, otherwise it is

$$P[y_i = Y_i | (A = z) \wedge (t_i = T_i)] = 0$$

In addition, the price change y_i only depends on time t_i and is independent of all $t_j, \forall j \neq i$, therefore,

$$P[y_i = Y_i | (A = z) \wedge (t_i = T_i)] = P[y_i = Y_i | (A = z) \wedge \{\bigwedge_{i=1}^N (t_i = T_i)\}]$$

Given the observations made so far, the probability that $A = z$ for any $z \in \{0, \dots, 80\}$ is

$$\begin{aligned} P[A = z | \{\bigwedge_{i=1}^N (t_i = T_i)\} \wedge \{\bigwedge_{i=1}^N (y_i = Y_i)\}] &= \\ \frac{P[(A = z) \wedge \{\bigwedge_{i=1}^N (y_i = Y_i)\} | \bigwedge_{i=1}^N (t_i = T_i)]}{P[\bigwedge_{i=1}^N (y_i = Y_i) | \bigwedge_{i=1}^N (t_i = T_i)]} &= \\ \frac{P[A = z | \bigwedge_{i=1}^N (t_i = T_i)] \cdot p_z}{\sum_{\zeta=0}^{80} \{P[A = \zeta | \bigwedge_{i=1}^N (t_i = T_i)] \cdot p_\zeta\}} = \frac{\frac{1}{81} \cdot p_z}{\sum_{\zeta=0}^{80} \frac{1}{81} \cdot p_\zeta} &\Leftrightarrow \\ P[A = z | \{\bigwedge_{i=1}^N (t_i = T_i)\} \wedge \{\bigwedge_{i=1}^N (y_i = Y_i)\}] &= \frac{p_z}{\sum_{\zeta=0}^{80} p_\zeta} \end{aligned}$$

where

$$p_z = P[\bigwedge_{i=1}^N (y_i = Y_i) | (A = z) \wedge \{\bigwedge_{i=1}^N (t_i = T_i)\}]$$

Given the value of A , $\{y_i = Y_i\}$ are independent of each other, thus

$$p_z = \prod_{i=1}^N P[y_i = Y_i | (A = z) \wedge \{\bigwedge_{i=1}^N (t_i = T_i)\}] = \prod_{i=1}^N P[y_i = Y_i | (A = z) \wedge (t_i = T_i)]$$

Since p_z is a product, that means that if any conditional probabilities that make up the product is zero then $p_z = 0$, as well. Therefore, for $p_z \neq 0$ it must be that

$$\forall i, -10 \leq Y_i \leq 10 + \frac{A \cdot T_i}{720} \Leftrightarrow \forall i, A \geq \frac{720 \cdot (Y_i - 10)}{T_i} \Leftrightarrow A \geq \max_i \frac{720 \cdot (Y_i - 10)}{T_i}$$

Therefore,

$$p_z = \prod_{i=1}^N \frac{1}{\lfloor \frac{A \cdot T_i}{720} \rfloor + 21}$$

if $A \geq \max_i \frac{720 \cdot (Y_i - 10)}{T_i}$, otherwise it is $p_z = 0$.

Thus, we conclude that

$$P[x = z | \{\bigwedge_{i=1}^N (t_i = T_i)\} \wedge \{\bigwedge_{i=1}^N (y_i = Y_i)\}] = \frac{p_z}{\sum_{\zeta=10}^{90} p_\zeta}$$

where

$$p_z = \prod_{i=1}^N \frac{1}{\lfloor \frac{(z-10) \cdot T_i}{720} \rfloor + 21}$$

if $z \geq \max_i \frac{720 \cdot (Y_i - 10)}{T_i} + 10$ and $z \leq 90$.

Otherwise $p_z = 0$.

Using these formulas, we can compute the probability $P[x = z]$ that the hidden parameter x is equal to a specific number z and use this in the agent's strategy for bidding in flight ticket auctions.

TAC 2004

Let $\{y_i, t_i\}, \forall i \in \{1, \dots, N\}$ be the pairs of the price changes y_i observed at times t_i , and let N be the number of such pairs. It is

$$P[x = z] = \frac{1}{41}, \forall z \in \{-10, \dots, 30\}$$

and

$$P[x = z] = 0, \forall z \notin \{-10, \dots, 30\}$$

Since x is independent of the times $\{t_i = T_i\}$, when the changes occur, therefore,

$$P[x = z] = P[x = z | \{\bigwedge_{i=1}^N (t_i = T_i)\}]$$

In addition, the price change y_i only depends on time t_i and is independent of all $t_j, \forall j \neq i$, therefore,

$$P[y_i = Y_i | (x = z) \wedge (t_i = T_i)] = P[y_i = Y_i | (x = z) \wedge \{\bigwedge_{i=1}^N (t_i = T_i)\}]$$

Given the observations made so far, the probability that $x = z$ for any $z \in \{-10, \dots, 30\}$ is

$$\begin{aligned} P[x = z | \{\bigwedge_{i=1}^N (t_i = T_i)\} \wedge \{\bigwedge_{i=1}^N (y_i = Y_i)\}] &= \\ \frac{P[(x = z) \wedge \{\bigwedge_{i=1}^N (y_i = Y_i)\} | \bigwedge_{i=1}^N (t_i = T_i)]}{P[\bigwedge_{i=1}^N (y_i = Y_i) | \bigwedge_{i=1}^N (t_i = T_i)]} &= \\ \frac{P[x = z | \bigwedge_{i=1}^N (t_i = T_i)] \cdot p_z}{\sum_{\zeta=-10}^{30} \{P[x = \zeta | \bigwedge_{i=1}^N (t_i = T_i)] \cdot p_\zeta\}} &= \frac{\frac{1}{41} \cdot p_z}{\sum_{\zeta=-10}^{30} \frac{1}{41} \cdot p_\zeta} \Leftrightarrow \\ P[x = z | \{\bigwedge_{i=1}^N (t_i = T_i)\} \wedge \{\bigwedge_{i=1}^N (y_i = Y_i)\}] &= \frac{p_z}{\sum_{\zeta=-10}^{30} p_\zeta} \end{aligned}$$

where

$$p_z = P[\bigwedge_{i=1}^N (y_i = Y_i) | (x = z) \wedge \{\bigwedge_{i=1}^N (t_i = T_i)\}]$$

Given the value of x , $\{y_i = Y_i\}$ are independent of each other, thus

$$p_z = \prod_{i=1}^N P[y_i = Y_i | (x = z) \wedge \{\bigwedge_{i=1}^N (t_i = T_i)\}] = \prod_{i=1}^N P[y_i = Y_i | (x = z) \wedge (t_i = T_i)]$$

This probability is known; we know that

$$x(z, T_i) = 10 + \frac{(z - 10) \cdot T_i}{540}$$

and thus, we have that

- if $x(z, T_i) > 0$ then

$$P[y_i = Y_i | (x = z) \wedge (t_i = T_i)] = \frac{1}{\lceil \frac{(z-10) \cdot T_i}{540} \rceil + 21}$$

when $-10 \leq Y_i \leq x(z, T_i)$, otherwise

$$P[y_i = Y_i | (x = z) \wedge (t_i = T_i)] = 0$$

- if $x(z, T_i) = 0$ then

$$P[y_i = Y_i | (x = z) \wedge (t_i = T_i)] = \frac{1}{21}$$

when $-10 \leq Y_i \leq 10$, otherwise

$$P[y_i = Y_i | (x = z) \wedge (t_i = T_i)] = 0$$

- if $x(z, T_i) < 0$ then

$$P[y_i = Y_i | (x = z) \wedge (t_i = T_i)] = \frac{1}{1 - \lceil \frac{(z-10) \cdot T_i}{540} \rceil}$$

when $x(z, T_i) \leq Y_i \leq 10$, otherwise

$$P[y_i = Y_i | (x = z) \wedge (t_i = T_i)] = 0$$

Using these formulas, we can compute the probability $P[x = z]$, and use this in the agent's strategy for bidding in flight ticket auctions.

5.2 Bid Aggressiveness: The Hotel Room Auctions

In the initial TAC game (2000), the hotel room auctions would close at the end of the game, unless there was a period of inactivity, which was randomly selected. However, the research teams figured out that by placing minimal bids at regular intervals, they could keep the auctions open until the end of the game, at which point the auctions

became sealed bid auctions, closing at the end of the game. Therefore, an optimal strategy was to place bids as close to the marginal utility, at the last seconds of the game. This strategy is also supported by the theoretical equilibria we compute in Chapter 6, when there is only one round left (round 8).

In TAC 2001 and later, the rules changed to what was described in Section 1.2. Bidding for hotel rooms poses some interesting questions under these rules. The main issue in this case is how *aggressively each agent should bid*, i.e., the level of the prices it submits in its bids. If it bids low, it might get outbid, and in that case, a significant loss of utility will occur. This is due to the fact that some other commodities that have already been purchased and cannot be sold back will be wasted, and to the fact that the new itinerary for the clients will give a lesser utility to the agent (otherwise it would have been selected by the optimizer as the optimal itinerary). On the other hand, if it bids high (aggressively), it is likely to enter into price wars with the other agents, and this will cause prices to skyrocket. The utility of all the agents going to suffer significantly in that case. Furthermore, assuming that the rising prices are the result mainly of the agent's bidding, this means that the commodities this particular agent wants mostly will be the ones whose price increases the most, and thus, it is the agent itself who is going to have the largest drop in utility compared to the other agents. This tradeoff must be addressed by the strategy selected for the hotel bidding modules.

As far as the timing of the bids is concerned, there is little ambiguity about what the optimal strategy is. The agent waits until the first hotel auction is about to close to place its first bid. The reason for this is that it does not wish to increase the prices earlier than necessary, and to allow other agents to respond to its bid in the current round of bids.

Boundary Strategies

The first step according to the methodology is to decide the boundary strategies. Since the only issue is the level of the price offered in each bid, the two obvious strategies are:

- to place a bid at the current bid price, plus a small increment ϵ , or
- to place a bid at the marginal utility (u_c^i).

In practice, it makes more sense to modify these boundary strategies a bit. The first boundary strategy, which is to place low bids, is modified so that the agent also bids progressively higher for each consecutive unit of a commodity for which it wants more than one unit. E.g. if the agent wants to buy 3 units of a hotel room, it might bid 210 for the first, 250 for the second and 290 for the third. This is the lowest possible level of aggressiveness, since the agent will never wish to bid less.

The other boundary strategy can be modified, so that the agent bids progressively closer to the marginal utility u_c^i of the i^{th} unit of commodity c that is traded in that auction as time passes.² Since the agent will likely lose money, if it bids above the marginal utility, this is the highest possible level of aggressiveness.

Thus, the boundary strategies we used in the experiments are:

- [*Lowest Aggressiveness*] Place a bid at the current bid price plus an increment, and bid progressively higher for each additional unit.
- [*Highest Aggressiveness*] Place a bid at progressively closer to $\frac{u_c^i}{\sqrt{z}}$ as time passes, where u_c^i is the marginal utility, and z is the number of rooms, which are still

²The marginal utility u_c^i for a particular hotel room of type c is the change in utility that occurs if the agent using the i^{th} desired unit fails to acquire it. In fact, for each customer i that needs a particular room, we bid $\frac{u_c^i}{\sqrt{z}}$ instead of u_c^i , where z is the number of rooms which are still needed to complete her itinerary. We do this, based on empirical observations, in order not to drive the prices up prematurely.

needed in order to complete the itinerary of the client who uses the i^{th} room of type c .

Intermediate Strategies

Once the boundary strategies are set, our methodology suggests that we try to combine these into intermediate strategies. We, therefore, selected the following intermediate strategy:

- Bid like the agent of highest aggressiveness for rooms that have a high marginal utility, and bids like the agent of lowest aggressiveness otherwise.

This is the agent of “medium” or intermediate aggressiveness. This is a reasonable strategy, because it bids low for rooms that are not crucial, and bids high for rooms that will have a large negative impact on the utility if they are not purchased.

Another way of combining the boundary strategies is to take a weighted sum of these. An appropriate weight seems to be the probability that a next round exists. In fact, if we observe the equilibria strategies in Chapter 6, we notice that they advocate bidding low initially (early rounds), and close to the marginal utility in the last rounds. Therefore, we decided on the following intermediate strategy:

- Place a bid equal to the weighted sum of the highest aggressiveness agent’s bid with weight $(1 - p_r)$ and of the lowest aggressiveness agent’s bid with weight p_r , where p_r is the probability that the auction will not have any more rounds of bidding.

The final intermediate strategy we used was:

- Use the equilibrium strategy for the case of an m^{th} price auction with multiple possible closing times.

This is computed in the next chapter and graphed (for some of the rounds) in Figures 6.24, 6.25, 6.27 and 6.29.

General Improvements

We observed, empirically, that an added feature which increases performance is to place bids for a small number of rooms at the beginning of the game, at a very low price (whether they are needed or not). In case these rooms are eventually bought, the agent pays only a very small price and gains increased flexibility in implementing its plan. However, this mainly works only against agents of lesser competence; in a fully competitive game (e.g. finals of the TAC competition), the other agents are less likely to allow our agent to buy hotel rooms that cheaply.

One further improvement, which was deemed necessary for the 2002 TAC competition and beyond, is to use historical data to determine the price of the hotel auction which closes first; if our bids are lesser than that price, then we increase them to become equal to this price, and we only do this just before the first hotel auction closes. We used this improvement in all of our strategies (boundary and intermediate), because we observed that our agent was often getting outbid at the first hotel auction, which closed while the bids were still low. The marginal utility just for that case was probably misleading as it is still low, since there is a lot of flexibility in the optimizer, because of the fact that a lot of commodities are still not purchased. However, the agent can't afford to lose too many rooms and let the other agents purchase them, even if the optimizer can deal with this issue.

5.3 Entertainment

The entertainment tickets do not present us with a challenging tradeoff. Therefore, we only used the following strategy. The agent buys the entertainment tickets that it needs for implementing its plan at a price equal to the current price, plus a small increment. It also sells the entertainment tickets that it does not need for implementing its plan at a price equal to the current price, minus a small increment.

The only exceptions to this rule are:

1. At the game's start and depending on how many tickets the agent begins with, it will offer to buy tickets at low prices, in order to increase its flexibility at a small cost. Even if these tickets are not used by the agent, it is highly likely that the agent will sell them for a profit.
2. The agent will not buy at a high price, and it will not sell at a low price, even if this is beneficial to its utility, because it helps other agents. This restriction is somewhat relaxed at the last minute of the game, in order for the agent to improve its score further, but it will still avoid some beneficial deals, if these would be very profitable for another agent.³

It places those bids immediately after the information is received from the optimizer. Therefore, the entertainment tickets are traded, for the most part, at the prices that the optimizer expected when it solved the optimization problem.

³This is introduced because in the competition the agent is interested in maximizing not just its own utility, but also the difference between its utility and the utilities of the other agents.

Chapter 6

Using Equilibrium Analysis to Generate

Strategies

In this chapter, we compute Bayes-Nash equilibria for first price single unit auctions and m^{th} price multi-unit auctions, when the auction has a set of possible closing times, one of which is chosen randomly for the auction to end at. Thus, the auctions have one or more rounds of sealed bids. We compute such equilibria for a wide range of assumptions, and give the algorithm used by an agent to generate these strategies. We then proceed to compute these equilibria for both a uniform probability distribution of the agents' utilities, and for a distribution sampled from the utilities that hotel rooms have for our agents in a large number of simulated games.

The part of the TAC game that we are interested in are the hotel room auctions. There are $m = 16$ rooms available each night at each of the two available hotels. Rooms for each of the days are sold by each hotel in separate, ascending, multi-unit, 16^{th} -price auctions. These auctions close at randomly determined times and, more specifically, a random auction will close every minute throughout the game. No prior knowledge of the closing order exists, and agents may not resell rooms. Between closing times the agents may place bids, but these are not opened until the next possible closing time; hence, each round that takes place between consecutive closing times is a sealed bid auction. Each agent wishes to buy, at most, 8 units. To simplify this game, we assume that each agent has 8 sub-agents, each of which is tasked with the purchase of a single hotel room; each agent i thus places a single bid $v_i = g(u_i)$ for one room, when its valuation is u_i . Therefore, each auction can be considered an m^{th} ($m = 16$) price auction with $N = 64$ agents participating, and with 8 possible closing times.

In the next sections, we will compute the equations that derive the equilibrium solution $g_r(u_i^r, Q_r)$ at round r , at which point the current price for the auction is Q_r . In Section 6.1, we only write this as $g(u_i)$, because we are interested in the solution of the first of only two rounds, when the price Q remains the same.

6.1 Equilibria For a Two Round Auction

Initially, we examined the problem in the case that there were only two possible closing times. We used the equilibria we computed in the 2-round case later, in order to compute equilibria for the R -round case.

6.1.1 Formal Definition of the Problem

We assume that N risk-neutral agents wish to buy 1 unit of a certain good each. An independent seller sells m units of the desired good in an m^{th} price auction, i.e., the good is sold to the agents which submitted the m highest bids at a price equal to the lowest winning bid. The agents have valuations (utilities) u_i , which are i.i.d. with probability distribution $F(u)$ in the first round. Each agent knows its own valuation and the distribution $F(u)$. There can be a second round with probability $(1 - p)$, where p is known. If a second round does exist, the agents have new i.i.d. utilities \tilde{u}_i drawn from some distribution $H(u)$, and can submit new bids as long as they are greater or equal to the bid price from the end of the first round. The assumptions about what each agent i knows about its utility \tilde{u}_i at the start of the game determine the different cases that we examine.

- \tilde{u}_i can be assumed to be similar to the utility of the first round u_i . This is reasonable in the case of TAC, because usually there is a correlation between the valuation of the same room over the course of the game. However, even though

\tilde{u}_i is similar to u_i ($\tilde{u}_i \simeq u_i$), they are not equal in reality. Theorem 7 provides the most general equilibrium for this case.

- The agent might not know anything about \tilde{u}_i other than that it is drawn from $H(u)$ (this is the same information that it has about the other agent's valuations). Theorem 8 provides the most general equilibrium for this case, when one substitutes $G(u)$ with $H(u)$.
- The agent might know something about \tilde{u}_i . In TAC, knowing the utilities at the previous rounds can allow the agent to compute that \tilde{u}_i is drawn from a more “tight” and accurate distribution $G(u)$, instead of $H(u)$. One example of this is that the utility at a later round is highly unlikely to decrease, so values $\tilde{u}_i < u_i$ can be discarded. Theorem 8 provides the most general equilibrium for this case.

We would like to point out that, even in the case that we assume that $\tilde{u}_i \simeq u_i$, knowledge of an agent's bid, does not lead to the information of the agent's utility \tilde{u}_i in the second round, since in reality $\tilde{u}_i \neq u_i$, even if they are assumed to be similar in some of the cases we analyze. Therefore, we do not have to consider the effect of information revelation in this multi-stage game.

The last rule of the auction is that agents may not subtract bids. This means that, if its utility drops in a later round below the current bid price (which we will denote Q), the agent cannot withdraw its previous bid, but it can adjust it to the current bid price. The effect of this is that, if the price Q is high enough that fewer than m agents have utilities $\tilde{u}_i \geq Q$, the rest of the rooms are sold to a random selection of the winners of the previous round which have $\tilde{u}_i < Q$, and these agents lose money. This may seem, at a first glance, to be different from the TAC rules for hotel auctions, but it is exactly what happens. The reason for this is that, in the case of ties, the bids that win are the earlier ones, and the agents will all try to bid as late as possible, if there is a chance that they

will have to buy some room at a loss. This means that every agent that is in this situation has the same probability of being a “winner”, and forced to buy an undesired room. Therefore, the unwanted rooms are sold to a random selection of the agents which were winners of the previous round and now (in the current round), their utility has dropped and is less than the current price Q .

We will use the following functions in the theorems: (multi-unit case)

$$\begin{aligned}\Phi(x) &= \sum_{i=0}^{m-1} C(N-1, i) \cdot (F(x))^{N-1-i} \cdot (1-F(x))^i \\ Y(x) &= \sum_{i=0}^{m-2} C(N-1, i) \cdot (F(x))^{N-1-i} \cdot (1-F(x))^i \\ \tilde{\Phi}(x) &= \sum_{i=0}^{m-1} C(N-1, i) \cdot (H(x))^{N-1-i} \cdot (1-H(x))^i \\ \tilde{Y}(x) &= \sum_{i=0}^{m-2} C(N-1, i) \cdot (H(x))^{N-1-i} \cdot (1-H(x))^i\end{aligned}$$

In the case that $m = 1$, it is

$$\begin{aligned}\Phi(x) &= (F(x))^{N-1} \\ \tilde{\Phi}(x) &= (H(x))^{N-1}\end{aligned}$$

and

$$Y(x) = \tilde{Y}(x) = 0$$

6.1.2 Useful Mathematical Knowledge

Before we proceed with our analysis, we need the following information found in numerical analysis textbooks (see [2]).

Theorem 2. *Let $f(x, z)$ and $\frac{\partial f(x, z)}{\partial z}$ be continuous functions of x and z at all points (x, z) in some neighborhood of the initial point (x_0, Y_0) . Then there is a unique function*

$Y(x)$ defined on some interval $[x_0 - \alpha, x_0 + \alpha]$, satisfying

$$Y'(x) = f(x, Y(x)), \quad \forall x : x_0 - \alpha \leq x \leq x_0 + \alpha \quad \text{and}$$

$$Y(x_0) = Y_0$$

All the equilibria when $p \neq 1$ are the solutions of differential equations of the form described by Theorem 2. This theorem guarantees the existence and unique solvability of the initial value problem for those differential equations, which in turn, means that the equilibrium does exist and is unique. We may not know a closed form solution, but a numerical solution can be easily calculated. The method that we decided to use is a Runge-Kutta method with variable step size; this is one of the most commonly used methods. More specifically, we used method `ode45` in `Matlab`. The requirement for a Runge-Kutta method in general is that the function $f(x, z)$ and several (this depends on the order of the Runge-Kutta method) of its derivatives be continuous in the interval for which the solution is computed.

We will also make use of the following lemmas:

Lemma 1. *If random variables $X_i, \forall i \in \{1, \dots, N\}$ are i.i.d. with probability distribution $f(x) = \text{Prob}[X_i \leq x]$ and $Y_N^{(k)}$ denotes the k^{th} order statistic of the variables X_i , then*

$$\text{Prob}[Y_N^{(k)} \leq y] = \sum_{i=0}^{k-1} C(N, i) \cdot (f(y))^{N-i} \cdot (1 - f(y))^i$$

where $C(N, i)$ is the total number of possible combinations of i items chosen from N .

Lemma 2. *If random variables $X_i, \forall i \in \{1, \dots, N\}$ are i.i.d. with probability distribution $f(x) = \text{Prob}[X_i \leq x]$, then the probability p_k that exactly k of these variables $X_i \geq T$ is*

$$p_k = C(N, k) \cdot (f(T))^{N-k} \cdot (1 - f(T))^k$$

Proof. Both of these lemmas can be found in any probabilities book (e.g. [74]).

Lemma 3. $\forall k, N, M \geq 0$ (non negative integers), such that $k \leq M \leq N$ the following formula holds

$$C(N, M) = \sum_{j=0}^k C(k, j) \cdot C(N - k, M - j)$$

Proof. Basic combinatorics lemma.

Lemma 4. A function $g(u)$ that satisfies the equation

$(u - g(u)) \cdot \frac{\Phi'(u)}{g'(u)} = T(u)$ is the following

$$g(u) = u - \frac{\Omega(u)}{T(u)} \cdot \int_C \frac{T(\omega)}{\Omega(\omega)} \cdot d\omega$$

where $\Omega(u) = e^{\int_D \frac{T(\omega) - \Phi'(\omega)}{T(\omega)} \cdot d\omega}$, C depends on the boundary conditions and D can have any value.

Proof. All that is needed to do is to replace $g(u)$ in the differential equation with the formula provided and check that the two sides of the equation are indeed equal.

6.1.3 Bayes-Nash Equilibria For a Single Unit Auction

Before analyzing the general case, we will compute equilibria for the first price auction. Therefore, in this section $m = 1$ and the single unit is sold to the agent which submitted the highest bid at a price equal to his bid. For Theorem 4, we assume that in the second round, the utilities are drawn from $F(u)$ and that $\tilde{u}_i \simeq u_i$. Each agent i submits a bid v_i in the first round. It is $Q = 0$, if no bids were placed, which is the case at the beginning of the first round, whereas $Q > 0$ equals the current bid price in the beginning of the second round. We compute a Bayes-Nash equilibrium $g(u)$ that maps utilities u_i to bids v_i .

In the case of $p = 1$ (only one round), and $Q = 0$, we know from auction theory (e.g. [44]) that each risk-neutral agent i with valuation u_i should bid

$$g(u_i) = u_i - \frac{1}{(F(u_i))^{N-1}} \cdot \int_0^{u_i} (F(\omega))^{N-1} \cdot d\omega \quad (6.1)$$

Theorem 3. *If the starting price is $Q \geq 0$ and the bidding lasts for exactly one round ($p = 1$), the equilibrium strategy is*

$$g(u_i) = u_i - \frac{\int_Q^{u_i} (F(\omega))^{N-1} \cdot d\omega}{(F(u_i))^{N-1}} \quad (6.2)$$

Proof. Since some bids have already been placed, then the current price $Q > 0$ and thus (i) some agents might have stopped participating in the auction, since the current price Q exceeds their private valuation u_i , and (ii) the probability distribution of the valuations $F(u)$ has changed, since now we know that the valuation of agents that still participate is $u_i \geq Q$. The new probability distribution is

$$F_Q(u) = \text{Prob}[U \leq u | U \geq Q] = \frac{\text{Prob}[U \leq u \wedge U \geq Q]}{\text{Prob}[U \geq Q]} = \frac{F(u) - F(Q)}{1 - F(Q)}$$

Therefore,

$$F_Q(u) = \frac{F(u) - F(Q)}{1 - F(Q)}, \text{ if } u \geq Q \ \& \ F_Q(u) = 0, \text{ if } u < Q \quad (6.3)$$

We also know the probability π_k of the event that exactly k agents participate in the auction at price Q ; it is the probability that exactly $k - 1$ of the other agents' valuations¹ u_i are $u_i \geq Q$, which is (see Lemma 2).

$$\pi_k = C(N - 1, k - 1) \cdot (F(Q))^{N-k} \cdot (1 - F(Q))^{k-1} \quad (6.4)$$

Let us assume that it is a Bayes-Nash equilibrium for each agent i to bid $v_i = g(u_i)$ and $g(u)$ is non-decreasing in u , meaning that higher valuations lead to higher bids. Each

¹Because, from the point-of-view of a participating agent, it does not know whether the other $N - 1$ agents participate.

agent submits its bid $v_i \geq Q$ and then the auction closes and the high bidder pays its bid and gets the item. Therefore, the expected utility $EU_i(v_i)$ of agent i , when it places a bid of v_i , can be computed by taking the expected utility conditional on the number of agents k that participate at the auction at price Q . If k agents participate, then the utility of agent i is 0, if it does not have the highest bid and $(u_i - v_i)$ if it does. Thus,

$$EU_i(v_i | \#agents = k) = (u_i - v_i) \cdot Prob\left[\bigwedge_{j \neq i} v_i \geq v_j\right] =$$

$$(u_i - v_i) \cdot \prod_{j \neq i} Prob[v_i \geq v_j]$$

Since $v_j = g(u_j)$, it follows that

$$Prob[v_i \geq v_j] = Prob[v_i \geq g(u_j)] = Prob[g^{-1}(v_i) \geq u_j] = F_Q(g^{-1}(v_i))$$

and therefore,

$$EU_i(v_i | \#agents = k) = (u_i - v_i) \cdot (F_Q(g^{-1}(v_i)))^{k-1} \quad (6.5)$$

Hence,

$$EU_i(v_i) = \sum_{k=1}^N \pi_k \cdot EU_i(v_i | \#agents = k) \Rightarrow$$

$$EU_i(v_i) = (u_i - v_i) \cdot \sum_{k=1}^N \pi_k \cdot (F_Q(g^{-1}(v_i)))^{k-1} \quad (6.6)$$

Let us set,

$$\Phi(x) = \sum_{k=1}^N \pi_k \cdot (F_Q(x))^{k-1}$$

Then for $x \geq Q$ it is

$$\Phi(x) = \sum_{k=1}^N \text{big}\{C(N-1, k-1) \cdot (F(Q))^{N-k} \cdot (1-F(Q))^{k-1}\} \cdot \left(\frac{F(x) - F(Q)}{1 - F(Q)}\right)^{k-1} =$$

$$\sum_{k=1}^N \{C(N-1, k-1) \cdot (F(Q))^{N-k} \cdot (F(x) - F(Q))^{k-1}\} \Leftrightarrow$$

$$\Phi(x) = (F(x))^{N-1}$$

If $x < Q$, it is

$$\Phi(x) = 0$$

because for $x < Q$, $F_Q(x) = 0$. Thus²

$$EU_i(v_i) = (u_i - v_i) \cdot \Phi(g^{-1}(v_i)) \quad (6.7)$$

The bid v_i that maximizes $EU_i(v_i)$ can be found by setting

$$\begin{aligned} \frac{dEU_i(v_i)}{dv_i} = 0 &\Leftrightarrow \\ -\Phi(g^{-1}(v_i)) + (u_i - v_i) \cdot \Phi'(g^{-1}(v_i)) \cdot (g^{-1}(v_i))' &= 0 \Leftrightarrow \\ (u_i - v_i) \cdot \Phi'(g^{-1}(v_i)) \cdot \frac{1}{g'(g^{-1}(v_i))} &= \Phi(g^{-1}(v_i)) \end{aligned}$$

Since we assumed that the optimal solution is $v_i = g(u_i)$, the previous equation becomes

$$(u_i - g(u_i)) \cdot \Phi'(u_i) \cdot \frac{1}{g'(u_i)} = \Phi(u_i)$$

The function $g(u)$ that satisfies this equation is (use Lemma 4 with $T(u) = \Phi(u)$)

$$g(u) = u - \frac{1}{\Phi(u)} \cdot \int_C^u \Phi(\omega) \cdot d\omega$$

and, since the boundary condition is $g(Q) = Q$,

$$g(u_i) = u_i - \frac{\int_Q^{u_i} (F(\omega))^{N-1} \cdot d\omega}{(F(u_i))^{N-1}}$$

Corollary 1. *The expected utility is*

$$\mathcal{U}(u_i, Q) = \int_Q^{u_i} (F(\omega))^{N-1} \cdot d\omega \quad (6.8)$$

²Note that this equation is the same as in the case that $Q = 0$.

Proof. The expected utility of agent i is

$$\mathcal{U}(u_i, Q) = EU_i(v_i)|_{v_i=g(u_i)}$$

and using Equations (6.7) and (6.2), we get

$$\begin{aligned} \mathcal{U}(u_i, Q) &= (u_i - g(u_i)) \cdot \Phi(u_i) = \\ &= \frac{\int_Q^{u_i} (F(\omega))^{N-1} \cdot d\omega}{(F(u_i))^{N-1}} \cdot (F(u_i))^{N-1} \Leftrightarrow \\ \mathcal{U}(u_i, Q) &= \int_Q^{u_i} (F(\omega))^{N-1} \cdot d\omega \end{aligned}$$

Note that $\mathcal{U}(u_i, Q) = 0, \forall Q > u_i$, since the agent does not participate in that case.

Theorem 4. *If the starting price is $Q = 0$, a second round of bidding exists with probability $(1 - p)$ ($p \neq 0, 1$) and the utility of the agents in the second round is drawn from the same distribution $F(u)$ (and each agent i in fact has utility of a similar value to the utility u_i of the first round), then the equilibrium strategy is the solution of the differential equation*

$$(u_i - g(u_i)) \cdot \frac{\Phi'(u_i)}{g'(u_i)} = \Phi(u_i) \cdot \Psi(g(u_i)) \quad (6.9)$$

where

$$\begin{aligned} \Phi(x) &= (F(x))^{N-1} \\ \Psi(x) &= 1 + \frac{1-p}{p} \cdot (F(x))^{N-1} \end{aligned}$$

and the boundary condition is $g(0) = 0$.

Proof. If the auction closes at the first round, then the expected utility is

$$U_i^{(1)} = (u_i - v_i) \cdot (F(g^{-1}(v_i)))^{N-1}$$

(see derivation of Equation 6.5). Once again, we assume that the strategy $v_i = g(u_i)$ is a Bayes-Nash equilibrium for all agents at the first round.

To compute the expected utility after the second round $U_i^{(2)}$, the price Q after the first round must be computed. The probability distribution of an agent's bid is:

$$Prob[V \leq v] = Prob[g(U) \leq v] = Prob[U \leq g^{-1}(v)] = F(g^{-1}(v))$$

Therefore, the probability distribution of the highest bid $B^{(1)}$ among all other $N - 1$ agents is

$$Prob[B^{(1)} \leq v] = (F(g^{-1}(v)))^{N-1}$$

(see Lemma 1).

The price $Q = v_i$, if $v_i \geq B^{(1)}$

and $Q = B^{(1)}$, if $v_i < B^{(1)}$.

Hence,

$$Prob[Q < v_i] = 0$$

$$Prob[Q = v_i] = Prob[v_i \geq B^{(1)}] = (F(g^{-1}(v_i)))^{N-1}$$

and

$$Prob[Q \leq v] = Prob[B^{(1)} \leq v] = (F(g^{-1}(v)))^{N-1}, \forall v > v_i$$

As a result, we can now compute $U_i^{(2)}$:

$$\begin{aligned} U_i^{(2)} &= \int_0^{+\infty} \mathcal{U}(\omega) \cdot Prob[Q = \omega] \cdot d\omega = 0 + \mathcal{U}(v_i) \cdot Prob[Q = v_i] + \\ &\int_{v_i^+}^{+\infty} \mathcal{U}(\omega) \cdot Prob[Q = \omega] \cdot d\omega = \mathcal{U}(v_i) \cdot (F(g^{-1}(v_i)))^{N-1} + \\ &\int_{v_i^+}^{u_i} \mathcal{U}(\omega) \cdot \frac{d}{d\omega} (F(g^{-1}(\omega)))^{N-1} \cdot d\omega \end{aligned}$$

Let us set

$$\Phi(x) = (F(x))^{N-1}$$

Then,

$$U_i^{(1)} = (u_i - v_i) \cdot \Phi(g^{-1}(v_i))$$

and

$$U_i^{(2)} = \mathcal{U}(v_i) \cdot \Phi(g^{-1}(v_i)) + \int_{v_i}^{u_i} \mathcal{U}(\omega) \cdot \frac{d}{d\omega} \Phi(g^{-1}(\omega)) \cdot d\omega$$

The expected utility is, therefore,

$$U_i(v_i) = p \cdot U_i^{(1)} + (1-p) \cdot U_i^{(2)} =$$

$$p \cdot (u_i - v_i) \cdot \Phi(g^{-1}(v_i)) + (1-p) \cdot \mathcal{U}(v_i) \cdot \Phi(g^{-1}(v_i)) + (1-p) \cdot \int_{v_i}^{u_i} \mathcal{U}(\omega) \cdot \frac{d}{d\omega} \Phi(g^{-1}(\omega)) \cdot d\omega \Leftrightarrow$$

$$U_i(v_i) = (p \cdot (u_i - v_i) + (1-p) \cdot \mathcal{U}(v_i)) \cdot \Phi(g^{-1}(v_i)) + (1-p) \cdot \int_{v_i}^{u_i} \mathcal{U}(\omega) \cdot \frac{d}{d\omega} \Phi(g^{-1}(\omega)) \cdot d\omega$$

and the bid v_i that maximizes $U_i(v_i)$ can be found by setting

$$\frac{dU_i(v_i)}{dv_i} = 0 \Leftrightarrow$$

$$(-p + (1-p) \cdot \mathcal{U}'(v_i)) \cdot \Phi(g^{-1}(v_i)) + p \cdot (u_i - v_i) \cdot \frac{d}{dv_i} \Phi(g^{-1}(v_i)) = 0 \Leftrightarrow$$

$$(u_i - v_i) \cdot \frac{\Phi'(g^{-1}(v_i))}{g'(g^{-1}(v_i))} = (1 - \frac{1-p}{p} \cdot \mathcal{U}'(v_i)) \cdot \Phi(g^{-1}(v_i))$$

We also have that $v_i = g(u_i)$, and it should also be noted that the function \mathcal{U} is actually a function both of Q and u_i , so $\mathcal{U}'(v_i)$ in the above equation is actually

$$\frac{\partial \mathcal{U}(u_i, Q)}{\partial Q} \Big|_{Q=v_i} = -\Phi(v_i) = -\Phi(g(u_i))$$

and, therefore, we get

$$(u_i - g(u_i)) \cdot \frac{\Phi'(u_i)}{g'(u_i)} = (1 + \frac{1-p}{p} \cdot \Phi(g(u_i))) \cdot \Phi(u_i)$$

and if we set

$$\Psi(u, Q) = 1 - \frac{1-p}{p} \cdot \frac{\partial \mathcal{U}(u_i, Q)}{\partial Q}$$

then the equation we need to solve is

$$(u_i - g(u_i)) \cdot \frac{\Phi'(u_i)}{g'(u_i)} = \Phi(u_i) \cdot \Psi(g(u_i)) \quad (6.10)$$

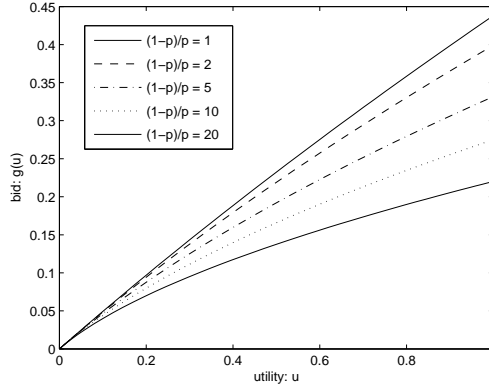


Figure 6.1: The graphs from top to bottom represent the solution of Equation 6.11, when $\frac{1-p}{p}$ takes values 1, 2, 5, 10 and 20 respectively.

As a special case, we can examine this equation when only $N = 2$ agents participate and their valuations $u_i \sim U[0, 1]$, which means that $F(u) = u, \forall u \in [0, 1]$. We need to compute $v_i = g_p(u_i), \forall u_i \in [0, 1]$. Equation 6.9 becomes:

$$g'_p(u_i) = \frac{u_i - g_p(u_i)}{u_i \cdot \left(1 + \frac{1-p}{p} \cdot g_p(u_i)\right)} \quad (6.11)$$

Even this equation, which is the simplest form that we can have for the two round auction, has no known closed form solution. In Figure 6.1, we graph the solution for various values of $\frac{1-p}{p}$. We provide this figure in order to contrast with the Bayes-Nash equilibrium, for the case that $p = 1$, which is $g_1(u_i) = \frac{u_i}{2}$. One may notice that as the probability $(1 - p)$ of a second round increases, the equilibrium strategy suggests that the agent should bid less. However, we can at least remove the parameter p from this computation. We can easily verify that

$$g_p(u) = \frac{p}{1-p} \cdot \tilde{g}\left(\frac{1-p}{p} \cdot u\right)$$

where $\tilde{g}(u)$ is the solution of differential equation

$$\tilde{g}'(u) = \frac{u - \tilde{g}(u)}{u \cdot (1 + \tilde{g}(u))}$$

6.1.4 Bayes-Nash Equilibria For a Multi-unit Auction

In this section, we examine the general case of an auction where m identical goods ($m > 1$) are sold and the goods are sold to the agents which submitted the m highest bids at a price equal to the lowest winning bid. For Theorem 7, we assume that in the second round, the utilities are drawn from $F(u)$ and that $\tilde{u}_i \simeq u_i$. In Theorem 8, we assume that the agent knows that its own utility \tilde{u}_i in the second round, is drawn from $G(u)$ and everyone else's from $H(u)$; this is the most general case that we can examine for the two-round auction. We will use the equations from Theorems 7 and 8 in Section 6.2, in order to provide the equilibria for the R -round auction.

Theorem 5. *If the starting price is $Q = 0$, and the bidding lasts for exactly one round ($p = 1$), the equilibrium strategy is*

$$g(u) = u - \frac{e^{\int_0^u \frac{-Y'(\omega)}{\Phi(\omega)-Y(\omega)} \cdot d\omega}}{\Phi(u) - Y(u)} \cdot \int_0^u \frac{\Phi(z) - Y(z)}{e^{\int_0^z \frac{-Y'(\omega)}{\Phi(\omega)-Y(\omega)} \cdot d\omega}} \cdot dz \quad (6.12)$$

Proof. The probability distribution of an agent's bid is:

$$Prob[V \leq v] = Prob[g(U) \leq v] = Prob[U \leq g^{-1}(v)] = F(g^{-1}(v))$$

Therefore, the probability distribution of the $(m-1)^{th}$ and m^{th} highest bids ($m > 1$), named $B^{(m-1)}$ and $B^{(m)}$ respectively, among all other $N-1$ agents are:

$$Prob[B^{(m-1)} \leq v] = \sum_{i=0}^{m-2} C(N-1, i) \cdot (F(g^{-1}(v)))^{N-1-i} \cdot (1 - F(g^{-1}(v)))^i$$

and

$$Prob[B^{(m)} \leq v] = \sum_{i=0}^{m-1} C(N-1, i) \cdot (F(g^{-1}(v)))^{N-1-i} \cdot (1 - F(g^{-1}(v)))^i$$

(see lemma 1).

Then,

$$Prob[B^{(m-1)} \leq v] = Y(g^{-1}(v))$$

and

$$Prob[B^{(m)} \leq v] = \Phi(g^{-1}(v))$$

If $B^{(m)} > v_i$ then the agent gets utility 0 (does not win).

If $B^{(m-1)} > v_i \geq B^{(m)}$, then the agent submitted the m^{th} price, so it gets 1 unit (of the m available units) and pays v_i getting a utility of $u_i - v_i$.

If $v_i \geq B^{(m-1)}$, then the agent gets 1 unit and pays $B^{(m-1)}$ (which is now the m^{th} price) getting a utility of $u_i - B^{(m-1)}$.

Hence, the expected utility is

$$\begin{aligned} EU_i(v_i) &= (u_i - v_i) \cdot Prob[B^{(m-1)} > v_i \geq B^{(m)}] + \int_0^{v_i} (u_i - \omega) \cdot Prob[B^{(m-1)} = \omega] \cdot d\omega = \\ &= (u_i - v_i) \cdot (Prob[B^{(m)} \leq v_i] - Prob[B^{(m-1)} \leq v_i]) + \int_0^{v_i} (u_i - v_i) \cdot Prob[B^{(m-1)} = \omega] \cdot d\omega \\ &\quad + \int_0^{v_i} (v_i - \omega) \cdot Prob[B^{(m-1)} = \omega] \cdot d\omega = \\ &= (u_i - v_i) \cdot (Prob[B^{(m)} \leq v_i] - Prob[B^{(m-1)} \leq v_i]) + (u_i - v_i) \cdot Prob[B^{(m-1)} \leq v_i] \\ &\quad + \int_0^{v_i} (v_i - \omega) \cdot \frac{d}{d\omega} Prob[B^{(m-1)} \leq \omega] \cdot d\omega = \\ &\quad (u_i - v_i) \cdot Prob[B^{(m)} \leq v_i] \\ &\quad + v_i \cdot \int_0^{v_i} \frac{d}{d\omega} Prob[B^{(m-1)} \leq \omega] \cdot d\omega - \int_0^{v_i} \omega \cdot \frac{d}{d\omega} Prob[B^{(m-1)} \leq \omega] \cdot d\omega \Leftrightarrow \\ EU_i(v_i) &= (u_i - v_i) \cdot \Phi(g^{-1}(v_i)) + v_i \cdot Y(g^{-1}(v_i)) - \int_0^{v_i} \omega \cdot (Y(g^{-1}(\omega)))' \cdot d\omega \end{aligned}$$

Note that,

$$\int_0^{v_i} \omega \cdot (Y(g^{-1}(\omega)))' \cdot d\omega = v_i \cdot Y(g^{-1}(v_i)) - \int_0^{v_i} (\omega)' \cdot Y(g^{-1}(\omega)) \cdot d\omega$$

thus,

$$EU_i(v_i) = (u_i - v_i) \cdot \Phi(g^{-1}(v_i)) + \int_0^{v_i} Y(g^{-1}(\omega)) \cdot d\omega \quad (6.13)$$

The bid v_i that maximizes $EU_i(v_i)$ can be found by setting

$$\frac{dEU_i(v_i)}{dv_i} = 0 \Leftrightarrow$$

$$-\Phi(g^{-1}(v_i)) + (u_i - v_i) \cdot \frac{\Phi'(g^{-1}(v_i))}{g'(g^{-1}(v_i))} + Y(g^{-1}(v_i)) = 0$$

Since $v_i = g(u_i)$, the previous equation becomes

$$-\Phi(u_i) + (u_i - g(u_i)) \cdot \frac{\Phi'(u_i)}{g'(u_i)} + Y(u_i) = 0$$

The function $g(u)$ that satisfies this equation considering the boundary condition $g(0) = 0$ is (use Lemma 4 with $T(u) = \Phi(u) - \Psi(u)$)

$$g(u) = u - \frac{e^{\int_0^u \frac{-Y'(\omega)}{\Phi(\omega) - Y(\omega)} \cdot d\omega}}{\Phi(u) - Y(u)} \cdot \int_0^u \frac{\Phi(z) - Y(z)}{e^{\int_0^z \frac{-Y'(\omega)}{\Phi(\omega) - Y(\omega)} \cdot d\omega}} \cdot dz$$

Theorem 6. *If the starting price is $Q \geq 0$, and the bidding lasts for exactly one round ($p = 1$), the equilibrium strategy is*

$$g(u) = u - \frac{e^{\int_Q^u \frac{-Y'(\omega)}{\Phi(\omega) - Y(\omega)} \cdot d\omega}}{\Phi(u) - Y(u)} \cdot \int_Q^u \frac{\Phi(z) - Y(z)}{e^{\int_Q^z \frac{-Y'(\omega)}{\Phi(\omega) - Y(\omega)} \cdot d\omega}} \cdot dz \quad (6.14)$$

Proof. $Q \geq 0$ because some bids may have already been placed, and thus, (i) some agents might have stopped participating in the auction, since the current price Q exceeds their private valuation u_i , and (ii) the probability distribution of the valuations $F(u)$ has changed, since now we know that the valuation of agents that still participate is $u_i \geq Q$.

The new probability distribution is given once more from Equation 6.3

$$F_Q(u) = \frac{F(u) - F(Q)}{1 - F(Q)}, \text{ if } u \geq Q \ \& \ F_Q(u) = 0, \text{ if } u < Q$$

We also know the probability π_k of the event that exactly $k \in [1, N]$ agents participate in the auction at price Q ; it is the probability that exactly $k - 1$ of the other agents' valuations³ u_i are $u_i \geq Q$, which is once more given by Equation 6.4

$$\pi_k = C(N - 1, k - 1) \cdot (F(Q))^{N-k} \cdot (1 - F(Q))^{k-1}$$

³Because from the point-of-view of a participating agent, it does not know whether the other $N - 1$ agents participate.

The probability distribution of an agent's bid is:

$$Prob[V \leq v] = Prob[g(U) \leq v] = Prob[U \leq g^{-1}(v)] = F(g^{-1}(v))$$

Therefore, the probability distribution of the $(m - 1)^{th}$ and m^{th} highest bids ($m > 1$), named $B^{(m-1)}$ and $B^{(m)}$ respectively, among all other $N - 1$ agents are:

$$Prob[B^{(m-1)} \leq v] = \widehat{Y}(g^{-1}(v))$$

and

$$Prob[B^{(m)} \leq v] = \widehat{\Phi}(g^{-1}(v))$$

where

$$\widehat{\Phi}(x) = \sum_{i=0}^{m-1} C(k-1, i) \cdot (F_Q(x))^{k-1-i} \cdot (1 - F_Q(x))^i$$

and

$$\widehat{Y}(x) = \sum_{i=0}^{m-2} C(k-1, i) \cdot (F_Q(x))^{k-1-i} \cdot (1 - F_Q(x))^i$$

(see Lemma 1).

If $k \leq m$, then the expected utility is:

$$EU_i(v_i | \#agents = k) = u_i - Q$$

If that is not the case, meaning that $k \geq m$ and $B^{(m-1)} \geq Q$, then:

If $B^{(m)} > v_i$, then the agent gets utility 0 (does not win).

If $B^{(m-1)} > v_i \geq B^{(m)}$, then the agent submitted the m^{th} price, so it gets 1 unit (of the m available units) and pays v_i getting a utility of $u_i - v_i$.

If $v_i \geq B^{(m-1)} \geq Q$, then the agent gets 1 unit and pays the m^{th} price, which is $B^{(m-1)}$, getting a utility of $u_i - B^{(m-1)}$.

The expected utility is

$$EU_i(v_i | \#agents = k) = (u_i - v_i) \cdot Prob[B^{(m-1)} > v_i \geq B^{(m)}] + \int_Q^{v_i} (u_i - \omega) \cdot Prob[B^{(m-1)} = \omega] \cdot d\omega =$$

$$\begin{aligned}
& (u_i - v_i) \cdot (\text{Prob}[B^{(m)} \leq v_i] - \text{Prob}[B^{(m-1)} \leq v_i]) \\
& + \int_Q^{v_i} (u_i - v_i) \cdot \text{Prob}[B^{(m-1)} = \omega] \cdot d\omega + \int_Q^{v_i} (v_i - \omega) \cdot \text{Prob}[B^{(m-1)} = \omega] \cdot d\omega = \\
& (u_i - v_i) \cdot (\text{Prob}[B^{(m)} \leq v_i] - \text{Prob}[B^{(m-1)} \leq v_i]) \\
& + (u_i - v_i) \cdot \text{Prob}[B^{(m-1)} \leq v_i] + \int_Q^{v_i} (v_i - \omega) \cdot \frac{d}{d\omega} \text{Prob}[B^{(m-1)} \leq \omega] \cdot d\omega = \\
& (u_i - v_i) \cdot \text{Prob}[B^{(m)} \leq v_i] + v_i \cdot \int_Q^{v_i} \frac{d}{d\omega} \text{Prob}[B^{(m-1)} \leq \omega] \cdot d\omega \\
& - \int_Q^{v_i} \omega \cdot \frac{d}{d\omega} \text{Prob}[B^{(m-1)} \leq \omega] \cdot d\omega \Leftrightarrow \\
EU_i(v_i | \#agents = k) &= (u_i - v_i) \cdot \widehat{\Phi}(g^{-1}(v_i)) + v_i \cdot \widehat{Y}(g^{-1}(v_i)) - \int_Q^{v_i} \omega \cdot (\widehat{Y}(g^{-1}(\omega)))' \cdot d\omega
\end{aligned}$$

Note that

$$\int_Q^{v_i} \omega \cdot (\widehat{Y}(g^{-1}(\omega)))' \cdot d\omega = v_i \cdot \widehat{Y}(g^{-1}(v_i)) - \int_Q^{v_i} (\omega)' \cdot \widehat{Y}(g^{-1}(\omega)) \cdot d\omega$$

thus,

$$EU_i(v_i | \#agents = k) = (u_i - v_i) \cdot \widehat{\Phi}(g^{-1}(v_i)) + \int_Q^{v_i} \widehat{Y}(g^{-1}(\omega)) \cdot d\omega \quad (6.15)$$

This equation also covers the case that $k \leq m$, since then it is

$$\widehat{\Phi}(u) = \widehat{Y}(u) = 1, \forall u \geq Q$$

The total expected utility is therefore:

$$\begin{aligned}
EU_i(v_i) &= \sum_{k=1}^N \pi_k \cdot EU_i(v_i | \#agents = k) \Rightarrow \\
EU_i(v_i) &= (u_i - v_i) \cdot \Phi(g^{-1}(v_i)) + \int_Q^{v_i} Y(g^{-1}(\omega)) \cdot d\omega \quad (6.16)
\end{aligned}$$

where

$$\Phi(x) = \sum_{k=1}^N \pi_k \cdot \sum_{i=0}^{m-1} C(k-1, i) \cdot (F_Q(x))^{k-1-i} \cdot (1 - F_Q(x))^i =$$

$$\sum_{i=0}^{m-1} \sum_{k=i+1}^N \pi_k \cdot C(k-1, i) \cdot (F_Q(x))^{k-1-i} \cdot (1 - F_Q(x))^i$$

and

$$\begin{aligned} Y(x) &= \sum_{k=1}^N \pi_k \cdot \sum_{i=0}^{m-2} C(k-1, i) \cdot (F_Q(x))^{k-1-i} \cdot (1 - F_Q(x))^i = \\ &= \sum_{i=0}^{m-2} \sum_{k=i+1}^N \pi_k \cdot C(k-1, i) \cdot (F_Q(x))^{k-1-i} \cdot (1 - F_Q(x))^i \end{aligned}$$

It is $C(k-1, i) = 0$, if $k \leq i$, and this is the reason why we changed the lower bound of the sum for k from 1 to $i+1$.

We will use the fact that $C(k-1, i) \cdot C(N-1, k-1) = C(N-1-i, N-k) \cdot C(N-1, i)$.

Substituting from Equations 6.3 and 6.4 and for any $i \in [0, m-1]$ and $x \geq Q$ it is:

$$\begin{aligned} & \sum_{k=i+1}^N \pi_k \cdot C(k-1, i) \cdot (F_Q(x))^{k-1-i} \cdot (1 - F_Q(x))^i = \\ & \sum_{k=i+1}^N C(k-1, i) \cdot C(N-1, k-1) \cdot (F(Q))^{N-k} \cdot (1 - F(Q))^{k-1} \cdot \\ & \quad \left(\frac{F(x) - F(Q)}{1 - F(Q)} \right)^{k-1-i} \cdot \left(\frac{1 - F(x)}{1 - F(Q)} \right)^i \\ &= \sum_{k=i+1}^N C(N-1-i, N-k) \cdot C(N-1, i) \cdot (F(Q))^{N-k} \cdot (F(x) - F(Q))^{k-1-i} \cdot (1 - F(x))^i \\ &= C(N-1, i) \cdot (1 - F(x))^i \cdot \sum_{k=i+1}^N C(N-1-i, N-k) \cdot (F(Q))^{N-k} \cdot (F(x) - F(Q))^{k-1-i} \\ & \quad = C(N-1, i) \cdot (1 - F(x))^i \\ & \quad \cdot \sum_{\lambda=0}^{N-i-1} C(N-1-i, N-i-1-\lambda) \cdot (F(Q))^{N-i-1-\lambda} \cdot (F(x) - F(Q))^\lambda \\ & \quad = C(N-1, i) \cdot (1 - F(x))^i \cdot (F(x))^{N-1-i} \end{aligned}$$

Therefore,

$$\Phi(x) = \sum_{i=0}^{m-1} C(N-1, i) \cdot (F(x))^{N-1-i} \cdot (1 - F(x))^i$$

and

$$Y(x) = \sum_{i=0}^{m-2} C(N-1, i) \cdot (F(x))^{N-1-i} \cdot (1-F(x))^i$$

Note if $x < Q$, then $\Phi(x) = Y(x) = 0$ and thus $EU_i(v_i) = 0$.

The bid v_i that maximizes $EU_i(v_i)$ can be found by setting

$$\frac{dEU_i(v_i)}{dv_i} = 0 \Leftrightarrow -\Phi(g^{-1}(v_i)) + (u_i - v_i) \cdot \frac{\Phi'(g^{-1}(v_i))}{g'(g^{-1}(v_i))} + Y(g^{-1}(v_i)) = 0$$

Since $v_i = g(u_i)$, the previous equation becomes

$$-\Phi(u_i) + (u_i - g(u_i)) \cdot \frac{\Phi'(u_i)}{g'(u_i)} + Y(u_i) = 0$$

The function $g(u)$ that satisfies this equation, given that the boundary condition is $g(Q) = Q$ is (use lemma 4 with $T(u) = \Phi(u) - \Psi(u)$)

$$g(u) = u - \frac{e^{\int_Q^u \frac{-Y'(\omega)}{\Phi(\omega) - Y(\omega)} \cdot d\omega}}{\Phi(u) - Y(u)} \cdot \int_Q^u \frac{\Phi(z) - Y(z)}{e^{\int_Q^z \frac{-Y'(\omega)}{\Phi(\omega) - Y(\omega)} \cdot d\omega}} \cdot dz$$

Note that Equation 6.16 can be written as

$$EU_i(v_i) = (u_i - v_i) \cdot \Phi(g^{-1}(v_i)) + \int_Q^{g^{-1}(v_i)} Y(u) \cdot g'(u) \cdot du$$

and the maximal expected utility $\mathcal{U}(u_i, Q)$ of agent i is therefore,

$$\begin{aligned} \mathcal{U}(u_i, Q) &= EU_i(v_i) \Big|_{v_i=g(u_i)} \Leftrightarrow \\ \mathcal{U}(u_i, Q) &= (u_i - g(u_i)) \cdot \Phi(u_i) + \int_Q^{u_i} Y(\omega) \cdot g'(\omega) \cdot d\omega \end{aligned} \quad (6.17)$$

We will use this utility in the next theorems.

Theorem 7. *If the starting price is $Q_s \geq 0$, a second round of bidding exists with probability $(1-p)$ ($p \neq 0, 1$), the utility of the agents in the second round is drawn from distribution $H(u)$, and each agent i , in fact, has utility in the second round $\tilde{u}_i \simeq u_i$, that*

is similar to the utility in the first round. Then the equilibrium strategy is the solution of the differential equation

$$(u_i - g(u_i)) \cdot \frac{\Phi'(u_i)}{g'(u_i)} = (\Phi(u_i) - Y(u_i)) \cdot \Psi(u_i, g(u_i)) \quad (6.18)$$

where

$$\Psi(u_i, Q) = 1 + \frac{1-p}{p} \cdot \frac{\tilde{\Phi}(Q) - \tilde{Y}(Q)}{\tilde{\Phi}(u_i) - \tilde{Y}(u_i)} \cdot e^{\int_Q^{u_i} \frac{-\tilde{Y}'(\omega)}{\tilde{\Phi}(\omega) - \tilde{Y}(\omega)} \cdot d\omega} \cdot \left(\tilde{\Phi}(u_i) + \int_Q^{u_i} \frac{\tilde{Y}(\omega) \cdot \tilde{\Phi}'(\omega)}{\tilde{\Phi}(\omega) - \tilde{Y}(\omega)} \cdot d\omega \right)$$

and the boundary condition is $g(Q_s) = Q_s$.

Proof. If the auction closes at the first round, then the expected utility is given by Equation 6.16

$$U_i^{(1)}(v_i) = (u_i - v_i) \cdot \Phi(g^{-1}(v_i)) + \int_{Q_s}^{v_i} Y(g^{-1}(\omega)) \cdot d\omega$$

In the second round, the agent's expected utility $\mathcal{U}(u_i, Q)$ is given by Equation 6.17, where $g(u)$ is the equilibrium strategy (given by Equation 6.14), which is used to maximize the utility. Note that in this case, we must substitute $\Phi(x)$ and $Y(x)$ with $\tilde{\Phi}(x)$ and $\tilde{Y}(x)$, since in the second round the utilities are drawn from $H(u)$.

The price Q at start of the second round depends on the bids placed in the first round:

If $B^{(m)} > v_i$ then $Q = B^{(m)}$.

If $B^{(m-1)} > v_i \geq B^{(m)}$, then the agent submitted the m^{th} price, so $Q = v_i$.

If $v_i \geq B^{(m-1)}$, then $Q = B^{(m-1)}$.

Note that (see derivation of Equation 6.13)

$$\text{Prob}[B^{(m-1)} \leq v] = Y(g^{-1}(v))$$

and

$$\text{Prob}[B^{(m)} \leq v] = \Phi(g^{-1}(v))$$

and that

$$\text{Prob}[Q = v_i] = \text{Prob}[B^{(m-1)} > v_i \geq B^{(m)}] = \text{Prob}[B^{(m)} \leq v_i] - \text{Prob}[B^{(m-1)} \leq v_i] =$$

$$\Phi(g^{-1}(v)) - Y(g^{-1}(v))$$

It is also $\mathcal{U}(u_i, Q) = 0, \forall Q > u_i$, since the agent does not wish to participate in that case. In addition $Q \geq Q_s$, since the agents cannot subtract bids, and thus, can only increase the bid price of the auction.

As a result, we can now compute $U_i^{(2)}$ as follows:

$$\begin{aligned} U_i^{(2)} &= \int_{Q_s}^{u_i} \mathcal{U}(u_i, \omega) \cdot \text{Prob}[Q = \omega] \cdot d\omega = \\ &\int_{Q_s}^{v_i^-} \mathcal{U}(u_i, \omega) \cdot \text{Prob}[Q = \omega] \cdot d\omega + \mathcal{U}(u_i, v_i) \cdot \text{Prob}[Q = v_i] \\ &\quad + \int_{v_i^+}^{u_i} \mathcal{U}(u_i, \omega) \cdot \text{Prob}[Q = \omega] \cdot d\omega = \\ &\int_{Q_s}^{v_i} \mathcal{U}(u_i, \omega) \cdot \text{Prob}[B^{(m-1)} = \omega] \cdot d\omega + \mathcal{U}(u_i, v_i) \cdot \text{Prob}[Q = v_i] \\ &\quad + \int_{v_i}^{u_i} \mathcal{U}(u_i, \omega) \cdot \text{Prob}[B^{(m)} = \omega] \cdot d\omega \Leftrightarrow \\ U_i^{(2)} &= \int_{Q_s}^{v_i} \mathcal{U}(u_i, \omega) \cdot \frac{d}{d\omega} Y(g^{-1}(\omega)) \cdot d\omega + \mathcal{U}(u_i, v_i) \cdot \{\Phi(g^{-1}(v_i)) - Y(g^{-1}(v_i))\} \\ &\quad + \int_{v_i}^{u_i} \mathcal{U}(u_i, \omega) \cdot \frac{d}{d\omega} \Phi(g^{-1}(\omega)) \cdot d\omega \end{aligned}$$

The expected utility for both rounds is thus,

$$\begin{aligned} EU_i(v_i) &= p \cdot U_i^{(1)} + (1 - p) \cdot U_i^{(2)} \Leftrightarrow \\ EU_i(v_i) &= p \cdot \{(u_i - v_i) \cdot \Phi(g^{-1}(v_i)) + \int_{Q_s}^{v_i} Y(g^{-1}(\omega)) \cdot d\omega\} \\ &+ (1 - p) \cdot \left\{ \int_{Q_s}^{v_i} \mathcal{U}(u_i, \omega) \cdot \frac{d}{d\omega} Y(g^{-1}(\omega)) \cdot d\omega + \mathcal{U}(u_i, v_i) \cdot \{\Phi(g^{-1}(v_i)) - Y(g^{-1}(v_i))\} \right. \\ &\quad \left. + \int_{v_i}^{u_i} \mathcal{U}(u_i, \omega) \cdot \frac{d}{d\omega} \Phi(g^{-1}(\omega)) \cdot d\omega \right\} \end{aligned} \quad (6.19)$$

The bid v_i that maximizes $EU_i(v_i)$ can be found by setting

$$\frac{dU_i(v_i)}{dv_i} = 0 \Leftrightarrow$$

$$\begin{aligned}
& p \cdot \left\{ (u_i - v_i) \cdot \frac{d}{dv_i} \Phi(g^{-1}(v_i)) - \Phi(g^{-1}(v_i)) + Y(g^{-1}(v_i)) \right\} \\
& + (1-p) \cdot \frac{\partial \mathcal{U}(u_i, v_i)}{\partial v_i} \cdot (\Phi(g^{-1}(v_i)) - Y(g^{-1}(v_i))) = 0 \Leftrightarrow \\
& (u_i - v_i) \cdot \frac{\Phi'(g^{-1}(v_i))}{g'(g^{-1}(v_i))} = \left(1 - \frac{1-p}{p} \cdot \frac{\partial \mathcal{U}(u_i, v_i)}{\partial v_i} \right) \cdot (\Phi(g^{-1}(v_i)) - Y(g^{-1}(v_i)))
\end{aligned}$$

Since $v_i = g(u_i)$ we get

$$(u_i - g(u_i)) \cdot \frac{\Phi'(u_i)}{g'(u_i)} = \left(1 - \frac{1-p}{p} \cdot \frac{\partial \mathcal{U}(u_i, v_i)}{\partial v_i} \Big|_{v_i=g(u_i)} \right) \cdot (\Phi(u_i) - Y(u_i))$$

and if we set

$$\begin{aligned}
\Psi(u_i, Q) &= 1 - \frac{1-p}{p} \cdot \frac{\partial \mathcal{U}(u, Q)}{\partial Q} = \\
& 1 + \frac{1-p}{p} \cdot \frac{\tilde{\Phi}(Q) - \tilde{Y}(Q)}{\tilde{\Phi}(u_i) - \tilde{Y}(u_i)} \cdot e^{\int_Q^{u_i} \frac{-\tilde{Y}'(\omega)}{\tilde{\Phi}(\omega) - \tilde{Y}(\omega)} \cdot d\omega} \cdot \left(\tilde{\Phi}(u_i) + \int_Q^{u_i} \frac{\tilde{Y}(\omega) \cdot \tilde{\Phi}'(\omega)}{\tilde{\Phi}(\omega) - \tilde{Y}(\omega)} \cdot d\omega \right)
\end{aligned}$$

the differential equation becomes

$$(u_i - g(u_i)) \cdot \frac{\Phi'(u_i)}{g'(u_i)} = (\Phi(u_i) - Y(u_i)) \cdot \Psi(u_i, g(u_i))$$

If $u_i = Q_s$, then the agent must bid $v_i = Q_s$, hence, the boundary condition.

Theorem 8. *If the starting price is $Q_s \geq 0$, a second round of bidding exists with probability $(1-p)$ ($p \neq 0, 1$), the utility of the other agents in the second round is drawn from distribution $H(u)$, and each agent i knows (more accurately for itself) that its own utility \tilde{u}_i is drawn from distribution $G(u)$, then the equilibrium strategy is the solution of the differential equation*

$$(u_i - g(u_i) + \frac{1-p}{p} \cdot U_L(g(u_i))) \cdot \frac{\Phi'(u_i)}{g'(u_i)} = (\Phi(u_i) - Y(u_i)) \cdot \Psi(g(u_i)) \quad (6.20)$$

where

$$\begin{aligned}
\Psi(Q) &= 1 + \frac{1-p}{p} \cdot \\
& \left\{ \int_Q^{+\infty} G'(z) \cdot \frac{\tilde{\Phi}(Q) - \tilde{Y}(Q)}{\tilde{\Phi}(z) - \tilde{Y}(z)} \cdot e^{\int_Q^z \frac{-\tilde{Y}'(\omega)}{\tilde{\Phi}(\omega) - \tilde{Y}(\omega)} \cdot d\omega} \cdot \left(\tilde{\Phi}(z) + \int_Q^z \frac{\tilde{Y}(\omega) \cdot \tilde{\Phi}'(\omega)}{\tilde{\Phi}(\omega) - \tilde{Y}(\omega)} \cdot d\omega \right) \cdot dz + \right.
\end{aligned}$$

$$\begin{aligned}
& \sum_{k=0}^{m-1} \frac{N \cdot (m-k)}{m \cdot (N-k)} \cdot C(N-1, k) \cdot \\
& \left\{ (N-1-k) \cdot H'(Q) \cdot (H(Q))^{N-2-k} \cdot (1-H(Q))^k \cdot \int_0^Q G(\omega) \cdot d\omega \right. \\
& \quad - k \cdot H'(Q) \cdot (H(Q))^{N-1-k} \cdot (1-H(Q))^{(k-1)} \cdot \int_0^Q G(\omega) \cdot d\omega \\
& \quad \left. + (H(Q))^{N-1-k} \cdot (1-H(Q))^k \cdot G(Q) \right\} \tag{6.21}
\end{aligned}$$

and

$$U_L(Q) = - \sum_{k=0}^{m-1} \left\{ \frac{N \cdot (m-k)}{m \cdot (N-k)} \cdot C(N-1, k) \cdot (H(Q))^{N-1-k} \cdot (1-H(Q))^k \cdot \int_0^Q G(\omega) \cdot d\omega \right\} \tag{6.22}$$

The boundary condition is $g(Q_s) = Q_s$.

Proof. Initially, we must compute the expected gain of utility (actually it's negative, so it's a loss) $U_L(Q)$, if the agent is a winner in the first round, and in the second his utility $\tilde{u}_i < Q$. The agent is forced (by the rules) to keep a bid of at least Q in the auction; so it puts a bid $\tilde{v}_i = Q$.

Let us assume that exactly k of the other agents wish to buy the goods.

If $k \geq m$, there is no problem, and the utility difference is 0.

However if $k < m$, there are $(m-k)$ units that must be sold randomly to some of the previous winners. We can compute that the probability of the event B , that any of the $(m-k)$ winners in the first round (including our agent) will be forced to buy a ticket equal to⁴

$$\text{Prob}(B) = \frac{N \cdot (m-k)}{m \cdot (N-k)}$$

In order to compute this probability, we must assume that $j : 0 \leq j \leq k$ out of those k agents were winners also in the first round. The probability of this event A_j , that exactly

⁴This is bigger than $\frac{m-k}{m}$, because some of the k agents might have been winners on the first round as well.

j out of the k winners in the second round were also winners in the first round, can be computed as follows. Since our own agent was also a winner in the first round, there are $(N - 1)$ other agents from which $(m - 1)$ won the first round. Since the probability distributions for the utilities of the other $(N - 1)$ agents are identical, this means that each possible combination of winners has the same probability, which is $\frac{1}{C(N-1, m-1)}$. However, for the case that we are interested in, we know that exactly j out of the group of k winners in the second round, and exactly $(m - 1 - j)$ out of the remaining $(N - 1 - k)$ agents were winners in the first round. Therefore, the number of combinations that fit our desired event is $C(k, j) \cdot C(N - 1 - k, m - 1 - j)$. Thus, the probability that exactly j out of the k winners in the second round were also winners in the first round is

$$Prob(A_j) = \frac{C(k, j) \cdot C(N - 1 - k, m - 1 - j)}{C(N - 1, m - 1)}$$

Note that (see Lemma 3)

$$\sum_{j=0}^k \frac{C(k, j) \cdot C(N - 1 - k, m - 1 - j)}{C(N - 1, m - 1)} = 1$$

There are $(m - j)$ initial winners left that do not wish to purchase an item in the second round; the remaining $(m - k)$ items must be divided among the $(m - j)$ agents randomly, with the same probability to each agent. Therefore, the probability that an agent will have to purchase an undesirable item in the case that j agents were winners in both rounds is

$$Prob(B|A_j) = \frac{m - k}{m - j}$$

Hence, the probability of the event B that any of the $(M - k)$ winners in the first round will have to buy a ticket is

$$Prob(B) = \sum_{j=0}^k Prob(B|A_j) \cdot Prob(A_j) \Leftrightarrow$$

$$Prob(B) = \sum_{j=0}^k \frac{m - k}{m - j} \cdot \frac{C(k, j) \cdot C(N - 1 - k, m - 1 - j)}{C(N - 1, m - 1)} =$$

$$\begin{aligned} & \sum_{j=0}^k \left(1 - \frac{k-j}{m-j}\right) \cdot \frac{C(k, j) \cdot C(N-1-k, m-1-j)}{C(N-1, m-1)} = \\ & \quad \sum_{j=0}^k \frac{C(k, j) \cdot C(N-1-k, m-1-j)}{C(N-1, m-1)} \\ & \quad - \sum_{j=0}^{k-1} \frac{k-j}{m-j} \cdot \frac{C(k, j) \cdot C(N-1-k, m-1-j)}{C(N-1, m-1)} \end{aligned}$$

From Lemma 3, we know that

$$\sum_{j=0}^k \frac{C(k, j) \cdot C(N-1-k, m-1-j)}{C(N-1, m-1)} = 1$$

and also it is

$$\begin{aligned} & \sum_{j=0}^{k-1} \frac{k-j}{m-j} \cdot \frac{C(k, j) \cdot C(N-1-k, m-1-j)}{C(N-1, m-1)} = \\ & \sum_{j=0}^{k-1} \frac{k}{N-k} \cdot \frac{C(k-1, j) \cdot C(N-k, m-j)}{C(N-1, m)} \cdot \frac{N-m}{m} = \\ & \frac{k}{N-k} \cdot \frac{N-m}{m} \cdot \sum_{j=0}^{k-1} \frac{C(k-1, j) \cdot C(N-k, m-j)}{C(N-1, m)} = \\ & \quad \frac{k}{N-k} \cdot \frac{N-m}{m} \end{aligned}$$

(because of Lemma 3)

Thus,

$$\begin{aligned} Prob(B) &= 1 - \frac{k}{N-k} \cdot \frac{N-m}{m} \Leftrightarrow \\ Prob(B) &= \frac{N \cdot (m-k)}{m \cdot (N-k)} \end{aligned}$$

The probability that k of the other agents will have utilities $\tilde{u}_j \geq Q$ is (see Lemma 2)

$$C(N-1, k) \cdot (H(Q))^{N-1-k} \cdot (1-H(Q))^k$$

and, if selected, the utility difference is

$$\tilde{u}_i - Q$$

Thus, the expected utility difference $U_L(\tilde{u}_i, Q)$ is

$$U_L(\tilde{u}_i, Q) = \sum_{k=0}^{m-1} \left\{ \frac{N \cdot (m-k)}{m \cdot (N-k)} \cdot C(N-1, k) \cdot (H(Q))^{N-1-k} \cdot (1-H(Q))^k \cdot (\tilde{u}_i - Q) \right\}$$

if $\tilde{u}_i < Q$.

If $\tilde{u}_i \geq Q$, then we can't have this case, since the agent would wish to participate. Since the value of the utility \tilde{u}_i is not known, but the distribution $G(u)$, from which it is drawn is, the total utility difference (loss actually), because of forced bidding of the agents that won the first round, is

$$U_L(Q) = \int_0^Q U_L(\tilde{u}_i, Q) \cdot Prob[\tilde{u}_i = \omega] \cdot d\omega$$

Note also, that

$$\int_0^Q (\tilde{u}_i - Q) \cdot Prob[\tilde{u}_i = \omega] \cdot d\omega = \int_0^Q G(\omega) \cdot d\omega$$

It is therefore,

$$U_L(Q) = - \sum_{k=0}^{m-1} \left\{ \frac{N \cdot (m-k)}{m \cdot (N-k)} \cdot C(N-1, k) \cdot (H(Q))^{N-1-k} \cdot (1-H(Q))^k \cdot \int_0^Q G(\omega) \cdot d\omega \right\}$$

The expected utility for the agent at the second round (without the inclusion of the utility difference stated in Equation 6.22) is

$$\tilde{U}(Q) = \int_Q^{+\infty} \mathcal{U}(\omega, Q) \cdot Prob[\tilde{u}_i = \omega] \cdot d\omega$$

This is because, if $\tilde{u}_i < Q$, then the utility gained is 0.

If $\tilde{u}_i \geq Q$, then the utility gained is $\mathcal{U}(\omega, Q)$, and it is computed by Equations 6.14 and 6.17, in which $\Phi(x)$ and $Y(x)$ are replaced by

$$\tilde{\Phi}(x) = \sum_{i=0}^{m-1} C(N-1, i) \cdot (H(x))^{N-1-i} \cdot (1-H(x))^i$$

and

$$\tilde{Y}(x) = \sum_{i=0}^{m-2} C(N-1, i) \cdot (H(x))^{N-1-i} \cdot (1-H(x))^i$$

Both utilities $\tilde{U}(Q)$ and $U_L(Q)$ depend on the price Q at the start of the second round, which in turn depends on the bids placed in the first round.

If $B^{(m)} > v_i$ then $Q = B^{(m)}$; in this case, the agent does not win in the first round, so there is no utility loss in the second round.

If $B^{(m-1)} > v_i \geq B^{(m)}$, then the agent submitted the m^{th} price, so $Q = v_i$, and if $v_i \geq B^{(m-1)}$, then $Q = B^{(m-1)}$; in both of these cases, there is an expected loss in the second round.

Note that (see proof of Theorem 6)

$$\text{Prob}[B^{(m-1)} \leq v] = Y(g^{-1}(v))$$

and

$$\text{Prob}[B^{(m)} \leq v] = \Phi(g^{-1}(v))$$

and that

$$\begin{aligned} \text{Prob}[Q = v_i] &= \text{Prob}[B^{(m-1)} > v_i \geq B^{(m)}] = \\ &= \text{Prob}[B^{(m)} \leq v_i] - \text{Prob}[B^{(m-1)} \leq v_i] = \Phi(g^{-1}(v)) - Y(g^{-1}(v)) \end{aligned}$$

As a result, we can now compute the expected utility, if a second round does exist, $U_i^{(2)}$ as follows:

$$\begin{aligned} U_i^{(2)} &= \int_0^{u_i} \mathcal{U}(\omega) \cdot \text{Prob}[Q = \omega] \cdot d\omega = \\ &= \int_0^{v_i^-} (\tilde{U}(\omega) + U_L(\omega)) \cdot \text{Prob}[Q = \omega] \cdot d\omega + (\tilde{U}(v_i) + U_L(v_i)) \cdot \text{Prob}[Q = v_i] \\ &\quad + \int_{v_i^+}^{u_i} \tilde{U}(\omega) \cdot \text{Prob}[Q = \omega] \cdot d\omega \Leftrightarrow \\ U_i^{(2)} &= \int_0^{v_i} (\tilde{U}(\omega) + U_L(\omega)) \cdot \frac{d}{d\omega} Y(g^{-1}(\omega)) \cdot d\omega \\ &\quad + (\tilde{U}(v_i) + U_L(v_i)) \cdot \{\Phi(g^{-1}(v_i)) - Y(g^{-1}(v_i))\} + \int_{v_i}^{u_i} \tilde{U}(\omega) \cdot \frac{d}{d\omega} \Phi(g^{-1}(\omega)) \cdot d\omega \end{aligned}$$

The expected utility if the auction closes at the first round $U_i^{(1)}$ is given by Equation 6.16, when Q is replaced by Q_s is inserted, since the price is Q_s at the beginning of the first

round.

The expected utility for both rounds is

$$\begin{aligned}
EU_i(v_i) &= p \cdot U_i^{(1)} + (1-p) \cdot U_i^{(2)} \Leftrightarrow \\
EU_i(v_i) &= p \cdot \left\{ (u_i - v_i) \cdot \Phi(g^{-1}(v_i)) + \int_{Q_s}^{v_i} Y(g^{-1}(\omega)) \cdot d\omega \right\} \quad (6.23) \\
&\quad + (1-p) \cdot \left\{ \int_{Q_s}^{v_i} (\tilde{U}(\omega) + U_L(\omega)) \cdot \frac{d}{d\omega} Y(g^{-1}(\omega)) \cdot d\omega \right. \\
&\quad \left. + (\tilde{U}(v_i) + U_L(v_i)) \cdot \left\{ \Phi(g^{-1}(v_i)) - Y(g^{-1}(v_i)) \right\} + \int_{v_i}^{u_i} \tilde{U}(\omega) \cdot \frac{d}{d\omega} \Phi(g^{-1}(\omega)) \cdot d\omega \right\}
\end{aligned}$$

We find the equilibrium by setting

$$\frac{dU_i(v_i)}{dv_i} = 0$$

and then substituting

$$v_i = g(u_i)$$

In the end, we get the differential equation

$$\begin{aligned}
&(u_i - g(u_i) + \frac{1-p}{p} \cdot U_L(g(u_i))) \cdot \frac{\Phi'(u_i)}{g'(u_i)} = \\
&(1 - \frac{1-p}{p} \cdot (\tilde{U}'(g(u_i)) + U_L'(g(u_i)))) \cdot (\Phi(u_i) - Y(u_i))
\end{aligned}$$

Let us set

$$\Psi(Q) = 1 - \frac{1-p}{p} \cdot (\tilde{U}'(Q) + U_L'(Q))$$

therefore,

$$\begin{aligned}
&\Psi(Q) = 1 + \frac{1-p}{p} \cdot \\
&\left\{ \int_Q^{+\infty} G'(z) \cdot \frac{\tilde{\Phi}(Q) - \tilde{Y}(Q)}{\tilde{\Phi}(z) - \tilde{Y}(z)} \cdot e^{\int_Q^z \frac{-\tilde{Y}'(\omega)}{\tilde{\Phi}(\omega) - \tilde{Y}(\omega)} \cdot d\omega} \cdot (\tilde{\Phi}(z) + \int_Q^z \frac{\tilde{Y}'(\omega) \cdot \tilde{\Phi}'(\omega)}{\tilde{\Phi}(\omega) - \tilde{Y}(\omega)} \cdot d\omega) \cdot dz + \right. \\
&\quad \left. \sum_{k=0}^{m-1} \frac{N \cdot (m-k)}{m \cdot (N-k)} \cdot C(N-1, k) \cdot \right.
\end{aligned}$$

$$\begin{aligned} & \{(N-1-k) \cdot H'(Q) \cdot (H(Q))^{N-2-k} \cdot (1-H(Q))^k \cdot \int_0^Q G(\omega) \cdot d\omega \\ & -k \cdot H'(Q) \cdot (H(Q))^{N-1-k} \cdot (1-H(Q))^{(k-1)} \cdot \int_0^Q G(\omega) \cdot d\omega \\ & + (H(Q))^{N-1-k} \cdot (1-H(Q))^k \cdot G(Q)\} \end{aligned}$$

where (as stated earlier)

$$\tilde{\Phi}(x) = \sum_{i=0}^{m-1} C(N-1, i) \cdot (H(x))^{N-1-i} \cdot (1-H(x))^i$$

and

$$\tilde{Y}(x) = \sum_{i=0}^{m-2} C(N-1, i) \cdot (H(x))^{N-1-i} \cdot (1-H(x))^i$$

Then the differential equation becomes

$$(u_i - g(u_i) + \frac{1-p}{p} \cdot U_L(g(u_i))) \cdot \frac{\Phi'(u_i)}{g'(u_i)} = (\Phi(u_i) - Y(u_i)) \cdot \Psi(g(u_i))$$

The boundary condition is $g(Q_s) = Q_s$, as the agent only has the choice of bidding $v_i = Q_s$, when $u_i = Q_s$.

6.2 Bayes-Nash Equilibria for Multiple Round Auctions

In this section, we present the extensions of the equilibria that were computed for the two-round auction to the R -round auction, where $R \geq 2$.

By following the methodology that we applied in Section 6.1, we can derive the systems of differential equations that are necessary, in order to compute an equilibrium for an auction that closes at multiple possible times, and not just two, as was the case in the theorems that were presented. We break the auction into more rounds, compute the utility for the last two rounds (e.g. solving Equation 6.20 and then substituting in Equation 6.23), and then recursively compute the expected utilities for the previous rounds, using the last utility function as the next round's utility function, until the first round is reached.

6.2.1 Formal Definition of the Problem

We assume that N risk-neutral agents wish to buy 1 unit each of a certain good. An independent seller sells m units of the desired good in an m^{th} price auction, i.e., the good is sold to the agents which submitted the m highest bids at a price equal to the lowest winning bid. The agents have valuations (utilities) u_i^r at each round r which are i.i.d. with probability distribution $F_r(u)$. Each agent knows its own valuation and the distribution $F_r(u)$. The probability that a round r will be the final one (when the auction closes), is known and is equal to p_r ; therefore, there can be one or more later rounds with probability $(1 - p_r)$. At each round, the agents can submit new bids as long as they are greater or equal to the bid price Q_r , from the end of the previous round. Once more, we can make the same assumptions about the information that each agent i has about its utility $u_i^{r'}$ in a later round r' , when it is currently evaluating its bid for round r .

- $u_i^{r'}, \forall r' > r$ can be assumed to be similar to the utility u_i^r of the current round r . Theorem 9 provides the system of differential equations that define the most general equilibrium for this case.

We implement this equilibrium solution and compute it for both the uniform and experimentally derived distribution of utilities for functions $F_r(u)$ in Section 6.2.3.

- The agent might not know anything about $u_i^{r'}, \forall r' > r$, other than that they are drawn from $F_r(u)$ (this is the same information that it has about the other agent's valuations). Theorem 10 provides the system of differential equations that define the most general equilibrium for this case, when $G_r(u)$ is substituted by $F_r(u)$.
- The agent might know that its future utilities $u_i^{r'}, \forall r' > r$ are drawn from a more “tight” and accurate distribution $G_r(u)$ instead of $F_r(u)$. Theorem 10 provides the system of differential equations that define the most general equilibrium for

this case.

We implement this equilibrium solution and compute it for both the uniform and experimentally derived distribution of utilities for functions $F_r(u)$ and $G_r(u)$ in Section 6.2.3.

Once more the agents may not subtract bids, and therefore, must bid for any room they were winning in the previous round at a price $v_i^r \geq Q_r$, where Q_r is the current price at this round. Thus, any undesired rooms are awarded randomly to winners from the previous rounds.

We will use the following functions in the theorems that follow:

$$\Phi_r(x) = \sum_{i=0}^{m-1} C(N-1, i) \cdot (F_r(x))^{N-1-i} \cdot (1 - F_r(x))^i$$

and

$$Y_r(x) = \sum_{i=0}^{m-2} C(N-1, i) \cdot (F_r(x))^{N-1-i} \cdot (1 - F_r(x))^i$$

6.2.2 Computing the Equilibria

The following Theorems (9 and 10) are based on Theorems 7 and 8. The derivation process is almost the same, with the only difference being that the expected utility of the next round is not substituted into the equations. In the end, when we solve the R -round equilibrium, we get a system of ordinary differential equations, the solution of each depending on the solution of the next round. Note also, that the differential equations of both theorems are dependent on Q_r , only because of the boundary condition, which is $g(Q_r, Q_r) = Q_r$ for any given Q_r .

Theorem 9. *If the starting price of the current round r is $Q_r \geq 0$, the next round of bidding ($r+1$) exists with probability $(1-p_r)$ ($p_r \neq 0, 1$) and the utility of the agents in round r is drawn from the distribution $F_r(u)$ (and each agent i in fact has utility u_i^r of a*

similar value to the utility u_i of the first round), then the equilibrium strategy $g_r(u_i, Q_r)$ is the solution of the differential equation

$$(u_i - g_r(u_i, Q_r)) \cdot \frac{\Phi'_r(u_i)}{\frac{\partial g_r(u_i, Q_r)}{\partial u_i}} = (\Phi_r(u_i) - Y_r(u_i)) \cdot \Psi_r(u_i, g_r(u_i, Q_r)) \quad (6.24)$$

where

$$\Psi_r(u, x) = 1 - \frac{1 - p_r}{p_r} \cdot \frac{\partial \mathcal{U}_{r+1}(u, x)}{\partial x}$$

and $\mathcal{U}_{r+1}(u_i, Q_{r+1})$ is the expected utility at round $(r + 1)$, when the agent's utility is u_i and the starting price is Q_{r+1} .

The boundary condition is $g(u_i, Q_r) = Q_r$, when $u_i = Q_r$.

The expected utility at round r given that $g_r(u_i, Q_r)$ has been computed is

$$\begin{aligned} \mathcal{U}_r(u_i, Q_r) = & p_r \cdot \left\{ (u_i - g_r(u_i, Q_r)) \cdot \Phi_r(u_i) + \int_{Q_r}^{u_i} Y_r(\omega) \cdot g'_r(\omega, Q_r) \cdot d\omega \right\} \\ & + (1 - p_r) \cdot \left\{ \int_{Q_r}^{u_i} \mathcal{U}_{r+1}(u_i, g_r(\omega, Q_r)) \cdot Y'_r(\omega) \cdot d\omega + \mathcal{U}_{r+1}(u_i, g_r(u_i, Q_r)) \cdot \{ \Phi_r(u_i) - Y_r(u_i) \} \right. \\ & \left. + \int_{u_i}^{g_r^{-1}(u_i, Q_r)} \mathcal{U}_{r+1}(u_i, g_r(\omega, Q_r)) \cdot \Phi'_r(\omega) \cdot d\omega \right\} \quad (6.25) \end{aligned}$$

Proof. The proof is exactly the same as the proof of Theorem 7, with the exception that the utility of the next round is not substituted into the equations. The expected utility \mathcal{U}_r is then computed by substituting the solution $g_r(u_i, Q_r)$ into Equation 6.19. It should be noted that because of the boundary condition, g_r as it appears in Theorem 7 is also dependent on the value of Q_r , so it is really $g_r(u_i, Q_r)$.

It is now possible to compute the strategy at any round $r, r \in \{1, \dots, R\}$. At the beginning, the agent computes the strategy $g_R(u, Q)$ and utility $\mathcal{U}_R(u_i, Q)$ of round R (last round), using Equations 6.24 and 6.25 as it is suggested in Theorem 9. We then apply the same equations one more time to compute the strategy and utility at round $(R - 1)$ using $\mathcal{U}_R(u_i, Q)$, etc., until round 1 is reached.

Theorem 10. *If the starting price of the current round r is $Q_r \geq 0$, the next round of bidding ($r+1$) exists with probability $(1-p_r)$ ($p_r \neq 0, 1$), and the utility of the agents in round r is drawn from the distribution $F_r(u)$ (and each agent i knows more accurately that, in fact, its utility u_i^r is drawn from distribution $G_r(u)$), then the equilibrium strategy $g_r(u_i^r, Q_r)$ is the solution of the differential equation*

$$(u_i^r - g_r(u_i^r, Q_r) + \frac{1-p_r}{p_r} \cdot U_L^r(g_r(u_i^r, Q_r))) \cdot \frac{\Phi_r'(u_i)}{\frac{\partial g_r(u_i^r, Q_r)}{\partial u_i^r}} = (\Phi_r(u_i^r) - Y_r(u_i^r)) \cdot \Psi_r(g_r(u_i^r, Q_r)) \quad (6.26)$$

where

$$\Psi_r(x) = 1 - \frac{1-p_r}{p_r} \cdot \frac{d}{dx} (\tilde{U}^{r+1}(x) + U_L^{r+1}(x))$$

and

$$U_L^{r+1}(Q) = - \sum_{k=0}^{m-1} \left\{ \frac{N \cdot (m-k)}{m \cdot (N-k)} \cdot C(N-1, k) \cdot (F_{r+1}(Q_{r+1}))^{N-1-k} \cdot (1 - F_{r+1}(Q_{r+1}))^k \cdot \int_0^Q G_{R=1}(\omega) \cdot d\omega \right\}$$

and the expected utility at round r when u_i^r is drawn from $G_r(u)$ is

$$\tilde{U}^r(Q_r) = \int_{Q_r}^{+\infty} \mathcal{U}_r(\omega, Q_r) \cdot \frac{d}{d\omega} G_r(\omega) \cdot d\omega \quad (6.27)$$

The boundary condition is $g(u_i, Q_r) = Q_r$, when $u_i = Q_r$.

$\mathcal{U}_r(u_i^r, Q_r)$ is the expected utility at round r , when the agent's utility is u_i^r and the starting price is Q_r and given that $g_r(u_i^r, Q_r)$ has been computed it is

$$\begin{aligned} \mathcal{U}_r(u_i^r, Q_r) = & p_r \cdot \left\{ (u_i^r - g_r(u_i^r, Q_r)) \cdot \Phi_r(u_i^r) + \int_{Q_r}^{u_i^r} Y_r(\omega) \cdot g_r'(\omega, Q_r) \cdot d\omega \right\} \\ & + (1-p_r) \cdot \left\{ \int_{Q_r}^{u_i^r} \tilde{U}^{r+1}(g_r(\omega, Q_r)) \cdot Y_r'(\omega) \cdot d\omega + \tilde{U}^{r+1}(g_r(u_i^r, Q_r)) \cdot \left\{ \Phi_r(u_i^r) - Y_r(u_i^r) \right\} \right. \\ & \left. + \int_{u_i^r}^{g_r^{-1}(u_i^r, Q_r)} \tilde{U}^{r+1}(g_r(\omega, Q_r)) \cdot \Phi_r'(\omega) \cdot d\omega \right\} \quad (6.28) \end{aligned}$$

Proof. The proof is exactly the same as the proof of Theorem 8, with the exception that the utility of the next round is not substituted into the equations. The expected utility \mathcal{U}_r is then computed by substituting the solution $g_r(u_i, Q_r)$ into Equation 6.23.⁵ It should be noted that because of the boundary condition, g_r as it appears in Theorem 8, is also dependent on the value of Q_r , so it is really $g_r(u_i, Q_r)$.

Another issue that is important to note is that Ψ_r , \tilde{U}^r and U_L^r may depend also on u_i^{r-1} ; this only happens if $G_r(u)$ depends on u_i^{r-1} , as will be the case in the second part of the next section.

Once more, it is possible to compute the strategy at any round r , $r \in \{1, \dots, R\}$ by computing the solution for round R and then continuing to apply Theorem 10 for the previous rounds, until round 1 is reached.

6.2.3 Numerical Evaluation Of Equilibrium Strategies

To apply the equilibria that we described in Section 6.2.2, we implemented the algorithms described by Theorems 9 and 10 using `Matlab`. Figure 6.2 provides the algorithm for this process for Theorem 9. We also applied Theorem 10, by considering that $G_r(u)$ is equal to

$$G_r(u) = F_r(u|u \geq u_{r-1})$$

meaning that we disregard any value of u that is smaller than the utility in the previous round; this is reasonable for TAC, since utilities rarely decrease between rounds. We only need to add one step to the algorithm of Figure 6.2: at the end of the “for loop”, we add one more step (see Equation 6.27) in order to compute the utility of this round when the utility u_i^r is only known to be drawn from distribution $G_r(u)$.

⁵Note that $\tilde{U}^r(Q_r)$ is given by Equation 6.27 and that this integration can be taken out of the other integrals in Equation 6.23, and performed after all the other integrations are completed.

```

 $EU_{r+1}(u, Q) := 0, \forall u, Q$ 

for r := 8 to 1 do
   $\Psi_r := 1 - \frac{1-p_r}{p_r} \cdot \frac{\partial}{\partial Q} EU_{r+1}(u, Q), \forall u$ 
   $\forall Q$  compute  $g_r(u, Q)$  by solving Differential Eq.
   $\forall u, Q$  compute  $EU_r(u, Q)$ 
end for

```

Figure 6.2: Algorithm for solving the system of Theorem 9.

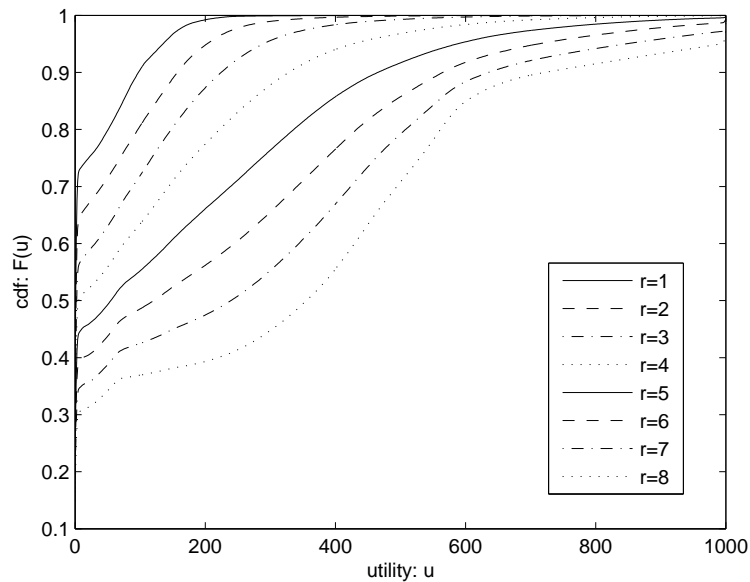


Figure 6.3: The experimental distribution $F_r(u)$ for round r (r takes values between 1 and 8 as shown by the lines from top to bottom respectively) that we used for our experiments. It has been created by sampling the utilities from a large number of real TAC games.

We computed both of these equilibria for three different cases:

1. Uniform distribution, $N = 2$ agents buying $m = 1$ item; this case was used in order to compare the results of the algorithm with the known solution (e.g. for 2 rounds as shown in Figure 6.1).
2. Uniform distribution, $N = 64$ agents buying $m = 16$ items; this uses the uniform distribution, and the proper number of agents and items for a real TAC game.
3. Experiment derived distribution, $N = 64$ agents buying $m = 16$ items. Since our stated goal was to apply these equilibria to a TAC agent in order to generate one more “partial” strategy, we collected the utilities of the hotel rooms from a large number of actual games and used these to create the distributions $F_r(u)$ and $G_r(u)$. The cdf of the distribution $F_r(u)$ that we use in the formulas for the equilibria is presented in Figure 6.3. We make sure that the distributions and their derivatives are continuous, so that we can apply a Runge-Kutta method. To do this, we group all the samples in buckets of size 2, and then to make it continuous, we expand each bucket to a normal distribution with $\sigma = 2$ and center equal to the center of the bucket.⁶

In the next pages, we graph the bid functions $g_r(u, Q)$ and expected utilities $EU_r(u, Q)$ for the 3 cases mentioned.

Uniform Distribution, $N = 2$ agents, $m = 1$ item

Figures 6.4 through 6.14 show the equilibria computed, based on both Theorems (9 and 10), as well as the expected utilities for some of the rounds. The rounds not presented here have equilibria which have similarities to the previous and next rounds, and

⁶This is the most common way for turning sampled discrete distributions into continuous ones used in statistics.

thus, we omit them. One may notice that in earlier rounds the agents bids less than in the later rounds. Furthermore, by comparing Figures 6.1 and 6.5 we verify that they are identical for the case that the current price $Q = 0$.

Note also that the equilibrium for round 8 is the same for both versions (based on both theorems) since the utility of the next round is 0 (a next round does not exist).

In addition, we observe that the equilibrium solution given by Theorem 10 is lower for $\forall u, Q$ at every round, which is something that we expected. The reason for this is that the expected utility in each round is now higher; this can be easily verified by comparing the corresponding figures (e.g. Figures 6.11 and 6.12).

The last observation that can be made is that the solution at each round is very similar to the solution that we would get if, instead of several later rounds (with probability $(1 - p)$ that the auction will close later), we only have one possible later round (again with probability $(1 - p)$ that the auction does not close at this round, but at the next one). This indicates that, from the point-of-view of the agent, the number of possible rounds does not matter, as long as the probability of having more rounds is the same.

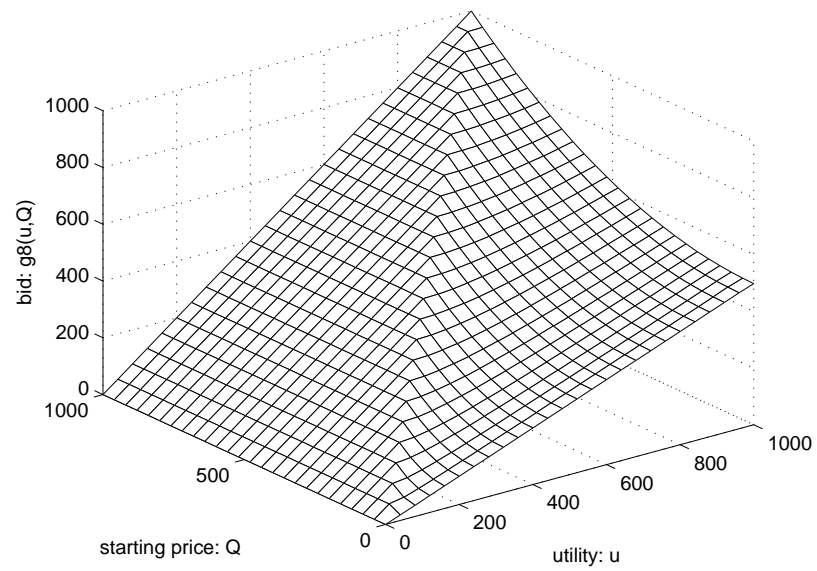


Figure 6.4: Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 8 in the case of a uniform distribution with $N = 2$ and $m = 1$. The equilibrium from Theorem 10 is the same.

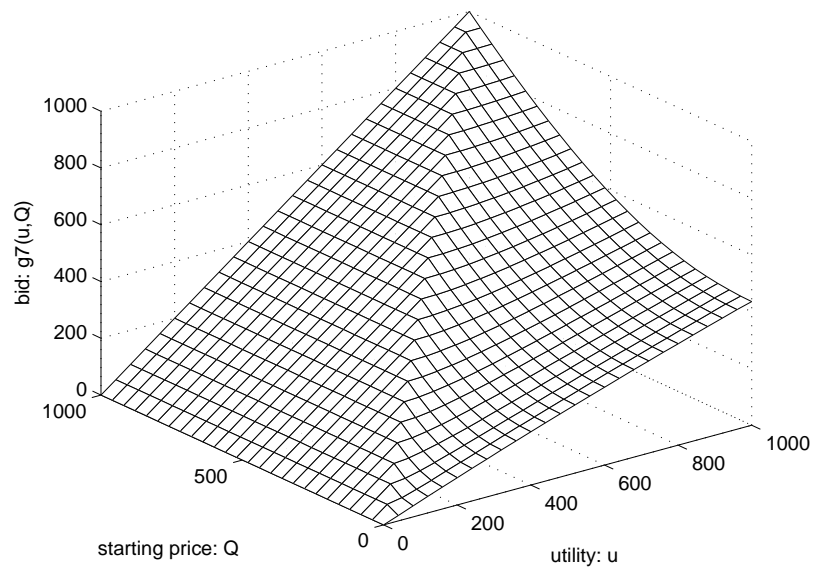


Figure 6.5: Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 7, in the case of a uniform distribution with $N = 2$ and $m = 1$.

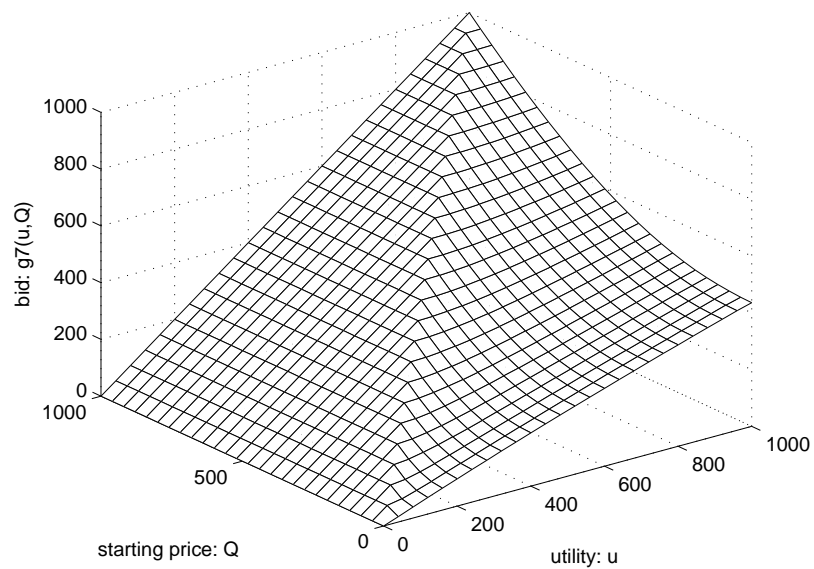


Figure 6.6: Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 7, in the case of a uniform distribution with $N = 2$ and $m = 1$.

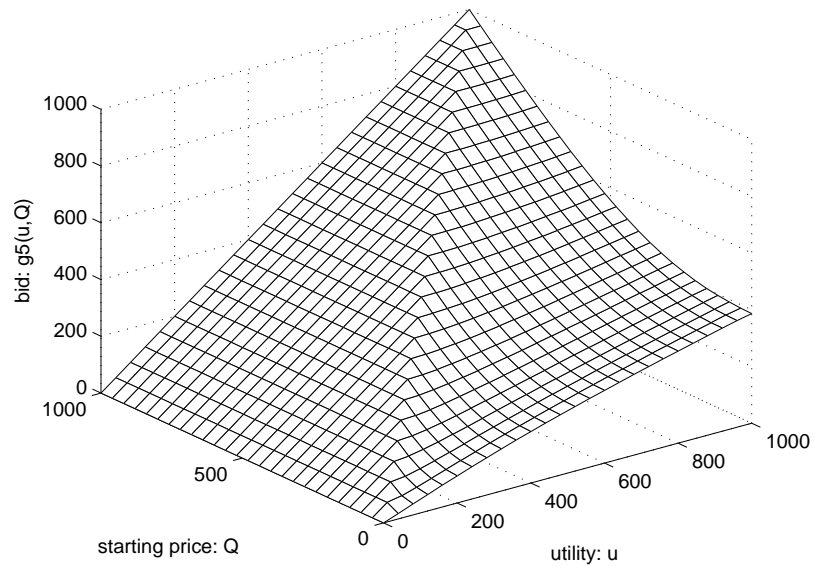


Figure 6.7: Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 5, in the case of a uniform distribution with $N = 2$ and $m = 1$.

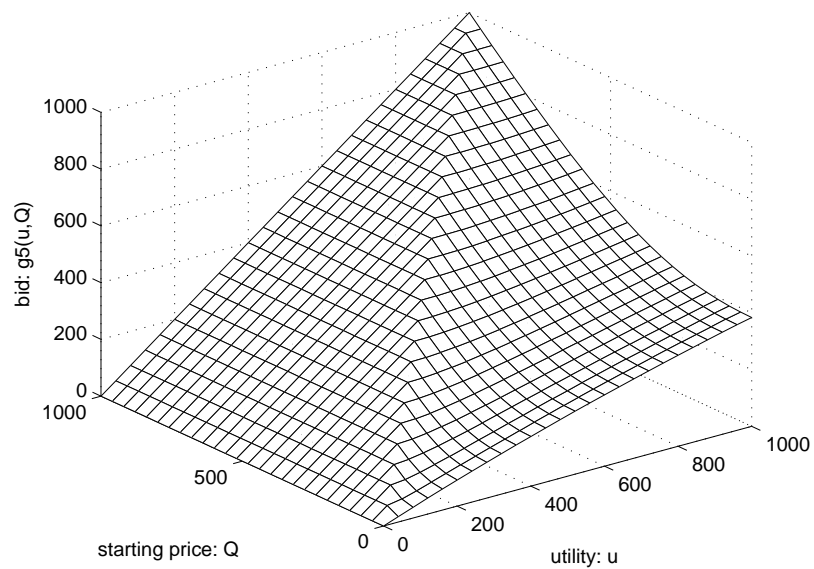


Figure 6.8: Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 5, in the case of a uniform distribution with $N = 2$ and $m = 1$.

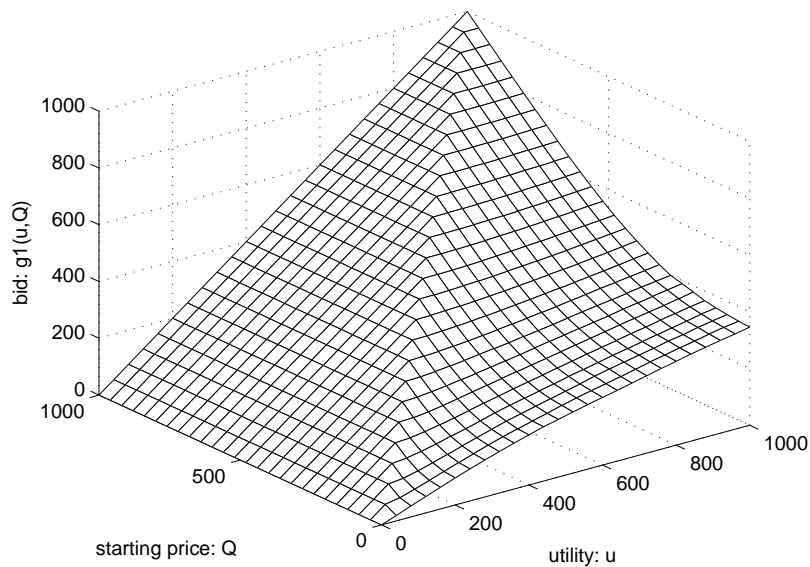


Figure 6.9: Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 1, in the case of a uniform distribution with $N = 2$ and $m = 1$.

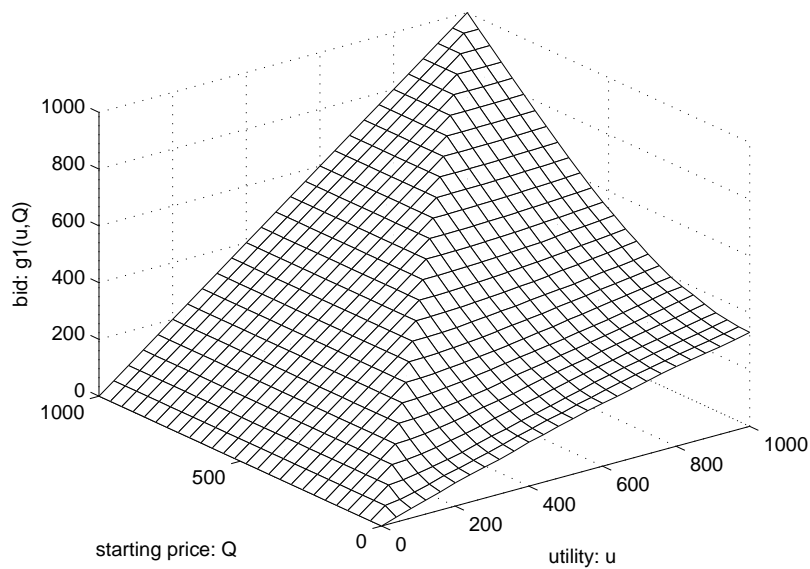


Figure 6.10: Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 1, in the case of a uniform distribution with $N = 2$ and $m = 1$.

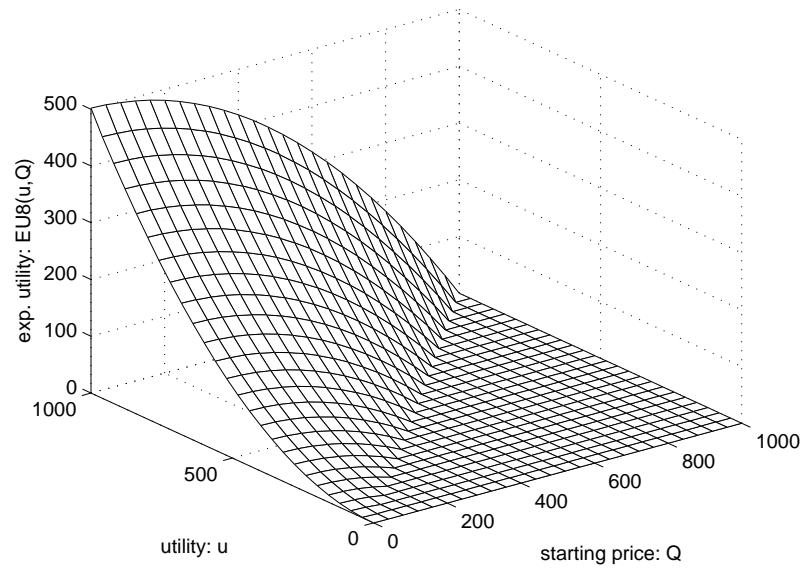


Figure 6.11: Expected utility $EU_r(u, Q)$ from Theorem 9 for round 8, in the case of a uniform distribution with $N = 2$ and $m = 1$.

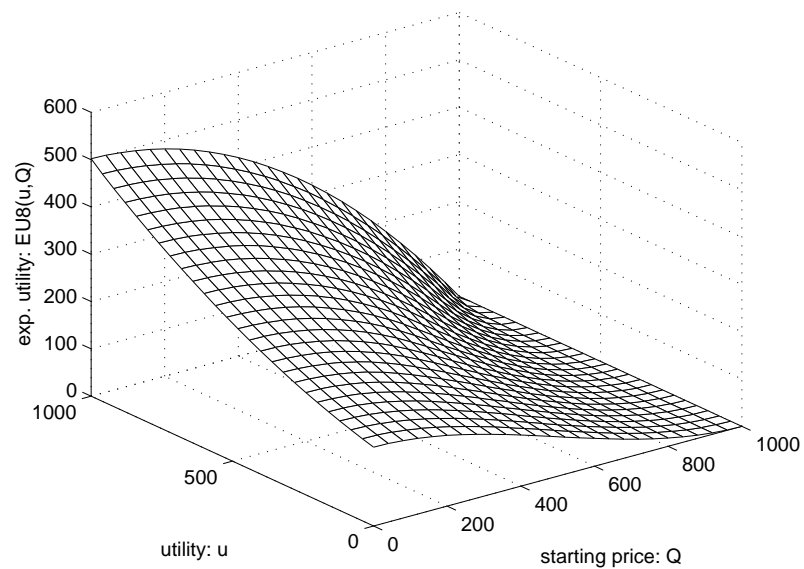


Figure 6.12: Expected utility $EU_r(u, Q)$ from Theorem 10 for round 8, in the case of a uniform distribution with $N = 2$ and $m = 1$.

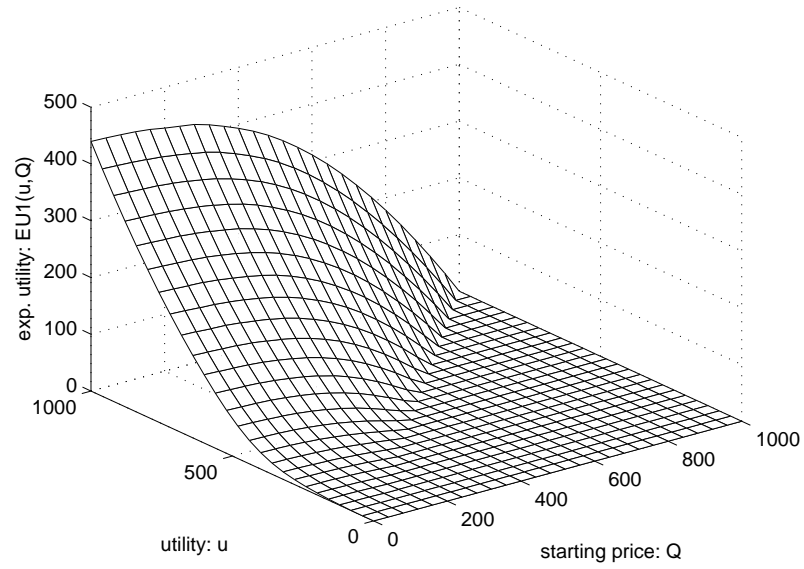


Figure 6.13: Expected utility $EU_r(u, Q)$ from Theorem 9 for round 1, in the case of a uniform distribution with $N = 2$ and $m = 1$.

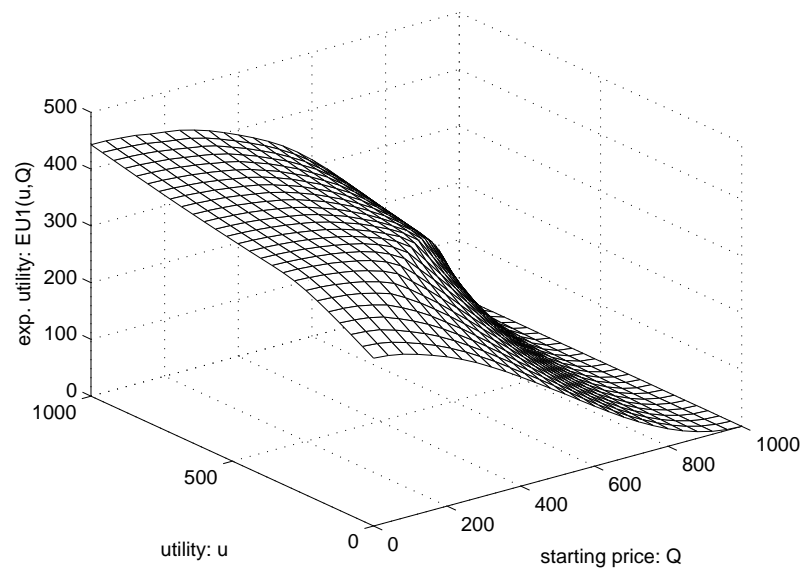


Figure 6.14: Expected utility $EU_r(u, Q)$ from Theorem 10 for round 1, in the case of a uniform distribution with $N = 2$ and $m = 1$.

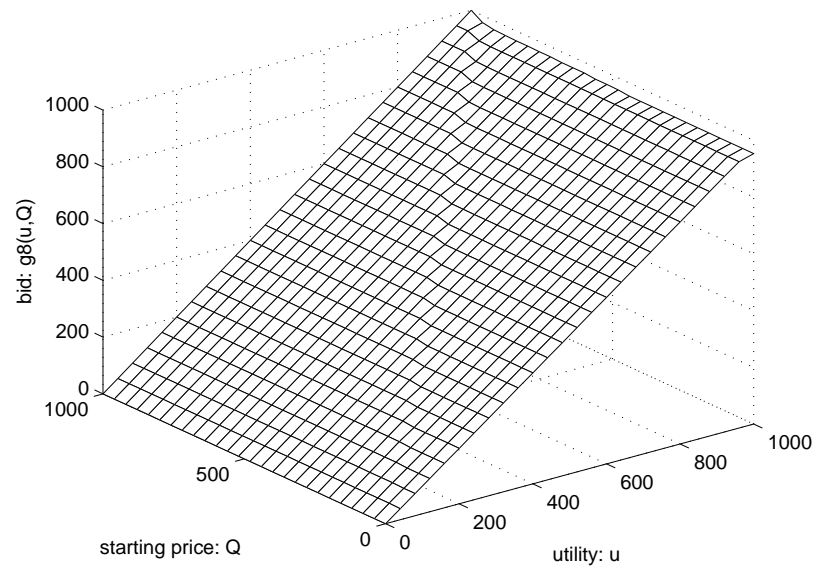


Figure 6.15: Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 8, in the case of a uniform distribution with $N = 64$ and $m = 16$. The equilibrium from Theorem 10 is the same.

Uniform Distribution, $N = 64$ agents, $m = 16$ item

Figures 6.15 through 6.23 show the equilibria computed, based on both Theorems (9 and 10), as well as the expected utilities for some of the rounds. The rounds not presented here have equilibria which have similarities to the previous and next rounds, and thus, we omit them. Once more the optimal strategy suggests that the agent bid less in the earlier rounds, but the difference is not significant.

An interesting observation that can be made here is that, because of the fact that many more agents participate, compared to the number of items offered, the equilibrium solution in all rounds is to bid very close to the actual utility. In this case, the agent makes a profit by the fact that its bid is likely not to be the m^{th} price, and thus, pays less most of the time than its actual bid, if it wins.

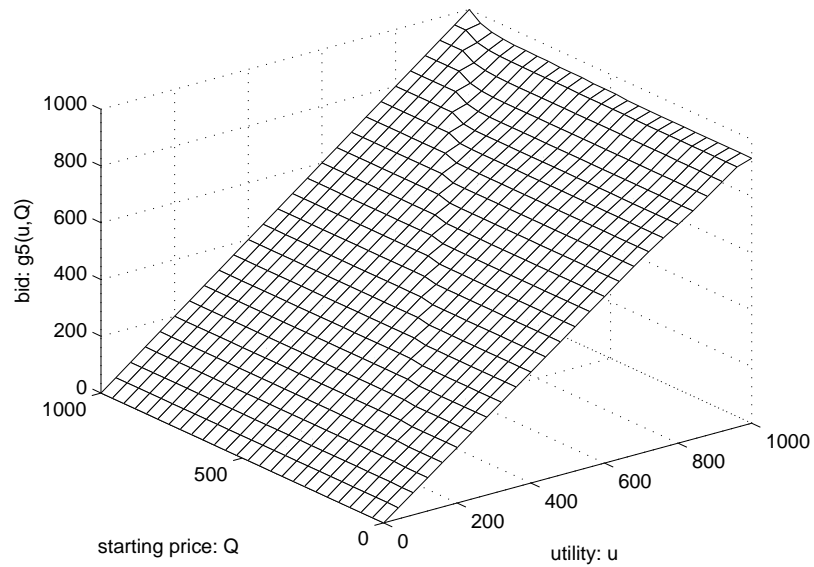


Figure 6.16: Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 5, in the case of a uniform distribution with $N = 64$ and $m = 16$.

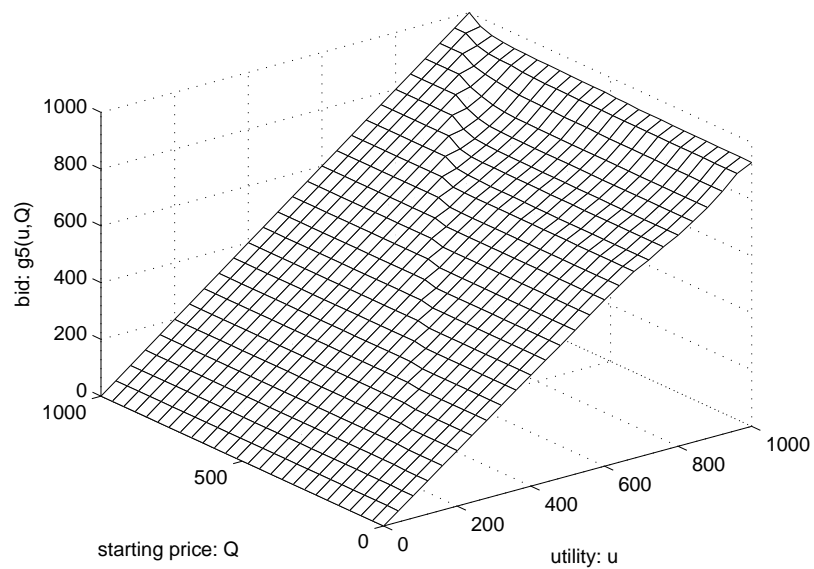


Figure 6.17: Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 5, in the case of a uniform distribution with $N = 64$ and $m = 16$.

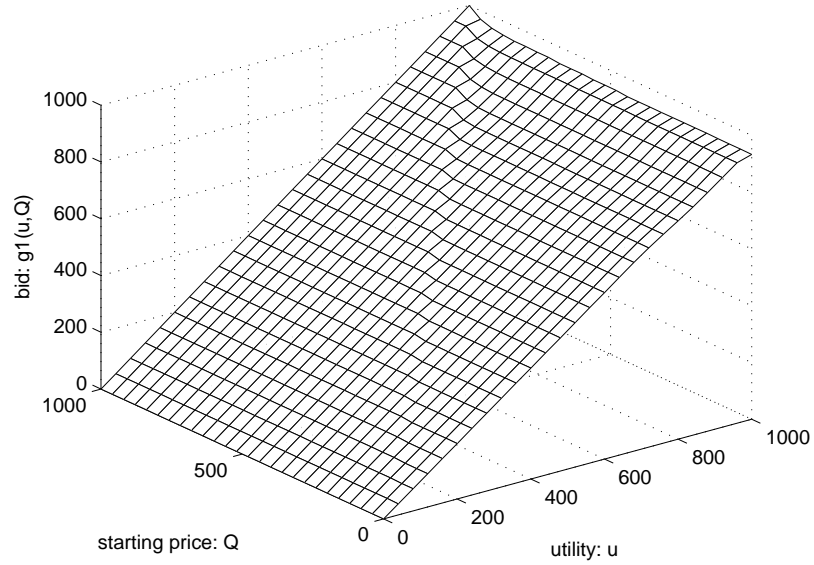


Figure 6.18: Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 1, in the case of a uniform distribution with $N = 64$ and $m = 16$.

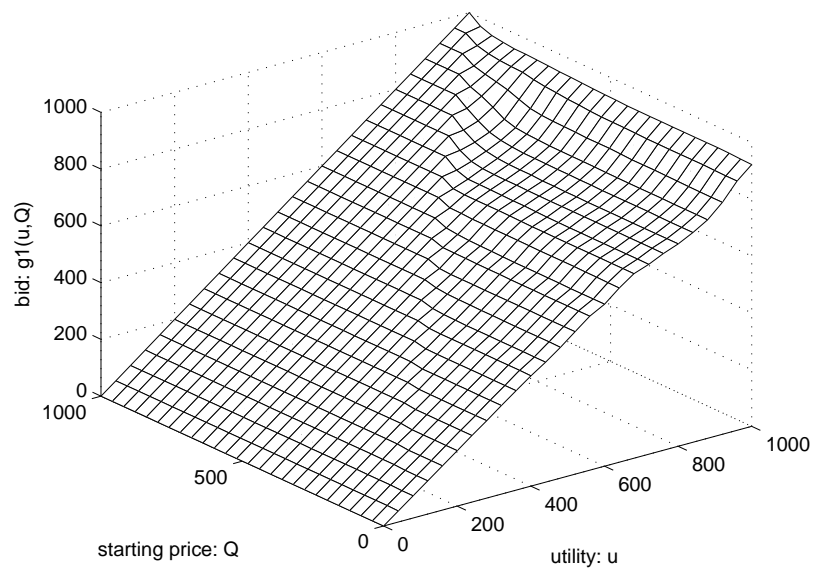


Figure 6.19: Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 1, in the case of a uniform distribution with $N = 64$ and $m = 16$.

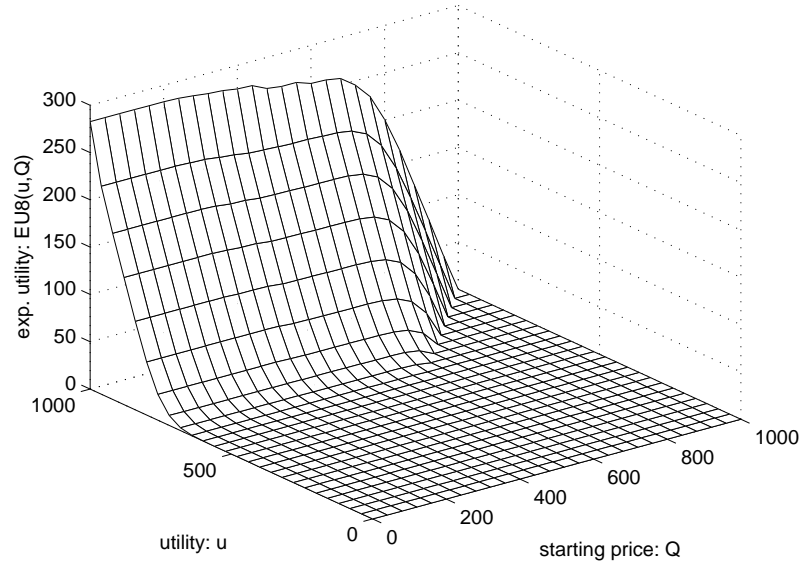


Figure 6.20: Expected utility $EU_r(u, Q)$ from Theorem 9 for round 8, in the case of a uniform distribution with $N = 64$ and $m = 16$.

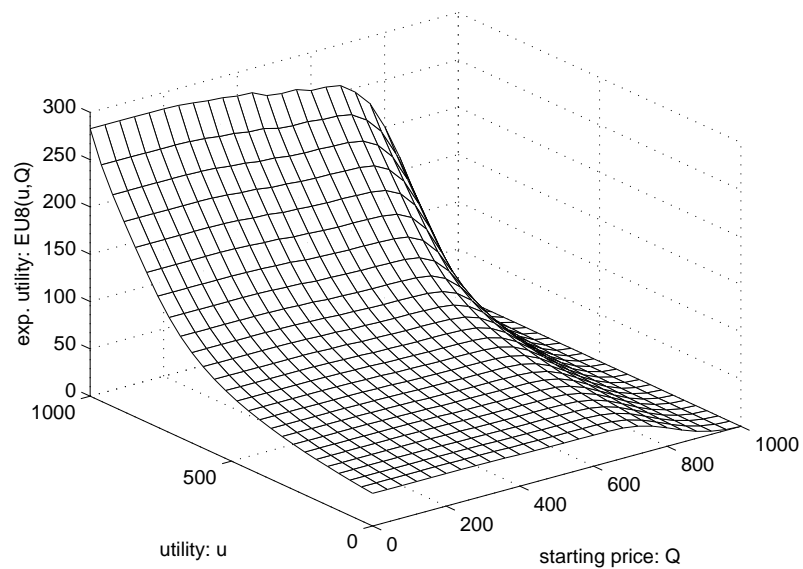


Figure 6.21: Expected utility $EU_r(u, Q)$ from Theorem 10 for round 8, in the case of a uniform distribution with $N = 64$ and $m = 16$.

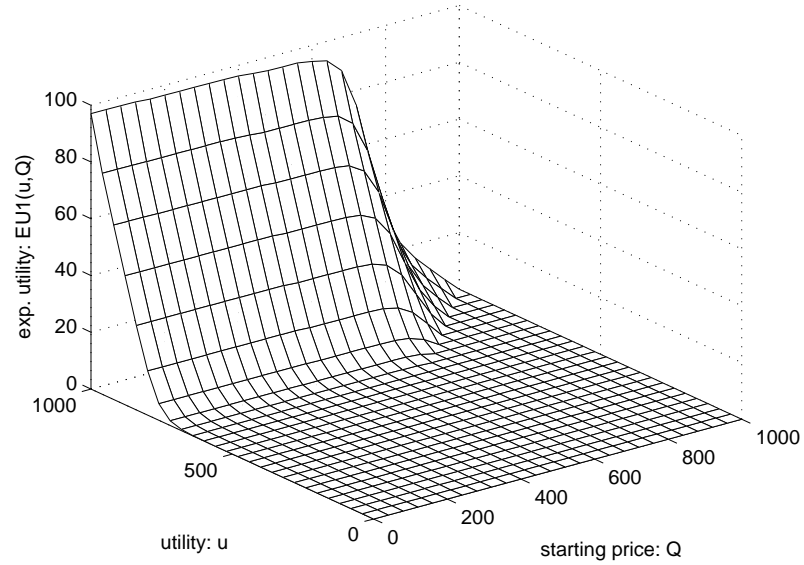


Figure 6.22: Expected utility $EU_r(u, Q)$ from Theorem 9 for round 1, in the case of a uniform distribution with $N = 64$ and $m = 16$.

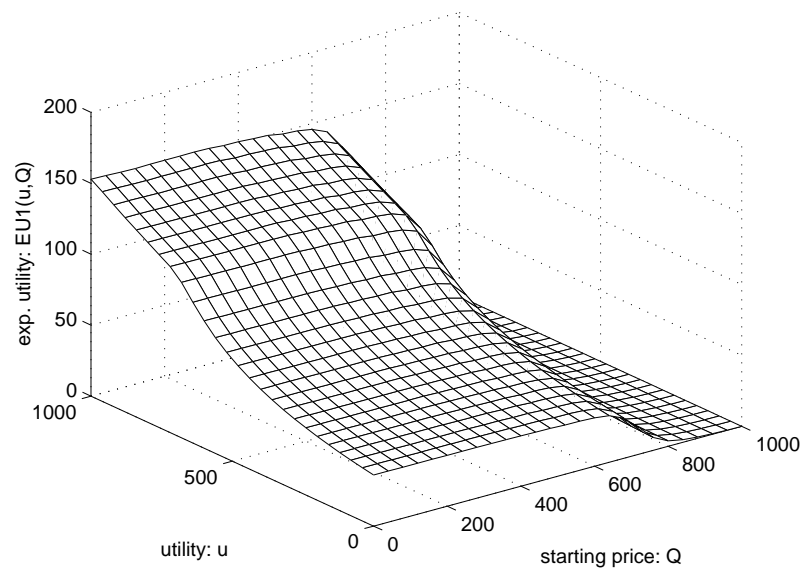


Figure 6.23: Expected utility $EU_r(u, Q)$ from Theorem 10 for round 1, in the case of a uniform distribution with $N = 64$ and $m = 16$.

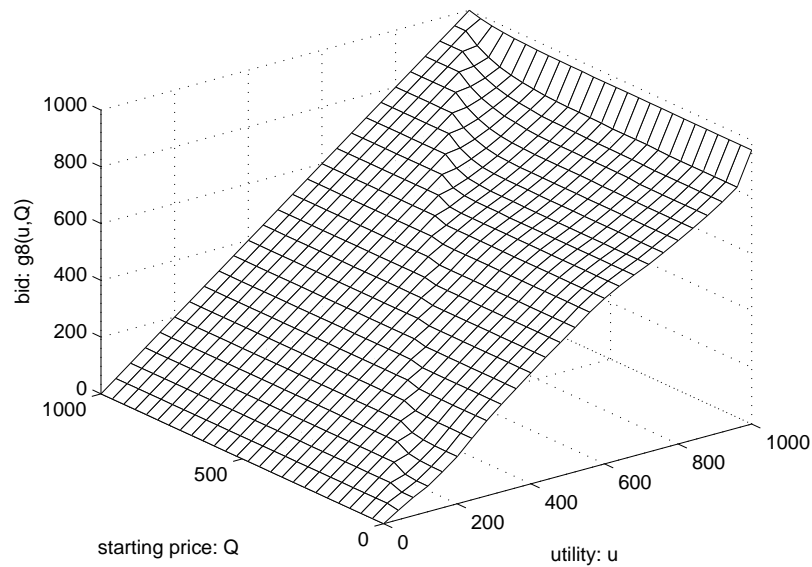


Figure 6.24: Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 8, in the case of a uniform distribution with $N = 64$ and $m = 16$. The equilibrium from Theorem 10 is the same.

Experiment Derived Distribution, $N = 64$ agents, $m = 16$ item

Figures 6.24 through 6.34 show the equilibria computed, based on both Theorems (9 and 10), as well as the expected utilities for some of the rounds.

In this section, we give the bid function $g_r(u, Q)$ for most rounds; the fact that the utility distribution $F_r(u)$ is different in every round causes the bid function to vary significantly between different rounds. In the later rounds, it is advisable to bid close to the utility that the agent has, but in the early rounds, it is preferable to bid closer to the current price Q .

We used the results of the equilibrium from Theorem 9 in our agent to generate a “partial” strategy that bids in the way that is directed by function $g_r(u, Q)$.

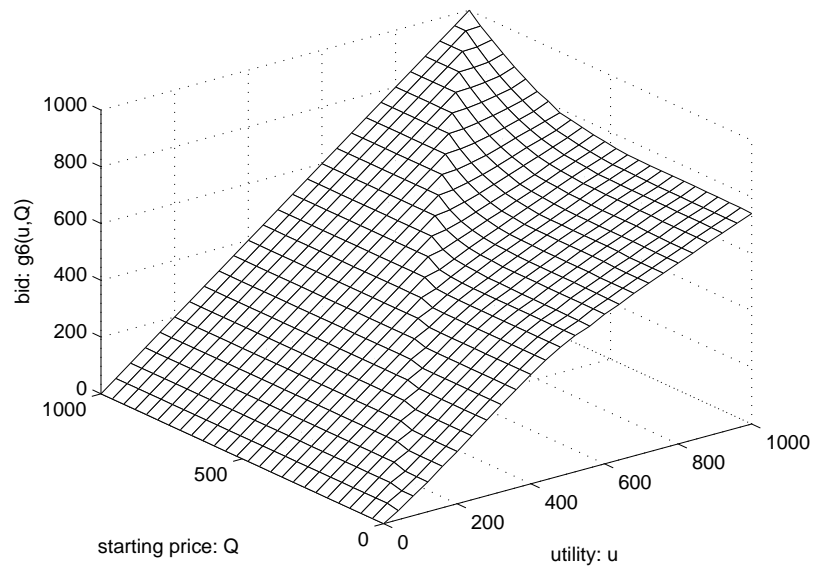


Figure 6.25: Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 6, in the case of the experiment derived distribution with $N = 64$ and $m = 16$.

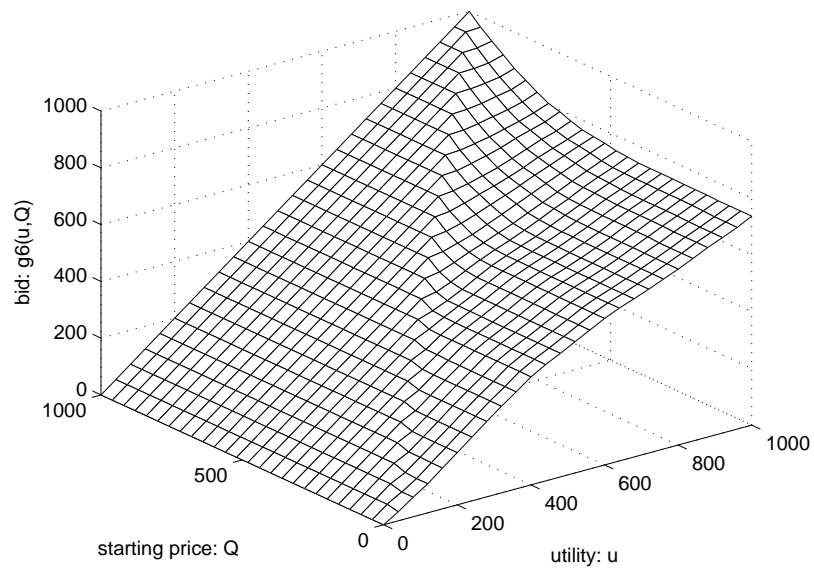


Figure 6.26: Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 6, in the case of the experiment derived distribution with $N = 64$ and $m = 16$.

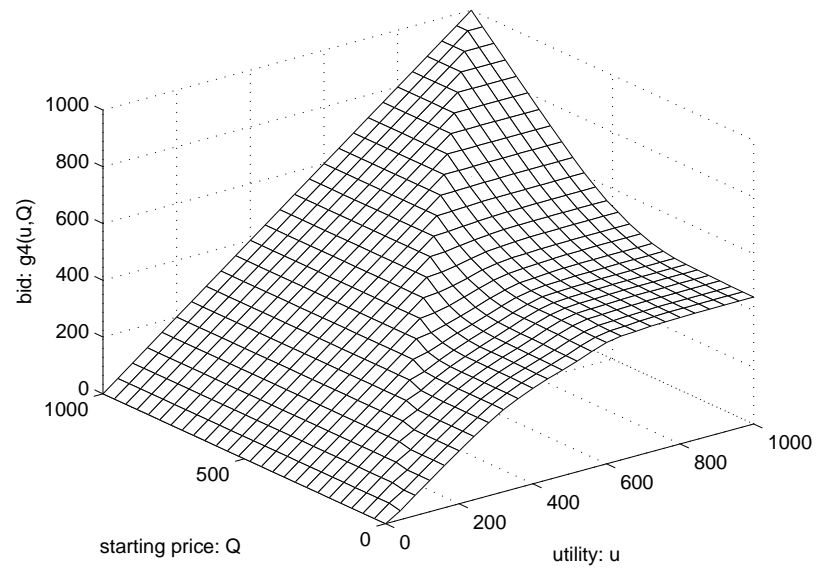


Figure 6.27: Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 4, in the case of the experiment derived distribution with $N = 64$ and $m = 16$.

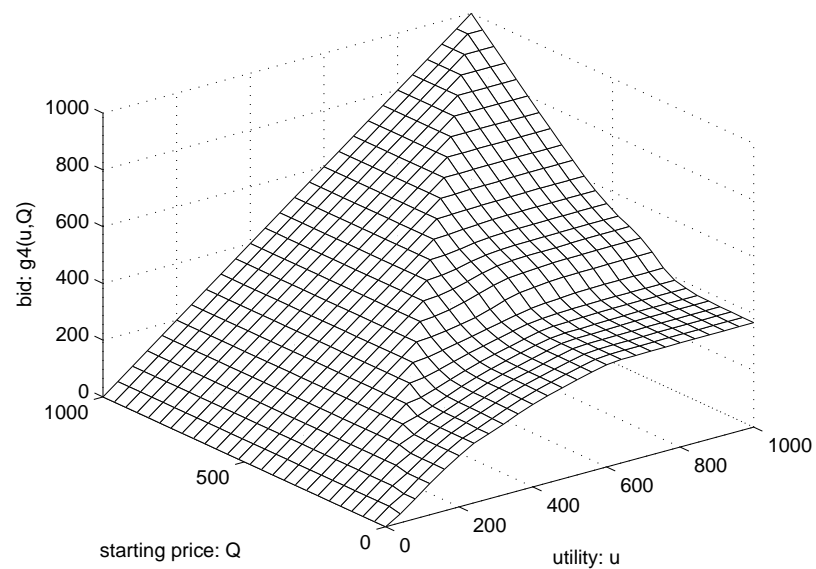


Figure 6.28: Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 4, in the case of the experiment derived distribution with $N = 64$ and $m = 16$.

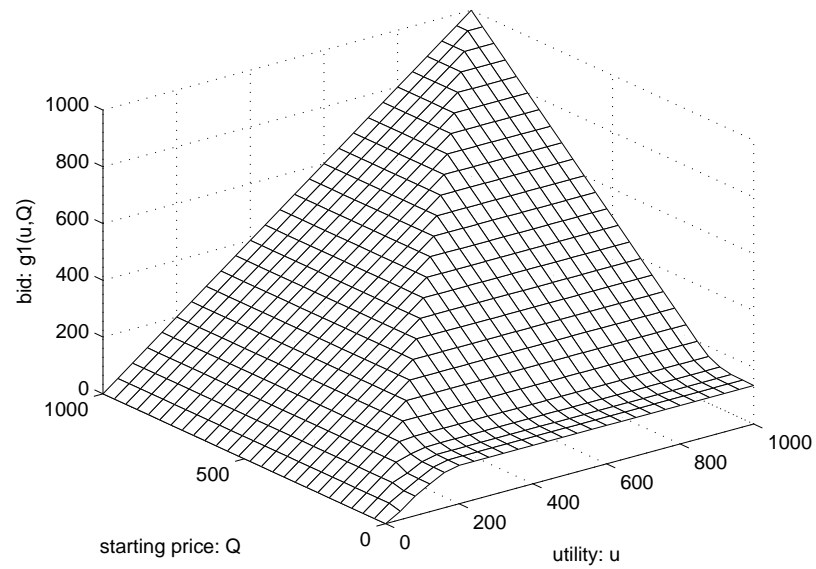


Figure 6.29: Equilibrium strategy $g_r(u, Q)$ of Theorem 9 for round 1, in the case of the experiment derived distribution with $N = 64$ and $m = 16$.

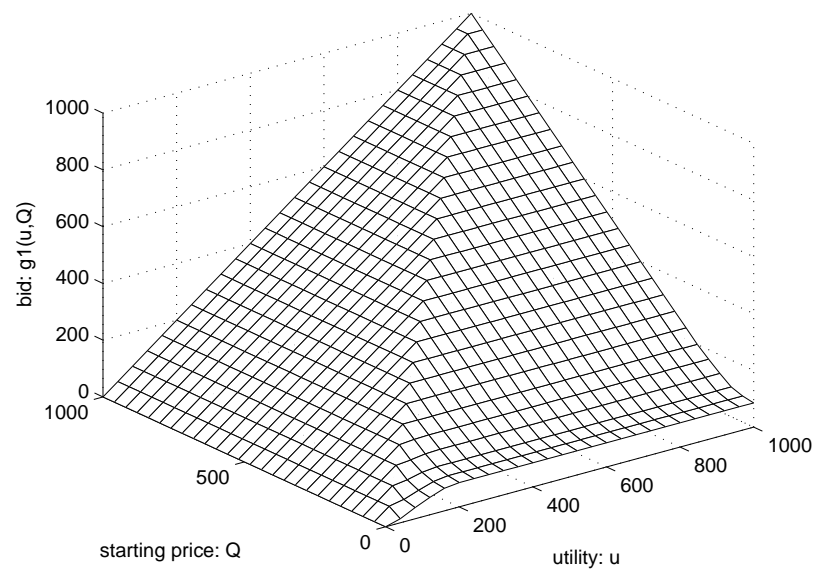


Figure 6.30: Equilibrium strategy $g_r(u, Q)$ of Theorem 10 for round 1, in the case of the experiment derived distribution with $N = 64$ and $m = 16$.

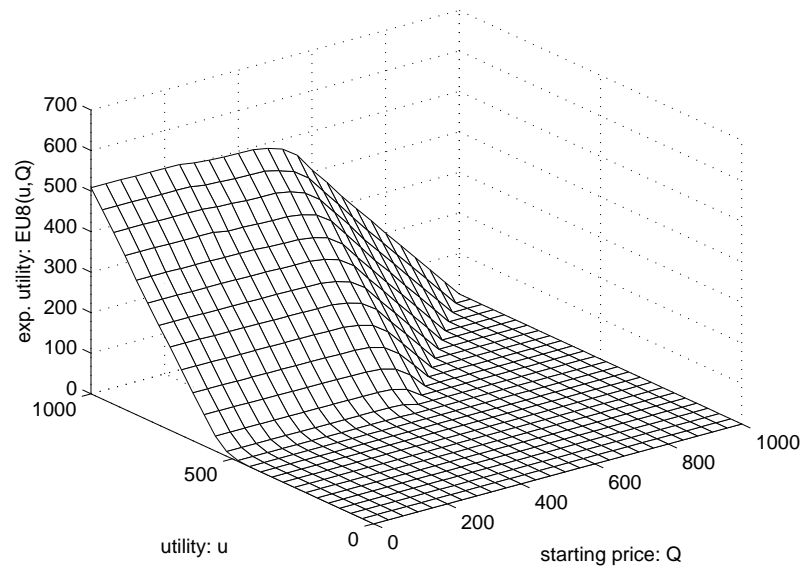


Figure 6.31: Expected utility $EU_r(u, Q)$ from Theorem 9 for round 8, in the case of the experiment derived distribution with $N = 64$ and $m = 16$.

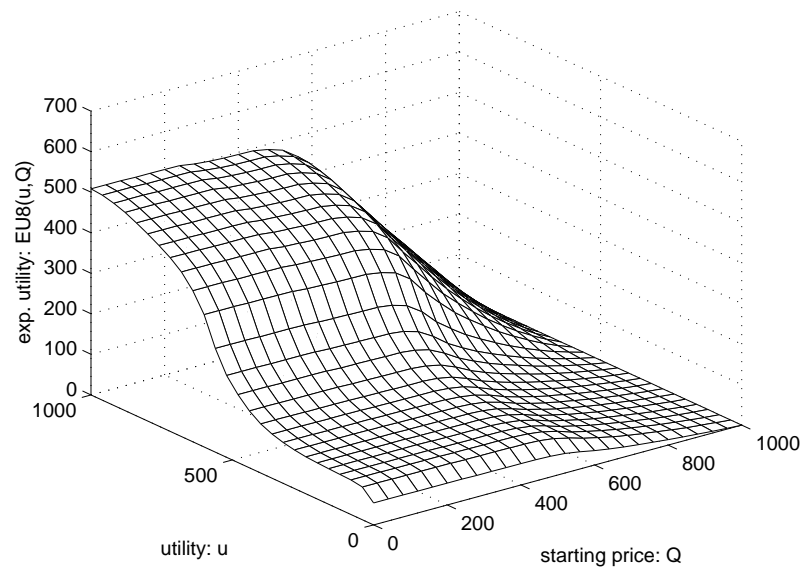


Figure 6.32: Expected utility $EU_r(u, Q)$ from Theorem 10 for round 8, in the case of the experiment derived distribution with $N = 64$ and $m = 16$.

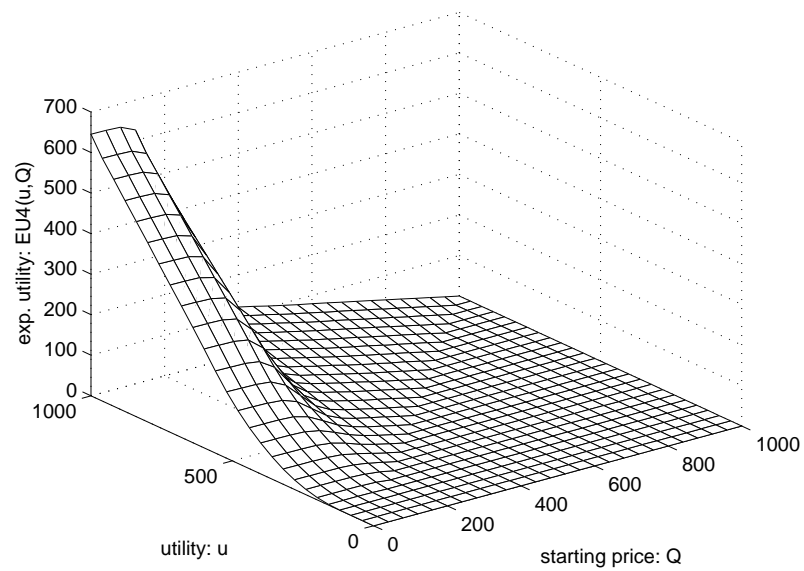


Figure 6.33: Expected utility $EU_r(u, Q)$ from Theorem 9 for round 4, in the case of the experiment derived distribution with $N = 64$ and $m = 16$.

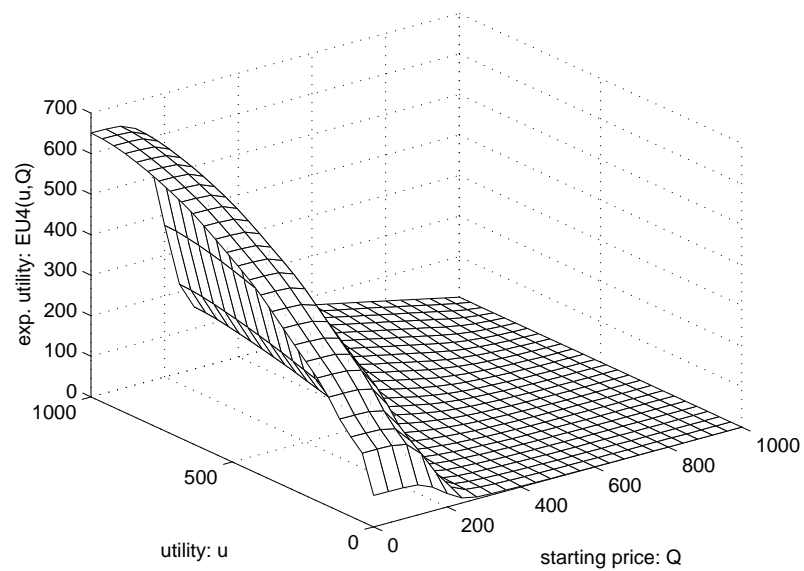


Figure 6.34: Expected utility $EU_r(u, Q)$ from Theorem 10 for round 4, in the case of the experiment derived distribution with $N = 64$ and $m = 16$.

Chapter 7

Experiments

In this chapter, we describe how the second part of our methodology is applied, and we present the controlled experiments we performed in order to determine the overall best combination of partial strategies for both instances of the TAC game. We present the conclusions we drew from them, concerning the tradeoffs and the performance of the strategies described in Chapter 5. After all the experiments have been presented in each case, we discuss which version of the agent was used in each TAC competition, and the performance of our agent against the agents of other research teams.

7.1 Performance Evaluation Between Competing Agents

In Chapter 3, we did not mention what statistical test is appropriate for evaluating the performance difference between competing agents. Since we are interested in the difference in score between agents that participate in the same games (same experiment), the most relevant statistical test is the **paired t-test**. Given that, in most experiments we use pairs, and, in general, we have more than 1 instance of a each version participating in every experiment, we compute the t-test for all possible combinations of instances. This means that 8 t-tests will be computed if we have 2 instances of version A and 4 of version B. *We consider the difference between the scores of the two versions to be significant, if all the tests produce values below 10% and, furthermore, if most of these values are less than 5%.* In most experiments that we present later on, we have run enough tests to have the values be well below 5%. Determining that an agent is performing better than another is very important for our conclusions.

On the other hand, in order to conclude that the differences are relatively similar, we are considerably less strict. The reason for this is that, on a number of experiments,

applying a paired t-test on the scores of two different instances of the same agent, yields probabilities of the two distributions being equal, which can be as low as 50%.¹ Furthermore, determining similarity in performance is somewhat less important than determining difference. Therefore, in general, we decided that the performance of two agents is relatively similar, if all the probabilities given by the t-test for all possible combinations of agents of those types, is above 50%.²

7.1.1 Reducing the Number of Required Runs By Estimating the Effect of Random Parameters

The biggest drawback in our methodology is probably the fact that in order to get results with a reasonable amount of confidence for each experiment, we need to run a large number of games. In general, even in the related work of other researchers, e.g. [73, 96, 99], it is suggested that the best way to achieve statistical significance is to run a large number of games. If one also considers the fact that quite a few experiments are needed in order to explore the strategy space, this leads to a huge number of games and that takes a very long time to run. The large number of games, which is needed, is mostly due to the fact that the TAC game has an overwhelming number of randomness involved. So even when two instances of the same agent are running, they can get considerably different results in the same experiment, if only a few games are played. This is more of a problem during the finals of the Trading Agent Competition, when a very small number of games are played. To partially alleviate this problem, the University of Michigan group decided to perform a linear regression of the scores an agent gets over the finals

¹Usually this value is around 70 to 75%, but in one instance it fell to 30%; this happened in the last experiment presented in Section 7.3.1.

²We might make an exception for one value, and allow it to be as low as 30%, if all the rest (at least 3 more combinations) are above 60%. That value is considered an outlier.

against a number of parameters that affect the scores; based on [48], we expect the bias in this method to go to 0 as the number of games increases. This was an interesting idea and a good starting point.³ However, we did not want to assume that two different games can be characterized just by these random parameters. In the case of our experiments, we have enough data to do more. Given that in most of our experiments we run pairs of the same agent, we can run a linear regression in order to minimize the difference between the scores of two instances of the same agent in the same game. This is the “least imposing” assumption that one could make about the scores in any game: that the same agent should perform similarly under similar conditions. We used this idea and we calculated the regressions and the adjustments for the experiment presented in Table 7.7. The average adjustments are presented next to the corresponding scores in the table. We also did the same for the experiment of Table 7.8. In both cases, the adjustments made by the agent were rather small, and it did not entirely succeed in equalizing the scores of the agents. However, an examination of the results demonstrates that, in most cases, the statistical confidence after adjusting the scores has increased (particularly in the first case), which indicates that we could get more accurate results with fewer experiments. Given that the adjustments that this process makes to the scores are rather small, we can reasonably argue that we are justified in doing this, while not altering the results of each experiment. For the record, the parameters found in each of the experiments to be the most relevant to the score (among the 10 parameters that we used) were:

1. the sum of the number of days an agent’s clients want their trips to last,
2. the sum of the rooms, per day, that an agent wants, weighted by the total demand for each day among all clients, and
3. the sum of the n highest valuations of its clients’ entertainment preferences, where

³They have since refined their analysis. For the latest details on how they do this see [99].

n depends on the length of the client's trip.

Another parameter that was included, in some cases, was the sum of the hotel bonuses for each client. The correlation of the linear regression predictions against the actual values was between $R = 0.3$ to $R = 0.35$ in all these cases, which is quite good, considering the high number of randomness of the TAC game.

We also applied the same process on the experiments presented in Table 7.9. While this increases, to some degree, the statistical significance of the results, we got the impression that it was a bit less accurate than in the previous cases that we had used this method. The correlation of the linear regression predictions against the actual values was between $R = 0.27$ to $R = 0.34$ in all the experiments of that table. For more information on this see Section 7.3.1.

7.2 Experiments for Determining TAC 2001-2003 Agent

Since in the TAC 2001 game, we have two possible tradeoffs. We need to explore the strategy space in both dimensions by varying the strategies for both tradeoffs. We also evaluate the performance difference when adding three individual features:

1. Modeling the flight auctions' prices and estimating the expected prices increase for each flight auction;
2. Using historical average prices when determining the Price Estimate Vectors;
3. Using the prices that constitute a Walrasian equilibrium as the values of the Price Estimate Vectors.

In the next sections, we provide the experimental results in each of these cases.

Notation

To distinguish between the different strategies (or versions) of the agent, we use the following notation:

- $WB - xyz$, if it is a version based on our TAC 2001 agent;
- $WB * xyz$, if it is a version based on our TAC 2002 and 2003 agent;⁴

where the parameters x, y, z denote

- x denotes the features being used in the optimizer (mainly), and can be
 1. $x = N$, if none of the features are used,
 2. $x = M$, if modeling of flight auctions' price is used,
 3. $x = A$, if, in addition to modeling, the historical average prices are used in the price predictions, and
 4. $x = W$, if, in addition to modeling and use of historical average prices, the agent used the prices from the Walrasian equilibrium at the beginning of the game.
- y denotes the strategy used for the flight ticket auctions and can be
 1. $y = 0$, if all tickets are bought at the beginning of the game (early bidder), which is one of the two boundary strategies,
 2. $y = 2$, if the other boundary strategy is used, which causes the agent to buy all tickets after two hotel room auctions have closed,⁵

⁴The reason why the two agents have different notations is because there was a significant number of changes and bug fixes between the two agents, which may or may not affect the results of the experiments.

⁵Not taking into account that this might be modified by feature $x = M$ (price predictions of flight auctions).

3. $y = 1$, if the agent uses the strategy that causes it to buy all tickets after the first hotel auction closes, and
 4. $y = S$, if, the strategy based on strategic demand reduction is used.
- z denotes the strategy used for the hotel room auctions and can be
 1. $z = L$, if the strategy of lowest aggressiveness is used, that is the agent bids reasonably close to the current bid price of the auction,
 2. $z = H$, if the other boundary strategy of highest aggressiveness is used, which causes the agent to bid progressively closer to the marginal utility,
 3. $z = M$, if the strategy of medium aggressiveness is used, which mixes the behavior of the boundary strategies based on the marginal utility,
 4. $z = P$, if the strategy, which places bids at a price equal to a weighted sum of bids that would be placed by the boundary strategies, is used, and
 5. $z = E$, if the equilibrium strategy from Chapter 6 is used.

7.2.1 Modeling the Flight Auctions' Prices

The first set of experiments was aimed at verifying our observation, that modeling the plane ticket prices improves the performance of the agent. We expected an improvement⁶, since the agent uses this information to bid later for tickets whose price will not increase much (therefore, achieving a greater flexibility at low cost), while bidding earlier for tickets whose price increases faster (reducing its cost). We ran 2 experiments with the following 4 versions: WB-N2L, WB-M2L, WB-M2M and WB-M2H. In the first, we ran 2 instances of each agent, while in the second we ran only one, and the other 4 slots were filled with the standard agent provided by the TAC support team. The

⁶A gain of 120 to 150 was expected, according to a rough estimate.

Table 7.1: Average scores of agents WB-N2L, WB-M2L, WB-M2M and WB-M2H. For experiment 1, the scores of the 2 instances of each agent type are also averaged. *The number inside the parentheses is the total number of games for each experiment, and this will be the case for every table.*

Experiments		WB-N2L	WB-M2L	WB-M2M	WB-M2H
Exp 1 (144)	agent 1	2087	2238	2387	2429
	agent 2	2087	2274	2418	2399
	average	2087	2256	2402	2414
Exp 2 (200)		3519	3581	3661	3656

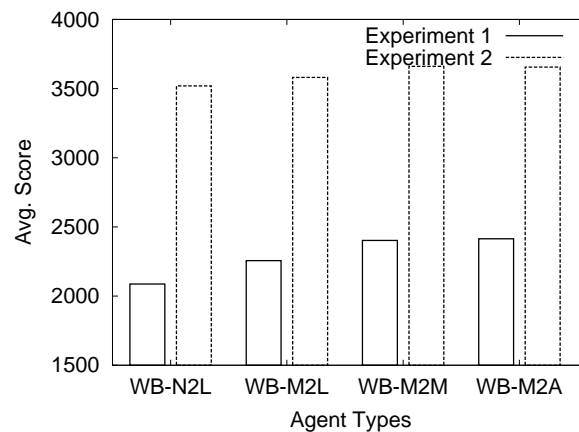


Figure 7.1: Average scores for some agents that do, and some that do not, model the flight ticket prices. Two different experiments are presented. For the exact scores, see Table 7.1.

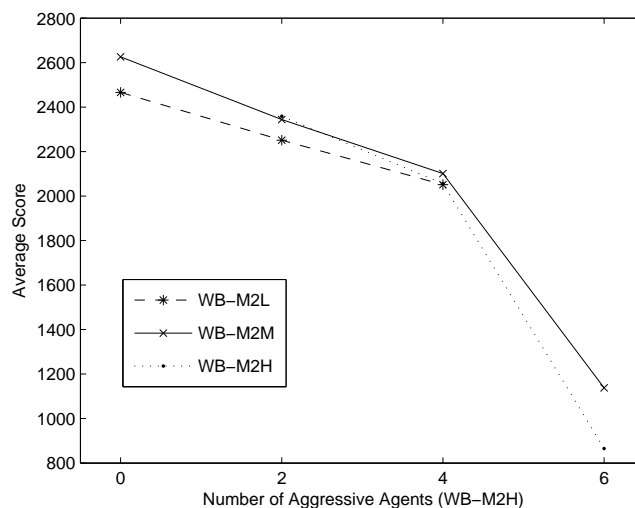


Figure 7.2: Changes in agents' average scores as the number of aggressive agents participating in the game increases. For the exact scores, see Table 7.2.

results are presented in Table 7.1, and in Figure 7.1. The agents which model the plane ticket prices perform better than agent WB-N2L, which does not do so. The differences between WB-N2L and the other agents are statistically significant, except for the one between WB-N2L and WB-M2L in experiment 2. We also observed that WB-M2L is outperformed by agents WB-M2M and WB-M2H, which in turn achieve similar scores; these results are statistically significant for both experiments. Having determined that this modeling leads to significant improvement, we concentrated our attention only to agents using this feature. This means that in all the remaining experiments all the agents use this feature.

7.2.2 Evaluating the Aggressiveness Tradeoff

The next experiment was designed to explore the tradeoff of bid aggressiveness in the hotel room auctions. As proposed by our methodology, we used agents WB-M2z ($z=L,M,H$), keeping all other partial strategies fixed, and we used a constant number of 2

Table 7.2: Scores for agents WB-M2L, WB-M2M and WB-M2H as the number of aggressive agents (WB-M2H) participating increases. In each experiment, agents 1 and 2 are instances of WB-M2M. The agents above the stair-step line are WB-M2L, while the ones below are WB-M2H. The averages scores for each agent type are presented in the next rows. In the last rows, \checkmark indicates statistically significant difference in the scores of the corresponding agents, while \times indicates statistically similar scores.

#WB-M2H	Agent Scores							
	1	2	3	4	5	6	7	8
0 (178)	2614	2638	2490	2463	2421	2442	2526	2455
2 (242)	2339	2350	2269	2265	2229	2241	2347	2371
4 (199)	2130	2073	2072	2029	2046	2098	2048	2033
6 (100)	1112	1165	796	843	920	884	848	898

#WB-M2H	Average Scores			Statistically Significant Difference?		
	WB-M2L	WB-M2M	WB-M2H	M2L/M2M	M2M/M2H	M2L/M2H
0 (178)	2466	2626	N/A	\checkmark		
2 (242)	2251	2344	2359	\checkmark	\times	\checkmark
4 (199)	2051	2101	2056	\times	\times	\times
6 (100)	N/A	1138	865		\checkmark	

instances of agent WB-M2M, while the number of agents WB-M2H was increased from 0 to 6. The rest of the slots were filled with instances of version WB-M2L. The result of this experiment are presented in Table 7.2, and in Figure 7.2. By increasing the number of agents which bid more aggressively, there is more competition between agents and the hotel room prices increase, leading to a decrease in scores. While the number of aggressive agents WB-M2H is less or equal to 4, the decrease in score is relatively small for all agents and is approximately linear with respect to the number of aggressive agents WB-M2H participating. The aggressive agents (WB-M2H) do relatively better in less competitive environments, and the non-aggressive agents (WB-M2L) do relatively better in more competitive environments, although the latter still do not perform good enough, compared to the WB-M2M and WB-M2H agents. Overall, WB-M2M (medium aggressiveness) performs comparably or better than the other agents in almost every instance. Agents WB-M2L are at a disadvantage compared to the other agents, because they do not bid aggressively enough to acquire the hotel rooms that they need. When an agent fails to get a hotel room it needs, its score suffers a double penalty: (i) it will have to buy at least one more plane ticket at a high price in order to complete the itinerary, or else it will end up wasting at least some of the other commodities it has already bought for that itinerary, and (ii) since the arrival and/or departure date will probably be further away from the customer's preference, and the stay will be shorter (hence, less entertainment tickets can be assigned), there is a significant utility penalty for the new itinerary. On the other hand, aggressive agents (WB-M2H) will not face this problem and they will score well in the case that prices do not go up. In the case that there are a lot of them in the game though, the price wars will hurt them more than other agents. The reasons for this are: (i) aggressive agents will pay more than other agents, since the prices will rise faster for the rooms that they need the most in comparison to other rooms, which are needed mostly by less aggressive agents, and (ii) the utility

Table 7.3: Scores for agents WB-M2M and WB-M0H as the number of early bidding agents (WB-M0H) participating increases. The agents above the stair-step line are WB-M2M, while the ones below are WB-M0H.

#WB-M0H	Agent Scores								Average Scores		Diff.?
	1	2	3	4	5	6	7	8	M2M	M0H	M/H
2 (343)	1607	1522	1564	1523	1497	1517	1531	1501	1517	1540	×
4 (282)	1398	1425	1401	1387	1292	1333	1265	1341	1403	1308	✓
6 (69)	1570	1602	1278	1178	1241	1289	1151	1207	1586	1224	✓

penalty for losing a hotel room becomes comparable to the price paid for buying the room, so non-aggressive agents suffer only a relatively small penalty for being outbid. Agent WB-M2M performs “reasonably well” in every situation, since it bids enough to maximize the probability that it is not outbid for critical rooms, and avoids “price wars” to a larger degree than WB-M2H. Based on these results, we did not use low aggressiveness agents in the next experiments, when we keep this part of the strategy fixed; we concentrate primarily on the medium aggressiveness ($z = M$) agents and secondarily on the high aggressiveness ($z = H$) agents.

7.2.3 Evaluating the “Paying For Flexibility” Tradeoff

The next set of experiments intended to further explore the trade-off of bidding early for plane tickets, against waiting longer in order to gain more flexibility in planning. Initially, we ran a smaller experiment with 2 instances of each of the following agents: WB-M2M and WB-M2H together with WB-M1M (which bids for most of its tickets at the beginning), and WB-M0H (which immediately buys all the plane tickets that it

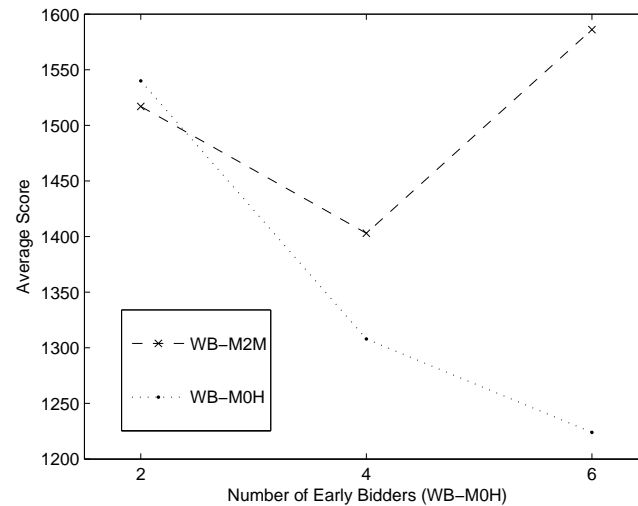


Figure 7.3: Changes in agents' average scores as the number of early bidders participating increases. For the exact scores, see Table 7.3.

needs and bids aggressively for hotel rooms.⁷ We ran 78 games and observed that WB-M2M scores slightly higher than the other agents, while WB-M2H scores slightly lower. These results are, however, not statistically significant. A larger experiment was done to examine the behavior of the two main strategies against each other. We varied the mixture of agents WB-M2M and WB-M0H (also called early-bidders, because they buy everything as soon as possible), as shown in Table 7.3, and in Figure 7.3. When only 2 of the agents were WB-M0H, the WB-M0H's scored on average close to the score of the WB-M2M's, but as their number increased, their score dropped and, when they are the majority, the WB-M0H's performed much worse than the WB-M2M's. In this case, the WB-M2M's try to stay clear of rooms whose price increases too much (usually, but not always, successfully), while the early-bidders do not have this choice, due to the reduced flexibility in changing their plans. One interesting result, which we did not

⁷An early-bidder must be aggressive, because, if it fails to get a room, it will pay a substantial cost for changing its plan, due to the lack of flexibility in planning.

Table 7.4: The effect of using historical averages in the PEVs. Early bidding agents benefit the most from this.

		WB*xSM	WB*xSH	WB*x2M	WB*x2H
#	x=A	1941	1887	1744	1677
games	x=M	1729	1645	1686	1706
(206)	Difference?	✓	✓	×	×

expect, is that the score of the WB-M2M's increases when there are 2 instances of them, compared to the case when there are 4. However, hotel room prices are higher in the former case, so this result seems contradictory! The explanation for this is that the prices tend to increase quite fast for the rooms that are needed by the early-bidders, so the 2 WB-M2M's avoid these rooms when possible and try to position themselves mostly on the other rooms, so they do not have to pay so much. This behavior also happens in the case that there are 4 WB-M2M's, but in this case there are many WB-M2M's, and when they try to move away from the rooms that the early-bidders want, they end up with similar rooms (so the reason it's harder to find the good deals is because they stop being "deals" much more often, once the other WB-M2M's go after them).

7.2.4 Using Historical Averages In Price Predictions

These results would normally allow us to conclude that it is usually beneficial not to bid for everything at the beginning of the game, but there is a minor catch: without using historical prices, the early-bidding agents buy goods "blindly". Therefore, we introduced this feature, and ran an experiment in which we examined the benefit that agents WB*M2M, WB*M2H, WB*MSM and WB*MSH gain if historical prices are used. Note that we did not use WB*M0z, because the agent WB*MSz also buys the

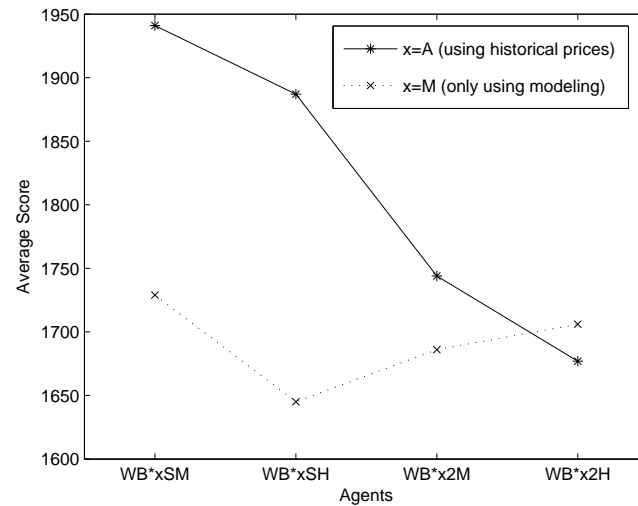


Figure 7.4: The effect of using historical averages in the PEVs. For the exact scores, see Table 7.4.

vast majority of its tickets at the beginning (but not all). The results are presented in Table 7.4, and in Figure 7.4. We observed that the agents that bid earlier, are the ones who benefited from the use of this feature, while the benefits for WB*M2M and WB*M2H are virtually non-existent. The increase of the price estimate has the effect that the planner generates itineraries, which use slightly fewer rooms than before. This decreases the price wars between agents and improves their scores.

7.2.5 Re-evaluating the “Paying For Flexibility” Tradeoff

The last experiment extends the experiment presented in Section 7.2.3. This time, we examine the effect on agents WB*AyM and WB*AyH (the medium and high aggressiveness with historical prices in the PEVs at the beginning of the game) when $y = 0$, $y = 2$ and $y = S$. Since $y = S$ is the intermediate strategy, we always keep 2 agents WB*ASz ($z=M,H$) in the mixture of agents and change the number of the other agents (which use the boundary strategies), as described by our methodology; half of these are

Table 7.5: Scores for agents WB*A2z, WB*ASz and WB*A0z (where z=M or H), as the number of early bidding agents (WB-A0z) participating increases. In each experiment, agent 1 is WB*ASM and 2 is WB*ASH. The agents above the stair-step line are WB*A2z, while the ones below are WB*A0z (the scores when z=M are presented in *italic*). In the last rows, we compared the scores for the M and H aggressiveness of the two versions; so M✓ in the A2z/Asz box means that the difference between A2M and ASM is significant.

#WB*A0z	Agent Scores							
	WB*ASM	WB*ASH	3	4	5	6	7	8
0 (413)	<i>2175</i>	2213	<i>1948</i>	<i>1954</i>	<i>1907</i>	1899	1939	1876
2 (650)	<i>2115</i>	2110	<i>2014</i>	<i>2031</i>	1841	1882	<i>2103</i>	2130
4 (438)	<i>2404</i>	2419	<i>2261</i>	<i>2207</i>	<i>2305</i>	<i>2390</i>	2408	2362
6 (1023)	<i>2430</i>	2442	<i>2370</i>	<i>2368</i>	<i>2364</i>	2384	2384	2378

#WB*A0z	Average Scores				Statistically Significant Difference?		
	A2M	A2H	A0M	A0H	A2z/ASz	ASz/A0z	A2z/A0z
0 (413)	<i>1936</i>	1904	N/A	N/A	M✓	H✓	
2 (650)	<i>2023</i>	1862	<i>2103</i>	2130	M✓	H✓	M× H× M✓ H✓
4 (438)	<i>2261</i>	2207	<i>2347</i>	2385	M✓	H✓	M× H× M✓ H✓
6 (1023)	N/A	N/A	<i>2367</i>	2382			M✓ H✓

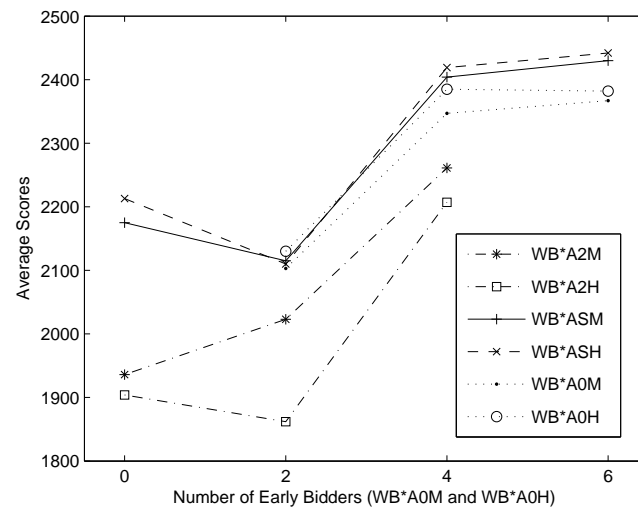


Figure 7.5: Changes in agents' average scores as the number of early bidders participating increases. For the exact scores, see Table 7.5.

of medium and half of high aggressiveness. The results are presented in Table 7.5, and in Figure 7.5. We observed that the strategy $y = 2$, which leaves the highest number of unpurchased tickets, performs worse than the other two. The other two perform similarly overall. The only case, in which the WB*ASz's performance is statistically better than that of the early bidders, is when there are lots of early bidders. From these results, we determine that the strategic demand agent is performing most consistently, and that is the reason we used it in the TAC. Another observation is that the scores of all agents tend to go up as the prices go higher. We believe that this is a result of the fact that the historical prices are used in the PEVs mainly at the beginning of the game and, when some auctions have closed, we do not any more. The later bidding agents ($y = 2$) observe the lower prices and try to purchase more rooms, which in turn drives the prices up; this happens only in the case that a lot of these agents participate. On the other hand, as their number decreases the economy becomes more efficient and all the agents profit.

Table 7.6: Scores for agents WB*AyM and WB*AyH (where $y=S$ or 0) as the number of aggressive agents (WB*AyH) participating increases. The agents above the stair-step line are WB*AyM, while below are WB*AyH (the scores when $y=0$ are presented in *italic*).

#WB*AyH	Agent Scores							
	1	2	3	4	5	6	7	8
2 (428)	2458	2423	2412	2394	2418	2400	2474	2377
4 (421)	2577	2591	2575	2560	2545	2615	2520	2515
6 (318)	2503	2330	2387	2382	2353	2355	2424	2366

#WB*AyH	Average Scores				Stat. Significant Difference?	
	WB*ASM	WB*A0M	WB*ASH	WB*A0H	ASM/ASH	A0M/A0H
2 (428)	2431	2404	2474	2377	×	×
4 (421)	2584	2568	2580	2518	×	×
6 (318)	2503	2330	2374	2382	✓	×

7.2.6 Re-evaluating the Aggressiveness Tradeoff

Having determined that the strategic ($y = S$) and early bidding ($y = 0$) agents are the best strategies for the plane tickets, we need to revisit our experiment for the hotel room bids, since in Section 7.2.2, we did it with the agents using strategy $y = 2$, which as was concluded from the last experiment is not the best strategy. We vary the number of aggressive agents (H) from 2 to 6, while reducing the number of medium aggressiveness agents (M). It should be noted that due to the fact that agents of low aggressiveness do not perform competitively in this setting, we only used agents of medium and high aggressiveness. We remedy this in the next experiment (Table 7.7), where we perform this test again with two intermediate strategies. Furthermore, half of them use strategy

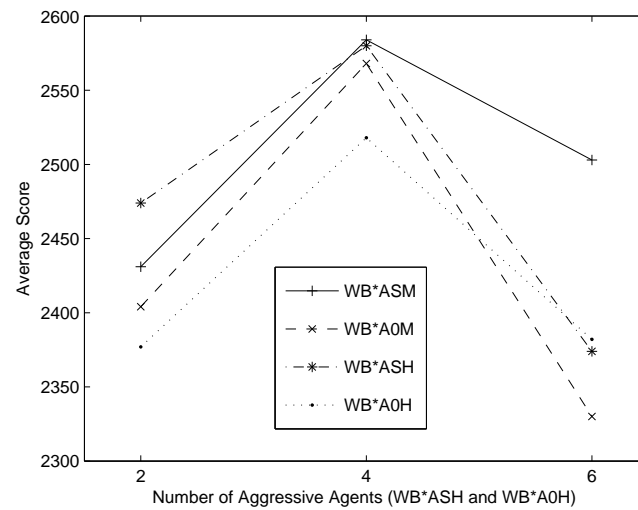


Figure 7.6: Changes in agents' average scores as the number of aggressive agents participating in the game increases. For the exact scores, see Table 7.6.

$y = 0$ and half $y = S$. The results are presented in Table 7.6, and in Figure 7.6. One can notice that, in most cases, the agents perform quite similarly to each other. Furthermore, agent WB*ASM outperforms the other agents when the aggressive agents are in the majority, and does statistically similar or better in all other cases. Another observation is that the overall scores remain similar, even when the number of aggressive agents increases (as opposed to the experiment presented in Table 7.2), mainly due to the fact that these agents use historical prices in the optimizer. Since, overall, agent WB*ASM performs at least as good as the other agents, both when we vary the strategy for the hotels and for the plane tickets. *It follows that we have explored the strategy space that we had, and we should use agent WB*ASM in TAC, which we did.*

Later, based on initial evaluation of the equilibria strategies presented in Chapter 6, we generated a different intermediate strategy for the hotel bids, that submits a bid equal to a weighted sum of the boundary strategies ($y = P$). As our methodology suggests, in order to evaluate it, we need to keep a fixed number of agents using the intermediate

Table 7.7: Scores for agents WB*ASL, WB*ASM, WB*ASP and WB*ASH as the number of aggressive agents (ASH) participating increases. In each experiment, agents 1 and 2 are WB*ASP and 3 and 4 are WB*ASM. The agents above the stair-step line are WB*ASL, while the ones below are WB*ASH.

#ASH	Agent Scores							
	1	2	3	4	5	6	7	8
2 (279)	3308 +6	3316-14	3310 -7	3304+12	3064+13	3120+12	3223+ 0	3268-22
4 (285)	2776 -5	2830-16	2798+19	2705 +8	2720-12	2732-17	2609+10	2689+14

#ASH	Average Scores				Significant Difference?		
	ASP	ASM	ASL	ASH	ASP/ASM	ASP/ASH	ASM/ASH
2 (279)	3312	3307	3092	3245	×	✓	✓
4 (285)	2803	2751	N/A	2687	×	✓	✓

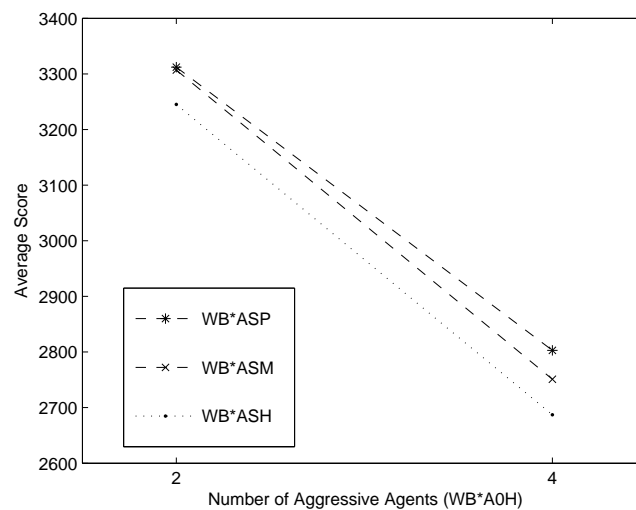


Figure 7.7: Changes in agents' average scores as the number of aggressive agents participating in the game increases. For the exact scores, see Table 7.7.

strategies $y = M$ and $y = P$ (we use 2 of each), while varying the number of agents using the boundary strategies $y = L$ and $y = H$. However, we decided not to run an experiment with 4 agents using strategy $y = L$, because from our experiments and our experiences in TAC, we know that it is unlikely to find many agents that use a non-aggressive strategy. The results of this experiment are presented in Table 7.7, and in Figure 7.7. We can see that the new strategy performs quite similarly to the medium aggressiveness agent. In fact, one might argue that this is expected, given that they both bid higher as time passes. Furthermore, as auctions close, and the probability p that the current round will be the last for an auction increases, the bids that agent WB*ASP places increase, and the same happens to the bids of agent WB*ASM because the marginal utility tends to increase significantly as time passes, even though the two versions do not increase their bids at the same rate. The number of games that we have run is not enough to guarantee statistical significance for all comparisons, however once the scores have been adjusted (see Section 7.1.1), the scores of WB*ASP and WB*ASM become statistically similar, and the difference between the scores of WB*ASP and WB*ASM on one side, and WB*ASH becomes significant. Based on the fact that WB*ASP and WB*ASM perform very similarly, either agent can be considered to have an “overall best strategy”.

7.2.7 Using Walrasian Equilibrium In Price Predictions

One of the areas where we have not put much effort, as far as our agent design is concerned, was the prediction of the closing prices for the hotel auctions. Most other agents have spent considerable effort in this area, whereas we tried to discount the effect of our lack of a good prediction, as a result of the bidding strategies that we have chosen. It was, however, our intention to implement some better prediction, in order to examine the result on the performance of the agent. To this effect, we chose to implement

Table 7.8: Evaluating the use of prices derived from a Walrasian equilibrium at the beginning of the game.

Agent		1	2	3	4	Average Score
#games (239)	WB*ASM	2279	2265	2224	2279	2262
	WB*WSM	2285	2211	2255	2240	2248

the price prediction algorithm of the Walverine team (it uses tatonment to compute the prices, for more details see [100]), since the authors show it to give some of the best predictions among all TAC agent implementations, and it does not use any knowledge of who the opponents are or how they behave (unlike most other methods used). We have tested this method by running four instances of agent WB*ASM using this prediction to set the prices used by the optimizer at the beginning of the game (let us call this agent WB*WSM) and four others that did not use this. We ran 239 games and the performance of the two agents was actually quite similar; even after adjusting the scores, we could not observe much of an appreciable difference between the two versions. The results of this experiment are presented in Table 7.8. The average scores for agents WB*ASM and WB*WSM were 2262 and 2248 respectively, which were statistically similar. This seems to indicate that the strategy we selected deals successfully with the errors (which are not catastrophic) in the price predictions that the optimizer uses. We only use this prediction at the beginning of the game though, because we believe that it is there, that accurate prices are the most useful in order to make a good guess about what plane tickets to buy, and also because this method requires a considerable amount of running time. So perhaps the agent might show some improvement if we allow this algorithm to set the price predictions throughout the game.

7.2.8 Agent Design For TAC Competition

In the 2001 TAC, the *WhiteBear* variant we used in the competition was **WB-M2M**. The reason for this is that we had only performed the first two experiments described in this section. The 4 top scoring agents (scores in parentheses) that year were: *livingagents* (3670), *ATTac* (3622), *WhiteBear* (3513) and *Urlaub01* (3421).

In the finals of TAC 2002 and in 2003, we used version **WB*ASM**, which performed the most consistently in our experiments. By that point, we had performed all of the experiments except the ones presented in Table 7.7 and 7.8. The performance of our agent in all the competition rounds that followed would seem to indicate that our choice was the correct one, since it managed to achieve the highest scores in almost all the rounds that followed and was always one of the top scoring agents (the exception being the finals of TAC 2003).

In 2002, the agent won the semi-finals and finals. The 4 top scoring agents were: *WhiteBear* (3556), *SouthamptonTAC* (3492), *Thalis* (3351) and *UMBCTAC* (3320).

In 2003, it won the preliminary rounds, and the semifinals with a significant score difference (a score of 3729 compared to the second ranked agents score of 3536), and was 3rd in the finals. This was partly due to a problem with the local server on which the agent was running. The 4 top scoring agents were: *ATTac01* (3200), *PackaTAC* (3163), *WhiteBear* (3142) and *Thalis* (3133).

In the 2001 TAC, the agents who performed the best were those that were buying almost everything at the beginning, e.g. *livingagents* [19], which followed a strategy similar to *WB-A0H*. This agent capitalized on the fact that the other agents were careful not to be very aggressive and that prices remained quite low. This strategy works well within the confines of an “efficient economy”. This was observed in our experiments as well. In the 2002 and 2003 competitions, more balanced agents (like ours) were performing better. Another observation that can be made is that the level of competence

of all agents has improved each time, so that in 2003 all finalists achieved comparable results. The agents would bid more aggressively and they would buy most of the plane tickets at the beginning. However, this did not lead to an inefficient economy, as most agents were more adaptive and restricted their itineraries to somewhat fewer rooms (in the same way that the inclusion of historical average prices in the optimizer produced the same effect on our agent).

7.3 Experiments for Determining TAC 2004-2005 Agent

Since in the TAC 2004 game we only have one tradeoff to examine, we need to explore the strategy space in just one dimension; thus we vary the strategies for the hotel auctions.

As notation, we are only going to use the z parameter (see notation in Section 7.2), which takes values L (i.e., low aggressiveness), H (i.e., highest aggressiveness), M (i.e., moderate aggressiveness), P (which mixed the boundary strategies with weight equal to probability p of the current round being the last one), and E (which is based on the equilibrium strategy).

7.3.1 Evaluating the Aggressiveness Tradeoff

The rules changed in 2004 leaving only one tradeoff to be examined: the hotel bidding strategy. Therefore, only one experiment set is necessary in this case, one in which the strategy for the hotel room bidding is varied. We use 2 agents of moderate (M) bidding aggressiveness and vary the number of agents of low (L) and high (H) aggressiveness. The results are presented in Table 7.9, and in Figure 7.8. The differences between the different versions of the agent are now much smaller; this is a result of the fact that the agents no longer buy the flight tickets early, and thus, are able to change their plans

Table 7.9: Scores for agents of low (L), moderate (M) and high (H), bidding aggressiveness as the number of aggressive agents participating increases. In each experiment, agents 1 and 2 are of moderate (M) aggressiveness. The agents above the stair-step line are of low aggressiveness (L), while the ones below are the most aggressive (H). The average scores for each agent type are presented in the next rows. In the last rows, ✓ indicates statistically significant difference in the scores of the corresponding agents, while × indicates statistically similar scores.

#H ag.	Agent Scores							
	1	2	3	4	5	6	7	8
0 (259)	3310-15	3275-17	3264+15	3295-21	3260 -8	3276 +1	3258+21	3223+24
2 (350)	3262-11	3292-15	3209 -7	3228 +4	3226 -7	3187 +5	3264+20	3240+10
4 (405)	3192 +3	3191+12	3160 -6	3199-12	3214+22	3268-11	3240-16	3193 +9
6 (413)	2899-31	2855 -9	2818+15	2842+23	2815+16	2934-20	2854 -1	2847 +6

#H agents	Average Scores			Stat. Significant Difference?		
	Low	Moderate	High	Low/Mod	Mod/High	Low/High
0 (259)	3263	3293	N/A	×		
2 (350)	3212	3277	3252	✓	×	✓
4 (405)	3180	3192	3229	×	×	✓
6 (413)	N/A	2877	2852		×	

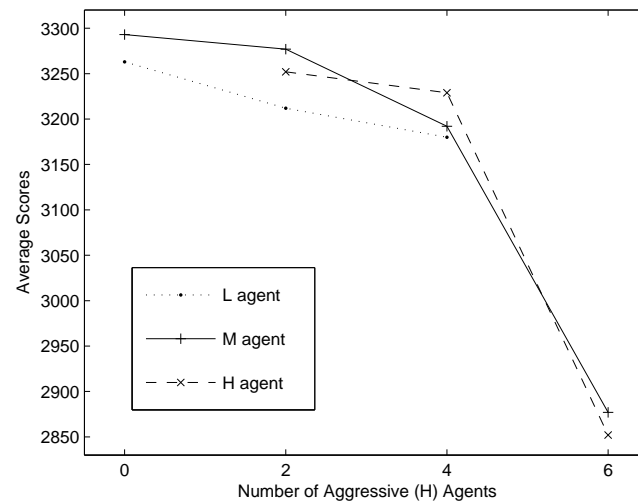


Figure 7.8: Changes in agents' average scores as the number of aggressive agents participating increases. For the exact scores, see Table 7.9.

much more easily, which in turn reduces the performance difference between the various strategies. From this experiment set, we can conclude that the low aggressiveness strategy is still dominated by the other two, but less so than in the case of TAC 2003. It is unclear which strategy (moderate or high aggressiveness) is best, as they perform quite similarly and the differences are not statistically significant. In fact, in most cases, there is a high probability of similarity in the score distributions. Once the method presented in Section 7.1.1 was applied to account for the randomness in the scores the results became statistically significant. However, the improvement was less than in the case of the TAC 2001 game. In TAC 2004, we used the moderate strategy, but based on these experiments the aggressive strategy would have performed quite well.

Once the equilibrium strategy was computed, we had 3 intermediate strategies to test, in addition to the two boundary ones. According to our methodology, we must organize a tournament among the intermediate strategies in order to determine the best strategy.

Table 7.10: Scores for agents of low (L), moderate (M) and high (H) bidding aggressiveness, as well as the equilibrium (E) agent, as the number of aggressive agents participating increases. In each experiment, agents 1 and 2 are equilibrium (E) agents, and agents 3 and 4 are of moderate (M) aggressiveness. The agents above the stair-step line are of low aggressiveness (L), while the ones below are the most aggressive (H). The average scores for each agent type are presented in the next rows. In the last rows, ✓ indicates statistically significant difference in the scores of the corresponding agents, while × indicates statistically similar scores. (The performance of the equilibrium agent is dominated by the performance of both the H and M agent, thus is not included in the last rows.)

#H agents	Agent Scores							
	1	2	3	4	5	6	7	8
0 (600)	3290	3294	3359	3364	3326	3327	3322	3335
2 (599)	3238	3242	3305	3342	3271	3257	3327	3352
4 (626)	2952	2955	3040	3051	3011	3013	3015	3007

#H agents	Average Scores				Statistically Significant Difference?		
	E	M	L	H	Low/Mod	Mod/High	Low/High
0 (600)	3292	3361	3327	N/A	✓		
2 (599)	3240	3323	3264	3339	✓	×	✓
4 (626)	2954	3046	N/A	3011		✓?	

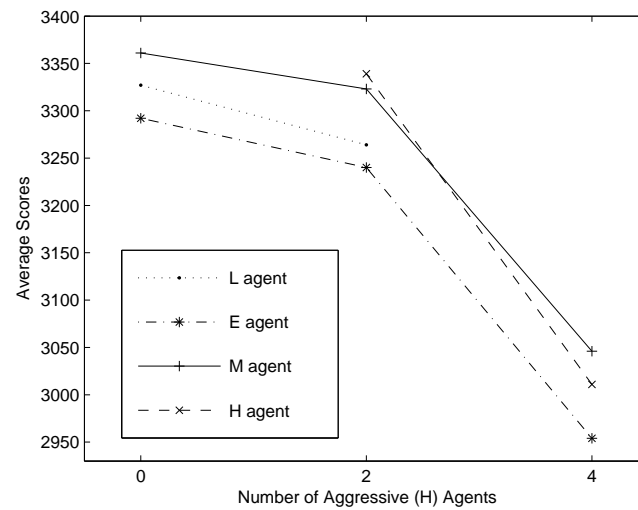


Figure 7.9: Changes in agents' average scores as the number of aggressive agents participating increases. For the exact scores, see Table 7.10.

In the first phase, we used intermediate strategies **M** and **E**. The results are presented in Table 7.10, and in Figure 7.9. Based on this, we can see that the **M** agents outperform the **E** agents. The most probable reason why this happens is because the equilibrium strategy assumes that the other agents do not deviate from the equilibrium, which is clearly not the case in this setting. This is a general problem with the Nash equilibrium, as opposed to dominant strategies, which is a stronger solution concept, and does not have this problem. Another possible reason is the fact that we assume that each agent is represented by 8 sub-agents, each of which wishes to buy 1 hotel room and these do not know the valuation of the other sub-agents, which in reality is not the case. It should be noted that the difference between the **M** and **H** agents is not statistically significant, but it looks like, with more experiments, it probably would be and thus is denoted by $\checkmark?$. This is not important however, as we found out what we wanted from this phase, namely that we should promote the **M** agent to the next phase of the tournament.

Thus, in the last phase, we used intermediate strategies **M** and **P**. The results are

Table 7.11: Scores for agents of low (L), moderate (M) and high (H) bidding aggressiveness as well as the P agent, as the number of aggressive agents participating increases. In each experiment, agents 1 and 2 are P agents, and agents 3 and 4 are of M agents. The agents above the stair-step line are of low aggressiveness (L), while the ones below are the most aggressive (H). The averages scores for each agent type are presented in the next rows. In the last rows, \checkmark indicates statistically significant difference in the scores of the corresponding agents, while \times indicates statistically similar scores. (The performance of the L agent is dominated by the performance of the P and the M agent in most cases, thus is not included in the last rows.)

#H agents	Agent Scores							
	1	2	3	4	5	6	7	8
0 (545)	3408	3399	3405	3430	3364	3381	3363	3385
2 (573)	3269	3282	3244	3240	3256	3214	3223	3259
4 (603)	2995	3004	2987	2974	3006	3015	2996	3008

#H agents	Average Scores				Statistically Significant Difference?		
	P	M	L	H	P/Mod	Mod/High	P/High
0 (545)	3403	3417	3373	N/A	\times		
2 (573)	3275	3242	3235	3241	\times	$\times?$	$\times?$
4 (603)	2999	2981	N/A	3006	\times	\times	\times

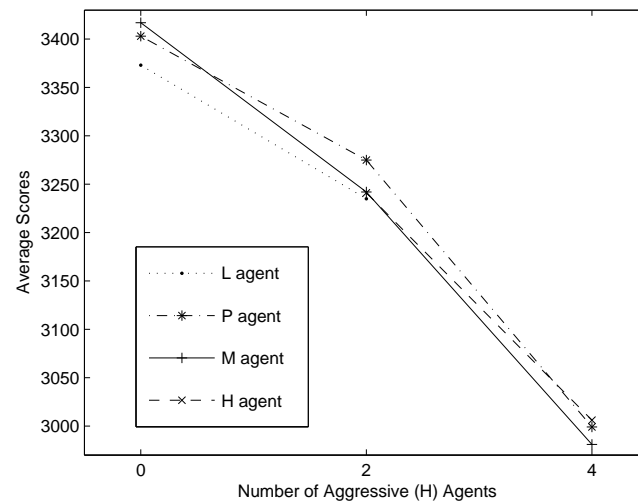


Figure 7.10: Changes in agents' average scores as the number of aggressive agents participating increases. For the exact scores, see Table 7.11.

presented in Table 7.11, and in Figure 7.10. Based on this, we can see that the M and the P agents outperform the L agents. The other agents seem to perform reasonably similar between themselves, and in fact, the performances were statistically similar for most cases as denoted in the table. The only exceptions to this rule were the performance of the P agents as opposed to that of the M and H agents in the case that 2 agents of each type participated. It looks like the results are similar (as half the t-test values are at about 50% or higher). It should be noted that in this experiment, we observed some rather low values when the t-test was applied to instances of the same agent; in one case, the value was 30%! Based on this, we can say that the P agent is probably performing best, but that the M and H agents are also performing rather similarly.

7.3.2 Agent Design For TAC Competition

In the 2004 TAC, the *WhiteBear* variant we used in the competition was the **M** agent. The agent won every round from the seeding round through to the finals with over-

whelming differences in score. In fact, in the finals *the score difference between our agent and the second one was statistically significant*, as the t-test value was less than 1%, which has never happened before in a TAC final; this happened despite the fact that only 35 games were run. The 4 top scoring agents (scores in parentheses) that year were: WhiteBear04 (4122), Walverine (3849), LearnAgents (3737) and SICS02 (3708).

In TAC 2005, we started by using once more agent **M**, but we are also experimenting with agent **P**, since it performed very well in our controlled experiments. So far WhiteBear05 has been on top of both preliminary rounds.

Chapter 8

Conclusions

Here we would like to summarize the work presented in this dissertation and provide directions for future work.

Our thesis statement was that by fully decomposing the problem faced by an agent participating in several auctions, it becomes feasible to analyze each component individually and generate strategies for the various components that lead to the creation of an extremely competent agent. We demonstrated this by using our methodology to design our agent *WhiteBear*, which became the most consistent (and best) performer at the TAC competitions (2001-2004).

In Chapter 3 the main contribution is the *novel methodology* that we provide, accompanied by the corresponding agent architecture. We describe in detail the information that the various modules exchange and are able to analyze each component individually. We describe the methodology for generating strategies and the experimental process through which the strategies are evaluated. Further work in this direction examines how the methodology can be extended to the case in which the utility of the acquired set of goods is not guaranteed but depends on various other parameters. An example of this is when the utility depends on the ability of other agents to provide better offers to our agent's clients.

In Chapter 4, we formulated the optimization problem faced by our agent and presented the optimization component of our TAC agent. As is the case with all the other modules, we designed the optimizer with scalability and speed in mind. We have shown that that an optimal algorithm is not necessary for competitive performance and we explored to a limited degree the effect of better price predictions when forming the optimization problem. This is the direction in which further work should proceed too,

namely to further explore the effect of a more accurate price prediction. An example of this would be to apply the Walrasian equilibrium algorithm to the whole game, despite the fact that it has a significant run time, or to examine the effect of a better learning algorithm.

In Chapter 5, we elaborated on how strategies for dealing with the various tradeoffs that the agent faces may be generated in a more systematic way than the mostly heuristic approaches that were prevalent in the literature. We presented the strategies generated for the various versions of the TAC games. Although we generated enough strategies to cover, to a sufficient degree, the entire strategy space, we are also examining the possibility of further automating this process and focusing the choices of an agent's designer. However we are not discounting the importance of empirical observations as they help to concentrate the search on the part of the strategy space where the most efficient candidate strategies lie.

In Chapter 6, we provided *several novel Bayes-Nash equilibria* for the case of m^{th} price auctions, when the closing time is randomly selected from a set of possible closing times. Furthermore, we extended our methodology to incorporate candidate strategies based on such equilibrium strategies. We solved this equilibrium for experimentally derived input distributions and applied this to our TAC agent. Even though its performance was dominated, this was done by the agent that won the 2004 TAC competition. However we are currently exploring ways of improving this strategy. We are trying to extend the equilibria to cover the case that an agent wishes to buy several units of the commodity sold in each auction. This is the setting of an individual hotel room auction, with almost no simplifications. Some limited initial work on this shows that, even for the case of a single round, one would probably need to solve a system of differential equations in order to compute the equilibrium.

In Chapter 7, we elaborated on how our proposed systematic method for exploring

the entire strategy space of our problem and determining the best combination of strategies can be applied. We were able to significantly limit the number of experiments that are needed in order to accomplish this, especially compared to related work. We also provided *a complete set of experiments* for determining our TAC agent entry, which performed very competitively. Furthermore, we discussed various issues concerning the comparisons between various agents and dealing with the inherent randomness of these problems. Further work would involve identifying other sources of randomness in the experiments and a more efficient method for reducing the statistical variance and thus the number of experiments needed in order to determine the best strategy.

BIBLIOGRAPHY

- [1] P. Anthony, W. Hall, V. Dang, and N. R. Jennings. Autonomous agents for participating in multiple on-line auctions. In *Proc IJCAI Workshop on E-Business and Intelligent Web*, pages 54–64, 2001.
- [2] K. Atkinson and W. Han. *Elementary Numerical Analysis*. John Wiley & Sons Inc., 2004.
- [3] E. Aurell, M. Boman, M. Carlsson, J. Eriksson, N. Finne, S. Janson, P. Kreuger, and L. Rasmusson. A trading agent built on constraint programming. In *Eighth International Conference of the Society for Computational Economics: Computing in Economics and Finance*, 2002.
- [4] R. Bejar, I. A. Vetsikas, C. P. Gomez, H. Kautz, and B. Selman. Structure and phase transition phenomena in the vtc problem. In *TASK-PI Darpa Workshop, Santa Fe, New Mexico*, Apr. 2001.
- [5] K. Binmore. *Fun and Games, A Text on Game Theory*. Lexington-Toronto: D.C. Heath and Company, 1992.
- [6] C. Boutilier, M. Goldszmidt, and B. Sabata. Sequential auctions for the allocation of resources with complementarities. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 527–534, 1999.
- [7] J. Boyan and A. R. Greenwald. Bid determination in simultaneous auctions: An agent architecture. In *Proc of the 3rd ACM Conference on Electronic Commerce (EC-01)*, pages 210–212, Oct. 2001.
- [8] P. Brewer and C. Plott. A binary conflict ascending price (bicap) mechanism for the decentralized allocation of the right to use railroad tracks. *International Journal of Industrial Organization*, 14, pages 857–886, 1996.
- [9] R. Brooks. Intelligence without reason. In *Proc of the 12th International Joint Conference on Artificial Intelligence*, 1991.
- [10] D. Cliff and J. Bruten. Minimal-intelligence agents for bargaining behaviors in market-based environments. *Technical Report HPL-97-91*.
- [11] W. Conen and T. Sandholm. Preference elicitation in combinatorial auctions: Extended abstract. In *Proceedings of the ACM Conference on Electronic Commerce*, 2001.
- [12] E. Dahlgren and P. R. Wurman. Packatac: A conservative trading agent. *ACM SIGecom Exchanges* 4(3), 38-45, 2004.

- [13] R. Das, J. Hanson, J. O. Kephart, and G. Tesauro. Agent-human interactions in the continuous double auction. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-01)*, pages 1169–1176, 2001.
- [14] S. de Vries and R. Vohra. Combinatorial auctions: A survey. *INFORMS Journal on Computing*, 15(3), 2003.
- [15] S. Fatima, M. Wooldridge, and N. R. Jennings. An analysis of sequential auctions for common and private value objects. In *Proc. AAMAS Workshop on Agent-Mediated Electronic Commerce*, 2005.
- [16] S. Fatima, M. Wooldridge, and N. R. Jennings. Optimal agendas for sequential auctions for common and private value objects. In *Proc. IJCAI Workshop on Game Theory and Decision Theory*, 2005.
- [17] S. Fatima, M. Wooldridge, and N. R. Jennings. Sequential auctions for objects with common and private values. In *Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-05)*, 2005.
- [18] Y. Feng, R. Yu, and P. Stone. Two stock-trading agents: Market making and technical analysis. In P. Faratin, D. C. Parkes, J. A. Rodriguez-Aguilar, and W. E. Walsh, editors, *Agent Mediated Electronic Commerce V: Designing Mechanisms and Systems*, volume 3048 of *Lecture Notes in Artificial Intelligence*, pages 18–36. Springer Verlag, 2004.
- [19] C. Fritsch and K. Dorer. Agent oriented software engineering for successful TAC participation. In *Proc of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2002.
- [20] D. Fudenberg and J. Tirole. *Game theory*. MIT Press, 1991.
- [21] Y. Fujishima, K. Leyton-Brown, and Y. Shoham. Taming the computational complexity of combinatorial auctions: Optimal and approximate approaches. In *Proc of the 16th International Joint Conference on Artificial Intelligence*, pages 548–553, Aug. 1999.
- [22] P. Garcia, E. Gimenez, L. Godo, and J. A. Rodriguez-Aguilar. Possibilisticbased design of bidding strategies in electronic auction. In *In Proceedings of the Thirteenth European on Artificial Intelligence (ECAI-98)*, pages 575–579, 1998.
- [23] E. Gimenez-Funes, L. Godo, J. A. Rodriguez-Aguilar, and P. Garcia-Calves. Designing bidding strategies for trading agents in electronic auctions. In *In Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS 98)*, pages 136–143, 1998.
- [24] S. Gjerstad and J. Dickhaut. Price formation in double auctions. *Games and Economic Behavior*, 22: 129, 1998.

- [25] P. J. Gmytrasiewicz and E. H. Durfee. A rigorous, operational formalization of recursive modeling. In *In Proceedings of the First International Conference on Multi-Agent Systems (ICMAS 95)*, pages 125–132, 1995.
- [26] D. K. Gode and S. Sunder. Allocative efficiency of markets with zerointelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119-137, 1993.
- [27] J. K. Goeree and T. Offerman. Competitive bidding in auctions with private and common values. *The Economic Journal* 113(489):598-613, 2003.
- [28] R. Gonen and D. Lehmann. Optimal solutions for multi-unit combinatorial auctions: Branch and bound heuristics. In *Proceedings of the ACM Conference on Electronic Commerce*, 2000.
- [29] R. Gonen and D. Lehmann. Linear programming helps solving large multi-unit combinatorial auctions. *Technical Report TR-2001-8, Leibniz Center for Research in Computer Science*, 2001.
- [30] A. Greenwald and P. Stone. Autonomous bidding agents in the trading agent competition. *IEEE Internet Computing*, April, 2001.
- [31] A. R. Greenwald. The 2002 trading agent competition: An overview of agent designs. *AI Magazine*, April 2003.
- [32] A. R. Greenwald and J. Boyan. Bidding algorithms for simultaneous auctions: A case study. In *Proc of the 3rd ACM Conference on Electronic Commerce (EC-01)*, pages 115–124, Oct. 2001.
- [33] A. R. Greenwald and J. O. Kephart. Shopbots and pricebots. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, pages 506–511, 1999.
- [34] A. R. Greenwald, J. O. Kephart, and G. J. Tesauero. Strategic pricebot dynamics. In *Proceedings of the ACM Conference on Electronic Commerce (EC-99)*, pages 58–67, Nov. 1999.
- [35] E. A. Hansen and S. Zilberstein. Monitoring and control of anytime algorithms: A dynamic programming approach. *Artificial Intelligence*, 126(1-2):139-157, 2001.
- [36] J. C. Harsanyi. Bargaining in ignorance of the opponent's utility function. *Econometrica* 18:155-162, 1962.
- [37] M. He and N. R. Jennings. Southamptonac: Designing a successful trading agent. In *Proc. European Conference on Artificial Intelligence (ECAI-02)*, 2002.

- [38] M. He and N. R. Jennings. Designing a successful trading agent using fuzzy techniques. *IEEE Trans on Fuzzy Systems* 12 (3), pages 389–410, 2004.
- [39] M. He, H. F. Leung, and N. R. Jennings. A fuzzy logic based bidding strategy for autonomous agents in continuous double auctions. *IEEE Trans on Knowledge and Data Engineering*, 15 (6), pages 1345–1363, 2003.
- [40] H. Hoos and C. Boutilier. Solving combinatorial auctions using stochastic local search. In *Proceedings of the 7th Conference on Artificial Intelligence (AAAI-00)*, pages 22–29, 2000.
- [41] B. Hudson and T. Sandholm. Effectiveness of query types and policies for preference elicitation in combinatorial auctions. In *In the proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2004)*, pages 386–393, 2004.
- [42] T. Ito, N. Fukuta, T. Shintani, and K. Sycara. Biddingbot: a multiagent support system for cooperative bidding in multiple auctions. In *Proceedings of the Fourth International Conference on MultiAgent Systems*, pages 399 – 400, July 2000.
- [43] P. Klemperer. Auction theory: A guide to the literature. *Journal of Economic Surveys* Vol13(3), July 1999.
- [44] V. Krishna. *Auction theory*. San Diego : Academic Press, 2002.
- [45] K. Larson and T. Sandholm. Bargaining with limited computation: Deliberation equilibrium. *Artificial Intelligence*, 132(2): 183-217, 2001.
- [46] K. Larson and T. Sandholm. Miscomputing ratio: Social cost of selfish computing. In *In the proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003)*, 2003.
- [47] K. Larson and T. Sandholm. Experiments on deliberation equilibria in auctions. In *In the proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2004)*, 2004.
- [48] P. L’Ecuyer. Efficiency improvement and variance reduction. In *Winter Simulation Conference*, pages 122–132, 1994.
- [49] D. Lehmann, L. O’Callaghan, and Y. Shoham. Truth revelation in rapid, approximately efficient combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce*, 1999.
- [50] C. Lemke and J. Howson. Equilibrium points of bimatrix games. *Journal of the Society for Industrial and Applied Mathematics* 12:413-423, 2001.
- [51] K. Leyton-Brown. *Resource Allocation in Competitive Multiagent Systems*. PhD thesis, Stanford University, August 2003.

- [52] K. Leyton-Brown, E. Nudelman, G. Andrew, J. McFadden, and Y. Shoham. A portfolio approach to algorithm selection. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2003.
- [53] K. Leyton-Brown, E. Nudelman, and Y. Shoham. Learning the empirical hardness of optimization problems: The case of combinatorial auctions. In *Constraint Programming (CP)*, 2002.
- [54] K. Leyton-Brown, M. Pearson, and Y. Shoham. Towards a universal test suite for combinatorial auction algorithms. In *ACM Conference on Electronic Commerce*, 2000.
- [55] K. Leyton-Brown, Y. Shoham, and M. Tennenholtz. An algorithm for multi-unit combinatorial auctions. In *American Association for Artificial Intelligence (AAAI)*, 2000.
- [56] K. Leyton-Brown, Y. Shoham, and M. Tennenholtz. Bidding clubs in first-price auctions. In *American Association for Artificial Intelligence (AAAI)*, 2002.
- [57] R. Lipton, E. Markakis, and A. Mehta. Playing large games using simple strategies. In *Proceedings of the ACM Conference on Electronic Commerce*, 2003.
- [58] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic theory*. New York: Oxford University Press, 1995.
- [59] R. P. McAfee and J. McMillan. Auctions and bidding. *Journal of Economic Literature*, 25:699–738, June 1987.
- [60] R. D. McKelvey, A. M. McLennan, and T. L. Turocy. Gambit: Software tools for game theory, version 0.2005.06.13. <http://econweb.tamu.edu/gambit>. 2005.
- [61] P. Milgrom. The economics of competitive bidding: A selective survey, social goals and social organization. *A Volume in Honor of Elisha Pazner, edited by L. Hurwicz, D. Schmeidler and H. Sonnenschein, Chapter 9*, pages 261–289, 1985.
- [62] P. Milgrom. Putting auction theory to work: The simultaneous ascending auction. *Technical Report 98-0002, Department of Economics, Stanford University*, 1998.
- [63] J. Nash. The bargaining problem. *Journal of Conflict Resolution*, 1950.
- [64] J. Nash. Equilibrium points in n-person games. In *Proc. of the National Academy of Sciences* 36:48-49, 1950.
- [65] N. Nisan. Bidding and allocation in combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce*, 2000.
- [66] N. Nisan and A. Ronen. Computationally feasible vcg mechanisms. In *Proceedings of the ACM Conference on Electronic Commerce*, 2000.

- [67] E. Nudelman, J. Wortman, K. Leyton-Brown, and Y. Shoham. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In *Proc of the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 04)*, 2004.
- [68] D. C. Parkes. *Iterative Combinatorial Auctions: Achieving Economic and Computational Efficiency*. PhD thesis, Department of Computer and Information Science, University of Pennsylvania, May 2001.
- [69] D. C. Parkes and L. H. Ungar. Iterative combinatorial auctions: Theory and practice. In *Proceedings of the 7th Conference on Artificial Intelligence (AAAI-00)*, pages 74–81, 2000.
- [70] R. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a nash equilibrium. In *Proceedings of National Conference on Artificial Intelligence (AAAI-04)*, 2004.
- [71] C. Preist, , A. Byde, and C. Bartolini. Economic dynamics of agents in multiple auctions. In *Proceedings of the International Conference on Autonomous Agents*, 2001.
- [72] C. Preist, C. Bartolini, and I. Phillips. Algorithm design for agents which participate in multiple simultaneous auctions. In *Agent Mediated Electronic Commerce III (LNAI)*, Springer-Verlag, Berlin, pages 139–154, 2001.
- [73] D. M. Reeves, K. M. Lochner, S. Cheng, and R. Suri. Approximate strategic reasoning through hierarchical reduction of large symmetric games. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-05)*, July 2005.
- [74] J. A. Rice. *Mathematical Statistics and Data Analysis*. Duxbury Press, California, 1995.
- [75] M. Rothkopf, A. Pekec, and R. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8), pages 1131–1147, 1998.
- [76] T. Sandholm. An algorithm for optimal winner determination in combinatorial auctions. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-99)*, 1999.
- [77] T. Sandholm. Distributed rational decision making. In *Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence*, G. Weiss, ed., MIT Press, pages 201–258, 1999.
- [78] T. Sandholm and S. Suri. Improved algorithms for optimal winner determination in combinatorial auctions. In *Proc 7th Conference on Artificial Intelligence (AAAI-00)*, pages 90–97, July 2000.

- [79] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. Cabob: A fast optimal algorithm for combinatorial auctions. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-01)*, 2001.
- [80] Y. Shoham and M. Tennenholtz. Fair imposition. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-01)*, 2001.
- [81] P. Stone, M. L. Littman, S. Singh, and M. Kearns. ATTac-2000: an adaptive autonomous bidding agent. In *Proc 5th International Conference on Autonomous Agents*, pages 238–245, May 2001.
- [82] P. Stone, R. Schapire, M. Littman, J. Csirik, and D. McAllester. Attac-2001: A learning, autonomous bidding agent. In *Agent Mediated Electronic Commerce IV. LNCS, volume 2531. Springer Verlag, Berlin.*, 2002.
- [83] M. Tennenholtz. Some tractable combinatorial auctions. In *Proceedings of the 7th Conference on Artificial Intelligence (AAAI-00)*, 2000.
- [84] H. R. Varian. Economic mechanism design for computerized agents. In *Proceedings of the First Usenix Conference on Electronic Commerce*, 1995.
- [85] I. A. Vetsikas and B. Selman. Using equilibria to handle design tradeoffs of autonomous trading agents (in preparation).
- [86] I. A. Vetsikas and B. Selman. Design tradeoffs for autonomous trading agents. In *Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming (CP-02)*, pages 790–791, 2002.
- [87] I. A. Vetsikas and B. Selman. Whitebear: An empirical study of design tradeoffs for autonomous trading agents. In *Proceedings AAAI 2002 Workshop on Game Theoretic and Decision Theoretic Agents*, 2002.
- [88] I. A. Vetsikas and B. Selman. A principled study of the design tradeoffs for autonomous trading agents. In *Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-03)*, pages 473–480, 2003.
- [89] I. A. Vetsikas and B. Selman. A methodology and equilibria for design tradeoffs of autonomous trading agents. In *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-04)*, pages 1286–1287, 2004.
- [90] I. A. Vetsikas and B. Selman. Autonomous trading agent design in the presence of tradeoffs. In *Proceedings of the 7th International Conference on Electronic Commerce, Xian, China*, Aug. 2005.
- [91] I. A. Vetsikas and B. Selman. Bayes-nash equilibria for m-th price auctions with multiple closing times. *ACM SIGecom Exchanges (to appear)*, 2005.

- [92] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 18:8-37, 1961.
- [93] B. von Stengel. Computing equilibria for two-person games. In *Aumann, R., and Hart, S., eds., Handbook of Game Theory, volume 3:1723-1759*, 2002.
- [94] P. Vytelingum, R. K. Dash, E. David, and N. R. Jennings. A risk-based bidding strategy for continuous double auctions. In *Proc. 16th European Conference on Artificial Intelligence*, pages 79–83, 2004.
- [95] W. E. Walsh, R. Das, G. Tesauro, and J. O. Kephart. Analyzing complex strategic interactions in multi-agent systems. In *Proceedings of the AAAI-02 Workshop on Game Theoretic and Decision Theoretic Agents*, 2002.
- [96] W. E. Walsh, D. C. Parkes, and R. Das. Choosing samples to compute heuristic-strategy nash equilibrium. In *Proc. AAMAS Workshop on Agent Mediated Electronic Commerce (AMEC V)*, Melbourne, Australia, 2003.
- [97] R. Weber. Making more from less: Strategic demand reduction in the fcc spectrum auctions. *Journal of Economics and Management Strategy Vol6(3)*, pages 529–548, 1997.
- [98] M. P. Wellman, J. K. MacKie-Mason, D. M. Reeves, and S. Swaminathan. Exploring bidding strategies for market-based scheduling. In *Proceedings of the ACM Conference on Electronic Commerce*, 2003.
- [99] M. P. Wellman, D. M. Reeves, K. M. Lochner, and R. Suri. Searching for walverine 2005. In *Proceedings IJCAI 2005 Workshop on Trading Agents Design and Analysis*, 2005.
- [100] M. P. Wellman, D. M. Reeves, K. M. Lochner, and Y. Vorobeychik. Price prediction in a trading agent competition. In *Journal of Artificial Intelligence Research*, 2004.
- [101] M. P. Wellman, P. R. Wurman, K. O’Malley, R. Bangera, S. de Lin, D. Reeves, and W. E. Walsh. Designing the market game for TAC. *IEEE Internet Computing, April*, March/April 2001.
- [102] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2), pages 115–152, 1995.
- [103] P. R. Wurman and M. P. Wellman. AkBA: A progressive, anonymous-price combinatorial auction. In *Proceedings of the ACM Conference on Electronic Commerce (EC-00)*, pages 21–29, 2000.
- [104] R. Yu and P. Stone. Performance analysis of a counter-intuitive automated stock-trading strategy. In *Proceedings of the Fifth International Conference on Electronic Commerce (ICEC-03)*, October 2003.

- [105] E. Zurel and N. Nisan. An efficient approximate allocation algorithm for combinatorial auctions. In *Proceedings of the ACM Conference on Electronic Commerce*, 2000.