AUTOMATED AUCTION MECHANISM DESIGN WITH COMPETING MARKETPLACES

by

JINZHONG NIU

A dissertation submitted to the Graduate Faculty in Computer Science
in partial fulfillment of the requirements for the degree of Doctor of Philosophy,
The City University of New York

2011

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

Date

Simon Parsons
Chair of Examining Committee

Date

Theodore Brown
Executive Officer

Susan L. Epstein

Peter McBurney

Elizabeth Sklar

Supervision Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

AUTOMATED AUCTION MECHANISM DESIGN WITH COMPETING MARKETPLACES

by

JINZHONG NIU

Advisor: Professor Simon Parsons

Resource allocation is a major issue in multiple areas of computer science. Auctions are commonly used in optimizing resource allocation in these areas, since well designed auctions achieve desirable economic outcomes including high allocative efficiency and fast response to supply and demand changes.

This dissertation presents a *grey-box* approach to automated auction mechanism design using reinforcement learning and evolutionary computation methods. In contrast to the traditional approaches that try to design complete auction mechanisms manually, which is tedious and error-prone, the grey-box approach solves the problem through an automated search in a parameterized space of auction mechanisms. This space is defined by a novel, parameterized structure for auction mechanisms—a big white box—and a set of auction rules—each as a small black box—that can fit into the structure. The grey-box approach uses reinforcement learning to explore the composition of the structure, relates the performance of auction mechanisms to that of auction rules that form

the mechanisms, and utilizes a Hall of Fame, a technique from evolutionary computation, to maintain viable auction mechanisms. The evaluation of auction mechanisms in the grey-box approach is conducted through a new strategic game, called CAT, which allows multiple marketplaces to run in parallel and compete to attract traders and make a profit. The CAT game helps to address the imbalance between prior work in this field that studied isolated auctions and the actual competitive situation that marketplaces face.

Experiments were carried out to examine the effectiveness of the grey-box approach. A comparison against the genetic algorithm approach showed that the grey-box approach was able to produce mechanisms with significantly better overall performance. The best produced mechanisms from the grey-box experiments were able to outperform both the standard mechanisms which were used in evaluating sampled mechanisms during the grey-box search and carefully hand-coded mechanisms which won tournaments based on the CAT game. These best mechanisms also exhibited better performance than some existing mechanisms from the literature even when the evaluation did not take place in the context of CAT games.

To whom I love and who loves me . . .

Acknowledgements

First, I would like to express my deep gratitude to my advisor, Professor Simon Parsons, for his supervision and guidance and for his patience on the long journey that I have taken. His commitment and rigor in research set a high standard and an excellent example for me in pursuing my own career. If Simon is like the chief scientist in the Agents Lab at CUNY, Professor Elizabeth Sklar plays the role of the CEO of the group and acted as a second advisor to me. Her neat organization and clear articulation made group meetings and many other occasions a very pleasant experience. I sincerely appreciate Simon and Betsy's willingness to go through with me the ups and downs of my life over the years, and the support they provided.

I had been in the team that was led by Professor Peter McBurney organizing the CAT competitions, which set the stage for my dissertation work. Peter is a great leader, has a broad vision of research, and is a collaborator whom I always find stimulating and comfortable to work with. I am greatly indebted to Peter for his financial support and invaluable career advices. Team work has been the key in the success of the CAT events and made it possible for me to carry out my research. I would like to thank the entire team, particularly Kai Cai for his close collaboration on a daily basis in building the software package of JCAT, Dr. Steve Phelps for his pioneering work that built the solid foundation for my work, and Dr. Enrico Gerding for his help in building JCAT and in preparing publications.

A special thanks goes to Distinguished Professor Gilbert Baumslag for the inspiration and excitement he, as a world-class scientist, brought to me and for his financial support during the first two years of my doctoral studies, and to Professor Ted Brown for his constant support during my years in the Ph.D. Program of Computer Science at the CUNY Graduate Center. My thanks also goes to Professor Xin Yao and Dr. Maria Fasli for allowing me to continue my dissertation research while I worked with them respectively at the University of Birmingham and the University of Essex, UK and for their invaluable inputs that made this work only more solid. I would like to thank Professor Susan Epstein and Professor Leanne Ussher for their time and effort in reading my dissertation and their help that made this dissertation what it is now. My thanks also goes to the research computing team at the CUNY Graduate Center for their excellent job in maintaining the Linux cluster on which much of the experimental work in this dissertation was carried out.

Being part of the Agents Group at CUNY, I have had the chance to enjoy the life or panic together with M. Q. "Sumon" Azhar, Kai Cai, Arif Tuna Ozgelen, Yuqing Tang, and other Agents folks. I wish them the best in their research and future career. I have been blessed and helped in different ways by many friends during my studies, for which I particularly would like to thank Bin Tang, Haizheng Zhang, Joe Driscoll, Bin Wu, Qingmin Guo, Guanjun Feng, and Yi Feng.

I owe my deepest appreciation to my entire family for their unconditional support and love. My wife, Wu, more than anybody else, gave me the strength and hope in my arduous journey of pursuing my Ph.D. and sacrificed her own career to take care of the whole family while I was away. My son, York, has been such a good boy and enjoyed his staying wherever he was brought, with or without the company of parents. My parents and parents-in-law have always had faith in me and encouraged me to pursue my academic career. My sister and brother have always made themselves available out there to share my joy and frustration, and tried their best to avoid asking me about

my progress. For all these and those that cannot be enumerated here, I thank you, and I love you all.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Markets, as ‘an invisible hand’, have shown their effectiveness in matching supply and demand, and allocating resources. *Auctions* provide opportunities of trading in marketplaces with strict regulations governing the information available to traders in the marketplaces and the possible actions they can take. They have been widely used in: structuring stock or futures exchanges, selling collectible items, choosing offers of goods or services in government procurements, and allocating computational resources in distributed systems. This is due to the fact that auctions, when well designed [Klemperer, 2002], achieve desired economic outcomes like high *allocative efficiency*. The creation of well-designed auctions in electronic commerce and computer-based control presents a grand challenge for auction mechanism design.

Traditionally, marketplaces have only involved human interactions, whereas in ecommerce, computer programs are widely used to make decisions on behalf of human traders in order to process a much higher volume of information at a much faster speed. As a result, auction mechanism design is no longer the exclusive domain of economists, but becomes an inter-disciplinary area where fields including distributed artificial intelligence, theoretical computer science, economics, and game theory meet. The design of competitive programs for algorithmic trading and electronic auction mechanisms that are both computationally feasible and economically efficient are, in particular, interesting to computer scientists.

1.1 Auction mechanisms in electronic commerce

In the Internet era, ecommerce has flourished and penetrated almost every corner of human life. The greater amount of available information, the lower cost of communication, and other reductions in economic frictions makes the world 'flatter' than ever before [Greenwald et al., 2003; Kephart, 2002].[1]

In financial markets, traders have continuously turned to automated algorithmic trading services to deal with faster transactions and more complex market dynamics [Schwartz et al., 2006b]. According to an article from *The Economist* [The Economist, 2007], algorithmic trading accounted for a third of all share trades in the United States in 2007, and the figure was estimated around 70% in 2009 [Clark, 2010]. News and events usually affect market predictions and lead to high price volatility, which in turn creates opportunities for arbitrage between markets. Unpredictable dynamics and complex linkages between markets make more robust, efficient market mechanisms very desirable.

Online auction sites like eBay provide a way for consumers to buy a wide range of items, such as common consumer electronic products and broken laser pointers.[2] Since its establishment in 1995, eBay has expanded into dozens of countries and now makes billions of dollars each year. The auction mechanisms used by eBay and other successful auction sites however are not perfect. For example, an eBay auction typically finishes at a fixed time, allowing a bidder to bid only

[1] A relevant issue is that researchers commonly believe that electronic marketplaces help to promote competition and increase allocative efficiency, but they often disagree on what changes brought by electronic marketplaces actually led to these results [Li et al., 2006].

[2] According to `www.wikipedia.org`, the first item sold through eBay was a broken laser pointer for $14.83. The buyer turned out to be a collector of these items.

moments before the auction terminates and steal a deal from bidders who would offer higher prices if given the chance [Greenwald, 2006]. This means a loss of revenue for both sellers and eBay. Another issue, and one to which many researchers have paid much attention, is that eBay runs many *simultaneous sequential auctions* [Gerding et al., 2007; Juda and Parkes, 2006]. In other words, on eBay, hundreds, even thousands, of on-going auctions may sell the same kind of goods. It is difficult for a potential buyer to select which auction to bid in. As a result, a successful bid in one auction may be lower than a failed bid in another, leading to complaints from both sellers and bidders, lower efficiency of the auctions and, in time, less revenue for eBay.

Electronic auctions have also been used to sell things that are not goods in a traditional sense. For example in *sponsored search*, search engines like Google and Yahoo, in the role of publisher, typically use auctions to select and show relevant advertisements along with search results on their web sites. For each keyword-based search query, an *ad auction* is run to select bids from advertisers. Each selected advertiser provides an ad to display in one of a certain number of *ad positions* on the search result page. Better positions, which draw more attention from users, are allotted to advertisers that bid higher. An advertiser usually pays on a *per-click* basis rather than on the *per-impression* basis in the traditional media. Publishers have commonly used variants of an auction mechanism called the *generalized second-priced auction* to determine winning bids from advertisers and their ad positions. Although ad auctions generate many dollars in income each year for these companies, the issue of how to analyze the current practices and design more effective ad auction mechanisms is still a major concern. For instance, Lahaie and Pennock [Lahaie and Pennock, 2007] compared the ranking rule used by Yahoo—based on the prices of bids—and that used by Google—based on the expected profits of bids to Google, and concluded that neither rule consistently outperforms the other.

All these scenarios from e-commerce challenge the designers of electronic auction mechanisms to design more desirable mechanisms.[3] This opens up new lines of research in computer science, such as inventing new algorithms for deciding the winning bid in auctions [Lehmann et al., 2006], deciding how best to bid in multiple auctions [Schwartz et al., 2008], and how to build the software infrastructure to run such auctions [Niu et al., 2008c].

1.2 Auction mechanisms in agent-based computing

The Internet also significantly boosts the adoption of distributed computing, in particular *agent-based computing*. From an AI perspective, an *agent* is a computational entity that perceives and acts upon its environment. It is usually assumed to be,

rational: it has some goal and carries out tasks that lead to the goal;

autonomous: it makes decisions based on its own knowledge instead of being controlled by an external entity; and

adaptive: it adjusts its behavior dynamically over time, allowing it to respond to changes in its environment.

An agent does not generally operate in isolation, and may interact with other agents directly or indirectly via the environment. It is often natural and convenient to view a computer system as a collection of multiple interacting agents, since this scheme grasps the very nature of what such a system tries to model in the real world. As a result, the agent-based paradigm has grown and flourished since 1980s [Wooldridge, 2001].

Taking a programming language perspective, this agent-based paradigm is an extension of *structured programming* (SP) and *object-oriented programming* (OOP). SP increases the reusability

[3]Designers of trading agents and designers of electronic marketplaces usually face an arms race, and both sides try to take advantage of the weaknesses on the other side and maximize their own profit.

of data by declaring data structure types and the reusability of functionality by defining functions and methods. OOP further encapsulates data and actions into objects and allows access to objects only via interfaces, building highly cohesive classes and maintaining loose coupling between those classes. However, both SP and OOP basically feature central control,[4] which decides when and what actions should be taken upon which piece of data. When the complexities of data and actions increase, the complexity of the control strategy will increase exponentially, making the task of system design formidably difficult. The agent-based paradigm, however, slices up the central control, grants the individual agents local control upon their own behaviors, and relies on the interactions between agents to achieve system goals. According to Maes [Maes, 1994], interactive dynamics can build complexity from simple components.[5] A system built in this bottom-up fashion is more robust, more flexible, and more fault-tolerant than those organized in a pre-programmed, top-down way.

Despite these advantages, the agent-based paradigm has its own difficulties, in particular those in solving the following two problems:

1. how to design strategies for rational agents to maximize their individual utilities?

2. how to design an interaction mechanism so as to achieve desired global outcomes?

In different scenarios, these problems present a different level of challenge to a system designer. When agents sit inside the boundary of the system, the first problem is not part of the designer's concern, meaning that the utility of each individual agent is either proportional to the global outcomes or not a concern at all. In this case, the system designer only needs to come up with an

[4] Strictly speaking, both SP and OOP may have distributed control, depending upon the concrete implementation, however distributed control is an inherent feature in the agent-based paradigm and is not so in SP or OOP.

[5] Maes [Maes, 1994] described three such scenarios that take place at different levels: (1) simple internal modules that work together can lead to emergent functionality; (2) simple atomic capabilities together with feedback mechanisms can produce complex behaviors; and (3) agents with simple behaviors can compose a social system that can exhibit advanced structures or functionality.

interaction mechanism that performs well with a fixed set of agent strategies. When the agents themselves are not part of the system that is being constructed and their utilities do not have a direct connection to the global outcomes, the interaction mechanism needs to perform well with virtually any combination of agent strategies. The system designer must take care to avoid being taken advantage of by the designer of individual agents and improve the interaction mechanism when a flaw is found.[6]

A major tool for multi-agent system designers has been *game theory* (GT). GT provides a framework for studying strategic, interacting individuals and solution concepts—usually various equilibria—with the assumption of the rationality of individuals. GT thus helps to compare the outcomes of an interaction mechanism to the optimal ones in theory, but it does not give a dynamic model that explains how to reach optimal outcomes, nor presents much guidance on how to maximize global outcomes when some agents in the system are not as rational as presumed.

Auction mechanisms are an ideal candidate to provide this missing model.[7] Auction mechanisms are inherently interaction mechanisms between trading agents. The similarity between an auction and a multi-agent system—both involving multiple self-interested individuals and concerning certain global outcomes—makes it easy to map a problem in multi-agent systems to one in auctions. This, together with the effectiveness of auction mechanisms in the real world, has led to various market-based approaches to multi-agent coordination and resource allocation problems in cluster and grid computing environments [Horling and Lesser, 2005; Stöber and Neumann, 2008; Yeo and Buyya, 2006]. These approaches have demonstrated superior performance to those

[6]Therefore a research topic in this scenario is how to design a *strategy-proof* mechanism, with which the optimal strategy for an agent is pre-determined and well-known.

[7]Other approaches to the problem of designing the interaction dynamics include multi-agent learning and biology-inspired paradigms. Multi-agent learning (MAL) [Shoham et al., 2007; Tuyls and Parsons, 2007; Vohra and Wellman, 2007] explores how agents learn and adapt to achieve certain goals. MAL typically takes a game theoretic perspective, modeling the situation as a game and using game theoretic solutions to analyze the interaction of agents with identical or varying learning strategies [Hu and Wellman, 1999, 2003; Littman, 1994, 2001]. Biology-inspired paradigms stem from the coordination among a swarm of homogeneous, little creatures like ants and bees that each follow a simple scheme [Dorigo et al., 1996].

pre-existing, non-market solutions, in terms of a combination of performance, scalability, and reliability. However, the market mechanisms adopted by these approaches are usually selected arbitrarily or based on certain heuristics. It is unknown whether these market mechanisms are optimal solutions, or whether there are better options.

1.3 Auction mechanism design

Facing the challenges in both electronic commerce and market-based control, we need to solve the following problem: *Given a certain set of restrictions and desired outcomes, how can we design a good, if not optimal, auction mechanism; and when the restrictions and goals alter, how can the current mechanism be adjusted to handle the new scenario?*

Traditionally, economists and mathematicians view auctions as games and have successfully applied analytic methods from game theory to some kinds of auctions [Maskin and Riley, 1985], for example the *second-price sealed-bid auctions* [Vickrey, 1961]. The high complexity of the dynamics of some other auction types, especially *double-sided auctions* [Friedman, 1993], however makes it difficult to go further in this direction [Madhavan, 1992; Satterthwaite and Williams, 1993; Walsh et al., 2002].

As a result, researchers turned to experimental approaches. Starting in 1955, Smith [Smith, 1962] ran a series of experimental auctions involving human traders, which revealed many of the properties of double auctions. Later researchers [Cliff and Bruten, 1997; Gode and Sunder, 1993a; Phelps et al., 2003; Rust et al., 1993; Tesauro and Das, 2001; Walsh et al., 2002], especially computer scientists, deployed software agents to study auction mechanisms. With configurable simulated environments, they were able to compare experimental results with theoretical predictions and explore what affects the performance of auction mechanisms. The experimental work to some extent helped to obtain more insights that are not readily available with the theoretical approaches. However, experiments associated with manual investigation were still burdensome. The problems with the approach are exactly those that dog any manual process—it is slow, error-prone, and restricted to just a handful of individuals with the necessary skills and knowledge.

Automated mechanism design (AMD) aims to overcome the problems of the manual process by designing auction mechanisms automatically. AMD considers design to be a search through some space of possible mechanisms. For example, Cliff *et al.* [Cliff, 2001a, 2003] and Phelps *et al.* [Phelps et al., 2002, 2003] explored the use of evolutionary algorithms to optimize different aspects of the continuous double auction. Around the same time, Conitzer and Sandholm [Conitzer and Sandholm, 2003] were examining the complexity of building a mechanism that fitted a particular specification.

These different approaches were all problematic. The algorithms that Conitzer and Sandholm considered dealt with exhaustive search, and naturally the complexity was exponential. In contrast, the approaches that Cliff and Phelps pursued were computationally more appealing, but gave no guarantee of success and were only searching tiny sections of the search space for the mechanisms they considered. As a result, one might consider the work of Cliff and Phelps, and indeed the work that is describe here, to be what Conitzer and Sandholm [Conitzer and Sandholm, 2007] call "incremental" mechanism design, where one starts with an existing mechanism and incrementally alters parts of it, aiming to iterate towards an optimal mechanism. Similar work, though work that uses a different approach to searching the space of possible mechanisms has been carried out by [Vorobeychik et al., 2007] and has been applied to several different mechanism design problems [Schvartzman and Wellman, 2009a].

The problem with taking the automated approach to mechanism design further is how to make it scale—though framing it as an incremental process is a good way to look at it, it does not provide much practical guidance about how to proceed. In addition, all the previous work in this area has one common theme—it all studies single marketplaces or compares different auction mechanisms of multiple marketplaces indirectly. Real marketplaces compete against each other just as traders do in a single marketplace. This dissertation introduces a game that models the competition between marketplaces in a single market, and presents an approach to automated auction mechanism design based on this game as well as a parameterized framework of auction mechanisms that can be searched for a solution to the automated mechanism design problem.

This dissertation is organized as follows. Chapter 2 introduces the basic concepts in auction theory. Chapter 3 gives an overview of experimental approaches to trading agent design and auction mechanism design. Chapter 4 considers a scenario involving competing marketplaces and defines a game that makes it possible to evaluate auction mechanisms at these marketplaces in this scenario. Chapter 5 presents an analysis of entries in the TAC Market Design Tournament, which is based on this game and the insights obtained in the analysis finally lead to Chapter 6, which proposes a novel grey-box approach to automated auction mechanism design and examines its effectiveness. Chapter 7 finally relates this work to prior work in the literature, discusses potential future work beyond this dissertation, and concludes.

The work introduced in this dissertation (Chapter 4 to Chapter 6) was carried out either independently by myself or in collaboration with colleagues, and to the collaborated work covered here, I am the main contributor. To be consistent in this dissertation, regardless of the nature of the different pieces of this work, I will use the first person plural pronoun, '*we*', to refer to the people who carried out the work, and will only use '*I*' when I refer to myself.

The first part of Chapter 4 and the white-box analysis in Chapter 5 was published in the AAMAS conference [Niu et al., 2008b] and the software that we built to run the market design game was selected for demonstration in the conference [Niu et al., 2008c]. The black-box analysis in Chapter 5 covers part of the paper that we published at the IAT conference [Niu et al., 2008a]. The two conference papers were later combined, revised, and further extended, and were published as a AAMAS journal paper [Niu et al., 2010c]. The second part of Chapter 4 was presented in the TADA workshop [Niu et al., 2007a]. Part of Chapter 6 was accepted by the AAMAS conference as an extended abstract [Niu et al., 2010a], and later presented at the AMEC workshop as a regular paper [Niu et al., 2010b]. A much extended version of this paper will appear in the ECRA journal [Niu et al., 2012].

Chapter 2

Auctions

2.1 Auction types

A *market* is a set of arrangements by which buyers and sellers, collectively known as *traders*, are in contact to exchange goods or services in physical or virtual marketplaces.[8] *Auctions* associate marketplaces with strict regulations governing the information available to traders in the marketplaces and the possible actions they can take.

One common kind of auction is the *English auction*, in which there is a single seller, and multiple buyers compete by making increasing bids for the commodity (good or service) being auctioned; the one who offers the highest price wins the right to purchase the commodity. Since only one type of trader—buyers—makes offers in an English auction, the auction belongs to the class of *single-sided auctions*. Another common single-sided auction is the *Dutch auction*, in which the auctioneer initially calls out a high price and then gradually lowers it until one bidder

[8]The term '*market*' in mainstream economics usually refers to a system, structure, or institution where the trading of a certain type of goods or service may happen within or without a geographical or political boundary, for example, the personal computer market in the United States, although it is often used to mean a concrete marketplace colloquially and in finance. One such example is to say '*market clearing*', where the clearing occurs really in a single marketplace to execute transactions.

indicates they will accept that price.

Another class of single-sided auctions is the class of *sealed-bid auctions*, in which all buyers submit a single bid and do so simultaneously, i.e., without observing the bids of the others or if the others have bid. Two common sealed-bid auctions are the *first-price auction* and the *second-price auction* or *Vickrey auction* [Vickrey, 1961]. In both types of sealed-bid auctions, the highest bidder obtains the commodity. In the former, the highest bidder pays the price they bid, while in the latter, they pay the second highest price that was bid.

These four single-sided auctions—English, Dutch, first-price sealed-bid, and Vickrey—are commonly referred to as the *standard auctions* [Klemperer, 1999] and were the basis of much early research on auctions [Rothkopf and Park, 2001; Wolfstetter, 1996].

In addition, there are *double-sided auctions* or DAs,[9] in which both sellers and buyers make offers, or *shouts*. The two most common forms of DA are *clearing houses* or CHs[10] and *continuous double auctions* or CDAs. In a CH, an auctioneer first collects *bids*—shouts from buyers—and *asks*—shouts from sellers, and then clears the market at a price where the quantity of the commodity supplied equals the quantity demanded. This type of market clearing guarantees that if a given trader is involved in a transaction, all traders with more competitive offers are also involved.[11] In a CDA, a trader can make a shout and accept an offer from someone at any time. This design makes a CDA able to process many transactions in a short time, but permits traders with less competitive offers to make deals. Both kinds of DA are of practical importance, with, for example, CDA variants being widely used in real-world stock or trading exchanges including the New York Stock Exchange (NYSE) and the Chicago Mercantile Exchange (CME).

[9] The terminology is not standardized, and sometimes these are called *bid-ask auctions*. Note that [Friedman, 1993] used the term "double auction" to refer to what we call a continuous double auction.

[10] These are sometimes called *call markets* or *static double auctions*.

[11] That is only *intra-marginal* traders are involved in transactions. The concepts of *intra-marginal* and the opposite *extra-marginal* will be introduced in Section 2.2.

In some auctions, traders can place shouts on combinations of items, or "packages", rather than just individual items. They are called *combinatorial auctions* [Cramton et al., 2006]. Combinatorial auctions present a host of new challenges as compared to traditional auctions, including the so-called *winner determination problem* [Lehmann et al., 2006], which is how to efficiently determine the allocation of the items among the traders once the bids have been submitted to the auctioneer.

Traders, in some cases, are allowed to both sell and buy during an auction. Such traders are called *two-way traders*, while those that only buy or only sell are called *one-way traders*.

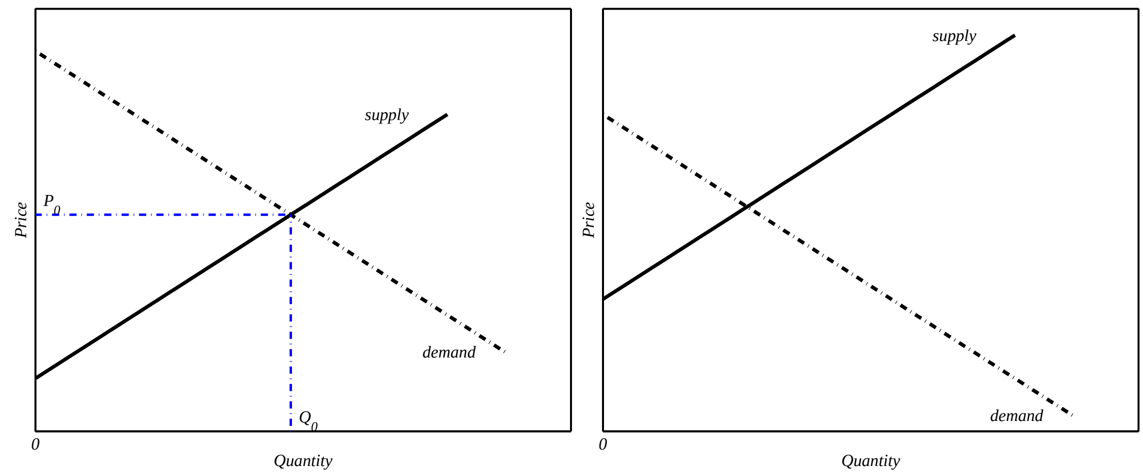
The work that is presented in this dissertation involves only non-combinatorial DAs populated by one-way traders.

2.2 Supply, demand and equilibrium

A central concern in studies of auction mechanisms are the supply and demand schedules in a market. The quantity of a commodity that buyers are prepared to purchase at each possible price is referred to as the *demand*, and the quantity of a commodity that sellers are prepared to sell at each possible price is referred to as the *supply*. Thus if *price* is plotted as a function of *quantity* following the convention in economics,[12] the *demand curve* slopes downward and the *supply curve* slopes upward, as shown in Figure 2.1a, since the greater the price of a commodity, the more sellers are inclined to sell and the fewer buyers are willing to buy.[13] Typically, there is some price at

[12] This graphical representation is usually attributed to Alfred Marshall, and has commonly followed in economics ever since, although it is more intuitive to view *quantity* as a function of *price*, e.g., *quantity* on the vertical axis and *price* on the horizontal axis, particularly so in the idealized scenario discussed in this dissertation where there is a market of some kind of goods and trading in the market is independent from the prices and quantities in other markets. In a real economy, *price* and *quantity* can have a causal relation each way. For example, when the price of some market goes up, more sellers are willing to sell or are able to sell at no loss, which may lead traders from outside of the market to join in for profit, providing additional supply and as a result pulling down the market price.

[13] The straight-line plots in Figure 2.1 are for illustration only. As a matter of fact, supply and demand curves usually have varying slopes at different points, or have a stairwise shape if the goods or service of concern is indivisible.



(a) Underlying (private values) (b) Apparent (offers)

Figure 2.1: Typical supply and demand curves.

which the quantity demanded is equal to the quantity supplied. Graphically, this is the intersection of the supply and demand curves. The price is called the *equilibrium price*, and the corresponding quantity of commodity that is traded is called the *equilibrium quantity*. The equilibrium price and equilibrium quantity are denoted as P_0 and Q_0 respectively in Figure 2.1a.

Each trader in an auction presumably has a limit price, called its *private value*, below which sellers will not sell and above which buyers will not buy. Traders whose private value is no less competitive than the equilibrium price are called *intra-marginal* whereas the rest of the traders are called *extra-marginal*. The supply and demand of intra-marginal traders form the supply and demand curves on the left hand side of the intersection point in Figure 2.1a whereas the supply and demand of extra-marginal traders form the supply and demand curves on the right hand side of the intersection point.

The private values of traders are not publicly known in most practical scenarios. What is known instead are the prices that traders offer. Self-interested sellers will presumably offer higher prices than their private values to make a profit and self-interested buyers tend to offer lower prices than their private values to save money. The prices and quantities that are offered also make a set of supply and demand curves, called the *apparent supply and demand curves*, while the curves based on traders' private values are called the *underlying supply and demand*.[14] Figure 2.1b shows that the apparent supply curve shifts up compared to the underlying supply curve in Figure 2.1a, while the apparent demand curve shifts down. Similar to traders, shouts that form the supply and demand on the left hand side of the intersection point in Figure 2.1a are called intra-marginal and those on the right hand side are called extra-marginal. In a CH, all intra-marginal shouts and only the intra-marginal ones are matched when the market is cleared.

When traders are excessively greedy, the apparent supply and demand curves do not intersect and thus no transactions can be made between sellers and buyers unless they compromise on their profit levels and adjust their offered prices, a nice illustration of which can be found in [Zhan and Friedman, 2007]. Sometimes even the underlying supply and demand curves do not intersect and no transaction is possible unless some traders accept a loss.

As the private values of traders are typically unknown, it is usually assumed in the analysis of a market that these values follow certain models. The simplest model is called the *independent private-value model* [Klemperer, 1999; Parsons et al., 2011], according to which every trader in the market knows the value of the goods being traded, and these values are all private and independent of each other. In the real world, the private values are not fixed and may affect each other. A more realistic model is called the *pure common-value model* [Wilson, 1969], which assumes that the actual value of the goods is the same for everyone, but traders have different private information about what that value actually is. In these cases, a trader will change her estimate of the value if she learns another trader's estimate, in contrast to the independent private-value case in which her value would be unaffected by learning any other trader's preferences or information. A general model encompassing both the independent private-value model and the pure common-value model

[14] Following the terminology in [Cliff and Bruten, 1997].

as special cases is the *correlated-value model* [Milgrom and Weber, 1982]. This model assumes that each trader receives a private information signal, but allows each trader's value to be a general function of *all* the signals.[15]

In addition, in a real economy that comprises multiple markets that each involve a type of goods, the values and prices of goods in one market may be influenced by many factors including, for example, the prices of goods in related markets and government policies.

The work that is presented in this dissertation considers a rather simplified economy that includes a single *market* in which trading may occur in multiple competing *marketplaces*, traders are entitled goods or money exogenously, and their values of goods follow the independent private-value model. Even so, the scenario of multiple competing marketplaces takes one step further than the typical scenarios in experimental economics that involve only a single marketplace.[16]

2.3 A typical time series of shouts

Auction mechanisms usually allow traders to place shouts and modify them over a certain period of time. In a CDA, for instance, buyers and sellers not only 'haggle' on prices in a collective manner, but they also face competition from opponents on the same side of the market. Thus buyers, for example, are not only collectively trying to drive prices down, against the wishes of sellers,

[15] That is, trader i receives signal t_i and would have value $v_i(t_1,\dots,t_n)$ if all traders' signals were available to her. In the independent private-value model, $v_i(t_1,\dots,t_n)$ is a function only of t_i. In the pure common-value model, $v_i(t_1,\dots,t_n) = v_j(t_1,\dots,t_n)$ for all i and j.

[16] Note that the definitions and use of '*market*' and '*marketplace*' vary across fields. For example, in finance, a *stock exchange* is often called a *stock market*, which is virtually a *marketplace* for stock trading, and all the stock exchanges may be collectively called the *global stock market*. For another example, places where mobile apps are purchased are named in many different ways, including, for instance, *Windows Mobile Marketplace*, *Android Market*, and *iPhone AppStore*. It seems to be the case as well in academic publications. In particular, when the market of interest involves only one marketplace, the two concepts are often used interchangeably, or 'marketplace' is not used at all, as in the previous work that will be reviewed in Chapter 3. To avoid ambiguity in this dissertation, when multiple marketplaces are involved, 'marketplace' is used to refer to one of the multiple places where trading may occur, and only the whole system is called a 'market', however the use of 'stock market' and the like in finance will be followed as long as there is no ambiguity.

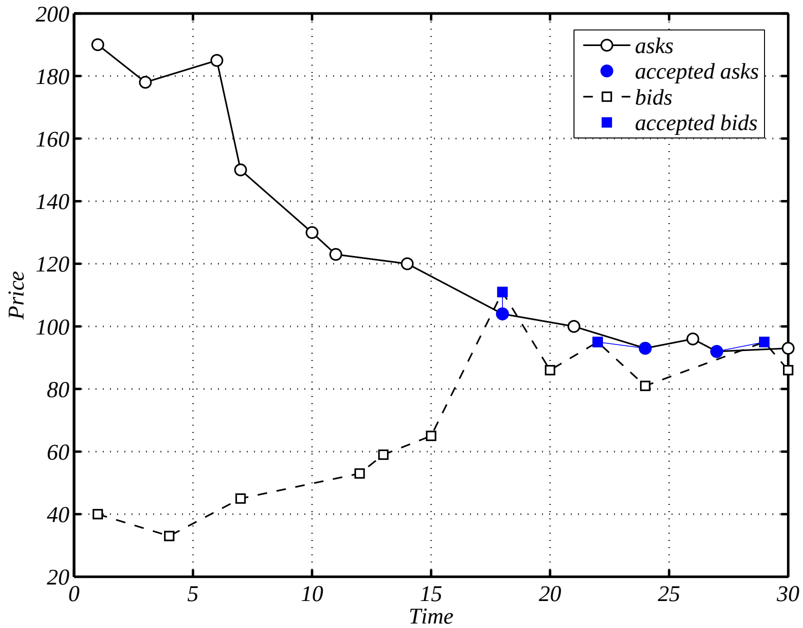


Figure 2.2: A typical time series of asks and bids.

but they are also individually trying to ensure that they, rather than other buyers, make profitable trades. This leads to shouts becoming more and more competitive over time in a given auction. Figure 2.2 shows a typical time series of shouts in a DA. Ask prices usually start high while bid prices start low. Gradually, traders adjust their offered prices, or make new shouts, closing the gap between standing asks and bids until the price of a bid surpasses that of an ask. Such an overlap results in a transaction, shown as solid bars between the matched asks and bids in Figure 2.2.

In the auction depicted in Figure 2.2, newly placed bids (asks) do not have to beat the outstanding bids (asks). However in some variants of the CDA, including the mechanism used by the NYSE, new shouts must improve on existing ones. This requirement is commonly referred to as the *NYSE shout improvement rule* [Easley and Ledyard, 1993].

In some real-world stock markets, including the NYSE and the NASDAQ markets, trades are made through *specialists* or *market makers*, who buy or sell stock from their own inventory to keep the market liquid or to prevent rapid price changes.[17] Each specialist is required to publish on a regular and continuous basis both a *bid quote*, the highest price it will pay a trader to purchase securities, and an *ask quote*, the lowest price it will accept from a trader to sell securities. The specialist is obligated to stand ready to buy at the bid quote or sell at the ask quote up to a certain number of shares. The range between the bid quote and the ask quote is called the *bid-ask spread* (the bid quote is lower than the ask quote), and according to stock exchange regulations, the bid-ask spread must be suitably small. If buy orders temporarily outpace sell orders, or conversely if sell orders outpace buy orders, the specialist is required to use its own capital to minimize the imbalance. This is done by buying or selling against the trend of the market until a price is reached at which public supply and demand are once again in balance. Maintaining a bid-ask spread creates risk for a specialist, but when well maintained, also brings huge profits, especially in an active market [Bao, 2001].

Markets involving specialists that present quotes are called *quote-driven markets*. Another class of markets are *order-driven markets*, in which all of the orders of buyers and sellers are displayed. This contrasts with quote-driven markets where only the orders of market makers are shown. An example of an order-driven market is the market formed by *electronic communication networks* or ECNs. These are electronic systems connecting individual traders so that they can trade directly between themselves without having to go through a middleman like a market maker. The biggest advantage of this market type is its transparency. The drawback is that in an order-driven market, there is no guarantee of order execution, meaning that a trader has no guarantee of making a trade at a given price, while it is guaranteed in a quote-driven market. There are markets that combine attributes from quote- and order-driven markets to form hybrid systems.

Our discussion above may give the impression that in real markets, trade orders are made

[17] Traditionally, in the NYSE, a given stock is traded through a single specialist, and in the NASDAQ, a stock may be dealt with by multiple competing market makers. As the NYSE has gradually adopted electronic trading in recent years, less and less stock has been traded through specialists.

directly by the individuals who want to buy or sell stock. In practice, traders commonly place orders through brokerage firms, which then manage the process of executing the orders through a stock exchange.[18]

2.4 Performance metrics

Auctions with different rules and populated by different sets of traders may vary greatly in performance. Popular performance measurements include, but are not limited to, *allocative efficiency* and the *coefficient of convergence*. These are the measures that will be used in this dissertation.

2.4.1 Allocative efficiency

The allocative efficiency of an auction, denoted as E_a, is used to measure how much social welfare is obtained through the auction. The *theoretical* or *equilibrium profit*, P_e, of an auction is

$$P_e = \sum_i |v_i - p_0| \tag{2.1}$$

for all intra-marginal traders, where p_0 is the equilibrium price and v_i is the private value of trader i who can trade at p_0 without a loss. The *actual overall profit*, P_a, of an auction is

$$P_a = \sum_j |v_j - p_j| \tag{2.2}$$

[18]`http://www.sec.gov/investor/pubs/tradexec.htm` gives a detailed illustration of how a trade order is executed through a brokerage firm.

where p_j is the transaction price of a trade completed by trader j and v_j is the private value of trader j, where j ranges over all traders who trade. Given these

$$E_a = \frac{100P_a}{P_e} \tag{2.3}$$

E_a is thus a measure of the proportion of the theoretical profit that is achieved in practice.

2.4.2 Convergence coefficient

The convergence coefficient, denoted as α, was introduced by Smith [Smith, 1962] to measure how far an active auction is away from the equilibrium point. It actually measures the relative root mean squared (RMS) deviation of transaction prices from the equilibrium price

$$\alpha = \frac{100}{p_0}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - p_0)^2} \tag{2.4}$$

Since auctions with human traders often trade close to the equilibrium price, α is used as a way of telling how closely artificial traders approach human trading performance [Cliff and Bruten, 1997].

Chapter 3

Experimental approaches

Traditionally, economists and mathematicians view auctions as games and have successfully applied analytic methods from game theory to some kinds of auctions [Maskin and Riley, 1985], for instance Vickrey auctions [Vickrey, 1961]. However, as, for example Friedman [Friedman, 1993], has pointed out, DAs, particularly CDAs, are too complex to analyze in this way since at every moment, a trader must compute expected utility-maximizing shouts based on the history of shouts and transactions and the time remaining in the auction. This difficulty led researchers to seek experimental approaches.

Researchers from economics have tended to run laboratory experiments with human subjects, while computer scientists appeal to computer-based market simulations and use software agents to automate trading. This chapter reviews the literature, focusing on how more and more sophisticated trading strategies evolved, and how trading strategies and auction mechanisms were designed or optimized in an automated fashion.

3.1 Smith's experiments

Researchers need certain market data to conduct analysis. As in other disciplines, researchers in economics have long based their studies on field data, from large-scale on-going markets [Schwartz et al., 2006b]. Field data has the most relevance to the real-world economy, but does not reveal some important information, e.g., the private values of traders, and hence puts limits on what can be achieved.

Smith [Smith, 1962] pioneered the research falling into the field of *experimental economics* by running a series of experiments with human subjects. The human subjects in laboratory experiments presumably inherit the same level of intelligence and incentive to make a profit as in real markets, and the experiments are a series of CDA simulations that adopt mechanisms that are similar to those in major stock and commodity exchanges and are described as follows:

- Every trader in the CDA markets is given a private value. The set of private values form the underlying supply and demand curves.
- Each experiment was run over a sequence of *trading days*, or *periods*,[19] the length of which depends upon how many traders are involved but are typically several minutes in duration. Different experiments may have different numbers of days.
- For simplicity, in most experiments, a trader is allowed to make a transaction for the exchange of only a single commodity in each day.
- Traders are free at any time to make a bid/ask or to accept a bid/ask.
- Once a transaction occurs, the transaction price, as well as the two traders' private values, are recorded.

[19]Smith used the term *periods* to refer to what are called *days* in this dissertation.

- For each new day, a trader may make up to one transaction with the same private value as before no matter whether she has made one in the previous day. Thus the supply and demand curves each correspond to a single trading day. The experimental conditions of supply and demand are held constant over several successive trading days in order to give any equilibrating mechanisms an opportunity to establish an equilibrium over time, unless it is the aim to study the effect of changing conditions on market behavior.

Smith's experiments effectively revealed many of the properties of CDAs. Of particular interest in this dissertation, he showed that in many different cases even a handful of traders can lead to high allocative efficiency, and transaction prices can quickly converge to the theoretical equilibrium. Smith's experimental framework has been widely adopted by later studies and these properties have been the basis and benchmark for much subsequent work.

3.2 Trading agents for double auctions

Experiments with human subjects are expensive in terms of time[20] and money[21] needed. Computer-aided simulation is a less expensive alternative and can be repeated as many times as needed. However traders' strategies in these simulations are not endogenously chosen as in auctions with human traders, but are specified exogenously by the experiment designers, which raises the question of whether the conclusions of this approach are trustworthy and applicable to practical situations.

Gode and Sunder [Gode and Sunder, 1993a] invented a naïve trading strategy that always randomly picks a profitable price to bid or ask. Surprisingly, their experiments with CDAs exhibited high efficiency despite the lack of intelligence of the traders. Indeed, software agents, armed with various learning algorithms and optimization techniques, have been shown capable of producing

[20] The experiments are run using a physical clock and need take into consideration the response time of human traders.

[21] Usually human subjects are monetarily rewarded according to their performance.

outcomes similar to those obtained with human subjects [Cliff and Bruten, 1997; Gode and Sunder, 1993a] and generating higher individual profits [Das et al., 2001].

3.2.1 Trading strategies

This section enumerates the common trading strategies for double auctions in the literature.

Zero intelligence

The main focus of Smith's pioneering experiments was on the convergence of transaction prices in different scenarios rather than examining why high efficiency is obtained. In a computerized world, a question that arises naturally is whether these results can be replicated in electronic auctions. In Smith's experiments, as in real markets traditionally, the traders are human beings, but computer programs are supposed to be automatic and work without human involvement. Obviously humans are intelligent creatures, but programs are not, at least for the foreseeable future. Is it intelligence that contributes to the high efficiency, or something else?

Gode and Sunder [Gode and Sunder, 1993a,b] were among the first to address this question, claiming that no intelligence is necessary for the goal of achieving high efficiency; so the outcome is due to the auction mechanism itself.

They reached this position having introduced two trading strategies: *zero intelligence without constraint* or ZI-U and *zero intelligence with constraint* or ZI-C. ZI-U, the more naïve version, shouts an offer at a random price without considering whether it is losing money or not, while ZI-C, which lacks the motivation of maximizing profit and picks a price in a similar way to ZI-U, simply makes shouts that guarantee no loss.

It was shown that ZI-U performed poorly in terms of making a profit, but ZI-C generated high efficiency solutions so that markets populated by ZI-C traders exhibited efficiency that compared well with that of markets populated by human traders and the efficiency obtained can be considered to be a lower bound on the efficiency of markets [Gode and Sunder, 1993b].

Gode and Sunder's experiments were set up with similar rules as in Smith's. They designed five different supply and demand schedules and tested each of them respectively with the three kinds of homogeneous traders: ZI-U, ZI-C, and human traders.

Prices in the ZI-U market exhibited little systemic pattern and no tendency to converge toward any specific level, but on the contrary, prices in the human market, after some initial adjustments, settled in the proximity of the equilibrium price. Gode and Sunder then raised the question: how much of the difference between the market outcomes with ZI-U traders and those with human traders is attributable to intelligence and profit motivation, and how much is attributable to market discipline?

They argued that, after examining the performance of the ZI-C markets, it was market discipline that played a major role in achieving high efficiency. Though in the ZI-C market, the price series showed no signs of improving from day to day, and the volatility of the price series was greater than the volatility of the price series from the human market, the series converged slowly toward equilibrium within each day. Gode and Sunder's explanation was that it was due to the progressive narrowing of the opportunity sets of ZI-C traders, e.g., the set of intra-marginal traders. Despite the randomness of ZI-C, buyers with higher private values tend to generate higher offered prices and they are likely to trade with sellers earlier than those buyers further down the demand curve. A similar statement also holds for sellers. Thus as the auction goes on, the upper end of the demand curve shifts down and the lower end of the supply curve moves up, which means the feasible range of transaction prices narrows as more commodities are traded, and transaction prices will converge to the equilibrium price. The fact that ZI-C traders lack profit motivation and have only the minimal intelligence (just enough to avoid losing money) suggested that the market mechanism was the key to obtaining high efficiency.

Zero intelligence plus

Gode and Sunder's results were, however, questioned by Cliff and Bruten [Cliff and Bruten, 1997]. Cliff and Bruten agreed on the point that the market mechanism played a major role in achieving high efficiency, but disputed whether in ZI-C markets transaction prices will always converge on an equilibrium price. They argued that the mean or expected value of the transaction price distribution was shown quantitatively to get close to the equilibrium price only in situations where the magnitude of the gradient of linear supply and demand curves was roughly equal, and used this to infer that zero-intelligence traders are not sufficient to account for convergence to equilibrium.

Cliff and Bruten further designed an adaptive trading strategy called *zero intelligence plus* or ZIP. Like ZI-C, ZIP traders make stochastic bids, but can adjust their prices based on the auction history, i.e., raising or lowering their profit margins dynamically according to the actions of other traders in the market. More specifically, ZIP traders raise the profit margin when a less competitive offer from the competition[22] is accepted, and lower the profit margin when a more competitive offer from the competition is rejected, or an accepted offer from the other side of the market would have been rejected by the subject. At every step, the profit margin is updated according to a learning algorithm called the *Widrow-Hoff delta rule* [Widrow and Hoff, 1960] in which the value being learned is adapted gradually towards a moving target, and the past targets leave discounting momentum to some extent.

Cliff and Bruten concluded that the performance of ZIP traders in the experimental markets was significantly closer to that of human traders than was the performance of ZI-C traders, based on the observation that ZIP traders rapidly adapted to give profit dispersion[23] levels that were in

[22] That is, sellers compete against sellers to get asks accepted and buyers compete against buyers to get bids accepted.

[23] Profit dispersion is the root mean squared difference between actual and equilibrium profits, and can be expressed

some cases approximately a factor of ten less than those of ZI-C traders.

Preist and van Tol [Preist and van Tol, 1998] introduced a revised version of ZIP, which uses simpler rules than those in ZIP and we call PVT. Their experiments incorporated *consistent shouts*, i.e., shouts that will continue to exist until they are matched or the period of trading ends, while the experiments by Cliff and Bruten allowed at most one shout to be active at any moment and the active shout will be disregarded if it is not matched immediately. With PVT and the adoption of persistent shouts, Preist and van Tol reported faster convergence to equilibrium and robustness to changes in parameter configuration .

Roth and Erev

Other learning methods have been adopted to design even more complex trading strategies than ZIP and its variants. Roth and Erev [Roth and Erev, 1995] proposed a stimuli-response strategy, which we call RE. The RE strategy uses reinforcement learning to choose from n possible profit margins over the agent's private value based on a reward signal computed as a function of profits earned in the previous round of bidding. It totally eliminates the dependence upon the information about transactions and shouts in ZIP and therefore presents a universal solution to trading in all kinds of auctions.

Gjerstad and Dickhaut

Taking a different path from RE and trying to make better use of information than ZIP, Gjerstad and Dickhaut [Gjerstad and Dickhaut, 1998] suggested a best-response-based strategy, which is

---

as

$$\sqrt{\frac{1}{n}\Sigma_i(a_i-\pi_i)^2},$$

where a_i, i.e., $|v_j - p_j|$ as in (2.2) on Page 19), and π_i, i.e., $|v_i - p_0|$ as in (2.1) on Page 19, are the *actual* and *theoretical equilibrium profits* of trader i, $i = 1, \cdots, n$.

commonly referred to as GD. GD traders keep a sliding window of the history of the shouts and transactions and calculate the probabilities of their offers being accepted at different prices. The traders use a cubic interpolation on the shouts and transaction prices in the sliding window in order to compute the probability of future shouts being accepted. They then use this to calculate the expected profit of those shouts. The expected profit at a price is the product of the probability of the price being accepted and the difference between the price and the private value. GD traders then always choose to bid or ask at a price that maximizes their expected profit. GD is much more computation-intensive than those strategies above, and generated the best record both for allocative efficiency and the speed of convergence to equilibrium.

In contrast to maximizing expected immediate profits as GD does, Tesauro and Bredin [Tesauro and Bredin, 2002] considered maximizing the discounted cumulative profits of individual traders and proposed a strategy called GDX that extends GD with an additional dynamic programming (DP) component to do this. The DP component in GDX considers iterative steps along the combination of two dimensions, the number of units of commodities that the trader has yet to trade and the number of trading opportunities that the trader has until the end of the auction. At each moment, GDX solves the DP model based on its expected returns at future steps and chooses the action—either to shout at a certain price or not to shout at all—that maximizes the long-term return. They showed through experiments that GDX was increasingly capable of beating GD and ZIP in terms of the discounted long-term return when future profits were weighted higher.

Risk-Based and Adaptive-Aggressiveness

Following the framework of ZIP, Vytelingum *et al.* [Vytelingum et al., 2004] introduced a *risk-based* trading strategy, or in short RB, for CDAs. RB adapts its shout price towards a moving target price, just as ZIP does, but calculates the target price in a more sophisticated way, based on a *risk* *factor*, r, and an estimated theoretical equilibrium price, p^*. With $r \in [-1,1]$, the target price is calculated through an interpolation function, satisfying the condition that for an intra-marginal buyer—whose private value is higher than or equal to p^*—the target price equals p^* when $r = 0$ (risk-neutral), equals 0 when $r = 1$ (risk-seeking), and equals the trader's private value when $r = -1$ (risk-averse), and for an extra-marginal buyer—whose private value is lower than p^*—the target price equals 0 when $r = 1$ and the trader's private value when $r \leq 0$. The target price is determined in a symmetric way for a seller. The risk factor starts with some initial value and is adjusted to go up or down according to transaction and shout prices in the market, akin to the way the profit margin is adjusted in ZIP.[24] As transaction prices are expected to converge to the equilibrium price, the average of transaction prices within a sliding window is used for p^*.[25] The interpolation function also relies upon a fixed parameter, θ, which controls the gradient of the curve of the function. Vytelingum *et al.* showed that CDAs populated by RB traders obtain allocative efficiency comparable to those populated by ZIP traders, and RB traders make more profit than both ZI-C and ZIP when all three types of trader exist in a single market.

A more advanced version of RB was introduced by the same authors in 2008 and called a strategy of *adaptive aggressiveness*, or AA [Vytelingum et al., 2008a]. AA allows θ to change dynamically and r to be updated at a varying rate in response to different degrees of volatility of market prices. Experimental results showed that AA outperforms ZIP and GDX in CDAs with either homogeneous or heterogeneous populations.

[24] The risk factor, r, can be interpreted as a normalized profit margin. The term 'risk' here is somewhat misleading. It is awkward to say that an agent changes its attitude towards risk, as r changes, in such a microscopic context as within an auction. Indeed in AA, which is described immediately below, the modified version of RB, r is said to stand for the degree of *aggressiveness* of traders.

[25] The same idea was adopted in auction rules as well to clear the market at the estimated equilibrium price (see PN in Section 6.2.5) and to avoid shouts that have little chance to get matched (see AE in Section 6.2.3).

3.2.2 Interaction of heterogeneous trading strategies

Early empirical work on DAs and trading strategies, including that by Smith [Smith, 1962], Gode and Sunder [Gode and Sunder, 1993a], and Cliff and Bruten [Cliff and Bruten, 1997], has employed either human traders or homogeneous trading agents, demonstrating high efficiency and fast convergence to equilibrium. From the viewpoint of individual traders, however, a major goal is to maximize their own profit. It is no surprise that the authors of GDX, RB, and AA ran auctions that were populated by heterogeneous trading agents to examine the relative superiority of their strategies.

There are both theoretical and practical reasons for considering heterogeneous traders. As Rust *et al.* argued in [Rust et al., 1993]:

> Although current theories of DA markets have provided important insight into the nature of trading strategies and price formation, it is fair to say that none of them has provided a satisfactory resolution of "Hayek's problem".[26] In particular, current theories assume a substantial degree of implicit coordination by requiring that traders have common knowledge of each other's strategies (in game-theoretic models), or by assuming that all traders use the same strategy (in learning models). Little is known theoretically about price formation in DA markets populated by heterogeneous traders with limited knowledge of their opponents.
>
> . . . the assumption that players have common knowledge of each other's beliefs and strategies . . . presumes an unreasonably high degree of implicit coordination amongst the traders ... Game theory also assumes that there is no *a priori* bound on traders' ability to compute their BNE[27] strategies. However, even traders with infinite, costless

[26] The Hayek's problem is how the trading process aggregates traders' dispersed information, driving the market towards competitive equilibrium.

[27] This is the abbreviation of *Bayesian Nash equilibrium*. In game theory, *players* interact with each other in a

> computing capabilities may still decide to deviate from their BNE strategies if they believe that limitations of other traders force them to use a sub-optimal strategy.

They went on to argue that ZI-C and other strategies' striking performance strongly suggests that the nice properties have more to do with the market mechanism itself than the rationality of traders. In addition, strategies that are more individually rational than ZI-C may display less collective rationality since clever strategies can exploit unsophisticated ones such as the truth-telling strategy, or TT,[28] and ZI-C so that a more-intelligent extra-marginal trader has more chances to finagle a transaction with an intra-marginal trader, causing market efficiency to fall.

To observe heterogeneous auctions, the Santa Fe Double Auction Tournament (SFDAT) was held in 1990 and prizes were offered to entrants in proportion to the trading profits earned by their programs over the course of the tournament. Thirty programs from researchers in various fields and industry participated. According to [Rust et al., 1993], the majority of the programs encoded the entrant's "market intuition" using simple rules of thumb. The top-ranked program was KAPLAN, named after the entrant. KAPLAN and the runner-up strategy were remarkably similar. Both "wait in the background and let the others do the negotiating, but when bid and ask get sufficiently close, jump in and steal the deal".

The overall efficiency levels in the markets used in the tournaments originally appeared to be somewhat lower than that observed in experimental markets with human traders (around 93% versus, for example, 99% or higher in most of the experiments with human traders in [Gode and Sunder, 1993a]), but experiments without the last-placed players produced an efficiency of around

game. Each player is associated with a set of actions and chooses among these actions to maximize its utility. A rational player takes the action that is the best response to the opponents' actions. When every player's action is a best response to the actions of the rest players, the joint action forms a *Nash equilibrium*. When a player has no complete information about its opponents, it maintains a belief about the characteristics of these opponents in the form of, for example, probability distributions, updates the belief as it interacts with the opponents using *Bayes' theorem*, and always chooses the action with the highest expected utility. When all the players maintain their own beliefs in such a manner, and take actions that are best responses to each other, the joint action forms a *Bayes Nash equilibrium*.

[28] A TT trader honestly offers to trade at its private value.

97%. This is further evidence that the properties of traders also affect the outcome of DA markets to some extent.

Besides high efficiency levels and convergence to competitive equilibrium, other "stylized facts" of human DA markets observed in the SFDAT included: reductions in transaction-price volatility and efficiency losses in successive trading days that seem to reflect apparent learning effects, coexistence of extra-marginal and intra-marginal efficiency losses, and low-rank correlations between the *realized order of transactions* and the *efficient order*.[29]

Rust *et al.* reported after thorough examination of efficiency losses in the tournaments and post-tournament experiments that the success of KAPLAN was due to its patience in waiting to exploit the intelligence or stupidity of other trading strategies.[30]

The volume of e-commerce nowadays creates another motivation for evaluating trading strategies in a heterogeneous environment. Electronic agents, on behalf of their human owners, can automatically make strategic decisions and respond quickly to the changes in various kinds of markets. In the foreseeable future, these agents will have to compete with a variety of agents using a range of trading strategies and human traders. As more complex trading strategies appear, it is natural to speculate on how these electronic minds will compete against their human counterparts.

Das *et al.* [Das et al., 2001] ran a series of CDAs allowing persistent orders[31] populated by a mixed population of automated agents (using modified GD and ZIP strategies) and human traders. They found that though the efficiency of the CDAs was comparable with prior research, the agents outperformed the humans in all the experiments, obtaining about 20% more profit. Das *et al.*

[29] The *efficient order* is the transaction sequence that maximizes surplus, meaning that the first transaction occurs between the buyer with the highest private value and the seller with the lowest private value, the second transaction occurs between the buyer and seller next to them, and so on. The *realized order of transactions* is the actual order in which transactions are made.

[30] The usual higher efficiency of CHs than CDAs can also be viewed as the proactive elimination of the effect of traders' impatience.

[31] In the SFDAT and the CDA testing ZIP in [Cliff and Bruten, 1997], shouts that are outbid are removed from the market, which is however not typical of real marketplaces.

speculated that this was due to human errors or weakness, and human traders were observed to improve their performance as they got familiar with using the trading software. Das *et al.* also suggested that the weaknesses of trading agents may be found when human experts take them on and thus improvement can be made to the algorithms of the agents.

Tesauro and Das [Tesauro and Das, 2001] executed experiments with both homogeneous and heterogeneous trading agents with varying trader population composition, making it possible to gain more insights into the relative competitiveness of trading strategies. In either the so-called "one-in-many"[32] tests or "balanced-group"[33] tests, GD and ZIP (and their variants) exhibited superior performance over ZI-C and KAPLAN even when the market mechanisms vary to some extent.[34] Furthermore, MGD, a variant of GD due to Das *et al.* [Das et al., 2001], outperformed all the other strategies. The same configurations of one-in-many and balanced-group tests were also employed in comparing RB with ZI and ZIP by Vytelingum *et al.* [Vytelingum et al., 2004].

The above approaches nevertheless all employed a fixed competition environment. In practice, when a strategy dominates others, it tends to flourish and be adopted by more people. Rust *et al.* were among the first to conduct evolutionary experiments, where the relative numbers of the different trading strategies in a SFDAT market changed over time, so that more profitable strategies became more numerous than less profitable ones. Such an analysis revealed that although KAPLAN agents outperformed others when traders of different types were approximately evenly distributed, they later exhibited low overall efficiency as they became the majority, making the evolution process a cycle of ups and downs.

Walsh *et al.* [Walsh et al., 2002] gave a more formal analysis combining the game-theoretic

[32]A single agent of one type competes against an otherwise homogeneous population of a different type.

[33]Buyers and sellers are evenly split between two types, and every agent of one type has a counterpart of the other type with identical limit prices.

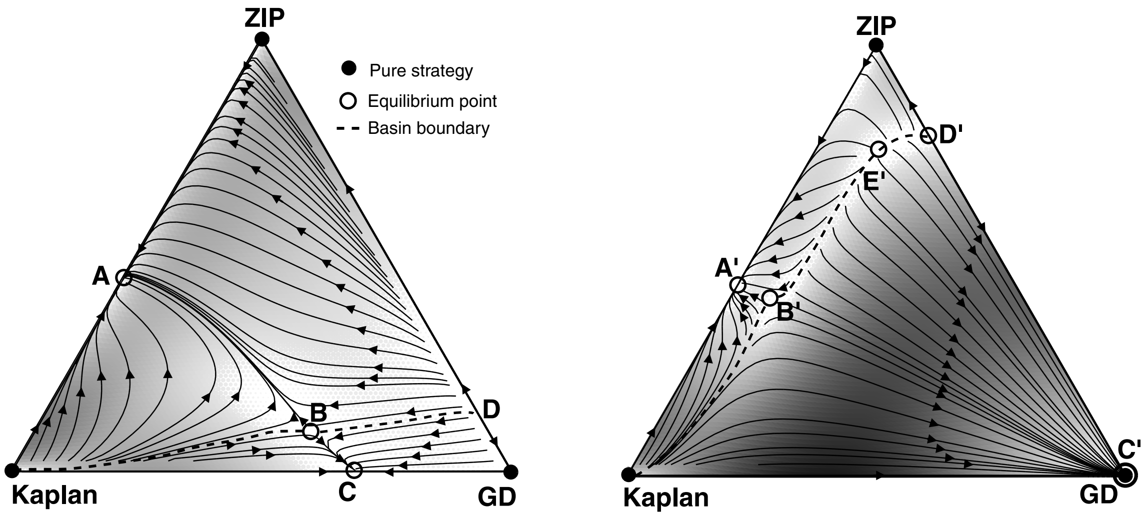
[34][Tesauro and Das, 2001] tested both with and without the NYSE shout improvement rule, a standing shout queue, and allowance of shout modification.

solution concept of Nash equilibrium, or NE, and *replicator dynamics*. This analysis is commonly referred to as *evolutionary game-theoretic analysis*, or EGTA. In EGTA, heuristic strategies, rather than the atomic actions like a bid or ask, are treated as primitive, and expected payoffs of each individual strategy are computed at certain points of the joint heuristic strategy space.[35] This method reduces the model of the game from a potentially very complex, multi-stage game to a one-shot game in normal form. At points where one strategy gains more than others, *replicator dynamics* dictates that the whole population moves to a nearby point where the winning strategy takes a larger fraction of the population. This process continues until an equilibrium point is reached where all strategies are equally competitive in terms of their expected payoffs. There may be multiple equilibrium points[36] 'absorbing' areas of different sizes, *basins* of the equilibria, which together compose the whole strategy space.

For example, Figure 3.1a, taken from [Walsh et al., 2002], shows the replicator dynamics of a CDA market with three strategies. A, B, C, and D are all equilibrium points, but B and D are not stable since a small deviation from them will lead to one of the other equilibria. The replicator dynamics gives an overview of the interaction of the three strategies and their relative competitiveness. The arrows show the directions in which the whole population moves at different points, and the shading indicates how much the strategies differ in terms of their expected payoffs, which may tell how fast the population moves towards the equilibrium points. What's more, a technique called *perturbation analysis* can be used to evaluate the potential to improve on a strategy. Figure 3.1b, also from [Walsh et al., 2002], shows the replicator dynamics of the same strategies after small portions of both ZIP and KAPLAN's payoffs are shifted to GD. This shift significantly changes the landscape of the space, and GD dominates in most of possible combinations. This shows that a 'tiny' (in the words of [Walsh et al., 2002]) improvement in the GD strategy may greatly affect its

[35] That is a space of a mixture of strategies when their relative proportions vary.

[36] Each equilibrium point also represents a mixed strategy, a homogeneous population of which makes a NE.



(a) before perturbation

(b) after perturbation

Figure 3.1: The replicator dynamics of CDA with ZIP, KAPLAN, and GD. Originally Figure 2 in [Walsh et al., 2002].

competitiveness against the other strategies.

Phelps *et al.* [Phelps et al., 2005, 2006] took a similar approach in comparing the RE, TT, and GD strategies, showed the potential of RE, and demonstrated that a modified RE strategy could be evolved by optimizing its learning component.

The main drawback of this approach is an exponential dependence on the number of strategies, which limits its applicability to real-world domains where there are potentially many heuristic strategies. Walsh *et al.* [Walsh et al., 2003] proposed information theoretic approaches to deliberately choose the sample points in the strategy space through an interleaving of equilibrium calculations and payoff refinement, thus reducing the number of samples required.

3.2.3 Automated strategy acquisition

Designing heuristic strategies to a great extent depends on the intelligence and experience of the strategy designer. Prior studies have also demonstrated that the performance of heuristic strategies hinges on the selection of parameter values. It is preferable that automatic optimization is used to find best parameter combinations and further identify better strategies automatically. Pioneering work [Cliff, 2001b; Phelps, 2007] along this avenue adopted evolutionary computation to address the challenge.

Evolutionary computation

A *genetic algorithm* (or GA) is a search technique used in computing to find true or approximate solutions to optimization and search problems. Genetic algorithms are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as *inheritance*, *mutation*, *selection*, and *crossover* [Forrest, 1993; Goldberg, 1989; Holland, 1975].

A typical GA maintains a set of *individuals* of some kind, each as a candidate solution to some problem of interest. These individuals, can be *selected* and *matched* to *reproduce* generation by generation just as a population of biological individuals do in the natural world. The chance of an individual getting selected for reproduction is based on its *fitness*, which measures the quality of the solution to the problem. Basically, the fitter an individual is, the better chance it will be selected.

Each individual in a GA is expressed as an instance of the *genetic representation* of the solution domain. The representation is also called the *genotype* or *chromosome* of the solution species. A standard representation of solution is as an array of bits. Arrays of other types, trees, or other kinds of structures can be used in essentially the same way.

The fitness of an individual is determined by a *fitness function*, which is defined over the genetic representation of a solution and measures the quality of the solution. The fitness function is always problem dependent. For instance, in the knapsack problem we want to maximize the total value of objects that can be put in a knapsack of some fixed capacity. Assuming that there are n objects to fill the knapsack of capacity S and item i has a value of v_i and a size of s_i, a solution to this problem might be represented by an array of n bits, where the ith bit determines whether or not to put object i into the knapsack with 1 for *yes* and 0 for *no*. Not every such representation is valid, as the size of objects may exceed the capacity of the knapsack. The fitness of the solution can then be defined as the sum of values of all objects in the knapsack if the representation is valid, or 0 otherwise. That is, given an individual, $b_{n-1} \dots b_1 b_0$, the fitness of the individual is

$$F(b_{n-1} \dots b_1 b_0) = \begin{cases} \sum_{i=0}^{n-1} (v_i b_i) & \text{if } \sum_{i=0}^{n-1} (s_i b_i) \leq S \\ 0 & \text{otherwise.} \end{cases}$$

For some problems, it is hard or even impossible to define the fitness expression, and external components like human interactions may be used to evaluate solutions, as in interactive genetic algorithms. For some other problems, there are no absolute metrics of fitness that exist and the fitness function needs to take more than one solution as input to compute a relative fitness of one solution against the others, e.g., in searching for better strategies for some multi-player game where no dominating strategy exists.

After the individuals in the population are evaluated, their fitnesses are used to select individuals with high fitness for reproduction to construct the next generation of individuals. The selection and reproduction steps require different types of *operators*, typically including *selection operators*, *mutation operators*, and *crossover operators*. A selection operator controls how to select individuals from the previous generation. One example is to select individuals probabilistically, in proportional to their fitnesses. A mutation operator governs how to tweak a single, selected individual to generate a usually slightly different individual so as to have a random search around the parent individual in the solution space. When an array of bits is used as the representation, the most common mutation operator is to flip the bits in the array independently with a relatively low probability, i.e., what is called the *mutation rate*. A crossover operator supports multiple, usually two, individuals to switch part of their genetic materials to construct offsprings. For instance, when the individuals are represented by a fixed-length array of bits, a crossover operator may switch bits between two parent individuals that are located in the same segment in the two arrays. These operators can be designed and organized in many different ways to form a pipeline to produce offsprings.

In a GA, the initial population of individuals is typically generated randomly and evolves generation by generation after iteratively applying the operators. The fitnesses of individuals basically improve across generations, at least in part, due to the random exploration at the beginning, the biased search in selection, and the combination of partial solutions in crossover [Forrest, 1996].

The introduction of GAs here is limited to the simple, classic model of GA in many aspects, but this model is exactly what was used in the early work on automated acquisition of trading strategies and auction mechanisms.

Optimizing parameter combination in ZIP

Cliff addressed the labor-intensive manual parameter optimization for the ZIP strategy, automatically optimizing parameter selection using a GA [Cliff, 2001b]. He identified eight parameters in ZIP: lower and upper bounds of the learning rate β (how fast to move towards the target), momentum γ (how much past momentum[37] to carry over), initial profit margin μ, and the upper bounds of the ranges defining the distributions of absolute and relative perturbations on learned prices,

[37]The momentum is a a discounted sum of distances to the targets in the past.

respectively denoted as c_a and c_r. These real parameters make an eight-dimensional space and any parameter value combination corresponds to a point in that space. The vector of the eight parameters defines an ideal genotype. To evaluate a combination of parameter values for ZIP, or more correctly, a ZIP variant, Cliff ran a CDA market populated by 22 homogeneous trading agents that all adopted the same ZIP variant, and used a weighted sum of daily coefficient of convergence[38] of the market as the fitness of this ZIP variant. He showed that this GA-based approach was effective in optimizing the parameter space of the ZIP strategy.

Combining GA and EGTA

Phelps *et al.* [Phelps et al., 2005, 2006] took this track a step further and focused on how to optimize trading strategies to maximize profits of individual traders. They combined the EGTA method and a GA, identified a strategy as the basis for optimization, and successfully evolved the strategy and acquired an optimized strategy that can beat GD, commonly considered then one of the most competitive strategies.

Since it is not realistic to seek "best", or even "good", strategies that can beat all potential opponents because an absolutely dominating strategy does not appear to exist in the CDA trading scenario—since the performance of a strategy depends greatly on the types of the opponents—Phelps *et al.* proposed using a small finite population of randomly sampled strategies to approximate the game with an infinite strategy population consisting of a mixture of all possible strategies. In particular, RE, TT, and GD were chosen as sample strategies. Following the EGTA and perturbation methods in [Walsh et al., 2002], RE was found to have the potential to dominate TT and GD.

As described above, RE traders adapt their trading behavior by learning from their profits in

[38] The α we defined in Section 2.4.2.

successive auction rounds. Potentially, the RE learning algorithm may be replaced by a number of learning algorithms, including SQ (stateless Q-learning), NPT (a modified version of RE used in [Nicolaisen et al., 2001]), and DR (a control algorithm which selects a uniformly random action regardless of reward signal). Phelps *et al.* then encoded the genotype to select any of these algorithms together with their parameters. The evolutionary search procedure they used is similar to Cliff's except that the individuals in a generation are evaluated again with EGTA and the basin size is used as a measure of fitness. The experiment finally found a SQ algorithm with a particular parameter combination, which together with TT composes a Nash equilibrium that captures 97% of the strategy space populated by the learned strategy, TT, RE, and GD.

Combining reinforcement learning and EGTA

Schvartzman and Wellman [Schvartzman and Wellman, 2009b] conducted the most comprehensive EGTA of trading strategies for CDAs to date and interleaved the EGTA steps with reinforcement learning steps, culminating in strategies that were able to deviate from Nash equilibria formed by strategies from the literature and arrive at new equilibria supported by these new strategies.

First, Schvartzman and Wellman defined a new framework that used reinforcement learning (RL) [Sutton and Barto, 1998] to decide how to trade. An RL agent aims to maximize its long-term return through trial-and-error. The agent models its situated environment with *states*, explores *actions* that are available at different states, and takes the best actions to its knowledge. The states summarize market conditions with information involving prices of recent transactions, prices of outstanding shouts, time elapsed, number of units of commodities left to trade, and private values of these units. The actions at each state are mapped to prices to shout.

Next, Schvartzman and Wellman ran an EGTA of a given set of strategies chosen from the known strategies in the literature,[39] calculated the Nash equilibrium strategy, and trained agents playing

[39] No more than three strategies were involved at each step of EGTA, probably a deliberate choice of the authors to

the RL strategy against agents playing the equilibrium strategy until the RL strategy converged. If the payoff of the obtained RL strategy was higher than that of the equilibrium strategy,[40] meaning the RL strategy could deviate from the equilibrium, this new RL strategy joined the pure strategies that were present in the equilibrium for the next step of EGTA and a new RL strategy was trained. If the equilibrium strategy itself was a pure strategy or the obtained RL strategy failed to deviate from the equilibrium, another strategy from the known strategies in the literature was selected and put into the set of strategies for the next step of EGTA.[41] This process repeated until all the known strategies had been considered.

The strategies from the literature that were used in the experiments included KAPLAN, ZI-C, GD, GDX, AA, and RB as well as ZI_{btq}, a variant of ZI-C devised by the authors, which picks up a price randomly from a distribution between the private value and the corresponding market quote.[42] In the end, the iterative process of EGTA and RL reached a mixed equilibrium that was made up of two new strategies. A comparison between one of the two strategies and GDX revealed that the new strategy was more willing to give up profit in certain circumstances and acted faster to accept shouts with good prices than GDX.[43]

---

avoid higher computational costs with more strategies.

[40] At each step of EGTA it turned out that only one equilibrium existed in their experiments. In theory, multiple equilibria may exist and when this happens, one way to calculate the payoff is to use the average across these equilibria weighted by the size of their basins [Phelps et al., 2004].

[41] This pattern was not followed all the time as the experiments themselves were trial-and-error. For example, when a Nash equilibrium that was a pure GD was reached, the authors tweaked the RL model several times and trained the new models so as to find a RL strategy that could deviate from the pure equilibrium.

[42] ZI_{btq} is therefore less resistant in giving up profit than ZI and, not surprisingly, dominated by ZI in their experiments.

[43] It should be noted that the strategies derived through the iterative process were more competitive in the heterogeneous scenarios as shown by the EGTAs, but were not the best choice for homogeneous populations of traders where the aim was to maximize the social welfare. For example, the experiments in [Schvartzman and Wellman, 2009b] showed that the Nash equilibrium supported by a pure GD strategy yielded a higher payoff than any other homogeneous strategy equilibrium.

3.3 Agent-based auction mechanism design

The story of trading strategies in the preceding section was only one facet of research on auctions. Gode and Sunder's results suggested that auction mechanisms played an important role in determining the outcome of an auction, and this was further borne out by the work of Walsh *et al.* [Walsh et al., 2002], which also pointed out that results hinged on both auction design and the mix of trading strategies used.

According to classical auction theory, if an auction is *strategy-proof* or *incentive compatible*, traders need not bother to conceal their private values and in such auctions complex trading agents are not required. Indeed, mechanism design traditionally focused on designing strategy-proof auctions [Conitzer and Sandholm, 2007]. However, typical DAs are not strategy-proof. McAfee [McAfee, 1992] has derived a form of double auction that is strategy-proof, though this strategy-proofness comes at the cost of lower efficiency.

It has been common in the domain of DAs for researchers to take empirical approaches using machine learning techniques, sometimes combined with methods from traditional game theory. Instead of trying to design optimal auction mechanisms, the computational approach looks for relatively good auctions and aims to make them better, in a noisy economic environment with traders that are not perfectly rational.

3.3.1 A parameterized space of auctions

One can think of different forms of auctions as employing variations of a common set of the auction rules, forming a parameterized auction space. Wurman *et al.* [Wurman et al., 2001, 2002] and others [Rothkopf and Park, 2001] parameterized auction rules in a way that can be summarized as follows:

- Bidding rules: determine the semantic content of messages, the authority to place certain types of bids, and admissibility criteria for submission and withdrawal of bids.
  - How many sellers and buyers are there?
  - Are both groups allowed to make shouts?
  - How is a shout expressed?
  - Does a shout have to beat the corresponding market quote if one exists?
- Information revelation:
  - When and what market quotes are generated and announced?
  - Are shouts visible to all traders?
- Clearing policy:
  - When does clearing a market take place?
  - When does a market close?
  - How are shouts matched?
  - How is a transaction price determined?

The idea of parameterizing the auction space not only eases the heuristic auction mechanism design, but also makes it possible to 'search' for better mechanisms in an automated manner [Cliff et al., 2002; Phelps et al., 2002; Sandholm, 2003].

It is not yet clear how auction design, and thus the choice of parameter values, contributes to the observed performance of auctions. Thus it is not clear how to create an auction with a particular specification. It *is* possible to design simple mechanisms in a provably correct manner from a specification, as shown by Conitzer and Sandholm [Conitzer and Sandholm, 2003, 2004].

However it is not clear that this kind of approach can be extended to mechanisms as complex as DAs. As a result, it seems that double auction mechanisms have to be designed experimentally, at least for the foreseeable future.

Of course, doing things experimentally does not solve the general problem. A typical experimental approach is to fix all but one parameter, creating a one-dimensional space, and then measure performance across a number of discrete sample points in the space, obtaining a fitness landscape that is expected to show how the factor in question correlates to a certain type of performance and how the auction can be optimized by tweaking the value of that factor [Phelps et al., 2003]. In other words, the experimental approach examines one small part of a mechanism and tries to optimize that part.[44] The situation is complicated when more than one factor needs to be taken into consideration—the search space then becomes complex and multidimensional, and the computation required to map and search it quickly becomes prohibitive.

3.3.2 Evolving auction mechanisms

Instead of manual search, some researchers have used evolutionary computation to automate mechanism design [Phelps et al., 2010] in a way that is similar to the evolutionary approach to optimizing trading strategies. Indeed, the work has been carried out by some of the same researchers.

Cliff [Cliff, 2001a] explored a continuous space of auction mechanisms by varying the probability of the next shout (at any point in time) being made by a seller, denoted by Q_s. The continuum includes the CDA ($Q_s = 0.5$) and also two purely single-sided mechanisms that are similar to the English auction ($Q_s = 0.0$) and the Dutch auction ($Q_s = 1.0$). Cliff's experiments used genetic algorithms and found that a Q_s that corresponds to a completely new kind of auction led to a better α value than that obtained for other markets using ZIP traders. Walia *et al.* [Walia et al.,

[44]And of course there are rarely any guarantees as to the optimality of the results.

2002] and [Cliff et al., 2002] (the same authors but in a different order) continued with this work, showing that the approach is also effective in markets using ZI-C traders, and the new "irregular" mechanisms can lead to high efficiency with a range of different supply and demand schedules as well. The visualization of fitness landscapes in this work, using plots including 3D histograms and contours, is also noteworthy.

Byde [Byde, 2003] took a similar approach in studying the space of auction mechanisms between the first and second-price sealed-bid auctions. The winner's payment in an auction sampled from the space is determined as a weighted average of the two highest bids, with the weighting determined by the auction parameter. For a given population of bidders, the revenue-maximizing parameter is approximated by considering a number of parameter choices over the allowed range, using a GA to learn the parameters of the bidders' strategies for each choice, and observing the resulting average revenues. For different bidder populations (varying bidder counts, risk sensitivity, and correlation of signals), different auction parameter values are found to maximize revenue.

Taking another tack, Phelps *et al.* [Phelps et al., 2003] explored the use of genetic programming to determine auction mechanism rules automatically. *Genetic programming* (or GP), another form of evolutionary computation, evolves programs (or expressions) rather than the binary strings evolved in GAs. This makes automatic programming possible, and in theory allows even more flexibility and effectiveness in finding optimal solutions in the domain of concern. In GP, programs are traditionally encoded as *tree structures*. Each branching node has an operator function and each terminal node has an operand, making it easy to evolve and evaluate the tree structure. With this type of structure, crossover is applied on an individual by simply switching one of its nodes with another node from another individual in the population. Mutation can replace a whole node in the selected individual, or it can replace just the information of that node. Replacing a node means replacing the whole branch. This adds greater effectiveness to the crossover and mutation operators [Koza, 1992].

Phelps *et al.* demonstrated how GP can be used to find an optimal point in a space of pricing policies, where the notion of optimality is based on a combination of allocative efficiency and trader market power. In DA markets, there are two popular pricing policies: the k-DA pricing rule [Satterthwaite and Williams, 1993] and the uniform pricing policy. The former calculates the transaction price for a matched ask-bid pair as:

$$p = k \cdot p_a + (1-k) \cdot p_b$$

where $k \in [0,1]$, and p_a and p_b are ask and bid prices. The latter executes all transactions at the same price, typically the middle point of the interval between the market ask and bid quotes. Searching in the space of arithmetic combinations of shout prices and market quotes including the above two rules as special cases, led to a complex expression that is virtually indistinguishable from the $k = 0.5$ version of the k-DA pricing rule. This shows that the middle-point transaction pricing rule not only reflects the traditional practice but also can be technically justified.

Noting that the performance of an auction mechanism always depends on the mix of traders participating in the mechanism, and that both the auction mechanism and the trading strategies may adapt simultaneously, Phelps *et al.* [Phelps et al., 2002] further investigated the use of co-evolution in optimizing auction mechanisms. They first co-evolved buyer and seller strategies and then evolved trader strategies together with auction mechanisms. The approach was able to produce outcomes with reasonable efficiency in both cases.

3.3.3 Evaluating auction mechanisms

Phelps *et al.* [Phelps et al., 2004] proposed a novel way to evaluate and compare the performances of market mechanisms using heuristic strategy analysis.

Despite the fact that the performance of an auction mechanism may vary significantly when the mechanism engages different sets of trading agents, previous research on auctions analyzed the properties of DA markets using an arbitrary selection of homogeneous trading strategies. A more sound approach is to find the equilibria of the game between the participating trading strategies and measure the auction mechanism at those equilibrium points. As Sections 3.2.2 and 3.2.3 have discussed, the EGTA analysis calculates equilibria among a representative collection of strategies. This makes the method ideal for measuring market mechanisms at those relatively stable equilibria.

The representative strategies selected by Phelps *et al.* included RE, PVT, and TT. The EGTA analysis revealed that: (1) neither the CDA nor the CH mechanism was strategy-proof since TT was not dominant in either market; (2) increasing the number of agents in the CH led to the appearance of an equilibrium basin for an equilibrium near TT, which agreed with the conclusion drawn through the approximate analysis in [Satterthwaite and Williams, 1989] that BNE strategies converged to TT and inefficiency vanished fast as the market increases in size; and (3) the CH had higher efficiency than the CDA in the sense that the three equilibrium points[45] in the dynamics field for the CH all generated 100% efficiency while the only equilibrium[46] for the CDA produced 98% efficiency. One can interpret the small efficiency loss when moving to a CDA from a CH as a post-hoc justification of the NYSE's use of a CDA rather than a CH for faster transactions and higher volumes.

One avenue of future research is to combine this evaluation method with evolutionary computation to optimize DA mechanisms.

[45] Each fell onto one of the three pure strategies, though the sizes of their basins varied.
[46] Pure RE strategy.

3.3.4 Adaptive auction mechanisms

Considering that the information about the population of traders is usually unknown to the auction mechanism, and many analytic methods depend on specific assumptions about traders, Pardoe and Stone [Pardoe and Stone, 2005] advocated a self-adapting auction mechanism that adjusts auction parameters in response to past auction results.

Their framework included an *evaluator* module, which can create an auction mechanism for online use, monitor the performance of the mechanism, and use the economic properties of the mechanism as feedback to guide the discovery of better parameter combinations. This process then created better auction mechanisms that continued to interact with traders which were themselves possibly evolving at the same time. A classic algorithm for n-armed bandit problems, ε-greedy [Sutton and Barto, 1998], was used in the evaluator module to make decisions on parameter value selection.

The main feature of their work is that auction mechanisms were optimized during their operation while the mechanisms in the approaches discussed previously in this chapter remained static even when they faced a set of traders that were different from those used in searching for the mechanisms. In theory, traders and the auction mechanism may form a co-evolutionary system and both sides try to adapt towards a best response to the rest of the system.

Chapter 4

Competing marketplaces

Despite the variety of the work on auction mechanism design, it has one common theme—it all studies single marketplaces. In contrast, not only do traders in an auction compete against each other, real market institutions, like stock and commodity exchanges, face competition from other marketplaces [Shah, 1997]. Company stock is frequently listed on several stock exchanges. US companies, for example, may be listed on both the NYSE, the NASDAQ, and, in the case of larger firms, non-US stock markets like the London Stock Exchange (LSE). Indian companies can be listed on both the National Stock Exchange (NSE) and the Bombay Stock Exchange (BSE) [Shah and Thomas, 2000].

Such multiple marketplaces for the same goods induce complex interactions. The simplest example of this is the work of *arbitrageurs* who exploit price differences between marketplaces to buy low in one and sell high in another.[47] More complex dynamics occur when marketplaces compete, as when the NSE opened and proceeded to claim much of the trade volume from the established BSE [Shah and Thomas, 2000], or when the newly created Singapore International

[47] In addition, futures exchanges make it possible for dealers in a particular commodity to offset their risks by trading options—commitments to buy or sell at a future date at a certain price—in that commodity, and provide further opportunities for arbitrage.

Monetary Exchange (SIMEX) did the same to Japanese stock markets for index futures on Nikkei 225 in the late 1980s [Shah, 1997]. These changes took place over a long period of time, but inter-market dynamics can have much shorter timescales, as was the case in the flow between the CME and the NYSE during the global stock market crash of 1987 [Miller et al., 1988]. This kind of interaction between marketplaces has not been widely studied, least of all using automated traders.

In addition, previous studies usually present comparisons of auction mechanisms indirectly, using different proprietary settings that differ in available information, computational resources and so on. As a result, mechanisms are difficult to compare, and it is desirable to have some platform that helps to evaluate auction mechanisms in a direct, uniform way.

This chapter will first introduce a game that allows multiple competing marketplaces to run in parallel and traders to move between them, and then provide an analysis of scenarios in which trading agents choose between marketplaces that impose different levels of charges. Although the complexity of the game is much less than that in the real world, this is the first step to address the imbalance between the prior experimental work and the reality, and provides a useful tool through controlled simulations to gain insights on how to design effective auction mechanisms in such a competitive environment.

4.1 CAT games

The Trading Agent Competition (TAC)[48] Market Design Tournament [Niu et al., 2008b], also known as the CAT Tournament or CAT Game, was initiated in 2007 to evaluate market mechanisms

[48]The Trading Agent Competition [Wellman et al., 2003] was organized to promote and encourage high quality research into trading agents. Under the TAC umbrella, besides the CAT Game described here, a series of tournaments have been held, including TAC Classic, which ran from 2000 to 2006, TAC Supply Chain Management, which has run since 2003, and the latest TAC Ad Auctions, which has run twice since 2009. More information about TAC can be found at `http://tradingagents.org/`.

in a competitive environment.[49] Other TAC competitions have competing trading agents that aim to maximize their payoffs by interacting in a single marketplace. CAT games do just the opposite. Each entrant in the competition provides a *specialist* that regulates a marketplace with a set of auction rules, and these specialists compete against each other to attract traders and make profit. Traders in CAT games are provided by the game organizer and each of them learns to choose the best marketplace to trade in.[50]

The CAT tournaments were organized jointly by our Agents Lab at the City University of New York (CUNY) and collaborators from two universities in the United Kingdom, University of Liverpool and University of Southampton. The CUNY team was responsible for the detailed design of the game and the development of the supporting software platform, JCAT [Niu et al., 2008c]. I was the lead developer and the main contributor to this effort.

4.1.1 Game procedure

A CAT game lasts a certain number of *day*s, each day consists of *round*s, and each round lasts a certain number of *tick*s, or milliseconds. Each game involves traders, who buy or sell goods, and specialists, who provide marketplaces for those goods, enabling the trade. All traders and specialists are required to check in with the game server prior to the start of a game, and the list of all clients, including both traders and specialists, is broadcast to each client afterwards.

Before the opening of each day, the specialists are required to announce their price lists, which are then forwarded to all clients by the game server. After a day is opened, traders can register with one of the specialists (and only one specialist). Their choice of specialist depends on both the announced fees for that day and the profits obtained in previous days. Traders always tend to go to

[49]The CAT tournament has been held four times so far, affiliated with an academic conference each time. The first event was with AAAI'07, the second with AAAI'08, the third with IJCAI'09, and the fourth with EC'10.

[50]Theoretically a CAT game may have both traders and specialists submitted by entrants. Thus two competitions, one for traders and the other for markets, could be coupled together.

specialists where they expect the highest profits.

Actual trade is allowed only during a round, during which traders submit shouts to the specialists they are registered with. A specialist has the option to either accept or reject a shout. A shout becomes active once it is accepted, and remains active until it is successfully matched with another shout or the trading day ends. A specialist may match asks and bids any time during a round. A matched bid must have a price no lower than the corresponding ask, and the transaction price that is set must fall in between the bid and the ask.

After a day closes, information on the profit made by each specialist and the number of traders registered with it is disclosed, and this allows specialists to adapt or learn to improve their competitiveness.

4.1.2 Traders

Each trading agent is assigned private values for the goods to be traded, based on the independent private-value model that was described in Section 2.2. The private values and the number of goods to buy or sell determine the supply and demand of the markets. The private values remain constant during a day, but may change from day to day. Each trading agent is also endowed with a *trading strategy* and a *market selection strategy* to do two tasks. Respectively, one is to decide how to make offers, and the other is to choose the marketplace to make offers in. These two tasks allow our traders to exhibit intelligence in two, orthogonal, ways.

Trading strategies

Each trader uses one of the four trading strategies: ZI-C, ZIP, RE, and GD. The reason for picking these among numerous others is that these four strategies have been extensively researched in the literature and some of them have been shown to work well in practice. In particular, we pick

ZI-C because it is not making bids with any intelligence. Any effects that are observed have to be a result of market structure, rather than a consequence of the trading strategy, and hence will be robust across marketplaces inhabited by different kinds of trader. In addition, we would like to see whether an entrant in the CAT game can design a mechanism that takes advantage of this naïve strategy. The reason for picking ZIP and RE is that the former strategy is typical of the behavior of automated traders, while the latter is a good model of human bidding behaviors. Using both will give us results indicative of markets with both human and software traders. As for GD, our consideration is that the sophisticated strategy can help us to compare situations that involve effective traders and situations that do not. ZIP and GD require information about the offers made by other traders and the results of those offers that ZI-C and RE do not need, and so traders that use these strategies may incur higher costs when specialists impose charges on information about shouts and transactions.

Market selection strategies

The market selection strategies that may be adopted by a trading agent include:

- random: the trader randomly picks a marketplace;
- ε-greedy: the trader treats the choice of marketplace as an n-armed bandit problem which it solves using an ε-greedy exploration policy [Sutton and Barto, 1998, Section 2.2]. An ε-greedy trader takes daily profits as rewards when updating its value function.

  An ε-greedy trader chooses what it estimates to be the best marketplace with probability $1-\varepsilon$, and with probability ε chooses one of the remaining marketplaces (picking between them with equal probability). ε may remain constant or be variable over time, depending upon the value of a parameter α. If α is 1, ε remains constant, while if α takes any value in $(0,1)$, ε will decrease over time by a factor of α each step.

- softmax: the trader is similar to an ε-greedy trader except that it uses a softmax exploration policy in the n-armed bandit algorithm [Sutton and Barto, 1998, Section 2.3].

  Unlike an ε-greedy trader, a softmax trader does not treat all marketplaces, other than the best marketplace, exactly the same. If it does not choose the best marketplace, it weights the choice among the remaining marketplaces so that it is more likely to choose better marketplaces. There are of course a large number of different ways to do so, and the most common way is following a *Boltzman distribution*, according to which, the probability to choose action a among n actions is

$$\frac{e^{Q(a)/\tau}}{\sum_{i=0}^{n-1} e^{Q(a_i)/\tau}} \tag{4.1}$$

  where $Q(a_i)$ is the estimated value of action a_i and the positive parameter τ is called the *temperature*. τ is used to control the relative importance of the weights assigned to actions, or in the market selection scenario, marketplaces. Similar to ε, τ may be fixed or variable depending upon the value of α. When τ is much larger than the likely values of $Q(a_i)$, the trader chooses all marketplaces with approximately equal probabilities, while when $\tau \to 0$, the trader stops exploring and greedily chooses the best marketplace according to its experience. As the effect of τ depends upon its relative value compared to $Q(a_i)$, we normalize the estimated values of actions in this market selection strategy before using them in (4.1) such that the largest $Q(a_i)$ is always 1 and τ can be configured independent of the domain.

4.1.3 Specialists

Specialists facilitate trade by matching asks and bids and determining the trading price in an exchange marketplace. Each specialist operates its own exchange marketplace and may choose what-ever auction rules it desires. Specialists are permitted and even encouraged to have adaptive strategies such that the policies change during the course of a game in response to market conditions.

A specialist can set its fees, or *price list*, which are charged to traders and other specialists who wish to use the services provided by the specialist. Each specialist is free to set the level of the charges (from zero up to some reasonable upper bounds). These are the following:

- *Registration fees*. Fees charged for registering with a specialist.
- *Information fees*. Fees for receiving market information from a specialist.
- *Shout fees*. Fees for successfully placing asks and bids.[51]
- *Transaction fees.* A flat charge for each successful transaction.
- *Profit fees*. A share of the profit made by traders, where a trader's profit is calculated as the difference between the shout and transaction price.

The first four types of fees are each a flat charge, and the last one is a percentage charged on the profit made by the traders involved in the transaction. A trader pays the registration and information fees at most once every trading day.

4.1.4 Assessment

The performance of specialists in a CAT game is assessed every day on multiple criteria. To encourage sustainable operation,[52] not all the trading days are used for assessment purposes, despite

[51]In a CAT game, a specialist could reject a shout placed by a trader so that the shout will not even have a chance to get matched with a shout from the opposite side. There will be no shout fee charged if a shout is rejected.

[52]Traders provided in a CAT game are of limited intelligence and the limited intelligence to much extent relies upon their exploration between marketplaces over time. Entrants who simply want to win the game might therefore, for example, charge wildly high fees once in a while, and do not make much effort to design auction mechanisms that perform well in a long term. The probabilistic selection of assessment days increases the risk of specialists in charging high fees spontaneously.

the fact that the game has a start-day and an end-day, and the selected assessment days are kept secret to entrants until they have been passed.

Each specialist is assessed on three criteria on each assessment day:

- *profit share*: the profit share score of a specialist on a particular day is given by the total profits obtained by that specialist on that day as a proportion of the total profits obtained by all specialists on that same day.
- *market share*: of those traders who have registered with a specialist on a particular day, the market share score of a specialist on that day is the proportion of traders that have registered with that specialist on that day.
- *transaction success rate*: the transaction success rate score for a specialist on a given day is the proportion of asks and bids placed with that specialist on that day which that specialist is able to match. In the case where no shouts are placed, the transaction success rate score is calculated as zero.

Each of these three criteria results in a value between 0 and 1 for each specialist for each day. The three criteria are then weighted equally and added together to produce a combined score. Scores are then summed across all assessment days to produce a final game score for each specialist. The specialist with the highest final game score will win the game.

4.1.5 Competition platform

JCAT [Niu et al., 2008c] is the platform that was built to support CAT games. JCAT extends the single-threading JASA package [Phelps, 2005] and adopts a client/server architecture. As Figure 4.1 illustrates, the CAT server works as a communication hub and the central time controller, and CAT clients—either specialists or traders—communicate with each other via the server. On one side, the

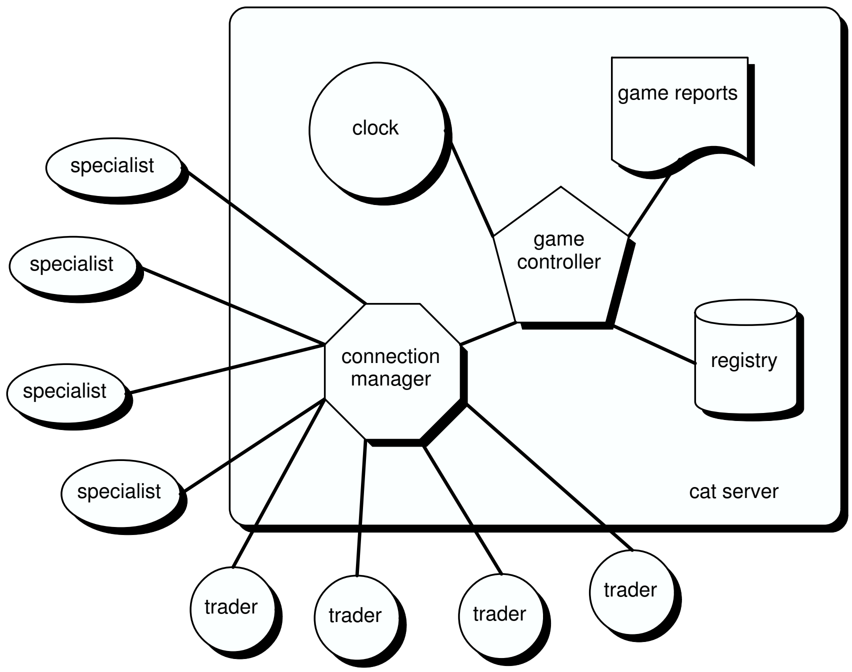


Figure 4.1: The architecture of JCAT.

CAT server takes traders' requests, including registering with a specialist, placing and modifying shouts, and forwards them to specialists; on the other side, specialists notify the CAT server of matching shouts and, via the server, inform traders. The behaviors of the CAT server and CAT clients are regulated by the CAT Protocol, or CATP [Niu et al., 2007b]. The CAT server uses a registry component to record all game events and validate requests from traders and specialists. Various game report modules are available to process subsets of game events, calculate and output different metrics for post-game analysis.

JCAT implements a parameterized framework for double auctions, which was published in [Niu et al., 2008b]. The framework can be easily extended to accommodate new auction rules. Chapter 6 will describe this framework in detail and discuss how to employ this structure to construct auction mechanisms.

4.2 On the behavior of competing marketplaces populated by automated traders

To examine the effectiveness of CAT games in evaluating the relative strength of market mechanisms, we ran a series of experiments that include marketplaces with a range of charging policies and traders that use simple reinforcement learning rules to select among marketplaces. This section will present the results that we obtained. The results were first published in [Niu et al., 2007a].

In doing this, our work has a different focus from the work on market mechanisms that was reviewed in Chapter 3. That work is focused on how the performance of traders helps achieve economic goals like high efficiency [Gode and Sunder, 1993a] and trading near equilibrium [Cliff and Bruten, 1997], or how traders compete amongst themselves to achieve high profits [Tesauro and Das, 2001]. In contrast, we are interested in competition between *marketplaces*, what the movement of traders is when they are faced with a variety of market mechanisms that are deployed in these marketplaces, and what effect their movement has on the profits of those marketplaces.

4.2.1 Experimental setup

Traders, each using a trading strategy and a market selection strategy—one of those described above—are not only learning how best to make offers, which they will have to do anew each time they change marketplace, but they are also learning which marketplace has the best auction mechanism for them. Of course, which marketplace is best will depend partly on the properties of different marketplaces, but also on which other traders are in those marketplaces.

Marketplaces may levy charges on traders, as real marketplaces do. While we can set up marketplaces to charge traders in a variety of ways, we have concentrated on charging traders a proportion of the surplus on a transaction in which they are involved—the profit fee described in

Section 4.1.3. We focus on this because it mirrors the case of the competition between the NSE and the BSE [Shah and Thomas, 2000] where the BSE, had a much higher charge on transactions than the new stock market.

In particular, we experimented with four basic charging policies, one fixed and three simple adaptive policies:

- *Fixed charging* (GF)[53] sets charges at a specified fixed level.
- *Charge-cutting charging* (GC) sets the charges by scaling down the lowest charges of marketplaces imposed on the previous day. This is based on the observation that all other things being equal traders will prefer marketplaces with lower charges.
- *Bait-and-switch charging* (GB) makes a specialist cut its charges until it captures a certain market share, and then slowly increases charges to increase profit. It will adjust its charges downward again if its market share drops below a certain level.
- *Learn-or-lure-fast charging* (GL) adapts its charges towards some desired target following the scheme used by the ZIP trading strategy. If the specialist using this policy believes that the traders are still exploring among specialists and have yet to find a good one to trade in, the specialist would adapt charges towards 0 to lure traders to join and stay; otherwise it learns from the charges of the most profitable marketplace. GL uses an exploring monitor component to determine whether traders are exploring or not. A simple exploring monitor, for example, examines the daily distribution of market shares of specialists. If the distribution is flat, the traders are considered to be exploring, and otherwise not. This is based on the observation that traders all tend to go to the best marketplace and cause an imbalanced distribution. Another scheme for the exploring monitor, which we did not implement, is to

[53]The abbreviation follows the norm used in [Niu et al., 2008b]. Since CX is already used for market **c**learing policies, we use GX to represent char**g**ing policies.

check the trader distribution in the most recent days and use the relative market share gain and loss to determine whether it is good at luring traders.

In common with much work in computational economics [Friedman, 1998], the strategies used both by traders to choose between marketplaces and by marketplaces to decide how to charge traders are very simple. Our choice of market strategies was driven by the desire to first establish the relative performance of simple charging policies, and thus the basic structure of the problem of competing marketplaces, before trying more complex policies.

Each of the experiments is setup to run for 200 or 400 trading days, with every day being split into 10 rounds, each of which is one second long. The marketplaces are populated by 100 traders, evenly split between buyers and sellers. Each trader is permitted to buy or sell at most one unit of goods per day, and each trader has a private value for that good which is drawn from a uniform distribution between \$50 and \$150.

4.2.2 Experimental results

Results are given in Figures 4.2a to 4.10d. These show values averaged over 100 runs of each experiment.

Fixed-charge marketplaces

The first set of experiments explores the properties of marketplaces with fixed charges, respectively 20%, 40%, 60% and 80% of the surplus on a transaction. These marketplaces are denoted as $M_{0.2}$, $M_{0.4}$, $M_{0.6}$, and $M_{0.8}$ respectively. Figures 4.2a and 4.2b show that traders that pick marketplaces randomly have no discernable pattern of movement between the GF marketplaces, just as we would expect. As a result, the marketplace with the highest charges makes the most profit. In contrast, Figure 4.2c and 4.2d, when traders pick marketplaces based on their personal profits, they move towards the GF marketplace with lowest fixed costs. While GF marketplaces with high charges make initial windfall profits, the trend is for the lower charging marketplace to gain greater cumulative profit as the number of trading days increases.

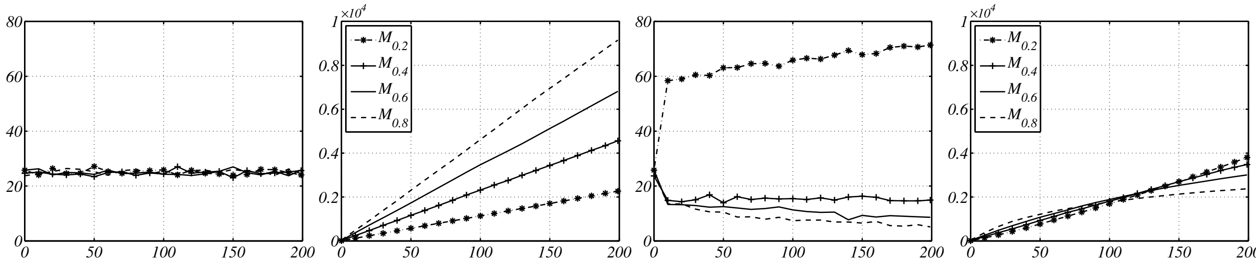
Figures 4.3a–4.3d show that these results are robust against the ability of traders to make sensible trades since broadly the same results are observed when some or all of the traders make their bidding decisions randomly. Figures 4.4a–4.4d test the sensitivity of the results to the strategy that traders use to learn which marketplace to choose. Decreasing ε over time (Figures 4.4a and 4.4b) does not seem to have much effect, but switching to the softmax strategy (Figures 4.4c and 4.4d) reduces the speed of convergence. The softmax strategy reduces the attractiveness of the lowest charging marketplace since traders that do not pick it tend to pick marketplaces where they can still make good profits and this reduces the incentive to pick $M_{0.2}$.

The final results for the experiments with only fixed charges, Figures 4.5a–4.5d, show that the results obtained so far are very sensitive to the length of time agents have to learn about the marketplaces. When some traders start learning afresh every day, simulating traders leaving and entering the marketplaces (4.5c and 4.5d), the lowest charging marketplace might still capture most of the traders, but it captures less of them, and the remaining marketplaces attract enough traders to have the same profit profile as when there is no learning (Figures 4.2a and 4.2b).

Thus, for the fixed charge marketplaces, provided that there is no turnover of traders, it is a winning strategy to undercut the charges of the other marketplaces.

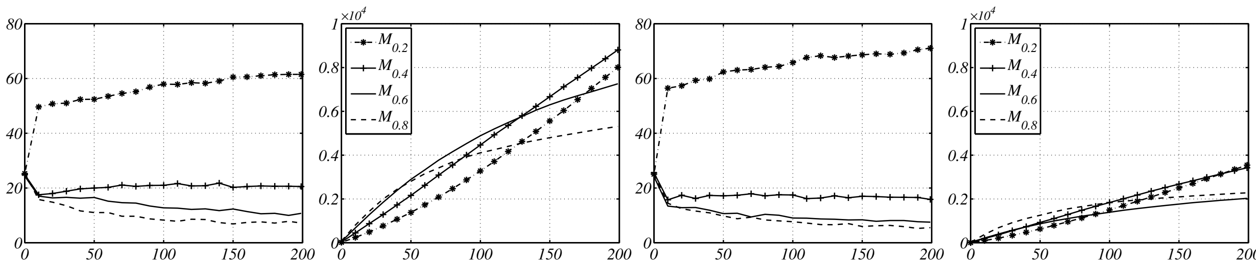
Homogeneous, adaptive-charge, marketplaces

Turning to the adaptive charging strategies, we first tested them against copies of themselves. In these experiments we ran four copies of each kind of marketplaces against each other with different initial profit charges, using the same values of the charges as we used in the experiments above. In



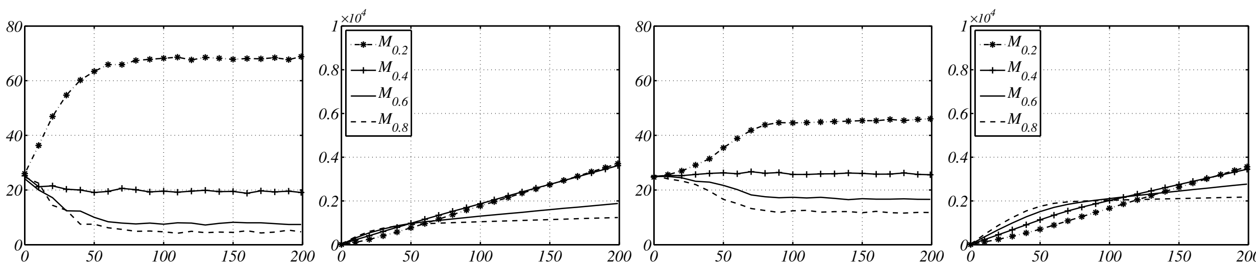
(a) Number of traders daily. (b) Cumulative profit. (c) Number of traders daily. (d) Cumulative profit.

Figure 4.2: Baseline experiments. GD traders, (a) and (b) with random market selection, (c) and (d) with ε-greedy market selection ($\varepsilon = 0.1$, $\alpha = 1$).



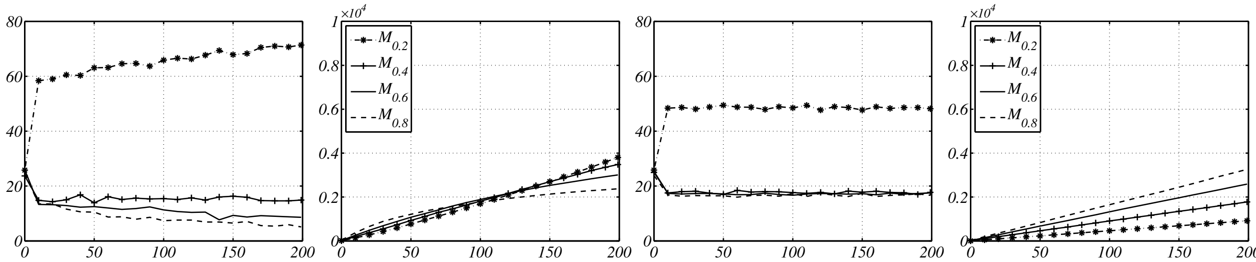
(a) Number of traders daily. (b) Cumulative profit. (c) Number of traders daily. (d) Cumulative profit.

Figure 4.3: Robustness experiments. (a) and (b) show ZI-C traders, and (c) and (d) show a mixture of GD and ZI-C traders, all traders use ε-greedy market selection ($\varepsilon = 0.1$, $\alpha = 1$).



(a) Number of traders daily. (b) Cumulative profit. (c) Number of traders daily. (d) Cumulative profit.

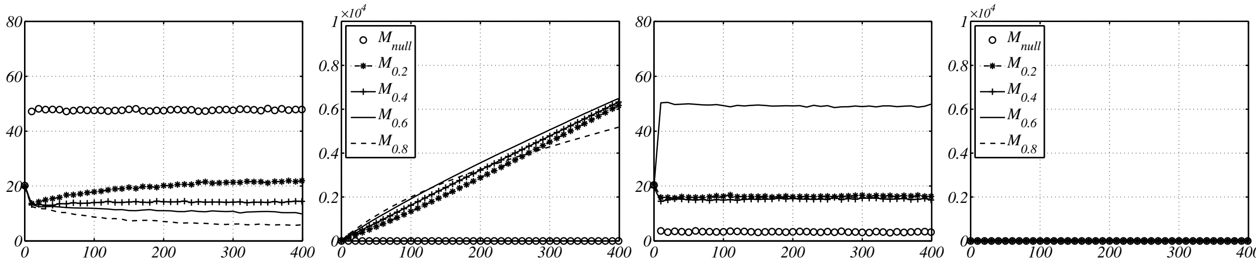
Figure 4.4: Learning experiments. GD traders, (a) and (b) with ε-greedy traders ($\varepsilon = 1$, $\alpha = 0.95$), (c) and (d) with softmax traders ($\tau = 1$, $\alpha = 0.95$).



(a) Number of traders daily. (b) Cumulative profit. (c) Number of traders daily. (d) Cumulative profit.

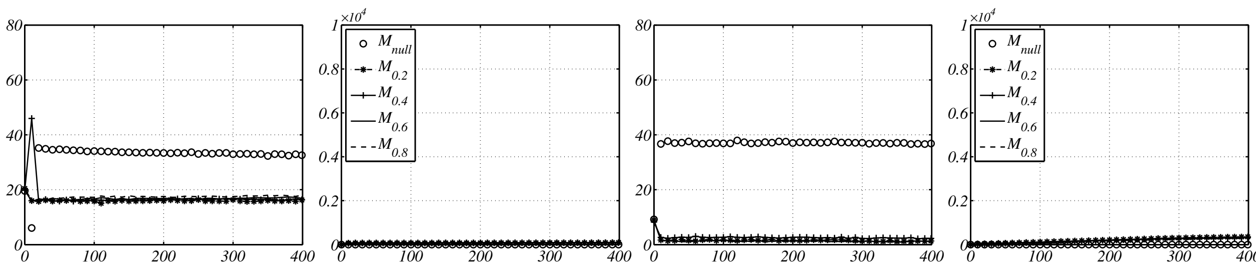
Figure 4.5: Population experiments. GD traders, all traders use ε-greedy market selection ($\varepsilon = 0.1$, $\alpha = 1$). In (a) and (b), all traders learn continuously through the experiment. In (c) and (d), 10% of the traders re-start learning every day.

each experiment we also provided a "null" marketplace, denoted as M_{null}, which made no charges and executed no trades—the idea of this is to allow traders who cannot trade profitably to have a mechanism for not trading—and, for completeness, carried out the same experiment with the GF marketplaces. For all of these experiments, and all subsequent experiments, we used traders that made bids using GD, that selected marketplaces using an ε-greedy policy, and that continued learning for all 400 days. The results of these experiments can be seen in Figures 4.6a to 4.7d. The GF marketplaces, in Figures 4.6a and 4.6b, attract fewer traders in the presence of the null marketplace, but make similar profits (since the traders who tend to the null marketplace do not often trade). The charge-cutting or GC marketplaces in Figures 4.6c and 4.6d, get into a price war which they do not have the intelligence to get out of, and the bait-and-switch or GB marketplaces (Figures 4.7a and 4.7b) are similarly unable to generate a significant profit. The learn-or-lure-fast or GL marketplaces, in Figures 4.7c and 4.7d, adjusting their profit margins to fit what the traders will allow, manage to do better, but generate nowhere near as much profit as the GF marketplaces do.



(a) Number of traders daily. (b) Cumulative profit. (c) Number of traders daily. (d) Cumulative profit.

Figure 4.6: Homogeneous marketplaces: GF marketplaces and GC marketplaces. GD traders, all traders use ε-greedy market selection ($\varepsilon = 0.1,\ \alpha = 1$). (a) and (b) are four homogeneous GF marketplaces, (c) and (d) are four homogeneous GC marketplaces.



(a) Number of traders daily. (b) Cumulative profit. (c) Number of traders daily. (d) Cumulative profit.

Figure 4.7: Homogeneous marketplaces: GB marketplaces and GL marketplaces. GD traders, all traders use ε-greedy market selection ($\varepsilon = 0.1,\ \alpha = 1$). (a) and (b) are four homogeneous GB marketplaces, (c) and (d) are four homogeneous GL marketplaces.

Heterogeneous, adaptive-charge, marketplaces

While the homogenous marketplace experiments give some idea of market performance, it is more interesting to examine how the adaptive charging strategies work in competition against one another. To explore this, we carried out a series of mixed marketplace experiments along the lines of the trading strategy work of [Tesauro and Das, 2001]. For each of the adaptive charging policies, we ran an experiment in which all but one marketplace used that policy and the remaining marketplace used another policy, carrying out one such "one-to-many" experiment for each of the

Table 4.1: Results of one-to-many experiments. For each experiment, the table gives the cumulative profit of the "one" policy vs. the cumulative profit of the best of the "many" over a certain period of time during the experiment, and shows whether the "one" is greater or less than the "many" at the 90% confidence level, determined by a t-test. Standard deviations are shown in *italics*.

(a) Cumulative profit over all days only.

	Many GC			**Many** GB			**Many** GL		
One GC		-		0.8	<	84.1	6502.2	>	6043.6
				7.5		*105.6*	*1527.1*		*2159.7*
One GB	82.0	>	0.7		-		6545.7	>	5743.8
	56.7		*6.8*				*2325.0*		*1581.8*
One GL	2289.6	>	0.8	1773.5	>	166.9		-	
	1118.9		*8.5*	*633.0*		*264.8*			

(b) Cumulative profit over the last 100 days.

	Many GC			**Many** GB			**Many** GL		
One GC		-		0.0	<	7.2	1727.5	>	1475.3
				0.0		*33.5*	*438.8*		*610.6*
One GB	5.9	>	0.0		-		2048.0	>	1397.7
	40.2		*0.0*				*829.3*		*432.1*
One GL	206.1	>	0.0	147.2	>	70.2		-	
	173.4		*0.0*	*54.4*		*227.6*			

other policies. In other words, we tested every "one-to-many" combination of the adaptive strategies. For all of these experiments, we measured the cumulative profit of a marketplace using the charging policies and ran the marketplaces alongside the same null marketplace as before. As in the experiments with homogeneous, adaptive-charge, marketplaces, there are in total four marketplaces besides the null one, and the one-to-many ratio is always 1 : 3. The similar configurations make it possible to compare the results of the two sets of experiments.

Table 4.1a gives the results of "one-to-many" experiments, giving the cumulative profits of the "one" marketplace against the best performing "many" marketplace for each combination of the adaptive marketplaces. The table also indicates which profit is the greater at the 90% confidence level (as determined by a t-test). "$>$" means the "one" marketplace is better than the best "many" marketplace at the 90% confidence level and "$<$" means the best "many" marketplace is better. The day by day results are also given in Figures 4.8a-4.10d. The results show that one GC marketplace is effective against many GL marketplaces, since it can capture more traders, as Figure 4.9a shows. In such a case, both types of marketplace generate good profits. However, when there is more than one GC, such marketplaces get into a price war and drive their charges down to zero.

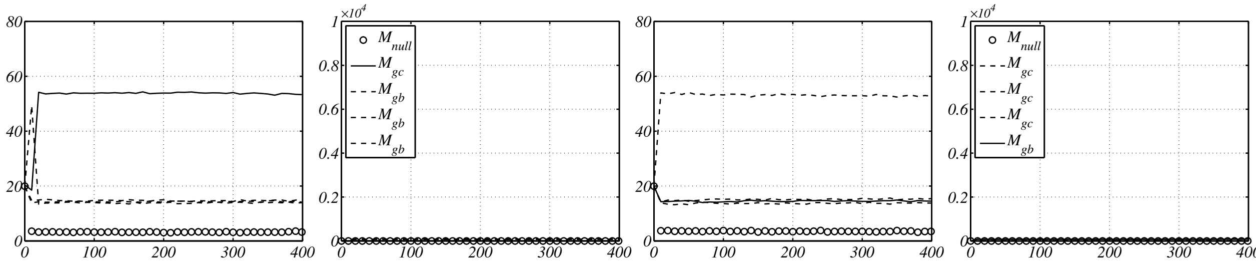
The GB policy was envisaged as a more sophisticated version of GC, one that exploited its market share by increasing charges on traders it had attracted through low charges. The results in Table 4.1a suggest that GB achieves this intention, outperforming GC both when one GB takes on multiple GCs, and when a single GC competes against multiple GB marketplaces. However, as is the case with GC, when there is more than one marketplace using GB, they may end up cutting charges in a futile attempt to increase market share and hence do not make much profit—this is what happens when there are many GB marketplaces running against a single GL marketplace.

The GL policy, designed to get out of price wars by increasing charges when it can, performs well against both GC and GB marketplaces when it is in the minority. When there is only one GC or GB against many GL marketplaces, the GC and GB may outperform GL. However, even when this is the case, as Figure 4.9b and Figure 4.10b show, GL can still make more profit than the other policies in the short run (before 200 days have elapsed).

The results in Table 4.1a are cumulative over the entire 400 days of the experiment. Since the early days of the experiment often contain a lot of noise from the initial exploration of the traders, it is interesting to also look at the profits over just the later stages of the experiments, when trader movement has settled down. Such results are presented in Table 4.1b. These results suggest that when it is in the majority, the GL strategy is clearly outperformed by both a single GC and a single GB marketplace. Since Figures 4.9a and 4.10a suggest that there is no longer much movement of traders at this point, the results in Table 4.1b simply reinforce those in Table 4.1a.

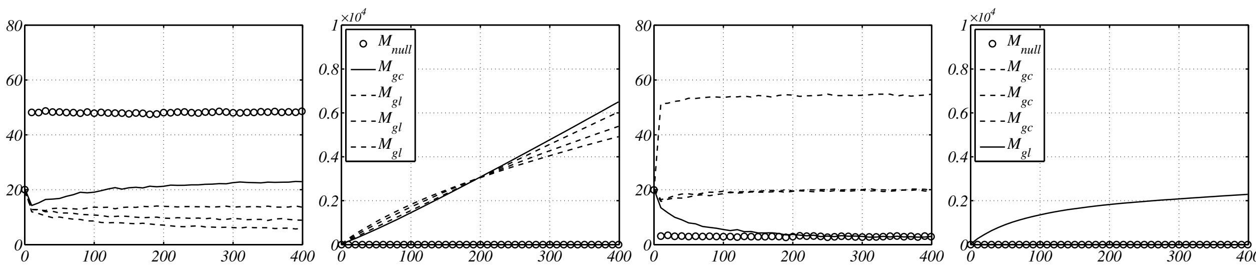
In our experiments described above, market performance depends on the mix of market strategies being considered. This suggests that, as is the case for trading strategies [Tesauro and Das, 2001], it may be difficult to find a dominant strategy for deciding market charges, though such a conclusion must wait until market strategies have been investigated further. This is particularly important since the strategies that we have considered were, quite intentionally, about the simplest we could imagine (starting with simple strategies seemed a good way to understand the problem we are considering).

Some of the experimental results might look obvious, but they are useful in validating the design of the CAT game and the implementation of JCAT. In addition, the results obtained with marketplaces that adopt simple auction mechanisms build a solid foundation for understanding the much more complicated interaction between specialists in the actual CAT tournaments.



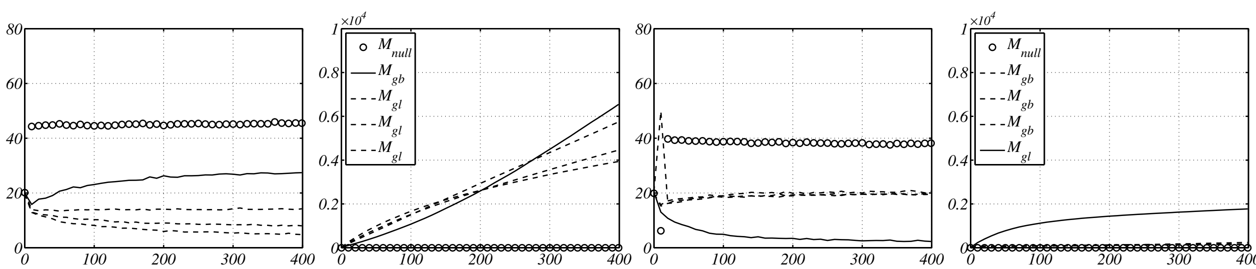
(a) Number of traders daily. (b) Cumulative profit. (c) Number of traders daily. (d) Cumulative profit.

Figure 4.8: Heterogeneous marketplaces: GC against GB. GD traders, all traders use ε-greedy market selection ($\varepsilon = 0.1$, $\alpha = 1$). In (a) and (b) one GC marketplace (solid line) competes with three GB marketplaces, in (c) and (d), one GB marketplace (solid line) competes with three GC marketplaces.



(a) Number of traders daily. (b) Cumulative profit. (c) Number of traders daily. (d) Cumulative profit.

Figure 4.9: Heterogeneous marketplaces: GC against GL. GD traders, all traders use ε-greedy market selection ($\varepsilon = 0.1$, $\alpha = 1$). In (a) and (b) one GC marketplace (solid line) competes with three GL marketplaces, in (c) and (d), one GL marketplace (solid line) competes with three GC marketplaces.



(a) Number of traders daily. (b) Cumulative profit. (c) Number of traders daily. (d) Cumulative profit.

Figure 4.10: Heterogeneous marketplaces: GB against GL. GD traders, all traders use ε-greedy market selection ($\varepsilon = 0.1$, $\alpha = 1$). In (a) and (b) one GB marketplace (solid line) competes with three GL marketplaces, in (c) and (d), one GL marketplace (solid line) competes with three GB marketplaces.

Chapter 5

An analysis of entries in the First TAC Market Design Tournament

The previous chapter introduced CAT games and showd that the migration of trading agents with simple reinforcement learning capabilities can effectively distinguish marketplaces that differ in charging policies they adopt. This suggests that CAT games can be used to evaluate auction mechanisms in a competing environment.

More specifically, CAT games enable us to do two things:

1. to zoom in on the detailed interaction between competing marketplaces and trading agents, and examine, through the dynamic movement of trading agents in particular, how a combination of auction rules, each regulating one aspect of a market, work together to form an effective mechanism; and

2. to zoom out, viewing a complete CAT game as an atomic interaction between marketplaces, and examine how a market mechanism can win out in the long run through repeated games.

These two perspectives are actually akin to white-box testing and black-box testing in software engineering. As the two testing methods complement each other to produce correct code, the two perspectives above together help to evaluate the effectiveness of auction mechanisms, identify the weaknesses, and make further improvements.

5.1 Strategy evaluation in competitive games

Trading competitions like CAT games have been an effective tool in fostering innovative approaches, creating enthusiasm, and advocating exchange among researchers [Stone and Greenwald, 2005; Wellman et al., 2003]. However, the competitions themselves usually cannot provide a complete view of the relative strength and weakness of entries. In a competition, the performance of one player closely depends upon the composition of its opponents and the competition configuration, and the scenarios considered are usually limited. Thus we typically turn to post-competition analysis to scrutinize the design of each entry.

Ideally, such an analysis will cover all possible scenarios, but this usually presents too large a possible space. As a result, a common practice is to deliberately select a limited number of representative strategies and run games corresponding to a set of discrete points or trajectories in the infinite space, assuming that the results are representative of what would happen in the whole space were one to explore it [Sodomka et al., 2007].

There are two common types of approaches to post-competition analysis: *white-box* approaches and *black-box* approaches. A white-box approach attempts to relate the internal logic and features of strategies to game outcomes. In the Santa Fe Double Auction Tournament and post-tournament experiments [Rust et al., 1993], a thorough examination of auction efficiency losses indicated that the success of the KAPLAN trading strategy was due to its patience in waiting to exploit other trading strategies. In Axelrod's Computer Prisoner's Dilemma Tournament [Axelrod, 2006], the strong showing of TIT FOR TAT was attributed to the fact that it was forgiving as well as being cooperative. While a white-box approach is often domain-dependent, the insights obtained in the domain of interest may still be extended to other domains. For instance, the payoff structure in the Iterated Prisoner's Dilemma problem captures the nature of many other issues that are faced by parties with conflicting interests.

A black-box approach, on the other hand, considers strategies as atomic entities. One perspective is an *ecological* one based on replicator dynamics, from which the entities are biological individuals in an infinitely large population and a sub-population playing a particular strategy grows in proportion to how well that strategy performs relative to the whole population on average [Fudenberg and Levine, 1998]. Examples include some work that were introduced in Chapter 3. Walsh *et al.* [Walsh et al., 2002] combined the game-theoretic solution concept of Nash equilibrium and replicator dynamics, turning a potentially very complex, multi-stage game of trading strategies into a one-shot game in normal form, and used perturbation analysis to evaluate whether a strategy can be improved further. Phelps *et al.* [Phelps et al., 2005, 2006] and Schvartzman and Wellman [Schvartzman and Wellman, 2009b] successfully applied this approach in acquiring better trading strategies for DA markets, a variant of RE and a variant of GD respectively. Jordan *et al.* [Jordan et al., 2007] took a similar approach to the evaluation of entries in TAC SCM and other games [Jordan and Wellman, 2007].

This chapter introduces our work in taking both a white-box approach and a black-box approach to the analysis of the entries to the First CAT Tournament (CAT 2007) that are available in the TAC agent repository.[54] The results of this work have been published in [Niu et al., 2008b] (white-box) and [Niu et al., 2008a] (black-box) respectively. In the white-box analysis, we attempted to relate market dynamics to the auction rules adopted by these entries and their adaptive strategies via a set of post-tournament experiments. Based on this analysis, we speculated about the design of effective auction mechanisms, both in the setting of the competition and in the more

[54] `http://www.sics.se/tac/showagents.php`.

general case. In the black-box analysis, we examined the relative strength and weakness of the specialist agents across several scenarios, which demonstrated some vulnerabilities of entries that placed highly in the competition.

5.2 A white-box analysis of CAT 2007 entries

We ran a series of games with the same setup as in the CAT 2007 games.[55] Every game in our experiment ran for 500 trading days with 10 rounds per day and 1 second per round. The trader population comprised 180 ZIP traders, 180 RE traders, 20 ZI-C traders, and 20 GD traders. Buyers and sellers were evenly split in each trader sub-population. The private values of all the traders were independently drawn from a uniform distribution between $50 and $150, and each trader was allowed to buy or sell up to three units of a commodity per day. The specialists in our games include all eight of the 2007 specialists that have been released in the TAC agent repository. The same scoring criteria were used as in the competition, which have been described in Section 4.1.4, but, unlike the competition, all the game days were assessed. The results and plots shown below were averaged over a total of ten games. To obtain a clearer view, plots were smoothed out with each datum being the average of a ten-day sliding window around it.

The results of our experiments broadly agree with the rankings in the competition.[56] The CAT 2007 champion, `IAMwildCAT`, still wins in our experiments and `PSUCAT`, which placed second in the competition, comes second as shown in Table 5.1. The only changes in ranking are due to `TacTex` and `MANX` increasing their scores over what they achieved in the competition since they could participate in every game. In CAT 2007 both missed part of the competition due to unexpected travel delays.

[55] Subsequent to the analysis undertaken here, two teams have reported on their specialist strategies [Petric et al., 2008; Vytelingum et al., 2008b].

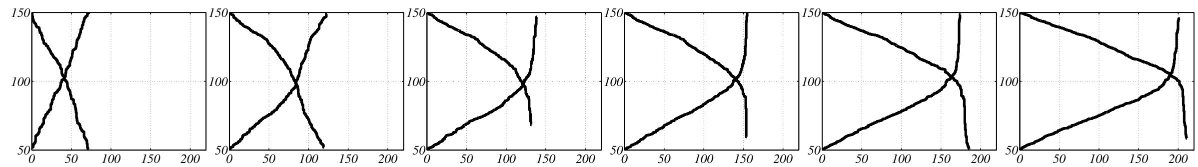
[56] `http://www.marketbasedcontrol.com/blog/index.php/?p=30`.

Table 5.1: The scores of specialists from CAT 2007 in the post-tournament experiments.

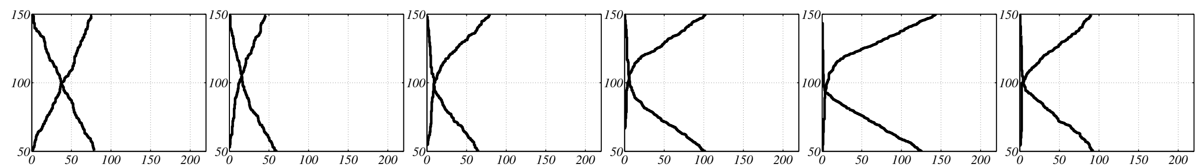
Specialist	Score	SD
IAMwildCAT	240.22	2.82
PSUCAT	209.26	12.01
CrocodileAgent	179.64	17.53
jackaroo	182.80	24.30
PersianCat	128.82	5.57
Mertacor	100.11	8.57
TacTex	166.66	8.99
MANX	140.09	31.03

Among many other things that we reported in [Niu et al., 2008b], the migration of intra-marginal traders and extra-marginal traders reflects the competition among specialists. Traders migrate based on estimates of expected profits, where the estimate for trading with a given specialist is based on past experience with that specialist. Generally speaking, the more intra-marginal traders and the fewer extra-marginal traders in a marketplace, the more potential profit there is, and the easier it is to make transactions and achieve a high transaction success rate. The intuitive way to explore how the intra-marginal and extra-marginal traders move between the marketplaces is to look at the snapshots of supply and demand in the marketplaces. Figure 5.1 illustrates the supply and demand curves on six selected days in one of the ten games (from left to right in each row, days 0, 50, 100, 150, 300, and 499) for three of the eight marketplaces, `IAMwildCAT`, `PersianCat`, and `MANX`. The curves show different trends in the three marketplaces. `IAMwildCAT` attracted an increasing number of intra-marginal traders, which at the end of the game account for almost two thirds of the intra-marginal traders in the global market.[57] `PersianCat` basically did the opposite,

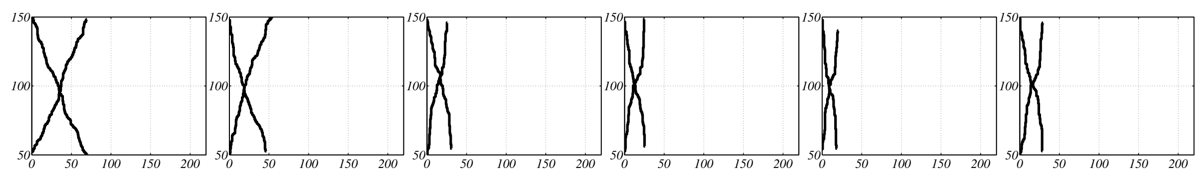
[57] Due to the uniform distribution of private values, the number of intra-marginal traders and the number of extra-marginal traders in a game are basically same, around 100. As each trader is entitled to trade up to three units of goods with its assigned private value, the intra-marginal supply/demand and the extra-marginal supply/demand in the global market are about 300 units of goods apiece. Figure 5.1 shows that almost 200 units of supply/demand in the marketplace of `IAMwildCAT` are on the intra-marginal side, which is about two thirds of the global intra-marginal supply/demand.



IAMwildCAT.



PersianCat.



MANX.

Figure 5.1: Supply and demand curves for individual marketplaces over time. Each graph has quantity on the x-axis and price on the y-axis. The leftmost graph gives supply and demand on day 0, and the remaining graphs in each row are those from days 50, 100, 150, 300, and 499 respectively. These graphs are from the same single run of the game.

with more extra-marginal traders and fewer intra-marginal traders coming in. MANX maintained a balanced supply or demand between the intra-marginal side and the extra-marginal side, though supply and demand on both sides shrank by 50% approximately at the end of the game. Though the supply and demand curves are intuitive in exhibiting the profitabilities of different marketplaces, it would be better to have a metric that makes it possible to compare the profitabilities quantitatively in a succinct way. To this end, we introduce the *marginal coefficient*, β, to measure the balance of intra-marginal and extra-marginal supply and demand. For demand,

$$\beta_D = \frac{D_i}{D_i + D_e} \tag{5.1}$$

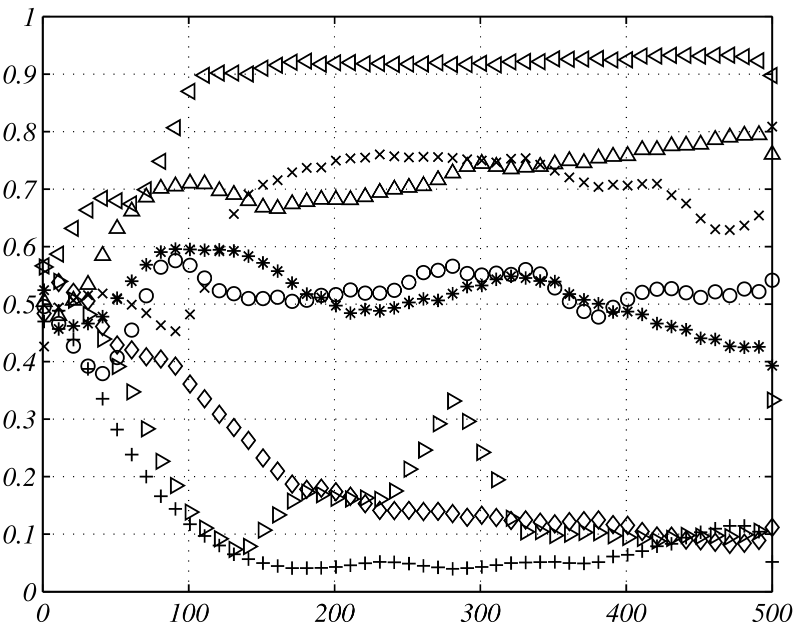
where D_i is the intra-marginal demand—the equilibrium—and D_e is the extra-marginal demand. The marginal coefficient for supply, β_S, can be defined similarly. β_D varies between 0 and 1. A value of 0 indicates that all the buyers in the market are extra-marginal while 1 indicates that all the buyers are intra-marginal. Figure 5.2a shows the daily value of β_D in the individual marketplaces managed by the specialists.

As Figure 5.2a shows, β_D is approximately 0.5 in all the marketplaces when the game starts. Then β_D of `IAMwildCAT`, `TacTex`, and `PSUCAT` increases while that of `CrocodileAgent`, `PersianCat`, and `Mertacor` decreases. Since a falling β_D indicates losing intra-marginal traders and/or gaining extra-marginal traders, these changes indicate that intra-marginal traders and extra-marginal traders have different preferences over the different marketplaces.

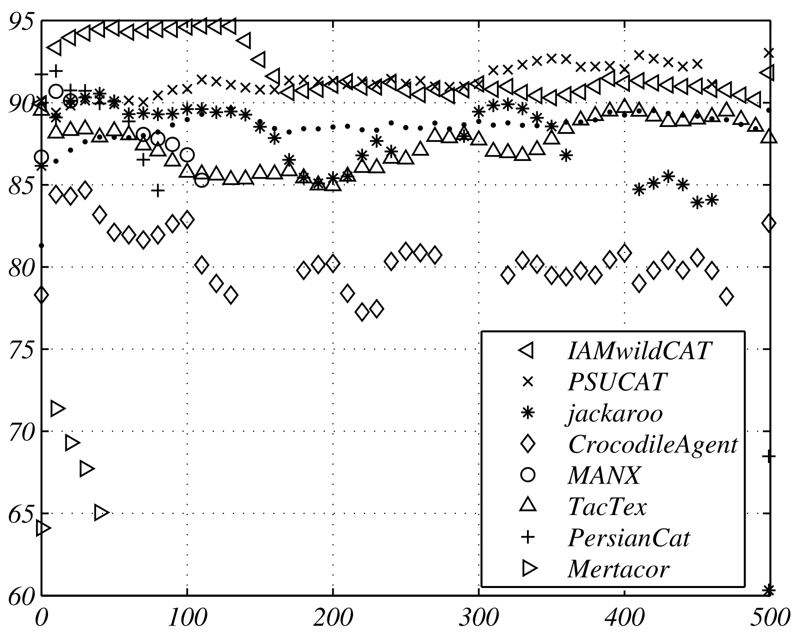
A close examination of `Mertacor`'s mechanism found that it strategically executes extra-marginal trades so as to increase its transaction success rate, and it has a bug leading to unrealized intra-marginal trades. These two issues are further confirmed by the low allocative efficiency of `Mertacor`, shown in Figure 5.2b, and provide sufficient 'excuse' for intra-marginal traders to flee.

`PersianCat` and `CrocodileAgent` both lose traders due to imposing high profit charges. `PersianCat` charges 100% on profit for the whole game, and this drives β_D down very quickly. `CrocodileAgent` levies a lower fee than `PersianCat` and therefore has a modestly decreasing β_D. The decrease of β_D in `PSUCAT` and `jackaroo` starting from days 250–300 follows the aggressive increase in the profit fee.

The rest of the specialists have much higher β_D despite their use of similar policies. `IAMwildCAT`, for instance, though adopting a matching component similar to the one used in `Mertacor`, refrains from using it in the early rounds of a day, which are usually sufficient to realize most intra-marginal trades. `MANX`, though levying a high, yet volatile, profit fee, also levies other fees without bias considerations, which together scare away both extra-marginal traders and intra-marginal



(a) Daily marginal coefficient of demand, β_D.



(b) Daily allocative efficiency.

Figure 5.2: Properties of individual marketplaces within a CAT game of 500 days.

traders at approximately the same pace. Its β_D therefore zigzags around 0.5. The three specialists that obtain a β_D higher than 0.6 during the most time of the game, `IAMwildCAT`, `PSUCAT`, and `TacTex`, all generate allocative efficiency higher than 85%, again suggesting the importance of matching policies in keeping a high-quality trader population.

The white-box analysis in [Niu et al., 2008b] presented a complete view of the interaction dynamics in the post-tournament experiments and speculated about effective auction mechanisms both for CAT games and for more general cases. This section aims to give a taste of this type of analysis. The white-box analysis, together with the black-box analysis that is to be presented in the next section, inspired the work of the grey-box approach to automated auction mechanism design, which will be introduced in Chapter 6 and is the major piece of this dissertation.

5.3 A black-box analysis of CAT 2007 entries

The above white-box analysis is feasible only when the internal structure of a mechanism is known and can only be conducted in a very limited number of scenarios because this analysis requires a thorough, manual examination of game dynamics. A black-box analysis abstracts away the internal structure of a mechanism and many details of the dynamics during the interaction of strategies, making it possible to consider many more scenarios. However, it may still involve high complexity. This is due to the fact that a game may have an arbitrary number of players and an arbitrary number of strategies. The results of n-player, m-strategy games may not necessarily agree with the results of $(n+1)$-player, m-strategy games, or n-player, $(m+1)$-strategy games. For instance, player A beating player B in a bilateral game does not necessarily imply that A would still beat B when an additional player C is added, no matter whether C uses either of the strategies used by A and B, or a third, new strategy. This difficulty suggests, for example, that the replicator dynamics fields reported in [Phelps et al., 2006] based on 6-agent auction games or in [Jordan et al., 2007] based on 6-agent TAC SCM games may possibly change when a different set of game profiles are used to approximate the interaction of a player population with a certain composition of strategies.

To shed more light on the interaction of possible scenarios and limit the possible distortion brought by the sampled game profiles, we ran two sets of experiments to analyze entries to CAT 2007: multi-lateral simulations with games involving all the entries and bilateral simulations with games each involving two specialists. The two sets of experiments can be viewed as the two ends of a spectrum along which the number of players and strategies in a game varies.

The simulations were inspired by the experiments conducted by Axelrod [Axelrod, 2006] and Rust *et al.* [Rust et al., 1993], in which more copies of successful strategies, and less copies of unsuccessful strategies were run for each successive game. Following Axelrod, the simulations that are reported here are called *ecological* simulations.

In the multi-lateral games, we could run a large population of specialists in each game and adjust the presence of different types of specialist in the population in the ecological way. However, constrained by the number of specialists that we could practically have in a single CAT game, we instead modified each strategy's playing time in proportion to its score.[58] That is, in a game that included all specialists, we decreased the number of trading days for less successful strategies, and increased the days for more successful strategies. Figure 5.3a shows the result of this simulation. The distribution on the y-axis gives the proportion of the total number of trading days for all specialists that are allotted to each specialist. Figure 5.3a demonstrates that

[58]Our experiments ran on a Linux cluster where a single run of a game is confined to a single host. That is, all the specialists and traders have to share the resources of the single host. Each specialist or trader runs as a thread, each thread requires a certain mount of memory, and we found that the total number of threads that can run smoothly without crashing the machine was about 750. As we typically run 15 or more traders per specialist in a game to obtain reliable and stable results, 750 traders means that we can run at most 50 specialists in a game, which is not sufficient for an ecological experiment that involves eight different types of specialist. Therefore, instead of running a large population of specialists and adjusting the presence of different types of specialist in the population, we ran games that always involve exactly eight specialists, one for each type, and we limited the number of days on which a specialist is available for traders to choose. The number of days is proportional to the share of the type of specialist in the population.

Table 5.2: The payoff matrix of bilateral CAT games between CAT 2007 entries. Entry (i, j) is the payoff of specialist i in the game against specialist j and entry (j, i) is the payoff of specialist j in the same game.

Specialist	IAM	PSU	jack	Croc	MANX	Tac	Pers	Mert
IAMwildCAT	0.6568	0.7207	0.6793	0.7681	0.7070	0.8008	0.6145	0.7632
PSUCAT	0.5687	0.6152	0.5534	0.6950	0.6121	0.6420	0.7409	0.8307
jackaroo	0.5926	0.6989	0.6279	0.7537	0.7088	0.7839	0.6902	0.8602
CrocodileAgent	0.5245	0.5420	0.5145	0.4865	0.4614	0.6210	0.5879	0.7257
MANX	0.5930	0.6067	0.5790	0.5101	0.6434	0.7150	0.6166	0.6944
TacTex	0.4123	0.5743	0.4344	0.6271	0.5369	0.5546	0.6126	0.7238
PersianCat	0.6200	0.5155	0.5925	0.7041	0.6686	0.6399	0.6446	0.7710
Mertacor	0.3831	0.2947	0.3172	0.5068	0.4026	0.4479	0.4650	0.5503

- the results of this analysis agree with the results we obtained in the white-box analysis, again confirming that IAMwildCAT was the strongest entry in the 2007 competition; and

- the days allotted to PersianCat shrink more slowly than those allotted to other losing specialists. This agrees with the results of bilateral games between IAMwildCAT and PersianCat (described below) and suggests that PersianCat was a strong entry, stronger than its overall position suggests.

One-on-one games closely examine the strength and weakness of a specialist when it faces different opponents. As a result, we ran 64 experiments in total between the eight specialists. Eight of these are self-play games. Table 5.2 shows the resulting payoffs of specialists—their average daily scores—in these CAT games, averaged over ten runs. Entry (i, j) is the payoff of specialist i in the game against specialist j and entry (j, i) is the payoff of specialist j in the same game. IAMwildCAT, the CAT 2007 champion, surprisingly loses, albeit narrowly, against PersianCat, which placed sixth in the competition. This provides an explanation for the fact that in Figure 5.3a the days for PersianCat shrink more slowly than those for other specialists—it does well against the increasingly dominant `IAMwildCAT`. `IAMwildCAT`'s loss, given the defeat of `PersianCat` by `PSUCAT` and `jackaroo`, suggests that `IAMwildCAT` has some particular weakness that happened to be taken advantage of by `PersianCat`.

A close look at the bilateral game between `IAMwildCAT` and `PersianCat` reveals that `PersianCat` charges 100% fee on profit and no other kinds of fees all the way through the game. Although this scares away intra-marginal traders gradually, it helps to attract extra-marginal ones, which do not usually make profit in either marketplace and would have to pay registration fee if they stay with `IAMwildCAT`. As a result, `PersianCat` is able to maintain a market share comparable to `IAMwildCAT`. Meanwhile, `IAMwildCAT` does not charge anything during the beginning phase of a game and charges only modestly for most of the time afterwards—less than a 1% fee on profits. In a bilateral game, `PersianCat` can exploit this. Even though its high fees will tend to drive away traders, it will still attract enough traders to make a decent share of the total profits. In contrast, the effect of the greediness of `PersianCat` is diluted in the more crowded multi-lateral games,[59] where some of the traders who do not choose `IAMwildCAT` choose marketplaces other than `PersianCat`.

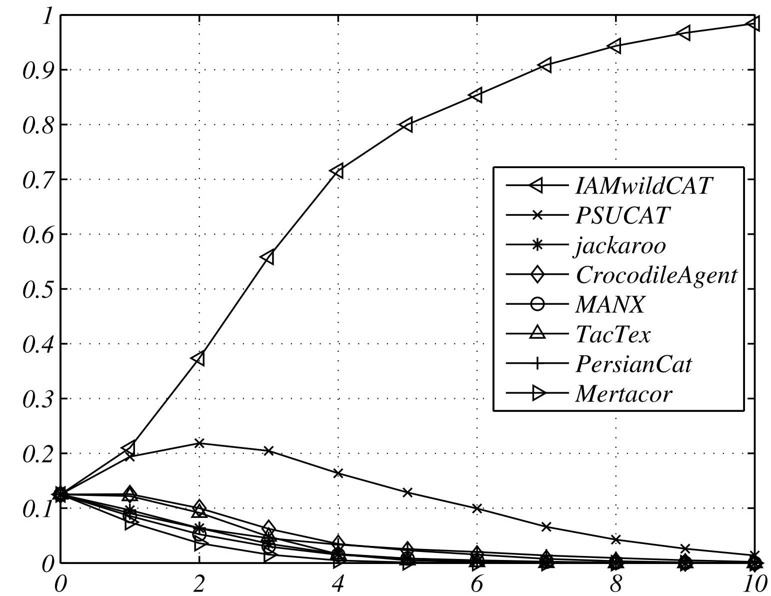
The payoff table for the bilateral CAT games can be used to approximate ecological dynamics for populations involving more than two specialist types. The payoff of each specialist type for a certain population mixture is computed as the expected payoff for this specialist assuming that each specialist obtains the payoff it would have obtained had it competed one-on-one with each of the other specialists in the mix. Under this assumption, Figure 5.3b shows how a population that starts with an even distribution of specialists evolves over time when, as in [Axelrod, 2006], every specialist plays against every other specialist in every generation in bilateral games, and the number of specialists in any generation is proportional to the payoff achieved by that "breed" of

[59] In the CAT 2008 tournament, `PersianCat` charged at a much lower level, adopted a better shout improvement rule based on the one presented in [Niu et al., 2006], and won the game.

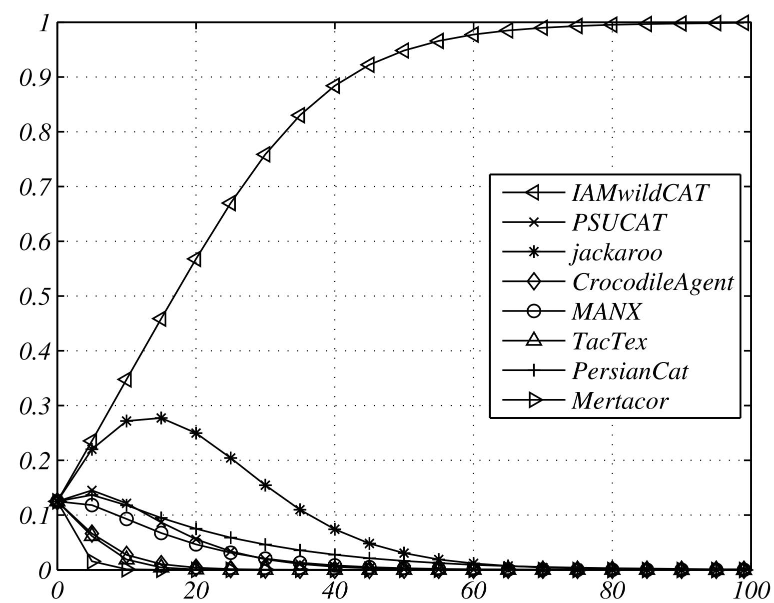
specialist in the previous generation.

Comparing Figure 5.3b with Figure 5.3a, shows that while the winning strategies are the same, the ecological simulations based on multi-lateral games converge much faster than those based on bilateral games (the scales on the x-axis are very different in the two plots). This may be explained by the fact that bilateral games give strategies a chance to benefit from the ability to perform well against specific opponents, whereas in the multi-lateral games they have to be good against all opponents in order to survive. Another noticeable phenomenon is that `PSUCAT` performs much worse in the simulations with bilateral games than those with multi-lateral games, while `jackaroo` and `IAMwildCAT` do the opposite. These discrepancies indicate that, as one might expect, different game setups may lead to very different results. However, our results may be helpful to identify the weakness in strategies by looking at the particular scenario in which a strategy performs poorly.

Similar to the previous section, this section aims to describe briefly a piece of work that inspired the work of the grey-box approach to automated auction mechanism design that will be introduced in the next chapter. The ecological analysis of auction mechanisms in particular leads to the handling of components of auction mechanisms in an ecological manner in the grey-box approach.



(a) Based on multi-lateral CAT games and over 10 generations.



(b) Based on bilateral CAT games and over 100 generations.

Figure 5.3: The distribution of specialists over generations in the ecological simulations of CAT 2007 entries, starting from an event distribution. Note that the scales on the x-axis are very different in the two plots. The population of specialists in the multi-lateral simulation (a) becomes a homogeneous `IAMwildCAT` one after evolving for 10 generations,[a] so the simulation stops, while in the bilateral simulation (b) the population will never become a homogeneous one as the population is assumed to be infinitely large and there are no truncation issues.

[a]The scores of other types of specialist are too low for them to participate in the game even for a single day.

Chapter 6

Automated auction mechanism acquisition through CAT games

The white-box analysis and the black-box analysis in the previous chapter make a good combination for examining the strengths and weaknesses of entries in the CAT 2007 games. Both approaches have advantages and disadvantages. The white-box approach is capable of relating the internal design of an auction mechanism to its performance and revealing which part of the design may cause vulnerabilities, but it requires knowledge of the internal structure of the mechanism and involves manual examination. The black-box does not rely upon the accessibility of the internal design of a mechanism. It can be applied to virtually any strategic game, and is capable of evaluating a design in many more situations, e.g., facing different sets of opponents. However, the black-box approach tells us little about what may have caused a strategy to perform poorly and provides little in the way of hints as to how to improve the strategy. It is desirable to combine these two approaches in order to benefit from the advantages of both. Following the GA-based approach to trading strategy acquisition and auction mechanism design in [Cliff, 2003; Phelps et al., 2006, 2003], we propose what we call a *grey-box* approach to automated mechanism design that combines the white-box approach and the black-box approach, and solves the problem of automatically creating a complex mechanism by searching a structured space of auction mechanisms extending the parameterized framework in [Wurman et al., 2001].

6.1 A grey-box approach to automated auction mechanism design

In the white-box analysis of the CAT entries, we look inside the auction mechanisms and try to relate each component of a mechanism to a specific part of the behavior of the mechanism. In the black-box approach, we do not care about the internals of the mechanism and instead just look at the conditions (in terms of its competitors in the game). To augment this work, we propose a *grey-box* approach in which we concentrate on the components of the mechanisms (as in the white-box approach), but take a black-box view of the components, evaluating their effectiveness by looking at their performance against that of their peers.

More specifically, we view a market mechanism as a combination of auction rules, each as an atomic building block. We consider the problem: *how can we find a combination of rules that is better than any known combination according to a certain criterion, based on a pool of existing building blocks?* The black-box analysis in the previous chapter maintains a population of auction mechanisms and evolves them generation by generation based on their fitnesses. Here we intend to follow a similar approach. In the grey-box analysis, we maintain a population of building blocks, associate each block with a *quality score*, which reflects the fitnesses of auction mechanisms using this block, explore more in the part of the space of auction mechanisms that involves building blocks of higher quality, and keep the best mechanisms we explored through the process.

The following sections discuss issues involved in this method in more detail.

6.2 A search space of double auctions

The first two issues we need to address are *what composite structure is used to represent auction mechanisms?* and *where can we obtain a pool of building blocks?*

Viewing an auction as a structured mechanism is not a new idea. Wurman *et al.* [Wurman et al., 2001] introduced a conceptual, parameterized view of auction mechanisms, which was described in Section 3.3.1. Our prior work on analyzing CAT entries [Niu et al., 2008b] extended this framework for auction mechanisms competing in CAT games and provided a classification of entries in the first CAT competition that was based upon it. The extended framework includes multiple intertwined components, or *policies*, each regulating one aspect of a marketplace.[60] We adopt this framework, include more candidates for each type of policy, and take into consideration parameters that are used by these policies.

These policies were either inferred from the literature, e.g. [McCabe et al., 1993; Wurman et al., 1998], or from our previous work [Niu et al., 2008a,b, 2006; Niu and Parsons, 2011], or contributed by entrants to the CAT competitions. These policies, each as a building block, form a solid foundation for the grey-box approach.

Figure 6.1 illustrates the building blocks as a tree structure. We describe these building blocks in detail below and discuss how we search the space based on the tree structure in the next section.

[60]In the context of CAT games, there are different types of decision making. We deliberately distinguish them by using different terms, which though may have similar meanings or each alone be interpreted in different ways in everyday English. We say that trading agents in a market use trading *strategies* to make an offer and market selection *strategies* to choose a marketplace; we refer to overall forms of auction as *mechanisms*; and we call individual rules in a mechanism—essentially those we enumerate in Section 6.2—*building blocks*, *components*, or *policies*.

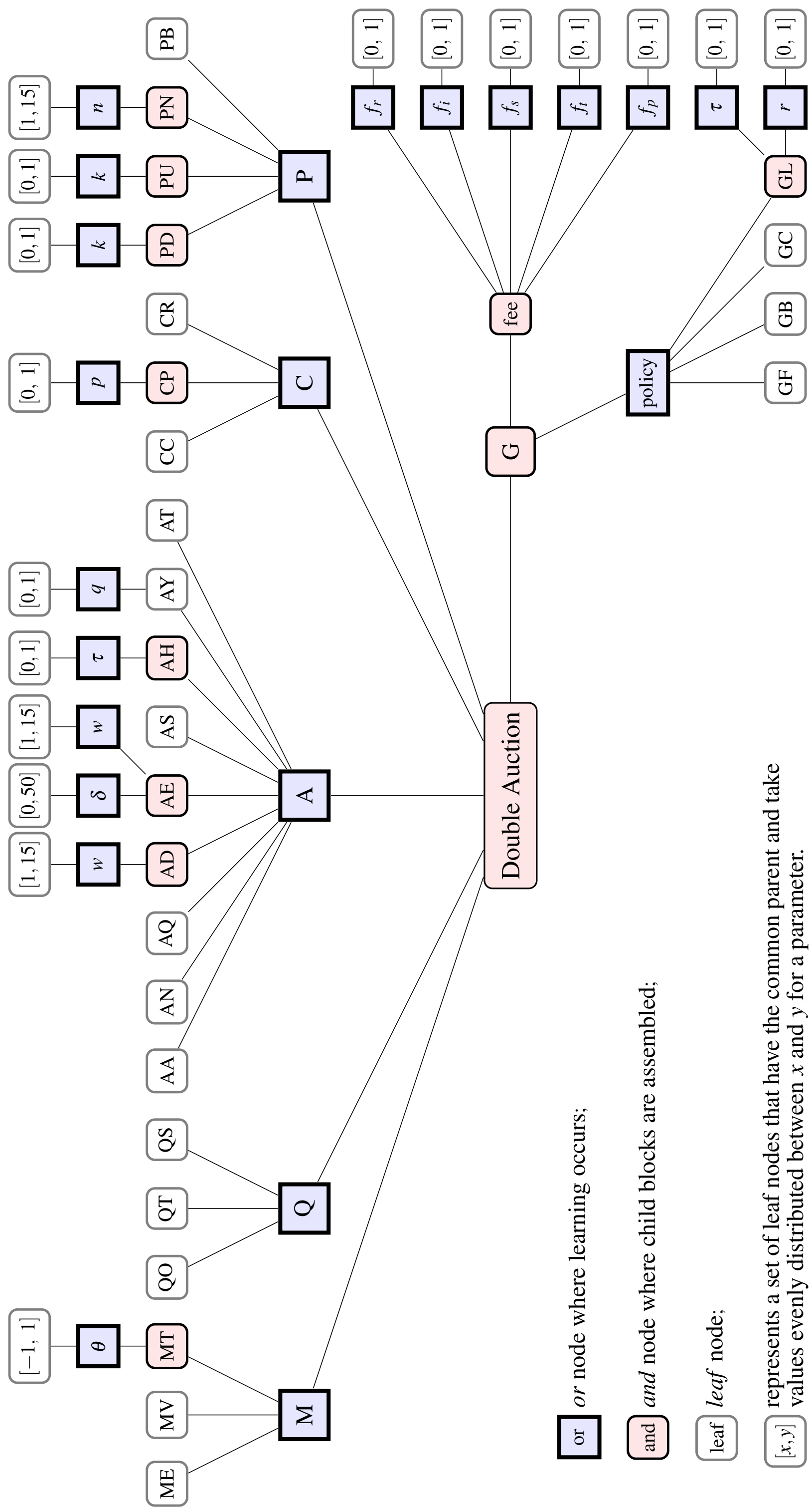


Figure 6.1: The search space of double auctions modeled as a tree, discussed in detail in Sections 6.2 and 6.3.

6.2.1 Matching policy

Matching policies, denoted as $\boxed{\text{M}}$ in Figure 6.1, define how a market matches shouts made by traders.

Equilibrium matching (ME) is the most commonly used matching policy [McCabe et al., 1993; Wurman et al., 1998]. It clears the market at the *reported* equilibrium price and matches intra-marginal asks with intra-marginal bids.

Max-volume matching (MV) [Niu and Parsons, 2011] aims to increase transaction volume based on the observation that a high intra-marginal bid can match with a lower extra-marginal ask, though with a profit loss for the buyer. It does so to realize the maximal transaction volume that is possible.

A generic, parameterized, matching policy can be defined to include ME and MV as two special cases. This policy, denoted as MT [Niu and Parsons, 2011], uses a parameter, θ, which can be any value in $[-1,1]$. When θ is -1, MT does not match any shout; when θ is 0, MT becomes ME; and when θ is 1, MT becomes MV. For any other values of θ, MT tries to realize a transaction volume that is proportional to 0 and those realized in ME and MV.

6.2.2 Quote policy

Quote policies, denoted as $\boxed{\text{Q}}$ in Figure 6.1, determine the quotes issued by markets. Typical quotes are ask and bid quotes, which respectively specify the upper bound for asks and the lower bound for bids that may be placed in a quote-driven market.

Two-sided quoting[61] (QT) defines the ask quote as the minimum of the lowest tentatively matchable bid and lowest unmatchable ask and defines the bid quote as the the maximum of the highest tentatively matchable ask and highest unmatchable bid.

[61] The name follows [McCabe et al., 1993] since either quote depends on information on both the ask side and the bid side.

One-sided quoting (QO) is similar to QT, but considers only the standing shouts closest to the reported equilibrium price from the unmatchable side. When the market is cleared continuously (see below), QO is identical to QT, but otherwise forms a possibly looser restriction on placing shouts.

Spread-based quoting (QS) extends QT to maintain a higher ask quote and a lower bid quote for use with MV. With QS, when the ask quote is lower than the bid quote, the former is set somewhere above their average and the latter below the average, and the spread between the two is a fixed value. QS helps relax the constraint put on shouts with too low an ask quote and too high a bid quote.

6.2.3 Shout accepting policy

Shout accepting policies, denoted as $\boxed{\text{A}}$ in Figure 6.1, judge whether a shout made by a trader should be permitted in the market.

Always accepting (AA) accepts any shout, and *never accepting* (AN) does the opposite.

Quote-beating accepting (AQ) allows only those shouts that are more competitive than the corresponding market quote. This has been commonly used in both experimental settings and real stock markets, and is exactly the NYSE shout improvement rule that was mentioned in Section 2.3.

Self-beating accepting (AS) accepts all first-time shouts but only allows a trader to modify its standing shout with a more competitive price.

Equilibrium-beating accepting (AE) learns an estimate of the equilibrium price based on the past transaction prices in a sliding window, and requires bids to be higher than the estimate and asks to be lower. AE uses a parameter, w, to specify the size of the sliding window in terms of the number of transactions, and a second parameter, δ, which can be added to the estimate to relax the restriction on shouts. This policy was suggested in [Niu et al., 2006] and found to be effective in reducing transaction price fluctuation and increasing allocative efficiency in markets populated by ZI-C traders.

A variant of AE, denoted as AD and introduced by the PSUCAT team in the first CAT competition, uses the standard deviation of transaction prices in the sliding window rather than a constant δ to relax the restriction on shouts.

History-based accepting (AH) is derived from the GD trading strategy [Gjerstad and Dickhaut, 1998] and we found it to be a crucial component of one particular strong market mechanism for CAT games [Niu et al., 2008a]. GD computes how likely a given offer is to be matched, based on the history of previous shouts, and AH uses this same probability computation to accept only shouts that will be matched with probability no lower than a specified threshold, $\tau \in [0, 1]$.

Transaction-based accepting (AT) tracks the most recently matched asks and bids, and uses the lowest matched bid and the highest matched ask to restrict the shouts to be accepted. In a CH, the two bounds are expected to be close to the estimate of equilibrium price in AE, while in a CDA, AT may produce much looser restriction since extra-marginal shouts may steal a deal.

Shout type-based accepting (AY) allows shouts based merely on their types, i.e. asks or bids. This mimics the continuum of auctions presented in [Cliff, 2001a], including retailer markets where only sellers shout, procurement auctions where only buyers shout, as well as general double auctions.

6.2.4 Clearing condition

Clearing conditions, denoted as $\boxed{\text{C}}$ in Figure 6.1, define when to clear the market and execute transactions between matched asks and bids.

Continuous clearing (CC) attempts to clear the market whenever a new shout is placed.

Round clearing (CR) clears the market after all traders have submitted their shouts. This was the original clearing policy in NYSE, but was replaced by CC later for faster transactions and higher volumes. With CC, an extra-marginal trader may have more chance to steal a deal and get matched.

Probabilistic clearing (CP) clears the market with a predefined probability, p, whenever a shout is placed. It thus defines a continuum of clearing rules with CR ($p = 0$) and CC ($p = 1$) being the two ends.

6.2.5 Pricing policy

Pricing policies, denoted as $\boxed{\text{P}}$ in Figure 6.1, set transaction prices for matched ask-bid pairs. The decision making may involve only the prices of the matched ask and bid, or more information including market quotes.

Discriminatory k-pricing (PD) sets the transaction price of a matched ask-bid pair at some point in the interval between their prices. The parameter $k \in [0, 1]$ controls which point is used and usually takes value 0.5 to avoid a bias in favor of buyers or sellers.

Uniform k-pricing (PU) is similar to PD, but sets the transaction prices for all matched ask-bid pairs at the same point between the ask quote and the bid quote. A transaction price set by PU may or may not fall into the range between the matched ask and bid, depending upon the matching policy and the quote policy in the auction mechanism. When it falls outside, whichever of the ask and the bid is closer to the computed transaction price will be used as the final transaction price.

n-pricing (PN) is a pricing policy that we introduced in [Niu et al., 2006] to set the transaction price as the average of the latest n pairs of matched asks and bids. If the average falls out of the price interval between the ask and bid to be matched, the nearest end of the interval is used. We found that this policy can help reduce transaction price fluctuation and has little impact on allocative efficiency.

Side-biased pricing (PB) is basically PD with an internal k dynamically adjusted so as to split the profit in favor of the side on which fewer shouts exist. Thus the more that asks outnumber bids in the current market, the closer k is set to 0.

6.2.6 Charging policy

Charging policies, denoted as $\boxed{\mathrm{G}}$ in Figure 6.1, determine the charges imposed by a market. This is typically not an issue in research on auctions in isolation, but would affect the selection of marketplaces by traders directly in an environment of multiple competing marketplaces, each associated with an auction mechanism, as in CAT games. The four charging policies introduced in Section 4.2.1—*Fixed charging* (GF), *Charge-cutting charging* (GC), *Bait-and-switch charging* (GB), and *Learn-or-lure-fast charging* (GL)—are included in the parameterized framework here.

All these charging policies require an initial set of fees on different activities, including fee for registration, fee for information, fee per shout, fee per transaction, and fee on profit, denoted as f_r, f_i, f_s, f_t, and f_p respectively in Figure 6.1.

GL needs additional parameters in adapting its charges. It employs a simple exploring monitor component, to determine whether traders are still exploring so as to find a good marketplace to trade in. The exploring monitor component monitors the daily distribution of traders among marketplaces and uses the degree of flatness of the distribution—the standard deviation of the distribution relative to the mean of the distribution—as an indicator for the degree of exploration by traders. When the degree of exploration is higher than a threshold, $\tau \in [0,1]$, GL lowers the charges to lure traders, or otherwise it learns from the charges imposed by the most profitable marketplace. GL also uses a learning rate parameter, $r \in [0,1]$, to control how fast the marketplace adapts its charges, which is identical to the way ZIP adapts trading prices.

6.2.7 A tree model

The tree model of double auctions in Figure 6.1 illustrates how building blocks are selected and assembled level by level. There are and nodes, or nodes, and leaf nodes in the tree. An and node, rounded and filled, combines a set of building blocks, each represented by one of its child nodes, to form a compound building block. The root node, for example, is an and node to assemble policies, one on each aspect described above, to obtain a complete auction mechanism. An or node, rectangular and filled, represents the decision making of selecting a building block from the candidates represented by the child nodes of the or node based on their quality scores. This selection occurs not only for those major aspects of an auction mechanism, i.e. M, Q, A, P, C, and G (at G's child node 'policy' in fact), but also for minor components, for example, a learning component for an adaptive policy (in a similar way to that in which Phelps *et al.* learnt a trading strategy [Phelps et al., 2006]), and for determining optimal values of parameters in a policy, like θ in MT and k in PD. A leaf node represents an atomic block that can either be for selection at its or parent node or be further assembled into a bigger block by its and parent node. A special type of leaf node in Figure 6.1 is that with a label in the format of $[x,y]$. Such a $[x,y]$ node is a convenient representation of a set of leaf nodes that have a common parent—the parent of this special leaf node—and take values evenly distributed between x and y for the parameter labeled at the parent node.

6.3 The GREY-BOX-AMD algorithm

This section presents the grey-box algorithm for automated acquisition of auction mechanisms for CAT games. The grey-box algorithm combines techniques from reinforcement learning, e.g., solutions to n-armed bandit problem [Sutton and Barto, 1998], and evolutionary computation, e.g., the use of a Hall of Fame [Rosin, 1997]. The general idea of this algorithm is to use n-armed bandit learners to choose building blocks when needed so as to construct auction mechanisms based on the tree model in Figure 6.1, to run CAT games to evaluate the constructed mechanisms, and to keep good mechanisms in a Hall of Fame.

In the tree model, $\boxed{\text{or}}$ nodes contribute to the variety of auction mechanisms in the search space and are where exploitation and exploration occur. We model each $\boxed{\text{or}}$ node as an n-armed bandit learner that chooses among candidate blocks, and we use the simple softmax method to solve this learning problem.[62] Solving all the n-armed bandit learners in the tree will uniquely determine a configuration of an auction mechanism, which is exactly how an auction mechanism is sampled in the search space. The sampled mechanisms can then be put into a CAT game for evaluation. The game score of a sampled mechanism not only suggests how good the mechanism itself is, but is also an indicator of the performance of the building blocks that are used in the mechanism. If a building block is due to the selection of an n-armed bandit learner among the child nodes of the corresponding $\boxed{\text{or}}$ node, the game score can be readily used as the feedback for the building block. All such feedback to a building block cumulatively serves as the expected return, or what we call the *quality score*, of the building block. Thus, after a game completes, the quality scores of building blocks that are children of an $\boxed{\text{or}}$ node are updated, and so are the way how an auction mechanism is sampled in the space in later steps.

Given a set of building blocks, $\mathbb{B}$, that are in the form of the tree model, and a set of fixed market mechanisms, $\mathbb{FM}$, as opponents for sampled mechanisms to beat in CAT games, we define the skeleton of the grey-box algorithm in Algorithm 1, denoted as the GREY-BOX-AMD algorithm, which we describe in detail as follows.

[62] The same solution was adopted in designing market selection strategies for trading agents in CAT games, which was described in Section 4.1.2. However the two scenarios may need different parameter values. The market selection scenario should favor choices that give a good profit—a cumulative measure—while here we require effective exploration to find a good mechanism in the foreseeable future—a one-time concern.

Algorithm 1: The GREY-BOX-AMD algorithm.

```
Input: 𝔹, 𝔽𝕄
Output: ℍ𝕆𝔽
begin
    ℍ𝕆𝔽 ← ∅
    for s ← 1 to NUM_OF_STEPS do
        G ← Create-Game()
        𝕊𝕄 ← ∅
        for m ← 1 to NUM_OF_SAMPLES do
            M ← Create-Market()
            for t ← 1 to NUM_OF_POLICYTYPES do
                B ← Select(𝔹_t, 1)
                Add-Block(M, B)
            𝕊𝕄 ← 𝕊𝕄 ∪ {M}
        𝔼𝕄 ← Select(ℍ𝕆𝔽, NUM_OF_HOF_SAMPLES)
        Run-Game(G, 𝔽𝕄 ∪ 𝔼𝕄 ∪ 𝕊𝕄)
        foreach M ∈ 𝔼𝕄 ∪ 𝕊𝕄 do
            Update-Market-Score(M, Score(G, M))
            if M ∉ ℍ𝕆𝔽 then
                ℍ𝕆𝔽 ← ℍ𝕆𝔽 ∪ {M}
            if CAPACITY_OF_HOF < |ℍ𝕆𝔽| then
                ℍ𝕆𝔽 ← ℍ𝕆𝔽 − {Worst-Market(ℍ𝕆𝔽)}
            foreach B used by M do
                Update-Block-Score(B, Score(G, M))
```

The GREY-BOX-AMD algorithm runs a certain number of steps. At each step, a single CAT game is created and a set of market mechanisms are prepared for the game. This set of market mechanisms includes all market mechanisms in $\mathbb{FM}$, a certain number of market mechanisms sampled from the search space, denoted as $\mathbb{SM}$, and a certain number of market mechanisms, denoted as $\mathbb{EM}$, chosen from a Hall of Fame, $\mathbb{HOF}$. All these market mechanisms, each run by a specialist in its marketplace, are put into the game, which evaluates the performance of these market mechanisms. The $\mathbb{HOF}$ has a fixed capacity, and maintains market mechanisms that performed well in games at previous steps in terms of their average scores across games in which they participated.

The $\mathbb{HOF}$ is empty initially, updated after each game, and returned in the end as the output of the GREY-BOX-AMD algorithm.

Each market mechanism in $\mathbb{SM}$ is constructed based on the tree model in Figure 6.1. After an 'empty' market mechanism, M, is created, building blocks can be incorporated into M. There are a certain number of different policy types, and from each group of policies of the same type, denoted as $\mathbb{B}_t$ where t specifies the type, a building block is chosen for M. For simplicity, this algorithm illustrates only what happens to the $\boxed{\text{or}}$ nodes at the high level, including `M`, `Q`, `A`, `C`, and `P`. Market mechanisms in $\mathbb{EM}$ are chosen from the $\mathbb{HOF}$ in a similar way.

After a CAT game, G, completes at each step, the game score of each participating market mechanism $M \in \mathbb{SM} \cup \mathbb{EM}$, `Score(`$G$, M`)`, is recorded and the game-independent score of M, `Score(`M`)`, is updated. If M is not currently in the $\mathbb{HOF}$ and `Score(`M`)` is higher than the lowest score of market mechanisms in the $\mathbb{HOF}$, it replaces that corresponding market mechanism.

`Score(`G, M`)` is also used to update the quality score of each building block used by M. Both `Update-Market-Score()` and `Update-Block-Score()` in Algorithm 1 calculate respectively game-independent scores of market mechanisms and quality scores of building blocks by averaging feedback `Score(`G, M`)` over time. Because choosing building blocks occurs only at $\boxed{\text{or}}$ nodes in the tree, only child nodes of an $\boxed{\text{or}}$ node have quality scores and receive feedback after a CAT game. Initially, quality scores of building blocks are all 0, so that the probabilities of choosing them are even. As the exploration proceeds, fitter blocks score higher and are chosen more often to construct better mechanisms.

6.4 Experiment Set I: Learning against classic double auction mechanisms

We carried out two sets of experiments to acquire auction mechanisms using the grey-box approach. The first set of experiments search the space of auction mechanisms presented above for mechanisms that are competitive in CAT competitions. This work was reported briefly in [Niu et al., 2010a] and with more detail in [Niu et al., 2010b].

6.4.1 Experimental setup

We extended JCAT with the parameterized framework of double auctions and all the individual policies described in Section 6.2. To reduce the computational cost, we eliminated the exploration of charging policies by focusing on mechanisms that impose a fixed charge of 10% on trader profit, which we denote as $\mathsf{GF}_{0.1}$. Analysis of CAT games [Niu et al., 2008a] and what entries have typically charged in actual CAT competitions, especially in the latest three events, suggest that such a charging policy is a reasonable choice to avoid losing either intra-marginal or extra-marginal traders. Even with this cut-off, the search space still contains more than $1,200,000$ different kinds of auction mechanisms, due to the variety of policies for aspects other than charging and the choices of values for parameters.

The experiments that we ran to search the space each last 200 steps. At each step, we sample two auction mechanisms from the space, and run a CAT game to evaluate them against four fixed, well known, mechanisms plus two mechanisms that performed well at previous steps and are members of the Hall of Fame. The scores of the sampled and Hall of Fame mechanisms are used as feedback for every building block that an individual mechanism uses and is associated with a quality score.

To sample auction mechanisms, the softmax exploration method used by $\boxed{\text{or}}$ nodes starts with a relatively high temperature ($\tau = 10$) so as to explore randomly, then gradually cools down, τ scaling down by 0.96 (α) each step, and eventually maintains a temperature ($\tau = 0.5$) that guarantees a non-negligible probability of choosing even the worst action any time.[63] After all, our goal in the grey-box approach is not to converge quickly to a small set of mechanisms, but to explore the space as broadly as possible and avoid being trapped in local optima.

The fixed set of four auction mechanisms in every CAT game includes two CHs—CH_l and CH_h—and two CDAs—CDA_l and CDA_h—with one of each charging 10% on trader profit, like $\text{GF}_{0.1}$ does, and the other charging 100% on trader profit (denoted as $\text{GF}_{1.0}$). The CH and CDA mechanisms are two common double auctions and have been used in the real world for many years, in financial markets in particular due to their high allocative efficiency. Earlier experiments we ran, involving CH and CDA mechanisms against entries from CAT competitions, indicate that it is not trivial to win over these two standard double auctions. Auction mechanisms with different charge levels are included to avoid any sampled mechanisms taking advantage otherwise. Based on the parameterized framework in Section 6.2, the CH and CDA mechanisms can be represented as follows:

$$\text{CH}_l = \text{ME} + \text{QT} + \text{AQ} + \text{CR} + \text{PU}_{k=0.5} + \text{GF}_{0.1}$$

$$\text{CH}_h = \text{ME} + \text{QT} + \text{AQ} + \text{CR} + \text{PU}_{k=0.5} + \text{GF}_{1.0}$$

$$\text{CDA}_l = \text{ME} + \text{QT} + \text{AQ} + \text{CC} + \text{PD}_{k=0.5} + \text{GF}_{0.1}$$

$$\text{CDA}_h = \text{ME} + \text{QT} + \text{AQ} + \text{CC} + \text{PD}_{k=0.5} + \text{GF}_{1.0}$$

The Hall of Fame that we maintain during the search contains up to ten members. After each CAT game, the two sampled mechanisms are compared with those Hall of Famers. If the score of a sampled mechanism is higher than the lowest average score of the Hall of Famers, the sampled mechanism is inducted into the Hall of Fame and replaces the corresponding Hall of Famer. The replaced Hall of Famer may be re-inducted if an identical mechanism happens to be sampled from

[63] At around the 75th step, τ reaches 0.5 and remains constant later on.

the space again and scores highly enough to promote its average score to surpass the lowest score of Hall of Famers. In addition, the softmax method used to choose two out of the ten Hall of Famers involves a constant $\tau = 0.3$. Since the scores of the Hall of Famers gradually converge in the experiments and the difference between the best and the worst Hall of Famers is less than 25% (see Figure 6.2b below), this value of τ guarantees that the bias towards the best Hall of Famers is modest and all Hall of Famers have a fairly big chance to be chosen.

Each CAT game is populated by 120 trading agents, using ZI-C, ZIP, RE, and GD strategies, a quarter of the traders using each strategy. Half the traders are buyers, half are sellers. The private values of these traders are drawn from a uniform distribution between \$50 and \$150. Each CAT game lasts 500 days with ten rounds for each day. This setup is similar to that of actual CAT competitions except for a smaller trader population that helps to reduce computational costs. The results that are reported in the next section are averaged over 40 runs of the grey-box experiment. Each run is an execution of the GREY-BOX-AMD algorithm shown in Algorithm 1 and uses a different seed for the random number generator that is used in JCAT and the n-armed bandit learners for selecting building blocks and auction mechanisms. Each run lasts 200 steps with a single CAT game at each step. That is to say that the CAT game is used as the evaluation function for auction mechanisms in these experiments, each invocation of the function evaluates eight mechanisms, and there are 200 invocations totally in a single run of the experiment. The experiments were carried out on a 64-node Linux cluster at the CUNY Graduate Center. A 200-step grey-box experiment takes around 12 hours on a node that typically runs at 2.8GHz and has a 4GB memory. All the 40 experiments lasted about three days due to the limited number of jobs that each user is allowed to run on the cluster.[64]

Table 6.1 summarizes the values of parameters and inputs of Algorithm 1 in our experiments.

[64]The maximum number of tasks that a user is allowed to run at any time on the cluster is 10 when the experiments were carried out.

Table 6.1: The values of parameters and inputs of the GREY-BOX-AMD algorithm in the first set of experiments.

Parameter/Input	**Value**
NUM_OF_STEPS	200
NUM_OF_SAMPLES	2
NUM_OF_HOF_SAMPLES	2
CAPACITY_OF_HOF	10
NUM_OF_POLICYTYPES	5
initial τ_0*	10
minimal τ_0*	0.5
α_0*	0.96
τ_1†	0.3
α_1†	1
$\mathbb{FM}$	$\{\text{CH}_l, \text{CH}_h, \text{CDA}_l, \text{CDA}_h\}$

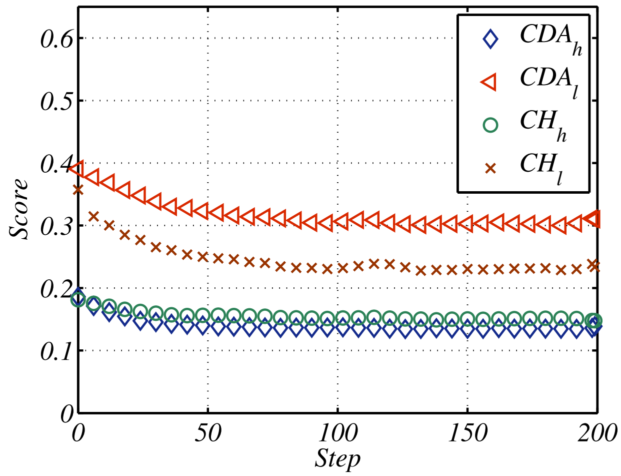
* τ_0 and α_0 are parameters in the softmax solver used by the `Select(`$\mathbb{B}_t$`, 1)` function in Algorithm 1.

† τ_1 and α_1 are parameters in the softmax solver used by the `Select(`$\mathbb{HOF}$`, NUM_OF_HOF_SAMPLES)` function in Algorithm 1.

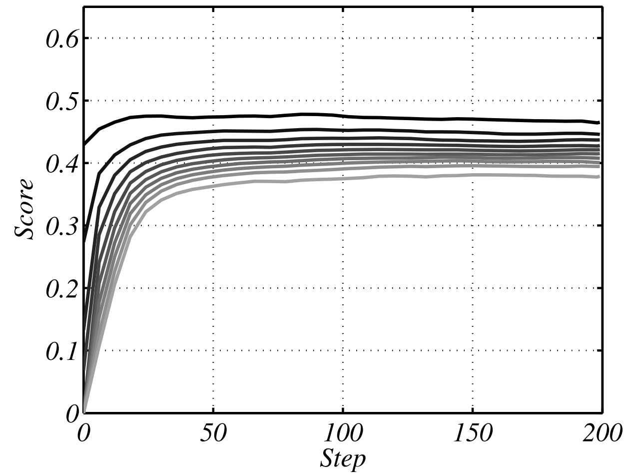
6.4.2 Experimental results

We collected data and checked whether the grey-box approach is successful in searching for good auction mechanisms in four different ways.

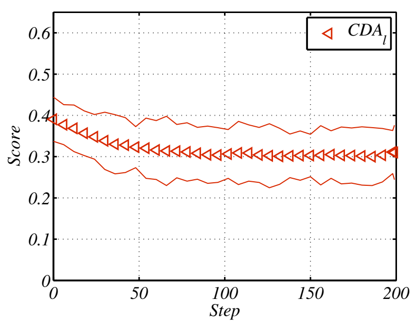
First, we measured the performance of the generated mechanisms indirectly, through their effect on other mechanisms. Since the four standard market mechanisms participate in all the CAT games, their performance over time reflects the strength of their opponents—they will do worse as their opponents get better—which in turn reflects whether the search generates increasingly better mechanisms. Figure 6.2a shows that the scores of the four market mechanisms (more specifically the average daily scores of the market mechanisms in a game) decrease over 200 steps, especially over the first 100 steps, suggesting that the mechanisms we are creating get better as the learning process progresses.



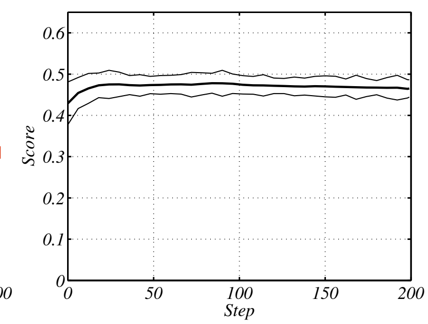
(a) Fixed markets.



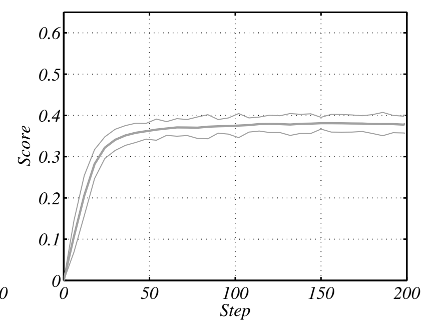
(b) Hall of Famers with the best at the top and the worst at the bottom.



(c) The best fixed market mechanism.



(d) The best Hall of Famer(s).



(e) The worst Hall of Famer(s).

Figure 6.2: Scores of market mechanisms across 200 steps in the first set of grey-box experiments, averaged over 40 runs.

Second, we measured the performance of the set of mechanisms we created more directly. The mechanisms that are in the Hall of Fame at a given point represent the best mechanisms that we know about at that point and their performance tells us more directly how the best mechanisms evolve over time. Figure 6.2b shows the scores of the ten Hall of Famers at each step over the 40 200-step runs.[65] As in Figure 6.2a, the first 100 steps sees a clear, increasing trend. Even the

[65]Note that the Hall of Famers may be different mechanisms at different steps in the process, so, for example, the curve for the best Hall of Famer in the figure may reflect the scores of many different mechanisms, the highest we

Table 6.2: The average daily scores of the best fixed market mechanism and the best and worst Hall of Famers in the CAT games at the end of the first set of grey-box experiments.

Market mechanism	**Mean**	**SD**
Best fixed mechanism (CDA$_l$)	0.3101	0.0659
Best Hall of Famers	0.4652	0.0210
Worst Hall of Famers	0.3790	0.0219

scores of the worst of the ten at the end are above 0.35, higher than the highest of the four fixed market mechanisms from Figure 6.2a. Indeed, Table 6.2 lists respectively the average scores of the best fixed market mechanism, and the best and worst Hall of Famers at the end of the grey-box experiments as well as the standard deviations. At the 95% confidence level, the score of the worst Hall of Famers is significantly higher than that of the best fixed market mechanism, CDA$_l$. Thus we know that our approach will create mechanisms that outperform standard mechanisms, though we should not read too much into this since we trained our new mechanisms directly against them.

It should be noted that in Figure 6.2b and in Figure 6.2d the scores of the top Hall of Famers descend slightly or reach a plateau after around 100 steps. This is due to two reasons. On one hand, these Hall of Famers face stronger and stronger opponents as the grey-box experiments go on and better mechanisms are sampled and put into the games at latter steps—the same reason caused the descending scores of the fixed market mechanisms. On the other hand, as the grey-box experiments go on, no new mechanisms can be found and inducted into the Hall of Fame that are able to produce significantly better performance than those existing Hall of Famers, thus failing to hold the increasing records. The time when the plateau begins and the level where the plateau resides are both quantitative indicators of the effectiveness of the search process, and provide guidance on, for example, how long a grey-box experiment should run to obtain stable results.

know of up to the the point when we collected the data.

A better test of the new mechanisms is to run them against those mechanisms that we know to be strong in the context of CAT games, asking what would have happened if our Hall of Fame members had been entered into prior CAT competitions and had run against the carefully hand-coded entries in those competitions. We chose three Hall of Famers from the ten Hall of Famers obtained in one of the 40 runs. These Hall of Famers are internally labeled as `SM7.1`, `SM88.0`, and `SM127.1` and can be represented in the parameterized framework in Section 6.2 as follows:

$$\texttt{SM7.1} = \mathsf{ME} + \mathsf{QO} + \mathsf{AH}_{\tau=0.4} + \mathsf{CP}_{p=0.3} + \mathsf{PN}_{n=11} + \mathsf{GF}_{0.1}$$

$$\texttt{SM88.0} = \mathsf{ME} + \mathsf{QT} + \mathsf{AA} + \mathsf{CP}_{p=0.4} + \mathsf{PU}_{k=0.7} + \mathsf{GF}_{0.1}$$

$$\texttt{SM127.1} = \mathsf{ME} + \mathsf{QS} + \mathsf{AS} + \mathsf{CP}_{p=0.4} + \mathsf{PU}_{k=0.7} + \mathsf{GF}_{0.1}$$

We ran these three mechanisms against the best recreation of past CAT competitions that we could achieve given the contents of the TAC agent repository,[66] where competitors are asked to upload their entries after the competition. The CAT games were set up in a similar way to the competitions, populated by 500 traders that are evenly split between buyers and sellers and between the four trading strategies—ZI-C, ZIP, RE, and GD—and the private values of sellers or buyers were drawn from a uniform distribution between \$50 and \$150. For the recreated competitions, we ran three games for 2007 and 2008 (like in the actual competitions) and ten games for 2009.[67,68]

Tables 6.3a, 6.3b and 6.12a list the average cumulative scores of all the market mechanisms across the games along with the standard deviations of those scores against entries from CAT 2007, 2008, and 2009 respectively.[69] The three new mechanisms we obtained from the grey-box experiments beat the actual entries from CAT 2007 and CAT 2008 by a comfortable margin in both cases.

[66] `http://www.sics.se/tac/showagents.php`.

[67] It is desirable to run more games for each recreated competition. However some of the entries from prior CAT competitions use a graphical interface, e.g., `MyFuzzy` for CAT 2008 and `IAMwildCAT` and `UMTac` for CAT 2009, which makes it difficult to run games involving these entries repeatedly in an automated manner on our cluster.

[68] When we ran these experiments, CAT 2010 had been held but no entries had been made available in the TAC agent repository so we were unable to recreate the latest competition.

[69] The data for CAT 2009 is placed in a separate table so as to be compared with the data from the second set of grey-box experiments that is to be described later.

Table 6.3: The scores of market mechanisms in the CAT games including the best mechanisms from the first set of grey-box experiments and entries from prior CAT competitions, averaged over three CAT games respectively for 2007 and 2008.

(a) Against CAT 2007 entries.

Mechanism	**Score**	**SD**
SM7.1	199.4500	5.9715
SM88.0	191.1083	10.3186
SM127.1	180.1277	9.0289
MANX	154.6953	1.3252
Croc'Agent	142.0523	9.0867
TacTex	138.4527	5.8224
PSUCAT	133.1347	5.6565
PersianCat	124.3767	11.2409
jackaroo	108.8017	8.6851
IAMwildCAT*	106.8897	4.4006
Mertacor	89.1707	4.9269

(b) Against CAT 2008 entries.

Mechanism	**Score**	**SD**
SM7.1	196.7240	9.2843
SM88.0	186.9247	4.2184
SM127.1	183.5887	9.7835
jackaroo	177.5913	2.5722
Mertacor	161.5440	5.8741
MANX	147.3050	15.7718
IAMwildCAT	142.9167	8.9581
PersianCat	139.1553	17.9783
DOG	130.2197	18.9782
MyFuzzy	125.9630	1.9221
Croc'Agent*	71.4820	5.8687
PSUCAT*	68.3143	6.7389

* IAMwildCAT from CAT 2007, and CrocodileAgent (abbreviated as Croc'Agent in the table) and PSUCAT from CAT 2008 worked abnormally during the games and tried to impose invalid fees, probably due to competition from the three new, strong opponents. Although we modified JCAT to avoid kicking out these entries on those trading days when they impose invalid fees—which JCAT does in an actual CAT tournament—these market mechanisms still perform poorly, in contrast to their rankings in the tournaments.

The fact that we can take mechanisms that we generate in one series of games (against the fixed opponents and other new mechanisms) and have them perform well against a separate set of mechanisms suggests that the grey-box approach learns robust mechanisms. The three new mechanisms failed to win the competition against entries from CAT 2009, but were able to perform better than some of them. The second set of grey-box experiments that is to be described in the next section aims to search for mechanisms that perform well against entries from CAT 2009.

In passing, we note that the rankings of the entries from the repository do not reflect those in the actual CAT competitions. This is to be expected since the entries now face new opponents and different market mechanisms will, in general, respond differently to this. Excluding the market mechanisms that attempt to impose invalid fees and are marked with '*', we can see that the overall performance of entries from the two recent, actual CAT competitions is significantly better than that of those into the competitions in the previous year respectively when they face the three new, strong, opponents, reflecting the improvement in the entries over time. `Mertacor`, which did not win the actual 2009 CAT competition, surprisingly beat all other mechanisms by a huge margin. It is unclear whether this is due to a different, improved version of `Mertacor` uploaded to the TAC agent repository, or some other reason.

Finally, we tested the performance of `SM7.1`, `SM88.0`, and `SM127.1` when they are run in isolation, applying the same kind of test that auction mechanisms are traditionally subject to. We tested the mechanisms both for allocative efficiency and, following our work in [Niu et al., 2006], for the extent to which they trade close to theoretical equilibrium as measured by the coefficient of convergence, α, even when populated by minimally rational traders. In [Niu et al., 2006] we proposed a class of double auctions, called NCDAEE, which can be represented as:

$$\text{NCDAEE} = \mathsf{ME} + \mathsf{AE}_{w,\delta} + \mathsf{CC} + \mathsf{PN}_n$$

The advantage of NCDAEE is that it can give significantly lower α—faster convergence of transaction prices—and higher allocative efficiency (E_a) than a CDA when populated respectively by homogeneous ZI-C traders and can perform comparably to a CDA when populated by homogeneous GD traders.

We replicated these experiments using JCAT and ran additional ones for the three new mechanisms with similar configurations. The results of these experiments are shown in Table 6.4.[70] The best result in each column is shaded. We can see that both `SM7.1` with ZI-C traders and `SM88.0` with GD traders give higher E_a than the best of the existing market mechanisms respectively, and

[70]The results we get there are slightly different from those we reported in [Niu et al., 2006] (in which we used a different platform), but the pattern of these results still holds. In addition, we ran an NCDAEE variant ($\delta = 30$) that was not tested in [Niu et al., 2006], observing that those with $\delta \leq 20$ do not perform well when populated by GD traders.

Table 6.4: Economic properties of the best mechanisms from the first set of grey-box experiments and the auction mechanisms explored in [Niu et al., 2006]. All NCDAEE mechanisms are configured to have $w = 4$ in their AE policies and $n = 4$ in their PN policies. The best result in each column is shaded. Data in the first four rows are averaged over 1,000 runs and those in the last four are averaged over 100 runs.

Mechanism	**ZI-C**				**GD**			
	E_a		α		E_a		α	
	Mean	**SD**	**Mean**	**SD**	**Mean**	**SD**	**Mean**	**SD**
CDA	97.464	3.510	13.376	4.351	99.740	1.553	4.360	3.589
NCDAEE$_{\delta=0}$	98.336	3.262	4.219	3.141	9.756	28.873	14.098	1.800
NCDAEE$_{\delta=10}$	98.912	2.605	5.552	2.770	23.344	41.727	7.834	5.648
NCDAEE$_{\delta=20}$	98.304	2.562	7.460	3.136	89.128	30.867	4.826	3.487
NCDAEE$_{\delta=30}$	97.708	3.136	8.660	3.740	99.736	1.723	4.498	3.502
SM7.1	99.280	1.537	4.325	2.509	58.480	47.983	4.655	4.383
SM88.0	98.320	2.477	11.007	4.251	99.920	0.560	4.387	2.913
SM127.1	97.960	3.225	11.152	4.584	99.520	1.727	4.751	3.153

both of these increases are statistically significant at the 95% level. Both cases also lead to low α, not the lowest in the column but close to the lowest, and the differences between them and the lowest are not statistically significant at the 95% level. Thus the grey-box approach can generate mechanisms that perform as well in the single marketplace case as the best mechanisms from the literature.

6.4.3 Choosing parameter values

The performance of the mechanisms obtained in the grey-box experiments suggests that the parameter values we chose in these experiments, those listed in Table 6.1, are reasonable, however it needs further investigation to determine whether the choices of parameter values play an important role in the grey-box algorithm.

In Table 6.1, $\mathbb{FM}$ and NUM_OF_POLICYTYPES are part of the input. NUM_OF_SAMPLES and NUM_OF_HOF_SAMPLES are set to 2 due to the maximum number of marketplaces that are typically allowed in a CAT game involving 120 traders. Our experience in running CAT games suggested that at least 15 traders per marketplace are needed in a CAT game to obtain statistically reliable results while still incorporating the diversity of traders in terms of their roles as sellers or buyers, their competitiveness based on their private values, and the trading strategies they may adopt.[71] Having more traders in the game will allow us to sample more mechanisms at each step of the grey-box search but it will significantly increase the computational cost. Our test showed that increasing the number of traders in a game will lead to a quadratic or possibly exponential increase of the time cost. The parameters used in softmax solvers—the initial τ_0, the minimal τ_0, α_0, τ_1, and α_1—take values in a way to balance exploitation and exploration in selecting building blocks and Hall of Famers respectively. The values of τ_0 and α_0 make sure that the softmax selection is close to random in the first 20 or so steps, and the temperature stops cooling down after about 40 steps and remains at 0.5 so as to give just a modest advantage to the best choice, the normalized return of which is always set to be 1.0.[72] τ_1 is a constant 0.3 with $\alpha_1 = 1$ so that the selected Hall of Famer is very likely to be one of those whose scores fall into the range between 0.35 and 0.48 (estimated to be normalized to 0.73 and 1.0 respectively) as shown in Figure 6.2b, while those that perform poorly, particularly those that are inducted into the Hall of Fame during the early stage of the grey-box search and score typically below 0.1 (estimated to be normalized to 0.21), will not have much chance to be selected. Finally, CAPACITY_OF_HOF is set to 10 to balance the diversity

[71] This minimum number of traders we observed from our experimentation is approximately in the same range as the number of traders in [Smith, 1962], [Gode and Sunder, 1993a], and [Cliff and Bruten, 1997]—22 (in one case), 12, and 22 respectively.

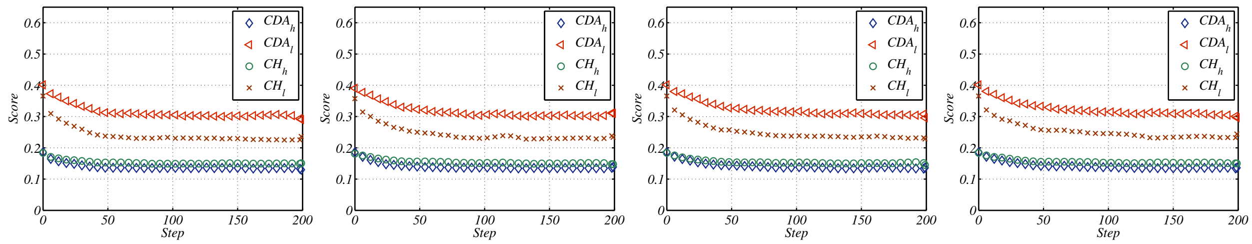
[72] After 20 steps (2 samples each step), the value of τ_0 in the softmax solvers for $\boxed{\text{or}}$ nodes including M, Q, A, P and C in the tree model becomes approximately 2.0 ($10 \times 0.96^{20\times 2}$), still much higher than even the maximum return, which is 1.0 after normalization. That is to say that τ is between 10.0 and 2.0 while $Q(a_i)$ is at most 1.0 in (4.1) on Page 54. τ_0 comes below 0.5 after 37 steps (37 is the solution to $10 \times 0.96^{2x} = 0.5$).

of mechanisms in the Hall of Fame and the number of times a Hall of Famer is typically selected to compete in games so that its average score may give a reliable estimate of its performance.

The parameters could be tweaked. Searching for optimal values for all these parameters is desirable but prohibitive due to the limited computational resources we have access to, as we have to run multiple runs of the grey-box search for each configuration of parameters and these may take years to complete in total. Therefore, we chose to focus on one of the parameters, `CAPACITY_OF_HOF`. We ran additional experiments to check whether the grey-box experiments are sensitive to this parameter and report our analysis in this section.

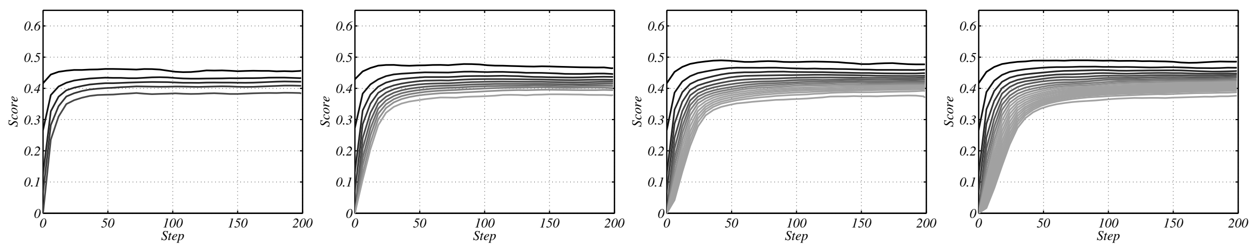
We ran three additional sets of grey-box experiments, each using a different capacity of the Hall of Fame, 5, 15, and 20 respectively in contrast to 10 in the initial set of experiments. The experimental setup remained otherwise the same as before. Illustrated in Figures 6.3 and 6.4 are the scores of the four fixed market mechanisms and the Hall of Famers in these experiments respectively. The results from the initial set of grey-box experiments where $|\mathbb{HOF}| = 10$ are also included for comparison, which are already shown in Figure 6.2. Figure 6.3 suggests that market mechanisms sampled from these different sets of experiments produced similar conditions of competition to the fixed market mechanisms as the performances of these fixed mechanisms follow similar patterns. The first row of Table 6.5 lists the average daily scores of the best fixed market mechanism—CDA$_l$ in all the cases—at the end of these experiments respectively. The differences between these scores are not statistically significant at the 95% confidence level.

The scores of the Hall of Famers indicate a different trend however. Figure 6.4 shows that the top Hall of Famers tend to perform better as the capacity of the Hall of Fame increases. The scores at the end of the experiments with $|\mathbb{HOF}| = 15$ or 20, given in the second row of Table 6.5, are significantly better than that from the experiments with $|\mathbb{HOF}| = 10$, both at the 95% confidence level. This indicates that there is merit in increasing the capacity of the Hall of Fame. A close look



(a) capacity of 5. (b) capacity of 10. (c) capacity of 15. (d) capacity of 20.

Figure 6.3: Scores of the four fixed market mechanisms across 200 steps in different sets of grey-box experiments, each set using a different capacity for the Hall of Fame, averaged over 40 runs. (b) is identical to Figure 6.2a.



(a) capacity of 5. (b) capacity of 10. (c) capacity of 15. (d) capacity of 20.

Figure 6.4: Scores of the Hall of Famers across 200 steps in different sets of grey-box experiments, each set using a different capacity for the Hall of Fame, averaged over 40 runs. (b) is identical to Figure 6.2b.

at the top Hall of Famers and the games they participated in reveals that many more of the top Hall of Famers in the cases of $|\mathbb{HOF}| = 15$ and $|\mathbb{HOF}| = 20$ (26 and 29 out of 40 respectively) compared to the other two cases (11 and 5 out of 40 respectively) are evaluated five times or fewer. On one hand, a larger Hall of Fame means that each individual has fewer chances to be selected during the 200 games each run; on the other hand, a larger Hall of Fame lowers the barrier of induction to weak market mechanisms, and the opponents of these bottom Hall of Famers in a game thus benefit from the latter's weakness, resulting in higher scores of the former. An examination of the top Hall of Famers that were selected and competed only once confirmed this.[73] That is to say that

[73] It is possible that mechanisms identical to these Hall of Famers were actually selected and competed in games

Table 6.5: The average daily scores of the best fixed market mechanism and the best Hall of Famers in the CAT games at the end of the sets of grey-box experiments, each set using a different capacity for the Hall of Fame. In parentheses are the standard deviations. The scores in shaded cells are significantly different, at the 95% confidence level, from their counterpart from the initial set of experiments ($|\mathbb{HOF}| = 10$).

Market mechanism	$\lvert\mathbb{HOF}\rvert = 5$	$\lvert\mathbb{HOF}\rvert = 10$	$\lvert\mathbb{HOF}\rvert = 15$	$\lvert\mathbb{HOF}\rvert = 20$
Best fixed mechanism (CDA_l)	0.2901 (0.0726)	0.3101 (0.0659)	0.2957 (0.0767)	0.2963 (0.0638)
Best Hall of Famers (all)†	0.4572 (0.0227)	0.4652 (0.0210)	0.4772 (0.0238)	0.4860 (0.0202)
Best Hall of Famers (reliable)‡	0.4551 (0.0136)	0.4603 (0.0164)	0.4619 (0.0122)	0.4589 (0.0199)

† Data is calculated based on all the Hall of Famers generated from the experiments.

‡ Data is calculated based on the Hall of Famers that were sampled and evaluated at least five times during the experiments, which gives relatively more *reliable* observations.

the other selected Hall of Famer in the game is much weaker and the two sampled mechanisms from the search space are not strong either. These observations led us to recalculate the scores of the top Hall of Famers in the experiments by excluding those that appeared in five games or fewer and using the scores of the Hall of Famers that come next after the top Hall of Famers in the Hall of Fames respectively and were recorded more appearances. This new calculation resulted in the data shown in the third row of Table 6.5, labeled as *reliable*. For example, Table 6.6 lists the 20 Hall of Famers from one run of the grey-box experiment with $|\mathbb{HOF}| = 20$. The top score, 0.490, is used in calculating the data in the second row of Table 6.5 while the second highest score, 0.458, is used in producing the third row. Statistical tests show that the scores in the third row do not differ significantly at the 95% confidence level.

Based on these experiments and the analysis above, we conclude that the grey-box algorithm is not sensitive to the capacity of the Hall of Fame in the range that we checked (5 to 20). As only 400 mechanisms at most are sampled from the search space in each run of experiments, choosing a capacity that is well beyond 20 (5% of the number of sampled mechanisms already) may introduce

before their induction into the Hall of Fame. But these appearances are not counted in the number of games a Hall of Fame participated, e.g., those shown in the third column in Table 6.6.

Table 6.6: The Hall of Famers from one of the grey-box experiments with $|\mathbb{HOF}| = 20$.

Rank	Score	Games	Market mechanism
1	+0.490	1	$\mathsf{MT}_{\theta=-0.2}$ + QT + AS + CR + $\mathsf{PN}_{n=5}$ + $\mathsf{GF}_{0.1}$
2	+0.458	7	ME + QO + AS + $\mathsf{CP}_{p=0.4}$ + $\mathsf{PN}_{n=5}$ + $\mathsf{GF}_{0.1}$
3	+0.444	2	ME + QO + AS + $\mathsf{CP}_{p=0.7}$ + $\mathsf{PN}_{n=13}$ + $\mathsf{GF}_{0.1}$
4	+0.442	1	MV + QO + $\mathsf{AE}_{w=13,\delta=50}$ + CR + $\mathsf{PN}_{n=7}$ + $\mathsf{GF}_{0.1}$
5	+0.440	4	$\mathsf{MT}_{\theta=-0.8}$ + QS + AS + $\mathsf{CP}_{p=0.3}$ + $\mathsf{PU}_{k=0.8}$ + $\mathsf{GF}_{0.1}$
6	+0.437	14	ME + QS + AA + $\mathsf{CP}_{p=0.4}$ + $\mathsf{PN}_{n=15}$ + $\mathsf{GF}_{0.1}$
7	+0.431	13	$\mathsf{MT}_{\theta=0.8}$ + QO + AS + $\mathsf{CP}_{p=0.1}$ + $\mathsf{PN}_{n=15}$ + $\mathsf{GF}_{0.1}$
8	+0.431	25	ME + QS + AA + $\mathsf{CP}_{p=0.3}$ + $\mathsf{PD}_{k=0.2}$ + $\mathsf{GF}_{0.1}$
9	+0.430	23	$\mathsf{MT}_{\theta=0.6}$ + QS + $\mathsf{AH}_{\tau=0.1}$ + $\mathsf{CP}_{p=0.4}$ + $\mathsf{PU}_{k=0.7}$ + $\mathsf{GF}_{0.1}$
10	+0.426	15	ME + QT + AA + $\mathsf{CP}_{p=0.3}$ + $\mathsf{PN}_{n=13}$ + $\mathsf{GF}_{0.1}$
11	+0.416	6	ME + QS + AA + $\mathsf{CP}_{p=0.5}$ + $\mathsf{PN}_{n=15}$ + $\mathsf{GF}_{0.1}$
12	+0.412	12	ME + QO + AS + $\mathsf{CP}_{p=0.4}$ + $\mathsf{PN}_{n=11}$ + $\mathsf{GF}_{0.1}$
13	+0.411	4	$\mathsf{MT}_{\theta=0.2}$ + QS + $\mathsf{AE}_{w=13,\delta=50}$ + $\mathsf{CP}_{p=0.3}$ + $\mathsf{PN}_{n=7}$ + $\mathsf{GF}_{0.1}$
14	+0.408	14	$\mathsf{MT}_{\theta=1.0}$ + QO + AS + $\mathsf{CP}_{p=0.4}$ + $\mathsf{PN}_{n=9}$ + $\mathsf{GF}_{0.1}$
15	+0.403	23	ME + QS + AA + $\mathsf{CP}_{p=0.2}$ + $\mathsf{PN}_{n=3}$ + $\mathsf{GF}_{0.1}$
16	+0.400	8	$\mathsf{MT}_{\theta=0.2}$ + QO + AQ + $\mathsf{CP}_{p=0.4}$ + $\mathsf{PD}_{k=0.3}$ + $\mathsf{GF}_{0.1}$
17	+0.394	2	MV + QT + $\mathsf{AE}_{w=7,\delta=25}$ + $\mathsf{CP}_{p=0.7}$ + $\mathsf{PU}_{k=0.8}$ + $\mathsf{GF}_{0.1}$
18	+0.393	5	MV + QO + $\mathsf{AH}_{\tau=0.3}$ + $\mathsf{CP}_{p=0.4}$ + $\mathsf{PN}_{n=11}$ + $\mathsf{GF}_{0.1}$
19	+0.393	11	$\mathsf{MT}_{\theta=-0.6}$ + QT + $\mathsf{AE}_{w=13,\delta=40}$ + $\mathsf{CP}_{p=0.3}$ + $\mathsf{PN}_{n=7}$ + $\mathsf{GF}_{0.1}$
20	+0.376	4	MV + QS + AS + $\mathsf{CP}_{p=0.8}$ + $\mathsf{PU}_{k=0.7}$ + $\mathsf{GF}_{0.1}$

too many spam mechanisms in the Hall of Fame, lower the 'pressure' on sampled mechanisms, and result in a prolonged convergence.

6.4.4 A comparison to the genetic algorithm

We have shown in previous sections that the grey-box experiments are successful in producing strong market mechanisms for CAT games. A more convincing way to demonstrate the effectiveness of the grey-box approach is to compare it with other search methods, especially the evolutionary computation approaches that have been applied successfully in the domain of experimental auction mechanism design as in [Cliff, 2001a, 2005; Niu et al., 2006; Phelps, 2007]. In this section, we describe the additional experiments that we carried out based on the classic GA [Forrest, 1993; Holland, 1975] to search in the space of auction mechanisms that was introduced in Section 6.2, the same space as explored in the initial set of grey-box experiments in Section 6.4.1.

In these GA experiments, each individual auction mechanism is represented by a tree structure, which is based on the tree model in Figure 6.1 but differs slightly. As each individual auction mechanism can be viewed as the result of making selections at the $\boxed{\text{or}}$ nodes in the tree model (it is exactly the case in the grey-box experiments), the individual can be conveniently represented by the tree structure after the unselected branches of the $\boxed{\text{or}}$ nodes are cut off from the tree model. For example, the tree on the left side in Figure 6.5 represents the market mechanism

$$\mathsf{ME} + \mathsf{QS} + \mathsf{AD}_{w=3} + \mathsf{CP}_{p=0.4} + \mathsf{PU}_{k=0.7} + \mathsf{GF}_{0.1}$$

where, for convenience, the $\boxed{\text{leaf}}$ node, $\mathsf{GF}_{0.1}$, replaces the subtree that selects GF at the $\boxed{\text{or}}$ node, *policy*, and selects 0, 0, 0, 0, and 0.1 respectively at the $\boxed{\text{or}}$ nodes of f_r, f_i, f_s, f_t, and f_p in Figure 6.1. Similar to the grey-box experiments above, in the GA experiments we explore only among the mechanisms that choose $\mathsf{GF}_{0.1}$ as their charging policy.

The tree-based encoding of an individual requires specialized mutation and crossover operators, due to the hierarchical construction and the different types of node in the tree. The diversity of auction mechanism individuals in the space originates from the $\boxed{\text{or}}$ nodes, so mutation and crossover occur only at $\boxed{\text{or}}$ nodes. To apply mutation to an individual, it is decided probabilistically, based on the *mutation rate*, at each $\boxed{\text{or}}$ node in its tree-based encoding whether the node selects a different child node from the tree model. If yes, the original child (and its children if any) is replaced by the new child, which is uniformly selected from all the possible choices other than the original one. If the new child requires its own descendants, the whole subtree is added. Descendants that are $\boxed{\text{or}}$ nodes make their selections randomly, in contrast to the way in the grey-box experiments where selections are made based on the quality scores of different choices. Figure 6.5

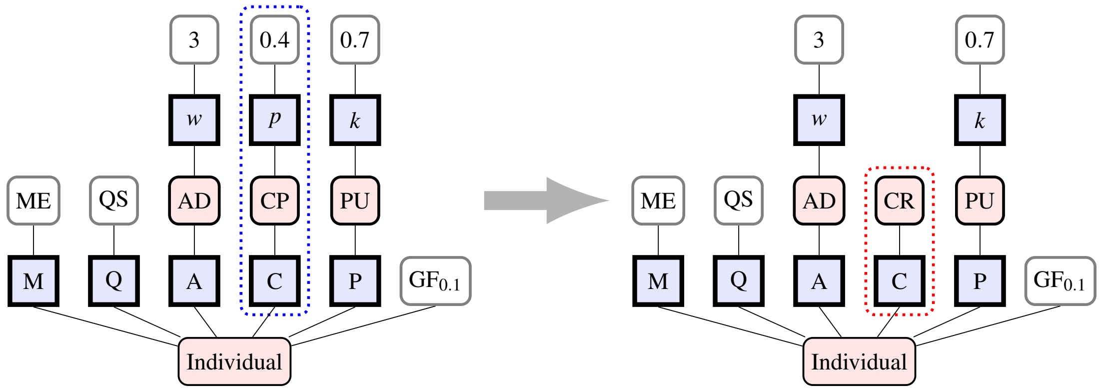


Figure 6.5: An illustration of mutation in the GA on the mechanism denoted as $\mathsf{ME} + \mathsf{QS} + \mathsf{AD}_{w=3} + \mathsf{CR} + \mathsf{PU}_{k=0.7} + \mathsf{GF}_{0.1}$. On the left side is the individual before mutation, and on the right side is the individual after mutation. The replaced and replacing subtrees are both enclosed by dotted lines. The $\boxed{\text{leaf}}$ node, $\mathsf{GF}_{0.1}$, is a convenient representation of the charging policy that is fixed in each mechanism in the search space.

demonstrates an example of mutation on the auction mechanism given above, with the encoding before mutation on the left side and the encoding after mutation on the right side. The node C is the only place where mutation occurs and as a result the branch $\mathsf{CP}_{p=0.4}$ is replaced by CR, both enclosed by dotted lines. The new mechanism is denoted as

$$\mathsf{ME} + \mathsf{QS} + \mathsf{AD}_{w=3} + \mathsf{CR} + \mathsf{PU}_{k=0.7} + \mathsf{GF}_{0.1}$$

Crossover occurs between two auction mechanism individuals in the GA experiments, and only at $\boxed{\text{or}}$ nodes similar to what happens with mutation. To perform crossover, indeed single-point crossover, between two individuals, the $\boxed{\text{or}}$ nodes that appear in both trees and have different children respectively in the two trees are collected; then one of these collected nodes is selected randomly as the place to possibly perform the crossover; and finally it is decided probabilistically, based on the *crossover rate*, whether or not to perform the crossover, and if yes, the two appearances of the selected node in the two trees switch their children. Figure 6.6 demonstrates the crossover between the two individuals—identified as *a* and *b* in the figure respectively—below:

$$\mathsf{MT}_{\theta=0.4} + \mathsf{QO} + \mathsf{AA} + \mathsf{CP}_{p=0.7} + \mathsf{PU}_{k=0.2} + \mathsf{GF}_{0.1}$$

$$\mathsf{ME} + \mathsf{QS} + \mathsf{AA} + \mathsf{CP}_{p=0.2} + \mathsf{PN}_{n=7} + \mathsf{GF}_{0.1}$$

In Figure 6.6, the or nodes at which crossover can be performed are marked with •, including M, Q, P, and p. A and C are excluded because their children in the two trees respectively are also identical, while θ and p in individual a and n in individual b are excluded because they appear in only one of the two trees. Random selection among the eligible nodes picks P. After a probabilistic test based on the crossover rate is taken and turns out to be positive, the subtrees $\mathsf{PU}_{k=0.2}$ in a and $\mathsf{PN}_{n=7}$ in b, both enclosed by dotted lines in the figure, are swapped, producing two new individuals:

$$\mathsf{MT}_{\theta=0.4} + \mathsf{QO} + \mathsf{AA} + \mathsf{CP}_{p=0.7} + \mathsf{PN}_{n=7} + \mathsf{GF}_{0.1}$$

$$\mathsf{ME} + \mathsf{QS} + \mathsf{AA} + \mathsf{CP}_{p=0.2} + \mathsf{PU}_{k=0.2} + \mathsf{GF}_{0.1}$$

which are identified as a' and b' respectively in the figure.

The skeleton of the GA algorithm that is used in our GA experiments is given in Algorithm 2. These GA experiments adopt the same search space of auction mechanisms, the same set of fixed market mechanisms to evaluate the fitnesses of the mechanisms sampled from the space, and the same idea of using a Hall of Fame to produce output as in the grey-box experiments described earlier.

The initial generation of auction mechanism individuals in each GA experiment is created by randomly sampling the search space in exactly the same way as at the beginning of the grey-box search until a certain number (`SIZE_OF_POPULATION`) of individuals are obtained (see Function Init-Population). Each of the subsequent generations is created through steps of selection, crossover, and mutation from the previous generation. The selection step, shown in Function Select-Population, is a combination of *elitism* and *roulette wheel selection*. Elitism selection keeps a certain number of fitter individuals in the next generation based on the *elitism rate*, which determines

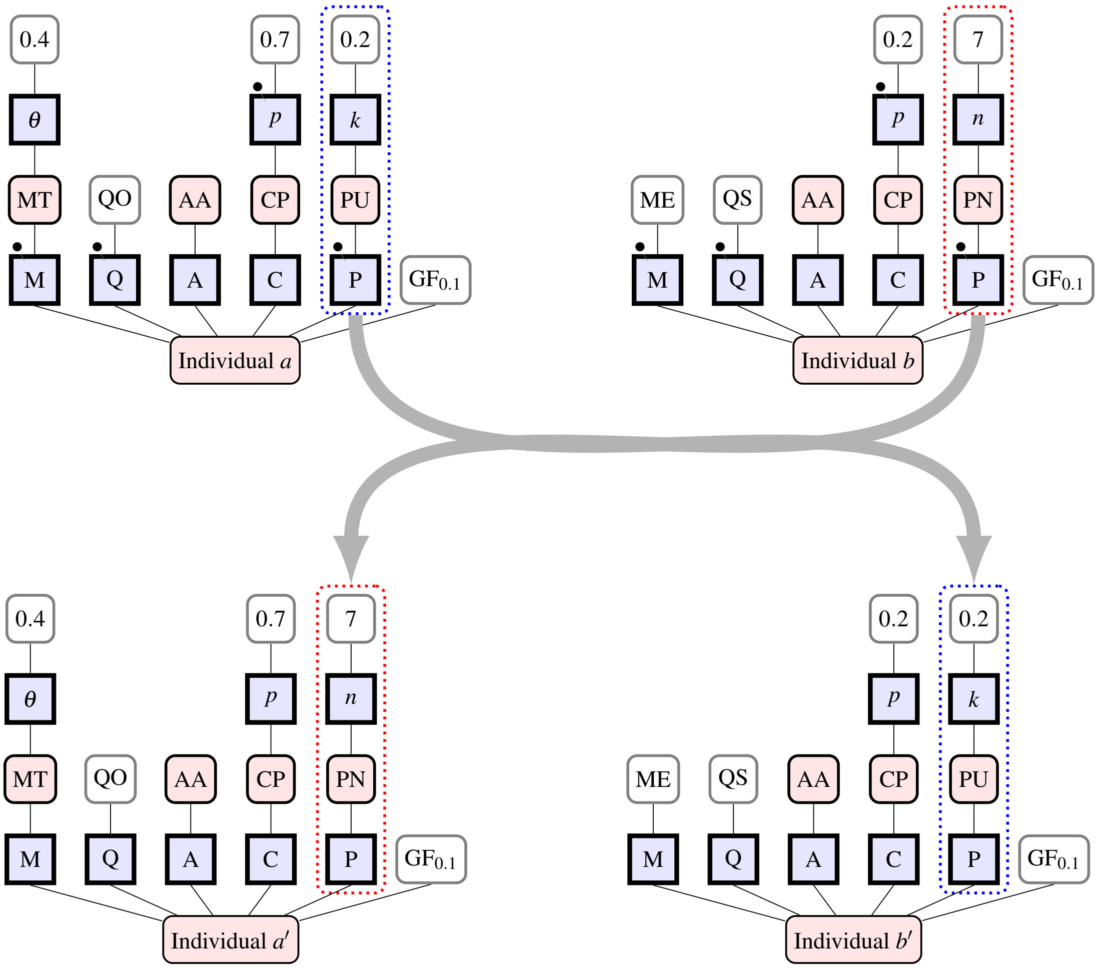


Figure 6.6: An illustration of crossover in the GA between two individuals, $\mathsf{MT}_{\theta=0.4}$ + QO + AA + $\mathsf{CP}_{p=0.7}$ + $\mathsf{PU}_{k=0.2}$ + $\mathsf{GF}_{0.1}$ and ME + QS + AA + $\mathsf{CP}_{p=0.2}$ + $\mathsf{PN}_{n=7}$ + $\mathsf{GF}_{0.1}$, identified respectively as a and b, producing two new individuals, $\mathsf{MT}_{\theta=0.4}$ + QO + AA + $\mathsf{CP}_{p=0.7}$ + $\mathsf{PN}_{n=7}$+ $\mathsf{GF}_{0.1}$ and ME + QS + AA + $\mathsf{CP}_{p=0.2}$ + $\mathsf{PU}_{k=0.2}$ + $\mathsf{GF}_{0.1}$, identified respectively as a' and b'. The $\boxed{\text{or}}$ nodes at which crossover can be performed are marked with • in the original encodings. P is selected to be the place where crossover is actually performed. The two subtrees with P as the root in the two trees are swapped and enclosed by dotted lines.

the size of the portion of the population to be considered as elite individuals. Roulette wheel selection fills the rest of the population by probabilistically selecting among all the individuals in the previous generation. The probability of an individual being selected each time is proportional to

Algorithm 2: The GA-AMD algorithm.

```
Input: 𝔹, 𝔽𝕄
Output: ℍ𝕆𝔽
begin
    ℍ𝕆𝔽 ← ∅
    for g ← 1 to NUM_OF_GENERATIONS do
        if g = 1 then
            ℙ ← Init-Population(𝔹)
        else
            ℙ ← Select-Population(ℙ)
            ℙ ← Crossover-Population(𝔹, ℙ, r_co)
            ℙ ← Mutate-Population(𝔹, ℙ, r_m)
        ℙ ← Randomize(ℙ)
        for i ← 1 to |ℙ|/NUM_OF_SAMPLES do
            G ← Create-Game()
            𝕊𝕄 ← ∅
            for m ← 1 to NUM_OF_SAMPLES do
                𝕊𝕄 ← 𝕊𝕄 ∪ {ℙ[(i − 1) ∗ NUM_OF_SAMPLES + m]}
            𝔼𝕄 ← Select(ℍ𝕆𝔽, NUM_OF_HOF_SAMPLES)
            Run-Game(G, 𝔽𝕄 ∪ 𝔼𝕄 ∪ 𝕊𝕄)
            foreach M ∈ 𝔼𝕄 ∪ 𝕊𝕄 do
                Update-Market-Score(M, Score(G, M))
                if M ∉ ℍ𝕆𝔽 then
                    ℍ𝕆𝔽 ← ℍ𝕆𝔽 ∪ {M}
                if CAPACITY_OF_HOF < |ℍ𝕆𝔽| then
                    ℍ𝕆𝔽 ← ℍ𝕆𝔽 − {Worst-Market(ℍ𝕆𝔽)}
```

its fitness, which is its average daily score in the game that it participated in during the evaluation of the previous generation. This type of selection has a known problem that individuals with low fitnesses have little chance to get selected when the fitnesses of individuals differ dramatically. Due to the scoring scheme of the CAT game, the typical daily score of a market ranges from 0.1 to 0.5, so the usual drawback of roulette wheel selection does not have big impact in this GA algorithm. The individuals that are picked in roulette wheel selection then go through the crossover and mu-

Function Init-Population.

Input: $\mathbb{B}$
Output: $\mathbb{P}$

```
begin
    ℙ ← ∅
    for i ← 1 to SIZE_OF_POPULATION do
        M ← Create-Market()
        for t ← 1 to NUM_OF_POLICYTYPES do
            B ← Select(𝔹_t, 1)
            Add-Block(M, B)
        ℙ ← ℙ ∪ {M}
```

Function Select-Population.

Input: $\mathbb{P}$
Output: $\mathbb{P}'$

```
begin
    ℙ' ← ∅
    Descending-Sort(ℙ)
    n_e ← SIZE_OF_POPULATION ∗ r_e
    for i ← 1 to n_e do
        ℙ' ← ℙ' ∪ {ℙ[i]}
    s ← 0
    for i ← 1 to SIZE_OF_POPULATION do
        s ← s + Score(ℙ[i])
    for i ← n_e to SIZE_OF_POPULATION do
        k ← SIZE_OF_POPULATION
        r ← Uniform(0,s)
        for j ← 1 to SIZE_OF_POPULATION do
            r ← r − Score(ℙ[i])
            if r <= 0 then
                k ← j
                break
        ℙ' ← ℙ' ∪ {ℙ[k]}
```

Function Crossover-Population.

Input: $\mathbb{B}$, $\mathbb{P}$
Output: $\mathbb{P}'$

1 **begin**
2 $\quad \mathbb{P}' \leftarrow \varnothing$
3 $\quad n_e \leftarrow \texttt{SIZE\_OF\_POPULATION} * r_e$
4 $\quad$ **for** $i \leftarrow 1$ **to** n_e **do**
5 $\quad\quad \mathbb{P}' \leftarrow \mathbb{P}' \cup \{\mathbb{P}[i]\}$
6 $\quad$ **for** $i \leftarrow 1$ **to** $(\texttt{SIZE\_OF\_POPULATION} - n_e)/2$ **do**
7 $\quad\quad \mathbb{P}' \leftarrow \mathbb{P}' \cup \{\texttt{Crossover-Individuals}(\mathbb{B}, \mathbb{P}[n_e + i * 2 - 1], \mathbb{P}[n_e + i * 2], r_{co})\}$

Function Mutate-Population.

Input: $\mathbb{B}$, $\mathbb{P}$
Output: $\mathbb{P}'$

1 **begin**
2 $\quad \mathbb{P}' \leftarrow \varnothing$
3 $\quad n_e \leftarrow \texttt{SIZE\_OF\_POPULATION} * r_e$
4 $\quad$ **for** $i \leftarrow 1$ **to** n_e **do**
5 $\quad\quad \mathbb{P}' \leftarrow \mathbb{P}' \cup \{\mathbb{P}[i]\}$
6 $\quad$ **for** $i \leftarrow n_e$ **to** $\texttt{SIZE\_OF\_POPULATION}$ **do**
7 $\quad\quad \mathbb{P}' \leftarrow \mathbb{P}' \cup \{\texttt{Mutate-Individual}(\mathbb{B}, \mathbb{P}[i], r_m)\}$

tation steps. In the crossover step, shown in Function Crossover-Population, individuals are paired up and each pair is probabilistically recombined (`Crossover-Individuals()` in Line 7) as we described above and illustrated in Figure 6.6. In the mutation step, shown in Function Mutate-Population, individuals are each probabilistically mutated (`Mutate-Individual()` in Line 7) as we described above and illustrated in Figure 6.5.

To evaluate a generation of auction mechanism individuals, all the mechanisms are randomly divided into groups. For each group, a CAT game is created, and, similar to those games in the grey-box experiments, this CAT game also includes a set of fixed market mechanisms and a certain number of mechanisms sampled from the Hall of Fame. After the game, the Hall of Fame is updated to incorporate the scores of the participating Hall of Famers and include new individuals from the generation that performed well. The way in which the Hall of Fame is manipulated is exactly the same as in the grey-box experiments. As mentioned above, the average daily scores of the individuals are used as their fitnesses in the selection step.

In the GA experiments, each game is configured to evaluate two individuals from the population as in the grey-box experiments. To compare the performances of the two approaches, the population consists of 20 individual auction mechanisms at each generation and evolves over 20 generations so that each GA experiment makes use of approximately the same number of CAT games in total (200) as in a grey-box experiment.[74] Some experiments based on the GA may have a population of thousands of individuals or even more. Our experiment cannot support a population of this size due to the high computational cost of running CAT games. The 20 generations and the population of 20 individuals are the result of balancing the two parameters under the constraint of the total number of CAT games to run. The elitism rate, r_e, the crossover rate, r_{co}, and the mutation rate, r_m, are set to be 0.1, 0.7, and 0.05, which are typical in the GA experiments reported in the literature [De Jong, 1975; Goldberg, 1989; Haupt and Haupt, 2004]. Table 6.7 summarizes the values of parameters and inputs of Algorithm 2 in our GA experiments.

To provide a better comparison, we ran two sets of GA experiments, one without crossover and the other with it. We ran the GA experiments on the same Linux cluster at the CUNY Graduate Center and plotted in the usual way the daily scores of the four fixed market mechanisms and the top Hall of Famers over time. Figures 6.7 and 6.8 show the results of the two sets of GA experiments together with that from the initial set of grey-box experiments. All the results are averaged over 40

[74] As the Hall of Fame is empty at the beginning of each GA experiment, the first CAT game includes four individuals from the population, so the total number of games to evaluate the 20 generations is actually 199. But the difference of one game can be negligible. In theory, it is possible to design the experiments to run exactly the same number of CAT games as long as $\texttt{NUM\_OF\_GENERATIONS} \times \texttt{SIZE\_OF\_POPULATION} = 402$ and $\texttt{SIZE\_OF\_POPULATION}\%2 = 0$, however the integer solutions—201 and 2, or 67 and 6—to this equation are not practical for the GA as SIZE_OF_POPULATION is too small.

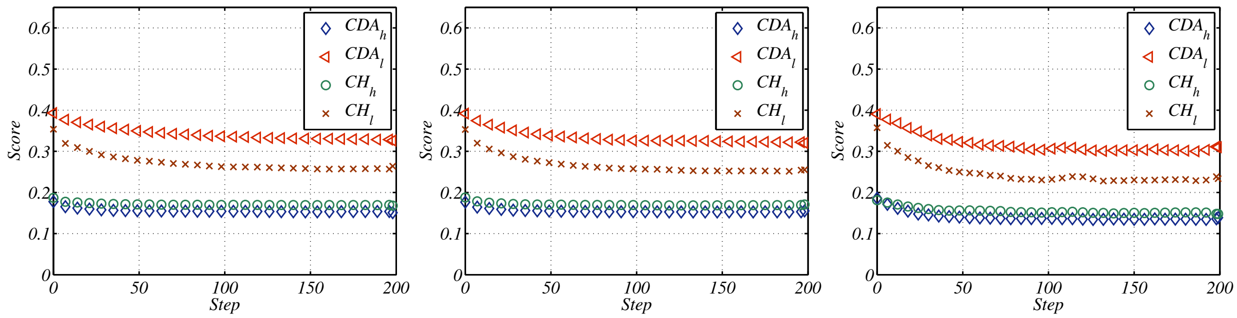
Table 6.7: The values of parameters and inputs of the GA experiments.

Parameter/Input	**Value**
`NUM_OF_GENERATIONS`	20
`SIZE_OF_POPULATION`	20
`NUM_OF_SAMPLES`	2
`NUM_OF_HOF_SAMPLES`	2
`CAPACITY_OF_HOF`	10
`NUM_OF_POLICYTYPES`	5
r_e	0.1
r_{co}	0.7
r_m	0.05
$\tau^\dagger$	0.3
$\alpha^\dagger$	1
$\mathbb{FM}$	$\{\text{CH}_l, \text{CH}_h, \text{CDA}_l, \text{CDA}_h\}$

† τ and α are parameters in the softmax solver used by the `Select`($\mathbb{HOF}$, `NUM_OF_HOF_SAMPLES`) function, identical to τ_1 and α_1 that are listed in Table 6.1 for the `Select`($\mathbb{HOF}$, `NUM_OF_HOF_SAMPLES`) function in the grey-box search.

runs. Note that the x axes in the subfigures are *step* (as in the grey-box experiments), or equivalently the number of games that have been run, rather than *generation* that is common in plotting results from GA experiments. This presentation aims to make it easier to compare the results of the GA experiments with those from the grey-box experiments.

Plots in Figures 6.7a and 6.7b, from the two sets of GA experiments respectively, exhibit the similar pattern as those in Figure 6.7c, which are from the initial grey-box experiments. The scores of the four fixed market mechanisms are at approximately the same positions across the three cases and then all descend until they settle down around certain values. These market mechanisms ended up with the same relative ranking positions in these different cases. The difference is that in the end each of the four market mechanisms settles down with different scores in different cases, the highest in the GA without crossover and the lowest in the grey-box search. This suggests that

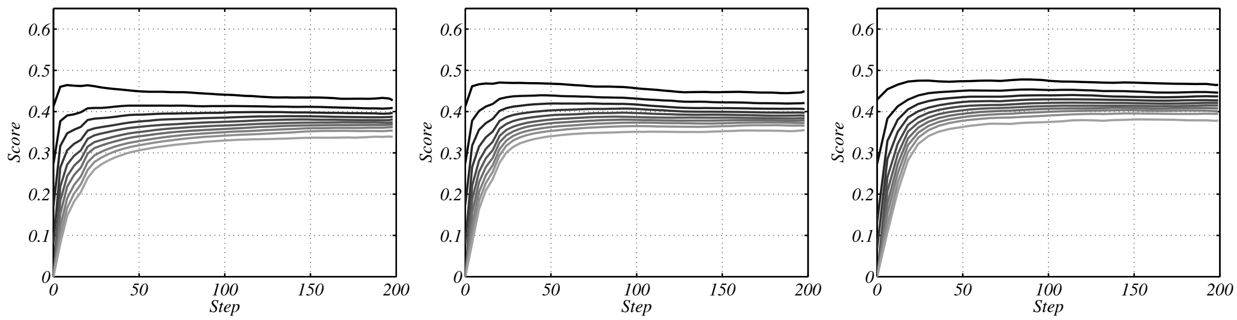


(a) GA without crossover. (b) GA with crossover. (c) Greybox.

Figure 6.7: Scores of the four fixed market mechanisms in the two sets of GA experiments, one without crossover and the other with crossover, and those in the first set of grey-box experiments, each averaged over 40 runs. (c) is identical to Figure 6.2a.

the auction mechanisms explored in the grey-box experiments are overall the most competitive while those explored in the GA experiments without crossover are the least competitive. This further indicates that the grey-box search is more effective than both versions of the GA search and as expected crossover plays an important role in the GA. Figure 6.8 indicates exactly the same. Figures 6.8a and 6.8b, from the two sets of GA experiments respectively, show that the scores of the Hall of Famers increase dramatically at the beginning of the experiments and flatten out at the end around certain positions that are lower than those in Figure 6.8c.

Table 6.8 lists respectively the average scores of the best fixed market mechanism, and the best and worst Hall of Famers at the end of the two versions of GA experiments and the grey-box experiments. At the 95% confidence level, any two values in the second row or any two values in the third row are significantly different from each other. That is to say that the Hall of Famers produced by the grey-box experiments are significantly better than those produced by the GA experiments. The scores of the best fixed market mechanism in the three cases agree to this finding, but they are not significantly different. This less significance is possibly due to the fact that the CAT game is not a zero-sum game, since the transaction success rate of a marketplace in a CAT



(a) GA without crossover. (b) GA with crossover. (c) Greybox.

Figure 6.8: Scores of the Hall of Famers in the two sets of GA experiments, one without crossover and the other with crossover, and those in the first set of grey-box experiments, each averaged over 40 runs. (c) is identical to Figure 6.2b.

game is relatively independent from the performance of its opponents, which counts for one third of its total score. Thus the gain of a stronger market mechanism does not necessarily mean the same amount of loss of the losing mechanism given that all the rest of the configuration remains the same.[75]

To further investigate the effectiveness of the grey-box search in comparison with the GA search, we ran additional experiments to let the Hall of Famers produced by the grey-box experiments and the two sets of GA experiments compete against each other directly. Each of the three sets of experiments produced dozens of the Hall of Famers (69 from the grey-box experiments, 45 from the GA experiments without crossover, and 71 from the GA experiments with crossover).[76] We ran 100 CAT games with eight market mechanisms in each game, which includes two of the fixed market mechanisms, CDA_l and CH_l, and two randomly selected market mechanisms from each of the three set of Hall of Famers. Other than this, the CAT games are configured exactly the same as we did in the grey-box experiments and the GA experiments. Table 6.9 lists the average

[75] One example is that the scores of CDA_h and CH_h flatten out much earlier during the experiments than the scores of CDA_l and CH_l in all the three cases in Figure 6.7.

[76] A Hall of Famer may come from more than one run of the same experiment.

Table 6.8: The average daily scores of the best fixed market mechanism and the best Hall of Famers in the CAT games at the end of the GA experiments, and those at the end of the first set of grey-box experiments. In parentheses are the standard deviations. The scores in the second row are significantly different from each other at the 95% confidence level and so are those in the third row.

Market mechanism	**GA without crossover**	**GA with crossover**	**greybox**†
Best fixed mechanism (CDA_l)	0.3260 (0.0224)	0.3203 (0.0230)	0.3101 (0.0659)
Best Hall of Famers	0.4275 (0.0233)	0.4496 (0.0340)	0.4652 (0.0210)
Worst Hall of Famers	0.3389 (0.0255)	0.3554 (0.0192)	0.3790 (0.0219)

† The values in this column are originally from Table 6.2.

Table 6.9: The average daily scores of the Hall of Famers produced by the GA experiments and the first set of grey-box experiments in direct competition in CAT games. In parentheses are the standard deviations. The scores are significantly different from each other at the 95% confidence level.

GA without crossover	**GA with crossover**	**greybox**
0.3481 (0.0201)	0.3643 (0.0188)	0.4155 (0.0291)

daily scores of the three set of market mechanisms. At the 95% confidence level, the scores of the Hall of Famers from the grey-box experiments are significantly higher than those from either set of the GA experiments.

Section 6.4.2 showed that the grey-box search was able to find mechanisms that are stronger than well known double auction mechanisms when competing directly in CAT games and are better than mechanisms that were reported in the literature in term of various economic properties. Section 6.4.3 examined the sensitivity of the grey-box search to the changes of parameter values and confirmed that the grey-box search can consistently produce similar results when, for example, the capacity of the Hall of Fame varies. This section provides one more piece of evidence for the superiority of the grey-box approach by comparing the results of the grey-box experiments and those of experiments based on different versions of the classic GA.[77]

6.5 Experiment Set II: Learning against entries from CAT 2009

As the mechanisms we found in the first set of grey-box experiments fail to win over entries in CAT 2009, we carried out a second set of grey-box experiments to show how the grey-box approach scales by searching in an extended space that includes policies used in the auction mechanisms of strong entries from CAT 2009. Although there is no formal guarantee, we do expect, in running the second set of grey-box experiments, either to find mechanisms that are able to beat all CAT 2009 entries in a reproduced competition or to confirm that certain entries from CAT 2009 are indeed strong and are identified among the best mechanisms found in the search.

6.5.1 Experimental setup

When a grey-box search fails to produce mechanisms that meet our goal, just as the mechanisms we found in the first set of experiments are unable to win in the reproduced CAT 2009 competition, there are at least two improvements we can make: first to introduce new auction policies into the search space, and second, to use stronger mechanisms in the fixed set of market mechanisms. We consider both types of improvement in the second set of grey-box experiments.

Although the search space in the first set of experiments already includes a variety of policies and some of them are further parameterized, all these policies are simple and fixed, and do not adapt over time within a duration of a single CAT game. The entries in the actual CAT competitions, on the other hand, often adapt the values of parameters in their policies, or switch to different

[77] We actually ran additional sets of GA experiments with crossover, each with a different crossover rate, 0.1, 0.4, or 1.0, in contrast to 0.7 that was used in the GA experiments described in the text. It turned out that the GA experiments using 0.7 produced the best results and hence only the results of this set of experiments were included in the text in the comparison against those of the grey-box experiments.

policies over time in response to the adaptation of their opponents [Niu et al., 2008b]. Intuitively, to combat against these complex mechanisms, the policies in our space should incorporate comparable complexity. As our focus in the work of grey-box search is how to automatically search for effective combinations of building blocks, we do not endeavor to design new, complex building blocks manually, which is contrary to our intention of having an approach of automated design. What we can do however is to directly incorporate policies used by these CAT 2009 entries into our search space.

We intended to incorporate at least policies used by those entries that ranked higher than the mechanisms we found in the first set of experiments as shown in Table 6.12a, including `Mertacor`, `cestlavie`, `IAMwildCAT`, `jackaroo`, `UMTac`. Both `IAMwildCAT` and `UMTac` however include a graphical component, which will make it impossible to run grey-box experiments on our cluster iteratively. So we eventually considered policies used by the other three entries.

`Mertacor` relies upon collecting information about shouts and transactions in the marketplaces regulated by its opponents. This is different from all the mechanisms we considered so far, in which only information from the corresponding marketplace itself is collected and used in its decision making. We introduce a new type of auction policy into the parameterized framework presented in Section 6.2 that regulates this aspect. We call the new type of policy a *subscribing policy*, denoted as S, and the default choice, *self subscribing* or SS.

Policies used by the three CAT 2009 entries are either among those we introduced previously or their own brew. We name policies in the latter case in such a scheme as, for example, Gaj for the charging policy of `jackaroo` and Sam for the subscribing policy of `Mertacor`.

We also introduce a new matching policy, *adaptive matching* or MA, which is a variant of MT. MA sets its parameter θ at 0 to clear the market at the equilibrium point in the first few rounds of a day and increases the value of θ modestly in later rounds of the day so as to increase the transaction

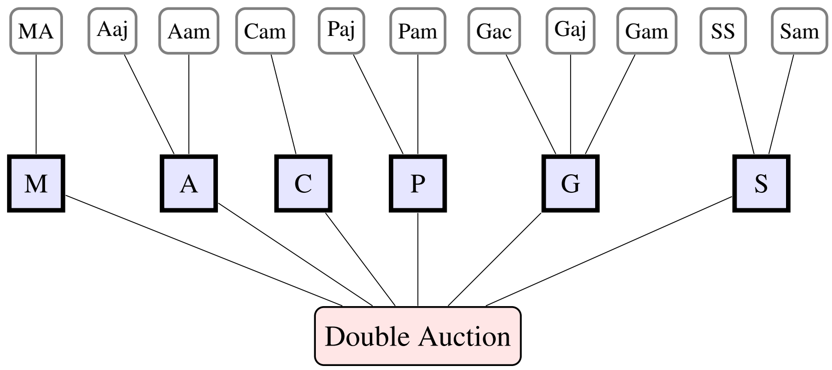


Figure 6.9: The extension of the search space of double auctions in the second set of grey-box experiments.

success rate.

We add all these new policies into the search space and depict this extension of the tree model in Figure 6.9. The three CAT 2009 entries can thus be represented respectively as follows:

$$\texttt{Mertacor} = \mathsf{ME} + \mathsf{Q^*} + \mathsf{Aam} + \mathsf{Cam} + \mathsf{Pam} + \mathsf{Gam} + \mathsf{Sam}$$

$$\texttt{cestlavie} = \mathsf{ME} + \mathsf{Q^*} + \mathsf{AE}_{w=10,\delta=25} + \mathsf{CP}_{p=0.7} + \mathsf{Pac} + \mathsf{Gac} + \mathsf{SS}$$

$$\texttt{jackaroo} = \mathsf{ME} + \mathsf{QT} + \mathsf{Aaj} + \mathsf{CR} + \mathsf{Paj} + \mathsf{Gaj} + \mathsf{SS}$$

where `Q*` represents an arbitrary quote policy as neither `Mertacor` nor `cestlavie` uses the market quotes.

In addition to extending the search space, we replace three members in the fixed set of market mechanisms in the first set of experiments with `Mertacor`, `cestlavie`, and `jackaroo`, and keep the best one, CDA_l, only. Stronger fixed market mechanisms may help to speed up the search in the extended space and to some extent avoid the search being trapped in local optima.

The second set of grey-box experiments are set up in a similar way to the first set of experiments except that each run of the grey-box experiments lasts 600 steps as the search space is bigger and the fixed market mechanisms are more difficult to beat. Table 6.10 lists the part of configuration that differs from that in the first set of experiments.

Table 6.10: The values of parameters and inputs of the GREY-BOX-AMD algorithm in the second set of grey-box experiments that differ from those in the first set of experiments.

Parameter/Input	**Value**
NUM_OF_STEPS	600
$\mathbb{FM}$	{Mertacor, cestlavie, jackaroo, CDA$_l$}

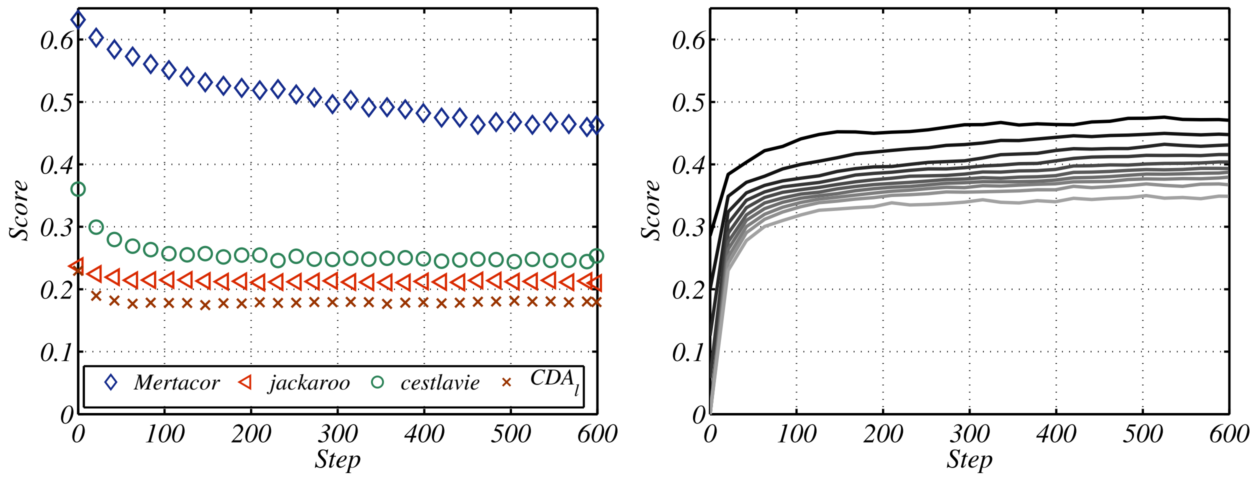
Table 6.11: The average daily scores of the best fixed market mechanism and the best and worst Hall of Famers in the CAT games at the end of the second set of grey-box experiments.

Market mechanism	**Mean**	**SD**
Best fixed mechanism (Mertacor)	0.4628	0.1216
Best Hall of Famers	0.4708	0.0197
Worst Hall of Famers	0.3488	0.0200

6.5.2 Experimental results

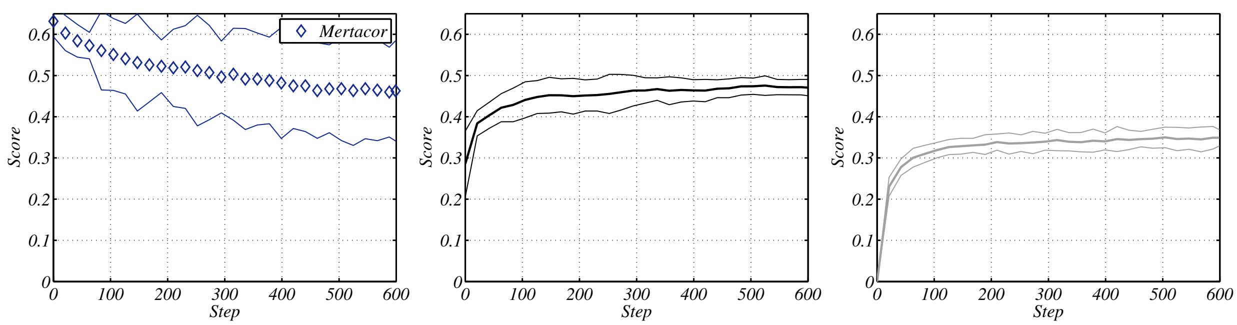
As previously, we generate the plots of the scores of the fixed market mechanisms and the Hall of Famers in the second set of grey-box experiments averaged over 30 runs, which are shown in Figure 6.10. The best member of the fixed set of market mechanisms in the first set of experiments, CDA$_l$, achieves the lowest score as we expected among the new fixed set of market mechanisms, 0.1795 at the last step, which is much lower than its score in the first set of experiments, 0.3101. Also unsurprisingly, as shown in both Figure 6.10 and Table 6.11, Mertacor obtains the highest score among the fixed set of market mechanisms, 0.4628 at the last step. This score is slightly lower than the score of the top Hall of Famers, 0.4708, however the difference is not significant at the 95% confidence level.

Further examination of the Hall of Famers from the 30 runs of the grey-box experiments shows that the mechanism of Mertacor was picked as the top Hall of Famer in steadily more runs over



(a) Fixed market mechanisms.

(b) Hall of Famers with the best at the top and the worst at the bottom.



(c) The best fixed market mechanism.

(d) The best Hall of Famer(s).

(e) The worst Hall of Famer(s).

Figure 6.10: Scores of market mechanisms in the second set of grey-box experiments across 600 steps, averaged over 30 runs.

time and was picked in almost half of the runs by the end of the experiment (Figure 6.11). The mechanisms that are identified as the top Hall of Famer at the last step in the other runs, though not identical to the mechanism of `Mertacor`, adopt many of the individual policies of `Mertacor`:

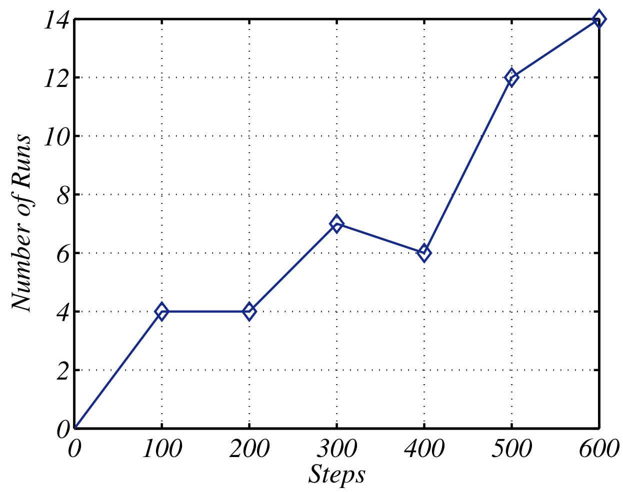


Figure 6.11: The number of runs of grey-box search out of a total of 30 runs in the second set of grey-box experiments that pick `Mertacor` as the best mechanism.

HM0 = ME + Q* + Aam + Cam + Pam + **Gac** + Sam

HM1 = **MA** + Q* + Aam + Cam + Pam + Gam + Sam

HM2 = ME + Q* + **AS** + Cam + Pam + Gam + Sam

HM3 = **MT**$_{\theta=0.2}$ + Q* + Aam + Cam + Pam + Gam + Sam

where **bold** indicates the policies that differentiate the mechanisms from that of `Mertacor`. `HM1` and `HM2` appeared in five and nine runs respectively while `HM0` and `HM3` appeared in a single run each.

In the same way as we examine the performance of the mechanisms we found in the first set of experiments, we ran a reproduced CAT 2009 competition between the CAT 2009 entries and the Hall of Famers listed above. Table 6.12b shows the cumulative scores of these mechanisms averaged over ten games. `Mertacor` still claims the victory, but it scores much less this time than previously if we compare Table 6.12b with Table 6.12a. This is to a great extent due to the strong competition from `HM0` – `HM3`, which are virtually variants of `Mertacor` itself.[78] These `Mertacor` variants take

[78]This is also to some extent due to a larger set of players in the games.

Table 6.12: The scores of market mechanisms in the CAT games including the best mechanisms from the grey-box experiments and entries from CAT 2009, averaged over ten CAT games in both cases.

(a) With mechanisms from the first set of experiments.

Mechanism	Score	SD
Mertacor	241.5715	10.5360
cestlavie	178.8957	3.3455
IAMwildCAT	171.4209	8.3065
jackaroo	161.3124	13.0854
UMTac†	158.6552	7.7849
SM88.0	157.4959	7.9758
SM127.1	150.6758	12.5501
SM7.1	149.7483	15.1307
CUNY.CS	137.5801	5.6975
PSUCAT	134.5170	11.1125
TWBB‡	113.2514	19.8423

(b) With mechanisms from the second set of experiments.

Mechanism	Score	SD
Mertacor	176.5365	24.1721
HM2	176.4945	20.6140
HM3	156.1061	21.1483
HM1	152.3192	18.0645
HM0	152.1263	27.6663
cestlavie	126.8365	14.6078
IAMwildCAT	114.6787	18.2257
jackaroo	114.5572	8.4117
CUNY.CS	93.2921	6.5482
UMTac†	91.5155	17.1831
PSUCAT	90.6562	22.9281
TWBB‡	68.0193	17.6970

† UMTac from CAT 2009 uses fuzzy logic in its mechanism and has a graphical interface to accept certain parameters. As we do not know what parameters should be used, we ran UMTac simply without setting those parameters. It may perform better if the parameters are properly set.

‡ TWBB from CAT 2009 requires a MySQL database in its market making. We were not able to run this on the cluster where we ran the experiments, so the scores of TWBB may not reflect its full capabilities.

the second place through the fifth, pushing down those entries that performed well previously, such as cestlavie, jackaroo, and IAMwildCAT. These observations, together with the high scores of Mertacor as shown in Figure 6.10a, suggest that Mertacor may be the best mechanism that can be found in our extended space of auction mechanisms for CAT games. It also suggests that the competitiveness of Mertacor in CAT games is attributed to its mechanism as a whole and does not hinge upon one or two individual policies alone, as replacing one policy in the mechanism tends to lower the performance of the overall mechanism. It is noteworthy that the score of Mertacor is very close to that of the runner-up, HM2. This indicates that Aam, the only policy that distinguishes

Mertacor from HM2, brings little improvement to the mechanism of Mertacor compared to the known policies like AS in HM2.[79]

Overall, identifying Mertacor as potentially the best mechanism in the search space suggests that our grey-box approach is effective in exploring the search space and scales well when new building blocks are introduced into the search space.

[79] Aam is actually a hybrid of AS and AE. It behaves in the same way as AS most of the time and switches to AE only for new shouts that are placed during a certain period of time in a game.

Chapter 7

Related work, future work, and conclusions

This chapter further discusses resemblances and differences between the pieces of our work that were described in previous chapters and related work in the literature, then outlines several pieces of potential future work, and finally concludes this dissertation.

7.1 On experimental approaches to mechanism design

The work that was presented in this dissertation falls into the area called *agent-based computational economics* [Tesfatsion, 2006], or the one that covers a broader range of topics, *experimental economics* [Roth, 1988; Smith, 1982] in the language of economists. The typical methodology in the area is that experiments are used to validate economic theories and construct market mechanisms through simulations. Although it is common in orthodox economics to assume that mechanism design can be done without experiments, experimental approaches play an important role in validating mechanisms that are based on game theoretic analysis and in turning a conceptually simple design to a complex, operational system. An example is the design of the FCC spectrum auctions in which electromagnetic spectrum licenses were auctioned to competing service providers.

According to the accounts of Nik-Khah [Nik-Khah, 2005], the design of the FCC spectrum auctions was an mixture of efforts from various parties with different backgrounds, including orthodox economists who believed in game theory as well as experimentalists. Although most of the competing proposals for the design of the auctions were from game theorists, including the design that was accepted by the FCC eventually and that was from John McMillan of the University of California San Diego, who was hired by the FCC to deliver an independent perspective, the winning design, one that game theorists claimed to be "simple and would not require the use of specialized auction software", could not be put into use until additional contracts were issued to a group of experimentalists to turn the "simple" design into an operational one. Ironically, among these experimentalists were John Ledyard and David Porter of Caltech, who were involved in preparing the proposal from the NTIA[80] which advocated a fully-operational electronic auction. Indeed, as Plott [Plott, 1997, Page 637] pointed out:

> While the use of laboratory experimental methodology is still in its infancy, it seems clear that the value of the techniques was decisively demonstrated in the development of the FCC auctions.

Market mechanisms used to be processes of human interaction, processes of "collecting bids and reporting back the information described above" [Milgrom and Wilson, 1993], but are more and more becoming automated processes that comprise a system of computers and software [Kephart, 2002; Mirowski, 2007; Schwartz et al., 2006a]. Software agents, acting on behalf of human owners, have limited resources at their disposal and limited access to information, and so have a limited capability of making rational decisions in the economy. Experiments based on software agents or human subjects are ideal tools to address issues that cannot be addressed in game theory, where players in a game are typically assumed to be rational and have unlimited computational resources

[80] The federal agency that regulates the governmental use of spectrum.

and access to information.

The role of computer science is not just running simulations to justify designs by game theorists. Once a marketplace and its patrons are all computerized, advanced methods and tools from computer science can be used to do what cannot be done otherwise. Just as in the case of bioinformatics, which took shape and boomed after information from genetic materials could be digitalized, a computerized economic system leaves the door wide open to carry out the kind of work presented in this dissertation. Furthermore, computational economics is useful not just in scenarios where trading and commerce occur in the first place, but also in solving, for example, many optimization problems in computer science, where the concept of markets was irrelevant originally [Wang et al., 2011; Yeo and Buyya, 2006].

7.2 On marketplaces running in parallel

Our work in this dissertation concerns multiple marketplaces running in parallel. Ladley and Bullock [Ladley and Bullock, 2005] studied the market dynamics involving multiple marketplaces as well, but their work differed from ours on multiple aspects. First, they were concerned with the information available to traders. Traders in their analysis—homogeneous ZIP ones—were each fixed to a certain location in a spatial network and could only trade with and receive information from their neighbors, while traders in our work, though spreading across multiple marketplaces in a similar way, have different capabilities of making offers as they use heterogeneous trading strategies and have the capability of moving between these marketplaces based on their desire to maximize their profits. Second, Ladley and Bullock were concerned with how the different levels of traders' accessibility to market information affected the convergence of the whole market to the theoretical equilibrium, while our work focuses upon whether the movement of traders and the information they obtain by moving around the marketplaces help to differentiate the properties of those marketplaces. Third, Ladley and Bullock constantly used a single classic CDA mechanism in their experiments, which was identical to the one used by Cliff and Bruten [Cliff and Bruten, 1997], while our work involves multiple marketplaces in direct competition that are each associated with an auction mechanism and our work is concerned with how to design effective auction mechanisms to regulate a marketplace and win the competition. In addition, our open source JCAT platform supports traders being able to subscribe for market information by paying a fee and all requests for subscription are permitted without constraint. This design can be easily extended with a spatial constraint, allowing traders to receive information about others that are close to them only, as in [Ladley and Bullock, 2005]. This extension would make it possible to carry out the kind of work by Ladley and Bullock using JCAT, in which a comprehensive collection of trading strategies and auction mechanisms are available for use.

Our work also has similarities to that of Greenwald and Kephart [Greenwald and Kephart, 1999]. In the latter, shoppers choose between different merchants, and the merchants set prices that depend on the prices set by other merchants. Here shoppers and merchants are respectively analogous to traders and specialists (or marketplaces) in CAT games in our work. While some of the results obtained by Greenwald and Kephart, especially the price wars induced by myopic price-setting, look similar to some of ours, the scenario we are considering is considerably more complex. The traders in our scenario learn over time using their profits in the marketplaces as feedback in selecting marketplaces while the shoppers in Greenwald and Kephart's scenario either choose a merchant randomly or choose the merchant that offered the lowest price. The expected return of choosing a marketplace in our work is non-deterministic and hinges upon various factors that a trader has no way to know exactly about in advance or at all, e.g., other traders that go to the marketplace at the same time and the mechanism adopted by the marketplace. In contrast, shoppers in the work of Greenwald and Kephart know exactly about their utilities if they choose to buy from a particular merchant in the retail market. Indeed, the transaction prices are set by the merchants in this scenario, while in our case the prices are determined by the traders. As a result, when traders pick a marketplace in our scenario, they do not know for sure if they will even be able to trade, much less about the price at which goods will change hands. From the perspective of the marketplaces, it is possible to attract many traders who, because of their values for the goods being traded, do not end up trading. The effect of these subtleties is worth further investigation.

Trading agents in our experiments choose marketplaces based on their experience of past profits rather than their knowledge of the concrete auction rules used in the marketplaces. A related, but different, line of work is how to present auction mechanisms that are machine understandable so that trading agents can deduct the game theoretic properties of these mechanisms by themselves and take optimal actions hereafter. For example, Tadjouddine *et al.* [Tadjouddine et al., 2008] explored how to verify the strategy-proof property of an auction mechanism represented in a certain, formal, fashion with model-checking techniques. Although this approach utilizes more information about an auction mechanism in decision making and helps to reduce the cost of trial-and-error behaviors, it has the same limitation as the traditional, analytic, approach to mechanism design, i.e., being applicable only to auction mechanisms that are tractable theoretically. Some other work concerns market selection based on different criteria or feedback. For example, Ganchev *et al.* [Ganchev et al., 2010] very recently discussed an algorithmic trading problem with *censored* feedback: how to spread big-block trades across multiple dark-pool stock markets. They are concerned with how much liquidity each market has and how to spread the requests of trades in a way so as to maximize the volume of successful trades regardless of the relative profit that is made in doing so.

7.3 On evolutionary approaches to automated design

We can also compare our grey-box approach to prior work on automated auction mechanism and trading strategy acquisition based on simple GAs, including Cliff *et al.* [Cliff, 2003] and Phelps *et al.* [Phelps et al., 2006]. A simple GA, or SGA, evolves genomes, or binary strings, using selection, crossover, and mutation operators, while the grey-box approach evolves a vector of quality scores, each for a pre-defined building block, and explores the solution space by biasing those building blocks that lead to better solutions. A SGA maintains a set of sampling points in the solution space and tries to arrive at points of higher fitness that are accessible by applying the operators, while the grey-box approach tends to view the solution space along multiple dimensions simultaneously, maintain a hyperplane that divides the solution space into slices, adjust the sizes of the slices, and identify and explore more in those of high fitness.

A popular theory that intends to explain the effectiveness of SGAs in many optimization domains is the *building block hypothesis*, or BBH [Holland, 1975; Mitchell et al., 1991]. The BBH argues that certain building blocks of low *order* and low *defining length*, called *schemata*,[81] in the genome play a substantial role in constructing genomes of high fitness. The operators of SGAs enable the process to concentrate sampling in subspaces that are identified by these schemata and further in the common areas of these subspaces that have increasing fitness through mixing different schemata. Based on this argument, Thierens and Goldberg [Thierens and Goldberg, 1993] indicated that computational expense grows exponentially with the difficulty of the problem, in terms of the number of schemata and the orders of schemata. Efforts have been made to address this issue with SGAs and improvements to SGAs were proposed by either explicitly exploring to identify schemata or implicitly using special operators to avoid breaking possible schemata in the

[81] A schemata is typically represented in the form, for example, `****01*1***`, where `*` can match `0` or `1`. The defining length of a schemata is the maximal distance between bits with deterministic values, and the order of a schemata is the number of bits with deterministic values.

sampled solutions [Chen and Goldberg, 2005; Goldberg et al., 1989; Li and Goodman, 2008]. The grey-box approach has similarities to these advanced GAs[82] since the grey-box approach explicitly considers the building blocks for auction mechanisms and biases its search towards the corners in the search space that correspond to high quality blocks.

The idea of our approach is in particular similar to that of the *compact GA* , or CGA, which was introduced by Harik *et al.* [Harik et al., 1999]. A CGA represents the population as a probability vector, rather than as a set of binary strings, where the *i*th component of the probability vector gives the probability that the *i*th bit of an individual's genome is 1. Compared with SGAs, CGAs have compact representations and work well in practice. The view of evolving a vector of real-valued quality scores in our grey-box approach (and the use of a real-valued array of probabilities in the CGA) should be distinguished from *real-coded genetic algorithms* [Goldberg, 1990]. In the former case, the vector of real numbers maintains a global view of the conceived fitness landscape of the problem domain or can be considered as a summary of the whole population of individuals if such a population exists, while in the latter case, a vector of real-coded values uniquely determines an individual in the solution domain and one only sees a global view of the fitness landscape when considering all the individuals and their fitness values.

Another topic in evolutionary computation that is related to grey-box search is the problem of early convergence to suboptimal solutions. In the grey-box experiments, parameters of the softmax exploration method in the n-armed bandit problem solvers were carefully set up so that sampling in the search space starts with near randomness and gradually biases modestly towards areas that are fitter than others. More details can be found in Section 6.4.1. Techniques employed in evolutionary computation to address the problem of premature convergence, including *fitness sharing*, *crowding*, and mutation with high rate, are based on similar considerations [Holland, 1975; Sareni and Krähenbühl, 1998]. For example, fitness sharing lowers the fitness of an individual by a cer-

[82] These are sometimes called *competent genetic algorithms*.

tain amount, which basically reflects the number of similar individuals in the population, so that similar individuals with high fitness will not be able to prevail in the next generation. In so doing, the whole population could remain diverse, approaching multiple optima in the space in parallel if applicable. In fact, as a piece of future work, these techniques can be incorporated into the GA experiments in the previous chapter to see if the experiments can produce similar or even better results than the grey-box experiments.

Our grey-box approach should be distinguished from Ronald A. Fisher's work in *population genetics* [Burjorjee, 2008]. Fisher, in his research on Mendelian inheritance, assumed that—as paraphrased by Sewall Wright[83]—

> ...each gene is assigned a constant value, measuring its contribution to the character of the individual (here fitness) in such a way that the sum of the contributions of all genes will equal as closely as possible the actual measures of the character in the individuals of the population.

Wright disagreed with the view of the linear additive contribution of genes and insisted that, based on his experimental work, genes favorable in one combination are extremely likely to be unfavorable in another. Our grey-box approach is not based on Fisher's argument, although the vector of quality scores undoubtedly converges and better auction policies would obtain higher scores if the argument holds in the case of auction mechanisms. When the argument does not hold, which we believe is the case based on our experience with the experiments described in Chapter 5, our grey-box approach may help to obtain insights on which auction policies can make better or bad combinations, and on how to design new, better policies that work better with others.

Finally, the tree-based model of auction mechanisms in our work bears similarities on the surface to the tree structures in genetic programming, though the tree structure in the former case

[83] A co-founder of the field with Fisher and a critic of Fisher's approach.

represents the whole search space and quality scores of building blocks reflect the fitness landscape of the space while tree structures in the latter each represent one individual in the search space and contains no information themselves about how fit they are.

7.4 On the generalization properties of auction mechanisms

We showed in the previous chapter that the grey-box approach to automated auction mechanism design was able to acquire mechanisms that perform well against classic DA mechanisms and CAT entries in CAT games and produce desired economic properties when they run in isolation. This however comes with no formal guarantees that the effectivenesses of these acquired mechanisms are *general* enough so that the mechanisms perform well in some other scenario where a different set of assessment criteria is used. Indeed, Robinson *et al.* [Robinson et al., 2010] examined the generalization properties of certain entries from CAT 2008 and CAT 2009 and they showed, as we expected, that the rankings of these entries in recreated CAT competitions may change when they face different opponents or when the market is populated by a different set of traders.

It is no doubt desirable to design a mechanism that can fit into all kinds of scenarios, and this is possible in some cases. One such example can be found in the work by Sandholm *et al.* on automated mechanism design [Conitzer and Sandholm, 2003, 2004; Sandholm, 2003], where they, as we do in this dissertation, view automated mechanism design as a search problem, and try to design entire auction mechanisms subject to absolute guarantees on their performance, e.g., mechanisms that are strategy-proof and make traders bid truthfully. Such mechanisms, if they exist, would produce guaranteed performance when the traders in the market are well-informed and rational, as the optimal strategy of these traders, simply telling the truth, is known. Nevertheless, mechanisms with such properties or properties like strategy-proofness may not exist or may be difficult to find in many scenarios. Even when such a mechanism does exist, its deployment may come at a high cost. For instance, typical DAs are not strategy-proof; McAfee [McAfee, 1992] derived a type of DA that is strategy-proof but it comes at the cost of lower allocative efficiency.

A practical approach to mechanism design may be to treat the problem as an 'engineering' one [Roth, 2002] and solve it with the aid of experimental methods and tools like the grey-box search and JCAT. Although the experimental approach comes with no formal guarantees [Phelps et al., 2007], it is able to produce good, if not optimal, solutions to a specific version of the problem even when the problem is theoretically intractable with formal analysis. Later on, when the specification of the problem changes, the solutions can be tweaked accordingly or be used as the starting points to search for solutions to the current version of the problem, thus forming a 'step-wise iterative' process [Phelps et al., 2010].

7.5 Future work

There are limitations in our approach and experiments, which motivate several pieces of future work. First, we update the quality scores of building blocks in a mechanism *equally*—every building block receives exactly the same feedback regardless of their contributions—and *independently*—there is no record whether positive or negative feedback comes contingent upon the existence of another policy in the mechanism. This may lead to ineffective feedback and inefficient exploration. One improvement is that heuristic rules may be applied to generate different feedback for updating quality scores of different building blocks in a mechanism. For instance, a mechanism that obtains a bigger profit share than its opponents in a CAT game and charges only on shouts may either have charged higher fees or have had more shouts placed in the market. As a result, stronger feedback should be given to its charging policy and shout accepting policy than to other parts of the mechanism. Another improvement is that combinations of building blocks may be viewed as composite building blocks and added into the tree model in Figure 6.1, which helps in recognizing symbiotic building blocks. Auction policies listed in Section 6.2 and those introduced in Section 6.5.1, more often than not need cooperation of certain other policies, and their contributions to the performance of a market mechanism may hinge on the existence of their cooperative, partner policies. Strong mechanisms are certainly potential places where such symbiotic relations take place. We may add possible combinations of building blocks from these mechanisms into the tree as new branches, and later on identify those mistaken combinations and cut them off using reinforcements from other mechanisms. Here we do not mean to explore all possible combinations. After all, that will lead to an exponentially large search space and does not differ, in essence, from an exhaustive search. What we intend to do is to leverage symbiosis between building blocks, to some extent, so as to produce more accurate causal feedback and explore the space more effectively. The work on *linkage learning* in evolutionary computation, e.g., [Harik, 1997; Watson et al., 1998; Watson and Pollack, 2000] addresses the issues that may arise in this effort and provide guidance.

Second, different runs of the grey-box experiment very likely produce different sets of Hall of Famers and the total number of Hall of Famers from dozens of runs will be hundreds. The three market mechanisms from the first set of grey-box experiments were chosen rather arbitrarily from the 400 Hall of Famers we obtained from 40 runs, and the top Hall of Famers from the second set of experiments won out after a series of games for which the set of players are composed rather randomly. A question that arises is how to choose the best of the best in the end as the output of the grey-box experiments. One way to do so is to use the EGTA approach described in Section 3.2.2 and follow an iterative process that is similar to the one in [Schvartzman and Wellman, 2009b] to obtain those Hall of Famers that are more robust than others. The small set of Hall of Famers that are obtained this way may be further used as the fixed markets in another iteration of grey-box experiments so that better mechanisms used as targets may lead to new better mechanisms over iterations.

Third, the fact that new mechanisms that we obtained through the grey-box experiments failed to win games against entries from CAT 2009, `Mertacor` in particular, suggests that novel, better building blocks should be introduced into the pool of building blocks so that better mechanisms can be constructed. Designing brand new building blocks requires domain knowledge and does not contribute much to the state of the art in a broad range of research fields, however more intelligence and complexity can be incorporated by supporting building blocks of some type that *mixes* the existing ones of the same type. There are at least two kinds of mixing: *concurrent* and *sequential*. A concurrent mixed block selects one of multiple component blocks stochastically with a distribution of probabilities to fulfill its task, as in the concept of mixed strategies in the context of game theory, while a sequential mixed block keeps using one particular pure block over a period of time and switches to another for the next period. Indeed, these two kinds of mixing methods can be integrated in the framework of Markov decision processes in reinforcement learning. These RL-based mixed building blocks are able to significantly contribute to the variety of auction mechanisms, no matter whether these blocks are fixed or allowed to adapt after being incorporated into an auction mechanism. One piece of work that is related to this is [Darwen and Yao, 1997], where a genetic algorithm was used to acquire diverse, simple strategies for the Iterated Prisoner's Dilemma Tournament and these strategies were then combined to form a meta strategy which chooses the best response among these strategies based on its recent interaction against its opponent. This work provides insights upon how simple solutions can be utilized to build composite, adaptive solutions, although the much more complex interactions in CAT games present challenges.

Fourth and finally, we may allow the strategies of traders to evolve in parallel to the market mechanisms. We have used a fixed set of trading strategies in both the CAT games during the grey-box experiments and those CAT games against entries into prior CAT competitions. In the real world, traders tend to adapt their strategies based on their experience so as either to take advantage of weaknesses of market mechanisms or to behave more robustly. To this end, we can model the search space of trading strategies as a tree similar to the grey-box approach to auction mechanism design. The existing trading strategies in the literature, their implementations in JCAT, and prior work on trading strategy acquisition [Phelps et al., 2005; Schvartzman and Wellman, 2009b] together make this task easier. Then two search processes, one in the space of auction mechanisms and the other in the space of trading strategies, can run alternately and iteratively, forming a *co-evolutionary* process. That is, for example, at one step, we fix the space of trading strategies, generate a population of trading agents according to the landscape that is defined by the quality scores of building blocks of trading strategies, and allow exploration in the space of auction mechanisms through CAT games using those trading agents until a good set of Hall of Famers are obtained or the search reaches a plateau based on certain criteria. At the next step, we fix the space of auction mechanisms, run parallel marketplaces using the Hall of Famers obtained from the previous step, and allow exploration in the space of trading strategies. These alternate and iterative steps may run either for a number of iterations or until the search process in one of the two search spaces stops producing significantly different landscapes at two adjacent exploration steps. Given the large number of possible strategies and auction mechanisms, solution concepts like Nash equilibrium as used in [Schvartzman and Wellman, 2009b] may not be readily applicable in this scenario. Prior work on co-evolutionary computation using competitive games, e.g., [Rosin, 1997], might shed light on how to proceed in this direction. The interaction between marketplaces and traders involves open-ended problems, while the work in [Rosin, 1997] considered problems of less complexity or solutions that can fit into a limited representation. We believe that the alternate, iterative approach is promising to help obtain insights into the complex interaction between markets and trading agents, especially how one side responds to the changes on the other side, a topic on which so far as we are aware, little work has been done.

7.6 Summary and Contributions

This dissertation describes a practical approach to the automated design of complex mechanisms. This approach breaks a mechanism down into a set of components each of which can be implemented in a number of different ways, some of which are also parameterized. Given a method to evaluate candidate mechanisms, the approach then uses reinforcement learning to explore the space of possible mechanisms, each composed from a specific choice of components and parameters. The key difference between this approach and previous approaches to this task is that the score from the evaluation is not only used to grade the candidate mechanisms, but also the components and parameters, and new mechanisms are generated in a way that is biased towards components and parameters with high scores.

The specific case-study that we used to develop our approach is the design of new double auction mechanisms. Evaluating the candidate mechanisms using the infrastructure of the CAT tournament, we showed that we could either learn mechanisms that can outperform the standard mechanisms that were used to evaluate the learned mechanisms and the best entries from past CAT competitions or confirm the high competitiveness of a known mechanism in the search space. We also showed that the best mechanisms we learned could outperform mechanisms from the literature even when the evaluation did not take place in the context of CAT games. Additional comparisons demonstrated that the grey-box experiments were able to produce significantly better results than experiments using the GA. These results make us confident that we can generate robust double auction mechanisms with the grey-box approach and that the grey-box approach is an effective approach to automated mechanism design.

The grey-box search also has potential in identifying weaknesses of a particular mechanism.

Mechanisms like the CDA and CH mechanisms, for example, were used in some of our grey-box experiments to evaluate and acquire effective auction mechanisms. This arrangement can be employed in such a reverse manner so that the sampled mechanisms from the search space can be viewed as 'attackers' that help to thoroughly examine aspects of those fixed mechanisms. For instance, if we find in a CAT game that a fixed mechanism receives a score that is much lower than it usually does in other games, we may zoom into the dynamics of the game in a way that is similar to the white-box analysis to see whether a new mechanism takes advantage of flaws in the fixed one. This is to some extent comparable to the work of Rosin [Rosin, 1997] where a population of test cases was evolved along with a population of solutions to the problem of interest so that the arms race between the two species helped to obtain better solutions as well as difficult test cases. The grey-box method, though involving a single species, comes in handy in serving the very purpose in that it automatically produces a variety of new mechanisms. In this scenario, the fitness of a sampled mechanism as well as the quality scores of building blocks involved can be defined as the inverse of the game score of the mechanism under examination. Thus it does not matter much whether or not these new mechanisms are strong competitors in CAT games, as long as they can apply competitive pressure to the fixed mechanisms in one scenario and can help to identify the flaws in those fixed mechanisms.

Indeed, the fitness function for mechanisms in the grey-box search can be defined rather arbitrarily in response to a different scenario other than competing in CAT games and the grey-box search can be deployed as usual. With fewer parameters involved and explicit feedback to guide search, the grey-box search might be more effective than other search methods including the GA, especially in a problem domain where the computational cost of evaluating a solution is high. The superior results of the grey-box experiments to those of the GA experiments, both consuming the same number of evaluations, suggests exactly that.

The work in this dissertation is concerned with auction mechanism design, one of the most important issues in electronic commerce. The model of the DA mechanisms presented here is one of the most comprehensive ones that we are aware of in computational economics research. This model and the various auction policies that we introduced and implemented in JCAT have been commonly used by CAT entrants. For example, a slightly revised version of the history-based shout accepting policy, AH, is the centerpiece of the design of PersianCat, the champion of CAT 2008. Furthermore, the way that we manipulated DA mechanisms may provide guidance to addressing problems involving other kinds of auction mechanisms, e.g., the single-sided auctions used by online marketplaces like eBay and the ad auctions used to monetize search services by Google and Microsoft. And even beyond, agent-based economic paradigms are increasingly used for solving control and optimization problems in computer science [Clearwater, 1996; Gerding et al., 2010]. The methods and tools presented in this dissertation may be valuable in designing artificial economies for, for example, resource allocation and load balancing [Wang et al., 2011]. In this case, the artificial economies are typically isolated and confined to the computer systems of interest, and are thus easier to manipulate and design, avoiding much complexity that is inherent in a real economy.

Bibliography

Axelrod, R. (2006). *The Evolution of Cooperation*. Basic Books, Perseus Books Group, New York, NY.

Bao, X. (2001). Market making and the index futures market. *Perspectives*, 3(3).

Burjorjee, K. M. (2008). The fundamental problem with the building block hypothesis. *CoRR*, abs/0810.3356.

Byde, A. (2003). Applying evolutionary game theory to auction mechanism design. In *Proceedings of the Fourth ACM Conference on Electronic Commerce (EC'03)*, pages 192–193, New York, NY, USA. ACM Press.

Chen, Y.-P. and Goldberg, D. E. (2005). Convergence time for the linkage learning genetic algorithm. *Evolutionary Computation*, 13(3):279–302.

Clark, C. L. (2010). Controlling risk in a lightning-speed trading environment. *Chicago Fed Letter*. No. 272.

Clearwater, S. H., editor (1996). *Market-Based Control: A Paradigm for Distributed Resource Allocation*. World Scientific Publishing Co., Inc., River Edge, NJ, USA.

Cliff, D. (2001a). Evolution of market mechanism through a continuous space of auction-types. Technical Report HPL-2001-326, Hewlett-Packard Research Laboratories, Bristol, England.

Cliff, D. (2001b). Evolutionary optimization of parameter sets for adaptive software-agent traders in continuous double auction markets. Technical Report HPL-2001-99, Hewlett-Packard Research Laboratories, Bristol, England.

Cliff, D. (2003). Explorations in evolutionary design of online auction market mechanisms. Technical Report HPL-2003-80, Hewlett-Packard Research Laboratories, Bristol, England.

Cliff, D. (2005). ZIP60: Further explorations in the evolutionary design of online auction market mechanisms. Technical Report HPL-2005-85, Hewlett-Packard Research Laboratories, Bristol, England.

Cliff, D. and Bruten, J. (1997). Minimal-intelligence agents for bargaining behaviours in market-based environments. Technical Report HPL-97-91, Hewlett-Packard Research Laboratories, Bristol, England.

Cliff, D., Walia, V., and Byde, A. (2002). Evolved hybrid auction mechanisms in non-ZIP trader marketplaces. Technical Report HPL-2002-247, Hewlett-Packard Research Laboratories, Bristol, England.

Conitzer, V. and Sandholm, T. (2003). Automated mechanism design: Complexity results stemming from the single-agent setting. In *Proceedings of the Fifth International Conference on Electronic Commerce (ICEC'03)*, pages 17–24, Pittsburgh, PA, USA. ACM Press.

Conitzer, V. and Sandholm, T. (2004). An algorithm for automatically designing deterministic mechanisms without payments. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'04)*, pages 128–135, Washington, DC, USA. IEEE Computer Society.

Conitzer, V. and Sandholm, T. (2007). Incremental mechanism design. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, pages 1251–1256, Hyderabad, India.

Cramton, P., Shoham, Y., and Steinberg, R. (2006). *Combinatorial Auctions*. The MIT Press.

Darwen, P. J. and Yao, X. (1997). Speciation as automatic categorical modularization. *IEEE Transactions on Evolutionary Computation*, 1(2):101–108.

Das, R., Hanson, J. E., Kephart, J. O., and Tesauro, G. (2001). Agent-human interactions in the continuous double auction. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, Seattle, U.S.A.

De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, Ann Arbor, MI, USA.

Dorigo, M., Maniezzo, V., and Colorni, A. (1996). The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 26(1):29–41.

Easley, D. and Ledyard, J. (1993). Theories of price formation and exchange in double oral auctions. In Friedman and Rust [1993], chapter 3, pages 63–97.

Forrest, S. (1993). Genetic algorithms: Principles of adaptation applied to computation. *Science*, 261:872–878.

Forrest, S. (1996). Genetic algorithms. *ACM Computing Surveys*, 28:77–80.

Friedman, D. (1993). The double auction institution: A survey. In Friedman and Rust [1993], chapter 1, pages 3–25.

Friedman, D. (1998). Evolutionary economics goes mainstream: A review of "The Theory of Learning in Games". *Journal of Evolutionary Economics*, 8(4):423–432.

Friedman, D. and Rust, J., editors (1993). *The Double Auction Market: Institutions, Theories and Evidence*. Santa Fe Institute Studies in the Sciences of Complexity. Westview Press, Perseus Books Group.

Fudenberg, D. and Levine, D. K. (1998). *The Theory of Learning in Games*. MIT Press, Cambridge, MA.

Ganchev, K., Nevmyvaka, Y., Kearns, M., and Vaughan, J. W. (2010). Censored exploration and the Dark Pool Problem. *Communications of the ACM*, 53(5):99–107.

Gerding, E., McBurney, P., and Yao, X. (2010). Market-based control of computational systems: introduction to the special issue. *Autonomous Agents and Multi-Agent Systems*, 21:109–114.

Gerding, E. H., Dash, R. K., Yuen, D. C. K., and Jennings, N. R. (2007). Bidding optimally in concurrent second-price auctions of perfectly substitutable goods. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'07)*, pages 1–8, New York, NY, USA. ACM.

Gjerstad, S. and Dickhaut, J. (1998). Price formation in double auctions. *Games and Economic Behavior*, 22(1):1–29.

Gode, D. K. and Sunder, S. (1993a). Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119–137.

Gode, D. K. and Sunder, S. (1993b). Lower bounds for efficiency of surplus extraction in double auctions. In Friedman and Rust [1993], chapter 7, pages 199–219.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Goldberg, D. E. (1990). Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems*, 5:139–167.

Goldberg, D. E., Korb, B., and Deb, K. (1989). Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3:493–530.

Greenwald, A. (2006). Game theory and the design of electronic markets. Computing Reviews.

Greenwald, A., Jennings, N. R., and Stone, P. (2003). Guest editors' introduction: Agents and markets. *IEEE Intelligent Systems*, 18(6):12–14.

Greenwald, A. R. and Kephart, J. O. (1999). Shopbots and pricebots. In *Agent Mediated Electronic Commerce (IJCAI Workshop)*, pages 1–23.

Harik, G. R. (1997). *Learning linkage to efficiently solve problems of bounded difficulty using genetic algorithms*. PhD thesis, University of Michigan, Ann Arbor, Michigan, USA.

Harik, G. R., Lobo, F. G., and Goldberg, D. E. (1999). The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 3:523–528.

Haupt, R. L. and Haupt, S. E. (2004). *Practical genetic algorithms*. John Wiley and Sons, Inc., second edition.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI, USA; reprinted by MIT Press, Cambridge, MA, 1992.

Horling, B. and Lesser, V. (2005). A Survey of Multi-Agent Organizational Paradigms. *The Knowledge Engineering Review*, 19(4):281–316.

Hu, J. and Wellman, M. P. (1999). Multiagent reinforcement learning in stochastic games. Technical report, Artificial Intelligence Laboratory, University of Michigan.

Hu, J. and Wellman, M. P. (2003). Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4:1039–1069.

Jordan, P. R., Kiekintveld, C., and Wellman, M. P. (2007). Empirical game-theoretic analysis of the TAC supply chain game. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1188–1195, Honolulu, Hawaii.

Jordan, P. R. and Wellman, M. P. (2007). Best-first search for approximate equilibria in empirical games. In *Proceedings of AAAI-07 Workshop on Trading Agent Design and Analysis (TADA-07)*, Vancouver, Canada.

Juda, A. I. and Parkes, D. C. (2006). The sequential auction problem on eBay: an empirical analysis and a solution. In *Proceedings of the Seventh ACM Conference on Electronic commerce (EC'06)*, pages 180–189, New York, NY, USA. ACM.

Kephart, J. O. (2002). Software agents and the route to the information economy. *Proceedings of the National Academy of Sciences of the United States of America*, 99:7207–7213.

Klemperer, P. (1999). Auction Theory: A guide to the literature. *Journal of Economic Surveys*, 13(3):227–286. http://ideas.repec.org/a/bla/jecsur/v13y1999i3p227-86.html.

Klemperer, P. (2002). How (not) to run auctions: The European 3G telecom auctions. *European Economic Review*, 46(4–5):829–845.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge, MA, USA.

Ladley, D. and Bullock, S. (2005). Who to listen to: Exploiting information quality in a ZIP-agent market. In *Proceedings of IJCAI-05 Workshop on Trading Agent Design and Analysis (TADA-05)*.

Lahaie, S. and Pennock, D. M. (2007). Revenue analysis of a family of ranking rules for keyword auctions. In *Proceedings of the Eighth ACM Conference on Electronic Commerce (EC'07)*, pages 50–56, San Diego, California, USA. ACM Press.

Lehmann, D., Müller, R., and Sandholm, T. (2006). The winner determination problem. In Cramton et al. [2006], chapter 12, pages 297–318.

Li, X., Gupta, J., and Koch, J. (2006). Effect of technological breakthroughs on electronic markets. *Electronic Commerce Research*, 6:389–404.

Li, Z. and Goodman, E. D. (2008). Exploring building blocks through crossover. In Kang, L., Cai, Z., Yan, X., and Liu, Y., editors, *ISICA*, volume 5370 of *Lecture Notes in Computer Science*, pages 707–714. Springer.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning (ML-94)*, pages 157–163, New Brunswick, NJ. Morgan Kaufmann.

Littman, M. L. (2001). Friend-or-Foe Q-learning in general-sum games. In *Proceedings of Eighteenth International Conference on Machine Learning*, pages 322–328.

Madhavan, A. (1992). Trading mechanisms in securities markets. *Journal of Finance*, 47(2):607–641.

Maes, P. (1994). Modeling adaptive autonomous agents. *Artificial Life*, 1:135–162.

Maskin, E. S. and Riley, J. G. (1985). Auction theory with private values. *American Economic Review*, 75(2):150–155.

McAfee, R. P. (1992). A dominant strategy double auction. *Journal of Economic Theory*, 56(2):434–450.

McCabe, K. A., Rassenti, S. J., and Smith, V. L. (1993). Designing a uniform price double auction. In Friedman and Rust [1993], chapter 11, pages 307–332.

Milgrom, P. R. and Weber, R. J. (1982). A theory of auctions and competitive bidding. *Econometrica*, 50:1089–1122.

Milgrom, P. R. and Wilson, R. B. (1993). Affidavit of Paul R. Milgrom and Robert B. Wilson. Comments of PacBell PP Docket No. 93–253.

Miller, M. H., John D. Hawke, J., Malkiel, B. G., and Scholes, M. (1988). Findings of the committee of inquiry examining the events surrounding October 19, 1987. Technical report, Chicago Mercantile Exchange.

Mirowski, P. (2007). Markets come to bits: Evolution, computation and markomata in economic science. *Journal of Economic Behavior and Organization*, 63(2):209–242.

Mitchell, M., Forrest, S., and Holland, J. H. (1991). The royal road for genetic algorithms: Fitness landscapes and GA performance. In Varela, F. V. and Bourgine, P., editors, *Toward a Practice of Autonomous Systems: First European Conference on Artificial Life*, pages 245–254. MIT Press, Cambridge, MA.

Nicolaisen, J., Petrov, V., and Tesfatsion, L. (2001). Market power and efficiency in a computational electricity market with discriminatory double-auction pricing. *IEEE Transactions on Evolutionary Computation*, 5(5):504–523.

Nik-Khah, E. M. (2005). *Designs on the Mechanism: Economics and the FCC Auctions*. PhD thesis, University of Notre Dame, Notre Dame, Indiana, USA.

Niu, J., Cai, K., McBurney, P., and Parsons, S. (2008a). An analysis of entries in the First TAC Market Design Competition. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT)*, Sydney, Australia.

Niu, J., Cai, K., and Parsons, S. (2010a). A grey-box approach to automated mechanism design. In *Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, Toronto, Canada. Extended Abstract.

Niu, J., Cai, K., and Parsons, S. (2010b). A grey-box approach to automated mechanism design. In *Proceedings of the Twelfth Workshop on Agent-Mediated Electronic Commerce (AMEC XII)*, Toronto, Canada.

Niu, J., Cai, K., Parsons, S., Fasli, M., and Yao, X. (2012). A grey-box approach to automated mechanism design. *Journal of Electronic Commerce Research and Applications*. To appear.

Niu, J., Cai, K., Parsons, S., Gerding, E., and McBurney, P. (2008b). Characterizing effective auction mechanisms: Insights from the 2007 TAC Mechanism Design Competition. In Padgham, Parkes, Müller, and Parsons, editors, *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pages 1079–1086, Estoril, Portugal.

Niu, J., Cai, K., Parsons, S., Gerding, E., McBurney, P., Moyaux, T., Phelps, S., and Shield, D. (2008c). JCAT: A platform for the TAC Market Design Competition. In Padgham, Parkes, Müller, and Parsons, editors, *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pages 1649–1650, Estoril, Portugal. Demo Paper.

Niu, J., Cai, K., Parsons, S., McBurney, P., and Gerding, E. (2010c). What the 2007 TAC Market Design Game tells us about effective auction mechanisms. *Journal of Autonomous Agents and Multiagent Systems*, 21(2):172–203.

Niu, J., Cai, K., Parsons, S., and Sklar, E. (2006). Reducing price fluctuation in continuous double auctions through pricing policy and shout improvement rule. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006)*, pages 1143–1150, Hakodate, Japan.

Niu, J., Cai, K., Parsons, S., and Sklar, E. (2007a). Some preliminary results on competition between markets for automated traders. In *Proceedings of AAAI-07 Workshop on Trading Agent Design and Analysis (TADA-07)*, Vancouver, Canada.

Niu, J., Mmoloke, A., McBurney, P., and Parsons, S. (2007b). CATP Specification: A communication protocol for CAT games. Technical report, Department of Computer Science, Graduate School and University Center, City University of New York, New York, NY. Version 1.11.

Niu, J. and Parsons, S. (2011). Maximizing matching in double-sided auctions. Working paper, Department of Computer Science, Graduate School and University Center, City University of New York, New York, NY.

Pardoe, D. and Stone, P. (2005). Developing adaptive auction mechanisms. *ACM SIGecom Exchanges*, 5(3):1–10.

Parsons, S., Rodriguez-Aguilar, J. A., and Klein, M. (2011). Auctions and bidding: A guide for computer scientists. *ACM Comput. Surv.*, 43:10:1–10:59.

Petric, A., Podobnik, V., Grguric, A., and Zemljic, M. (2008). Designing an effective e-market: an overview of the CAT agent. In *Proceedings of AAAI-08 Workshop on Trading Agent Design and Analysis (TADA-08)*, Chicago, IL, USA.

Phelps, S. (2005). JASA — Java Auction Simulation API. `http://jasa.sourceforge.net/`.

Phelps, S. (2007). *Evolutionary Mechanism Design*. PhD thesis, University of Liverpool, Liverpool, UK.

Phelps, S., Cai, K., McBurney, P., Niu, J., Parsons, S., and Sklar, E. (2007). Auctions, evolution and multi-agent learning. In *Adaptive Agents and Multi-Agent Systems III. Adaptation and Multi-Agent Learning*, pages 188–210, Maastricht, The Netherlands.

Phelps, S., Marcinkiewicz, M., Parsons, S., and McBurney, P. (2005). Using population-based search and evolutionary game theory to acquire better-response strategies for the double-auction market. In *Proceedings of IJCAI-05 Workshop on Trading Agent Design and Analysis (TADA-05)*.

Phelps, S., Marcinkiewicz, M., Parsons, S., and McBurney, P. (2006). A novel method for automatic strategy acquisition in n-player non-zero-sum games. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'06)*, pages 705–712, New York, NY, USA. ACM Press.

Phelps, S., McBurney, P., and Parsons, S. (2010). Evolutionary mechanism design: a review. *Autonomous Agents and Multi-Agent Systems*, 21(2):237–264.

Phelps, S., McBurney, P., Parsons, S., and Sklar, E. (2002). Co-evolutionary auction mechanism design: a preliminary report. In *Proceedings of the Workshop on Agent Mediated Electronic Commerce IV (AMEC IV)*.

Phelps, S., Parsons, S., and McBurney, P. (2004). An evolutionary game-theoretic comparison of two double-auction market designs. In *Proceedings of the Workshop on Agent Mediated Electronic Commerce VI (AMEC VI)*.

Phelps, S., Parsons, S., Sklar, E., and McBurney, P. (2003). Using genetic programming to optimise pricing rules for a double auction market. In *Proceedings of the Workshop on Agents for Electronic Commerce*, Pittsburgh, PA.

Plott, C. (1997). Laboratory experimental testbeds: Application to the PCS Auction. *Journal of Economics and Management Strategy*, 6(3):605–638.

Preist, C. and van Tol, M. (1998). Adaptive agents in a persistent shout double auction. In *Proceedings of the 1st International Conference on Information and Computation Economies*, pages 11–18. ACM Press.

Robinson, E., McBurney, P., and Yao, X. (2010). How specialised are specialists? Generalisation properties of entries from the 2008 and 2009 TAC Market Design Competitions. In David, E., Gerding, E., Sarne, D., and Shehory, O., editors, *Selected and Revised Papers from AAMAS Workshop, AMEC 2009, Budapest, Hungary, May 2009, and IJCAI Workshop, TADA 2009,* *Pasadena, CA, USA, July 2009*, volume 59 of *Lecture Notes in Business Information Processing*. Springer.

Rosin, C. D. (1997). *Coevolutionary Search Among Adversaries*. PhD thesis, University of California, San Diego, San Diego, CA, USA.

Roth, A. E. (1988). Laboratory experimentation in economics: A methodological overview. *Economic Journal*, 98(393):974–1031.

Roth, A. E. (2002). The economist as engineer: Game theory, experimentation, and computation as tools for design economics. *Econometrica*, 70:1341–1378.

Roth, A. E. and Erev, I. (1995). Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term. *Games and Economic Behavior*, 8:164–212.

Rothkopf, M. H. and Park, S. (2001). An elementary introduction to auctions. Research Report RRR 2-2001, Rutgers Center for Operations Research, Rutgers University.

Rust, J., Miller, J. H., and Palmer, R. G. (1993). Behaviour of trading automata in a computerized double auction market. In Friedman and Rust [1993], chapter 6, pages 155–199.

Sandholm, T. (2003). Automated mechanism design: A new application area for search algorithms. In *Proceedings of the 9th International Conference on Principles and Practice of Constraint Programming (CP)*, volume 2833 of *Lecture Notes in Computer Science (LNCS)*, pages 19–36. Springer.

Sareni, B. and Krähenbühl, L. (1998). Fitness sharing and niching methods revisited. *IEEE Transactions on Evolutionary Computation*, 2(3):97–106.

Satterthwaite, M. and Williams, S. R. (1993). The Bayesian theory of the *k*-double auction. In Friedman and Rust [1993], chapter 4, pages 99–123.

Satterthwaite, M. A. and Williams, S. R. (1989). The rate of convergence to efficiency in the buyer's bid double auction as the market becomes large. *Review of Economic Studies*, 56(4):477–498.

Schvartzman, J. L. and Wellman, M. P. (2009a). Exploring large strategy spaces in empirical game modeling. In *Proceedings of the 11th Workshop on Agent-Mediated Electronic Commerce*.

Schvartzman, J. L. and Wellman, M. P. (2009b). Stronger CDA strategies through empirical game-theoretic analysis and reinforcement learning. In *Proceedings of the Eighth International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Budapest, Hungary. ACM Press.

Schwartz, R. A., Byrne, J. A., and Colaninno, A., editors (2006a). *Electronic vs. Floor Based Trading*. Zicklin School of Business Financial Markets Series. Springer, New York.

Schwartz, R. A., Byrne, J. A., and Colaninno, A., editors (2008). *Competition in a Consolidating Environment*. Zicklin School of Business Financial Markets Series. Springer, New York.

Schwartz, R. A., Francioni, R., and Weber, B. W. (2006b). *The Equity Trader Course*. John Wiley & Sons, Inc., Hoboken, New Jersey.

Shah, A. (1997). Competing exchanges: The international dimension. `http://www.mayin.org/ajayshah/MEDIA/1997/cmarkets.html`.

Shah, A. and Thomas, S. (2000). David and Goliath: Displacing a primary market. *Journal of Global Financial Markets*, 1(1):14–21.

Shoham, Y., Powers, R., and Grenager, T. (2007). If multi-agent learning is the answer, what is the question? *Artificial Intelligence*, 171(7):365–377.

Smith, V. L. (1962). An experimental study of competitive market behaviour. *Journal of Political Economy*, 70(2):111–137.

Smith, V. L. (1982). Microeconomics systems as an experimental science. *American Economic Review*, 72(5):923–955.

Sodomka, E., Collins, J., and Gini, M. L. (2007). Efficient statistical methods for evaluating trading agent performance. In *Proceedings of the 27th Conference on Artificial Intelligence*, pages 770–775, Vancouver, Canada.

Stöber, J. and Neumann, D. (2008). GREEDEX – a scalable clearing mechanism for utility computing. *Electronic Commerce Research*, 8(4):235–253.

Stone, P. and Greenwald, A. (2005). The first international trading agent competition: Autonomous bidding agents. *Electronic Commerce Research*, 5(2):229–265.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA.

Tadjouddine, E. M., Guerin, F., and Vasconcelos, W. (2008). Abstractions for model-checking game-theoretic properties of auctions. In *AAMAS '08: Proceedings of the Seventh international joint conference on Autonomous agents and multiagent systems*, pages 1613–1616, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems. Short Paper.

Tesauro, G. and Bredin, J. L. (2002). Strategic sequential bidding in auctions using dynamic programming. In *Proceedings of the First International Joint conference on Autonomous Agents and Multi-Agent Systems (AAMAS'02)*, pages 591–598, Bologna, Italy. ACM Press.

Tesauro, G. and Das, R. (2001). High-performance bidding agents for the continuous double auction. In *Proceedings of the Third ACM Conference on Electronic Commence*, pages 206–209.

Tesfatsion, L. (2006). Chapter 16 agent-based computational economics: A constructive approach to economic theory. In Tesfatsion, L. and Judd, K. L., editors, *Agent-Based Computational Economics*, volume 2 of *Handbook of Computational Economics*, pages 831–880. Elsevier.

The Economist (2007). Ahead of the tape — algorithmic trading. *The Economist*, 383:85.

Thierens, D. and Goldberg, D. E. (1993). Mixing in genetic algorithms. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 38–47, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Tuyls, K. and Parsons, S. (2007). What evolutionary game theory tells us about multiagent learning. *Artificial Intelligence*, 171(7):406–416.

Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–27.

Vohra, R. V. and Wellman, M. P. (2007). Foundations of multi-agent learning: Introduction to the special issue. *Artificial Intelligence*, 171(7):363–364.

Vorobeychik, Y., Reeves, D. M., and Wellman, M. P. (2007). Constrained automated mechanism for infinite games of incomplete information. In *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, pages 400–407.

Vytelingum, P., Cliff, D., and Jennings, N. R. (2008a). Strategic bidding in continuous double auctions. *Artificial Intelligence*, 172(14):1700–1729.

Vytelingum, P., Dash, R. K., David, E., and Jennings, N. R. (2004). A risk-based bidding strategy for continuous double auctions. In *Proceedings of Sixteenth European Conference on Artificial Intelligence*, pages 79–83, Valencia, Spain.

Vytelingum, P., Vetsikas, I. A., Shi, B., and Jennings, N. R. (2008b). IAMwildCAT: The winning strategy for the TAC Market Design Competition. In *Proceedings of 18th European Conference on Artificial Intelligence*, pages 428–434, Patras, Greece.

Walia, V., Byde, A., and Cliff, D. (2002). Evolving market design in zero-intelligence trader markets. Technical Report HPL-2002-290, Hewlett-Packard Research Laboratories, Bristol, England.

Walsh, W., Das, R., Tesauro, G., and Kephart, J. O. (2002). Analyzing complex strategic interactions in multi-agent systems. In Gmytrasiewicz, P. and Parsons, S., editors, *Proceedings of 2002 Workshop on Game-Theoretic and Decision-Theoretic Agents (GTDT-02)*, Edmonton, Alberta Canada. AAAI.

Walsh, W. E., Parkes, D. C., and Das, R. (2003). Choosing samples to compute heuristic-strategy Nash equilibrium. In *AAMAS 2003 Workshop on Agent Mediated Electronic Commerce*, Melbourne, Australia.

Wang, T., Lin, Z., Yang, B., Gao, J., Huang, A., Yang, D., Zhang, Q., Tang, S., and Niu, J. (2011). MBA: A market-based approach to data allocation and migration in cloud databases. *Science in China Series F: Information Sciences*. To appear.

Watson, R. A., Hornby, G., and Pollack, J. B. (1998). Modeling building-block interdependency. In Eiben, A. E., Bäck, T., Schoenauer, M., and Schwefel, H.-P., editors, *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature (PPSN V)*, volume 1498 of *Lecture Notes in Computer Science*, pages 97–108. Springer.

Watson, R. A. and Pollack, J. B. (2000). Recombination without respect: Schema combination and disruption in genetic algorithm grossover. In Whitley, L. D., Goldberg, D. E., Cantú-Paz, E., Spector, L., Parmee, I. C., and Beyer, H.-G., editors, *GECCO*, pages 112–119. Morgan Kaufmann.

Wellman, M. P., Cheng, S.-F., Reeves, D. M., and Lochner, K. M. (2003). Trading agents competing: Performance, progress, and market effectiveness. *IEEE Intelligent Systems*, 18(6):48–53.

Widrow, B. and Hoff, M. E. (1960). Adaptive switching circuits. In *1960 IRE Western Electric Show and Convention Record*, pages 96–104. The Institute of Radio Engineers, New York, NY.

Wilson, R. B. (1969). Competitive bidding with disparate information. *Management Science*, 15(7):446–448.

Wolfstetter, E. (1996). Auctions: An introduction. *Journal of Economic Surveys*, 10(4):367–420.

Wooldridge, M. (2001). *Introduction to Multiagent Systems*. John Wiley & Sons, Inc. New York, NY, USA.

Wurman, P. R., Walsh, W. E., and Wellman, M. P. (1998). Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, 24(1):17–27.

Wurman, P. R., Wellman, M. P., and Walsh, W. E. (2001). A parametrization of the auction design space. *Games and Economic Behavior*, 35:304–338.

Wurman, P. R., Wellman, M. P., and Walsh, W. E. (2002). Specifying rules for electronic auctions. *AI Magazine*, 23(3):15–23.

Yeo, C. S. and Buyya, R. (2006). A taxonomy of market-based resource management systems for utility-driven cluster computing. *Software: Practice and Experience*, 36(13):1381–1419.

Zhan, W. and Friedman, D. (2007). Markups in double auction markets. *Journal of Economic Dynamics and Control*, 31(9):2984–3005.