Constraint-based Co-evolutionary Genetic Programming for Bargaining Problems

Nanlin Jin

A Thesis Submitted for the Degree of Doctor of Philosophy

Department of Computer Science
University of Essex

2007

Contents

List of Figures

List of Tables

Acknowledgements

Thanks to my supervisor Professor Edward Tsang. He has guided me to build a scientific mind to conduct research and encouraged me to learn how to present, argue and defense my opinions. Thanks to my board meeting members Dr John Ford and Professor Riccardo Poli for their useful advices on my research progress. Special thanks to economists Professor Abhinay Muthoo and Professor Sheri M. Markose. Thanks to my thesis examiners: Dr Simon Lucas and Dr Ken Brown for their great comments and suggestions.

Dr Simon Lucas and Dr William B. Langdon have given me supports and comments on my oral presentations at IEEE CIG05, EuroGP06 and IEEE WCCI06. Their comments and advice have helped me improve my presentation skills. I have enjoyed and benefited greatly from chats with Dr W.B. Langdon and Robert Keller about artificial intelligence, evolutionary algorithms and British weather.

My tutor Alberto Moraglio has acted as an assistant of my supervisor to help with my research for more than three years. Along with Alberto, Tim Gosling, Aimin Zhou, Vikentia Provizionatou, Yossi Borenstein, Serafin Martinez Jaramillo, Dr Helmut Rainer, Hui Li and all the other fellow research students who have made my Essex years enjoyful and fruitful. I am very proud of having been a member of our Essex team.

Dick Williams, Bill Hart, John Tierney and Dr Norbert Volker have helped me improve my teaching skills. Marisa Bostock is my first contact point of all types of administrative issues in the department.

Thanks to all my personal friends, especially to Marisol, Spiridoula, Dong, Nadia, Lei, Alex and Elias. There are too many names to mention. Thanks all in Department of Computer Science and in University of Essex. You all have helped me in many ways during my years in Essex.

Last but not least, thanks to my parents for their forever love, mercy and their emotional and financial supports. More than that, 100% genes of the author of this thesis are luckily copied from theirs, if there is no mutation.

Abstract

This thesis applies evolutionary algorithms to tackle bargaining games. Evolutionary algorithms can discover efficient and stationary strategies for various bargaining games.

Game-theoretic method requires a substantial amount of mathematical reasoning. Thus this method restricts to simple problems. Moreover, game-theoretic solutions rest on the crucial assumption that every player is perfectly rational. These characteristics cast doubts on the applications of game-theoretic method to complex bargaining problems. To overcome such limitations of game-theoretic method, we adopt an alternative method, evolutionary algorithms. We assume that players are boundedly rational. We present a theoretic framework on the basis of co-evolutionary algorithms. We develop Constraint-based Co-evolutionary Genetic Programming system (CCGP) to simulate seven types of two-player bargaining scenarios. On the ground of experimental observations, the co-evolutionary algorithm successfully discovers satisfied and profitable solutions. The computational cost and human efforts of using the co-evolutionary algorithm for bargaining problems are affordable. The CCGP system is reusable.

In particular, this thesis

- simulates boundedly rational players' adaptive learning in two-player bargaining games;
- investigates fitness evaluations in co-evolutionary systems;
- presents a constraint handling technique integrated into evolutionary algorithms. This technique is able to handle situations where both hard and soft constraints exist;

- develops Constraint-based Co-evolutionary Genetic Programming system. It generates solutions with game-theoretic properties;
- demonstrates that artificial training makes nearly perfect.

Chapter 1

Introduction

1.1 Background

Game theory captures essential consideration and reasoning in decision making [Wik05a]. It explicitly explains why and how to make rational choices. A rational choice maximizes the player's utility. A player can be a person, an organization, a nation, an animal or a group. Many social studies and industries benefit from game theory. Moreover, biological evidences have shown that game theory nicely interprets some of animal behaviors too [May82].

Bargaining theory is one branch of game theory [Mut99, Nas50]. Bargaining refers to a process of achieving an agreement on how to divide a common interest between (among) players. Bargaining situations are ubiquitous in our social activities, ranging from political parties' coalition for election to negotiation between husband and wife about domestic affairs, from threats of nuclear war to reaching new international trade agreements. In the game-theoretic view, bargaining situations are typical dynamic games. In this thesis, we focus on two-player bargaining problems.

Game theory, including bargaining theory, studies abstract games [Gib92]. Abstract games are idealized and simplified models of real-world situations [NM44]. The major research method of game theory is the game-theoretic method [NM44, Mut99]. This method mathematically proves *efficient* and *stationary* [1] solutions for some abstract games. Game-theoretic solutions are justifiable under the assumption that game players are *perfectly rational*. Perfectly rational players have stable preferences, unlimited reasoning ability, and complete and perfect information about the game [Sim55].

1.2 Motivations

Game-theoretic method generates perfectly rational solutions. However, this method suffers three major weaknesses. These weaknesses confine applications of game theory to simple games. The three major weaknesses are:

1. The assumption of perfect rationality imposed by game-theoretic method is problematic. Humans are the practitioners of game theory and bargaining theory, but most of humans are *boundedly rational*, i.e. they are not perfectly rational [Sim55]. Users of bargaining theory may question whether the game-theoretic solutions are applicable to boundedly rational players at all;

2. Game theory only considers the most important determinants in an abstract bargaining game. A *bargaining determinant* is an influential factor that affects bargaining outcomes [Mut99]. For example, a discount factor is a bargaining determinant which measures a player's bargaining cost over time. The player takes it into consideration in bargaining. A player's *bargaining power* is the sum of forces from his bargaining determinants acting on the outcome. The bargaining power reflects a player's impact relative to the other player(s) on the bargaining outcome. Even for a simple abstract game with one or two determinants, the game-theoretic method requires substantial

[1] Efficiency and stationarity are defined in Section 2.3.1.

human efforts and expertise. In complicated bargaining situations, there exist many determinants that make various influences on bargaining outcomes. It is unlikely for using game-theoretic method to model many determinants in one game and to mathematically define the interrelationship of all these determinants. Consequently it is too difficult to solve such complicated problems by game-theoretic method;

3. Game-theoretic proofs are not easily reusable even when well-established bargaining models are changed slightly. The cost of providing game-theoretic solutions by this method is high.

These weaknesses of game-theoretic method motivate us to study bargaining problems in a different way. Recent research of biology [May82], psychology and experimental economics [KR95] on learning and evolution inspire us to consider the power of *evolution*. One of theories on human decision making suggests that in reality, it is unlikely that humans resolve real situations by game-theoretic method [Daw76]. Most of time, humans analyze and act through simple heuristics [GTG01]. Humans are boundedly rational. They often have heuristic learning ability and improve their performance based on trial-and-error experiences [GTG01]. Heuristics come from adaptation to the social environment where the decision maker survives [GTG01]. Such an adaptation is a part of social evolution. This is one of important connections between game theory and evolution, so is the theoretic foundation in respect of sociology and psychology, of our choice of applying evolutionary computation to bargaining problems.

1.3 Objectives

To overcome the limitations of game-theoretic method, we tackle bargaining problems by an alternative method: *evolutionary algorithms*. Evolutionary algorithms, a set of computational intelligence techniques, originate from natural evolution. Evolutionary algorithms are heuristic search methods. They have solved many problems in different fields under manageable time and affordable computational resources [Hol62, Gol89, Koz92, Koz94, KIAK99, KKS+03].

In this thesis we attempt to answer following questions:

1. Are game-theoretic solutions of any use for boundedly rational players?
2. Through adaptive learning, will boundedly rational players make good decisions? Good decisions are those of high efficiency and stationarity, according to game theory.
3. For complicated games whose game-theoretic solutions are unavailable yet, are boundedly rational players' decisions efficient and stationary after evolutionary training?
4. Is the system on the basis of evolutionary algorithms easily reusable and extensible for various bargaining problems?
5. Does the system on the basis of evolutionary algorithms require heavy computational resources and/or long time for solving bargaining problems?

1.4 Contributions

This work advances evolutionary algorithms for game theory applications. We study three classic bargaining problems with game-theoretic solutions and four complicated bargaining problems whose game-theoretic solutions are not solved yet. Evolutionary algorithms are tailored to simulate players' learning and decision-making in bargaining situations. Such simulations assume bounded rationality on players' behaviors. A set of new assumptions replaces the set of perfect rationality assumptions imposed by game-theoretic method. We establish a theoretic learning framework for these two boundedly rational bargainers. On the basis of this framework, we develop an extensible system for tackling bargaining problems, aiming to spend little extra efforts in handling these bargaining problems' variations.

In details, this study

- artificially simulates boundedly rational players' behaviors in playing interactive bargaining;
- handles the existence of different types of constraints in one problem which is to be tackled by evolutionary algorithms;
- develops an evolutionary-algorithms-based learning system which provides solutions with game-theoretic properties;
- empirically confirms that artificial adaptive learning and training make players' behaviors nearly perfect: Practice makes perfect.

The experiments demonstrate the dynamic interactions of co-evolving species (players) in the co-evolutionary algorithm. [2] The experimental findings and observations support the applications of evolutionary algorithms to bargaining problems. Using evolutionary algorithms is a practical approach for finding approximate game-theoretic solutions for more realistic bargaining problems.

[2]The co-evolutionary algorithm is a branch of evolutionary algorithms.

1.5 Organization of Thesis

The rest of thesis is organized as follows:

Chapter 2 reviews literature. It summarizes bargaining theory, evolutionary computation and applications of evolutionary computation to games. Three classic bargaining problems, their game-theoretic assumptions and their game-theoretic solutions are introduced.

Chapter 3 studies the basic alternating-offers bargaining problem, abbreviated as CRub82 bargaining problem. The only determinant of this bargaining problem is the bargaining cost over time. Time is valuable. The measure of such cost is discount factor. We establish a co-evolutionary framework to model the players' interactive behaviors and learning. Chapter 3 details the system design based on this framework. Chapter 3 emphasizes the dynamics of the fitness evaluation in the co-evolutionary system.

Chapter 4 reports and analyzes the experimental results generated by the co-evolutionary system. Experimental results are measured by their game-theoretic properties and computational resources consumed. Chapter 3 and 4 form the cornerstone for extensions. Some extensions are detailed in later chapters.

Chapter 5 addresses constraint satisfaction in CRub82 bargaining problem. How to handle constraints is another main concern in appropriately defining the fitness evaluation for this bargaining problem. Chapter 5 invents a constraint handling technique, Incentive method, to deal with the existence of both hard and soft constraints. The concept of the Incentive method is not only applicable to the CRub82 problem, but probably applicable to a wide range of problems, for example financial forecasting [LT99]. In this chapter, the co-evolutionary system is named as Constraint-based Co-evolutionary Genetic Programming system. It is abbreviated as CCGP system.

Chapter 6 studies the role of incomplete information. In CRub82 Problem, the players have complete information about the game before starting. Most of bargaining players in reality are only partially informed about game-relevant information. In Chapter 6, we explore whether the presence of incomplete information necessarily accounts for inefficient outcomes and costly delays after evolutionary processes. Four bargaining problems having different set-ups of incomplete information are examined.

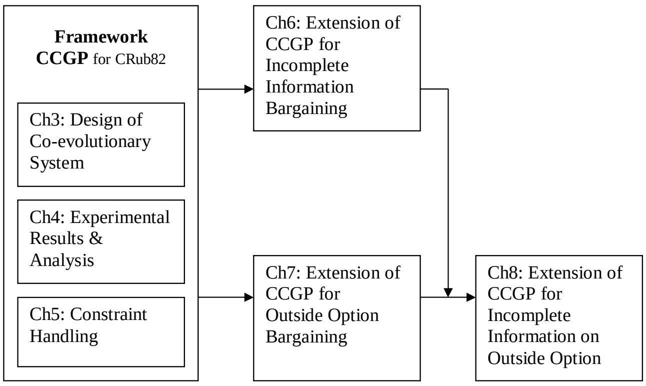


Figure 1.1: Flow of Chapters

Chapter 7 considers another bargaining determinant which is often observed in reality: outside option. Theoretically to input more determinants into a bargaining problem makes it more realistic and more complex to be solved. Chapter 7 studies the role of players' outside options and the compound effects of outside options and discount factors on bargaining outcomes.

Chapter 8 challenges CCGP system with a bargaining problem with incomplete information on outside options and with complete information on discount factors. This problem is so complicated that there is no game-theoretic solution known yet.

Figure 1.1 illustrates the main contents of the chapters excluding Chapter 1 Introduction, Chapter 2 Literature survey and Chapter 9 Conclusions.

Chapter 9 concludes findings and discoveries. It summarizes the seven bargaining prob-

Chapter	Name of the Chapter	Publications
3	Co-evolution to Tackle Alternating-Offers Bargaining Problem - Theoretic Framework and Design	[JT05a] [JT05b] [GTJ06]
4	Co-evolution to Tackle Alternating-Offers Bargaining Problem - Experimental Results and Observations	[JT05a] [JT05b]
5	Constraint Driven Search - Incentive Method	[TJ06]
6	CCGP for Bargaining Problems with Incomplete Information	[Jin05] [Jin]
7	CCGP for Bargaining Problem with Outside Options	[JT06]

Table 1.1: References of our publications on the ground of the initial works of the corresponding chapters.

lems studied in this thesis and the major findings. This chapter also discusses the use of evolutionary algorithms and remarks future studies.

Earlier works of some chapters in this thesis are published. Table 1.1 lists our publications.

Chapter 2

Literature Survey

This chapter reviews game theory, bargaining theory, evolutionary algorithms and highly related works to this thesis.

2.1 Game Theory and its Research Methods

This section briefly reviews the game theory and its four main research methods. These methods are: game-theoretically analytic method, behavioral method, evolutionary game theory approach and artificial simulations.

2.1.1 Game Theory

Game theory mathematically models and studies abstract and idealized human behaviors as well as animal behaviors [NM44, May82]. It is an "application of mathematics to model and analyze interactions with formalized incentive structures ("games")" [Wik05a]. Game theory has achieved great success not only in economics, but also in biology, politics and computer science.

There are various types of games. A *dynamic game* consists of more than one subgame [Gib92]. An *infinite-horizon game* is a dynamic game which can potentially last forever [Mut99]. In a *complete information game*, players have complete information on each other's preferences [Gib92]. In a *perfect information game* the player with the move knows the entire history of the game [Gib92]. Other games include noncooperative and cooperative games, symmetric and asymmetric games, zero sum and non-zero sum games, simultaneous and sequential games, and imperfect information games [Wik05a, Gib92].

2.1.2 Game-theoretic Method

The game-theoretic method was first established by von Neumann and Morgenstern in 1940s [NM44]. Soon after, Nash proved the existence of Nash Equilibrium for incooperative games [Nas51], which widely broadens the domain of game theory. These works lay the foundations of modern game theory.

The essential goal of game-theoretic analysis is to discover what equilibrium(s) are and how to reach them. Equilibriums are recognized as the consequences of the influential forces [oS94]. Game theorists present their theoretical solutions for rational players by mathematical reasoning [NM44]. It is assumed that every involver has perfect rationality as an 'economic man'. The rationality is elaborated in Section 2.2.

Complex games probably have multiple Nash Equilibriums. How to select equilibrium becomes a problem [Gib92]. Theorists therefore propose various definitions on rationality to eliminate some equilibriums in order to refine the Nash equilibrium [Har62, Rub82, Gib92].

2.1.3 Behavioral Method: Experimental Economics

Unlike analytic theorists, experimental economists along with social scientists and psychologists, collect the data of humans' responses to games through questionnaires and competitions [Sim82, BCT92, KR95]. They observe and analyze *actual* human behaviors, trying to explain why in some, specially simple cases, people perform rationally, *as if* they know theoretical equilibriums. In some other cases, often complex situations, people give intuitively reasonable responses, failing to make rational choices.

2.1.4 Evolutionary Game Theory: Evolutionarily Stable Strategy

Maynard Smith and Price (1973) initiated *Evolutionarily Stable Strategy* (ESS) which is the most influential work since the Nash Equilibrium in game theory. Evolutionarily Stable Strategy is mathematically defined in [May82]. Using ESS theory, one can mathematically measure whether a strategy is robust to continually evolutionary pressures. This theory is helpful for understanding efficiency and stability of evolutionarily stable solutions from another angle.

ESS explain little *how* a population adapts to such a stable strategy [Wei95]. In practice however, an ESS may not dominate a population during a certain period of time, due to strong stochastic components emerging in evolutionary process. Fogel et. al [FFA97,FAF98] show that "even in simple games, ESSs may not be stable under conditions that are pertinent in the real-world, such as finite population size and culling selection. Under proportional selection, large finite populations may tend to vary around an ESS, but large can be in the order of 5000 or more individuals in a population."

2.1.5 Artificial Simulation

Artificial simulation is an appealing technique for studying game theory in recent decades. As our understanding of intelligence grows, it seems inevitable that artificial intelligence plays an increasingly important role in computing theory and applications. We discuss it in details in Section 2.4.

2.2 Perfect Rationality and Bounded Rationality

There are two types of rationality: perfect rationality and bounded rationality. “In economics, sociology, and political science, a decision or situation is often called rational if it is in some sense optimal, and individuals or organizations are often called rational if they tend to act somehow optimally in pursuit of their goals” [Wik06a]. Herbert Simon [Sim55] coins the term“Economic Man”. An economic man is *perfectly rational* or *hyper-rational*. He has all relevant knowledge, well-defined and stable system(s) of preferences and of utility functions, and full computational capacity ([Sim55] [Sim82]). Rational players know that all involvers are rational and know the rules of the game [Sim55]. Players in many game-theoretic models are economic men. The assumption of perfect rationality makes game-theoretic solutions unique or makes much less equilibriums valid.

In contrast to perfect rationality, Simon models “administrative man” who is *boundedly rational* and “satisfices - looks for a course of action that is satisfactory or ‘good enough.’ ” ([Sim55] [Sim97]). Simon states “boundedly rational agents experience limits in formulating and solving complex problems and in processing (receiving, storing, retrieving, transmitting) information”. Humans are boundedly rational: we only have limited time, knowledge, information and computational resources. In addition, the definition of bounded rationality actually rooms many variations of bounded rationality, varying from complete randomness (exclusive) to perfect rationality (exclusive). Complete randomness can not be regarded as having any rationality at all.

Because of existing variations of bounded rationality, there is no concrete outcome of a game which is played by boundedly rational players, without fully defining the exact behaviors that such boundedly rational players are able to do. For this reason, we define a set of assumptions of bounded rationality on bargaining players' behaviors (see Section 3.3). When computation is concerned, bounded rationality can be interpreted as applying a certain algorithm, rule or program and using a certain hardware configuration for a certain period of time [1]. Our experiments and conclusions are subject to the set of assumptions defined in Section 3.3, the evolutionary algorithms in Section 3.4 and computational resources in Section 4.2.6.

There is a connection between (perfect) rationality and heuristics (a kind of bounded rationality). [GTG01] argues that "rationality can be found in the use of fast and frugal heuristics, inference mechanisms that can be simple and smart". This term "rationality" above can be understood as a kind of bounded rationality which approximates perfect rationality. [GTG01] further explains the reasons why such a simple heuristic exists and how it relates to (perfect) rationality: "these heuristics are successful to the degree they are ecologically rational". The mind's adaptive toolbox has "adapted to the structure of the information in which they are used to make decisions". Because of such an adaptive mechanism, heuristics as a result of adaptive learning perform fairly good in decision making while the laws of logic and probability analysis *seem* playing a less important role. This phenomenon has also been found by biologists. [Daw76] gives a few examples on how human make decisions "unconsciously" due to our adaptation to environments.

2.3 Bargaining Problems and Game-theoretic Solutions

Generally speaking, "bargaining" is the process of reaching agreements. In the bargaining problems of our interest, players offer and counter-offer over partition of a cake. In practice,

[1] For example, a boundedly rational approach to the travelling salesman problem [GBDJ54] could be applying Branch and bound algorithm [LD60] in a C++ implementation on a Windows XP operation system running on a Pentium IV, 3GHz machine with 1 GB RAM for 1 hour.

bargaining may cover many complex issues and terms, such as prices, quantity of products, deliverable time, payment methods, quality of service, compound goods and services. A theory studies the essentials of bargaining situations is bargaining theory. It is one of the subbranches of game theory.

Typically the bargaining theory models and studies a class of situations where participants (players) have a common interest, also called mutual benefit or surplus in game theory literature, but they conflict over how to divide the interest between(among) them. A common interest is often symbolized as ‘a cake’. Players bargain in order to make an agreement over the partition of a cake.

Nash bargaining model [Nas50] is the fundamental and abstract one. Researchers in bargaining theory follow the general framework of Nash bargaining model, adding more realistic components into it to study more practical situations. Rubinstein [Rub82] extends Nash bargaining model, considering the cost on bargaining time and specifying the alternating-offers bargaining procedure, also called sequential bargaining procedure. Later researchers specify bargaining procedures, consider bargaining cost, risk of breakdown and outside and/or inside options and investigate information completeness [Mut99].

In this thesis we study two-player bargaining problems. Two-player bargaining is fundamental. Game theorists often start with a two-player model, then extend to multi-player models [Mut99]. When more and more players join in, a game becomes more and more complicated. Darwen and Yao study multiple players playing IPD [DY94]. The results are far more complicated than two-player IPD. Gosling and Tsang tackle the simple supply chain model in which many buyers and sellers, together with middle men are bargaining over various products and services [GTJ06, GT06]. Their works demonstrate many features in reality.

Three classic bargaining problems from game theory literature are introduced shortly. We also present four bargaining problems which have not been treated by game theory yet. These four bargaining games are presented in the chapters where they are attempted to be solved.

2.3.1 Alternating-Offers Bargaining Problem - CRub82

Basic Alternating-Offers Bargaining Problem or so called Rubinstein Bargaining Problem [Rub82], (abbreviated as CRub82) is a dynamic two-player game with complete and perfect information. In a *complete information* game, all players' information are public. In a *perfect information* game, at each move the player with the move knows the full history of the game thus far [Gib92]. Board games are perfect information games: players observe the history of moves and states. So do the participants in most bargaining problems where each player witnesses the offers and counter-offers made by his counterpart(s) [2].

CRub82 bargaining problem describes a bargaining scenario in which the first-move player 1 makes an offer or count-offers to the second-move player 2 on dividing a cake $\pi = 1$ at even time t: 0, 2, 4, 6... Time t is a non-negative integer. Player 2 either accepts the offer from player 1 immediately or rejects it immediately. If player 2 rejects this offer, after one time interval Δ, player 2 makes a counteroffer to player 1. To make the problem simpler, we assume Δ is 1. Player 2 makes counter-offers at odd time 1, 3, 5, 7, ... The bargaining process ends once an offer or a counter-offer is accepted by the other player. At any given time when one makes an offer, the other can either accept it thus the game ends with an agreement, or reject it then the game continues. The player who has rejected the previous

[2] who can be either his cooperative partner(s) or rival(s).

offer makes a counter-offer. An offer or counter-offer on diving the cake by player i is x_i for himself and $x_j = 1 - x_i$ for player j. $i, j \in \{1, 2\}$.

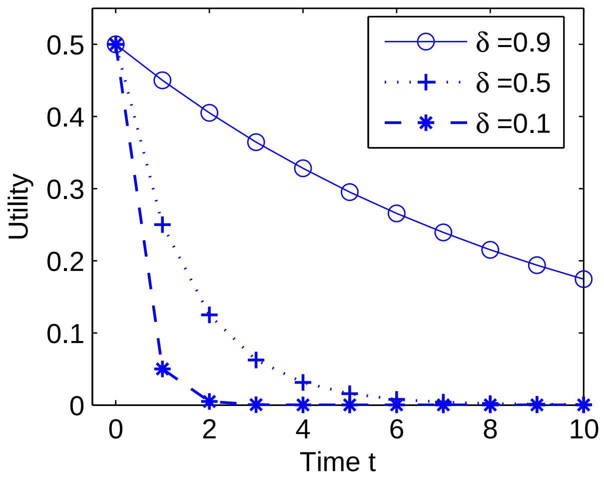


Figure 2.1: Utilities are decaying over time subject to discount factor δ.

In this problem the incentive for players to reach an agreement soon is their bargaining costs subject to time. This incentive is termed by the discount factors. *Discount factor* measures a player's time preference. The pair of discount factors (δ_1, δ_2) for the two players specifies the respective costs subject to bargaining time. The pair of discount factors is the important determinant that determines players' time preferences and thus their bargaining powers. Player i's discount factor δ_i is his bargaining cost per time interval, which means his partition from a cake, for example a slide of 0.5, shrinks to $0.5 \times \delta_i^t$ at the time t. Figure 2.1 illustrates that utilities decay over time. While t increases the cake shrinks exponentially. $\delta_i \equiv e^{-r_i}$ where r_i is player i's discount rate. In theory, $0 < \delta_i < 1$. $\delta_i = 0.1$ means that a cake of size 1 shrinks to 0.1 after one time interval for player i. $\delta_i = 0.9$ means that this cake becomes 0.9 after one time interval. $\delta_i = 0.1$ is a relatively lower discount factor and $\delta_i = 0.9$ is a relatively higher discount factor. A player who has a higher discount factor is relatively more patient, while a player with a lower discount factor is less patient in waiting for reaching an agreement. The player with a lower discount factor compromises more in bargaining as he pays more on bargaining cost per one time interval. The utility gained by player i who has the share x_i from the agreement that is reached at time t is determined by the utility function: $x_i \delta_i^t$. [3]

To help readers understand "discount factor", and its relation to players' patience, we give an example. Consider two friends are bargaining over dividing an ice-cream cake in a very hot summer afternoon at sunny beach, without drinking anything for hours. They are in a hurry to make a deal because ice-cream is melting very quickly and they are thirsty, the longer time they spend on bargaining, the less ice-cream they get in the end. This is a case when players are impatient. If these two guys meet again to divide another ice-cream in a wet and cold winter early morning after drinking plenty of warm juice, they do not even bother to make any deal today or next week. In the later case, players are very patient as there is no time pressure to make an agreement. Players in the summer case have much smaller discount factors than players in the winter case.

Other examples about time preferences in the context of economics research are available in [Mut99] and in a non-technical study [Mut00].

Assumptions and Subgame Perfect Equilibrium

In terms of game theory, CRub82 bargaining problem is an infinite-horizon game. All combinations of players' behaviors are infinite. As stated in Section 2.2, game theorists reduce the solution space by imposing strict assumptions on players' rationality: each player is an economic man. Game-theoretic solutions satisfy two game-theoretic properties [Mut99]: "*no delay*" and "*stationarity*". No delay means that "whenever a player has to make an offer,

[3] There is a subtle difference between payoff and utility in game theory literature [Mut99]. In this thesis, "payoff" and "utility" are exchangeable in this thesis.

her equilibrium offer is accepted by the other player". No delay also implies the *efficiency* of agreement. An efficient agreement should be reached at $t = 0$, so no cost is spent on bargaining. Stationarity requires "in equilibrium, a player makes the same offer whenever she has to make an offer". Theorists mathematically proves the existence of the *Subgame Perfect Equilibrium* (SPE) under such assumptions. In SPE players should offer nothing than the perfect equilibrium partition. Surely this offer will be accepted at time 0 [Rub82, Mut99]. Partitions thus are guaranteed before a bargain even starts, given the discounts factors. This game-theoretic solution SPE is unique and can be expressed analytically. The Subgame Perfect Equilibrium is the first player obtains x_1^*:

$$x_1^* = \frac{1 - \delta_2}{1 - \delta_1 \delta_2} \tag{2.1}$$

$$x_2^* = 1 - x_1^* \tag{2.2}$$

The second player obtains the rest of cake, $(1 - x_1^*)$.

Appendix B provides a non-technical description of game-theoretic analysis of CRub82. It aims to help readers understand the thoughts of game-theoretic reasoning. Technical treatments and proofs are available in [Rub82, Mut99, BF98].

2.3.2 Rubinstein Incomplete Information Bargaining Problem - ICRub85

In an *incomplete information* game, at least one player is uncertain about another player's game-relevant information [Gib92].

Computer science contributes to understanding of incomplete information games in different ways. [FWJ05] systematically analyzes incomplete information alternating-offers bargaining models, taking discounting factors, deadlines, and reservation prices into consideration.

In [FWJ05] three types of decision functions are discussed, namely linear, Boulware and Conceder. Authors of [FWJ05] mathematically prove the existence of such equilibriums that meet properties of uniqueness, symmetry and efficiency of solution. The players in [FWJ05] are designed as economic men without any evolutionary learning ability. [BPSS02]'s simulations of Poker, an incomplete information game and the modelling of the opponent of a Poker player involve artificial neutral network. [CR03a] examines evolutionary dynamics of agents in contribution games and subscription games under incomplete information. [Aus04] inspects the allocative efficiency in an alternative ascending-bid auction with incomplete information. [Eym01] studies a multi-agent system considering the constraints of incomplete information and time pressures in evolutionary learning environments.

Rubinstein Incomplete Information Bargaining Problem [Rub85], abbreviated as ICRub85 is an infinite-horizon game with incomplete and perfect information. On the basis of the complete information alternating-offers bargaining problem CRub82, Rubinstein [Rub85] further studies an alternating-offer bargaining problem with one-sided incomplete information. One-sided incomplete information games refer to the situations where only one of two players have incomplete information. ICRub85 specifies that player 1 only knows that the value of player 2's discount factor δ_2 is either δ_w or δ_s. δ_s is strictly larger than δ_w. Player 2 knows player 1's discount factor is δ_1. The possibility of player 1's initial belief on player 2's type being δ_w is ω_0.

Game-theoretic method proves that in the case of $\omega_0 < \omega^*$ (the definition of ω^* is in Equation (2.4)), player 1 offers a division of V_s to himself and the rest to player 2. This offer will be accepted by both 2_w whose discount factor is δ_w, and 2_s whose discount factor is δ_s. The game-theoretic solution is *Perfect Bayesian Equilibrium*, abbreviated as *PBE*. In PBE, player 1 gets V_s and player 2 gets $1 - V_s$. Interestingly, if $\delta_2 = \delta_s$ and $\omega_0 = 0$ the PBE (V_s, $1 - V_s$) is identical to the SPE for CRub82 bargaining problem.

$$V_s = \frac{1 - \delta_s}{1 - \delta_1 \delta_s} \tag{2.3}$$

$$\omega^* = \frac{V_s - \delta_1^2 V_s}{1 - \delta_w + \delta_1 V_s(\delta_w - \delta_1)} \tag{2.4}$$

When ω_0 is high enough such that $\omega_0 > \omega^*$ the unique PBE for player 2 is that player 1 offers x^{ω_0}:

$$x^{\omega_0} = \frac{(1 - \delta_w)(1 - \delta_1^2(1 - \omega_0))}{1 - \delta_1^2(1 - \omega_0) - \delta_1 \delta_w \omega_0} \tag{2.5}$$

and player 2_w should accept this offer and the game over. Player 2_s rejects it and subsequently makes a counter-offer: y^{ω_0} for player 1 and $1 - y^{\omega_0}$ for herself:

$$y^{\omega_0} = 1 - \frac{1 - x^{\omega_0}}{\delta_w} \tag{2.6}$$

which will be accepted by player 1. In this PBE player 1 acquires y^{ω_0} and player 2 obtains the rest of the cake. If $\omega_0 = \omega^*$, more than one bargaining PBE are possible. From the Perfect Bayesian Equilibria above the actual value of the δ_2 has no direct effect on player 1's first offer. Instead player 1's initial belief ω_0 and the combination of δ_w and δ_s decide player 1's first offer. Table 2.1 summarizes the PBE.

2.3.3 Bargaining Problem with Outside Options - COO

Binmore [Bin85] models and solves a two-player and one-cake problem with an "outside option" pair. Outside option is a player's alternative choice beside the bargaining. In addi-

	$\delta_2 = \delta_w$		$\delta_2 = \delta_s$	
	x_1^*	t^*	x_1^*	t^*
$\omega_0 < \omega*$	V_s	0	V_s	0
$\omega_0 > \omega*$	x^{ω_0}	0	y^{ω_0}	1

Table 2.1: Player 1' shares x_1^* in PBE agreements and the time t^* for reaching such agreements. Player 2's share in such agreements is $x_2^* = 1 - x_1^*$. All bargains start at time 0.

tion, the above solution is supported by Rubinstein bargaining model with an outside-option pair. [BSS89] investigates both the game-theoretic method and human-subject experiments. Binmore et. al [BPSS98] examines the Nash Demand game with outside options. The experimental study on this game shows that human subjects make inefficient outcomes are common phenomena. When the second player's outside option is sufficiently large, the mutual benefit for the bargaining often remains unexploited. Through experimental studies on demand and ultimatum games, Kahn and Murnighan [KM93] find that human-subject experimental data from human-entry, as a whole, argue against game-theoretic quantity predictions in terms of opening demands, bargaining outcomes and efficiency.

One of classic outside option bargaining problems is on the basis of the complete information alternating-offers bargaining problem CRub82. This outside option bargaining problem is abbreviated as COO. In CRub82 and ICRub85, when a player is given an offer, he has two choices: (1) acceptance thus ending the bargain with an agreement; or (2) rejecting the offer and making a counter offer after one time interval. COO integrates outside option(s) into CRub82. In the presence of an outside option, a player has one more choice besides the two mentioned: he can choose to (3) end the bargain by taking his outside option, as illustrated in Figure 2.2. One player has no more than one outside option. If both players have outside options larger than 0, such a situation is called *two-sided outside option.* If one player has no outside option (or having an outside option equals 0) and another has an outside option larger than 0, it is called *one-sided outside option*. We treat one-sided outside option situations as special cases of two-sided outside option bargaining problems.

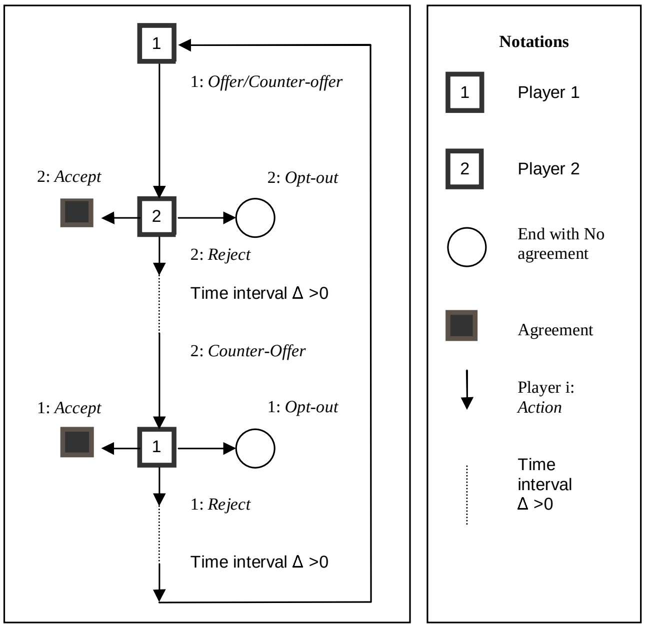


Figure 2.2: Outside Option Bargaining Scenario

In COO bargaining problem, there are two players bargaining over a partition of a cake of size 1. Two players are indexed by subscript i or j, $i, j \in \{1, 2\}$. All game related information are public. Their bargaining costs over time are measured by discount factors δ_1 and δ_2 respectively. Player 1 has his outside option $w_1 \in [0, 1)$ and the second player 2 has an outside option $w_2 \in [0, 1)$. If neither player has an outside option $w_i = 0$ and $w_j = 0$, the bargaining model becomes the CRub82 bargaining model in which discount factors are the sole determinant on bargaining powers. In this thesis, we consider cases where at least one player has an outside option: $w_i > 0$ or $w_j > 0$.

To ensure that the bargaining is worth continuing and no player prefers to withdrawal from bargaining, the conditions $0 \leq w_1 < 1$, $0 \leq w_2 < 1$ and $0 < w_1 + w_2 < 1$ must be satisfied. We include one-sided outside option cases as special cases where only one player has an outside option, $w_i = 0$ but $0 < w_j < 1$, where $i \neq j$.

Values of Outside Options	*Outside Option*	*Solutions*
$w_1 > 0$ AND $w_2 > 0$	Two-sided	Table 2.4
($w_1 = 0$ AND $w_2 > 0$) OR ($w_1 > 0$ AND $w_2 = 0$)	One-sided	Table 2.4
$w_1 = 0$ AND $w_2 = 0$	No	(μ_1, μ_2)

Table 2.2: Values of outside option in COO

If $w_1 = 1$, or $w_2 = 1$ or $w_1 + w_2 \geq 1$, there stands no mutual benefit for dividing the cake, from the viewpoint of game-theoretic analysis. [4]

When $w_i > 1$, a player's outside option is larger than the size of cake, there is meaningless even to start the bargaining because player i will take his outside option anyway. We examine the first two situations in Table 2.2, in which at least one player has an outside option larger than 0 and no outside option is larger than the size of cake. We furthermore categorize the outside option, according to the values of outside options in Table 2.3 and 2.4.

Each outside option stands statically since the bargaining starts and until the game ends. When an offer $x_i \in (0, 1)$ is accepted at the time t (t is a non-negative integer), player i receives an utility $u_i = x_i\delta_i^t$ and the other player gets $(1 - x_i)\delta_j^t$. If one player i opts out the bargaining at the time t and takes his outside option, he receives the utility $w_i\delta_i^t$ and another player j will take her outside option and gets $w_j\delta_j^t$. Note that a player's outside option is discounted over time subject to the same discount factor as his share of cake. This ensures that players make decisions under time pressure [5]. If both players perpetually disagree and do not take their outside options, then both players obtain 0.

[4]This condition is examined in Chapter 7 to see whether an evolutionary algorithm is able to identify the over-strong threats from outside option and explores the reasonable solutions.

[5]In game theory literature, outside options in some bargaining models have no connection with discount factors.

Category Name	*Outside Option*
Ineffective Threats	C1
Effective Threats	C2
Over-strong Threats	C3

Table 2.3: Types of Outside Option in COO

Game-theoretic Solutions: Subgame Perfect Equilibrium

The unique *Subgame Perfect Equilibrium* (SPE) solution (x_1^*, x_2^*) of the outside option bargaining model is stated in the Table 2.4 where,

$$\mu_1 = \frac{1-\delta_2}{1-\delta_1\delta_2} \tag{2.7}$$

$$\mu_2 = \frac{1-\delta_1}{1-\delta_1\delta_2} \tag{2.8}$$

Note that $(\mu_1,\ \mu_2)$ is the Subgame Perfect Equilibrium for the corresponding bargaining problem CRub82 whose outside options are $(w_1, w_2) = (0, 0)$.

Whenever a player makes an offer or a counteroffer, he only asks for the SPE share x_i (*Stationarity*). We assume that player 1 is the player who makes the first move. His offer x_1^* is accepted by player 2 immediately (*No delay*). Player 2 gets a share $1 - x_1^*$ from this agreement (except #e in Table 2.4). So the shares for player 1 and 2 from the SPE agreement are $(x_1^*,\ 1 - x_1^*)$. All SPE agreements should be reached at the bargaining time $t = 0$. Therefore, player 1 and 2 obtain utility x_1^* and $1 - x_1^*$ respectively. For Condition #e, theoretically players take their outside options straightway because for at least one player, his outside option is more beneficial than any possible share from the bargaining.

#	x_1^*	*Conditions (AND)*		C.
		I	II	
a	μ_1	$w_1 \leq \delta_1\mu_1$	$w_2 \leq \delta_2\mu_2$	C1
b	$1 - w_2$	$w_1 \leq \delta_1(1 - w_2)$	$w_2 > \delta_2\mu_2$	C2
c	$\delta_2 w_1 +$ $(1 - \delta_2)$	$w_1 > \delta_1\mu_1$	$w_2 \leq \delta_2(1 - w_1)$	C2
d	$1 - w_2$	$w_1 > \delta_1(1 - w_2)$	$w_2 > \delta_2(1 - w_1)$	C2
e	w_1	$w_1 + w_2 > 1$	-	C3

Table 2.4: SPE under 5 different conditions for outside option bargaining problem. Player 1 makes the first offer. The shares in a SPE agreement is $(x_1^*, 1 - x_1^*)$ under the condition of $\#a, \#b, \#c$ or $\#d$. Under the condition $\#e$, $x_1^* = w_1$ and $x_2^* = w_2$.

2.4 Evolutionary Algorithms

This section maps the family tree of Artificial Intelligence, with the evolutionary algorithms (EA) focused. Two important methods in EA: GA and GP which are highly relevant to this thesis are explained in greater details.

2.4.1 Artificial Intelligence

"Artificial intelligence (AI) is defined as intelligence exhibited by an artificial entity" [Wik05b]. AI is a science dealing with intelligent behavior, learning and adaption in machines. Typical problems to which AI methods have been applied include patten recognition, natural language processing, non-linear control and robotics, computer version and virtual reality, game theory and strategic planning, game AI and computer game bot, and artificial creativity. Figure 2.3 overviews important methods in the AI family grouped by their characteristics. *Conventional AI* includes methods which are classified as machine learning [Mit97], featured by formalism and statistical analysis. Another branch, *computational intelligence* involves iterative improvement and/or learning [Wik05b].

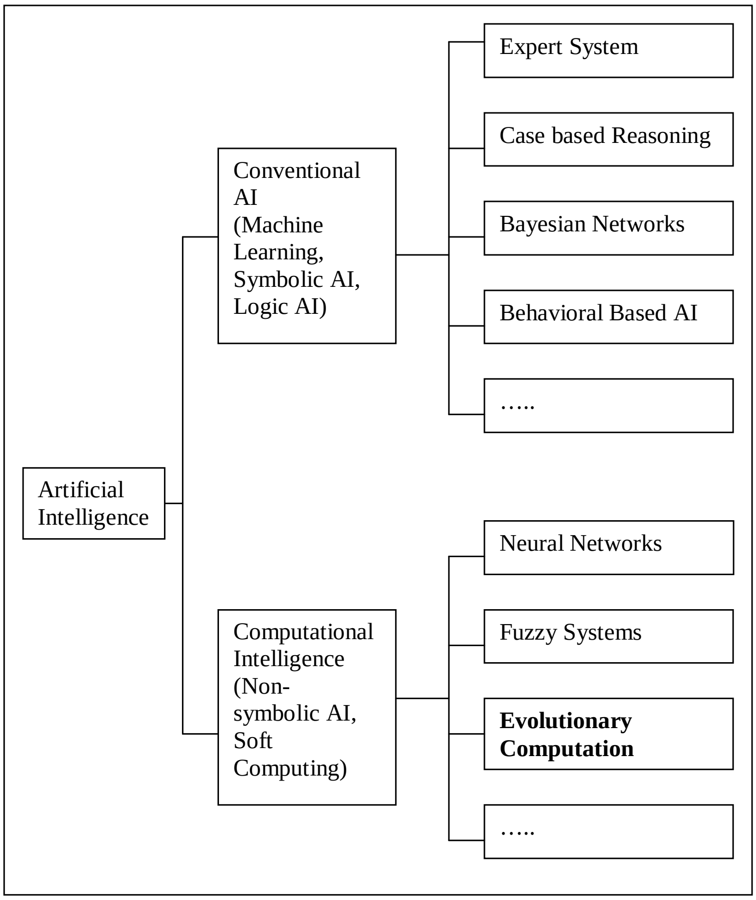


Figure 2.3: A simple family tree of Artificial Intelligence.

In this thesis, the main methodology to be applied is the Evolutionary Computation. Fogel reviews major contributions of simulated evolution to intelligence research [Fog99]. Evolutionary algorithms include a group of algorithms and methods, as seen in Figure 2.4. Among them, we pay special attention to Genetic Programming, one of Evolutionary Algorithms. Researchers also categorize methods according to their relationship to nature, see Figure 2.5.

Research on human beings provides insights to develop computational methods: "Biological systems such as human beings can be regarded as sophisticated information processing systems, and can be expected to provide inspiration for various ideas to science and engineering. Biologically motivated information processing systems can be classified into: brain-nervous systems (neural networks), genetic systems (evolutionary algorithms), and immune systems (artificial immune systems)." [Das05]

2.4.2 Evolutionary Algorithms - A Heuristic Search Method

Evolutionary Algorithms include a class of algorithms which benefit from the creative power of natural evolution, see Figure 2.4. Natural organisms populate the world through a process of reproduction (self-copy) and low-probability mutation. Mutation creates variation for exploring new genetic materials. The essences of natural evolution, namely selection, reproduction, variation and fitness measures are mechanized into evolutionary algorithms. Evolutionary algorithms are biologically, specially genetically, motivated information processing systems. It follows the general principles of natural evolution: the fittest survive under natural selection; the genetic materials of the new population emerge after recombination and variation of genetic materials of their parental individuals [Hol75]. Evolutionary algorithms feature heavily in the evolutionary simulation and have successfully solved many

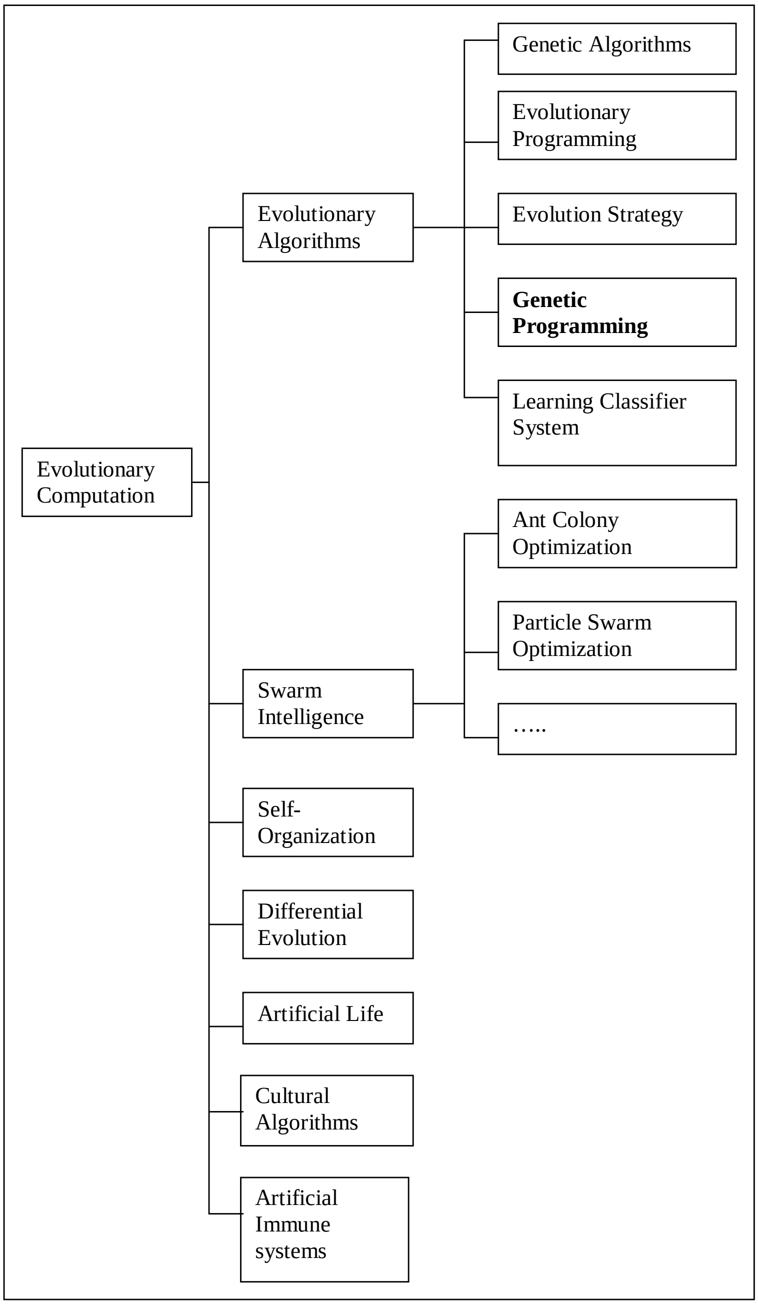


Figure 2.4: The family tree of Evolutionary Computation.

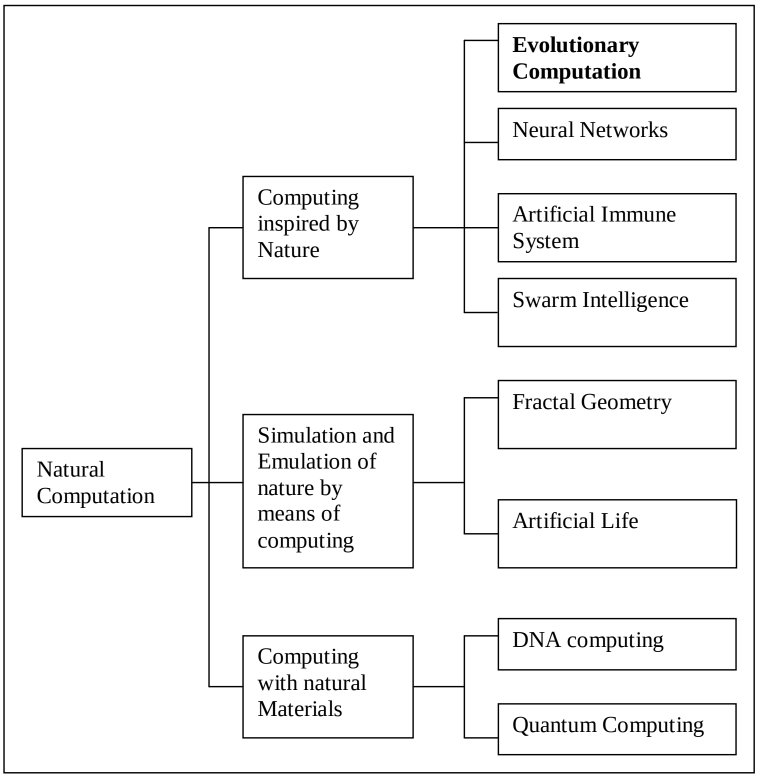


Figure 2.5: The family tree of Natural Computation.

classes of problems, including non-linear, epistatic, large search-space and multi-dimension problems [Gol89, Koz92, LP02].

Evolutionary algorithm is a population-based self improvement mechanism. One population consists of a set of individuals. Often individuals are candidate solutions to the problem. Individuals are selected based on their performance (fitness). Better individuals have higher probability to be selected as "raw material" to breed new offspring for the forthcoming generation. The offspring are then created by the genetic operators (crossover and mutation) from the "raw" genetic materials. Selective pressure pushes individuals (more specifically, the genetic materials) to continue improving their adaptation to the environments. Improvement of individuals' capability illustrates the process of acquiring behavior patterns adaptive to the environments [LP02]. [6]

[6]Although both are rooted in evolutionary biology, Evolutionary Algorithm is different from ESS in that

In many applications of game theory, the size of the search space is huge [7], thus exhaustive search is often impractical. An efficient way to search acceptable solutions within a reasonable time is therefore heavily in demand. Evolutionary algorithm is one of such methods. Evolutionary algorithms often serve as stochastic and heuristic search methods, which can shorten the search time. They are proposed to produce nearly optimal solutions. We apply EA to tackle bargaining problems, not only because they have succeeded in many other applications of games, but also because we aim to reuse the same mechanism to variations of bargaining scenarios.

Genetic Algorithms

Natural evolution is driven by natural selection, which favors individuals best fit to their environments, discovered by Charles Darwin's *The Origin of Species*. John Holland abstracts and formalizes the adaptation in natural and artificial systems. He presents *Genetic Algorithm* (GA) in [Hol62, Hol75]. Popular reading materials for studying Genetic Algorithms include [Gol89, BBM93a, BBM93b]. Problems which appear to be appropriate for using GA include timetabling, scheduling, optimization, high-dimensional problems and non-linear problems.

The following is the Pseudo-code algorithm for evolution inspired algorithms, for instance GA.

```
Create initial population Repeat
  Evaluate individuals' fitness
  Select pairs of individuals to reproduce
```

Evolutionary Algorithm is a way of simulation. The outcomes of simulations by EA do not necessarily converge to game-theoretic equilibriums or ESS.

[7] The size of search space also depends on how individuals are represented.

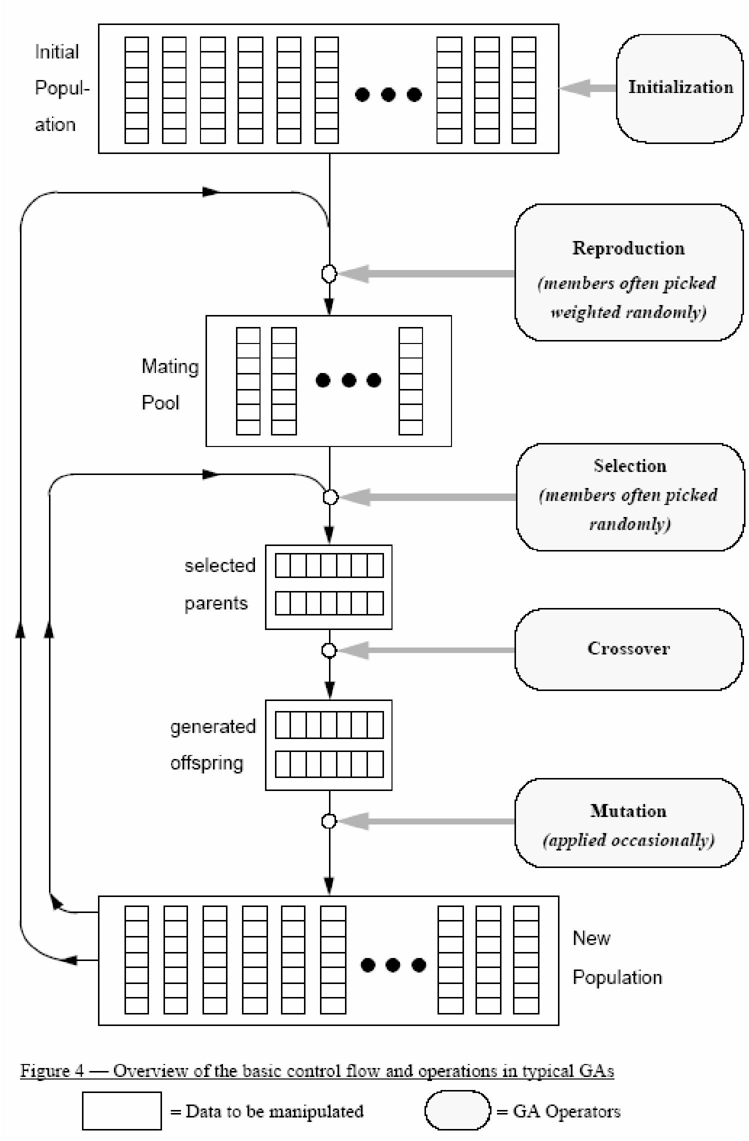
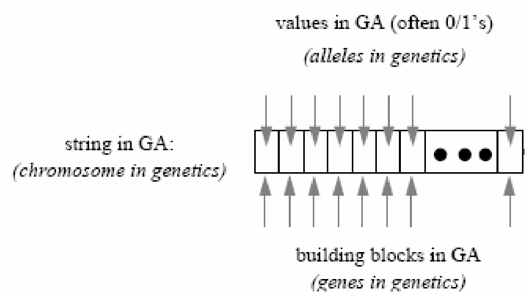


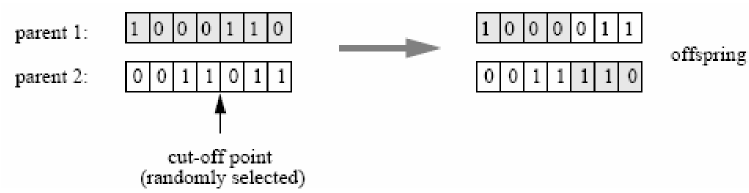
Figure 2.6: Overview of the basic control flow and operations in typical GAs [Tsa92]

```
    Breed new individual(s) through crossover and mutation
  Until terminating condition
```

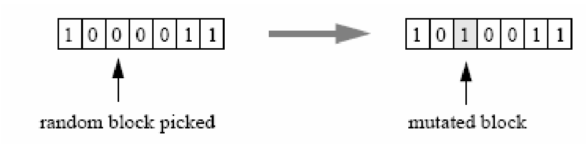
Figure 2.6 illustrates the basic control flow and operations in a typical GA. One species has a group of individuals; similarly one population in GA consists of a set of candidate solutions. The representation of candidate solutions in GA is normally in the format of strings with 0s and 1s. Crossover swaps parts of two selected strings to make new strings. Mutation randomly alters 1s to 0s or 0s to 1s, often with very low probability. Typical GA



GA: A typical representation of candidate solution



One-point crossover of two strings



One-point mutation

Figure 2.7: A typical way of representing a GA candidate solution with "0s" and "1s", of one-point crossover and of one-point mutation [Tsa92].

representation, one-point crossover and one-point mutation are shown in Figure 2.7.

The major difference between GA and GP lies in the representation of an individual. The concept of evolutionary procedure remains identical as shown in the Pseudo-code above. In the forthcoming subsection, GP is reviewed.

Genetic Programming

The very early experiments with Genetic Programming were reported by Stephen F. Smith 1980 [Smi80] and Nichael L. Cramer 1985 [Cra85]. Koza's works [Koz90, Koz92] popularize genetic programming. Individuals in Genetic programming are "general, hierarchial computer programs of dynamically varying size and shape" [Koz92]. Genetic programs are usually represented as syntax trees, unlike genetic algorithms which are represented as lines

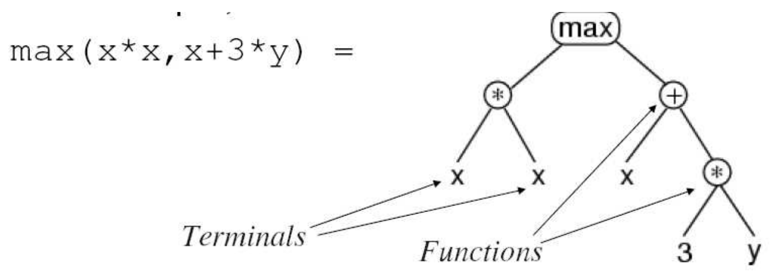


Figure 2.8: A syntax tree and the genetic program that it represents [LP02].

Type	*Examples*
Arithmetic	+, -, $\times$, $\div$
Mathematical	sin, cos, *log*, *exp*
Logic	AND, OR, NOT
Conditional	IF-THEN-ELSE
Looping	FOR, REPEAT
...	...

Table 2.5: Possible types of functions in a GP function set [LP02].

of code. Figure 2.8 is an example of a genetic program, representing $max(x \times x, x + 3 \times y)$.

Nodes of syntax trees can be categorized as either functions or terminals, defined in [Koz92]. A genetic program's *terminal set* consists of variables and constants as nodes. In the above example, the terminal set is $\{x, 3, y\}$. A genetic program's *functional set* consists of functions and operations which take members of the terminal set as input(s). The terminal set is $\{max, \times, +\}$ in the above example. Table 2.5 lists possible types of functions in GP and their examples. The sum of a GP's function set and its terminal set is its *primitive set*, $\{x, 3, y, max, \times, +\}$ in the above example. The search space of genetic programming is all possible computer programs composed of elements from its primitive set.

A typical one-point crossover of two GP programs is illustrated in Figure 2.9. An one-point mutation is illustrated in Figure 2.10.

Among many inventions of genetic programming, the most well-known and exciting one

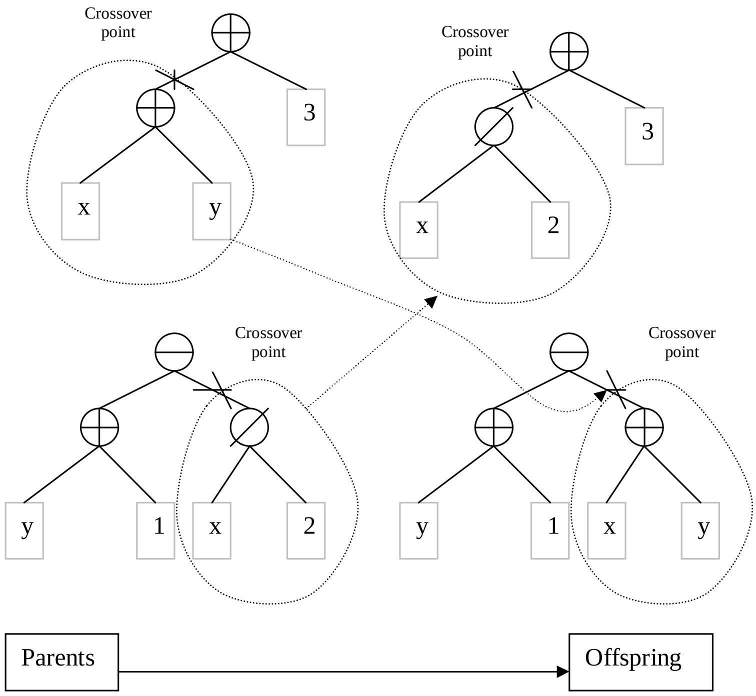


Figure 2.9: A typical one-point crossover of two genetic programs [Koz92].

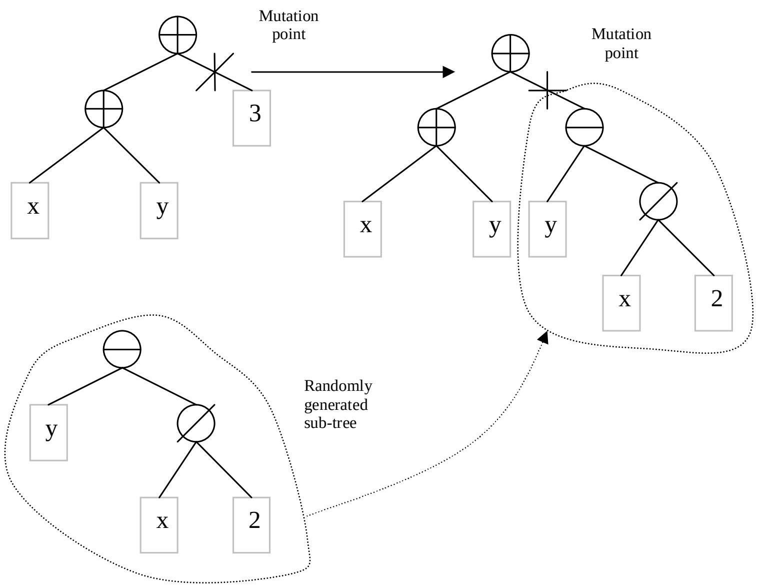


Figure 2.10: A typical one-point mutation [Koz92].

is that Genetic programming created an unique antenna for a NASA microsat experiment. It is the first "artificially evolved" object to be launched into space, according to NASA [8]. Many creations and patented inventions by genetic programming are detailed in [Koz92, Koz94, KIAK99, KKS+03].

2.4.3 Evolutionary Algorithm Simulations and Applications

A considerable amount of research has demonstrated that artificial simulations by evolution-inspired techniques can successfully provide ideal solutions for many games.

Axelrod [Axe87] starts the domain of interdisciplinary research of EA and games with his remarkable Genetic Algorithm experiments on the well-known game Iterated Prisoners Dilemma (IPD). According to [Axe87], the experimental results coincide with reciprocity phenomena found in the most competitive human entries to IPD competitions. Miller [Mil96] later uses automata to systematically restudy Iterated Prisoner's Dilemma. He builds a co-evolutionary [9] system for this problem. He concludes that cooperation is rather stable when there is no noise in the system and that the level of cooperation decreases as the noise increases. Gosling et al. [GJT05] study the one-population co-evolutionary system for IPD. [GJT05] compares population based incremental learning with guided mutation versus genetic algorithms on the iterated prisoners dilemma problem. [Koz92] employs co-evolution in conjunction with Genetic Programming to disclose minimax strategies for a two-player finite-size game in extensive-form under complete information. [LT99] by Tsang et al. successfully develops a co-evolutionary system EDDIE/FGP which aids investors to seek dealing opportunities in financial markets.

Lucas and Kendall's [LK06] reviews evolutionary computation and games. It demon-

[8] NASA feature article " 'Borg' Computer Collective Designs NASA Space Antenna" on 16 February 2006.

[9] The concept of co-evolution is introduced in the next subsection.

strates how computational intelligence makes entertainment games so fun to play. These games include classic board games such as chess, checkers and tic-tac-toe and real-time arcade and console games, such as Quake and Pac-Man. This article also explains why the involvement of evolution makes many games more interesting than otherwise.

2.4.4 Co-evolution

The original concept of co-evolution comes from nature. Biologists observe that, in nature, one species modifies itself to adapt to the changes of its co-existing species in their shared physical surrounding. Such modifications, in turn cause its co-evolving species to change themselves accordingly. This sort of reciprocal evolutionary changes in interacting species is known as *co-evolution* in biology.

Natural co-evolution is the mutually evolutionary influences between two (or more) species. The survival skills from co-evolving species in nature inspire computer scientists to borrow co-evolutionary principles to solve problems in which two dynamic elements interact with each other. Computer scientists formalize the co-evolutionary model [Sch04]. Figure 2.11 shows in an idealized two-species situation, the species A and B are co-evolving. One species's fitness is its current adaptation to the other species that is evolving simultaneously.

Holland [Hol92] initiates co-evolution for an artificial ecology system, "Echo". Co-evolution is also successful in many fields: factory organization [Lan99], robotics [KU99], predator-prey systems [Sim94, KU99, CM96], sorting networks [Hil90] and social sciences [Dor98].

2.5 Comparison with Related Works

This section compares and contrast our study with its closely related literature.

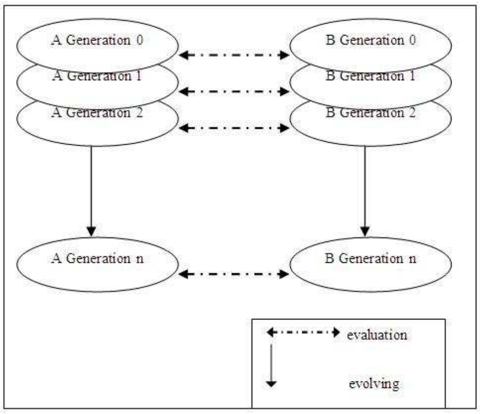


Figure 2.11: Two-species co-evolution

Axelrod [Axe87] uses genetic algorithms to simulate Iterated Prisoner's Dilemma. It has had a major influence on our work. The major similarities of our work to [Axe87] are:

- our work and [Axe87] both aim to understand players' behaviors in game-theoretic models;
- both works assume that players are boundedly rational, although [Axe87] does not explicitly specify this. Players improve their abilities through adaptive learning;
- both studies utilize artificial simulation, more specifically evolutionary algorithms to tackle the games.

The major differences between our work and [Axe87] are:

- Iterated Prisoner's Dilemma and bargaining games are dynamic games which can be split to sub-games. However, every subgame of IPD is a static game. Players make their decisions simultaneously. A sub-game of a bargaining game is not a static game. Only one player makes an action at one time [Mut99];

- [Axe87] assumes that two players are so identical that they take the same strategy even if they exchange their roles. So he evolves one population of strategies for both players. Our work assumes that two players in bargaining games are not identical. The first player in bargaining games may have a first-move advantage [Mut99]. Two players are not identical especially in incomplete information bargaining problems and outside options bargaining problem. Therefore, each player in our work has his own population of candidate solutions;

- [Axe87] concludes that TIT-FOR-TAT and its variants are the most competitive and stationary strategies. TIT-FOR-TAT is not the game-theoretic solutions for IPD. In contrast our work finds that evolutionary algorithms discover efficient and stationary strategies and/or approximate game-theoretic solutions.

Work of Braget, et al [vBGP02] shares a common interest with us in studying Rubinstein alternating-offers bargaining game. [vBGP00] simulates the bargaining by a multi-agent evolving system that is implemented by real number-coded genetic algorithms. Their experiments only test the situations when $\delta_1 = 0.6$ or $\delta_1 = 0.3$.

The major similarities of [vBGP02] and our work are:

- both [vBGP02] and our work study the classic bargaining model: CRub82 problem [Rub82];

- both use evolutionary algorithms, in particular two-population co-evolution;

- both works find that solutions from co-evolutionary algorithms approximate CRub82's game-theoretic solutions.

The major differences between our work and [vBGP02] are

- [vBGP02] uses genetic algorithms; we use genetic programming to implement the co-evolutionary system;
- [vBGP02] only measures the experimental results under $\delta_1 = 0.3$ and $\delta_1 = 0.6$; we cover the whole game-setting space $\delta_1 \times \delta_2 = \{0.1, 0.3, 0.5, 0.7, 0.9\} \times \{0.1, 0.3, 0.5, 0.7, 0.9\}$ with 25 evenly distributed game settings;
- besides CRub82 bargaining problem, we thoroughly study bargaining problems with incomplete information and outside options. We are in a better position to generalize our claims.

Fatima, Wooldridge and Jennings [FWJ05] examine incomplete information bargaining problems. The major similarities of our work to [FWJ05] are:

- our work and [FWJ05] both study incomplete information bargaining models, including Rubinstein incomplete information bargaining problem [Rub85];
- both consider multi-determinants: incomplete information and outside options.

The major differences between ours and [FWJ05] are:

- different research approaches: [FWJ05] mathematically proves the existence and uniqueness of equilibriums - using human intelligence. We apply evolutionary algorithms - using artificial intelligence;

- [FWJ05] takes three kinds of bidding functions into consideration. These three types of bidding functions are analyzed in Section 3.5. We choose the Conceder type of bidding function because the bargaining time pressure, measured by discount factor, forces bargainers to concede quickly in order to reach an agreement as soon as possible thus to reduce costs [10].

[10]Details are available in Section 3.5.

Chapter 3

Co-evolution to Tackle Alternating-Offers Bargaining Problem - Theoretic Framework and Design

3.1 Introduction

This chapter [1] aims to demonstrate how the Rubinstein [Rub82] alternating-offers bargaining problem, abbreviated as CRub82, can be tackled by an evolutionary algorithm, in particular the co-evolution algorithm.

CRub82 bargaining problem has one determinant on bargaining power, discount factor. Its game-theoretic assumptions and solutions are surveyed in Section 2.3.1 and is recapped shortly in Section 3.1.1. The computational complexity of this bargaining problem is analyzed in Section 3.2. This bargaining problem is so complicated that it is unlikely to be solved by an exhaustive search. A set of assumptions of bargaining players' bounded rationality sets up in Section 3.3.

The theoretic framework is established in Section 3.4. The system design and experiment design are in Section 3.5. Due to the complexity of the fitness function, this chapter focuses

[1] Most parts of this chapter have been published in [JT05a] and [JTng].

on the game fitness function which is a player's utility gained from playing a bargaining game. We leave the technical treatment for controlling constraints to Chapter 5.

Experimental results and observations are analyzed in the next Chapter.

3.1.1 Recapitulation of Alternating-Offer Bargaining Problem CRub82

Section 2.3.1 details the CRub82 bargaining problem. This subsection briefly recaps this problem and its game-theoretic solution.

CRub82 bargaining problem is a two-player dynamic game. Players have complete and perfect information about the game. Players' preferences are determined by their bargaining costs per time interval, termed as discount factors δ_1 for player 1, and δ_2 for player 2.

While bargaining, players' utilities are discounted over time, so both players have incentives to strike a deal as soon as possible. Player 1 starts bargaining over the divisible common interest (a cake) of the size 1 at the time $t = 0$. Two players make offers and counteroffers in a strictly alternating manner. An offer is either accepted immediately or rejected and after one time interval counter-offered. Once an offer x_i is accepted, an agreement is reached with a share x_i for player i and a x_j for player j, $x_i + x_j = 1$. If this agreement is reached at time t, the two players attain the utilities $u_i = x_i \delta_i^t$ and $u_j = x_j \delta_j^t$ respectively. The length of one bargaining period is one time interval Δ. To make things simple, we assume $\Delta = 1$.

In the unique *Subgame Perfect Equilibrium* (SPE), player 1 obtains:

$$x_1^* = \frac{1 - \delta_2}{1 - \delta_1 \delta_2} \tag{3.1}$$

$$x_2^* = 1 - x_1^* \tag{3.2}$$

The second player, player 2 obtains the rest of cake, $(1 - x_1^*)$.

3.2 Computational Complexity of CRub82 Bargaining Problem

CRub82 bargaining problem is intractable if an exhaustive search method is in use. One player can make infinitely possible offers. Moreover, from the definition of infinite-horizon, the bargaining procedure can last forever. To simplify this problem, we limit that a bargaining only lasts 10 time intervals at the maximum [2] and that any division of the cake is in 10^{-2} precision. Thus at any time, the offering or counter-offering player has 100 options (the cake size 1 is divided by 100) while the another player has two options: acceptance or rejection. So at a particular time, there are $100 \times 2 = 200$ possible outcomes. For 10 time intervals or 11 time points, there are $200^{11} \approx 10^{25}$ possibilities. If a machine tests possibilities at the rate of one possibility per nano-second (10^{-9} second), it requires 10^{16} seconds, or more than 200 million years to test all possibilities for an instance of CRub82 bargaining problem.

This bargaining problem has only one determinant, discount factor. Imagine when more bargaining determinants are taken into consideration while an exhaustive search is used, the complexity goes beyond the time constraint we can ever afford. On the other hand, game theorists spend years to solve such a problem [3]. Game-theoretic method requires excessive human efforts and expertise. Therefore an alternative method which is more efficient is heavily in demand. Artificial intelligence as a complementary method of human reasoning, in particular evolutionary algorithms as a heuristic search method, is hoped to generate fairly good solutions for bargaining problems within affordable time and computational expenses.

[2] The reasons of using 10 time intervals limitation are explained in Section 3.5.

[3] Rubinstein published the theoretic solution for CRub82 bargaining problem in 1982 [Rub82]. He later extended the CRub82 bargaining problem to the incomplete information bargaining problem in which one of two players has incomplete information. He published the game-theoretic solutions for the incomplete information bargaining problem in 1985 [Rub85]. We can not therefore conclude that he spent three years to solve one extension of CRub82 bargaining problem but can infer that it is not easy for a game theorist to work out game-theoretic solutions.

We will demonstrate the use of evolutionary algorithms on bargaining problems.

3.3 Assumptions of Players' Boundedly Rationality

In most of game-theoretic models, all players are perfectly rational "economics men". For the CRub82 bargaining problem, its game-theoretic assumptions imposed for obtaining Subgame Perfect Equilibrium are subject to perfect (unbounded) rationality hence other equilibriums and possibilities, which may arise in more realistic circumstances are ruled out. The perfect rationality and bounded rationality are reviewed in Section 2.2.

The beauty of perfect rationality is seldom seen in the real world. Boundedly rational players are common. It is of great interest to see whether players with bounded rationality also prefer SPE. Ordinary players often enable to learn from experiences and surely aim to maximize their utilities if possible. Individuals' adaptive learning modelled in evolutionary algorithms match these characteristics. The desire of assuming bounded rationality on bargaining players' behaviors is another main reason for using evolutionary algorithms.

This thesis does not intend to study human decision making in terms of their economic and social implications. Instead, we limit our attention to artificial simplification of humans. These artificial players are boundedly rational and much simpler than human beings. They are equipped with simplified adaptive learning ability. The learning mechanism that we will use is an evolutionary algorithm which is supposed to capture essentials of human adaptive learning at an abstract level.

This section specifies the assumptions on players' behaviors under bounded rationality. In fact, any assumption that is not perfect rationality, can be regarded as a form of bounded rationality. This set of assumptions is applicable to bargaining problems in this thesis. The mathematical descriptions of these assumptions and the theoretic framework are detailed in

Section 3.4 and 3.5.

A player can have many strategies. We assume that players learn on the basis of evolutionary algorithms: at beginning players offer, reject or accept partitions of cake randomly and after evolutionary process they offer, reject and/or accept such partitions that can be quickly accepted by the other player. Please note that in this work, a strategy does not learn. In terms of evolutionary algorithms, a player can be thought as a population full of candidate solutions (strategies in games). Commonly, in evolutionary algorithms, candidate solutions (also called individuals) in a population at a specific generation are static, which do not learn or change within that specific generation [Gol89] [Koz92]. It is two strategies, each from a player that play instances of the game. Table 3.2 gives an example of two strategies playing a bargaining game.

- **A-1**: A player tries to maximize his utility. A perfect utility maximizer in game-theoretic settings reasons its strategies not only on known utilities but also on players' preferences and other player's possible reactions [Rub85]. Unlike a perfect utility maximizer, the boundedly rational players use such strategies that merely take the option with the 'superficially' highest utility. A player's strategy does not do any further deliberation, such as possible delay and the possibility of the acceptance of the offer by the other player's strategy. Provided with two options of action α and β with $u(\alpha)$ and $u(\beta)$ as i's utilities of α and β respectively, player i' strategy chooses the action that rewards larger utility: if $u(\alpha) > u(\beta)$, this strategy takes α. If $u(\alpha) = u(\beta)$, it takes the action which makes an agreement sooner. Equation (3.5) in the next sub-section instantiates this assumption.

- **A-2**: A player' behaviors not only respond to but also influence the other's behaviors over evolutionary learning time. This means that a player's best strategy is determined by what strategies the other player has. Co-evolution captures this interdependent relationship between two players' strategies. A strategy' performance is explicitly expressed in the fitness function of the co-evolutionary framework as in Table 3.1.

- **A-3**: Players learn through trial-and-error training. Players improve their strategies' performance through playing bargaining games with the other player's strategies. Better strategies are reused or modified then reused with higher possibility. *Definition 3. State Transaction Equation* in the next section mathematically defines the process of players' learning and improvement.

- **A-4**: Players are assumed to be equipped with almost the same set-up. In terms of evolutionary algorithms, a player's strategy representation, operators and parameters of evolutionary algorithms are the same for both players. The system set-up of CRub82 bargaining problem is detailed in Section 3.5 and Table 3.3.

The following assumptions **A-5** to **A-8** clarify what players are unable to do:

- **A-5**: Players' strategies are incapable of game-theoretic reasoning. The strategy representation in Equation (3.4) and (3.5) of Section 3.5 shows that players' strategies simply compare the utility from the current offer versus the possible utility from the counter-offer one time interval after, and then take the action which rewards the player higher utility (as **A-1**). The ability of telling the larger one from two values is far simpler than the game-theoretic reasoning.

- **A-6**: A player's strategy is unable to identify the other player's strategies. From the strategy representation in Equation (3.5), when player i's strategy is trying to make a decision, it only measures its player's possible utilities from his own bidding function $b(g_i)$ and from x_j offered by other player j's strategy. Player i's strategies have neither information of j's strategies nor intention to identify j's strategies. Player i's strategies are unable to predict what j's strategies will do if player i's strategies accept or counter-offer.

- **A-7**: Player i's strategy has no memory of its own moves and the moves by player j's strategy. A strategy does not take actions of its own and of the other's strategy in past into consideration. As in Equation (3.5) of strategy representation, $t-1$ is not a component, which means that when player i's strategy makes a decision, it never considers his own actions nor j's actions [4].

- **A-8**: A strategy can not modify itself to respond the other strategy' actions during one bargaining encounter. Equation (3.5) clearly shows that a player i's strategy $s(g_i)$ is defined before the bargaining starts and remains unchanged throughout the bargaining encounter. It also shows that i does not change strategy $s(g_i)$ whatever action $s(g_j)$ takes.

- **A-9**: In this work, adaptive learning refers to the learning in the time frame of species-level, not the learning during the life-time of a single individual. Learning in an individual (strategy)'s life-time is theoretically feasible and is seen in playing games in reality. In this thesis, we assume that players learn. From the game-theoretic prospect, a bargaining agreement is inefficient if it is settled at any time $t > 0$. Strategies that make agreements very soon have higher utility and thus higher fitness so they are

[4] In literature, IPD players in [Axe87] [DY94] [GJT05] have the memory of the last three moves of both players. IPD players try to infer the other's strategy from the other's actions in preceding moves.

more likely to survive. In experiments we set the maximal bargaining time as 10 time intervals. Strategies do not know this maximal time and they play games as if the game can last forever. *Definition 3. State Transaction Equation* formulates players' learning mechanism. The upgrading of genetic materials and of strategies only happens at the time from one generation to another. Equation (3.5) further confirms that no learning ever conducts during the life time of a strategy. Further discussions on the learning time frames are available in [Mit97]. Learning in an individual's life-time is interesting, but is not studied in this thesis.

This set of assumptions is very simple in terms of intelligence. In short (1) a strategy is able to choose the better one from two choices provided with the utilities of these choices; (2) a player with a set of strategies adaptively learns on the basis of evolutionary algorithms.

Other bargaining problems in this thesis are more complicated than CRub82, in the sense of bargaining scenarios, bargaining determinants and players' information. Therefore assumptions **A-1** or **A-4** may need modifications for other bargaining problems. Such modifications only have marginal effects on the implementation of the system; but they have almost no impact on the theoretic framework.

3.4 Theoretic Framework of Co-evolution

In CRub82 bargaining problem, both players try to deploy the best response to the other's strategy. Both are learning how to find the best response through playing this bargaining game with the other. This co-adaptive learning relationship fits well with dynamics in co-evolutionary algorithms. The theoretic concept of co-evolutionary algorithms is reviewed in Section 2.4.4.

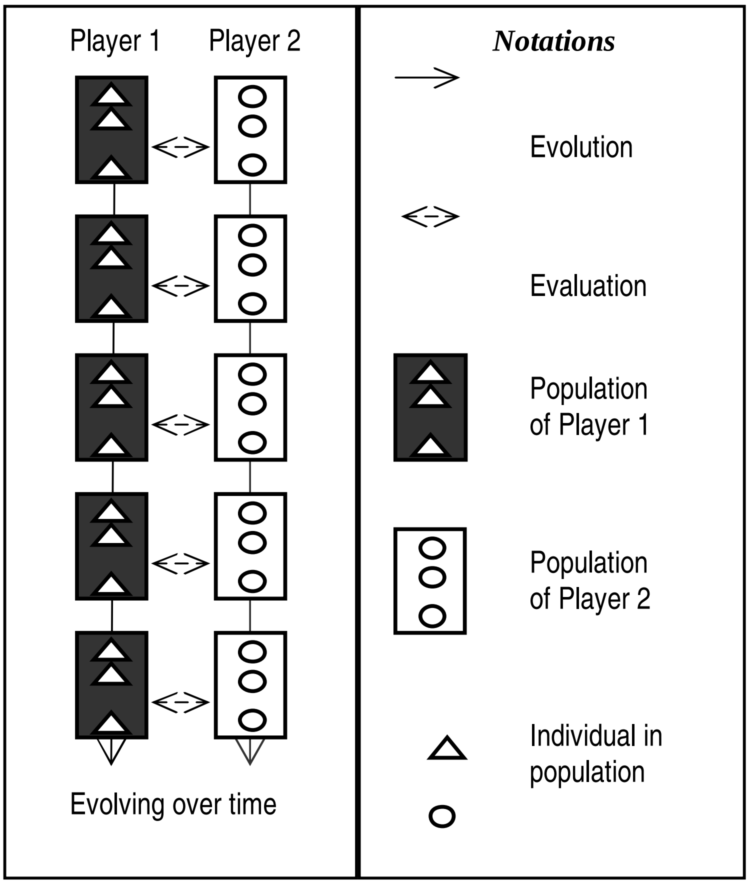


Figure 3.1: A Simple Two-population Co-evolution

3.4.1 A Learning Problem

Machine learning broadly speaking "includes any computer program that improves its performance at some task through experience" [Mit97]. Learning of artificial players in bargaining games under Assumptions **A-1** to **A-9** is a kind of machine learning.

We now define a machine learning problem, assuming that bargaining players learn from experiences with respect to the task and performance measure.

- *Task*: Player 1 and player 2 play a bargaining game;
- *Training Experience*: Player 1 and player 2 play such a bargaining game;
- *Performance Measure*: an important measure of a player's ability is his utility from agreements, i.e. his gains from the mutual benefit. Efficiency and stationarity of agreements are game-theoretic properties. So they are also important measures.

3.4.2 Training and Learning through Co-evolution

We design a co-evolutionary system. In this system, each player has a population of individuals that are strategies for the player. The two players' strategies evolve simultaneously. The strategies aim to maximize their player's utility from agreements. Individuals (strategies) in a population independently undergo selection based on their performance (fitness). Better performed individuals have higher probability to be taken as "raw materials" which are genetically modified in order to breed new individuals for the forthcoming generation. Newly created individuals bargain with individuals in the updated co-evolving population that has gone through a similar evolutionary process. This system simulates the trial-and-error learning and co-adaptation specified in Assumption **A-2** and **A-3**. During one generation, there is nothing changed for improvement. Learning develops not in an individual's life time but in the species (population)'s evolutionary time, as assumed in **A-9**. The co-evolving process is illustrated in Figure 3.1.

Definition 3.1. Populations of Co-evolution. A population P is a set of individuals or candidate solutions for that player. P_1 is the population for player 1. P_2 is the population for player 2. Two populations P_i and P_j directly interact in such way that individuals in one population are assessed dependently on individuals in the other population.

The sequence of training examples are controlled by designer not by learners. Playing bargaining games provides direct training examples. Training examples are identical to the test examples: CRub82 bargaining game, over which the training is later to be tested.

Training is to give appropriate *responses* or feedbacks to *guide* subjects to the *target* [5]. To achieve the goal of training, the design of an appropriate response is the key. In the co-evolutionary learning framework, appropriate responses refer to appropriate fitness functions; the target refers to the state of co-adaption; and the guiding force is the principle of evolution, in particular the selective pressure. The next subsection details the fitness functions in co-evolutionary systems.

3.4.3 Relative and Absolute Fitness in Co-evolution

This section analyzes how to measure a player's adaptation (fitness) in co-evolution.

In evolutionary algorithms, individuals are evolving against static objectives. The *absolute fitness* [Koz92] (also called objective fitness) evaluates the fitness upon static targets. Using the absolute fitness is implicit in applications of evolutionary algorithms where the

[5] "The law of effect is a principle of psychology described by Edward Thorndike in 1898. It holds that responses to stimuli that produce a satisfying or pleasant state of affairs in a particular situation are more likely to occur again in the situation. Conversely, responses that produce a discomforting, annoying or unpleasant effect are less likely to occur again in the situation. The law of effect [Her70] is important in understanding learning, especially as it relates to operant conditioning. " [Wik06b].

objective function is assumed to be identical over all generations, independent of the evolutionary time.

In most of applications of the co-evolutionary algorithms, absolute fitness functions usually are unavailable. For example, in the case of two-player bargaining games, the set of a player's possible strategies is huge in size. Any strategy cannot be judged properly in absence of the other player's behaviors. The fitness of a player's strategy depends on the other player's strategy. Because of the difficulty of discovering underlying objectives for a class of problems, researchers turn to use co-evolution to solve problems for which static objectives are unknown, whereas a kind of "reciprocally interactive" relationship between/among species (players in games) can be assumed. Koza [Koz92] terms *relative fitness*. It is also called "subjective fitness" which measures how much co-evolving species adapt to each other. de Jong and Pollack [dP04] examine fitness measure in co-evolution in great details.

Relative fitness functions are dynamic, updating over evolutionary time. In other words, individuals in different generations are evaluated by probably different objectives. As we assume, bargaining strategies are learning and improving their performance against dynamic objects (the other player's strategies), therefore their fitness should also be dynamic. The relative fitness measure is such a dynamic assessment subject to co-evolving population(s). Applications have shown that co-evolutionary algorithms using relative fitness functions are successful as surveyed in Section 2.4.4.

Researchers pay less attention to the use of absolute fitness functions in studying co-evolutionary algorithms. The property of absolute fitness attracts our attention to investigate whether the absolute fitness can behave as a monitor of a co-evolutionary system. Absolute fitness may provide information concerning the co-adaptive process.

To illustrate these two fitness concepts, “the classic analogy is the co-evolutionary arms race (in nature): a plant species has chemical defenses and an insect species evolves the biochemistry to detoxify these compounds. The plant in turn evolves new defenses that the insect in turn ‘needs’ to further detoxify” [Koz92]. The (relative) fitness of the insect depends on the evolutionary state of the plant, so the relative fitness directs the insect to adjust its behaviors to detoxify the plant’s *current* chemical defenses. From the insect’s and the plant’s relative fitness, we know the degree of co-adaptation of these two species at certain time. To discover how the insect adapts to the plant progressively, it is necessary to investigate the insect’ biochemistry (the insect’s absolute fitness) at every stage of the co-evolution.

In nature it is the relative fitness that motivates the co-evolutionary improvement. In the plant-and-insect example, to detoxify the plant’s present chemical defenses is what the insect tries to do now. On the other hand, the absolute fitness records the co-evolutionary history: at different evolutionary times, the insect has different biochemistry materials. It is unlikely to transform a relative fitness into an absolute one or vice versa, because the relative fitness is always a function of time and is generation-dependent, but the absolute fitness is not. Only if the time is frozen at a certain moment, can the relative fitness at that time be also interpreted as the absolute fitness at that time.

Recent researches disclose that under certain conditions, the absolute fitness may provide the same information as the relative fitness. de Jong and Pollack [dP04] have shown that a complete evaluation set exists which provides ideal evaluation, meaning objective evaluation. If one could evolve against a complete evaluation set (CES), this would have the same effect as having a single, fixed, objective fitness function as in normal GA’s. However, the question is how to find a CES. The delphi algorithm *approximates* the CES. Meanwhile, there are new algorithms such as the Nash memory and IPCA which guarantee monotonic progress for co-evolution. One purpose of introducing this ideal evaluation is to avoid inaccuracy in co-evolutionary algorithms. Luke and Wiegand [LW03] argue the possibility of existing an objective measure that may make evolutionary algorithms exhibit similar dynamics and generates similar results to a single-population co-evolution. Our work tries to analyze the absolute fitness and relative fitness, and furthermore to understand the behaviors of co-evolving strategies and their adaptive learning in a bargaining scenario.

Two studies on the well-known controversy of Iterated Prisoners' Dilemma are good expositions of how an absolute fitness function and a relative fitness function can generate different results for the same problem. Axelrod's GA experiments [Axe87] evolve player's strategies against a fix environment, "eight representatives". The set of these eight chosen representatives is an absolute fitness function that is independent of the evolutionary time [6]. His GA experimental results support the claim that TIT FOR TAT and its variants are the best responses to IPD problem. However, this claim is questioned by the results from co-evolutionary experiments where a relative fitness function is adopted. Darwen and Yao [DY94] follow the strategy representation by Axelrod's [Axe87] , but the fitness of a strategy is the scores it achieves from playing IPD against all the other strategies in its population. This set-up is a typical one-population co-evolution with the use of a relative fitness evaluation. At the end of their experiments, "only cooperative strategies survive" [DY94]. Moreover some behavioral patterns claimed to be parts of TIT FOR TAT such as "Be provocable" and "accept a rut" which play defections, have not been discovered in [DY94].

[6] [Axe87] uses eight representatives rather than a co-evolving population as the fitness evaluation was probably a consequence of the very expensive computational cost in 1980s.

	Player 1's Fitness Function	*Player 2's Fitness Function*
$P_1 - P_2$	$f_1(g_1, P_2(n))$	$f_2(g_2, P_1(n))$

Table 3.1: Fitness Functions of the two-population Co-evolution

3.4.4 Relative Fitness Evaluation for Two-Population Co-evolution

We now formalize the relative fitness in co-evolution. Suppose in a simple co-evolutionary system, two species exist in a stable physical environment. The two populations (P_1, P_2) are the sets of individuals of these two species which are simultaneously co-evolving over evolutionary time. Assume these two species start evolving at the same time and spend exactly same length of time per generation.

For each population in the two-population co-evolution, we adopt the relative fitness assessment. Thus the evaluation of individuals' performance is subject to co-evolving objects, not to static one(s).

Definition 3.2. Formalization of Fitness Functions of two-population Co-evolution. We denote the relative fitness functions of species i and of species j as f_i, and f_j respectively. g_i is an individual in the population for species i. The non-negative integer n is designated to the evolutionary time.

In this co-evolutionary system the relative fitness function f_i of an individual $g_i \in P_i$ is a function of g_i and of the state of the other population P_j: $f_i(g_i, P_j)$. If the generation is specified, we get: $f_i(g_i, P_j(n))$. It simply tells that the individual g_i's relative fitness depends on the evolutionary time n and P_j that is changing over time. In Table 3.1, $f_2(g_2, P_1(n))$ is interpreted as: the relative fitness of species 2's strategy g_2 is determined by the state of the population $P_1(n)$ at the same generation, in other words, by species 1's individuals at time n.

Unlike the relative fitness function f_i, the absolute fitness function f' evaluates an individual g_i not in connection with the co-evolving P_j but upon a static objective $f'(g_i)$ which is independent of the time n.

The selection operates on the basis of individuals' performance. The recombination of selected individuals produces new offsprings. The formal definition of the state transaction from one generation n to the next generation $n+1$ of the two-population co-evolution is as follows.

Definition 3.3. State Transaction Equations of the two populations from n to $n+1$. Var is the variation operation in evolution (mutation and/or crossover). Sel is the selection operation. Then the state transaction equation of two-population co-evolution is:

$$\begin{vmatrix} P_1(n+1) \\ P_2(n+1) \end{vmatrix} = \begin{vmatrix} Var(Sel(P_1(n), f_1(P_2(n)))) \\ Var(Sel(P_2(n), f_2(P_1(n)))) \end{vmatrix}. \tag{3.3}$$

It is interpreted as: $P_1(n)$ at generation n are selected on the basis of their relative fitness against individuals in $P_2(n)$ at the same generation n: $Sel(P_1(n), f_1(P_2(n))$. After selection, mutation and crossover are conducted. This is a generic expression for a two-population co-evolution. The coming section instantiates this expression for the CRub82 bargaining problem.

One-population co-evolution for CRub82

We have considered *one-population Co-evolution* for CRub82 bargaining problem, because of the symmetric property of players' information and of the game-theoretic solution SPE. Players then are assumed to use the same strategy if they swap their roles. Thus strategies could be identical for both players. In the one-population co-evolutionary system, one population consists of candidate solutions suitable for both players. We have executed experiments to examine the effectiveness of both the one-population system and the two-population system for CRub82 problem. The experimental study on CRub82 problem by the one-population co-evolutionary system is reported in Appendix C. Experimental results convince us that the two-population system generates more effective results than the one-population system. Moreover, the two-population system enables different strategies to be evolved for different players. The first player may have a first-move advantage [Mut99] so that player 2's strategy may not be suitable for player 1. Moreover, the two-population co-evolutionary system is extensible for more realistic situations: for example, one player has more information relevant to the game than the other. So the two-population system as a more general framework for two-player bargaining problems is finally chosen.

3.5 System and Experiment Design

This section specifies the co-evolutionary system for CRub82 bargaining problem with respect to strategy representation, fitness function, GP set-up, the system's parameters and the game settings.

3.5.1 Strategy Representation

An individual $g_i \in I$ is a candidate solution in player i's population I. GP trees could be evaluated once prior to the start playing an instance of game given the discount factors (δ_1, δ_2). For example, assume the genetic program $g_1 = \delta_1 - (1 - \delta_2)$. If $(\delta_1, \delta_2) = (0.9, 0.5)$, the $g_1 = 0.4$.

Although SPE does not explicitly involve time, time is an important element in any bargaining. We define a time-dependent bidding function: player i bids $b(g_i)$ at time t:

$$b(g_i) = g_i \times (1 - r_i)^t \tag{3.4}$$

where t is the bargaining time, a non-negative integer. r_i is the discount rate $\delta_i \equiv exp(-r_i)$. So in the example of $g_1 = 0.4$, this strategy will offer 0.4 at $t = 0$ and accept player 2's counter offers if they are larger than $0.4 \times (1 - r_1)^t$.

The part of $(1 - r_i)^t$ guarantees that players bid decreasing offers and counteroffers while time elapses. This is a kind of conceder function. We have considered linear, boulware and conceder types of strategies. [FWJ05] discusses these three types of functions: in the boulware type of function "the initial offer is maintained until time is almost exhausted, when the agent concedes up to its reservation value". Using the conceder type of function, "the agent goes to its reservation value very quickly". Using the linear type of function, shares in offers are increased linearly. Conceder type is more practical for the CRub82 bargaining problem than the other two, because both players are under the exponentially increasing pressures of bargaining costs over time measured by discount factors. Figure 2.1 in Section 2.3.1 clearly illustrates such exponential increase of bargaining costs. Thus while bargaining time goes, players are more eager to make an deal. Eagerness forces players to increasingly concede on their shares over time. The conceder function $(1 - r_i)^t$ captures this property of the discount factor δ_i.

A strategy determines what action (acceptance or making a counter- offer) a player takes at time t. g_i's corresponding strategy is $s(g_i)$. $s(g_i)$ accepts or rejects an offer from player j on dividing the cake as $(1 - x_j, x_j)$ at time t:

$$s(g_i) = \begin{cases} \textit{accept}: \\ \qquad \text{if } (1-x_j)\delta_i^t \geq b(g_i)_{(t+1)}\delta_i^{t+1} \\ \textit{counteroffer at } (t+1): \\ \qquad \text{if } (1-x_j)\delta_i^t < b(g_i)_{(t+1)}\delta_i^{t+1} \end{cases} \tag{3.5}$$

When player i with strategy $s(g_i)$ receives an offer $(1-x_j)$ from the other player j who asks for x_j as her share, player i compares the utility $(1-x_j)\delta_i^t$ of this offer $(1-x_j)$ with the possible utility that he expects if his counter-offer $b(g_i)_{(t+1)}$ is accepted at time $t+1$. If the utility from the later choice is not higher than that from the former choice, the offer of the former choice is accepted by player i. Just as the assumption **A-1** states, players try to maximize their utilities. This strategy expression also conforms to the assumptions **A-5**, **A-6**, **A-7** and **A-8**. $s(g_i)$ has neither the ability of game-theoretic reasoning nor a memory storage. It cannot identify the other player's strategies nor change its own strategy during a bargaining encounter.

An example of two strategies $s(g_1)$ and $s(g_2)$ playing a bargaining game is give as in Table 3.2. The discount factors $\delta_1 = 0.9$ and $\delta_2 = 0.7$. The superscripts indicate the order of actions by the two players. This example shows that two strategies make a division of cake as player 1 gets a 0.7337 slide of the cake and player 2 gets 0.2663. Because the agreement is reached at time 3, so the utilities of two strategies obtained are 0.5348 and 0.0913 respectively.

3.5.2 Genetic Programming Set-up

There are a few algorithms in the family of evolutionary algorithms, as shown in Figure 2.4. Genetic Programming is not the only way to implement the design of the two-population co-evolutionary system. We choose it because it copes well with the representation of functions and variables.

Bargaining Time $t =$	$b(g_1)$	$s(g_1)$'s action	$b(g_2)$	$s(g_2)$'s Action
0	1.0000	ask 1.0000 [1]	1.0000	reject [2]
1	0.8946	reject [4]	0.6433	ask 0.6433 [3]
2	0.8004	ask 0.8004 [5]	0.4139	reject [6]
3	0.7161	accept: GAME OVER [8]	0.2663	ask 0.2663 [7]
4	0.6406		0.1713	
5	0.5731		0.1102	
6	0.5127		0.0709	
7	0.4587		0.0456	
8	0.4104		0.0293	
9	0.3671		0.0189	
10	0.3285		0.0121	

Table 3.2: An example of two genetic programs g_1 and g_2 playing an instance of CRub82 bargaining game. $\delta_1 = 0.9$ $\delta_2 = 0.7$. The superscripts indicate the order of actions.

Firstly the game-theoretic solutions SPE consists of variables and arithmetic functions. We propose to compare the experimental results with the game-theoretic solutions. The structures of experimental results consisting of variables and functions therefore are preferred. GP fulfills this purpose.

Another reason GP is chosen is that GP implementation can have more extensions on various bargaining problems. In CRub82 bargaining problem, there are two game variables δ_1 and δ_2. In ICRub85 incomplete information bargaining problem, one of these two variables has two possible values. In the outside option bargaining problem four game variables need to be handled. We plan to establish a general system which should manage more variables. This demand favors the choice of GP.

GP Operators and Parameters

GP operators (selection method, crossover and mutation methods) and parameters (for instance, function set, terminal set, population size and number of generation) are stated in Table 3.3.

An individual in a population is a genetic program: g_i for player i, $i \in \{1, 2\}$. g_i is constructed with the function set {+, - , $\times$ and $\div$ (protected) } and the terminal set {1, -1, δ_i, δ_j }. Players both have complete information relevant to CRub82 bargaining game, therefore their information is $\{\delta_1, \delta_2\}$. Added the size of cake 1 and the -1 to change the sign, the terminal set for g_i is $\{\delta_1, \delta_2, 1, -1\}$. We choose arithmetic functions $\{+, -, \times, \div\}$ as the function set, because firstly they are fundamental and secondly, SPE is expressed arithmetically. So arithmetic functions should be sufficient for CRub82 problem. By changing the function and terminal sets, we could create different g_is. For example, if adding time t and $if - then - else$ into the terminal set, we would evolve the overall strategies $s(g_i)$.

There are more than one feasible way to represent genetic programs for solving this bargaining problem. We only evolve g_i in most parts of this thesis. We have tried to include the bargaining time t into the terminal set and evolve the overall strategy. The experimental results of evolving the overall strategies are assessed against those results of evolving g_i. Details of evolving the overall strategies are reported in Appendix D. Appendix D justifies our choice of evolving g_i. It is clear that when the variable t is also a member of the terminal set in genetic programs, the evolutionary stability is difficult to maintain under the same computational resources. It is probably because adding t into the terminal set expands the search space dramatically. t is not one variable but a vector of $[0, 10]$. Moreover, experimental results by using GP with t in its terminal set display less rational behaviors than those by

GP parameters	*Values*
Terminal set	$\{\delta_1, \delta_2, 1, -1\}$
Functional set	$\{+, -, \times, \div\}$ ($\div$ is Protected)
Population Size	100
Number of Generations	300
Initial Max Depth	5
Maximum nodes of a GP program	50
Initialization Method	Grow
Selection Method	3-member Tournament
Crossover Method	[Koz92]
Crossover rate	(0, 0.1)
Mutation Method	Sub-tree Mutation
Mutation rate	(0.01, 0.3)

Table 3.3: Summary of the Genetic Programming Parameters and Operators. The two populations have the exactly same GP set-up for solving CRub82 bargaining problem. For other bargaining problems, the terminal sets of two populations are not necessarily the same.

evolving g_i. We explain the reasons of such results in Appendix D.

"The depth of a tree is defined as the longest non-backtracking path from the root to an endpoint. " [Koz92]

The initialization method used is the "grow method". We select one function from the functional set at random to be the root of the tree. To select a function as the root guarantees that the tree consists of more than one nodes [Koz92].

When a node in a tree is labelled with a function, an element from the primitive set (from the functional set or from the terminal set) is randomly selected to be one endpoint of this node (for example, the root). According to the members in the functional set in this thesis, each function takes two arguments as its endpoints. An argument can be a member from the terminal set or from the functional set. If a member of the terminal set is selected as the endpoint of a functional node (for example, the root). The terminal node has no further arguments as its inputs [Koz92].

This process recursively grows a tree from left to right until the maximum depth reaches or until all endpoints are nodes from the terminal set. The grow method generates trees have variably shaped, under the constraint of the maximum depth.

We use 3-member tournament selection. A group of 3 random individuals is created. The individual with the highest fitness in the group is selected, the others are discarded (tournament).

We adopt the method of crossover in [Koz92], as seen in Figure 2.9. In sub-tree crossover, we randomly select a crossover point in each of two parent trees and swap the sub-trees rooted at the crossover points. The crossover points are selected 90% of the times at functions and 10% of the times at the nodes. [LS97] presents a systematic study on the effectiveness of this crossover method. [LP02] introduces and discusses another type of crossover where only sub-tree with same depth can be swaped.

Sub-tree Mutation: Mutation randomly selects a mutation point in a tree and substitutes the sub-tree rooted there with a randomly generated sub-tree. Mutation is sometimes implemented as crossover between a program and a newly generated random program (headless chicken crossover). We make sure that the randomly generated sub-tree has the same depth as the subtree cut.

The values of genetic operators of any evolutionary algorithm can affect the performance of the algorithm in a significant way [HME97]. As many researchers do, we choose values of parameters and of operators through experimentation. We try the values of GP operators suggested by [Koz92, LP02], and test them on the bargaining problem. Some researchers suggest other ways to find appropriate parameter values, surveyed in [HME97].

Under the GP setup in Table 3.3, it is typical that the fitness of two populations tends to be stabilized before 200 generations [7]. To ensure the stabilization and limited by computational resources, we terminate a run at the evolutionary time of 300^{th} generation.

According to the assumption **A-4**, the two players have the same level of learning ability. In terms of evolutionary algorithms, this means that the same genetic operators and the same values of parameters are employed for both populations.

3.5.3 Operators of Co-evolutionary System

In general, a co-evolutionary system is more sensitive to the values of genetic operators than an evolutionary system. In an evolutionary system, search undergoes against a static landscape. A high mutation rate or a low selective pressure, scatters the individuals in a

[7] Figure 4.6, 4.7, 4.8, and 4.9 illustrate the evolutionary stabilization process.

wider space and/or take longer time to converge.

By contrast, in the two-population co-evolutionary system, the search is conducted on two dynamic landscapes, both changing over generations. Each population is its co-evolving population's landscape. Any modification on one co-evolving population P_i updates its co-evolving population P_j's landscape, making improvement of P_j more difficult. In return such difficulty causes more uncertainty for P_i to cope with P_j that is P_i's landscape. Therefore, even a slight modification on one population magnifies its effects through dynamics and co-adaptation of the landscapes. So we should use smaller crossover, low selective pressure and mutation rates in a co-evolutionary system than those suitable for an evolutionary system.

This analysis is supported by our experimental results. A range of crossover rate, mutation rate and selection methods are tested to reduce their biases on experiential results. We find that high crossover rate and/or high mutation rate cause radical changes of genetic programs in a population. The fluctuations of landscapes are so massive that bargaining players receive heavily noisy responses from each other. As a result, it becomes difficult to co-adapt. High crossover and mutation rates are also the reason of of taking long time to stabilize. Low crossover and mutation rates are ideal for co-evolution, as long as they are large enough for players to do some learning. Selection methods with low selective pressures contribute to keeping landscape less dynamic. Our system performs relatively stable while the crossover rate is within 0 to 0.1, the mutation rate ranges from 0.01 to 0.3 and uses 3-member tournament selection method.

3.5.4 Relative and Absolute Fitness Functions for CRub82

One crucial part of successful applications of evolutionary algorithms is to design the fitness function properly. The features of the fitness function for this co-evolutionary system are emphasized.

First of all, the dynamic interactions between two players: co-evolutionary relation. We formally define the absolute and relative fitness function in co-evolutionary systems in Section 3.4. This section instantiates the relative and absolute fitness functions for the CRub82 bargaining problem. What roles they play in co-evolution is to be investigated. Experiments are designed to answer this question.

Another important feature is the constraint handling. The constraints in CRub82 bargaining problem need to be integrated into the fitness function. We design the Incentive method to adjust individuals' relative fitness, according to their satisfaction to constraints (see Section 5.4). We present the fitness function $F(g_i)$ shortly after presenting the constraint handling technique. Bear in mind that the co-evolutionary experiments in this and the next chapter are executed either using the $F(g_i)$ or using absolute fitness functions.

Relative Fitness Evaluation for CRub82

A player's goal is to maximize his utility. So the utility from bargaining agreements is an ideal measure of a player's ability to bargain and to obtain benefit from the mutual interest.

Game Fitness of a strategy $s(g_i)$ from population I, denoted by $GF(s(g_i))$ is defined as the average of $s(g_i)$'s utilities from agreements with individuals in the other population J. J is a set of m number of genetic programs which satisfy the constraint that $g_(i) \in (0,1]$, $j \in J$:

$$GF(s(g_i)) = \frac{\sum_{j \in J} u_{s(g_i) \to s(g_j)}}{m} \tag{3.6}$$

where $u_{s(g_i) \to s(g_j)}$ is the utility gained by $s(g_i)$ from an agreement with $s(g_j)$. $s(g_j)$ receives $u_{s(g_j) \to s(g_i)}$. In theory if players perpetually disagree, both players obtain utility 0. In experiments, if players do not agree after 10 time intervals, both get utility 0. [8]

$GF(s(g_i))$ returns the average utility that g_i receives from bargaining with either the co-evolving player or the monitor (the fixed representatives of absolute fitness functions). When J is a co-evolving population, $GF(s(g_i))$ calculates the relative fitness of g_i at the evolutionary time when J emerges. If J is static, $GF(s(g_i))$ returns g_i's absolute fitness.

On the basis of the relative fitness $GF(s(g_i))$, the fitness function $F(g_i)$ which drives the co-evolutionary system is established in Section 5.4. A constraint handling technique is imposed into the fitness function $F(g_i)$. This constraint handling technique controls one hard constraint: any share on the partition of cake should be a value within the size of cake: $x_i \in (0, 1]$ and two soft constraints: everything else being equal, the higher discount factor a player i has, the larger share x_i he obtains; everything else being equal, the higher discount factor the other player j has, the smaller share x_i player i gets. The constraint handling technique together with the fitness function $F(g_i)$ will be fully specified in Section 5.4.

A strategy's performance depends on the strategies with whom it is bargaining. Another choice of designing the fitness function is to use the absolute fitness function: a group of fixed representative strategies. This design has a risk that co-evolution may exploit the weaknesses of the pre-defined representatives, but may perform poorly against others. So the fitness of a strategy should be evaluated by its performance against the individuals in the co-evolving population. In other words, the relative fitness evaluation is an appropriate choice to address the fact that both players continually improve their co-adaptation.

[8] The reasons of limiting the bargaining time to 10 time intervals in experiments are explained in Section 3.5.5.

Absolute Fitness Evaluation for CRub82

Using the relative fitness function, the individuals in the populations at the end of evolutionary time are supposed to be co-adapted under a particular bargaining setting. Finding co-adaptive genetic programs are inadequate to answer questions that are important in observing the co-adapting process and in evaluating the co-adaptive strategies' performance. How does a population develop in order to find the best responses to its co-evolving population? Can co-adapted strategies out-perform the theoretical solution? During the evolutionary time, does co-evolutionary learning help players to adapt to more diverse environments, or only perform well to its co-adapted population J, a dynamic but known environment?

Taking into account the property of the absolute fitness and the knowledge of game-theoretic solutions, we design two absolute fitness functions to continually assess the co-adapting individuals, without replacing the relative fitness measure from the co-evolutionary system. At every generation, each co-evolving population is evaluated against the two absolute fitness functions which serve as external indicators: the set of SPE strategies and the static set of randomly generated strategies.

To differ notations used for the relative fitness, we use an alternative set of notations for absolute fitness evaluations, as shown in Table 3.4. First-move player 1's co-evolving population A_C starts from a randomly generated initial population A_R. A_P is his SPE strategy. Similarly, player 2' B_C, B_R, and B_P refer to player 2's co-evolving population, random population and SPE strategy respectively. Random genetic programs make offers and counter-offers uniformly distributed in the range of $[0, 1)$. The following sets of experiments are to be executed, in which the symbol "$\circ$" means "to bargain with":

- $A_C \circ B_R$ and $A_R \circ B_C$: random strategies bargain with co-evolving strategies;

Notations	Explanation
A	The first player 1's population
A_C	The first player 1's evolving population
A_P	The first player 1's population full of SPE strategies
A_R	The first player 1's population with randomly generated but static genetic programs
B	The second player 2's population
B_C	The second player 2's evolving population
B_P	The second player 2's population full of SPE strategies
B_R	The second player 2's population with randomly generated but static genetic programs
$\circ$	two populations evaluate each other through playing bargaining games

Table 3.4: Notations used for absolute fitness evaluations

- $A_C \circ B_P$ and $A_P \circ B_C$: co-evolving strategies bargain with SPE strategies.

3.5.5 Game Parameters

We select 25 game settings for testing, $\delta_1, \delta_2 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The combinations of selected δ_1 and δ_2 evenly distribute over the space $\delta_1 \times \delta_2$. 100 runs each with different random sequences are conducted for every pair of (δ_1, δ_2). 100-run is considered to be statistically sufficient to collect samples.

In experiments, a bargaining procedure lasts at most 10 time intervals. The reasons for limiting bargaining time to 10 intervals are:

- players pay the bargaining cost subject to time. When a bargaining process lasts 10 time intervals, player i pays $x_i(1-\delta_i)^{10}$ as the bargaining cost. For example, assume player 1 even wins the whole cake $x_1 = 1$ and he pays very low bargaining cost per time interval $\delta_1 = 0.95$. After 10 time intervals, he only gets about 60% of his original share $x_1 = 1$ or his utility $u_1 \approx 0.6$. Therefore, an agreement reached after 10 time intervals is very inefficient. Strategies that make agreements after long bargaining time receive relatively low utilities and thus low game fitness and fitness, so they have limited chance to survive in evolution.

- the game-theoretic solution SPE is achieved at time 0.
- we have limited computational resources, therefore we must focus on more efficient solutions.
- experimental results show that few best-of-generation genetic programs at the end of co-evolution make agreements at time $t > 4$ (Table 4.3). Details about bargaining time in experimental results are reported in the next chapter.

Limiting bargaining time to 10 time intervals is practical for solving bargaining problems having discount factors. Running games forever is not realistic in reality nor in computational resources. However, it might change the infinite bargaining game to an finite game. In theory, infinite games and finite games are very different [Gib92] [Mut99]. To make sure such a limitation on bargaining time has the minimal impacts on the experimental outcomes, we further ensure that: in experiments,

- players in the co-evolutionary system do not know that the maximal bargaining time is 10 time intervals;
- if players do not make an agreement after 10 time intervals, both gets 0 utility.

3.6 Summary

Bargaining problems are of high degree of computational complexity. Exhaustive search requires excessive computational resources. In addition, it is unclear on what extent the game-theoretic solutions apply to boundedly rational players.

We set up a set of assumptions on the bounded rationality of bargaining players' behaviors. Under such assumptions, we establish a theoretic framework and design a co-evolutionary system to simulate bargaining players' behaviors. In this co-evolutionary system, there are two populations, each for a player. These two populations are co-evolving, interacting and learning through bargaining experiences. Relative and absolute fitness of the co-evolutionary system are carefully designed. The coming chapter reports and analyzes experimental results of the CRub82 bargaining problem generated by the co-evolutionary system.

Chapter 4

Co-evolution to Tackle Alternating-Offers Bargaining Problem - Experimental Results and Observations

4.1 Introduction

In the previous chapter, we design the theoretic framework and a co-evolutionary system to deal with CRub82 bargaining problem. We assume that bargaining players are boundedly rational. System design and experimental set-up are described in great detail in Chapter 3.

This chapter reports and analyzes the experimental results. Experimental results are compared with game-theoretic Subgame Perfect Equilibrium. Additionally, the impact of discount factors on the divisions of the cake and on bargaining time are studied. Players' adaptive learning in a co-evolutionary system and computational resources are examined [1].

4.1.1 Statistic Measures

In this thesis, the major statistic measures to analyze the relationship between experimental results and game-theoretic solutions are t-test and the linear regression.

t-test

[1]Parts of this chapter have been published in [JT05b] and [JT05a].

We describe t-test here for clarifying the meanings of two terms: t-statistic value and t-critical value.

There are two samples. Each sample is a set of real numbers. One set is M, $m \in M$ and the mean of M is μ_M. Another set is L, $l \in L$ and the mean of L is μ_L. n is the size of two sets. The integer i, $1 \leq i \leq n$ is the index of the real numbers in both sets. The null hypothesis is that M is statistically same in pairs with L, in other words, $\mu_M - \mu_L = 0$.

Let the set N has n elements. Each element is $x_i = l_i - m_i$, $x_i \in N$. The mean of this set is μ, the variance is σ^2 and the standard deviation is σ.

$$\sigma^2 = \frac{\Sigma_1^n (x_i - \mu)^2}{n - 1} \tag{4.1}$$

The *t-statistic* value is defined as:

$$tsv = \frac{\mu \times \sqrt{n}}{\sigma} \tag{4.2}$$

t-critical value (two tail) under the 95% confidence level is notated as tc. If $|tsv| < |tc|$, the difference between two samples M and L is insignificant, so the null hypothesis can not be rejected.

Linear Regression

Linear regression examines the relationship between two variables X and Y. A linear regression creates the equation $Y = a + bX$. The slope of the line is b. The intercept is a which is the value of Y when $X = 0$. R-squared value R^2 is the square of the correlation coefficient. The correlation coefficient is a measure of the reliability of the linear relationship between the X and Y values. R^2 is a value between -1 and 1 indicating the strength of the association of the observed data for the two variables X and Y. A R^2 value close to 1 indicates an excellent linear reliability between X and Y.

We plan to test whether there is a relationship between the game-theoretic solutions x^* and experimental results $\bar{x}$. The linear regression helps us to investigate whether there is a significant association between x^* and $\bar{x}$.

4.2 Experimental Results and Observations

Generally speaking, the experimental results from the co-evolutionary system can be judged in two ways:

1. compare and contrast experimental results of the co-evolutionary system with game-theoretic properties and/or game-theoretic solutions;

2. compare and contrast experimental results of the co-evolutionary system with human behaviors observed in experimental economics. Findings and observations in literature of experimental economics vary greatly. These experiments were typically done on human subjects. Humans have different utility functions. Humans may change their minds so their utility functions are not stable. The differences in experimental subjects' age, educational level, culture and geographical location cause great discrepancies in findings. Therefore, there lacks commonly accepted conclusions, for example in [BSS89, GS93]. It deserves long-term studies.

In this thesis we mainly evaluate our experimental results by the first way and very briefly evaluate them by the second way. We leave the latter for future work.

This section analyzes experimental results. Bargaining theory emphasizes the properties of game-theoretic solutions at three aspects:

- *Partition* [2] of the cake: how the cake is split [Mut99].
- *Efficiency* of the agreement: whether the cake is solely split by two players so the agreement is the most efficient one. If players spend any portion of cake that he obtains for bargaining cost, the agreement is not efficient (also in Section 2.3.1 and [Mut99]).
- *Stationarity* of agreement: whether players intend to unilaterally withdraw from such an agreement (also in Section 2.3.1 and [Mut99]). Put it another way, if players want to make the same agreement next time, this agreement is stationary.

We regard these three aspects as the major points to investigate experimental data. Besides these three aspects, the adaptive learning in co-evolutionary process and the computational resources of using the co-evolutionary algorithm are also analyzed.

4.2.1 Partition of Cake in Agreement

In one run, x_1 is player 1's share from an observed agreement made by the pair of the best-of-generation (highest fitness) strategies, one from the population 1 and another from population 2, at the end of evolution (300^{th} generation in our experiments). $\bar{x_1}$ is the average of 100 x_1s from 100 runs for a given (δ_1, δ_2). The results of $\bar{x_2}$ is not reported for CRub82 problem because it is merely the complement of $\bar{x_1}$.

We execute 100 runs with different random seeds for each game setting (δ_1, δ_2). Totally 25 game settings are tested.

The SPEs of the first players (x_1^*s) are shown in Table 4.1. The differences between SPE x_1^* and $\bar{x_1}$ is: $(x_1^* - \bar{x_1})$ shown in Table 4.2. The t-test on the null hypothesis $(x_1^* - \bar{x_1} = 0)$ can not be rejected with 95% confidence where t-critical value two-tail = 2.0639 and t-statistic

[2] Some game theorists [Mut99] use the term "distribution" to refer partition.

δ_2 value	The SPE values				
	$\delta_1 = 0.1$	$\delta_1 = 0.3$	$\delta_1 = 0.5$	$\delta_1 = 0.7$	$\delta_1 = 0.9$
0.1	0.9091	0.9278	0.9474	0.9677	0.9890
0.3	0.7216	0.7692	0.8235	0.8861	0.9589
0.5	0.5263	0.5882	0.6667	0.7692	0.9091
0.7	0.3226	0.3797	0.4615	0.5882	0.8108
0.9	0.1099	0.1370	0.1818	0.2703	0.5263

Table 4.1: The SPE solutions: x_1^*s.

value $= -0.8513$. There is no statistical evidence to show that the experimental results are different from the game-theoretic solutions.

From the linear regression analysis on the results in Table 4.2, the correlation coefficient R^2 of the association between x_1^*s and $\bar{x_1}$s is 0.9928, which is very close to 1. The linear regression's coefficient variable is 0.9588 and its coefficient intercept is 0.0257. These two values tell on what extent SPEs are approximated by experimental results. In this case, the regression equation is $x_1^* = 0.9588 \times \bar{x_1} + 0.0257$. SPE is linearly approximated by the experimental results. The strength of the relationship between x_1^*s and $\bar{x_1}$s is illustrated in Figure 4.1. This scatter plot indicates an increasing trend. So the linear regression provides a useful model for this association [3].

We observe the distribution of x_1s. After 300 generations, the 100 x_1s cluster around SPE, having a minority of exceptions found. For most of the 25 tested game settings, the SPEs are within the distributions of experimental x_1s, for example, shown in Figure 4.4 and Figure 4.5. In Figure 4.4 and Figure 4.5 the x_1s of 100 runs are plotted according to their frequencies in the range of [0, 1]. Shown in Figure 4.4, 48% of the x_1s of 100 runs are in the range of [0.65, 0.70]; the 50% the x_1s of 100 runs are in the range of [0.70, 0.75]. The SPE $x_1^* = 0.6667$ is in the range of [0.65, 0.75]. Shown in Figure 4.5, 90% of the x_1s of 100

[3] If there appears no association between x_1^* and $\bar{x_1}$ variables (i.e., the scatter-plot does not indicate any increasing or decreasing trends), then fitting a linear regression model to the data probably will not provide a useful model.

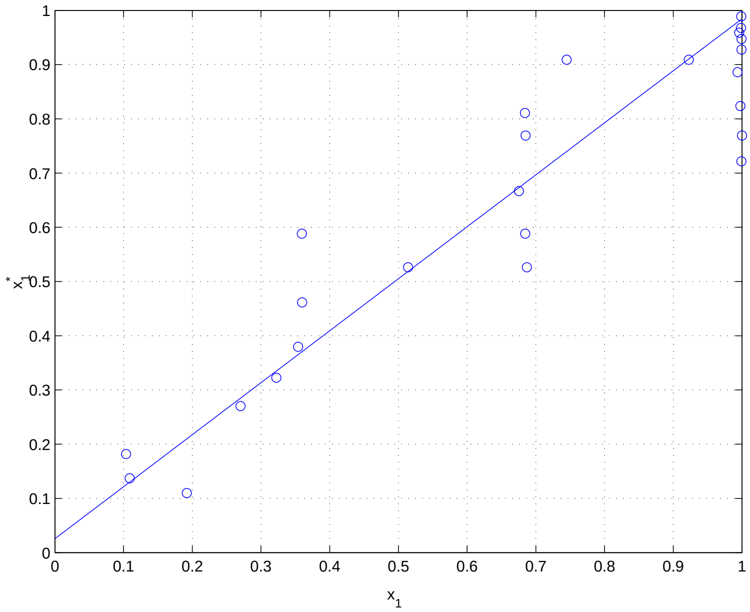


Figure 4.1: Linear regression of x_1^* and $\bar{x_1}$. The horizontal axis is $\bar{x_1}$ and the vertical axis is x_1^*.

runs are in the range of [0.98, 1.00]. The SPE $x_1^* = 0.9890$ is in the range of [0.98, 1.00]. Therefore it is very likely that the co-evolutionary system generates exactly same solutions as SPE, at a certain level of precision.

In Table 4.2, when $\delta_2 < 0.3 \cup (\delta_2 = 0.5 \cap \delta_1 \leq 0.5)$, the $(x_1^* - \bar{x_1})$s are negative, meaning that $\bar{x_1}$s are larger than x_1^*s. For the rest of game settings, $\bar{x_1}$s are smaller than x_1^*s. Figure 4.2 demonstrates how $(x_1^* - \bar{x_1})$ changes over the space $\delta_1 \times \delta_2$. For a given δ_1 the tendency is that while δ_2 increasing, $(x_1^* - \bar{x_1})$ starts with a negative value nearly 0. It deepens until it across a certain point around $\delta_2 = 0.5$. Then $(x_1^* - \bar{x_1})$ increases gradually and becomes a positive value. After some points around $\delta_2 > 0.7$, $(x_1^* - \bar{x_1})$ approaches to 0 again. Figure 4.3 displays $(x_1^* - \bar{x_1})$s over the space $\delta_2 \times \delta_1$. When $\delta_2 < 0.5$ $(x_1^* - \bar{x_1})$s are below 0; When $\delta_2 = 0.5$, $(x_1^* - \bar{x_1})$ increases from negative to positive values; when $\delta_2 > 0.5$, $(x_1^* - \bar{x_1}) > 0$. These patterns exhibit that two discount factors have different influences on $\bar{x_1}$s.

δ_2 value	$(x_1^* - \bar{x_1})$				
	$\delta_1 = 0.1$	$\delta_1 = 0.3$	$\delta_1 = 0.5$	$\delta_1 = 0.7$	$\delta_1 = 0.9$
0.1	-0.0135	-0.0715	-0.0520	-0.0308	-0.0098
0.3	-0.2775	-0.2308	-0.1741	-0.1075	-0.0371
0.5	-0.1606	-0.0963	-0.0087	0.0841	0.1643
0.7	0.0002	0.0258	0.1016	0.2287	0.1266
0.9	0.0821	0.0280	0.0780	0.0001	0.0122

Table 4.2: The differences between SPE x_1^*s and experimental $\bar{x}_1$s.

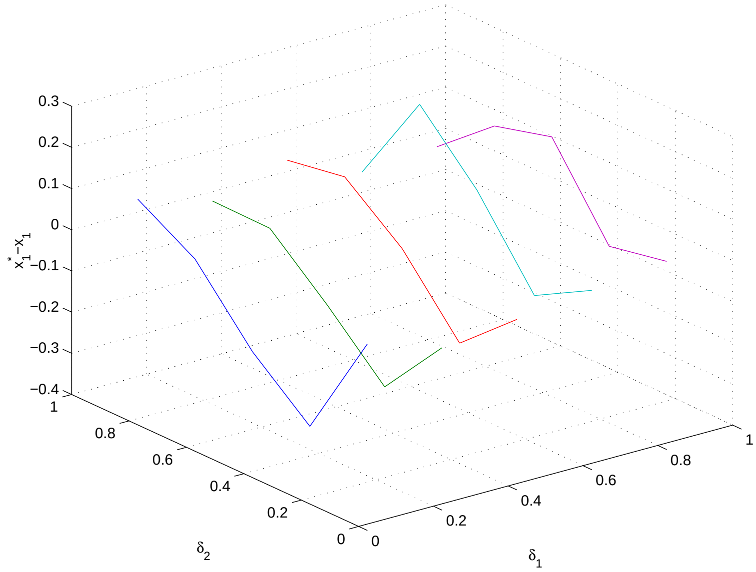


Figure 4.2: $(x_1^* - \bar{x}_1)$s over the space $\delta_1 \times \delta_2$

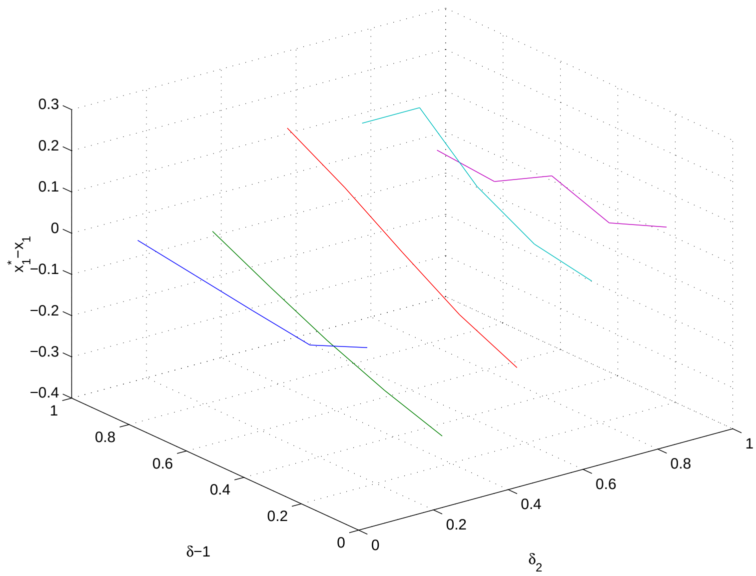


Figure 4.3: $(x_1^* - \bar{x}_1)$s over the space $\delta_2 \times \delta_1$

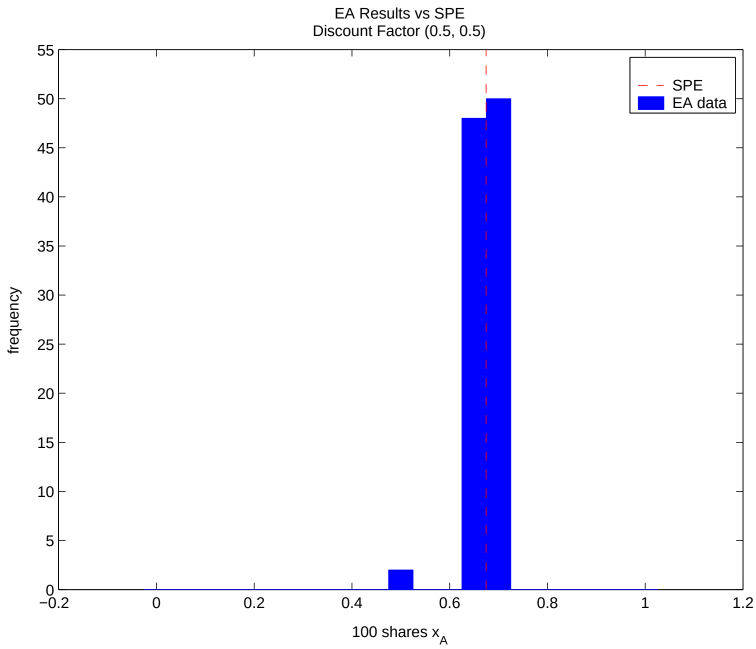


Figure 4.4: The frequency distribution of x_1s when $\delta_1 = 0.5$ and $\delta_2 = 0.5$. The vertical line is SPE $x_1^* = 0.6667$. The horizontal axis is 100 x_1s and the vertical axis is x_1's frequencies.

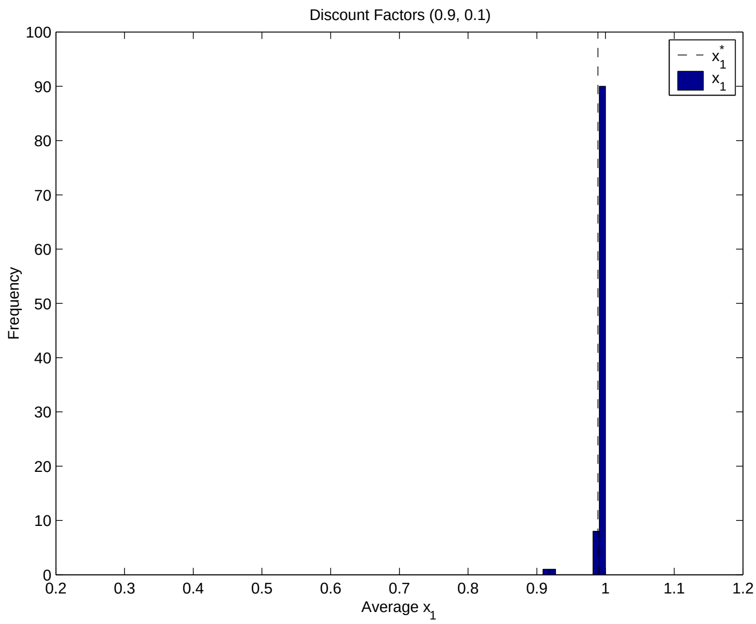


Figure 4.5: The frequency distribution of x_1s when $\delta_1 = 0.9$ and $\delta_2 = 0.1$. The vertical line is SPE $x_1^* = 0.9890$. The horizontal axis is x_1 and the vertical axis is x_1' frequency.

4.2.2 Bargaining Time and Efficiency of Agreement

In addition to the partition of cake, the time for reaching agreements is also important. A cake is divided most efficiently when the agreement is reached at $t = 0$. In such an agreement, the cake is solely shared by the two players and players spend nothing on the bargaining costs. The longer the bargaining time, the more bargaining costs are incurred and therefore the less efficiency of an agreement. t is an indicator of the efficiency of an agreement. Any $t > 0$ implies delay and inefficiency.

In SPE, all deals settle down at $t = 0$. In one run, t is the bargaining time for reaching an agreement by the best-of-generation individuals each from one of the two populations at the 300^{th} generation. $\bar{t}$ is the average of 100 ts of 100 runs for a given (δ_1, δ_2). $\bar{t}$s and their corresponding discount factors are reported in Table 4.3. In experiments, not every agreement is settled at the time $t = 0$. There is a tendency in $\bar{t}$s: when both δ_1 and δ_2 are small enough (≤ 0.5), delays never happen. Moreover δ_2 has more influence on the $\bar{t}$ than δ_1 in that $\delta_1 \geq 0.9$ does not necessarily lead to delays, but $\delta_2 \geq 0.9$ does certainly. Delays ($t > 0$) emerge as a consequence of players' preferences to higher utilities and of their expectations that higher utilities will obtain in future. This is especially true for more patient players who have large discount factors. Impatient players on the other hand, are eager to agree as soon as possible to avoid any delay otherwise they will afford relatively higher costs than that of patient players would.

4.2.3 Stationarity of Agreement

Game-theoretic equilibriums are stationary. In theory stationarity means that players have no intention to unilaterally withdraw from such an equilibrium [Mut99]. In game theory stationarity strictly constrains to the exactly game-theoretic solution(s).

	Experimental Bargaining Time $\bar{t}$				
δ_2 value	$\delta_1 = 0.1$	$\delta_1 = 0.3$	$\delta_1 = 0.5$	$\delta_1 = 0.7$	$\delta_1 = 0.9$
0.10	0.00	0.00	0.00	0.00	0.00
0.30	0.00	0.00	0.00	0.00	0.00
0.50	0.00	0.00	0.00	0.00	0.30
0.70	0.21	0.02	0.00	0.27	0.64
0.90	0.14	0.04	0.22	0.96	3.74

Table 4.3: The average of bargaining time $\bar{t}$ for a (δ_1, δ_2).

In experiments, due to the stochastic nature of evolutionary algorithms, we measure the stationarity by the deviation of the behaviors of genetic programs in a population at the last generation. If in a population, all genetic programs make the same or similar offers and counter-offers, this population and its player's behaviors are stationary. If both players' behaviors are stationary, the agreements that they achieve are certainly stationary as well. Moreover we assume that when the deviation is smaller than 0.05 the agreement made by the best-of-generation genetic programs is stationary.

There is a relevant concept "stability of evolutionary system" [LP02]. Strictly speaking when a population converges, individuals in the population have almost the same genotypes [Daw76]. This population is evolutionarily stable. *Genotype* in biology refers to the structure of DNA [4]. In genetic programming genotype refers to the structure of the syntax tree of a genetic program. *Phenotype* in biology is the characteristics and behaviors of an individual [5]. The phenotype of a genetic program in the co-evolutionary system is its player's behaviors in a bargaining game. Two individuals may have very similar genotypes, but they may have

[4] "The genotype is the specific genetic makeup (the specific genome) of an individual, in the form of DNA. Together with the environmental variation that influences the individual, it codes for the phenotype of that individual." [Wik06c]

[5] "The phenotype of an individual organism is either its total physical appearance and constitution or a specific manifestation of a trait, such as size, eye color, or behavior that varies between individuals. Phenotype is determined to some extent by genotype, or by the identity of the alleles that an individual carries at one or more positions on the chromosomes. Many phenotypes are determined by multiple genes and influenced by environmental factors. Thus, the identity of one or a few known alleles does not always enable prediction of the phenotype." [Wik06d]

very different phenotypes.

We consider for bargaining games, the convergence of genetic programs' phenotypes is more important. To use the convergence of individuals' phenotype implies that we do not focus on the individuals' genotypes. In the co-evolutionary system, we measure the stationarity of agreements by how different the phenotypes of genetic programs are in a population at the last generation. For evolutionary stability of the co-evolutionary system, as long as genetic programs make the same offers and counteroffers in the bargaining game, the co-evolving landscapes hardly change even if these genetic programs have very different genotypes.

In short for both the measure of evolutionary stability of the co-evolutionary system and the measure of stationarity of agreements, we adopt the measure on the phenotypes of genetic programs in populations at the end of evolution.

We use the deviation σ of x_1s as the statistic measure of the difference among phenotypes of genetic programs in a population. The value of σ also reflects the stability of the co-evolutionary population. σ is the indicator of both evolutionary stability and stationarity of agreements. We assume that when both populations have $\sigma \leq 0.05$, the co-evolutionary system is evolutionarily stable and agreements are stationary.

We report the σs for x_1s. The σs for x_2s are not reported because they are found to be very similar to σs of x_1s under the same (δ_1, δ_2). In the initial generations, the deviation σs can be as high as 10^3. As in Table 4.4, at the end of evolution, the deviation σs are relatively small: 19 out of 25 deviations are less than 0.05. 3 σs out of the rest are less than 0.09. Only 3 of 25 have deviations between 0.1 and 0.12. Such data indicate that the final populations remain evolutionarily stable even when the mutations apply on both populations and that observed agreements are of high stationarity.

δ_2 value	Deviation σ of x_1s				
	$\delta_1 = 0.1$	$\delta_1 = 0.3$	$\delta_1 = 0.5$	$\delta_1 = 0.7$	$\delta_1 = 0.9$
0.1	0.0308	0.0037	0.0030	0.0052	0.0030
0.3	0.0030	0.0000	0.0054	0.0079	0.0094
0.5	0.0128	0.0139	0.0271	0.0272	0.0847
0.7	0.1170	0.0397	0.0091	0.1148	0.0838
0.9	0.0288	0.0223	0.0432	0.0773	0.1194

Table 4.4: The values of deviations σs.

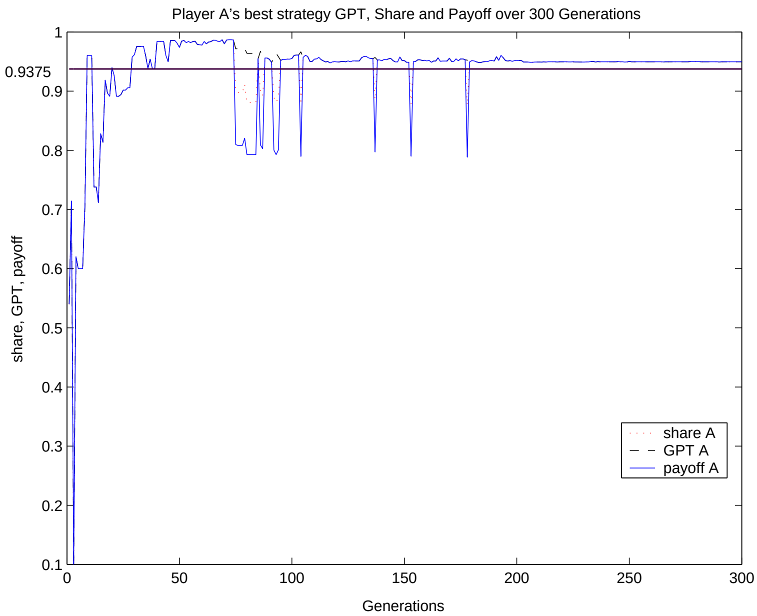


Figure 4.6: A typical run: the best-of-generation genetic programs for player 1, notated as A. The pair of discount factors is (0.9, 0.4). The line $y = 0.9375$ is the SPE of player 1. The overlaps of share x_1, p_1 (player i's utility), and g_1 imply that agreements are settled down at time $t = 0$. No bargaining cost incurs.

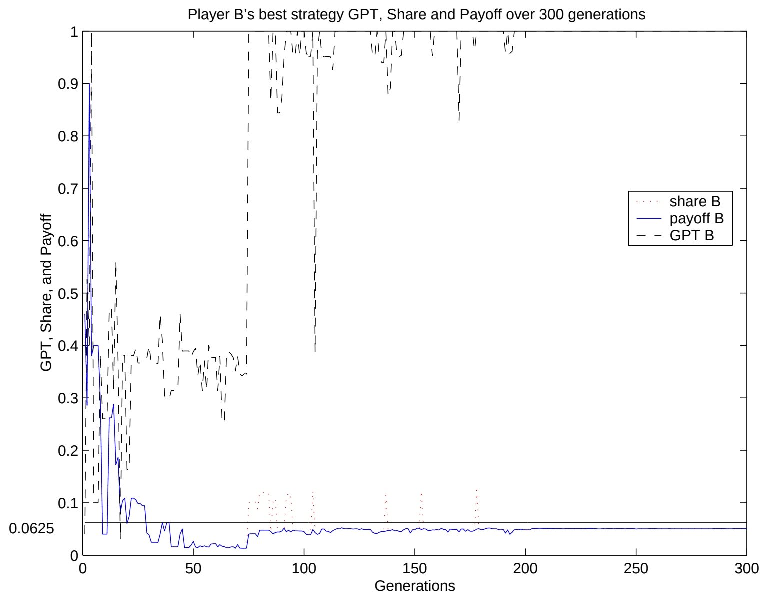


Figure 4.7: A typical run: the best-of-generation genetic programs for player 2, notated as B. The pair of discount factors is (0.9, 0.4). The line $y = 0.0625$ is the SPE of player 2.

4.2.4 Adaptive Learning Driven by Relative Fitness

Besides the partition of cake, the efficiency and stationarity of agreement, we are interested in how players learn to make efficient agreements from naive players at the beginning. The co-evolutionary system opens a window to observe the process of adaptive learning over evolutionary time. Neither can game-theoretic solution nor ESS provide such information. In Figure 4.8 and 4.9, two players' discount factors are $(\delta_1, \delta_2) = (0.5, 0.5)$. The horizontal line in Figure 4.8 is 0.6667 and the horizontal line in Figure 4.9 is 0.3333. They are SPEs for player 1 and 2 respectively. The other values displayed are the shares in agreements x_is and the utilities u_is. x_is and u_is are achieved by the best-of-generation genetic programs of every generation. Genetic programs continually update themselves over evolutionary time to co-adapt each other and come up with relatively stable x_is and u_is.

We comment now on modifications of phenotypes of genetic programs in this typical run in Figure 4.8 and 4.9. In the initial populations, genetic programs are generated randomly.

This means that a player obtains roughly 50% of the cake in average. Soon after player 1 learns that he can obtain more. He changes his first offer to as much high as he can in order to maximize his utility. In Figure 4.8, after 100 generations player 1's shares x_1s and utilities u_1s overlap SPE. He approaches the minimal value that player 2 accepts. It is very close to SPE. In the mean time, player 2 learns that she has to secure an agreement as soon as possible because it is not worthy for her to wait. So player 2 decreases her counter-offers and the acceptable shares. Thus players finally reach at the point where both of them are willing to agree at time 0 with no bargaining costs paid. No player wants to unilaterally withdraw from this agreement. This is verified by the fact that both utilities and shares in agreements stabilize and are very close to SPE horizontal lines. Players' behaviors after the evolutionary training fit nicely with the theoretical explanation.

Figure 4.6 and 4.7 illustrate another typical run with discount factors $(0.9, 0.4)$. It tells a similar story.

4.2.5 Learning Process Monitored by Absolute Fitness

The above findings from analyzing experimental results on partition of cake, efficiency of agreement and stationarity of agreements demonstrate that experimental results from the co-evolutionary system convincingly approximate the game-theoretic solution. These experimental results are generated by using the fitness function on the basis of relative fitness evaluation $GF(s(g_i))$. This indicates that the relative fitness successful guides the co-evolving players to perform nearly subgame perfect equilibrium strategies. It is also important to know what role the absolute fitness evaluation plays in this co-evolutionary system.

The experiments by using absolute fitness are designed in Subsection 3.5.4 with notations in Table 3.4. Two example runs are used to illustrate how absolute fitness functions work.

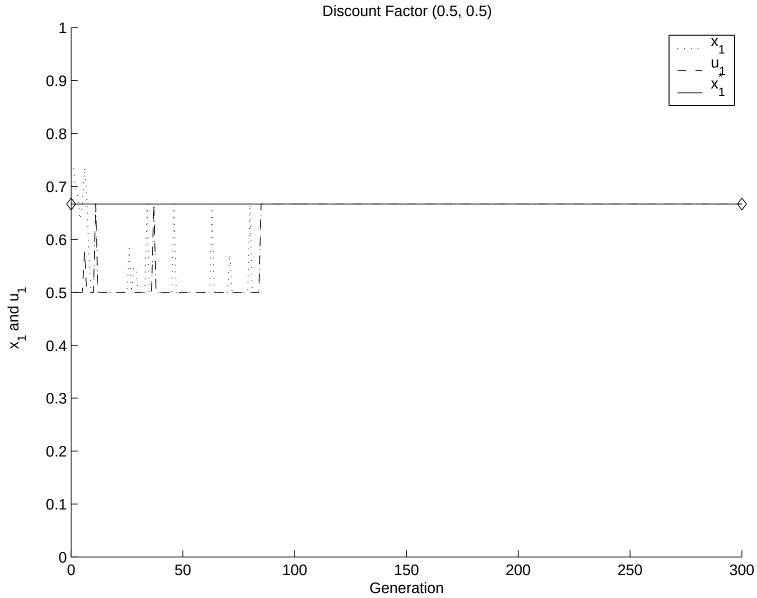


Figure 4.8: Player 1's shares and utilities by the best-of-generation genetic programs in Population 1 over 300 generations. (δ_1 =0.5 and δ_2 = 0.5). In the notation box, the first is x_1 which notates player 1' share from the agreement made by the best-of-generation individuals; the second is u_1 which is the utility corresponding to x_1; the third is x_1^* which is SPE solution.

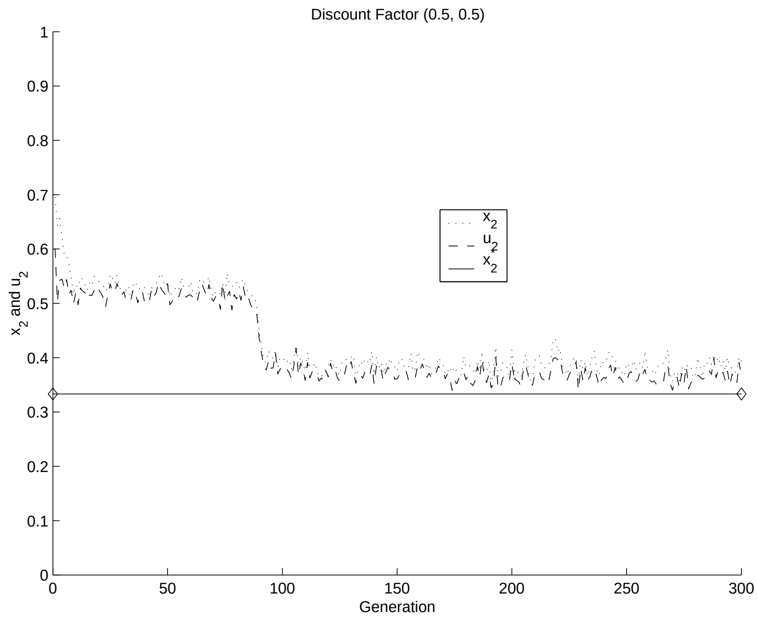


Figure 4.9: Player 2's shares and utilities by the best-of-generation genetic programs in Population 2 over 300 generations. (δ_1 =0.5 and δ_2 = 0.5). In the notation box, the first is x_2 which notates player 2' share from the agreement made by the best-of-generation individuals; the second is u_2 which is the utility corresponding to x_2; the third is x_2^* which is SPE solution.

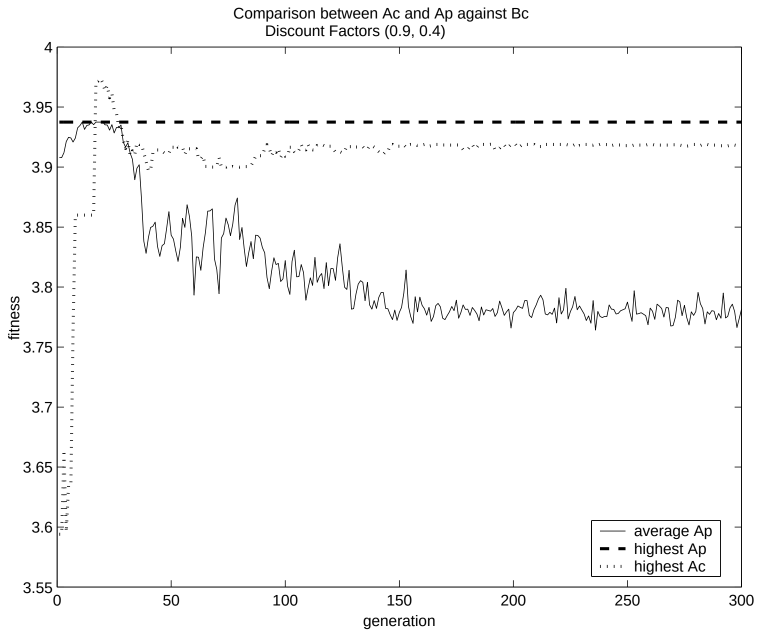


Figure 4.10: The highest and the average relative fitness of A_C, and the highest and the average absolute fitness of A_P, both against the co-evolving B_C. The average A_C does not display because it is much smaller than the other three values.

In a typical run shown in Figure 4.10, it is observed that in a very short period of time immediately after the beginning, the best-of-generation in the co-evolving population 1: A_C, performs better than A_P. It is because SPE is unable to exploit inexperienced strategies in B_C who offer or accept a partition of cake less than the x_2^* in SPE. But some strategies in A_C can take advantage of these weakly performed strategies in B_C by asking lager shares than SPE.

After first twenty of generations the average absolute fitness of A_P declines and stabilizes after around 150^{th} generation. This suggests that A_C's co-evolving B_C is learning to gain more utilities than B_C did at the beginning when B_C had no experience. Importantly B_C' such improvement does not reflect on his relative fitness against his co-evolving opponent A_C. Figure 4.10 tells that A_C's relative fitness remains stable after the 100^{th} generation. Therefore, it is the absolute fitness A_P or A_R that provides information concerning a co-

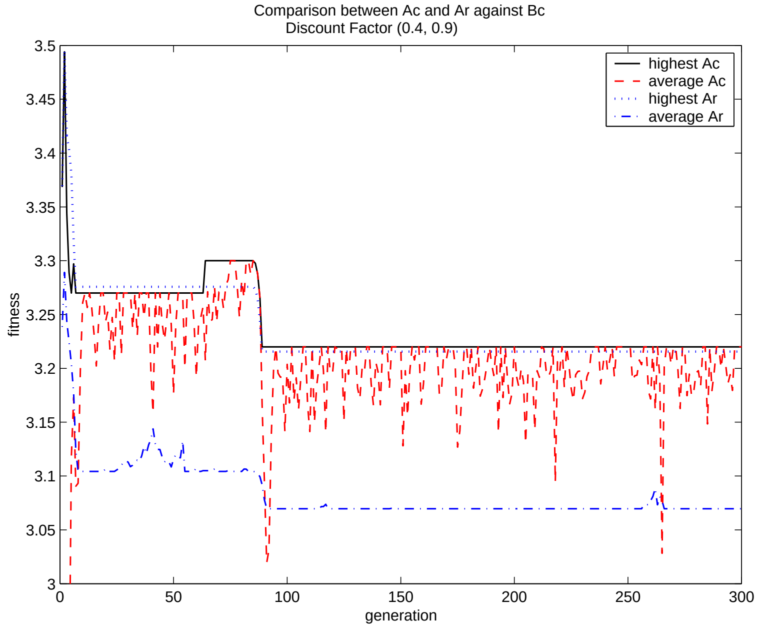


Figure 4.11: The highest and the average absolute fitness of A_R and the relative fitness of A_C against B_C.

evolving population's (B_C's) adaptation and improvement. Relative fitness of A_C tells little about how B_C improves.

Another absolute fitness function for studying the co-evolving population B_C is a static and randomly generated population A_R. In Figure 4.11, the highest absolute fitness of A_R almost overlaps the highest relative fitness of A_C. Both values quickly decline from 3.5 to 3.27. This implies that B_C quickly discovers the approximation to SPE and also improves its competitive strength against both A_C and A_R. Due to the diversity property of the random population A_R, the absolute fitness of A_R is less indicative on B_C's adaptation than the absolute fitness A_P is.

Let summarize the observations of relative and absolute fitness. In the application of the co-evolutionary algorithm to the bargaining problem CRub82, we analyze the co-adapting process of two populations through observing development of individuals' relative and absolute fitness. We gain insights into not only the empirical justification to the game-theoretic

SPE, but also the importance of adopting absolute evaluations to the co-evolutionary adaptive system. On the ground of experimental observations, the relative fitness continues pushing individuals to co-adapt. The absolute evaluation, on the other hand, provides information on the co-evolving process. Absolute fitness monitors the development of co-evolution. Having analyzed the learning behaviors of strategies based on their absolute fitness, we explain how co-evolving populations stabilize at the perfect equilibrium.

4.2.6 Computational Resources

We implement the co-evolutionary system by JAVA version "1.4.2" and use GNU Compiler for the Java Programming Language (GCJ). This Java compiler is developed by Sun. A Linux machine with an $athlon2400$ processor executes about 1 hour to test a game setting (δ_1, δ_2) with 100 runs. We only spent about a day to run experiments for 25 sets of game settings. Compared with either an exhaustive search or an ordinary human user of game theory, the artificial simulation, in particular evolutionary simulations, is dramatically more efficient for solving CRub82 problem.

4.2.7 Evolve Genetic Programs for All Game Settings

The design of experiments in Section 3.5 aims to evolve genetic programs for a specific game setting (δ_1, δ_2). We have tried to evolve genetic programs which adapt to all 25 game settings. It was hoped that the exact functions of game-theoretic solutions will be found.

A report about this experiments is detailed in Appendix H.

4.3 Concluding Summary

We treat the game-theoretic solutions as the benchmark, comparing them verse experimental results on the division of cake, on the efficiency of agreement and on stationarity of agreement. Comparative studies discover exciting findings as well as new understanding of co-evolution. Under the assumptions of players' bounded rationality, experimental results statistically approximate game-theoretic solutions. Experimental results also display relatively high efficiency and stationarity. It is confirmed that the relative fitness essentially drives co-evolution towards perfect equilibrium.

In addition, this study investigates the players' adaptive learning in the co-evolving process. The trial-and-error training experiences under the co-evolutionary framework guide both players to improve their adaptation to each other. Relative fitness is the driving force of co-evolution; absolute fitness plays the role of a monitor of adaptive learning. From experimental results, we understand whether and how the absolute fitness contributes to the development of co-adaptation.

The computational costs we have spent are rather low. It is much lower than the cost by using the game-theoretic method or an exhaustive search.

The next chapter concentrates on specifying the constraint satisfaction in the CRub82 problem. We invent a constraint handling technique, called the Incentive method, to manage the constraints. Then the relative fitness function is to be integrated by Incentive method.

Chapter 5

Constraint Driven Search - Incentive Method

5.1 Introduction

We establish a theoretic co-evolutionary framework and apply it to the CRub82 bargaining problem in the previous two chapters. The fitness function is so complicated that we need to elaborate one of its features in one chapter. Chapter 3 focuses on the co-evolutionary interactions and the relative fitness of individuals. This chapter aims to address another important concern of the fitness function on handling constraint satisfaction.

The CRub82 bargaining problem can be viewed as a multi-constraint optimization problem. It has both hard and soft constraints. Existing constraint handling techniques in evolutionary algorithms are to be surveyed in Section 5.1.1. Most of existing constraint handling techniques are not purposely designed for dealing with problems which have both hard and soft constraints at the same time. How to deal with such problems motivates us to design a constraint handling technique - Incentive method.

Section 5.2 formally defines the Incentive method. This method aims to handle both hard and soft constraints in an evolutionary algorithm.

Section 5.3 specifies both the hard and the soft constraints in CRub82 bargaining problem. Section 5.4 applies the Incentive method to the co-evolutionary system for CRub82 bargaining problem. Section 5.5 names the Constraint-based Co-evolutionary Genetic Programming System.

In Section 5.6, experimental results using Incentive method are compared with results using a penalty method and with results using a co-evolutionary algorithm without constraint handling technique. We end this chapter with the conclusion that on the ground of statistic analysis, the Incentive method is a more effective method than the other two techniques for the CRub82 problem [1].

5.1.1 Common Constraint Handling Techniques

Many optimization problems involve constraints. The well known constraints handling techniques used in evolutionary algorithms include: penalty methods; repair algorithms; multi-objective functions [Coe99, Mic95, MJ96] and co-evolutionary models [Coe02, CR03b]. We explain more on the penalty methods and repair methods. The penalty method [RPLH89, CEM01, CEvH03], penalizes infeasible or unfavorable individuals. In general, it transforms a constrained optimization problem "$max\ f(x)$ subject to $w(x) \leq C$" to an unconstrained problem "$max\ Y(x) = f(x) - Penalty(x)$" by defining the penalty function $Penalty(x)$. Given the same value of $f(x) - Penalty(x)$, we can not differ one $f(x_1) - Penalty(x_1)$ which has a high value of $f(x_1)$ and a high value of $Penalty(x_1)$ from another $f(x_2) - Penalty(x_2)$ which has a low value of $f(x_2)$ and a low value of $Penalty(x_2)$.

A death penalty method rejects any infeasible individual. Repair methods use domain specific operators to modify infeasible individuals to feasible ones. Repair methods have been used for solving many combinatorial optimization problems ([LV90, LP91, Mic95, MJ96]).

[1] Initial work of this chapter has been published in [TJ06].

Brown et al. [BLCF04] examine constraint satisfaction for multiple adversaries who have different objectives. Little [LFC03] shows that a multi-objective constraint optimization problem can be viewed as multiple-player game playing.

5.2 Formal Definition of the Incentive Method

There are two types of constraints in some optimization problems. *Hard constraints* describe feasibility of solutions. *Soft constraints* describe preferences, which often encode users' partial knowledge about good solutions. Candidate solutions can be classified into three qualitatively different sets: feasible, infeasible and preferred. The preferred set is not necessarily a subset of the feasible set. [2] No matter how many soft constraints a candidate solution satisfies, its satisfaction to hard constraints still has the first priority. Soft constraints may have different priorities among themselves. For such complicated problems, it is sometimes difficult to define penalties for penalty methods or to repair solutions to satisfy hard constraints while still taking soft constraints into consideration. It is also difficult to decide which of the many soft constraints can be sacrificed to ensure that the hard constraint(s) is (are) satisfied.

We present the Incentive method as a complementary method to penalty, repair methods and hybrid methods, to handle multiple constraints in evolutionary algorithms. By introducing the Incentive method, we attempt to deal with each type of constraints individually, through differentially rewarding individuals (candidate solutions) depending on the level of constraints they satisfy. Moreover, the Incentive method is designed to enable us to integrate

[2] There is an informal example to help understand the idea of hard and soft constraints. The optimization problem is to have a drink. "I am not allowed to drink in the seminar room by the department's regulation" is the hard constraint; "I prefer Coke to Pepsi" is a soft constraint. Coke is a preferred choice, no matter I am in the seminar room or not. In this simple case, the best choice is (a) "stay outside the seminar room and have a Coke". It satisfies both the hard and soft constraint. (b) "stay outside the seminar room and drink Pepsi" is the second order choice which satisfies the hard constraint but is not preferred; (c) "stay inside the seminar room" is the last choice as I can not achieve the goal of having a drink.

extra problem-specific knowledge into fitness functions. Constraints are used to guide the search [Tsa93], as opposed to being seen as obstacles to the search.

The Incentive method is especially suitable for problems whose solutions can be categorized into different groups by the nature of constraints. The main idea of the Incentive method is to define the relevance of each type of constraints to the quality of a solution. Thus all candidate solutions can be categorized into partially ordered sets. For example, the set of solutions violate hard constraints are definitely less favorable to the set of solutions that only violate some soft constraints, which are in turn less favorable to solutions that violate no constraints. Note that some soft constraints may not be strictly ordered. The partial order of the constraints is translated into the fitness function. Thus evolution rewards favorable candidate solutions according to this partial ordering mechanism. This helps to guide the evolutionary search to allocate more effort to the search areas that are more promising, without totally denying access to other areas.

The formal definition of Incentive method is below:

Definition 5.1: Incentive Method: An optimization problem has hard constraint(s) that define feasibility of solutions, and soft constraint(s) that define preference properties in solutions. The problem is to find $x \in S$ to optimize $f(x)$ where x can be a number, a vector or a computer program. Let S be the search space and $E \subseteq S$ be the set of feasible solutions. Further, let $Q \subseteq S$ be the set of solutions that meet soft constraints, i.e. the set has preferred individuals. Q may not be a subset of E.

$$R(x) = \begin{cases} f(x) + C & \text{if } x \in E \cap x \in Q \\ k(x) & \text{if } x \in E \cap x \notin Q \\ h(x) & \text{if } x \notin E \end{cases} \tag{5.1}$$

To instantiate $R(x)$, users must define the functions k and h. Functions k and h must satisfy the condition $h(x) < k(x) < f(x) + C$ where C is a constant. C is not strictly necessary; it is included so that k and h do not have to return negative values to meet the condition $h(x) < k(x) < f(x)+C$. Problem-dependent knowledge is required to define $k(x)$ and $h(x)$ in the Incentive method. Effective definitions of $k(x)$ and $h(x)$ help the search to allocate its effort more effectively. Therefore, the definition of $k(x)$ and $h(x)$ can be seen as a burden on users, but it can also be seen as an opportunity for channelling domain knowledge into the search method.

The conditions in Equation (5.1) ensure that solution sets are strictly ordered. Even if a feasible solution violates many soft constraints, it is still preferred than any infeasible one. A solution x that violates no constraints at all is preferred to a feasible solution x' that violates one or some soft constraints, whatever the values of $f(x)$ or $f(x')$ is. The Incentive method does not prevent a search from considering infeasible regions of the search space. This is because infeasible solutions may contain valuable genetic materials that are needed for finding global optimal solutions. However, the Incentive method discourages candidate solutions in infeasible regions to produce offspring.

5.3 Constraints in CRub82

The CRub82 bargaining problem implies a hard constraint: any offer or counter-offer of diving the cake must be within the size of cake. Any offer or counter-offer that does not obey this constraint is infeasible. Besides, the common sense tells that the player has a relatively higher discount factor (lower bargaining cost per time interval) is in a stronger position to bargain. This is because one time-interval delay costs him less than the same delay costs to the other player.

We list the hard constraint **C1** and two soft constraints **C2** and **C3**:

C 1. Any share on the partition of cake should be a value within the size of cake: $x_i \in (0, 1]$; This constraint must be satisfied.

C 2. Everything else being equal, the higher discount factor a player i has, the larger share x_i he obtains. This constraint is derived from problem-specific knowledge.

C 3. Everything else being equal, the higher discount factor the other player j has, the smaller share x_i player i gets. This constraint is derived from problem-specific knowledge.

5.4 Incentive Method in Fitness Function for CRub82

Success of applying evolutionary algorithms relies on appropriate fitness functions that evaluate performance of individuals. The relative fitness for CRub82 bargaining problem is defined in Section 3.5.4. How to integrate the constraints into the fitness function is the main purpose of this section. We apply Incentive method to fulfil the purpose.

The game fitness $GF(g_i)$ is designed for feasible individuals which satisfy **C1**. The features of soft constraints **C2** and **C3** help define $k(x)$ and $h(x)$ in Equation (5.1).

Sensibility Measure and Evaluation of Attribution

Obviously not all genetic programs meet the **C1**, **C2** and **C3** constraints, especially when the genetic programs are created randomly at the initial generation. *Sensibility Measure SM* is invented to measure whether a genetic program characterizes **C2** and/or **C3**. Let $g_i(p, q)$ be the instantiation of a genetic program g_i with δ_i being substituted by p and δ_j being substituted by q. An arbitrary real numbers $\alpha \in (0, 1)$.

Definition 5.2: Sensibility Measure of a genetic program g_i

$$SM_i(\delta_i, \delta_j, \alpha) = \begin{cases} -\frac{g_i(\delta_i,\delta_j)-g_i(\delta_i\times(1+\alpha),\delta_j)}{g_i(\delta_i,\delta_j)} \\ \quad \text{if } \delta_i \times (1+\alpha) < 1; \\ \frac{g_i(\delta_i,\delta_j)-g_i(\delta_i\times(1-\alpha),\delta_j)}{g_i(\delta_i,\delta_j)} \\ \quad \text{if } \delta_i \times (1+\alpha) \geq 1; \end{cases} \tag{5.2}$$

$$SM_j(\delta_i, \delta_j, \alpha) = \begin{cases} \frac{g_i(\delta_i,\delta_j)-g_i(\delta_i,\delta_j\times(1+\alpha))}{g_i(\delta_i,\delta_j)} \\ \quad \text{if } \delta_j \times (1+\alpha) < 1; \\ -\frac{g_i(\delta_i,\delta_j)-g_i(\delta_i,\delta_j\times(1-\alpha))}{g_i(\delta_i,\delta_j)} \\ \quad \text{if } \delta_j \times (1+\alpha) \geq 1; \end{cases} \tag{5.3}$$

SM describes **C2** and **C3** in a mathematical manner: when player i's discount factor increases from δ_i to $(\delta_i \times (1+\alpha)) \in (0,1)$, the genetic program g_i that positively correlates to δ_i should be rewarded. The amount of reward depends on the degree of the increment from $g_i(\delta_i, \delta_j)$ to $g_i(\delta_i \times (1+\alpha), \delta_j)$. In the case of α is too large to make $\delta_i \times (1+\alpha)$ satisfy the definition $\delta \in (0,1)$, we decrease the discount factor δ_i to $\delta_i \times (1-\alpha)$. The value of g_i should decrease accordingly. So $g_i(\delta_i, \delta_j)$ should be larger than $g_i(\delta_i \times (1-\alpha), \delta_j)$ under **C2**. When the other player j's discount factor is taken into consideration, the genetic program g_i that negatively correlates to δ_j should be rewarded. $SM_j(\delta_i, \delta_j, \alpha)$ returns such rewards. In short, positive values returned from $SM_i(\delta_i, \delta_j, \alpha)$ and $SM_j(\delta_i, \delta_j, \alpha)$ indicate that the genetic program g_i satisfies the constraints **C2** and **C3**, respectively.

Definition 5.3: Evaluation of Attribution (ATT) defines the incentive value to the genetic program g_i whose Sensibility Measures are SM_i and SM_j. The incentive is calculated by g_i's satisfaction to constraints.

$$ATT(i) = \begin{cases} 1 \\ \quad \text{if } SM_i(\delta_i, \delta_j, \alpha) > 0 \\ -e^{\frac{1}{SM_i(\delta_i,\delta_j,\alpha)}} \\ \quad \text{if } SM_i(\delta_i, \delta_j, \alpha) \leq 0 \end{cases} \tag{5.4}$$

$$ATT(j) = \begin{cases} 1 & \text{if } SM_j(\delta_i, \delta_j, \alpha) > 0 \\ -e^{\frac{1}{SM_j(\delta_i,\delta_j,\alpha)}} & \text{if } SM_j(\delta_i, \delta_j, \alpha) \leq 0 \end{cases} \tag{5.5}$$

When SM_i or SM_j returns a positive value, meaning that it satisfies the soft constraint, $ATT(i)$ or $ATT(j)$ gives the highest incentive value 1. When SM_i or SM_j returns a non-positive value, ATT gives an incentive less than 0. The exact incentive value depends on how close SM_i or SM_j is to 0. The closer SM_i or SM_j to 0, the higher incentive value is given by the $ATT(i)$ or $ATT(j)$. Here we adopt the function $-e^{\frac{1}{SM}}$ to control this incentive rewarding algorithm. For a non-positive value of SM, ATT is always negative in the range between $(-1, 0)$. For $SM \rightarrow 0^-$, ATT goes quickly to nearly 0. For $SM < -1$ and $SM \rightarrow -\infty$, ATT goes quickly to -1. The function $-e^{\frac{1}{SM}}$ is problem-dependent and is not the only way to implement the idea of incentive rewarding. It is chosen here for its simple structure.

Fitness Function

A genetic program g_i that satisfies the constraint **C1** is converted to a bargaining strategy $s(g_i)$. Strategies pair-wisely play the CRub82 bargaining game. An individual's game fitness depends on the composition of the other player's co-evolving population. Thus its game fitness against a co-evolving population is its relative fitness, changing over evolutionary time. Just recap that *Game Fitness* $GF(s(g_i))$ to the co-evolving population is defined as the average utility of the strategy $s(g_i)$ gained from bargaining against every $s(g_j)$ in the co-evolving j's population [3].

Definition 5.4: Fitness Function $F(g_i)$ *incorporated with the Incentive Method* determines the overall fitness of g_i whose Sensibility Measures are SM_i and SM_j and whose Evaluation

[3] Definition of $s(g_i)$ is in Section 3.5.4.

of Attribution are $ATT(i)$ and $ATT(j)$.

$$F(g_i) = \begin{cases} GF(s(g_i)) + 3 \\ \quad \text{if } g_i \in (0,1] \cap SM_i > 0 \cap SM_j > 0 \\ GF(s(g_i)) + ATT(i) + ATT(j) \\ \quad \text{if } g_i \in (0,1] \cap (SM_i \leq 0 \cup SM_j \leq 0) \\ ATT(i) + ATT(j) - e^{\frac{-1}{|g_i|}} \\ \quad \text{if } g_i \notin (0,1] \end{cases} \tag{5.6}$$

This is the top-level fitness function which integrates the Incentive method. It is applied to all genetic programs in both populations. It instantiates the formal definition of the Incentive method $R(x)$ in Equation (5.1): $f(x) + C = GF(s(g_i)) + 3$ returns the fitness value of any genetic program which meets the hard constraint $g_i \in (0, 1]$ and the two soft constraints $SM_i > 0 \cap SM_j > 0$. Genetic programs that satisfy all three constraints are rewarded a bonus of 3 plus their game fitness $GF(s(g_i))$. This ensures that they dominate the rest of genetic programs that fail to meet all three constraints and thus encourages desired genetic programs to propagate. Integer 3 is an experimental value. Other integers might be possible. If this value is too small, it may be insufficient to differentiate the genetic programs with the feasible and preferred features from the rest of them. From experimental results, the integer 3 is an appropriate value.

In Equation(5.6), $k(x) = GF(s(g_i))+ATT(i)+ATT(j)$: genetic programs that satisfy the constraint **C1**, but do not meet **C2** and/or **C3**, are still eligible for playing bargaining games. Their fitness are their game fitness adding a value $(ATT(i) + ATT(j))$. $(ATT(i) + ATT(j))$ reflects how close genetic programs satisfy these two soft constraints. $h(x) = ATT(i) + ATT(j) - e^{\frac{-1}{|g_i|}}$: such genetic programs that violate the constraint **C1** are not eligible for playing bargaining games. Instead they are allocated a fitness solely based on the structures of their genetic programs assessed by SM and ATT. Their fitness is definitely lower than any genetic program that satisfies at least **C1**. In $F(g_i)$, $GF(s(g_i)) + 3 > GF(s(g_i)) +$

$ATT(i) + ATT(j) > ATT(i) + ATT(j) - e^{\frac{-1}{|g_i|}}$.

5.5 Constraint-based Co-evolutionary Genetic Programming - CCGP

So far we have elaborated the three major features of the co-evolutionary system for the CRub82 problem: Constraint-based, co-evolutionary interactions and using genetic programming. This system therefore is named as *Constraint-based Co-evolutionary Genetic Programming*, or *CCGP*. The CCGP is designed to be extensible and modifiable for various two-player bargaining problems. The following three chapters witness adaptation of CCGP to six other bargaining problems.

5.6 Experimental Results and Observations

This section reports and compares the experimental results generated by the co-evolutionary system under three constraint handling techniques: Incentive method, Penalty method and imposing no constraints.

5.6.1 Results from using Incentive Method

Two statistic methods measure the experimental results using Incentive method:

(1) Reported in Section 4.2 for experimental results of the 25 tested game settings, a t-test over the hypothesis ($x_1^* - \bar{x}_1 = 0$) is done. The result of the t-test shows that there is no statistically significant difference between the experimentally observed $\bar{x}_1$s and game-theoretic x_1^*s, with 95% level of confidence. A linear regression test shows that there is a strong linear correlation between x_1^*s and $\bar{x}_1$s.

(2) Two types of variations are defined below to measure the difference between x_1^* and

experimentally observed $\bar{x_1}$ of a given game setting [4].

Definition 5.5: Absolute Variation (av) is an unsigned difference between the SPE x_1^* and experimentally observed $\bar{x_1}$ for a given (δ_1, δ_2).

$$av = |x_1^* - \bar{x_1}| \tag{5.7}$$

Definition 5.6: Relative Variation (rv) is an unsigned relative increment of $\bar{x_1}$ over x_1^*.

$$rv = \frac{|x_1^* - \bar{x_1}|}{x_1^*} = \frac{av}{x_1^*} \tag{5.8}$$

The reasons that we adopt two variation measures are that (i) for absolute variation, $|x_1^* - \bar{x_1}| = |x_2^* - \bar{x_2}|$ is true. But for relative variation, $\frac{|x_1^*-\bar{x_1}|}{x_1^*} = \frac{|x_2^*-\bar{x_2}|}{x_2^*}$ is always not true. Relative variation may produce two different results for analyzing one experimental run; (ii) absolute variation alone is insufficient to express the variation on the basis of the target point: SPE. For example, two sets ($x_1^* = 0.05, \bar{x_1} = 0.06$) and ($x_1^* = 0.95, \bar{x_1} = 0.96$), both have the same absolute variation $av = 0.01$, but the former set has a $rv = 0.17$ increment based on x_1^* and the latter set has only a $rv = 0.01$ increment based on x_1^*. In this sense, the relative variation is more informative and indicative. For these two reasons we use both absolute and relative variations.

The absolute variations of experimental results by the Incentive method are given in Table 5.1. Most majority of the absolute variations avs are very small, especially for game settings reported in Table 5.5. The relative variations by the Incentive method are in Table 5.2.

Having obtained a general view on the results in Chapter 4, we further look into the effectiveness of the Incentive method. The questions to be answered are: whether does

[4]The term "difference" is used for $(x_1^* - \bar{x_1})$, see Section 4.2. Here we use "variation" to notate $|x_1^* - \bar{x_1}|$ or $\frac{|x_1^*-\bar{x_1}|}{x_1^*}$.

δ_2 value	Incentive Method ($\lvert x_1^* - \bar{x_1}\rvert$)				
	$\delta_1 = 0.1$	$\delta_1 = 0.3$	$\delta_1 = 0.5$	$\delta_1 = 0.7$	$\delta_1 = 0.9$
0.1	0.0135	0.0715	0.0520	0.0308	0.0098
0.3	0.2775	0.2308	0.1741	0.1075	0.0371
0.5	0.1606	0.0963	0.0087	0.0841	0.1643
0.7	0.0002	0.0258	0.1016	0.2287	0.1266
0.9	0.0821	0.0280	0.0780	0.0001	0.0122

Table 5.1: Absolute Variation by the Incentive Method (x_1^* is the SPE. $\bar{x_1}$ is the average of x_1s).

δ_2 value	Incentive Method ($\lvert x_1^* - \bar{x_1}\rvert / x_1^*$)				
	$\delta_1 = 0.1$	$\delta_1 = 0.3$	$\delta_1 = 0.5$	$\delta_1 = 0.7$	$\delta_1 = 0.9$
0.1	0.0149	0.0771	0.0549	0.0318	0.0100
0.3	0.3845	0.3000	0.2114	0.1213	0.0387
0.5	0.3051	0.1638	0.0131	0.1094	0.1807
0.7	0.0007	0.0679	0.2201	0.3889	0.1562
0.9	0.7472	0.2045	0.4292	0.0004	0.0232

Table 5.2: Relative Variation by the Incentive Method (x_1^* is the SPE. $\bar{x_1}$ is the average of x_1s).

the Incentive method outperform some other constraint-handling techniques, for instance, widely applied Penalty method? Does the Incentive method produce better results than an evolutionary algorithm having no constraint handling technique?

5.6.2 Results from using a Penalty Method

To evaluate the performance of the Incentive method, a control experiment using a penalty method is conducted. The penalty method integrates to the fitness function $F(g_i)'$. It penalties infeasible genetic programs. To be fairly comparable to the fitness function $F(g_i)$ in which the Incentive method is used, $F(g_i)'$ implements the penalty function for infeasible genetic programs as $ATT(i) + ATT(j) - e^{\frac{-1}{|g_i|}}$.

Definition 5.7: Fitness Function $F(g_i)'$ incorporated with Penalty Method

$$F(g_i)' = \begin{cases} GF(s(g_i)) + ATT(i) + ATT(j) \\ \quad \text{if } g_i \in (0,1] \\ ATT(i) + ATT(j) - e^{\frac{-1}{|g_i|}} \\ \quad \text{if } g_i \notin (0,1] \end{cases} \tag{5.9}$$

δ_2 value	Penalty Method ($\lvert x_1^* - \bar{x_1}\rvert$)				
	$\delta_1 = 0.1$	$\delta_1 = 0.3$	$\delta_1 = 0.5$	$\delta_1 = 0.7$	$\delta_1 = 0.9$
0.1	0.0909	0.0722	0.0526	0.0323	0.0110
0.3	0.2784	0.2308	0.1765	0.1139	0.0411
0.5	0.1625	0.0986	0.0058	0.0813	0.1555
0.7	0.0066	0.0238	0.1025	0.2356	0.0855
0.9	0.0765	0.0184	0.0458	0.0021	0.0252

Table 5.3: Absolute Variation by Penalty Method (x_1^* is the SPE. $\bar{x_1}$ is the average of x_1s).

δ_2 value	Penalty Method ($\lvert x_1^* - \bar{x_1}\rvert / x_1^*$)				
	$\delta_1 = 0.1$	$\delta_1 = 0.3$	$\delta_1 = 0.5$	$\delta_1 = 0.7$	$\delta_1 = 0.9$
0.1	0.1000	0.0778	0.0556	0.0333	0.0111
0.3	0.3857	0.3000	0.2143	0.1286	0.0429
0.5	0.3088	0.1676	0.0087	0.1057	0.1710
0.7	0.0203	0.0627	0.2221	0.4005	0.1055
0.9	0.6965	0.1340	0.2516	0.0079	0.0479

Table 5.4: Relative Variation by Penalty Method (x_1^* is the SPE. $\bar{x_1}$ is the average of x_1s).

The same sequence of random seeds, the same genetic operators and the same sets of discount factor pairs as those for experiments using the Incentive method are used in this control experiment. The experimental results of $F(g_i)'$ on absolute variation and relative variation are reported in Table 5.3 and Table 5.4.

Table 5.5 lists the average absolute variations of the Incentive method and those of the penalty method, grouped by three ranges, namely $av < 0.15$, $av < 0.10$ and $av < 0.05$. Table 5.6 lists the average relative variations of the Incentive method and those of the penalty method, grouped by three ranges, namely $rv < 0.15$, $rv < 0.10$ and $rv < 0.05$. A clear pattern displays: for all these three groups, both on absolute variation and on relative variation, the Incentive method yields smaller variations than the penalty method does. $\bar{x_1}$s found by Incentive method approximate better to SPE than those found by the penalty method. Therefore, the Incentive method is more effective than the Penalty method in this case. Moreover, counting the number of game settings, we find that for much more game

Method	Average of Absolute variations $\frac{\sum av}{m}$ when $0 \leq av < 0.15$	Average of Absolute variations $\frac{\sum av}{m}$ when $0 \leq av < 0.10$	Average of Absolute variations $\frac{\sum av}{m}$ when $0 \leq av < 0.05$
Incentive Method	0.0509	0.0394	0.0166
m	19	16	10
Penalty Method	0.0519	0.0453	0.0212
m	19	16	10

Table 5.5: Average of absolute variations (av in Equation(5.7)) under the specified ranges. “m” is the number of avs in the specified range, amongst the 25 game settings.

Method	Average of relative variations $\frac{\sum rv}{n}$ when $0 \leq rv < 0.15$	Average of relative variations $\frac{\sum rv}{n}$ when $0 \leq rv < 0.10$	Average of relative variations $\frac{\sum rv}{n}$ when $0 \leq rv < 0.05$
Incentive Method	0.0433	0.0302	0.0166
n	13	11	8
Penalty Method	0.0628	0.0426	0.0246
n	15	11	7

Table 5.6: Average of relative variations (rv in Equation(5.8)) under the specified ranges. “n” is the number of rvs in the specified range, amongst the 25 game settings.

Absolute Variations	Incentive Method better	Tie	Penalty Method better
Number of Game Settings	16	1	8
Relative Variations	**Incentive Method better**	**Tie**	**Penalty Method better**
Number of Game Settings	16	1	8

Table 5.7: The number of game settings which Incentive method performs better than the penalty method, amongst the 25 game settings.

settings, Incentive method perform better than the penalty method, see Table 5.7.

5.6.3 Results from imposing No Constraint

To examine the effectiveness of the Incentive method, we further do experiments whose fitness functions control none of the three mentioned constraints **C1**, **C2** and **C3**. A genetic program's fitness function is its game fitness $GF(g_i)$. All genetic programs in populations play the bargaining game. Under this non-constrained fitness function $GF(g_i)$, experimental results show that for a given game setting (δ_1, δ_2), the majority of the 100 runs end with genetic programs that offer x_1s which are within the cake size, i.e. satisfying the hard constraint. However, their average value $\bar{x_1}$ may have very large absolute and/or relative variations. $\bar{x_1}$ deviates far away from the SPE, because a few runs might end up with some x_1s being exceptionally large or exceptionally small (including negative) values. For example, in 100 runs of experiments on game setting $(0.9, 0.1)$, 17 x_1s out of 100 x_1s end with asking for $x_1 > 1$. The highest $x_1 = 8 \times 10^{14}$. The rest of 83 runs end with $0 < x_1 < 1$. The reason is that the search probably had no chance to enter the area of (0, 1]. The average value of 100 runs, $\bar{x_1}$ thus does not meet the hard constraint $x_i \in (0, 1]$. Therefore, imposing no constraint into the fitness function is impractical in this case.

The co-evolutionary algorithm is one of constraint handling techniques [Coe02, CR03b]. In this study of two-player bargaining problems, the purpose of using co-evolution is not for constraint handling but for simulating the two-player interactive behaviors.

5.7 Concluding Summary

This chapter introduces the Incentive method, which is a novel constraint handling technique in evolutionary algorithms.

The main idea of the Incentive method is to modify the fitness function by awarding differential incentives according to the defined qualitative preferences. Solution sets are divided by their satisfaction to constraints. The Incentive method uses constraints to help allocating heuristic search effort more effectively. The Incentive method does not exclude individuals in evolutionary algorithms the right to access such search spaces that violate some or even all constraints.

We integrate the Incentive method into the game fitness of the co-evolutionary system. We test this integrated fitness function to CRub82 bargaining problem. We compare the experimental results from using incentive-based fitness function with those from using a penalty-based fitness function and with those from a fitness function without any constraint handling. We measure experimental results from different constraint handling techniques on how close their solutions approximate SPE. Analytic results suggest that the fitness function integrated with the Incentive method finds the best approximation to SPE than the other two fitness functions, one with penalty method and another with no constraint handling. This finding encourages us to apply the Incentive method to other bargaining problems and some optimization problems with similar constraints features.

This chapter names the co-evolutionary system incorporated with the Incentive method and implemented by genetic programming as *Constraint-based Co-evolutionary Genetic Programming*, or *CCGP*. CCGP system is reused in later chapters as the general system for tackling two-player bargaining problems.

Chapter 6

CCGP for Bargaining Problems with Incomplete Information

6.1 Introduction

It is well recognized that incomplete information situations are common occurrences in real world bargaining scenarios. The existence of incomplete information captures an essential aspect of realistic bargaining. In few circumstances players are fully informed about the other players' game relevant information.

The two-player bargaining problems that we attempt to study in this chapter are dynamic games in the presence of incomplete information and perfect information. The main purpose of this chapter is to empirically study the impact of players' information completeness on the outcomes in bargaining games. We aim to examine how players' information affects bargaining outcomes and whether the Constraint-based Co-evolutionary Genetic Programming generates inefficient or disadvantageous solutions for a player who is not fully informed.

To address these issues, we introduce four bargaining problems imposed by different inputs on the players' information. We restrict our attention to the problems having minor modifications on the basis of the complete information bargaining problem CRub82 in order to compare experimental results of incomplete information problems with those of the complete information problem. These four bargaining problems differ from each other on players' information. The bargaining scenario and utility functions remain unchanged.

We apply CCGP to solve four incomplete information bargaining problems and investigate the experimental results on partition of cake in agreements, the efficiency of agreements and the stationarity of agreements.

We modify the CCGP system to adapt these bargaining problems. The remainder of this chapter firstly recaps CRub82 and ICRub85 bargaining problems and then introduces three incomplete information bargaining problems in Section 6.2. The bounded rationality assumptions on players' behaviors are discussed in Section 6.3. Constraints are specified in Section 6.4. The adaption of CCGP to fit the assumptions and constraints for incomplete information bargaining problems is described in Section 6.5. Experimental results [1] are analyzed in Section 6.6. Findings are discussed in Section 6.7 and it concludes in Section 6.8 [2].

6.2 Incomplete Information Bargaining Problems

These incomplete information bargaining problems of interest together with CRub82 are outlined in Table 6.1. The incomplete information studied in this chapter is static. For example, in the Bilateral Ignorance Information bargaining problem, abbreviated as BGI, a player j does not know the other player i's time preference which is captured by the discount factor δ_i. δ_i exists as a constant.

The computational complexity of any one of these four incomplete information bargaining problem is higher than that of CRub82 bargaining problem (Section 3.2). The existence of

[1] The raw data of experimental results are available in Appendix E.
[2] Initial work of this chapter has been published in [Jin05].

Type	Full names of bargaining problems	Abbreviation	Descriptions in Chapter
Unilateral Incomplete Information	Rubinstein Bargaining 1985 [Rub85]	ICRub85	2
	Unilateral Imprecise Information	UII	6
	Unilateral Ignorance Information	UGI	6
Bilateral Incomplete Information	Bilateral Ignorance Information	BGI	6
Complete Information	Rubinstein Bargaining 1982 [Rub82]	CRub82	2

Table 6.1: Four incomplete information bargaining problems and their comparable complete information bargaining problem.

incomplete information increases the number of possible offers and counter-offers during a bargaining encounter.

6.2.1 Recapitulation of CRub82 and ICRub85

For the purpose of measuring the impact of incomplete information on bargaining outcomes, we use the experimental results in Chapter 4 of the comparable complete information bargaining problem CRub82. It is solved by CCGP. In CRub82 problem, both players' discount factors are publicly known. Its game-theoretic solution, the *Subgame Perfect Equilibrium* is the unique point where "no delay" and "stationarity" properties hold. In SPE player 1 offers

$$x_1^* = \frac{1-\delta_2}{1-\delta_1\delta_2} \tag{6.1}$$

Player 2 certainly accepts this offer x_1^* and receives $(1-x_1^*)$ of the cake at $t=0$. Therefore, two players' utilities are $u_1 = x_1^*$ and $u_2 = 1-x_1^*$. There is no bargaining cost spent.

ICRub85 bargaining problem and its game-theoretic solutions are introduced in great detail in Section 2.3.2. Table 6.2 lists the game variables and their properties. Player 1 only knows that the value of player 2's discount factor δ_2 is either δ_w or δ_s. δ_s is strictly larger than δ_w. Player 2 knows player 1's discount factor is δ_1. The possibility of player 1's initial

Player	*Variable*	*Explanation*	*Privacy*
1	δ_1	Play 1's discount factor	Public
2	δ_2	Player 2's real discount factor	Private
$2'$	δ_2'	Incorrect player 2's discount factor	Private
2_s	δ_s	$= MAX\{\delta_2, \delta_2'\}$ Strong type	Public
2_w	δ_w	$= MIN\{\delta_2, \delta_2'\}$ Weak type	Public
	ω_0	Player 1's initial belief of the possibility of $\delta_2 = \delta_w$	Public

Table 6.2: Notations of Variables in ICRub85.

	$\delta_2 = \delta_w$		$\delta_2 = \delta_s$	
	x_1^*	t^*	x_1^*	t^*
$\omega_0 < \omega*$	V_s	0	V_s	0
$\omega_0 > \omega*$	x^{ω_0}	0	y^{ω_0}	1

Table 6.3: Player 1' share x_1^* and bargaining time t^* for reaching the PBE agreement. Player 2's share in the PBE agreement is $x_2^* = 1 - x_1^*$. Bargaining starts at time 0.

belief on player 2's discount factor being δ_w is ω_0. Table 6.3 summarizes the game-theoretic solutions Perfect Bayesian Equilibrium (PBE), where

$$V_s = \frac{1 - \delta_s}{1 - \delta_1 \delta_s} \tag{6.2}$$

$$\omega^* = \frac{V_s - \delta_1^2 V_s}{1 - \delta_w + \delta_1 V_s(\delta_w - \delta_1)} \tag{6.3}$$

$$x^{\omega_0} = \frac{(1 - \delta_w)(1 - \delta_1^2(1 - \omega_0))}{1 - \delta_1^2(1 - \omega_0) - \delta_1 \delta_w \omega_0} \tag{6.4}$$

Player 1 correctly guesses the value of δ_2 when $\omega_0 = 1$ and $\delta_2 = \delta_w$ or when $\omega_0 = 0$ and $\delta_2 = \delta_s$.

The other three incomplete information bargaining problems are presented in the coming subsections. They are in the scope of bargaining over a divisible interest, infinite-horizon and incomplete information about time preferences (discount factor). They do not have known game-theoretic solutions yet. Therefore we exclude them from Chapter 2 Literature survey.

6.2.2 Unilateral Imprecise Information - UII

The unilateral imprecise information bargaining problem, abbreviated as UII interests us in that player 1 has imprecise information about δ_2, knowing that δ_2 is drawn from a Gaussian distribution. Player 1 in UII has less precise information on player 2's discount factor than player 1 in ICRub85 where δ_2 only has two possible values.

[OR86] analyzes a bargaining game where players learn others' specific characteristics (bargaining powers) which are drawn from a probability distribution. The major differences between the incomplete information bargaining problems in [OR86] and the UII are at two key aspects (1) [OR86] has several sellers. The buyer can switch sellers. Instead the UII is designed for two players, one of them only knows the other's discount factor is from a distribution. (2) In [OR86], sellers and the buyer have correlated values on the indivisible object. Their valuations to the object are partially private. The buyer wants to learn the sellers' valuations as well as their bargaining powers [FLT85]. By contrast, in UII the bargaining object, the cake, is public information. The sole purpose of learning is to discover the other player' discount factor. [GP85] also analyzes an incomplete information bargaining game on an indivisible object and the buyer's type is from a probability distribution.

6.2.3 Unilateral Ignorance Information - UGI

We introduce another incomplete information bargaining problem based on CRub82 problem. Player 1 has no information about the value of δ_2. However, player 2 knows the value of δ_1.

We call this problem the unilateral ignorance information bargaining problem, abbreviated as UGI.

[FT83] solves a two-person two-period bargaining problem under incomplete information. A buyer and a seller bargain over an indivisible object. The valuation of the seller to the object is common knowledge but the buyer's valuation is private. [FT83] proves the existence of an unique Perfect Bayesian Equilibrium.

ICRub85, UII and UGI are one-sided incomplete and asymmetric information bargaining problems. One-sided incomplete information means that only one of two players has incomplete information about the game. Asymmetric information means that two players have different information about the game. The following one is a two-sided incomplete information problem.

6.2.4 Bilateral Ignorance Information - BGI

In the bilateral ignorance information bargaining problem, abbreviated as BGI, no player has any information about each other's discount factor. [FT83] studies a similar bargaining problem where neither player knows his opponent's valuation to the bargaining object. [FT83] proves that there are multiple equilibria for the two-sided incomplete information problem. [FT83] bargaining problem is different from BGI in that the cake in BGI is divisible and players in BGI have time pressures to make an agreement soon.

6.3 Assumptions of Players' Boundedly Rationality

The set of assumptions on players' bounded rationality in Section 3.3 is fundamental for players in incomplete information bargaining problems as well.

Assumptions **A-1** to **A-9**, excluding **A-4**, are applicable to the CCGP system for incomplete information bargaining problems. The **A-4** for CRub82 problem assumes that both players have complete information about the game. In incomplete information problems however, at least one of two players does not have complete information about the game. Accordingly their terminal sets of genetic programs are different from each other. We update the **A-4** in Section 3.3 to the following:

The two bargaining players have different information. In terms of evolutionary algorithms, terminal sets in GP set-ups (Table 3.3) of the two populations of CCGP system are not the same. The strategy representation, learning mechanism and the game fitness functions of the two populations are the same. Players should still be regarded as having the same level of learning ability after such a modification on assumptions.

6.4 Constraints in Incomplete Information Bargaining Problems

The constraints of CRub82 bargaining problem are defined in Section 5.3. For an incomplete information problem, some features of these constraints change. In incomplete information bargaining problems, even a perfectly rational economic man having a larger discount factor is likely to get a smaller portion of a cake than a player who has a lower discount factor. The reason is that incomplete information may reversely influence a player's bargaining power. Therefore, the constraint properties for incomplete information bargaining problems are:

The hard constraint **C 1** $x_i \in (0, 1]$ is sustained in that a share of partition more than a cake or negative is impossible.

The soft constraints **C 2** and **C 3** are still valid, but their effects on the bargaining outcomes become unclear without substantial game-theoretic knowledge and analysis.

6.5 System and Experiment Design - Adaptation of CCGP

Constraint-driven Co-evolutionary Genetic Programming system (CCGP) is used to evolve co-adapted strategies for CRub82 bargaining problem and is designed to adapt the variations of CRub82 bargaining problem. The CCGP system is modified, mainly on the genetic programs' terminal sets and the fitness functions to match the specifications of four incomplete information bargaining problems.

The easy adaptation of CCGP to incomplete information bargaining problems demonstrates again the choice of adopting the two-population co-evolutionary system instead of the one-population co-evolutionary system, especially when two players have different information. Other reasons of choosing two-population system are detailed in Section 3.4 and Appendix C.

6.5.1 GP Terminal Sets

The four incomplete information bargaining problems and CRub82 complete information bargaining problem differ from each other on the players' information completeness about the other player's discount factor. For the ICRub85, player 1 with the probability ω_0, believes that player 2's actual discount factor is δ_w. Player 1's information set is therefore $\{\delta_1, \delta_w, \delta_s, \omega_0\}$. Added the size of cake 1 and the -1 to change the sign, the terminal set for the genetic programs in player 1' s population is $\{\delta_1, \delta_w, \delta_s, \omega_0, 1, -1\}$. Similarly, for the UGI problem player 1 has no information about player 2's discount factor and he has to guess the value of δ_2 from random. So his information set is $\{\delta_1, r_2\}$. r_2 is a random variable.

Table 6.4 lists the information sets of the two players in these bargaining problems. Information sets adding $\{1, -1\}$ become the terminal sets. In terminal sets, r_1, r_2 and r' are

Bargaining Problems	*Player 1's Information set*	*Player 2's Information set*
CRub82	$\{\delta_1, \delta_2\}$	$\{\delta_1, \delta_2\}$
ICRub85	$\{\delta_1, w_0, \delta_w, \delta_s\}$	$\{\delta_1, \delta_2\}$
UII	$\{\delta_1, r'\}$	$\{\delta_1, \delta_2\}$
UGI	$\{\delta_1, r_2\}$	$\{\delta_1, \delta_2\}$
BGI	$\{\delta_1, r_2\}$	$\{r_1, \delta_2\}$

Table 6.4: Information sets for the two players in the five bargaining problems. GP Terminal sets are the information sets added $\{1, -1\}$.

real and random variables.

r_1 and r_2 return values from an uniform distribution in the range of greater than or equal to 0.0 and less than 1.0. r' is from an approximative Gaussian distribution with the mean δ_2 and the deviation 0.1 but limited to the constraint $0 < r' < 1$. The value of r' is generated by sampling Gaussian values until a value within (0, 1) is sampled. Values of variables r_1, r_2 and r' keep as constants once created. A random variable has only one value in a genetic program and may have different values in different genetic programs. Mutation or crossover may bring new random values, but do not change the values of those existing.

The bargaining process remains unchanged. The strategy representation, the utility function and the game fitness function are the same as those defined in Section 3.5 and 3.5.4. The definition of Game Fitness of a strategy $s(g_i)$ is the average utility of what $s(g_i)$'s gains from the agreements with strategies in the co-evolving population J which satisfy the hard constraint:

$$GF(s(g_i)) = \frac{\sum_{j \in J} u_{s(g_i) \rightarrow s(g_j)}}{m} \tag{6.5}$$

In theory if two players perpetually disagree, both players obtain utility 0. In experiments, if players do not agree after 10 time intervals, both get utility 0.

6.5.2 Fitness Function

The application of the Incentive method to the fitness function can reduce the search space by integrating problem-specific knowledge on both hard and soft constraints (Section 5.4).

According to the constraints in incomplete information bargaining problems in Section 6.4, we continue using Sensibility Measure and Evaluation of Attribution. Both are formally defined in Section 5.4. For incomplete information bargaining problems, Sensibility Measure and Evaluation of Attribution do not play the role of separating the preferred individuals from the rest. Instead, they are used to give incentive values to infeasible individuals.

For an incomplete information bargaining problem, a genetic program's fitness $F(g_i)$ is:

$$F(g_i) = \begin{cases} GF(s(g_i)) + 3 \\ \quad \text{if } g_i \in (0, 1] \\ ATT(i) + ATT(j) - e^{\frac{-1}{|g_i|}} \\ \quad \text{if } g_i \notin (0, 1] \end{cases} \tag{6.6}$$

where $ATT(i) + ATT(j) - e^{\frac{-1}{|g_i|}}$ is restricted within $[0, 2]$ as we explain in Section 5.4. Thus genetic programs satisfy the hard constraint are guaranteed to obtain higher fitness values than the rest.

From the analysis above, we can see that only the fitness function and the terminal sets of GP set-up need to do minor changes. The rest of the system can be inherited directly. This saves our time and efforts. Comparably, it is much harder for game theorists to reuse the game-theoretic proofs for variant bargaining problems.

6.6 Experimental Results and Observations

For an incomplete information bargaining problem, each of its game setting is tested with 100 runs starting with different random sequences.

Problem	*Raw* $\bar{x_1}$	*Analyzed* $\bar{x_1}$	*Raw* $\bar{t}$	Analyzed $\bar{t}$	*Raw* σ	*Analyzed* σ
ICRub85	Table E.1, E.2, E.3	Table 6.10	Table E.4 E.5, E.6	Table 6.11	Table E.1 E.2, E.3	Table 6.12
UII	Table E.7	Table 6.10	Table E.7	Table 6.11	Table E.7	Table 6.12
UGI	Table E.8	Table 6.10	Table E.8	Table 6.11	Table E.8	Table 6.12
BGI	Table E.9	Table 6.10	Table E.9	Table 6.11	Table E.9	Table 6.12

Table 6.5: Summary of locations of the raw and the analyzed experimental results for these four incomplete information problems. The raw experimental data are in Appendix E. The analyzed data are inserted within this chapter.

The experimental results of ICRub85 are compared with its game-theoretic solutions PBE in Subsection 6.6.1. Game-theoretic solutions of UII, UGI and BGI are unknown, so their experimental results are compared with two benchmarks: the experimental results of CRub82 as well as the experimental results of ICRub85. Such comparisons are taken on three statistic results:

- $\bar{x_1}$, the observed shares of agreements (Subsection 6.6.2);
- $\bar{t}$, the average bargaining time and efficiency of agreements (Subsection 6.6.3);
- σ, stationarity and evolutionary time for stabilization (Subsection 6.6.4).

Table 6.5 lists the locations of the raw data and the analyzed data for four incomplete information problems.

6.6.1 Experimental Results of ICRub85

This subsection measures the relationship between the experimental results of ICRub85 and the game-theoretic solutions PBE.

Measures of the experimental results of ICRub85

For this specific problem, the experimental results $\bar{x_1}$s are examined as follows:

1. According to the game-theoretic analysis, given a game setting $(\delta_1, \delta_2, \delta_2', \omega_0)$, there exists an unique PBE x_1^*. The experimental results $\bar{x_1}$ should be compared with its corresponding x_1^*.

2. To understand whether and/or how δ_2' and ω_0 contribute to the experimental results $\bar{x_1}$, the relationship of $\bar{x_1}$ and the equilibria V_s, x^{ω_0} and y^{ω_0} need be examined on three contingencies: (1) $\omega_0 < \omega^*$; (2)$\delta_2 = \delta_s$ and $\omega_0 > \omega^*$; (3) $\delta_2 = \delta_w$ and $\omega_0 > \omega^*$. For $\omega_0 = \omega^*$, [Rub85] does not provide the game-theoretic solutions, therefore we do not examine this condition.

3. There are three possible equilibriums V_s, y^{ω_0}, and x^{ω_0} as in Table 6.3. The unique PBE is chosen from them on the condition of $\omega_0 > \omega^*$ and the real value of δ_2. We plan to analyze the association between the experimental results and the three possible equilibriums V_s, y^{ω_0}, and x^{ω_0} .

Notations

δ_2 is player 2's actual discount factor. δ_2' is another possible value of player 2's discount factor in player 1' initial belief. $\delta_2' \in \{\delta_w, \delta_s\}$. ω_0 is the possibility of player 1's belief of 2's type being $MIN(\delta_2, \delta_2')$. Experimental result x_1 is player 1' share from the bargaining agreement which is made by the best-of-generation genetic programs at the 300^{th} generation. $\bar{x_1}$ is the average of x_1s of 100 runs of a given game setting. $\bar{t}$ is the average bargaining time for reaching agreements of 100 runs of a given game setting. σ is the deviation of the x_1s of 100 runs for a given game setting.

Definition 6.1: Experimental results select the equilibrium if this equilibrium has the minimal absolute variations from the observed shares $\bar{x_1}$.

$\bar{x_1}$ chooses V_s if $MIN(|\bar{x_1} - V_s|, |\bar{x_1} - x^{\omega_0}|, |\bar{x_1} - y^{\omega_0}|) = |\bar{x_1} - V_s|$;

$\bar{x_1}$ chooses x^{ω_0} if $MIN(|\bar{x_1} - V_s|, |\bar{x_1} - x^{\omega_0}|, |\bar{x_1} - y^{\omega_0}|) = |\bar{x_1} - x^{\omega_0}|$;

$\bar{x_1}$ chooses y^{ω_0} if $MIN(|\bar{x_1} - V_s|, |\bar{x_1} - x^{\omega_0}|, |\bar{x_1} - y^{\omega_0}|) = |\bar{x_1} - y^{\omega_0}|$;

Observations

We test 54 game settings for the ICRub85 problem. $\delta_1 \in \{0.1, 0.5, 0.9\}$, $\delta_2 \in \{0.1, 0.5, 0.9\}$, $\delta_2' \in \{0.1, 0.2, 0.4, 0.8, 0.9\}$, and $\omega_0 \in \{0.1, 0.5, 0.9\}$ [3]. Tables E.1, E.2 and E.3 in Appendix E list the raw experimental results of $\bar{x_1}$s, their σs and x_1^*s in PBE. Tables E.4, E.5 and E.6 list the bargaining time of PBE agreements t^* and of the experiments $\bar{t}$.

Association between $\bar{x_1}$ and PBE x_1^* and between $\bar{x_1}$ and SPE x_1^*

For the 54 tested game settings, $\bar{x_1}$s approximate PBE x_1^*s. Hypothesis $(\bar{x_1} - x_1^*) = 0$ on the total 54 game settings cannot be rejected with 95% confidence (see #1 in Table 6.6). This means that there is no statistical evidence to show that observed $\bar{x_1}$s are significantly different from PBE x_1^*s. Please note that for a particular game setting $\bar{x_1}$ may not ideally approximate x_1^* but the set of $\bar{x_1}$s approximate the set of x_1^*s for the 54 tested game settings. In detail, we split the 54 game settings into two groups and have t-test on each group: (i) when $\omega_0 < \omega^*$ (30 game settings), $\bar{x_1}$s approximate PBE x_1^*s (see #1.1 in Table 6.6). (ii) when $\omega_0 > \omega^*$ (24 game settings), $\bar{x_1}$s approximate PBE x_1^*s (see #1.2 in Table 6.6).

Linear regression tests are used to further measure the correlation between x_1^* and $\bar{x_1}$. The linear regression statistics over x_1^* and $\bar{x_1}$ (see #1 in Table 6.6) $R^2 = 0.5364$ show that $\bar{x_1}$ and PBE x_1^* are not in a linear correlation.

[3]The full combination of $\delta_1 \times \delta_2 \times \delta_2' \times \omega_0$ has totally 135 possibilities. Considering the computational resources, we only choose two values from the set of δ_2' for every combination of $\delta_1 \times \delta_2 \times \omega_0$. The principle of choosing δ_2' is that one value of δ_2' is close to δ_2 and another one is far from δ_2. So the number of the total tests is $\delta_1 \times \delta_2 \times 2 \times \omega_0 = 54$.

Moreover, $\bar{x_1}$s also approximate SPE x_1^*s of the CRub82 bargaining problem. Hypothesis $(\bar{x_1} - SPEx_1^*) = 0$ can not be rejected with 95% confidence. In addition the linear regression statistics of 54 game settings ICRub85 $\bar{x_1}$ over CRub82 SPE x_1^* suggest a weak linear correlation between $\bar{x_1}$ and SPE x_1^* (see #3 in Table 6.6). When player 1 correctly guesses δ_2, the theoretic solution PBE of this incomplete information bargaining problem coincides the SPE of the CRub82. Experimental results exhibit a strong consistency with the SPEs as if player 1 correctly guesses the value of δ_2 before bargaining begins or when $\omega_0 = 1$ if $\delta_2 = \delta_w$ or $\omega_0 = 0$ if $\delta_2 = \delta_s$.

Association between $\bar{x_1}$ and three possible equilibriums

Table 6.7 and 6.8 display the number of and the frequency of a game-theoretic equilibrium chosen by experimental results. From them, we find that some game-theoretic equilibria are less frequently chosen by experimental results.

1. For 82% game settings of PBE y^{ω_0}s, experiments choose V_ss. This implies that after the evolutionary process player 1 discovers that the type of player 2 is δ_s, so he shifts the equilibrium from y^{ω_0} to V_s to increase the efficiency of agreements from one time delay y^{ω_0} to no delay V_s. For example: Table E.1 #9. PBE $x_1^* = 0.0561$ and the experimental result $\bar{x_1} = 0.6356$. This $\bar{x_1}$ approximates to $x^{\omega_0} = 0.6224$. This shows that experiments of this game setting do not choose the PBE but another possible equilibrium. From the summary of theoretic solutions in Table 6.3, x^{ω_0} and V_s theoretically achieves at time 0 but x_1^* (here y^{ω_0}) arrives at time 1. When an agreement settles at time 1, both players pay bargaining costs. This agreement is inefficient. This example and other 8 tested game settings whose experiments choose V_ss rather than PBE y^{ω_0}s show that experiments tend to choose more efficient equilibriums.

#	Observations	Statistic evidence(s) from experiments
1	Observed $\bar{x_1}$s approximate PBE x_1^*s (54 game settings)	Hypothesis $(\bar{x_1} - x_1^*) = 0$ can not be rejected with 95% confidence. (tsv = -1.5801 and tc = 2.0057) Linear Regression Statistics $R^2 = 0.5364$.
1.1	$\bar{x_1}$s approximate PBE x_1^*s when $\omega_0 < \omega^*$ (30 game setting)	Hypothesis $(\bar{x_1} - x_1^*) = 0$ can not be rejected with 95% confidence. (tsv = -1.1259 and tc = 2.0017)
1.2	$\bar{x_1}$s approximate PBE x_1^*s when $\omega_0 > \omega^*$ (24 game settings)	Hypothesis $(\bar{x_1} - x_1^*) = 0$ can not be rejected with 95% confidence. (tsv = 0.01392 and tc = 2.0129)
2	The behavior of the final populations remain evolutionary stable.	For a game setting, the deviation of the results $\bar{x_1}$ of 100 runs is very small (less than 0.05). see Table E.1, E.2 and E.3
3	$\bar{x_1}$s approximate SPE x_1^*s of the CRub82.	Hypothesis $(\bar{x_1} - \text{SPE}x_1^*) = 0$ can not be rejected with 95% confidence. (tsv = -0.4199 and tc = 1.9826) Linear Regression Statistics $R^2 = 0.9403$.
4	Results of shares ICRub85 $\bar{x_1}$ are significantly similar to results CRub82 $\bar{x_1}$. ICRub85 $\bar{x_1}$s and CRub82 $\bar{x_1}$s are strongly linearly related.	Hypothesis (CRub82 $\bar{x_1}$−ICRub85 $\bar{x_1}$) = 0 can not be rejected with 95% confidence. (tsv = -0.5586 and tc = 1.9826) Linear Regression Statistics $R^2 = 0.9939$.

Table 6.6: Observations and Statistic evidence(s) from experiments. tsv is the t-Statistic value; tc is the t-Critical value; PBE is the Perfect Bayesian Equilibrium for ICRub85; SPE is Subgame Perfect Equilibrium for CRub82; ω^* is the threshold value defined in Equation (6.3).

$\bar{x_1}$ *choose*	*PBE* x_1^*			$\bar{x_1}$
	V_s	x^{ω_0}	y^{ω_0}	subtotal
V_s	8	0	9	17
x^{ω_0}	18	12	1	31
y^{ω_0}	4	1	1	6
PBE subtotal	30	13	11	

Table 6.7: For ICRub85, the number of PBE x_1^* chosen by $\bar{x_1}$.

$\bar{x_1}$	PBE x_1^*		
choose	V_s	x^{ω_0}	y^{ω_0}
V_s	27%	0%	82%
x^{ω_0}	60%	92%	9%
y^{ω_0}	13%	8%	9%

Table 6.8: For ICRub85, the percentage of PBE x_1^* equilibriums chosen by $\bar{x_1}$.

2. That 92% of PBE x^{ω_0}s are chosen by experiments. Encountering a 2_w, the best offer for player 1 is x^{ω_0}, $x^{\omega_0} > MAX\{V_s, y^{\omega_0}\}$. x^{ω_0} is the largest shares for player 1 among the three equilibria, and it should be also acceptable to player 2_w. That more than 90% experiments whose PBE is x^{ω_0}s choose x^{ω_0}s suggests that the actual type of player 2 being δ_w, is uncovered by player 1 after the evolutionary training.

3. Theoretically bargaining with 2_w, the best equilibrium for player 1 is x^{ω_0}. Bargaining with 2_s the best equilibrium for player 1 is V_s. As the PBE V_s can be accepted by both δ_w and δ_s, 2_w takes advantage of player 1 when player 1 mistakenly thinks 2 is 2_s. From the 30 game settings whose PBE is V_s, there are 18 game settings that have $\delta_2 = \delta_w$ from which 9 choose x^{ω_0}, 7 choose V_s and 2 choose y^{ω_0}. There is an example. The game setting #4 in Table E.1 : $(\delta_1, \delta_2, \delta_2', \omega_0) = (0.1, 0.1, 0.9, 0.1)$ tells that at the beginning, player 1 believes that very likely player 2 is 2_s, but actually player 2 is 2_w whose discount factor is δ_w. In theory, player 1 prepares to encounter 2_s, so he uses V_s which is accepted by both 2_s and 2_w. The experimental result $\bar{x_1} = 0.9273$ do not choose the PBE (V_s) but instead choose $x^{\omega_0} = 0.9009$. Why? From Table 6.3 we know that theoretically, V_s is efficient for both $\delta_2 = \delta_w$ or $\delta_2 = \delta_s$. x^{ω_0} is efficient for $\delta_2 = \delta_w$. Experiments help player 1 to choose the equilibrium which reflect the real value of player 2's discount factor $\delta_2 = \delta_w = 0.1$ and $\delta_2' = \delta_s = 0.9$. Such results show that evolution is capable of reducing player 1's disadvantage caused by his incomplete

information on the type of player 2.

From the above analysis, experimental results do not always choose the PBE, and some times choose one of the other two possible equilibriums. One reason that experimental results choose one of other equilibriums but not the PBE, is probably that the other equilibrium may be more efficient (less bargaining time so less bargaining cost) than the PBE or that the other equilibrium is more likely to survive under selective pressures. Let take this point a bit further, evolution of population 1 actually discovers the real value of δ_2. After learning by means of the co-evolutionary algorithm, genetic programs in the first player population behave as if they know the actual type of player 2 simply because these genetic programs perform better than those which do not guess correctly about δ_2. Under the selection pressure, such genetic programs that correctly guesses out δ_2 survive and dominate the population.

Summary of ICRub85 Experimental Results

Observations and explanations suggest that after evolutionary training, the privilege of knowing more over the other player shows less impact in experiments than what is expected by game theory. Instead, that the efficiency of solutions has a higher priority in the evolutionary system than the information incompleteness, is observed in experiments.

6.6.2 Partition of Cake in Agreement

Game-theoretic equilibriums for UII, UGI and BGI problems have not been solved yet. Their experimental results are evaluated against the experimental results of ICRub85 and the experimental results of CRub82. In this subsection, the averages of shares, $\bar{x_1}$s in the observed agreements across these five problems are investigated.

Table 6.9 lists the $\bar{x_1}$s for the 5 problems. Each pair of discount factors is the actual

Game Setting (δ_1, δ_2)	*C Rub82* $\bar{x_1}$	*IC Rub85* $\bar{x_1}$	*UII* $\bar{x_1}$	*UGI* $\bar{x_1}$	*BGI* $\bar{x_1}$
0.1 , 0.1	1.0000	0.9162	0.9495	0.9517	0.9517
0.1 , 0.5	0.6886	0.6273	0.6405	0.6407	0.6413
0.1 , 0.9	0.1967	0.1486	0.1538	0.1452	0.1502
0.5 , 0.1	0.9993	0.9304	0.9522	0.9524	0.9524
0.5 , 0.5	0.6782	0.6309	0.6459	0.6468	0.6468
0.5 , 0.9	0.1137	0.1072	0.1099	0.1077	0.1073
0.9 , 0.1	0.9967	0.9269	0.9562	0.9594	0.9594
0.9 , 0.5	0.7479	0.7151	0.7266	0.7287	0.7304
0.9 , 0.9	0.4902	0.4708	0.4838	0.4819	0.4965

Table 6.9: For the five listed problems, each pair of discount factors is the actual values of δ_1 and δ_2. We examine whether the actual (δ_1, δ_2) make decisive roles in dividing the cake. Other values on player 2's discount factor are excluded.

	R^2 *Values*
ICRub85	0.99750
UII	0.99801
UGI	0.99760
BGI	0.99699

Table 6.10: R^2 values of linear regression tests. The $\bar{x_1}$s of CRub82 are the input y range. The $\bar{x_1}$s for a specified incomplete information problem are the input of x range.

values of δ_1 and δ_2. Other values on player 2's discount factor are excluded. An obvious pattern displays: given the real values of discount factors, the $\bar{x_1}$s across the five problems are very close to each other. From the linear regression statistics, the R^2 on the 5 sets of $\bar{x_1}$s yield significantly high values nearly 1 as shown in Table 6.10. They are strongly linear correlated among them. As we know, the essential difference among these problems are players' information completeness. Thus the information completeness does not attribute much to the similarity of solutions among the five problems. Instead, such observations suggest that the actual values of δ_1 and δ_2 which are applied to the utility functions $x_1\delta_1^t$ and $x_2\delta_2^t$ ultimately determine the bargaining outcomes on partition of cake.

When there is one unique equilibrium, the experimental results are attracted to that equilibrium, for example the experimental results of CRub82 problem shown in Chapter 4. If there exist multi-equilibria for a problem, for instance ICRub85 and probably for UGI, UII and BGI, the experimental results probably converge to one of equilibria where $u_1 + u_2 = 1$ holds (u_i is the utility that player i receives from an agreement). This is one of the most efficient equilibriums.

6.6.3 Bargaining Time and Efficiency of Agreement

Bargaining cost increases exponentially over time. This causes utilities to decrease. The bargaining time therefore determines the efficiency of agreements. Any delay ($t > 0$) causes bargaining costs to both players. Thus the sum of players' utilities is less than the size of cake, which means that the cake is not divided efficiently. If the sum of two players' utilities in an agreement is the exact size of the cake, this happens only if $t = 0$, the agreement is the most efficient one. The most efficient agreement reaches at the moment when the game starts thus no bargaining cost is incurred.

Tables E.4, E.5 and E.6 report experimental results on bargaining time of ICRub85 problem. Table 6.11 summarizes the average bargaining time $\bar{t}$ for the five problems. The bargaining time $\bar{t}$s of the game settings $(0.1, 0.1)$ $(0.5, 0.1)$ and $(0.9, 0.1)$ are all 0s across all the five problems. It is interesting to note that these three game settings all have $\delta_2 = 0.1$, although δ_1 varies from 0.1 to 0.9. Increasing the value of δ_2, the $\bar{t}$ correspondingly rises. This can be confirmed by comparing the $\bar{t}$ values inside the groups where each group has its δ_1 keeps unchanged: $\{(0.1, 0.1), (0.1, 0.5), (0.1, 0.9)\}$,$\{(0.5, 0.1)$, $(0.5, 0.5)$, $(0.5, 0.9)\}$ or $\{(0.9, 0.1), (0.9, 0.5), (0.9, 0.9)\}$. When $\delta_2 = 0.9$, $\bar{ts}$ of { (0.1, 0.9), (0.5, 0.9) and (0.9, 0.9)} only increase greatly when δ_1 approaches 0.9. Furthermore, when both δ_1 and δ_2 are not large enough (less than 0.9 in experiments shown), the impact of δ_1 on bargaining time $\bar{t}$ is

Game Settings (δ_1, δ_2)	*C Rub82* $\bar{t}$	*IC Rub85* $\bar{t}$	*UII* $\bar{t}$	UGI $\bar{t}$	BGI $\bar{t}$
0.1 , 0.1	0.00	0.00	0.00	0.00	0.00
0.1 , 0.5	0.00	0.04	0.03	0.04	0.04
0.1 , 0.9	0.14	0.34	0.35	0.38	0.30
0.5 , 0.1	0.00	0.00	0.00	0.00	0.00
0.5 , 0.5	0.00	0.05	0.04	0.03	0.03
0.5 , 0.9	0.13	0.20	0.34	0.28	0.30
0.9 , 0.1	0.00	0.00	0.00	0.00	0.00
0.9 , 0.5	0.28	0.32	0.30	0.31	0.25
0.9 , 0.9	3.82	3.99	4.05	3.90	3.32

Table 6.11: The average bargaining time $\bar{t}$s of five bargaining problems. Each pair of discount factors is the actual values of δ_1 and δ_2.

relatively small. In such cases, all $\bar{t}$s are less than 0.35, so most agreements are settled at the time 0. By contrast when both δ_1 and δ_2 are large enough, the bargaining time prolongs dramatically. Shown in Table 6.11 when the δ_2 rises from 0.5 to 0.9 while $\delta_1 = 0.9$, the bargaining time $\bar{t}$s go about 10 times longer. A large discount factor makes the efficiency of agreements decrease, but not to an extraordinary degree due to the players' very low costs on time. For example, t of one run of the game setting $(0.9, 0.9)$ is 3, then the average bargaining cost of a strategy in player 2' population is therefore $(1 - 0.9)^3 = 0.001$. t of another run of $(0.1, 0.9)$ is 0.20, then the average cost of a strategy in player 2' population is $(1-0.9)^{0.2} = 0.63$. These two examples show that even the average bargaining time of (0.9, 0.9) is 15 times longer than that of the (0.1, 0.9), the average bargaining cost of a player 2' strategy in (0.9, 0.9) is only 0.16% of the cost of a player 2' strategy in (0.1, 0.9).

Having found that $\bar{t}$s highly associate with the discount factors, we notice that given (δ_1, δ_2), it hardly distinguishes the type of players' information completeness from the difference of the experimental data $\bar{t}$s. In summary, the bargaining time is more likely influenced by the actual values of discount factors than by the players' information completeness.

Problem	*Average* σ	*Maximal* σ	*Minimal* σ
ICRub85	0.0264	0.0416	0.0172
UII	0.0272	0.0581	0.0051
UGI	0.0260	0.0641	0.0026
BGI	0.0295	0.0643	0.0088

Table 6.12: Average σs, maximal σs and minimal σs of four incomplete information problems.

6.6.4 Stationarity of Agreement

Stationarity of Agreements and Evolutionary Stability

From experimental results, players' behaviors in the final populations remain stationary. For a game setting of ICRub85 bargaining problem, the deviation σ of 100 x_1s from 100 runs is very small, less than 0.05, see Table 6.12. The raw data of σs are in Table E.1, E.2 and E.3. This indicates the players' stationary behaviors at the end of the co-evolving processes. No player (population) prefers to withdrawal from the chosen equilibrium (co-adaptation) even in the presence of other possible offers or counter-offers (mutations).

Similarly for UII, UGI and BGI bargaining problems, their average σs, maximal σs and minimal σs are very small (Table 6.12), showing that agreements made by the final populations are stationary.

Evolutionary Time to Stabilize

Evolutionary algorithms are used as a stochastic search method. The search process starts from random points scattered over the search space and gradually confines into the more promising spaces. Usually precise and sufficient information about the problems helps an efficient and effective search. Having the full information about each other, the experimental results of CRub82 are expected to converge to equilibrium(s) more quickly and more precisely than does an incomplete information bargaining problem.

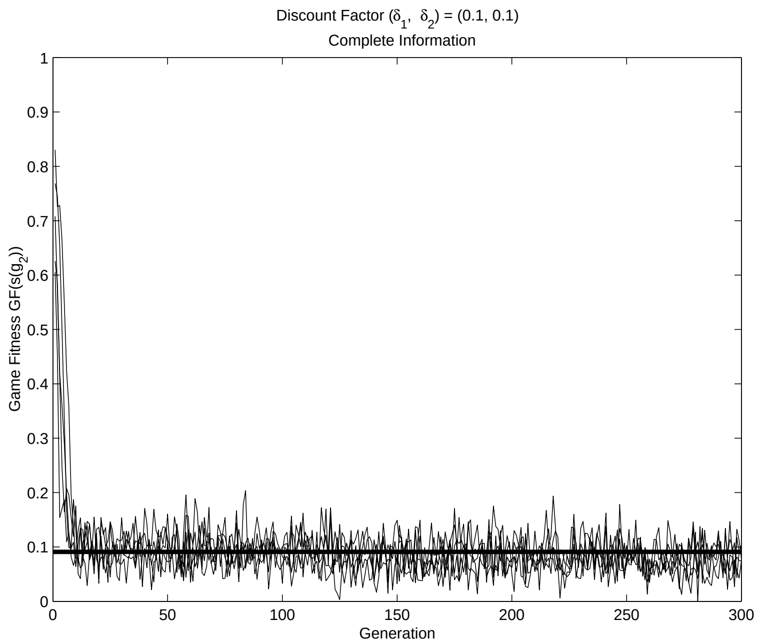


Figure 6.1: The game fitness of the best-of-generation genetic programs in player 2's population of a CRub82. 5 runs are shown. $\delta_1 = 0.1$, $\delta_2 = 0.1$

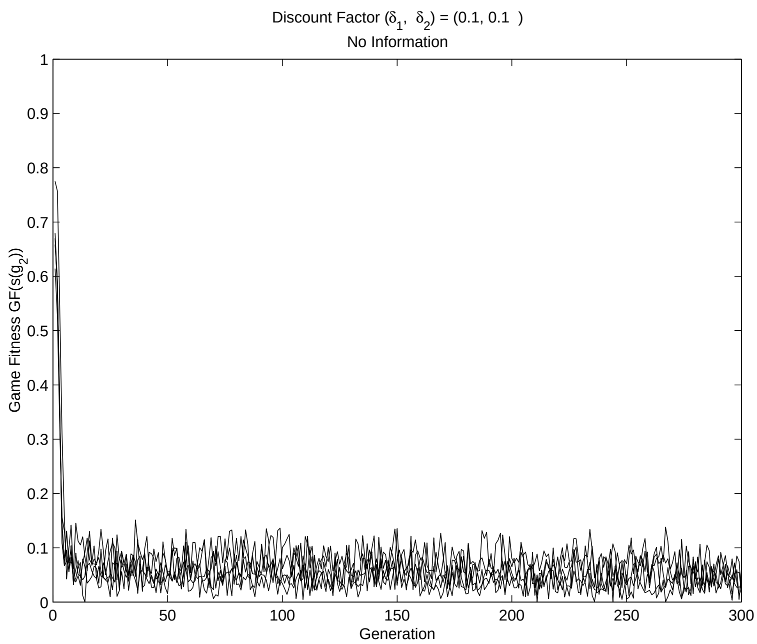


Figure 6.2: The game fitness of the best-of-generation genetic programs in player 2's population of a BGI bargaining problem. 5 runs are shown. $\delta_1 = 0.1$, $\delta_2 = 0.1$

Two sets of examples are given to address this concern. The first example includes a CRub82 bargaining problem and a BGI problem, both with the same discount factors ($\delta_1 = 0.1$, $\delta_2 = 0.1$). The second example is a CRub82 problem and a BGI problem with the same discount factors ($\delta_1 = 0.9$, $\delta_2 = 0.4$). Five runs are plotted. Experiments of two bargaining problems in each set of above examples use the same sets of random sequences. Figure 6.1 and 6.2 are for the first example, illustrating the game fitness of the best-of-generation genetic programs in player 2' populations. In Figure 6.1, both players have the complete information. The game fitness is around $0.8 \sim 0.9$ at the very beginning, quickly dropping down to the area $0.05 \sim 0.15$ and fluctuates in the proximity of SPE $x_2^* = 0.0909$. Surprisingly, Figure 6.2 shows the similar phenomenon, although neither player has any information about the other at all. In this example, the information about the other player has little effect on the time for finding out the co-adaptive strategies and stabilization. For the second example, Figure 6.3 and Figure 6.4 illustrate the game fitness of the best-of-generation genetic programs in player 1' population. It is interesting to see that, from as early as the first generations, the five runs of BGI leap to the point at which the population stabilizes for 300 generations. Some runs for the CRub82, however spend a significantly longer time to reach it: approaching the SPE in a series steps. This observation is probably due to the fact that for the BGI problem, a large number of random values as terminal nodes in the genetic programs of initial populations provide a greater diversity, scattering over much larger search space. In contrast, the terminal set of genetic programs for CRub82 only has δ_1 and δ_2, which restricts the diversity.

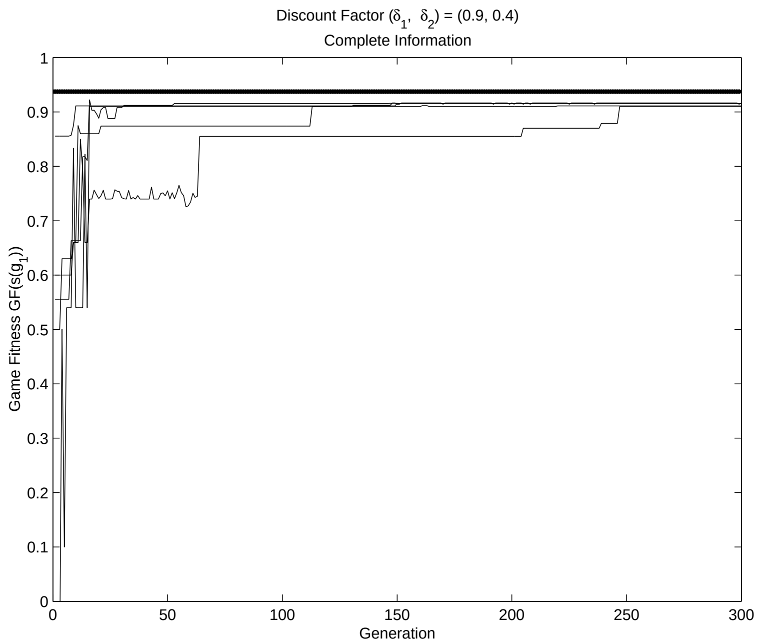


Figure 6.3: The game fitness of the best-of-generation genetic programs in player 1's population of a CRub82. 5 runs are shown. $\delta_1 = 0.9$, $\delta_2 = 0.4$

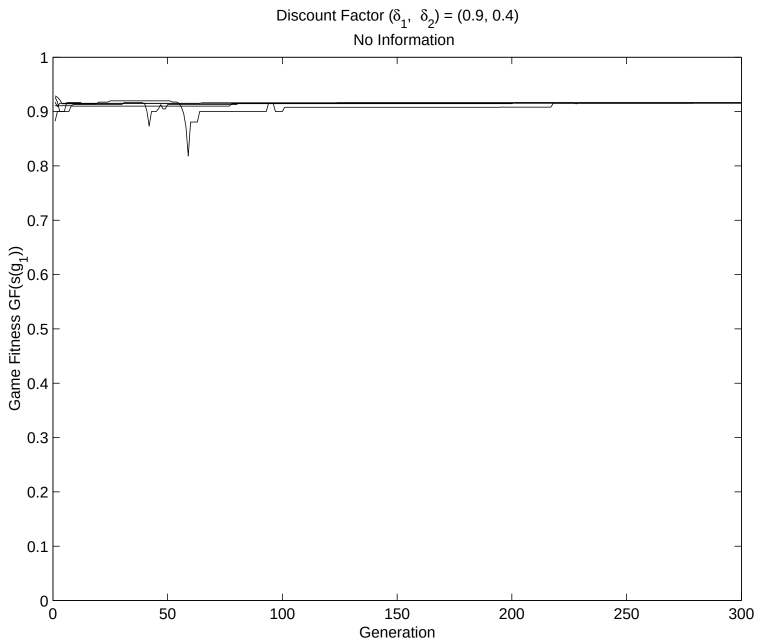


Figure 6.4: The game fitness of the best-of-generation genetic programs in player 1's population of a BGI bargaining problem. 5 runs are shown. $\delta_1 = 0.9$, $\delta_2 = 0.4$

6.6.5 Computational Resources

The computational resources require for running experiments for incomplete information bargaining problems are almost the same as those for complete information bargaining problem: a Linux machine with an $athlon2400$ processor runs about 1 hour to test a game setting with 100 runs.

6.7 Discussion

The common sense tells us that if a player has more game-related information he should attain more advantages in dividing a cake. However, the experimental results by co-evolutionary algorithms display no apparent indication that the amount or precision of information affects bargaining outcomes on partitions of the cake. Moreover, from the viewpoint of game-theoretic analysis, bargaining problems under both one-sided and two-sided incomplete information imply delay and inefficiency [FLT85]. But experimental results on bargaining time provide no convincing evidence that delays are the consequences of the incomplete information.

Equilibria selected by experiments are always the most efficient ones for the evolutionary systems. They are reached because, through the selection and variation mechanism (genetic learning), the information that exists but is unknown to the other population is induced via fitness evaluations and then embedded in the individuals at a genetic level. This means that the co-evolutionary process prefers (selects for reproduction) those genetic programs that act *as if* they were fully informed about all aspects of the game, because these individuals perform better.

The power of evolutionary algorithms achieve high efficiency and stationarity. This feature may cast doubts on applying evolutionary algorithms to one-off games. Some economists tend to believe that in one-off games, incomplete information certainly make inefficient delays, cheating, wrong signalling, unstable and/or unfair agreements possible. However, such a belief does not refute the existence of *honest* and *fair* players who make efficient and stable agreements.

On the other hand, although one of the efficient and stationary equilibria has always been experimentally found for a bargaining game, other possible equilibria have been discarded. This characteristic may limit the applications of evolutionary algorithms from such games that need to emphasize the importance of more than one feasible and preferable solutions. In addition, evolutionary algorithms' ability to discover private information restricts its potential applications from observing special features of some one-off games where the chance and time for learning are extremely limited and random elements play a strong role. Jin [Jin] further discusses the modelling and the applications of evolutionary algorithms for incomplete information games.

6.8 Concluding Summary

In summary, this chapter investigates incomplete information bargaining problems by means of co-evolutionary algorithms and by using the established CCGP system with minor modifications. As observed, experimental results demonstrate that evolutionary populations always successfully approximate one of the most efficient and stationary equilibria for ICRub85 problem. Moreover, the experimental results of the four incomplete information problems exhibit linear correlations among them as well as with the experimental results of the CRub82 complete information problem in terms of the partition of cake in agreements, irrespective of players' information completeness. The bargaining time is more likely associated with actual values of discount factors than player(s)' information on the values of discount factors. In particular, the experiments of ICRub85 choose the equilibria which ensure the high efficiency of agreements as if the player's initial belief about the other player's discount factor is always correct. Moreover, it is unnecessarily to spend longer time in finding out such equilibria in the absence of complete information.

Experimental results on partition of cake stabilize at (one of) the most efficient and stationary equilibria. This approach therefore, is capable of providing reasonably good solutions for those bargaining problems that have no game-theoretic solutions available yet due to their complexity. The game-theoretic method can only find solutions for simple bargaining problems. When these bargaining problems are extended and modified even slightly, game theorists need considerable efforts to solve them. In contrary, the co-evolutionary system, in particular CCGP can be easily reused to deal with variants of bargaining problems within an affordable budget of human efforts, computational resources and time.

Chapter 7

CCGP for Bargaining Problem with Outside Options

7.1 Introduction

In the previous chapters, we develop a Constraint-based Co-evolutionary Genetic Programming, CCGP system and use it to study five bargaining problems. In any of these five problems, there is only one bargaining determinant: discount factor. The CCGP system generates experimental results that are efficient and stationary and approximate game-theoretic solutions.

This chapter studies a bargaining problem with two determinants: discount factors and outside options. Besides discount factors, outside options also influence bargaining outcomes. Players have complete information on discount factors and outside options. We abbreviate this problem as COO bargaining problem.

We aim to investigate whether the CCGP system is able to reuse and generate efficient and stationary solutions for COO bargaining problem which has two determinants.

In the remainder of this chapter, first of all the outside option bargaining problem and its game-theoretic solutions are recaped in Section 7.1.1. The assumptions on players' bounded rationality and the constraints of COO bargaining problem are specified in Section 7.2 and

7.3 respectively. CCGP is supposed to adapt to such differences on assumptions and constraints from those of bargaining problems studied in the previous chapters.

The experimental design to modify CCGP is presented in Section 7.4. We measure whether the experimental results from CCGP approximate game-theoretic solutions and whether such solutions exhibit efficiency and stationarity. Followed by experimental results and observations in Section 7.5, this chapter ends with concluding summary in Section 7.6 [1].

7.1.1 Recapitulation of Outside Option Bargaining Problem - COO

The outside option bargaining problem (COO) and it game-theoretic solutions are introduced in Section 2.3.3. They are recaped as follows.

In COO outside option bargaining scenario, when a player i encounters an offer or a counter-offer from player j, he can choose one of three choices: (1) acceptance thus an agreement is settled; (2) rejecting this offer and making a counter offer after one time interval; (3) quitting bargaining and taking his outside option w_i.

The existence of outside options potentially threats bargaining players with withdrawal and ending up with nothing from the undergoing bargaining [Bin85]. The idea of outside option can be illustrated by the following example bargaining scenario, which is originally given in [Mut99]. Imagine University A and academic economist B is bargaining over B's annual salary. B has been offered a job by another institution with a fixed and nonnegotiable salary w_B. w_B is B's outside option. To make this problem simpler, let's assume that University A does not have an outside option (of replacing B) at the point of bargaining. The bargaining between University A and academic economist B is used to exemplify how

[1] Initial work of this chapter has been published in [JT06].

#	x_1^*	*Conditions (AND)*		Category
		I	II	
a	μ_1	$w_1 \leq \delta_1\mu_1$	$w_2 \leq \delta_2\mu_2$	Category 1
b	$1 - w_2$	$w_1 \leq \delta_1(1 - w_2)$	$w_2 > \delta_2\mu_2$	Category 2
c	$\delta_2 w_1 + (1 - \delta_2)$	$w_1 > \delta_1\mu_1$	$w_2 \leq \delta_2(1 - w_1)$	Category 2
d	$1 - w_2$	$w_1 > \delta_1(1 - w_2)$	$w_2 > \delta_2(1 - w_1)$	Category 2
e	w_1	$w_1 + w_2 > 1$	-	Category 3

Table 7.1: SPE under 5 different conditions for COO bargaining problem. Player 1 makes the first offer. The shares in a SPE agreement is $(x_1^*, 1 - x_1^*)$ under the condition $\#a, \#b, \#c$ or $\#d$. Under the condition $\#e$, $x_1^* = w_1$ and $x_2^* = w_2$. Detailed explanation is available in Section 2.3.3.

the existence of and the values of outside options affect the bargaining outcome throughout this chapter.

The unique *Subgame Perfect Equilibrium* (SPE) solution x_1^* of outside option bargaining problem is stated in the Table 7.1 where,

$$\mu_1 = \frac{1 - \delta_2}{1 - \delta_1\delta_2} \tag{7.1}$$

$$\mu_2 = \frac{1 - \delta_1}{1 - \delta_1\delta_2} \tag{7.2}$$

The share player 2 gets from a bargaining agreement is $(1 - x_1^*)$. If any player takes his outside option, $x_1^* = w_1$ and $x_2^* = w_2$. Player 1 and 2' discount factors are δ_1 and δ_2 respectively.

The computational complexity of COO bargaining problem is higher than that of CRub82 bargaining problem which is analyzed in Section 3.2. This is because that in COO bargaining problem one player has one more choice of action at a bargaining time: taking his outside option.

7.2 Assumptions of Players' Boundedly Rationality

The set of assumptions on players' bounded rationality in Section 3.3 is fundamental for players in COO bargaining problem as well.

Assumptions **A-2** to **A-9** continue to apply to COO bargaining players in the CCGP system. Both players in COO bargaining problem, have three options to respond an offer or a counter-offer: accept, reject and quit. So assumption **A-1** is changed to: player i's goal is to maximize his utility. Provided with three action choices, a player chooses the action which brings him the largest utility. Player i's three possible choices of action are: α, β and γ. $u(\alpha)$, $u(\beta)$ and $u(\gamma)$ are i's utilities of α, β and γ respectively. Player i takes the action which rewards him the highest utility, $MAX(u(\alpha), u(\beta), u(\gamma))$. If two or three actions return the same utility, player i takes the action which ends the bargaining soonest. This modification on **A-1** is an extension to the original idea of **A-1** in Section 3.3.

7.3 Constraints in Outside Option Bargaining Problem

The constraints of CRub82 bargaining problem are defined in Section 5.3. In COO bargaining problem, the properties of constraints change: even a perfectly rational economic man having a larger discount factor can get a smaller portion of a cake than a player having a lower discount factor, because outside option(s) might have a reverse influence on the bargaining power. Therefore the constraint properties for COO bargaining problem change to:

The hard constraint **C 1** $x_i \in (0, 1]$ is still valid. Any offer, counter-offer or a share of an agreement should not be larger than the size of the cake and should not be negative.

The soft constraints **C 2** and **C 3** are still valid, but their effects on the bargaining outcomes become unclear without substantial game-theoretic knowledge and analysis.

7.4 System and Experiment Design - Adaptation of CCGP

The CCGP system successfully applies to CRub82 bargaining problem and to the four incomplete information bargaining problems. Although the mathematical proofs of game-theoretic solutions for CRub82, for ICRub85 incomplete information bargaining problems and for COO outside option bargaining problem are different from each other, the CCGP system established in Chapter 3 is designed to be extensible and modifiable for various bargaining problems. We modify it to satisfy the specifications of the COO outside option bargaining problem at the following three aspects:

- *Bargaining Scenario*: the existence of an outside option provides an alternative choice for a player when he makes decisions. The bargaining procedure in CCGP needs to be consequently changed to allow a player to secede after refusing an offer and then to take up his outside option;
- *GP Terminal Set*: outside options are one of the two determinants on bargaining outcomes. The values of outside options need to be included into the terminal sets of genetic programs of both populations;
- *Utility Function*: a player's utility function becomes $x_i\delta_i^t$ if an agreement is settled at t or $w_i\delta_i^t$ if player i takes up his outside option w_i at t.

In the coming subsections, these three specifications are treated and CCGP is modified accordingly.

7.4.1 Bargaining Procedure

As in Section 3.5, a genetic program g_i's corresponding time-dependent bidding function is defined as $b(g_i) = g_i \times (1 - r_i)^t$, where r_i is the discount rate, $\delta_i \equiv exp(-r_i)$. $b(g_i)$ ensures that player i bids decreasing shares over time. The nonnegative integer t is the bargaining time.

Player i's strategy $s(g_i)$ determines what action player i takes at time t to respond the offer or the counter-offer by player j on dividing the cake as $(1 - x_j, x_j)$:

$$s(g_i) = \begin{cases} \textit{Accept} : x_i = 1 - x_j \\ \quad \text{if } (1 - x_j)\delta_i^t \geq MAX(b(g_i)_{(t+1)}\delta_i^{t+1}, w_i\delta_i^t) \\ \\ \textit{Opt-out} \ : x_i = w_i \\ \quad \text{if } w_i\delta_i^t > MAX(b(g_i)_{(t+1)}\delta_i^{t+1}, (1 - x_j)\delta_i^t) \\ \\ \textit{Counteroffer at } (t+1) : \\ \quad \text{if } b(g_i)_{(t+1)}\delta_i^{t+1} > MAX((1 - x_j)\delta_i^t, w_i\delta_i^t) \end{cases} \tag{7.3}$$

$s(g_i)$ expresses a player's decisions in order to maximize his utility. A player chooses the most beneficial one among the three possible options: accepting the current offer immediately, so his utility will be $(1 - x_j)\delta_i^t$; taking his outside option now, so his utility will be $w_i\delta_i^t$; or making a counter-offer after one time interval, so his utility will be $b(g_i)_{(t+1)}\delta_i^{t+1}$ if this counter offer is accepted by player j. If accepting the current offer brings no less utility than the other two options, the player accepts this offer; if taking his outside option returns higher utility than the other two options, the player takes his outside option; if counter-offering after one time interval will probably bring him higher utility than accepting now and taking his outside option now, the player counter-offers.

7.4.2 GP Terminal Sets

In CCGP system, each player has a population of genetic programs. Both players of the COO bargaining problem have complete information about the game, therefore their information set should be $\{\delta_1, \delta_2, w_1, w_2\}$. Added the size of cake 1 and the -1 to change the sign, the terminal set for g_i is therefore $\{\delta_1, \delta_2, w_1, w_2, 1, -1\}$. The function set, values of GP operators and the experimental set-ups keep the same as those defined in Table 3.3 and in Section 3.5.

7.4.3 Fitness Function

g_i's utility from an agreement with g_j in player j's population, or from taking player i's outside option, is notated as $u_{s(g_i)\to s(g_j)}$. If both players agree with a division of the cake as $(x_i, x_j) = (x_i, (1 - x_i))$ at the bargaining time t, g_i gets $x_i \times \delta_i^t$; if one of players decides to take his outside option, g_i gets $w_i \times \delta_i^t$.

$$u_{s(g_i)\to s(g_j)} = \begin{cases} x_i \times \delta_i^t \\ \quad \text{if } \textit{agreement made} \\ w_i \times \delta_i^t \\ \quad \text{if } w_i \textit{ taken} \end{cases} \tag{7.4}$$

The game fitness of g_i, $GF(s(g_i))$ is the average utility that $s(g_i)$ gains from agreements or from taking his outside option after bargaining with genetic programs in the co-evolving population J which is a set of m genetic programs, $j \in J$. Such genetic programs satisfy the hard constraint. The integer m is an experimental parameter.

$$GF(s(g_i)) = \frac{\sum_{j\in J} u_{s(g_i)\to s(g_j)}}{m} \tag{7.5}$$

In theory if players perpetually disagree and do not take their outside options, both players obtain utility 0. In experiments, if players do not agree and do not take their outside options after 10 time intervals, both get utility 0.

According to the analysis of the constraints in COO bargaining problem in Section 7.3, the hard constraint **C 1**: $x_i \in (0, 1]$ must be satisfied. If this constraint is satisfied, an extra value 3 is added on the top of its game fitness. This encourages feasible genetic programs to propagate in the new population. For genetic programs violate this hard constraint, a function $ATT(i) + ATT(j) - e^{\frac{-1}{|g_i|}}$ is applied. The constraint handling technique and the fitness function for COO can directly inherit those of the CCGP for incomplete information bargaining problems in Section 6.5:

For COO bargaining problem, g_i's fitness function $F(g_i)$ is:

$$F(g_i) = \begin{cases} GF(s(g_i)) + 3 \\ \quad \text{if } g_i \in (0, 1] \\ ATT(i) + ATT(j) - e^{\frac{-1}{|g_i|}} \\ \quad \text{if } g_i \notin (0, 1] \end{cases} \tag{7.6}$$

7.4.4 Game Settings

We test 115 game settings, covering all five conditions of the three categories defined in Table 7.1. It is necessary to group experiments according to their conditions. We study experimental results under three categories below. They are divided by the values of outside options and by outside options' relationship to discount factors, see Table 7.1.

1. *Category 1: Ineffective Threats*: the condition $\#a$ in the Table 7.1: the situations where $w_1 \leq \delta_1\mu_1$ and $w_2 \leq \delta_2\mu_2$. The existence of outside option(s) have no impact on bargaining outcomes because these outside options are too small to be considered by both players.

2. *Category 2: Effective Threats*: the conditions $\#b, \#c$ and $\#d$ in Table 7.1. Outside option(s) change players' bargaining powers but both players still prefer a mutually agreeable bargaining outcomes than taking outside options.

3. *Category 3: Over-Strong Threats*: condition $\#e$ in Table 7.1, including the situations where $(w_1 + w_2 > 1) \cap (0 < w_1 < 1) \cap (0 < w_2 < 1)$, and the situations where $(w_1 = 1) \cap (0 < w_2 < 1)$; $(0 < w_1 < 1) \cap (w_2 = 1)$. For at least one player, outside option(s) is better than any possible bargaining outcome.

Values of game settings are chosen as follows: δ_1 and $\delta_2 \in \{0.1, 0.5, 0.9\}$, representing small, middle and large discount factors. $w_1 \in \{0, 0.01, 0.03, 0.1,$
$0.2, 0.4, 0.5, 0.7, 1\}$ and $w_2 \in \{0, 0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.4, 0.5, 0.7, 1\}$. These values are not fully combined.

For a game setting, we execute 100 runs, starting with different random seeds. We record (1) the average experimental shares $\bar{x_1}$ and $\bar{x_2}$ of 100 runs. If $\bar{x_1} + \bar{x_2} = 1$, only $\bar{x_1}$ is reported. For the situations of $\bar{x_1} + \bar{x_2} \neq 1$, as expected for the condition $\#e$, both $\bar{x_1}$ and $\bar{x_2}$ are reported; (2) the average bargaining time $\bar{t}$ in experiments and; (3) the deviations σs of $\bar{x_1}$s and $\bar{x_2}$s, which examine the stationarity of agreements.

7.5 Experimental Results and Observations

Comparing experimental results with game-theoretic solutions, we measure their differences with respect to partition in agreements, bargaining time and stationarity of agreements. In addition, to demonstrate the efficiency of CCGP, the computational resource required is reported.

#	*t Statistical value* *tsv*	*t Critical two-tail* *tcv*
a	0.4592	2.0423
b	0.9112	2.0049
c	0.4020	2.0154
d	0.4704	2.0739
e	-1.6506	1.9769

Table 7.2: t-test results under 5 different conditions for COO outside option bargaining problem. The 95% confidence level applies.

7.5.1 Partition of Cake in Agreement

Experiments are split into three categories with five conditions as in Table 7.1. We make t-tests on the hypothesis $x_1^* - \bar{x_1} = 0$ for these five conditions. The hypotheses under all these five conditions are accepted with 95% confidence. The results of t-tests are in Table 7.2. So there is no statistical evidence showing that x_1^*s are significantly different from $\bar{x_1}$s.

From the linear regression test on x_1^*s and $\bar{x_1}$s, R^2 is 0.9777. In summary, the R^2 is nearly 1, showing a strong linear relationship between x_1^*s and $\bar{x_1}$s. The statistic evidences from t-tests and from the linear regression test demonstrate that $\bar{x_1}$s ideally approximate x_1^*s.

Now we analyze the compound impacts of the two determinants: discount factors and outside options on bargaining outcomes according to experimental results.

1. *Category 1: Ineffective Threats* $w_1 \leq \delta_1\mu_1$ and $w_2 \leq \delta_2\mu_2$: condition #a in Table 7.1. Theoretically the values of these outside options are too small to make any credible threats on the bargaining outcomes at all. Bargaining agreements are better off than players' outside options. Thus no player considers his outside option. Bargaining continues as if there is no outside option. The game-theoretic solution to this condition is the same as the SPE for CRub82 bargaining problem whose outside option are

both 0s. Both SPE for CRub82 and SPE for COO bargaining problem under $\#a$ are $(\mu_1, 1-\mu_1)$. Let return to our bargaining example of University A and economist B. Suppose B's current salary is $\pounds 50,000$ and University A obtains $\pounds 50,000$ from having B working at University A. The size of cake is $\pounds 100,000$. A and B each now has 50% of the cake. If B's alternative job offer provides a salary $\pounds 49,999$ (B' outside option w_B). B's threat to quit in order to get higher salary from A is incredible because University A just ignores this threat. Experimental results support the game-theoretic analysis. The raw experimental data are in Appendix F. For example, Table F.1: #2 $(\delta_1, \delta_2, w_1, w_2) = (0.1, 0.9, 0, 0.7)$ where player 1 has no outside option and player 2' outside option seems to be a threatening one $w_2 = 0.7$. The observed $\bar{x_1}$ is 0.1104 (SPE $x_1^* = 0.1099$). We compare SPE of COO problem with SPE of CRub82 problem in which both outside options are 0, to see whether the existence of a positive outside option makes any difference on bargaining outcomes. For the bargaining situation having no outside option $(\delta_1, \delta_2, w_1, w_2) = (0.1, 0.9, 0, 0)$, its CRub82 SPE x_1^* is 0.1099. It is clear that in the example #2, the bargaining outcome does not change because of the existence of the outside option $w_2 = 0.7$. Compared with the discount factors (δ_1, δ_2), the $w_2 = 0.7$ is too small to increase player 2' bargaining power. Why? Player 1 has a very low discount factor and has no (creditable) outside option. These force him to accept a small slide of cake. He gives up so much that player 2 obtains a higher utility than if player 2 takes her outside option.

2. *Category 2: Effective Threats* The conditions $\#b, \#c$ and $\#d$ in Table 7.1. Theoretically such outside options effectively account for the increase of one player' bargaining power while both players still prefer dividing the cake to taking their outside options.

Players do not take up their outside options and remain at negotiation table, but the presence of outside option(s) influences the partition of cake. "Credible threats and credible promises matter." [Mut00]. Suppose this time, economist B receives a job offer $w_B = £55,000$. B's threat to quit is now credible and he increases his bargaining power due to the value of w_B. University A compromises to the extent that A only needs to increase B's salary to the exact value of w_B in order to keep B working here. University A does not need to give £1 extra more than $w_B = £55,000$. One example in the experimental data in Appendix F is Table F.2: #1 $(\delta_1, \delta_2, w_1, w_2) = (0.1, 0.1, 0, 0.7)$. The observed $\bar{x_1}$ is 0.2663. For the bargaining situation having no outside option but have the same discount factors $(\delta_1, \delta_2, w_1, w_2) = (0.1, 0.1, 0, 0)$, the CRub82 SPE x_1^* is 0.9091 (so $x_2^* = 0.0909$). In this example, the existence of the outside option $w_2 = 0.7$ makes great difference on the bargaining outcome. w_2 significantly increases player 2's bargaining power from obtaining a 0.0909 slide of the cake to getting an offer from player 1 who instead offers 0.7337 to player 2 (in COO SPE $x_2^* = 0.7$), the same value as player 2's outside option 0.7. Player 1's bargaining power decreases accordingly. The only difference on game setting between this example Table F.2 #1 and the previous example Table F.1 #2 is the value of player 2' discount factor: 0.1 and 0.9 respectively. Because player 2 in Table F.1 #2 has a discount factor so high that w_2 makes relatively less impact. Player 2 in Table F.2 #1 has a discount factor so low that w_2 makes more influential impact on the outcome.

3. *Category 3: Over-Strong Threats* $0 < w_1 < 1$, $0 < w_2 < 1$, and $w_1 + w_2 > 1$: condition #*e* in Table 7.1. The game-theoretic analysis expects that at least one of players prefers his outside option to bargaining, so in equilibrium both players take

their outside options. If economist B is offered a job with the salary $w_B = £100,000$, it is intuitive that B will walk away from the negotiation table. So does University A. If increase B's salary equal to his outside option $£100,000$, University A obtains nothing from keeping B, because there is indifferent between to have B working here and to have nobody working for his position at all. This is a situation where there is no mutual benefit of bargaining: the cake disappears. Experimental results on the partition of cake agree with game theoretic solutions. From the experimental data in Appendix F, for example, Table F.5: #2 $(\delta_1, \delta_2, w_1, w_2) = (0.1, 0.1, 0, 1)$, player 2's outside option is the same size as the cake. The observed $\bar{x_1}$ is 0 and $\bar{x_2}$ is 1. Obviously, player 2 just takes his outside option, no point to carry out bargaining.

7.5.2 Bargaining Time and Efficiency of Agreement and Decision

To analyze the resulting bargaining time $\bar{t}$ and stationarity σ in experiments, we regroup experimental results by their game settings under "$\#a, \#b, \#c$ and $\#d$" or "$\#e$". The reason is that in theory, bargaining under conditions "$\#a, \#b, \#c$ and $\#d$" ends up with bargaining agreements. Bargaining under $\#e$ ends up with taking outside options.

In theory, for all conditions $\#a, \#b, \#c, \#d$ and $\#e$, the SPE solutions implies $t = 0$. The bargaining time t^* in COO SPE for all game settings are 0. As shown in the utility function $u_i = x_i \times \delta_i^t$ or $u_i = w_i \times \delta_i^t$, the utility deteriorates exponentially while t increases. When an agreement or a decision of taking outside option is made at $t = 0$, this agreement or decision is the most efficient one. Any $t > 0$ suggests inefficiency because players pay the cost for delays. Experimental results of average bargaining time are stated in Table 7.3. Generally speaking the experimental results $\bar{t}$s are very small. When threats from outside options are over strong, (condition $\#e$), players learn to take their outside option almost

range of $\bar{t}$	Probability of $\bar{t}$s corresponding to the left range under Conditions $\#a, \#b, \#c$ and $\#d$	under $\#e$
$\bar{t} = 0$	47%	67%
$0 < \bar{t} < 0.05$	20%	17%
$0.05 < \bar{t} < 0.10$	16%	8%
$0.10 < \bar{t} < 0.50$	14%	8%
$0.50 < \bar{t} < 4.00$	3%	0%
Average value of $\bar{t}$s	Condition $\#a, \#b, \#c$ and $\#d$	$\#e$
Average $\bar{t} =$	0.1237	0.0229

Table 7.3: Average bargaining time $\bar{t}$s under the five different conditions of COO outside option bargaining problem.

σ	Percentage of tests under Condition $\#a, b, c$ and d	under $\#e$
Average	0.0209	0.0139
Maximum	0.0430	0.0759

Table 7.4: Deviation σ of $\bar{x}_1$s under conditions $\#a, b, c, d$ or $\#e$ of COO outside option bargaining problem.

immediately: the large probability of $\bar{t} = 0$ and the low average $\bar{t}$ value. When players prefer to bargaining they spend slightly longer time to reach agreements. In Chapter 2, we survey related studies in the field of experimental economics on outside option bargaining problems. These studies find that outcomes of human decisions are inefficient in general. Compared with decisions made by human subjects, those by artificial players demonstrate higher efficiency. Inefficiency observed in our experiments is so small that it can be considered mainly as the consequence of the stochastic property of an evolutionary algorithm.

7.5.3 Stationarity of Agreement and Decision

We measure the stationarity of agreements by means of examining the deviation σ of x_1s. Experimental results of σs are shown in Table 7.4. Both the average and maximal values of deviations are very small, showing that no player wants to withdraw from such agreements under the conditions $\#a, \#b, \#c$ and $\#d$ or to take any choice other than outside options under the condition $\#e$.

7.5.4 Computational Resources

The computational resource for tackling COO problem is almost the same as that for tackling CRub82 bargaining problem. Adding another determinant into the bargaining problem does not make it more difficult nor computationally more expensive to be solved by CCGP.

7.6 Concluding Summary

This chapter studies the COO bargaining problem which has outside options and discount factors. Having two determinants, this bargaining problem is complicated and therefor more difficult to be solved by game-theoretic method. We investigate whether the CCGP system is able to generate fairly good solutions for COO bargaining problem. CCGP system is reused with slight modifications to conform with the specification of outside options.

From experimental results, the mutual benefits (the cakes) are partitioned in a way that approximate the Subgame Perfect Equilibrium. The compound effects of discount factors and outside options on bargaining outcomes demonstrated in experiments, support the game-theoretic analysis. The observed average bargaining time is very small, meaning that the agreements in experiments are of nearly perfect efficiency. Additionally, observed players' behaviors in making agreements or decisions show high stationarity.

The experimental results and observations enhance our assertion that the evolutionary algorithm, particularly the CCGP system is capable of finding out nearly perfect solutions within manageable computational resources and reasonable time for a variety of two-player bargaining problems.

Chapter 8

CCGP for Bargaining Problem with Incomplete Information on Outside Options

8.1 Introduction

The previous chapters discover that CCGP provides efficient and stationary solutions for

- complete information bargaining problem having one determinant, discount factors (CRub82);
- bargaining problems with incomplete information on discount factors (ICRub85, UGI, UII and BGI);
- complete information bargaining problem with two determinants: discount factors and outside options (COO).

Among them, the CRub82, ICRub85 and COO bargaining problems have game-theoretic solutions. Experimental results from CCGP approximate their game-theoretic solutions.

This chapter challenges CCGP with a bargaining problem with complete information on discount factors and with incomplete information on outside options. This bargaining problem is abbreviated as ICOO. The challenge of ICOO problem is that firstly its game-theoretic solution is not known yet; secondly it has complete information on one determinant and has incomplete information on another determinant. We attempt to provide such solutions from CCGP system that may inspire game theorists to reason equilibriums in near future.

The next section introduces the ICOO bargaining problem. Assumptions and constraints are specified in Section 8.3 and Section 8.4 respectively. Section 8.5 reuses the CCGP system for ICOO bargaining problem. Section 8.6 analyzes experimental results.

8.2 Bargaining Problem with Incomplete Information on Outside Option - ICOO

We present a bargaining problem with complete information on discount factors and with incomplete information on outside option, abbreviated as ICOO. In ICOO, one player has incomplete information on the another player's outside option, while his own outside option is publicly known.

We set that the uninformed player 1 knows that player 2's outside option is either w_l (large) or w_s (small) [1]. This model is supposedly easy for game-theoretic analysis. Another option of defining an incomplete information on outside option is the situation when one player's information is totally unknown. For this option the complexity of incomplete information is completely out of the control of game-theoretic analysis at the time being. We therefore prefer the simple structure of incomplete information of ICOO. Hopefully, game theorists will be able to solve ICOO problem within one or two years so that game-theoretic equilibriums can soon be compared with CCGP experimental results.

In details, in ICOO bargaining problem, there are two players bargaining over a partition of a cake of size 1. Except player 2's outside option, all other game related information are

[1] The information structure over a probability distribution is similarly designed to the incomplete information structure on discount factor δ as in ICRub85 problem [Rub85].

Player	*Variable*	*Explanation*	*Privacy*
1	δ_1	Player 1's discount factor	Public
2	δ_2	Player 2's discount factor	Public
1	w_1	Player 1's outside option	Public
2	w_2	Player 2's outside option	Private
2	w_l	$= MAX\{w_l, w_s\}$ a possible value of w_2	Public
2	w_s	$= MIN\{w_l, w_s\}$ another possible value of w_2	Public
	ω'_0	The possibility of $w_2 = w_s$ in player 1's initial belief	Public

Table 8.1: Notations of Variables in ICOO bargaining problem

known by both players. Their bargaining costs over time are measured by discount factors δ_1 and δ_2 respectively. Player 1 has his outside option w_1 and the second player 2 has her outside option w_2. Player 1 knows that player 2's outside option is either w_l (a larger outside option) or w_s (a smaller outside option). $w_l > w_s$. The actual $w_2 \in \{w_l, w_s\}$. Player 1 initially believes that player 2's outside option is w_s with ω'_0 possibility. Table 8.1 outlines the game variables and their properties of privacy.

These outside options w_1, w_2, w_l and w_s are static during one bargaining. The bargaining scenario is the same as what we define in Section 2.3.3 for COO bargaining problem. When an offer or a counter-offer x_i is accepted at the time t, player i receives utility $x_i\delta_i^t$ and the other player gets utility $(1 - x_i)\delta_j^t$. If player i opts out the bargaining at the time t and takes his outside option, he receives $w_i\delta_i^t$ and player j gets $w_j\delta_j^t$. In theory if they both perpetually disagree and do not take their outside options, both players obtain 0. In experiments, if players do not make an agreement and do not take their outside options after 10 time intervals, both get utility 0.

The computational complexity of ICOO bargaining problem is higher than that of COO bargaining problem which is analyzed in Section 7.1.1. This is because in ICOO bargaining problem one player has incomplete information on the other's outside option. Therefore the number of possible outcomes increases.

8.3 Assumptions of Players' Boundedly Rationality

The assumptions on players' bounded rationality: **A-2** to **A-9** in Section 3.3, excluding **A-1** and **A-4**, continue to serve for the CCGP system for the ICOO bargaining problem. Due to the complexity of ICOO problem, **A-1** and **A-4** need to update:

One of two players in ICOO does not have complete information about the game. Therefore two players have different information. **A-4** is changed to: the two players in ICOO bargaining problem have the same level of learning ability. Only the terminal sets of the GP set-ups of two populations in CCGP system are not completely the same. The rest of GP set-up of the two populations are the same.

In a bargaining problem with outside options, each player has three options upon an offer or a counter-offer: accept, reject and quit. So the assumption **A-1** is changed to : a player tries to maximize his utility. Player i has three possible choices of action, α, β and γ. $u(\alpha)$, $u(\beta)$ and $u(\gamma)$ are i's utilities of α, β and γ respectively. Player i takes the action which rewards him the highest utility, $MAX(u(\alpha), u(\beta), u(\gamma))$. If two or three actions bring the same utility, player i takes the action which ends the bargaining sooner.

Players' behaviors are still boundedly rational after the updates on **A-1** and **A-4** assumptions.

8.4 Constraints in ICOO Bargaining Problem

The three constraints of CRub82 bargaining problem are defined in Section 5.3. But for ICOO problem, even a perfectly rational economic man having a larger discount factor is likely to get a smaller portion of a cake than a player with lower discount factor. This is because incomplete information and/or outside option(s) might reverse the influence of discount factors on the outcome. Therefore, the constraint properties for ICOO bargaining problem are:

The hard constraint **C 1** $x_i \in (0, 1]$ is still valid, because any offer, any counter-offer or a share of an agreement should not be larger than the size of the cake and should not be negative.

The soft constraints **C 2** and **C 3** are still valid, but their impacts on bargaining outcomes are unclear without substantial game-theoretic analysis.

8.5 System and Experiment Design - Adaptation of CCGP

We reuse the CCGP system which solves the COO bargaining problem in the last chapter. We modify it to meet the requirements of the incomplete information on outside options.

8.5.1 Bargaining Procedure

We know that the bargaining procedure of ICOO is identical to that of COO bargaining problem. Therefore player i's strategy $s(g_i)$ for ICOO problem can be directly copied from the $s(g_i)$ for COO in Section 7.4.

8.5.2 GP Terminal Sets

Player 1's information set is $\{\delta_1, \delta_2, w_1, \omega_0', w_l, w_s\}$. Added the size of cake 1 and -1 to change the sign, the terminal set for g_1 is $\{\delta_1, \delta_2, w_1, w_l, w_s, \omega_0', 1, -1\}$. Player 2 has complete information relevant to bargaining game, therefore she knows all what player 1 knows. Her information set is $\{\delta_1, \delta_2, w_1, 1, -1\}$. The rest of values of GP parameters and operators are the same as those listed in the Table 3.3.

8.5.3 Fitness Function

The ICOO fitness function inherits the fitness function for COO bargaining problem. COO and ICOO bargaining problems share the same utility function $u_{s(g_i)\to s(g_j)}$, game fitness $GF(s(g_i))$, the properties of constraints and the fitness function $F(g_i)$. $u_{s(g_i)\to s(g_j)}$, $GF(s(g_i))$ and $F(g_i)$ are formally defined in Section 7.4.

In terms of game-theoretic knowledge, the ICOO bargaining problem is much more complicated than either the COO bargaining problem which has complete information on outside options or the incomplete information bargaining problems which have no outside option. The increasing complexity does not make it more difficult for CCGP to adapt to. The CCGP version for solving COO problem only needs to update the terminal sets in order to solve the ICOO. In addition, the way to deal with incomplete information is similar to that in CCGP version for solving incomplete information bargaining problems. The reusability of CCGP is a great advantage.

8.6 Experimental Results and Observations

For each game setting, we execute 100 runs. Runs start with different random seeds. The average of player 1's shares x_1s from the agreements by the best-of-generation genetic programs in the 300^{th} generations of 100 runs is notated as $\bar{x_1}$. We test 36 sets of game settings as shown in Appendix G: Tables G.1 and G.2. The properties of efficiency of agreements, stationarity of agreements and computational resources are examined.

8.6.1 Partition of Cake in Agreement

In theory as player 1 has incomplete information about player 2's outside option, he is presumably to have a disadvantage on the partition of the cake.

There is no game-theoretic solution for ICOO problem yet. We treat COO $\bar{x_1}$s as a reference to understand the impacts of the incomplete information of outside options on ICOO's $\bar{x_1}$s. The experimental results $\bar{x_1}$s of ICOO are compared with the experimental results $\bar{x_1}$s of COO under the same $(\delta_1, \delta_2, w1, w2)$. The last two columns of Table G.3 display the ICOO $\bar{x_1}$s and COO $\bar{x_1}$s. t-test on the hypothesis $ICOO\bar{x_1} - COO\bar{x_1} = 0$ can not be rejected with 95% confidence level. A linear regression test on these two compared data sets results in $R^2 = 0.9953$. $\bar{x_1}$ and COO $\bar{x_1}$ display a strong linear correlation.

Readers may ask why we do not directly compare the experimental results of ICOO bargaining problem with the game-theoretic solutions of COO. Such a comparison may not be justifiable because *in theory* ICOO and COO are different games and they have different structures of game settings. Bargaining problems with incomplete information are expected to have multiple equilibriums. It is inferred from the fact that CRub82 bargaining problem has an unique game-theoretic solution SPE and its corresponding incomplete information bargaining problem ICRub85 has multiple equilibriums PBEs. Therefore we only study the relationship between experimental results of COO and those of ICOO. The experimental results stand on the common co-evolutionary framework and CCGP so they are comparable.

	Experimental Results	
	$\bar{t}$	σ
Average	0.02	0.02
Max	0.08	0.06
Min	0.00	0.0064

Table 8.2: Experimental results on the average, maximal and minimal values of $\bar{t}$ and σ for ICOO bargaining problem.

8.6.2 Bargaining Time and Efficiency of Agreement and Decision

Bargaining time t is highly associated with the efficiency of an agreement or a decision (Section 4.2). The most efficient agreements or decisions of taking outside option should be made at time $t = 0$. Experimental results on bargaining time $\bar{t}$s in Tables G.1 and G.2 are very small. Their statistically analytic results are reported in Table 8.2. In Table 8.2, the average of bargaining time $\bar{t}$s from the experimental results of the 36 tested game settings is 0.02 and the maximal value among them is 0.08. It implies that most of agreements or decisions for taking outside options are made at $t = 0$ in experiments. The bargaining cost on time therefore is very small. This demonstrates that most of agreements and decisions in experiments are very efficient.

8.6.3 Stationarity of Agreement and Decision

The stationarity of agreements and decisions is measured by the deviation σ of x_1s from 100 runs of a given game setting. Experimental results of the deviation σs are very small, see Tables G.1 and G.2. From Table 8.2, the average of deviation σs of the 36 tested game settings is as low as 0.02 with their maximal value 0.06. Such statistic results show that most of observed agreements and decisions are very stationary. Moveover, genetic programs in the final populations are evolutionarily stable.

8.6.4 Computational Resources

The computational resource for tackling ICOO problem is almost the same as that for tackling CRub82 bargaining problem. Adding another determinant and incomplete information into the bargaining problem does not make it more difficult nor computationally more expensive to be solved by CCGP.

8.7 Concluding Summary

We only spent a few days to convert the CCGP system for solving COO bargaining problem to the CCGP system for solving ICOO bargaining problem. It takes about two days to execute the experiments of 36 game settings for ICOO bargaining problem.

On the basis of the above analysis on the experimental results, we conclude that CCGP system generates such solutions for ICOO bargaining problem that demonstrate the game-theoretic properties: efficiency and stationarity.

About the partition of cake in agreements, experimental results $\bar{x_1}$s of ICOO and $\bar{x_1}$s of COO are statistically similar. This fact together with ICOO results' high efficiency and high stationarity, suggest that COO SPE should be efficient and stationary to ICOO problem too. This further infers that COO SPE is probably one of game-theoretic solutions for ICCO [2].

We are hoping that game theorists solve the ICOO bargaining problem soon, then our expectation that the partition of cake between two players approximate one of the game-theoretic solutions will be further confirmed.

[2] CRub82's game-theoretic solution SPE is one of multiple equilibriums of ICRub85's game-theoretic solutions PBE: $(V_s, 1 - V_s)$ when $\delta_2 = \delta_s$ and $\omega_0 = 0$.

Chapter 9

Conclusions

9.1 Importance and Motivations

Bargaining is one of fundamental activities in economics, politics and many other social aspects of the society. Bargaining theory represents an idealization of important aspects of bargaining activities [Mut00]. The game-theoretic method mathematically derives perfect solutions for abstract bargaining games [Mut99]: game-theoretic equilibriums as rational choices. Game-theoretic analysis is based on idealized assumptions, such as perfect rationality, and their mathematical consequences. Perfect rationality is seldom observed in humans' decision making. Humans behave boundedly rationally. The use of perfect rationality by game-theoretic method leads to the question of the suitability of game-theoretic methods for more realistic applications. This argument urges researchers to investigate whether game-theoretic solutions are still applicable to boundedly rational players at all when boundedly rational players play game-theoretic games.

Typically, game theory models one or two essential determinant(s) in a bargaining problem, then examines how these one or two determinant(s) make impact on bargaining outcomes, outcomes' efficiency and outcomes' stationarity. It is rare to examine more than three determinants in one game-theoretic model. This is largely due to the high complexity that multiple determinants cause. When bargaining models increase the number of determinants, their complexity quickly goes beyond human's ability of mathematical reasoning.

Another weakness of game-theoretic method is that even for a simple game, game-theoretic method demands substantial human intelligent effort and expertise in order to prove game-theoretic equilibriums. Analytic complexity increases rapidly when more determinants are taken into account. One the other hand, the more determinants involved the more realistic a bargaining model is. The applications of bargaining theory demand the integration of multiple determinants.

The critical analysis of the limitations of game-theoretic method on its assumption of rationality, its capability and its cost of solving complex problems motivates us to consider an alternative method: computational intelligence. In this thesis, the computational intelligence comes from evolutionary algorithms. The applications of evolutionary algorithms in many fields successfully demonstrate that they are able to deal with many learning tasks and optimization problems which are impractically treated by traditional methods, for example mathematical proofs [Koz92, LP02]. Evolutionary algorithms are especially suitable for problems which are non-linear, having large search space (for instance NP hard problems), multi-dimensional and dynamic problems [LP02]. Bargaining problems are dynamic, having a large search space (Section 3.2) and non-linear. Evolutionary algorithms are suitable for solving such problems.

9.2 Innovations

This section summarizes how we overcome these limitations of game-theoretic method and how we solve seven bargaining problems.

Bounded Rationality

Bounded rationality is defined as not perfect rationality. There is no commonly accepted formal definition of bounded rationality. We define a set of assumptions **A-1** to **A-9** in Section 3.3 to model boundedly rational bargaining players. This set of assumptions is under the co-evolutionary learning principles. We equip artificial bargaining players with basic adaptive learning ability, a form of bounded rationality. Such ability is far from the perfect rationality assumed by the game-theoretic method.

Theoretic Framework

We establish the theoretic co-evolutionary framework for two-player bargaining problems. These artificial players learn how to make mutually acceptable agreements quickly from bargaining training experiences. Such training is patterned after simulated evolution.

CCGP system for Seven Bargaining Problems

On the basis of the assumptions and the theoretic framework, we develop Constraint-based Co-evolutionary Genetic Programming system. CCGP adapts the specifications of variant bargaining problems and solve:

- complete information bargaining problem having one determinant, discount factors (CRub82);
- bargaining problems with incomplete information on discount factors (ICRub85, UGI, UII and BGI);
- complete information bargaining problem with two determinants: discount factors and outside options (COO);
- bargaining problem with complete information on discount factors and with incomplete information on outside options (ICOO).

Determinants	*Complete* Information	*Incomplete Information* One-sided	Two-Sided
Discount Factors	CRub82 * (3, 4, 5)	ICRub85 * UII UGI (6)	BGI (6)
Discount Factors + Outside Options	COO * (7)	ICCO (8)	♢

Table 9.1: Seven bargaining problems are studied in this thesis. Problems with * are problems that have game-theoretic solutions. The numbers in the brackets are the numbers of chapters which examine corresponding bargaining problems. CRub82: Rubinstein complete information bargaining problem having one determinant, discount factors. ICRub85: Rubinstein incomplete information bargaining problem having one determinant, discount factors. UII: Unilateral Imprecise Information Bargaining Model. UGI: Unilateral Ignorance Information Bargaining Model. BGI: Bilateral Ignorance Information Bargaining Model. COO: Complete Information Outside Option Bargaining Model. ICOO: Incomplete Information on Outside Option Bargaining Model. Bargaining problems with two-sided uncertainty on outside options (♢) can be done relatively easily, but it does not enhance our conclusions.

We thoroughly examine seven types of bargaining problems as shown in Table 9.1. These problems are in the scope of sequential bargaining procedure, infinite-horizon dynamic, complete or incomplete information and perfect information. They have one or two bargaining determinants: discount factors and outside options.

Relative and Absolute Fitness Functions

We formalize the relative and absolute fitness functions for co-evolution. We instantiate the relative and absolute fitness functions for bargaining problems. Then we investigate their features and the roles that these two types of fitness functions play in co-evolution.

Constraint Handling

We present the constraint handling technique for evolutionary algorithms, namely Incentive method. The Incentive method enables us to deal with hard constraints and soft constraints simultaneously. The seven bargaining problems all have the hard constraint and soft constraints. Chapter 5, Section 6.6, Section 7.5 and Section 8.6 demonstrate the use of the Incentive Method for CRub82, Incomplete information bargaining problems, COO and ICOO bargaining problems respectively.

9.3 Discoveries

The experimental results and our conclusions are subject to the set of assumptions of bounded rationality defined in Section 3.3, the theoretic co-evolutionary framework in Section 3.4 and computational resources in Section 4.2.6.

The experimental results of CCGP are measured by their game-theoretic properties [1]: *efficiency* (how much bargaining costs spent) and *stationarity* (whether players have intentions to withdraw from agreements unilaterally). For those bargaining problems that have game-theoretic solutions available, we also compare and contrast experimental results with their game-theoretic equilibriums.

We have made significant progress towards efficiently solving bargaining problems with evolutionary algorithms. With reference to the objectives set in Section 1.3, we are ready to answer questions arisen in Introduction.

Approximation to Game-theoretic Solutions

We have found that the experimental results generated by CCGP system statistically approximate game-theoretic solutions. This finding suggests that after certain artificial training, boundedly rational players behave surprisingly similar to perfectly rational players especially in relatively simple games.

Q1. *Are game-theoretic solutions of any use for bounded rational players?* **Answer**: Yes. On the ground that the experimental results statistically approximate game-theoretic

[1] The definitions of these properties are in Section 4.2

solutions (Section 4.2 for CRub82; Section 6.6.1 for ICRub85 and Section 7.4 for COO), we conclude that even boundedly rational players behave in ways that resemble what game-theoretic solutions suggest. So game-theoretic solutions apply to certain boundedly rational players.

Game-theoretic Properties

Furthermore, experimental results of the seven bargaining problems exhibit game-theoretic properties (Section 4.2): efficiency and stationarity. In game theory stationarity strictly constrains to the exact game-theoretic solution(s). In experiments (Section 4.2), due to the stochastic nature of evolutionary algorithms, we measure the stationarity by the deviation of the phenotypes of genetic programs in the population at the last generation. Moreover we assume that when the deviation is smaller than 0.05 the agreement made by the best-of-generation genetic programs is stationary.

The observations indicate that evolutionary algorithms could achieve one of major goals that game-theoretic analysis pursues: to discover efficient and stationary strategies. The summary of experimental results are outlined in Table 9.2 corresponding to bargaining problems in Table 9.1.

Q2. *Through adaptive learning, will boundedly rational players make reasonable decisions?* **Answer**: Yes. For the seven bargaining problems studied by evolutionary algorithms, after training, players make efficient and stationary agreements, as seen in Section 4.2 for CRub82, Section 6.6 for ICRub85,UII, UGI and BGI, Section 7.4 for COO and Section 8.4 for ICOO. In experiments, after training player 1's first offer is always accepted by player 2 immediately. Such an agreement is always made at time 0. Both players have no motivation to withdraw from such an agreement.

Determinants	*Complete* Information	*Incomplete Information*	
		One-sided	Two-Sided
Discount Factors	Approximate SPE *	Approximate PBE * Efficient and stationary	Efficient and stationary
Discount Factors + Outside Options	Approximate SPE *	Efficient and stationary	◇

Table 9.2: Overall observations from experimental results on corresponding bargaining problems as in Table 9.1. Problems with * are problems that have game-theoretic solutions.

Q3. *For complicated games whose game-theoretic solutions are unavailable yet, are bounded rational players' decisions efficient and stationary after evolutionary training?* **Answer**: Yes. Experimental results demonstrate high efficiency and high stationarity (Section 6.6 for UII, UGI and BGI; and Section 8.4 for ICOO). Therefore, it is very likely that experimental results approximate (one of) game-theoretic solutions which are unknown.

Computational Efficiency

Q5. *Does CCGP require heavy computational resources and/or long time for solving bargaining problems?* **Answer**: It only takes about an hour to run statistically sufficient experiments for one game setting by a Linux machine with an *athlon*2400 processor. We implement the CCGP system by JAVA version "1.4.2" and use GNU Compiler for the Java Programming Language (GCJ). This Java compiler is developed by Sun.

Reusability

Q4. *Is the CCGP system easily reusable and extensible for various bargaining problems?* **Answer**: Yes. Once the co-evolutionary framework is established and CCGP system is implemented, it typically takes us a few days to extend the CCGP system to satisfy the specification of a variant bargaining problem. As Section 6.5, 7.4 and 8.5 witness, the CCGP for CRub82 bargaining problem adapts to solve Incomplete information bargaining problems (ICRub82, UII, UGI and BGI), outside option bargaining problem (COO), and the bargaining problem with incomplete information on outside option (ICCO).

Comparison with Human-Subject Experiments

Moreover in Section 6.7 we briefly contrast and compare experimental results from CCGP with human entries or human behaviors observed in experimental economics studies [BPSS98]. It is obvious that human behaviors demonstrate less efficiency and less stationarity than the experimental results from CCGP. It is probably because human learning is far more complicated than the artificial adaptive learning that we define in this thesis (Section 3.3 and Section 3.4). Many other factors in human learning affect learning outcomes.

9.4 Contributions

This study overcomes the limitations of game-theoretic method for solving complicated bargaining problems, in terms of assumption, cost and expertise. We have demonstrated that co-evolutionary algorithms can find approximation to game-theoretic equilibriums and/or find solutions with game-theoretic properties. The main contributions of this thesis are as follows:

- **Co-evolutionary Dynamics** We have deepened the understanding of the co-evolutionary algorithm and its dynamics from observing the interactive adaptation of two players simulated by the co-evolutionary algorithm. Moreover the study on co-evolutionary dynamics helps to explain why the values of evolutionary operators (crossover rate, mutation rate and selection pressure) should be smaller than values of operators in evolutionary algorithms [2]. Additionally in a co-evolutionary system relative fitness essentially drives co-evolving populations to adapt to each other. Absolute fitness monitors adaptive improvement, interpreting how co-adaptation reaches.

- **Constraints**, especially when different types of constraints exist in one problem, increase the difficulty of problem-solving. As demonstrated mainly in Chapter 5, the Incentive method outperforms a comparable penalty method and the co-evolutionary system having no constraint handling technique. Incentive method helps heuristic search to allocate search efforts more efficiently. In literature the application of the Incentive method to financial forecasting also exhibits its efficiency and effectiveness [LT99].

- **Simplicity and Complexity** The assumptions **A-1** to **A-9** defined in Section 3.3 are very primitive. One of the main abilities of a boundedly rational player is to choose the better or the best choice in front of him according to the known values (utilities) of these choices. Another main ability is the adaptive learning ability according to principles of evolution. The discoveries show that players equipped with such primitive abilities perform nearly perfect after evolutionary training. Simple intelligence can generate complex behaviors.

- **Learning Element** The simulations of boundedly rational players' adaptive learning reveal an interesting property of learning elements. Although the learning elements (the terminal set and the function set in Table 3.3 [3]) that we provide to the artificial players are very primitive: arithmetic functions and variables, such players' behaviors

[2] Full details about the values of evolutionary operators are in Section 3.5.
[3] Please note that there are variations in terminal sets for different bargaining problems, as the information one player has may be different from what the other player has.

after sufficient trial-and-error training approximate game-theoretic solutions and exhibit game-theoretic properties. It implies that combinations of simple elements can achieve complex structures and behaviors. This finding is consistent with observations in biology. The building blocks of DNA are called nucleotides. Nucleotides have only four types: adenine, cytosine, guanine and thymine. Merely these four types of nucleotides build genetic materials for millions of species on earth. These species vary greatly in genotypes [4] and phenotypes [5]. John Holland in *Emergence : from chaos to order* states that "complexity emerged from simple elements" [Hol00].

- **Rationality** The learning mechanism in the theoretic framework is based on the principles of genetic algorithms in Machine learning [Mit97]. Simplification is also taken into consideration. As observed, at earlier beginning of learning, players' behave naively. Then they quickly improve their performance. At the end of training, they demonstrate nearly perfect rationality. Just as the saying "practice makes perfect", adaptive learning makes players' behaviors nearly perfect. It indicates that the bounded rationality is not an unbeatable obstacle to reach perfect rationality if sufficient training is given. Importantly, bounded rationality can evolve to be perfect. This study sheds insights to rationality and its relationship to adaptive learning.

- **Complementary Method** CCGP system discovers good solutions for complex and dynamic games. We argue that the co-evolutionary framework, together with CCGP

[4] "The genotype is the specific genetic makeup (the specific genome) of an individual, in the form of DNA. Together with the environmental variation that influences the individual, it codes for the phenotype of that individual." [Wik06c]

[5] "The phenotype of an individual organism is either its total physical appearance and constitution or a specific manifestation of a trait, such as size, eye color, or behavior that varies between individuals. Phenotype is determined to some extent by genotype, or by the identity of the alleles that an individual carries at one or more positions on the chromosomes. Many phenotypes are determined by multiple genes and influenced by environmental factors. Thus, the identity of one or a few known alleles does not always enable prediction of the phenotype." [Wik06d]

system can serve as an alternative method to game-theoretic method for bargaining theory.

- **For Complex Bargaining Problems** Real-life bargaining situations typically have a large number of determinants that affect outcomes. Therefore, it is always difficult to predict bargaining outcomes and find equilibriums. Artificial simulations may help release such analytic burden with affordable time and cost.

9.5 Discussions

Despite achievements, we are aware of limitations of evolutionary algorithms. Firstly, there lack mathematical proofs on experimental results from the evolutionary-algorithms-based CCGP. A CCGP experiment is not designed to "mathematically solve" realistic bargaining problems. Instead, CCGP is expected to generate hints of possible outcomes and to recommend beneficial strategies.

Secondly, experimental results that evolutionary algorithms create are not always identical when inputting different random seeds. Therefore, it is necessary to collect statistically sufficient samples. For these concerns, Koza [Koz92] comments that genetic programming [6] disobeys seven ruling principles for science and engineering. These seven principles are *correctness*, *consistency*, *justifiability*, *certainty*, *orderliness* , *parsimony* and *decisiveness*. However, he argues that from a practical perspective, genetic programming has successfully solved a wide variety of problems and that this is supported by empirical evidences. We agree with his arguments on these concerns.

Thirdly, it may be difficult for CCGP to handle all determinants in a real situation. However, CCGP is capable of integrating more determinants into one bargaining model

[6] This remark also applies to GA.

than game-theoretic method.

These features of evolutionary algorithms and CCGP are paid off by evolutionary algorithms' and CCGP's great advantages: they can model more bargaining determinants than game-theoretic method. Additionally, the quality of solutions, the efficiency on resources and CCGP's reusability are attractive points for considering using such an artificial intelligence.

9.6 Future Study

Having summarized the main achievements and analyzed the limitations, we now outline improvements that merit further investigation.

- To extend CCGP to treat multi-players bargaining. Currently we focus on two-player bargaining games. Two-player models are fundamental ones. Real situations motive us to investigate how bargaining players behave where multiple players are involving. [YD94] extends Axelrod's two-player Iterated Prisoner's Dilemma (IPD) to multi-player IPD. [YD94] discovers such interesting findings that the increasing number of players in IPD causes less cooperation. We are looking forward to observing the behaviors of multiple bargaining players.
- To study bargaining situations where two players have different leaning abilities. In this thesis, we assume that two players equip with identical or almost the same intelligence, see Assumption **A-4** in Section 3.3. That co-evolving players (species) have differentially adaptive abilities is very common in society (nature). Some players quickly learn and promptly adapt to changes of environments, but others do not. That is why chess players might qualify different levels even after the same level of intensive training. This extension will definitely help to enhance our understanding of co-adaptation in a

more general set-up.

- To contrast and compare experimental results of CCGP with observations by experimental economics and to examine whether the beliefs, information states and preferences affect human's decisions in the same way as they affect artificial players' behaviors in simulations. They are great challenges. Such studies demand cooperation of social scientists (experimental economists and psychologists) and researchers of artificial intelligence.

We believe that this study will be selected as genetic materials for creating many new research in future.

THE END

Bibliography

[Aus04] Lawrence M. Ausubel. An efficient ascending-bid auction for multiple objects. *American Economic Review*, 94(5):1452–1475, 2004.

[Axe87] Robert Axelrod. The evolution of strategies in the iterated prisoner's dilemma. In L. Davis, editor, *Genetic Algorithms in Simulated Annealing*, pages 32–41. Pitman, London, 1987.

[BBM93a] David Beasley, David R. Bull, and Ralph R. Martin. An overview of genetic algorithms: Part 1, fundamentals. *University Computing*, 15(2):58–69, 1993.

[BBM93b] David Beasley, David R. Bull, and Ralph R. Martin. An overview of genetic algorithms: Part 2, research topics. *University Computing*, 15(4):170–181, 1993.

[BCT92] Jerome H. Barkow, Leda Cosmides, and John Tooby. *The Adapted mind : evolutionary psychology and the generation of culture*. New York : Oxford University Press, 1995, c1992.

[BF98] H. Bierman and L. Fernandes. *Game Theory with Economic Applications*. Addison-Wesley, 1998.

[Bin85] Ken Binmore. *Game-theoretic models of bargaining*, chapter Bargaining and coalitions, pages 269–304. Cambridge University Press, 1985.

[BLCF04] Kenneth N. Brown, James Little, Paidi J. Creed, and Eugene. C. Freuder. Adversarial constraint satisfaction by game-tree search. In *ECAI 2004*, pages 151–155, Valencia, Spain, 2004.

[BPSS98] Ken Binmore, Chris Proulx, Larry Samuelson, and Joe Swierzbinski. Hard bargains and lost opportunities. *Economic Journal*, 108(450):1279–98, 1998.

[BPSS02] Darse Billings, Lourdes Peña, Jonathan Schaeffer, and Duane Szafron. The challenge of poker. *Artificial Intelligence*, 134(1-2):201–240, 2002.

[BSS89] Ken Binmore, Avner Shaked, and John Sutton. An outside option experiment. *The Quarterly Journal of Economics*, 104(4):753–70, 1989.

[CEM01] B. G. W. Craenen, A. E. Eiben, and E. Marchiori. How to handle constraints with evolutionary algorithms. In Lance Chambers, editor, *The Practical Handbook of Genetic Algorithms, Applications, 2nd ed.*, pages 341–361. Chapman & Hall/CRC Press, Boca Raton, FL, 2001. Chapter 10.

[CEvH03] B. G. W. Craenen, A. E. Eiben, and Jano I. van Hemert. Comparing evolutionary algorithms on binary constraint satisfaction problems. *IEEE Trans. Evolutionary Computation*, 7(5), 2003.

[CM96] Dave Cliff and Geoffrey F. Miller. Co-evolution of pursuit and evasion II: Simulation methods and results. In Pattie Maes, Maja J. Mataric, Jean-Arcady Meyer, Jordan B. Pollack, and Stewart W. Wilson, editors, *From animals to animats 4*, pages 506–515, Cambridge, MA, 1996. MIT Press.

[Coe99] Carlos A. Coello Coello. A survey constraint handling techniques used evolutionary algorithms. Technical Report Technical Report Lania-RI-9904, Laboratorio Nacional de Informtica Avanzada, 1999.

[Coe02] Carlos A. Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191(11–12):1245–1287, 2002.

[CR03a] Christiane Clemens and Thomas Riechmann. Discrete public goods: Contribution levels and learning as outcomes of an evolutionary game. In *Royal Economic Society Annual Conference 2003*, 2003.

[CR03b] Carlos A. Coello Coello and Margarita Reyes Sierra. A Coevolutionary Multi-Objective Evolutionary Algorithm. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 1, pages 482–489, Canberra, Australia, December 2003. IEEE Press.

[Cra85] Nichael Lynn Cramer. A representation for the adaptive generation of simple sequential programs. In John J. Grefenstette, editor, *Proceedings of an International Conference on Genetic Algorithms and the Applications*. Carnegie Mellon University, 1985.

[Das05] Dipankar Dasgupta. *Artificial Immune Systems*, chapter Artificial Immune Systems. University of Memphis, 2005.

[Daw76] Richard Dawkins. *The selfish gene*. Oxford : Oxford University Press, 1976.

[Dor98] Jim Doran. Social simulation, agents and artificial societies. In *ICMAS*, 1998.

[dP04] Edwin D. de Jong and Jordan B. Pollack. Ideal Evaluation from Coevolution. *Evolutionary Computation*, 12(2):159–192, Summer 2004.

[DY94] Paul J. Darwen and Xin Yao. On evolving robust strategies for iterated prisoner's dilemma. In *Progress in Evolutionary Computation, Lecture Notes in Artificial Intelligence*, 1994.

[Eym01] T. Eymann. Co-evolution of bargaining strategies in a decentralized multi-agent system. In *AAAI Fall 2001 Symposium on Negotiation Methods for Autonomous Cooperative Systems*, 2001.

[FAF98] G Fogel, P Andrews, and D Fogel. On the instability of evolutionary stable strategies in small populations. *Ecological Modelling*, 109, 1998.

[FFA97] D Fogel, G Fogel, and P Andrews. On the instability of evolutionary stable strategies. *BioSystems*, 44, 1997.

[FLT85] D. Fudenbery, D. Levine, and J. Tirole. *Game-theoretic models of bargaining*, chapter Infinite-horizon models of bargaining with one-sided incomplete information. Cambridge University Press, 1985.

[Fog99] Lawrence J. Fogel. *Intelligence through simulated evolution : forty years of evolutionary programming*. New York : Wiley, c1999, 1999.

[FT83] Drew Fudenberg and Jean Tirole. Sequential bargaining with incomplete information. *Review of Economic Studies*, 50(2):221–47, 1983.

[FWJ05] S Fatima, M Wooldridge, and N Jennings. Bargaining with incomplete information. *Annals of Mathematics and Artificial Intelligence*, 44(3):207–232, 2005.

[GBDJ54] R. Fulkerson G. B. Dantzig and S. M. Johnson. Solution of a large-scale traveling salesman problem. *Operations Research*, 2, 1954.

[Gib92] Robert Gibbons. *A primer in game theory.* London : Harvester Wheatsheaf, 1992.

[GJT05] Tim Gosling, Nanlin Jin, and Edward Tsang. Population based incremental learning with guided mutation versus genetic algorithms: iterated prisoners dilemma. In D. Corne et al., editor, *Proceedings of the 2005 Congress on Evolutionary Computation (IEEE CEC)*, volume 1, pages 958–965, Edinburgh, UK, 2005. IEEE.

[Gol89] David E. Goldberg. *Genetic Algorithms in Search Optimization and Machine Learning.* Addison-Wesley, 1989.

[GP85] S Grossman and M Perry. Sequential bargaining under asymmetric information. Technical report, National Bureau of Economic Research, 1985.

[GS93] Dhananjay K Gode and Shyam Sunder. Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of Political Economy*, 101(1):119–137, 1993.

[GT06] Tim Gosling and Edward Tsang. Tackling the simple supply chain model. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, 2006.

[GTG01] Gerd Gigerenzer, Peter M. Todd, and ABC Research Group. *Simple heuristics that make us smart.* Oxford : Oxford University Press, 2001.

[GTJ06] Tim Gosling, Edward Tsang, and Nanlin Jin. *Handbook of Research on Nature Inspired Computing for Economy and Management*, chapter Game, Supply Chains and Automatic Strategy Discovery Using Evolutionary Computation. Idea Group, Inc., 2006.

[Har62] John C. Harsanyi. Bargaining in ignorance of the opponents' utility function. *Journal of Conflict Resolution*, 6(1):29–38, 1962.

[Her70] R. J. Herrnstein. On the law of effect. *Journal of the Experimental Analysis of Behavior*, 13:243–266, 1970.

[Hil90] D. W. Hillis. Co-evolving parasites improve simulated evolution in an optimization procedure. *Physica D*, 42:228–234, 1990.

[HME97] Robert Hinterding, Zbigniew Michalewicz, and A.E. Eiben. Adaptation in evolutionary computation: A survey. In *IEEECEP: Proceedings of The IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, 1997.

[Hol62] J. H. Holland. Outline for a logical theory of adaptive systems. *J. ACM*, 9:297–314, 1962.

[Hol75] John H. Holland. *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975.

[Hol92] John H. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* Cambridge, Mass. : MIT Press, 2 edition, 1992.

[Hol00] John H. Holland. *Emergence : from chaos to order.* Oxford : Oxford University Press, 2000.

[Jin] Nanlin Jin. Indirect co-evolution for understanding belief in an incomplete information dynamic game. In Mike Cattolico, editor, *Genetic and Evolutionary Computation Conference, GECCO 2006, Proceedings, Seattle, Washington, USA, July 8-12*.

[Jin05] Nanlin Jin. Equilibrium selection by co-evolution for bargaining problems under incomplete information about time preferences. In David Corne et al., editor, *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, volume 3, pages 2661–2668, Edinburgh, UK, 2005.

[JT05a] Nanlin Jin and Edward Tsang. Co-evolutionary strategies for an alternating-offer bargaining problem. In Graham Kendall and Simon Lucas, editors, *IEEE 2005 Symposium on Computational Intelligence and Games CIG'05*, pages 211–217, 2005.

[JT05b] Nanlin Jin and Edward P. K. Tsang. Relative fitness and absolute fitness for co-evolutionary systems. In Maarten Keijzer, Andrea Tettamanzi, Pierre Collet, Jano I. van Hemert, and Marco Tomassini, editors, *Proceedings of the 8th European Conference on Genetic Programming*, volume 3447 of *Lecture Notes in Computer Science*, Lausanne, Switzerland, 2005. Springer.

[JT06] Nanlin Jin and Edward Tsang. Co-adaptive strategies for sequential bargaining problems with discount factors and outside options. In *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, 2006.

[JTng] Nanlin Jin and Edward Tsang. *Humans, Automatons and Markets: Computational Microstructure Design*, chapter Co-evolutionary Adaptive Learning of Bargaining Strategies. Cambridge University Press, Forthcoming.

[KIAK99] John R. Koza, Forrest H Bennett III, David Andre, and Martin A. Keane. *Genetic Programming III: Darwinian Invention and Problem Solving*. MIT Press, Cambridge, MA, USA, 1999.

[KKS+03] John R. Koza, Martin A. Keane, Matthew J. Streeter, William Mydlowec, Jessen Yu, and Guido Lanza. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. MIT Press, Cambridge, MA, USA, 2003.

[KM93] Lawrence M Kahn and J Keith Munighan. A general experiment on bargaining in demand games with outside options. *American Economic Review*, 83(5):1260–80, 1993.

[Koz90] John R. Koza. A genetic approach to econometric modeling. In *Sixth World Congress of the Econometric Society*, Barcelona, Spain, 1990.

[Koz92] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA, 1992.

[Koz94] John R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge, MA, USA, 1994.

[KR95] John H. Kagel and Alvin E. Roth. *The handbook of experimental economics*. Princeton, N.J. : Princeton University Press, 1995.

[KU99] Maciej Komosinski and Szymon Ulatowski. Framsticks: Towards a simulation of a nature-like world, creatures and evolution. In *ECAL*, pages 261–265, 1999.

[Lan99] Richard N. Langlois. The coevolution of technology and organization in the transition to the factory system. In Paul L. Robertson, editor, *Authority and Control in Modern Industry*, pages 45–72. London: Routledge, 1999.

[LD60] A. H. Land and A. G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28, 1960.

[LFC03] J Little, E C Freuder, and P J Creed. Game based csps. In *Proc. 2nd Intl Workshop on Multiparadigm Constraint Programming Languages at the Eighth International Conference on Principles and Practice of Constraint Programming*, 2003.

[LK06] Simon Lucas and Graham Kendall. Evolutionary computation and games. *IEEE Computational Intelligence Magazine*, 1(1):10–18, February 2006.

[LP91] Gunar E. Liepins and W. D. Potter. A genetic algorithm aproach to multiple-fault diagnosis. In Lawrence Davis, editor, *Handbook of Genetic Algorithms*, pages 237–250. Van Nostrand Reinhold, New York, 1991.

[LP02] W. B. Langdon and Riccardo Poli. *Foundations of Genetic Programming*. Springer-Verlag, 2002.

[LS97] Sean Luke and Lee Spector. A comparison of crossover and mutation in genetic programming. In John R. Koza, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max Garzon, Hitoshi Iba, and Rick L. Riolo, editors, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 240–248, Stanford University, CA, USA, 13-16 July 1997. Morgan Kaufmann.

[LT99] Jin Li and Edward P. K. Tsang. Investment decision making using FGP: A case study. In *1999 Congress on Evolutionary Computation*, pages 1253–1259, Piscataway, NJ, 1999. IEEE Service Center.

[LV90] Gunar E. Liepins and Michael D. Vose. Representational issues in genetic optimization. *Journal of Experimental and Theoretical Artificial Intelligence*, 2:101–115, 1990.

[LW03] Sean Luke and R. Paul Wiegand. Guaranteeing coevolutionary objective measures. In Kenneth A. De Jong, Riccardo Poli, and Jonathan E. Rowe, editors, *Foundations of Genetic Algorithms 7*, pages 237–252. Morgan Kaufmann, San Francisco, 2003.

[May82] John Maynard Smith. *Evolution and the theory of games.* Cambridge University Press, 1982.

[Mic95] Zbigniew Michalewicz. A survey of constraint handling techniques in evolutionary computation methods. In *Evolutionary Programming*, pages 135–155. MIT Press, Cambridge, MA, 1995.

[Mil96] J Miller. The co-evolution of automata in the repeated prisoner's dilemma. *Journal of Economic Behavior and Organization*, 29(1), 1996.

[Mit97] Tom M. Mitchell. *Machine learning.* Boston, Mass. : McGraw-Hill, 1997.

[MJ96] Zbigniew Michalewicz and Cezary Z. Janikow. GENOCOP: A genetic algorithm for numerical optimization problems with linear constraints. *Commun. ACM*, 39(12es), 1996.

[Mut99] A. Muthoo. *Bargaining Theory and Applications.* Cambridge University Press, Cambrige UK, first edition, 1999.

[Mut00] Abhinay Muthoo. A non-technical introduction to bargaining theory. *World Economics*, 1(2), 2000.

[Nas50] John Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.

[Nas51] John Nash. Non-cooperative games. *The Annals of Mathematics*, 54(2):286–295, 1951.

[NM44] John Von Newmann and Oskar Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, 1944.

[OR86] Janusz A Ordover and Ariel Rubinstein. A sequential concession game with asymmetric information. *The Quarterly Journal of Economics*, 101(4):879–88, 1986.

[oS94] The Royal Swedish Academy of Sciences. Press release. *The Bank of Sweden Prize in Economic Sciences in Memory of Alfred Nobel*, 1994.

[RPLH89] J. Richardson, M. Palmer, G. Liepins, and M. Hilliard. Some guidelines for genetic algorithms with penalty functions. In J. David Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 191–197. Morgan Kaufmann, 1989.

[Rub82] Ariel Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50(1):97–110, 1982.

[Rub85] Ariel Rubinstein. A bargaining model with incomplete information about time preferences. *Econometrica*, 53(5):1151–72, 1985.

[Sch04] Lothar M. Schmitt. Classification with scaled genetic algorithms in a coevolutionary setting. In *Genetic and Evolutionary Computation Conference (GECCO)*, volume 2, pages 138–149, 2004.

[Sim55] Herbert A. Simon. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69(1):99–118, 1955.

[Sim82] Herbert A. Simon. *Models of bounded rationality.* Cambridge, Mass. : MIT Press, 1982.

[Sim94] Karl Sims. Evolving 3D morphology and behavior by competition. *Artificial Life*, 1(4), 1994.

[Sim97] Herbert A. Simon. *Administrative behavior : a study of decision-making processes in administrative organizations.* New York : Free Press, 4th ed edition, 1997.

[Smi80] S.F. Smith. *A Learning System Based on Genetic Adaptive Algorithms.* Phd dissertation, University of Pittsburgh, 1980.

[TJ06] Edward P. K. Tsang and Nanlin Jin. Incentive method to handle constraints in evolutionary algorithms with a case study. In Pierre Collet, Marco Tomassini, Marc Ebner, Steven Gustafson, and Anikó Ekárt, editors, *Genetic Programming, 9th European Conference, EuroGP 2006, Budapest, Hungary, April 10-12, 2006, Proceedings*, volume 3905 of *Lecture Notes in Computer Science*, pages 133–144. Springer, 2006.

[Tsa92] Edward Tsang. Problem solving with genetic algorithms. In *Science and Engineering Magazine*, number 6, pages 14–17. University of Essex Publication, 1992.

[Tsa93] Edward Tsang. *Foundations of Constraint Satisfaction*. Academic Press, 1993.

[vBGP00] D. D. B. van Bragt, E. H. Gerding, and J. A. La Poutre. Equilibrium selection in alternating-offers bargaining models – the evolutionary computing approach. In *266*, page 25. Centrum voor Wiskunde en Informatica (CWI), ISSN 1386-369X, May 31 2000.

[vBGP02] D.D.B. van Bragt, E.H. Gerding, and J.A. La Poutre. Equilibrium selection in alternating-offers bargaining models - the evolutionary computing approach. *The Electronic Journal of Evolutionary Modeling and Economic Dynamics*, 2002.

[Wei95] J. Weibull. *Evolutionary game theory*. Cambridge, Mass. : MIT Press, 1995.

[Wik05a] Wikipedia. *Wikipedia encyclopedia*, chapter Game Theory. Wikipedia, 2005.

[Wik05b] Wikipedia. *Wikipedia encyclopedia*, chapter Artificial intelligence. Wikipedia, 2005.

[Wik06a] Wikipedia. *Wikipedia encyclopedia*, chapter Rationality. Wikipedia, 2006.

[Wik06b] Wikipedia. *Wikipedia encyclopedia*, chapter Law of effect. Wikipedia, 2006.

[Wik06c] Wikipedia. *Wikipedia encyclopedia*, chapter Genotype. Wikipedia, 2006.

[Wik06d] Wikipedia. *Wikipedia encyclopedia*, chapter Phenotype. Wikipedia, 2006.

[YD94] Xin Yao and Paul J. Darwen. An experimental study of n-person iterated prisoner's dilemma games. In *Evo Workshops*, pages 90–108, 1994.

Appendix A

Notations and Abbreviations

Table A.1: Notations and Explanation

Explanation	Notation
The player who makes the first offer	1
The second player	2
A player	i
The other player	j
Population	P
Individual (Genetic Program)	g
An individual in population for the player i	g_i
g_i's bidding function	$b(g_i)$
Strategy of g_i	$s(g_i)$
Utility	u
g_i's utility from the outcome of bargaining with g_j	$u_{s(g_i)\to s(g_j)}$
Game fitness of g_i	$GF(g_i)$
Fitness of g_i	$F(g_i)$
Discount factor of i	δ_i
Discount rate of i	r_i
Discount factor of the weak player 2	δ_w
Discount factor of the strong player 2	δ_s
Incorrect discount factor of player 2	δ_2'
Player 2 whose discount factor is δ_w	2_w
Player 2 whose discount factor is δ_s	2_s
Player 2 whose discount factor is δ_2'	$2'$
The possibility of player 1' initial belief that player 2's discount factor is δ_w	ω_0
Outside option of player i	w_i
The larger one of two possible player 2's outside options	w_l
The smaller one of two possible player 2's outside options	w_s
The incorrect value of player 2's outside option	w_2'
The possibility of player 1' initial belief that player 2's outside option is w_s	ω_0'
Random value	r
Player i' share from an agreement in the theoretic solution	x_i^*
Player i' share of the agreement that is made by the best-of-generation individuals at last generation of one run	x_i
Player i' average x_is of 100 runs	$\bar{x}_i$
The bargaining time for reaching agreements by the game-theoretic solution	t^*
The bargaining time spent by the best-of-generation individuals at last generation of one run	t
The average bargaining time of ts of 100 runs	$\bar{t}$
Deviation	σ
t-test critical value	tc
t-test statistical value	tsv

Table A.2: Abbreviations and Explanation

Explanation	Abbreviations
Evolutionary Algorithms	EA
Genetic Programming	GP
Genetic Algorithms	GA
Constraint-based Co-evolutionary Genetic Programming	CCGP
Rubinstein Complete Information Bargaining Problem	CRub82
Rubinstein Incomplete Information Bargaining Problem	ICRub85
Unilateral Imprecise Information Bargaining Problem	UII
Unilateral Ignorance Information Bargaining Problem	UGI
Bilateral Ignorance Information Bargaining Problem	BGI
Complete Information Outside Option Bargaining Problem	COO
Incomplete Information on Outside Option Bargaining Problem	ICOO
Subgame Perfect Equilibrium	SPE
Perfect Bayesian Equilibrium	PBE

Appendix B

Non-technical Introduction to Game-theoretic Analysis of CRub82 Bargaining Problem

Rubinstein [Rub82] introduces an alternating-offer bargaining problem and establishes a mathematical model for this problem. We abbreviate this problem as CRub82 bargaining problem. Rubinstein makes game-theoretic assumptions and game-theoretic analysis, and provides subgame perfect equilibrium (SPE).

The purpose of this appendix is to give readers a taste of proof of subgame perfect equilibrium for CRub82 bargaining problem. This appendix mainly follows [Mut99]'s analysis. This appendix is a **non-technical** description of game-theoretic analysis. For rigorous and technical treatments please refer to [Rub82, Mut99, BF98].

Game-theoretic solutions must satisfy at least two properties below:

- **Efficiency**. Efficiency is instantiated as "**no delay**" for CRub82 bargaining problem. A player's equilibrium offer is accepted by the other player. Thus there is no bargaining cost spent, and the cake is solely split by two players.
- **Stationarity**. A player only offers the same division in equilibrium. Players do not unilaterally withdraw from equilibrium.

Let x_1 denote an offer made by player 1 and x_2 by player 2; x_1^* and x_2^* denote the shares of respective players in SPE. The size of the cake is 1. t, a non-negative integer is the time when an agreement is reached.

Consider at an arbitrary time t when player 1 (he) offers x_1 and player 2 (she) gets $1-x_1$ if she accepts this offer. According to Property **Stationarity**, player 1 only asks for x_1 and player 2 only asks for x_2 in equilibrium. In equilibrium x_1 and x_2 are static, not changing over bargaining time. Player 2 accepts x_1 if this offer brings her more utility than what she probably gets after her counteroffer:

$$(1-x_1)\times\delta_2^t > x_2\times\delta_2^{t+1} \tag{B.1}$$

$$1-x_1 > x_2\times\delta_2 \tag{B.2}$$

She rejects x_1 if the utility from x_1 gives her less than what she might get from her counteroffer:

$$(1-x_1)\times\delta_2^t < x_2\times\delta_2^{t+1} \tag{B.3}$$

$$1-x_1 < x_2\times\delta_2 \tag{B.4}$$

If

$$(1-x_1)\times\delta_2^t = x_2\times\delta_2^{t+1} \tag{B.5}$$

$$1-x_1 = x_2\times\delta_2 \tag{B.6}$$

acceptance or rejection of x_1 is indifferent to her. According to Property of **no delay**, she should accept this offer.

Now consider whether player 1 will offer a x_1 so that $1 - x_1 > x_2 \times \delta_2$. He will not do that because he can increase his utility by asking x_1' such that

$$1 - x_1 > 1 - x_1' > x_2 \times \delta_2 \tag{B.7}$$

until $1 - x_1' = x_2 \times \delta_2$. On the other hand, player 1 must offer a x_1 which player 2 does not reject. Therefore, player 1 offers the minimum that player 2 can accept:

$$1 - x_1 = x_2 \times \delta_2 \tag{B.8}$$

Similarly, when player 2 offers, her offer is accepted if

$$1 - x_2 = x_1 \times \delta_1 \tag{B.9}$$

According to Equation B.8 and Equation B.9, we solve the x_1 and x_2:

$$x_1 = \frac{1 - \delta_2}{1 - \delta_1 \delta_2} \tag{B.10}$$

$$x_2 = \frac{1 - \delta_1}{1 - \delta_2 \delta_1} \tag{B.11}$$

As the player 1 is the first-move player, he starts bargaining at time 0 and his offer is accepted by player 2 immediately. Therefore, player 1 gets what he offers and player 2 gets the rest of the cake. The shares in SPE are:

$$x_1^* = x_1 \tag{B.12}$$

$$= \frac{1 - \delta_2}{1 - \delta_1 \delta_2} \tag{B.13}$$

$$x_2^* = 1 - x_1^* \tag{B.14}$$

End of proof.

Appendix C

One-population Co-evolution for CRub82 Bargaining Problem

This appendix presents the study of applying one-population co-evolutionary algorithm to tackle CRub82 bargaining problem. The reasons of considering one-population system are explained. The experimental results of one-population system are analyzed.

Axelrod [Axe87] applies the one-population co-evolutionary algorithms to examine Iterated Prisoners' Dilemma. This design implicitly assumes that two prisoners' actions are symmetric, choosing the same or very similar strategies. After the evolutionary process, the population converges to one strategy and its minor variants. This survived strategy is TIT-FOR-TAT.

We propose an one-population co-evolutionary system for the CRub82 bargaining problem, considering the symmetry of the game-theoretic solution: Subgame Perfect Equilibrium. The proposed one-population co-evolutionary system presumes that players behave the same when they swap their roles. This design also conforms to the fact that two players in CRub82 have the same set of information.

We use genetic programming to implement this one-population co-evolutionary system. The terminal set is $\{\delta_{this}, \delta_{other}, 1, -1\}$. A genetic program is interpreted as a player's behaviors given this player's discount factor δ_{this} and the other's discount factor δ_{other}, so the exact same genetic program can be applied to two players.

We run experiments to measure the performances of the one-population and two-population co-evolutionary systems by inputting same game settings and genetic programming operators. The two-population co-evolutionary system are specified in Section 3.5. Table C.1 lists the experimental results both from the one-population system and from the two-population system. Experimental evidences show that the average partition of cake $\bar{x}_1$s by the two-population system are closer to SPE.

An explanation of this phenomenon is that the two-population system allows individuals in two populations to converge to different genetic programs, thus different strategies. On the other hand, individuals in the one-population system are tightly constrained by functions of δ_{this}, δ_{other}. Two players have to use the same genetic programs. The search aims for a general function suitable for both players with (δ_{this}, δ_{opp}), is much harder than a search for functions suitable for only one player in each population.

The one-population design limits the potential extensions for more complicated situations. The two-population co-evolutionary system is more general, therefore it is preferred for studying problems such as outside option or incomplete information problems in which two players have different information about the bargaining game. In addition, it is more realistic and probably easier to be understood that two bargaining players do not necessarily behave symmetrically.

This appendix has justified the reasons that we finally choose the two-population co-evolutionary system.

Game setting $(\delta_1,\ \delta_2)$	*SPE* x_1^*	*One-Population* $\bar{x_1}$	σ	*Two-population* $\bar{x_1}$	σ
(0.4 , 0.4)	0.7143	0.9588	0.0104	0.8973	0.0247
(0.4 , 0.6)	0.5263	0.4493	0.1429	0.5090	0.0096
(0.4 , 0.9)	0.1563	0.0667	0.0582	0.1469	0.1467
(0.9 , 0.4)	0.9375	0.9684	0.0194	0.9107	0.0106
(0.9 , 0.6)	0.8696	0.8221	0.1143	0.8000	0.1419
(0.9 , 0.9)	0.5263	0.5037	0.0164	0.5385	0.1194
(0.9 , 0.99)	0.0917	0.1918	0.0394	0.1474	0.1023

Table C.1: Experimental results from one-population system and two-population system for CRub82 Bargaining Problem.

Appendix D

GP Terminal Sets including Bargaining Time t for CRub82 Bargaining Problem

This appendix substantiates our choice of using the bidding function $b(g_i) = g_i \times (1 - r_i)^t$ and of representing strategy $s(g_i)$. $b(g_i)$ and $s(g_i)$ are detailed in Section 3.5.

The design of strategy representation of the two-population co-evolutionary system in Section 3.5 excludes bargaining time t from the genetic program's terminal set. This means that t is not a variable to be evolved. This appendix investigates what are the results if we introduce t into the terminal set and we evolve functions which may include t.

In this appendix, the strategy representation for CRub82 bargaining problem is completely generated by the co-evolutionary system. In terms of genetic programming, the primitive set is $\{\ t, \delta_1, \delta_2, 1, -1, +, -, \times, /\ \}$ where t, a non-negative integer, is the bargaining time. A strategy is a syntax tree consisting of the members of the primitive set. We call such strategy representation as T-strategy. The strategy representation in Section 3.5 is named as B-strategy.

Experimental results from using the T-strategy representation is displayed in Table D.1. From a regression statistics analysis, the R^2 value is as low as 0.2402. The SPE is approxi-

Game setting (δ_1, δ_2)	*SPE* x_1^*	*T-strategy* $\bar{x_1}$	σ	*B-strategy* $\bar{x_1}$	σ
(0.1 0.1)	0.9091	0.9169	0.1515	0.8956	0.0308
(0.1 0.4)	0.6250	0.8692	0.1666	0.9101	0.0117
(0.1 0.9)	0.1099	0.6322	0.3620	0.1920	0.0288
(0.4 0.1)	0.9375	0.8929	0.1852	0.9991	0.0054
(0.4 0.6)	0.5263	0.6708	0.3410	0.5090	0.0096
(0.4 0.9)	0.1563	0.5911	0.4346	0.1469	0.1467
(0.5 0.5)	0.6667	0.7762	0.2324	0.6580	0.0271
(0.9 0.1)	0.9890	0.7004	0.3449	0.9792	0.0030
(0.9 0.4)	0.9375	0.4687	0.3604	0.9107	0.0106
(0.9 0.9)	0.5263	0.5458	0.4078	0.5385	0.1194

Table D.1: Experimental results using T-strategy and B-strategy for CRub82 Bargaining Problem

mated by $x_1^* = 0.8050 \times \bar{x_1} + 0.1571$. Experimental results of using the B-strategy generate the regression result $x_1^* = 0.9588 \times \bar{x_1} + 0.0257$ with a remarkably high R^2 value 0.9928. Moreover, the σ from T-strategy is much higher than that of B-strategy in all game settings listed in Table D.1. These sharp distinctions clearly demonstrate that the B-strategy creates much closer results to the Subgame Perfect Equilibrium than the T-strategy does. Spending the same computational costs, the B-strategy representation outperforms the T-strategy representation. The reason of these observations is that t has 11 possible values [1]. Added t into the terminal set, it is more difficult for heuristic search to find out co-adapted strategies.

T-strategy representation is more general than B-strategy representation, but T-strategy has a much larger search space than the B-strategy. T-strategy system has to search sensible strategies which must makes offers and counter-offers within the constraint of (0, 1] for $t = 0$ to $t = 10$. In addition, the B-strategy guarantees that players offers and counter-offers decreasing shares over the bargaining time. Whereas T-strategy makes potentially any possible offers and counter-offers including such strategies that request even larger shares at

[1]bargaining starts at $t = 0$. It can last for 10 time intervals, so t's maximal value is 10.

the end of bargaining than at the beginning, which does not honestly reflect the exponentially increasing time pressure subject to discount factors.

As a conclusion, we finally choose the B-strategy representation in the main thesis.

Appendix E

Experimental Results for Chapter 6

This appendix provides raw experimental data for Chapter 6: *CCGP for Bargaining Problems with Incomplete Information*.

Exp	Game Setting				*PBE*				*Exp. Results*	
No.	δ_1	δ_2	δ_2'	ω_0	x_1^*	V_s	x^{ω_0}	y^{ω_0}	$\bar{x_1}$	σ
# 1	0.1	0.1	0.2	0.1	0.8163	0.8163	0.9009	0.0091	0.9162	0.0316
# 2	0.1	0.1	0.2	0.5	0.8163	0.8163	0.9045	0.0455	0.9269	0.0269
# 3	0.1	0.1	0.2	0.9	0.9082	0.8163	0.9082	0.0818	0.9367	0.0242
# 4	0.1	0.1	0.9	0.1	0.1099	0.1099	0.9009	0.0091	0.9273	0.0277
# 5	0.1	0.1	0.9	0.5	0.9045	0.1099	0.9045	0.0455	0.9335	0.0253
# 6	0.1	0.1	0.9	0.9	0.9082	0.1099	0.9082	0.0818	0.9387	0.0250
# 7	0.1	0.5	0.4	0.1	0.5263	0.5263	0.6024	0.0061	0.6273	0.0180
# 8	0.1	0.5	0.4	0.5	0.5263	0.5263	0.6123	0.0308	0.6329	0.0196
# 9	0.1	0.5	0.4	0.9	0.0561	0.5263	0.6224	0.0561	0.6356	0.0199
# 10	0.1	0.5	0.9	0.1	0.1099	0.1099	0.5025	0.0051	0.6318	0.0185
# 11	0.1	0.5	0.9	0.5	0.5129	0.1099	0.5129	0.0258	0.6362	0.0178
# 12	0.1	0.5	0.9	0.9	0.5236	0.1099	0.5236	0.0472	0.6318	0.0196
# 13	0.1	0.9	0.1	0.1	0.1099	0.1099	0.9009	0.0091	0.1486	0.0311
# 14	0.1	0.9	0.1	0.5	0.0455	0.1099	0.9045	0.0455	0.1507	0.0312
# 15	0.1	0.9	0.1	0.9	0.0818	0.1099	0.9082	0.0818	0.1492	0.0314
# 16	0.1	0.9	0.8	0.1	0.1099	0.1099	0.2016	0.0020	0.1488	0.0315
# 17	0.1	0.9	0.8	0.5	0.1099	0.1099	0.2084	0.0105	0.1572	0.0285
# 18	0.1	0.9	0.8	0.9	0.0194	0.1099	0.2155	0.0194	0.1543	0.0271

Table E.1: The experimental results and its corresponding PBE solutions. $\bar{x_1}$ is the average of player 1's shares from agreements of 100 runs. $\delta_1 = 0.1$.

Exp	Game Setting				*PBE*				*Exp. Results*	
No.	δ_1	δ_2	δ_2'	ω_0	x_1^*	V_s	x^{ω_0}	y^{ω_0}	$\bar{x}_1$	σ
# 19	0.5	0.1	0.2	0.1	0.8889	0.8889	0.9058	0.0584	0.9304	0.0312
# 20	0.5	0.1	0.2	0.5	0.8889	0.8889	0.9265	0.2647	0.9378	0.0250
# 21	0.5	0.1	0.2	0.9	0.8889	0.8889	0.9435	0.4355	0.9373	0.0244
# 22	0.5	0.1	0.9	0.1	0.1818	0.1818	0.9058	0.0584	0.9391	0.0254
# 23	0.5	0.1	0.9	0.5	0.9265	0.1818	0.9265	0.2647	0.9391	0.0233
# 24	0.5	0.1	0.9	0.9	0.9435	0.1818	0.9435	0.4355	0.9445	0.0246
# 25	0.5	0.5	0.4	0.1	0.6667	0.6667	0.6159	0.0397	0.6309	0.0194
# 26	0.5	0.5	0.4	0.5	0.6667	0.6667	0.6774	0.1935	0.6307	0.0207
# 27	0.5	0.5	0.4	0.9	0.3396	0.6667	0.7358	0.3396	0.6406	0.0182
# 28	0.5	0.5	0.9	0.1	0.1818	0.1818	0.5167	0.0333	0.6354	0.0172
# 29	0.5	0.5	0.9	0.5	0.5833	0.1818	0.5833	0.1667	0.6356	0.0184
# 30	0.5	0.5	0.9	0.9	0.6500	0.1818	0.6500	0.3000	0.6397	0.0183
# 31	0.5	0.9	0.1	0.1	0.1818	0.1818	0.9058	0.0584	0.1072	0.0191
# 32	0.5	0.9	0.1	0.5	0.2647	0.1818	0.9265	0.2647	0.1088	0.0195
# 33	0.5	0.9	0.1	0.9	0.4355	0.1818	0.9435	0.4355	0.1106	0.0256
# 34	0.5	0.9	0.8	0.1	0.1818	0.1818	0.2109	0.0136	0.1111	0.0206
# 35	0.5	0.9	0.8	0.5	0.1818	0.1818	0.2593	0.0741	0.1097	0.0233
# 36	0.5	0.9	0.8	0.9	0.1463	0.1818	0.3171	0.1463	0.1113	0.0196

Table E.2: The experimental results and its corresponding PBE solutions. $\bar{x}_1$ is the average of player 1's shares from agreements of 100 runs. $\delta_1 = 0.5$.

Exp	Game Setting				*PBE*				*Exp. Results*	
No.	δ_1	δ_2	δ'_2	ω_0	x_1^*	V_s	x^{ω_0}	y^{ω_0}	$\bar{x_1}$	σ
# 37	0.9	0.1	0.2	0.1	0.9756	0.9756	0.9309	0.3092	0.9269	0.0290
# 38	0.9	0.1	0.2	0.5	0.9756	0.9756	0.9736	0.7364	0.9400	0.0243
# 39	0.9	0.1	0.2	0.9	0.9756	0.9756	0.9870	0.8699	0.9415	0.0280
# 40	0.9	0.1	0.9	0.1	0.5263	0.5263	0.9309	0.3092	0.9293	0.0322
# 41	0.9	0.1	0.9	0.5	0.9736	0.5263	0.9736	0.7364	0.9443	0.0233
# 42	0.9	0.1	0.9	0.9	0.9870	0.5263	0.9870	0.8699	0.9446	0.0238
# 43	0.9	0.5	0.4	0.1	0.9091	0.9091	0.6919	0.2298	0.7151	0.0413
# 44	0.9	0.5	0.4	0.5	0.9091	0.9091	0.8602	0.6506	0.7287	0.0407
# 45	0.9	0.5	0.4	0.9	0.9091	0.9091	0.9267	0.8168	0.7225	0.0416
# 46	0.9	0.5	0.9	0.1	0.5263	0.5263	0.5996	0.1991	0.7338	0.0382
# 47	0.9	0.5	0.9	0.5	0.8041	0.5263	0.8041	0.6081	0.7272	0.0358
# 48	0.9	0.5	0.9	0.9	0.8940	0.5263	0.8940	0.7879	0.7304	0.0397
# 49	0.9	0.9	0.1	0.1	0.5263	0.5263	0.9309	0.3092	0.4708	0.0273
# 50	0.9	0.9	0.1	0.5	0.7364	0.5263	0.9736	0.7364	0.4779	0.0309
# 51	0.9	0.9	0.1	0.9	0.8699	0.5263	0.9870	0.8699	0.4817	0.0295
# 52	0.9	0.9	0.8	0.1	0.5263	0.5263	0.2724	0.0905	0.4817	0.0301
# 53	0.9	0.9	0.8	0.5	0.5263	0.5263	0.5064	0.3830	0.4864	0.0313
# 54	0.9	0.9	0.8	0.9	0.5978	0.5263	0.6782	0.5978	0.4780	0.0278

Table E.3: The experimental results and its corresponding PBE solutions. $\bar{x_1}$ is the average of player 1's shares from agreements of 100 runs. $\delta_1 = 0.9$.

Exp No.	Game Setting δ_1	δ_2	δ_2'	ω_0	*PBE* t^*	*Experimental* $\bar{t}$
# 1	0.1	0.1	0.2	0.1	0	0.00
# 2	0.1	0.1	0.2	0.5	0	0.00
# 3	0.1	0.1	0.2	0.9	0	0.00
# 4	0.1	0.1	0.9	0.1	0	0.00
# 5	0.1	0.1	0.9	0.5	0	0.00
# 6	0.1	0.1	0.9	0.9	0	0.00
# 7	0.1	0.5	0.4	0.1	0	0.04
# 8	0.1	0.5	0.4	0.5	0	0.05
# 9	0.1	0.5	0.4	0.9	1	0.05
# 10	0.1	0.5	0.9	0.1	0	0.05
# 11	0.1	0.5	0.9	0.5	0	0.05
# 12	0.1	0.5	0.9	0.9	0	0.05
# 13	0.1	0.9	0.1	0.1	0	0.34
# 14	0.1	0.9	0.1	0.5	1	0.40
# 15	0.1	0.9	0.1	0.9	1	0.48
# 16	0.1	0.9	0.8	0.1	0	0.41
# 17	0.1	0.9	0.8	0.5	0	0.37
# 18	0.1	0.9	0.8	0.9	1	0.41

Table E.4: PBE bargaining time t^* and experimental bargaining time $\bar{t}$. $\bar{t}$ is the average of the bargaining time for reaching agreements of 100 runs.

Exp No.	Game Setting				*PBE*	*Experimental*
	δ_1	δ_2	δ'_2	ω_0	t^*	$\bar{t}$
# 19	0.5	0.1	0.2	0.1	0	0.00
# 20	0.5	0.1	0.2	0.5	0	0.00
# 21	0.5	0.1	0.2	0.9	0	0.00
# 22	0.5	0.1	0.9	0.1	0	0.00
# 23	0.5	0.1	0.9	0.5	0	0.00
# 24	0.5	0.1	0.9	0.9	0	0.00
# 25	0.5	0.5	0.4	0.1	0	0.05
# 26	0.5	0.5	0.4	0.5	0	0.05
# 27	0.5	0.5	0.4	0.9	1	0.05
# 28	0.5	0.5	0.9	0.1	0	0.05
# 29	0.5	0.5	0.9	0.5	0	0.05
# 30	0.5	0.5	0.9	0.9	0	0.05
# 31	0.5	0.9	0.1	0.1	0	0.20
# 32	0.5	0.9	0.1	0.5	1	0.23
# 33	0.5	0.9	0.1	0.9	1	0.23
# 34	0.5	0.9	0.8	0.1	0	0.22
# 35	0.5	0.9	0.8	0.5	0	0.29
# 36	0.5	0.9	0.8	0.9	1	0.23

Table E.5: PBE bargaining time t^* and experimental bargaining time $\bar{t}$. $\bar{t}$ is the average of the bargaining time for reaching agreements of 100 runs.

Exp No.	Game Setting δ_1	δ_2	δ_2'	ω_0	*PBE* t^*	*Experimental* $\bar{t}$
# 37	0.9	0.1	0.2	0.1	0	0.00
# 38	0.9	0.1	0.2	0.5	0	0.00
# 39	0.9	0.1	0.2	0.9	0	0.00
# 40	0.9	0.1	0.9	0.1	0	0.00
# 41	0.9	0.1	0.9	0.5	0	0.00
# 42	0.9	0.1	0.9	0.9	0	0.00
# 43	0.9	0.5	0.4	0.1	0	0.32
# 44	0.9	0.5	0.4	0.5	0	0.26
# 45	0.9	0.5	0.4	0.9	0	0.32
# 46	0.9	0.5	0.9	0.1	0	0.28
# 47	0.9	0.5	0.9	0.5	0	0.34
# 48	0.9	0.5	0.9	0.9	0	0.33
# 49	0.9	0.9	0.1	0.1	0	3.99
# 50	0.9	0.9	0.1	0.5	1	3.64
# 51	0.9	0.9	0.1	0.9	1	3.98
# 52	0.9	0.9	0.8	0.1	0	3.76
# 53	0.9	0.9	0.8	0.5	0	3.73
# 54	0.9	0.9	0.8	0.9	1	3.83

Table E.6: PBE bargaining time t^* and experimental bargaining time $\bar{t}$. $\bar{t}$ is the average of the bargaining time for reaching agreements of 100 runs.

Exp No.	Game Setting δ_1	δ_2	*CRub82* x_1^*	*Experimental Results* $\bar{x_1}$	σ	$\bar{t}$
# 1	0.1	0.1	0.9091	0.9438	0.0254	0.00
# 2	0.1	0.4	0.6250	0.8476	0.0293	0.03
# 3	0.1	0.9	0.1099	0.1474	0.0371	0.22
# 4	0.4	0.1	0.9375	0.9456	0.0238	0.00
# 5	0.4	0.6	0.5263	0.4981	0.0063	0.02
# 6	0.4	0.9	0.1563	0.0999	0.0051	0.01
# 7	0.5	0.5	0.6667	0.6765	0.0083	0.01
# 8	0.9	0.1	0.9890	0.9836	0.0130	0.00
# 9	0.9	0.4	0.9375	0.8944	0.0128	0.00
# 10	0.9	0.6	0.8696	0.7144	0.0571	0.37
# 11	0.9	0.8	0.7143	0.6178	0.0581	1.91
# 12	0.9	0.9	0.5263	0.4965	0.0503	3.82

Table E.7: Experimental Results for UII: shares of player 1 $\bar{x_1}$s, bargaining time t^*s and stationarity σs.

Exp	Game Setting		*CRub82*	*Experimental Results*		
No.	δ_1	δ_2	x_1^*	$\bar{x_1}$	σ	$\bar{t}$
# 1	0.1	0.1	0.9091	0.9536	0.0229	0.00
# 2	0.1	0.4	0.6250	0.8529	0.0276	0.02
# 3	0.1	0.9	0.1099	0.1414	0.0408	0.20
# 4	0.4	0.1	0.9375	0.9546	0.0211	0.00
# 5	0.4	0.6	0.5263	0.4982	0.0066	0.01
# 6	0.4	0.9	0.1563	0.1016	0.0026	0.01
# 7	0.5	0.5	0.6667	0.6763	0.0091	0.01
# 8	0.9	0.1	0.9890	0.9859	0.0119	0.00
# 9	0.9	0.4	0.9375	0.8974	0.0118	0.01
# 10	0.9	0.6	0.8696	0.7275	0.0505	0.46
# 11	0.9	0.8	0.7143	0.6245	0.0641	1.76
# 12	0.9	0.9	0.5263	0.4901	0.0432	3.66

Table E.8: Experimental Results for UGI: shares of player 1 $\bar{x}_1$s, bargaining time t^*s and stationarity σs.

Exp	Game Setting		*CRub82*	*Experimental Results*		
No.	δ_1	δ_2	x_1^*	$\bar{x_1}$	σ	$\bar{t}$
# 1	0.1	0.1	0.9091	0.9536	0.0229	0.00
# 2	0.1	0.4	0.6250	0.8531	0.0261	0.03
# 3	0.1	0.9	0.1099	0.1306	0.0393	0.23
# 4	0.4	0.1	0.9375	0.9546	0.0211	0.00
# 5	0.4	0.6	0.5263	0.4951	0.0269	0.02
# 6	0.4	0.9	0.1563	0.1033	0.0141	0.03
# 7	0.5	0.5	0.6667	0.6761	0.0088	0.01
# 8	0.9	0.1	0.9890	0.9859	0.0119	0.00
# 9	0.9	0.4	0.9375	0.8971	0.0119	0.01
# 10	0.9	0.6	0.8696	0.7195	0.0514	0.35
# 11	0.9	0.8	0.7143	0.6484	0.0643	1.48
# 12	0.9	0.9	0.5263	0.4930	0.0550	2.96

Table E.9: Experimental Results for BGI: shares of player 1 $\bar{x}_1$s, bargaining time t^*s and stationarity σs.

Appendix F

Experimental Results for Chapter 7

This appendix provides raw experimental data for Chapter 7: *CCGP for Bargaining Problems with Outside Options*.

Exp	Game Setting				*SPE*	*Experimental Results*		
No.	δ_1	δ_2	$w1$	$w2$	x_1^*	$\bar{x_1}$	σ	$\bar{t}$
# 1	0.1	0.5	0	0.2	0.5263	0.6179	0.0275	0.02
# 2	0.1	0.9	0	0.7	0.1099	0.1104	0.0184	0.12
# 3	0.1	0.9	0	0.2	0.1099	0.1350	0.0321	0.20
# 4	0.5	0.9	0	0.7	0.1818	0.0869	0.0159	0.10
# 5	0.5	0.9	0	0.2	0.1818	0.0979	0.0190	0.18
# 6	0.9	0.9	0	0.2	0.5263	0.4326	0.0338	2.06
# 7	0.5	0.1	0.1	0	0.9474	0.9311	0.0250	0.00
# 8	0.5	0.5	0.1	0	0.6667	0.6292	0.0204	0.04
# 9	0.9	0.1	0.1	0	0.9890	0.9336	0.0234	0.00
# 10	0.9	0.1	0.5	0	0.9890	0.9376	0.0235	0.00
# 11	0.9	0.5	0.1	0	0.9091	0.7227	0.0419	0.25
# 12	0.9	0.5	0.5	0	0.9091	0.7129	0.0430	0.26
# 13	0.9	0.9	0.1	0	0.5263	0.4731	0.0281	3.73
# 14	0.1	0.9	0.01	0.02	0.1099	0.1496	0.0372	0.24
# 15	0.5	0.5	0.03	0.03	0.6667	0.6294	0.0218	0.03
# 16	0.9	0.9	0.4	0.4	0.5263	0.4350	0.0098	0.07

Table F.1: Category 1-a Ineffective Threats: The experimental results $\bar{x_1}$ and its corresponding SPE solutions x_1^*, σ and $\bar{t}$. $\bar{x_1}$ is the average of 100 runs for a game setting.

Exp	Game Setting				*SPE*	*Experimental Results*		
No.	δ_1	δ_2	$w1$	$w2$	x_1^*	$\bar{x_1}$	σ	$\bar{t}$
# 1	0.1	0.1	0	0.7	0.3000	0.2663	0.0111	0.00
# 2	0.1	0.1	0	0.2	0.8000	0.7017	0.0328	0.00
# 3	0.1	0.5	0	0.7	0.3000	0.2681	0.0126	0.00
# 4	0.5	0.1	0	0.7	0.3000	0.2676	0.0127	0.00
# 5	0.5	0.1	0	0.2	0.8000	0.7148	0.0243	0.00
# 6	0.5	0.5	0	0.7	0.3000	0.2643	0.0123	0.00
# 7	0.5	0.5	0	0.2	0.6667	0.6194	0.0222	0.02
# 8	0.9	0.1	0	0.7	0.3000	0.2663	0.0126	0.00
# 9	0.9	0.1	0	0.2	0.8000	0.7098	0.0329	0.00
# 10	0.9	0.5	0	0.7	0.3000	0.2687	0.0109	0.00
# 11	0.9	0.5	0	0.2	0.8000	0.6894	0.0340	0.07
# 12	0.9	0.9	0	0.7	0.3000	0.2530	0.0203	0.10
# 13	0.9	0.1	0.2	0.7	0.3000	0.2807	0.0077	0.00
# 14	0.9	0.1	0.7	0.2	0.8000	0.7438	0.0206	0.00
# 15	0.9	0.5	0.2	0.7	0.3000	0.2834	0.0060	0.00
# 16	0.9	0.5	0.7	0.2	0.8000	0.7457	0.0231	0.03
# 17	0.9	0.9	0.2	0.7	0.3000	0.2695	0.0111	0.06
# 18	0.5	0.1	0.1	0.5	0.5000	0.4569	0.0177	0.00
# 19	0.5	0.5	0.1	0.5	0.5000	0.4596	0.0160	0.00
# 20	0.9	0.1	0.1	0.5	0.5000	0.4571	0.0142	0.00
# 21	0.9	0.1	0.5	0.1	0.9000	0.8357	0.0235	0.00
# 22	0.9	0.5	0.1	0.5	0.5000	0.4635	0.0146	0.00
# 23	0.9	0.5	0.5	0.1	0.9000	0.7059	0.0411	0.15
# 24	0.9	0.9	0.1	0.5	0.5000	0.3426	0.0295	0.48
# 25	0.1	0.1	0.01	0.02	0.9091	0.8736	0.0375	0.00
# 26	0.5	0.1	0.4	0.05	0.9474	0.8603	0.0289	0.00
# 27	0.5	0.1	0.4	0.05	0.9474	0.8615	0.0323	0.00
# 28	0.9	0.1	0.4	0.01	0.9890	0.9123	0.0302	0.00

Table F.2: Category 2-b Effective Threats: The experimental results $\bar{x_1}$ and its corresponding SPE solutions x_1^*, σ and $\bar{t}$. $\bar{x_1}$ is the average of 100 runs for a game setting.

Exp	Game Setting				*SPE*	*Experimental Results*		
No.	δ_1	δ_2	$w1$	$w2$	x_1^*	$\bar{x_1}$	σ	$\bar{t}$
# 1	0.1	0.1	0.1	0	0.9100	0.9199	0.0294	0.00
# 2	0.1	0.1	0.5	0	0.9500	0.9349	0.0281	0.00
# 3	0.1	0.5	0.1	0	0.5500	0.6248	0.0243	0.04
# 4	0.1	0.5	0.5	0	0.7500	0.6873	0.0254	0.04
# 5	0.1	0.9	0.1	0	0.1900	0.1782	0.0364	0.20
# 6	0.1	0.9	0.5	0	0.5500	0.5101	0.0165	0.09
# 7	0.5	0.1	0.5	0	0.9500	0.9357	0.0289	0.00
# 8	0.5	0.5	0.5	0	0.7500	0.6886	0.0255	0.05
# 9	0.5	0.9	0.1	0	0.1900	0.1726	0.0070	0.09
# 10	0.5	0.9	0.5	0	0.5500	0.5156	0.0151	0.09
# 11	0.9	0.9	0.5	0	0.5500	0.5196	0.0181	0.18
# 12	0.1	0.9	0.2	0.7	0.2800	0.2563	0.0076	0.04
# 13	0.1	0.9	0.7	0.2	0.7300	0.6703	0.0161	0.00
# 14	0.5	0.9	0.2	0.7	0.2800	0.2574	0.0089	0.04
# 15	0.5	0.9	0.7	0.2	0.7300	0.6763	0.0164	0.01
# 16	0.9	0.9	0.7	0.2	0.7300	0.6880	0.0176	0.09
# 17	0.1	0.5	0.5	0.1	0.7500	0.6867	0.0218	0.03
# 18	0.1	0.9	0.1	0.5	0.1900	0.1704	0.0062	0.08
# 19	0.1	0.9	0.5	0.1	0.5500	0.5139	0.0157	0.08
# 20	0.5	0.5	0.5	0.1	0.7500	0.6832	0.0245	0.03
# 21	0.5	0.9	0.1	0.5	0.1900	0.1716	0.0094	0.08
# 22	0.5	0.9	0.5	0.1	0.5500	0.5171	0.0145	0.08
# 23	0.9	0.9	0.5	0.1	0.5500	0.5200	0.0199	0.18

Table F.3: Category 2-c Effective Threats: The experimental results $\bar{x_1}$ and its corresponding SPE solutions x_1^*, σ and $\bar{t}$. $\bar{x_1}$ is the average of 100 runs for a game setting.

Exp	Game Setting				*SPE*	*Experimental Results*		
No.	δ_1	δ_2	$w1$	$w2$	x_1^*	$\bar{x_1}$	σ	$\bar{t}$
# 1	0.1	0.1	0.2	0.7	0.3000	0.2789	0.0073	0.00
# 2	0.1	0.5	0.2	0.7	0.3000	0.2816	0.0072	0.00
# 3	0.1	0.5	0.7	0.2	0.8000	0.7420	0.0191	0.02
# 4	0.5	0.1	0.2	0.7	0.3000	0.2823	0.0067	0.00
# 5	0.5	0.1	0.7	0.2	0.8000	0.7382	0.0199	0.00
# 6	0.5	0.5	0.2	0.7	0.3000	0.2822	0.0068	0.00
# 7	0.5	0.5	0.7	0.2	0.8000	0.7391	0.0220	0.03
# 8	0.1	0.1	0.1	0.5	0.5000	0.4535	0.0176	0.00
# 9	0.1	0.1	0.5	0.1	0.9000	0.8219	0.0290	0.00
# 10	0.1	0.5	0.1	0.5	0.5000	0.4607	0.0166	0.00
# 11	0.5	0.1	0.5	0.1	0.9000	0.8207	0.0283	0.00
# 12	0.1	0.1	0.7	0.2	0.8000	0.7355	0.0212	0.00

Table F.4: Category 2-d Effective Threats: The experimental results $\bar{x_1}$ and its corresponding SPE solutions x_1^*, σ and $\bar{t}$. $\bar{x_1}$ is the average of 100 runs for a game setting.

Exp	Game Setting				*SPE*	*Experimental Results*		
No.	δ_1	δ_2	$w1$	$w2$	x^*	$\bar{x}$	σ	$\bar{t}$
# 1	0.1	0.1	1	0	1	0.9313	0.0260	0.00
					0	0.0687	0.0260	
# 2	0.1	0.1	0	1	0	0.0000	0.0000	0.00
					1	1.0000	0.0000	
# 3	0.1	0.5	1	0	1	0.8892	0.0451	0.08
					0	0.1108	0.0451	
# 4	0.1	0.5	0	1	0	0.0000	0.0000	0.00
					1	1.0000	0.0000	
# 5	0.1	0.9	1	0	1	0.9046	0.0359	0.08
					0	0.0954	0.0359	
# 6	0.1	0.9	0	1	0	0.0000	0.0000	0.00
					1	1.0000	0.0000	
# 7	0.5	0.1	0	1	0	0.0000	0.0000	0.00
					1	1.0000	0.0000	
# 8	0.5	0.1	1	0	1	0.9310	0.0256	0.00
					0	0.0690	0.0256	
# 9	0.5	0.5	0	1	0	0.0000	0.0000	0.00
					1	1.0000	0.0000	
# 10	0.5	0.5	1	0	1	0.8570	0.0759	0.20
					0	0.1430	0.0759	
# 11	0.5	0.9	0	1	0	0.0000	0.0000	0.00
					1	1.0000	0.0000	
# 12	0.5	0.9	1	0	1	0.9348	0.0293	0.09
					0	0.0652	0.0293	
# 13	0.9	0.1	0	1	0	0.0000	0.0000	0.00
					1	1.0000	0.0000	
# 14	0.9	0.1	1	0	1	0.9326	0.0266	0.00
					0	0.0674	0.0266	
# 15	0.9	0.5	0	1	0	0.0000	0.0000	0.00
					1	1.0000	0.0000	
# 16	0.9	0.5	1	0	1	0.9385	0.0283	0.26
					0	0.0615	0.0283	
# 17	0.9	0.9	0	1	0	0.0000	0.0000	0.00
					1	1.0000	0.0000	
# 18	0.9	0.9	1	0	1	0.9399	0.0272	0.11
					0	0.0601	0.0272	

Table F.5: Category 3-e Over strong threats: The experimental results $\bar{x}$ and its corresponding SPE solutions x^*, σ and $\bar{t}$. $\bar{x}$ is the average of 100 runs for a game setting. In the cells under SPE x^* and $\bar{x}$, the above values are x_1^* and $\bar{x}_1$ respectively. The below values are x_2^* and $\bar{x}_2$ respectively. In theory, players take their outside options immediately so $x_1 + x_2 \neq 1$.

Exp	Game Setting				*SPE*	*Experimental Results*		
No.	δ_1	δ_2	$w1$	$w2$	x^*	$\bar{x}$	σ	$\bar{t}$
# 19	0.1	0.1	0.7	0.5	0.7	0.6559	0.0194	0.01
					0.5	0.52	0.0125	
# 20	0.1	0.1	0.5	0.7	0.5	0.4715	0.0122	0.01
					0.7	0.7069	0.0094	
# 21	0.1	0.5	0.7	0.5	0.7	0.6665	0.0154	0.05
					0.5	0.5196	0.0100	
# 22	0.1	0.5	0.5	0.7	0.5	0.4781	0.0097	0.02
					0.7	0.7085	0.0059	
# 23	0.1	0.9	0.7	0.5	0.7	0.6698	0.0139	0.05
					0.5	0.5184	0.0090	
# 24	0.1	0.9	0.5	0.7	0.5	0.4806	0.0088	0.05
					0.7	0.7094	0.0047	
# 25	0.5	0.1	0.7	0.5	0.7	0.6621	0.0155	0.01
					0.5	0.5171	0.0109	
# 26	0.5	0.1	0.5	0.7	0.5	0.4760	0.0112	0
					0.7	0.708	0.0064	
# 27	0.5	0.5	0.7	0.5	0.7	0.6669	0.0145	0.07
					0.5	0.5183	0.0090	
# 28	0.5	0.5	0.5	0.7	0.5	0.4818	0.0104	0.04
					0.7	0.7054	0.0065	
# 29	0.5	0.9	0.7	0.5	0.7	0.6704	0.0156	0.04
					0.5	0.5166	0.0099	
# 30	0.5	0.9	0.5	0.7	0.5	0.4837	0.0104	0.04
					0.7	0.7078	0.0056	
# 31	0.9	0.1	0.7	0.5	0.7	0.6605	0.0162	0.01
					0.5	0.5168	0.0113	
# 32	0.9	0.1	0.5	0.7	0.5	0.4775	0.0092	0.01
					0.7	0.7061	0.0081	
# 33	0.9	0.5	0.7	0.5	0.7	0.6748	0.0132	0.06
					0.5	0.5138	0.0083	
# 34	0.9	0.5	0.5	0.7	0.5	0.4864	0.0083	0.04
					0.7	0.704	0.0053	
# 35	0.9	0.9	0.7	0.5	0.7	0.6778	0.0114	0.05
					0.5	0.5129	0.0074	
# 36	0.9	0.9	0.5	0.7	0.5	0.4869	0.0082	0.06
					0.7	0.7064	0.0046	

Table F.6: Category 3-e Over Strong Threats: The experimental results $\bar{x}$ and its corresponding SPE solutions x^*, σ and $\bar{t}$. $\bar{x}$ is the average of 100 runs for a game setting. In the cells under SPE x^* and $\bar{x}$, the above values are x_1^* and $\bar{x}_1$ respectively. The below values are x_2^* and $\bar{x}_2$ respectively. In theory, players take their outside options immediately so $x_1 + x_2 \neq 1$.

Appendix G

Experimental Results for Chapter 8

This appendix provides raw experimental data for Chapter 8: *CCGP for Bargaining Problems with Incomplete Information on Outside Options.*

Exp	Game Setting						*Experimental Results*		
No.	δ_1	δ_2	$w1$	ω_0'	w_2	w_2'	$\bar{x_1}$	σ	$\bar{t}$
# 1	0.1	0.1	0.2	0.5	0.7	0.5	0.2800	0.0071	0.00
# 2	0.1	0.1	0.7	0.5	0.2	0.5	0.7291	0.0203	0.00
# 3	0.1	0.1	0.2	0.5	0.7	0.0	0.2763	0.0078	0.00
# 4	0.1	0.1	0.7	0.5	0.2	1.0	0.6874	0.0532	0.00
# 5	0.1	0.5	0.2	0.5	0.7	0.5	0.2814	0.0069	0.00
# 6	0.1	0.5	0.7	0.5	0.2	0.5	0.7303	0.0295	0.03
# 7	0.1	0.5	0.2	0.5	0.7	0.0	0.2391	0.0365	0.00
# 8	0.1	0.5	0.7	0.5	0.2	1.0	0.7033	0.0562	0.02
# 9	0.1	0.9	0.2	0.5	0.7	0.5	0.2526	0.0078	0.05
# 10	0.1	0.9	0.7	0.5	0.2	0.5	0.6684	0.0147	0.01
# 11	0.1	0.9	0.2	0.5	0.7	0.0	0.2556	0.0069	0.06
# 12	0.1	0.9	0.7	0.5	0.2	1.0	0.6535	0.0306	0.01
# 13	0.5	0.1	0.2	0.5	0.7	0.5	0.2798	0.0081	0.00
# 14	0.5	0.1	0.7	0.5	0.2	0.5	0.7360	0.0249	0.00
# 15	0.5	0.1	0.2	0.5	0.7	0.0	0.2783	0.0081	0.00
# 16	0.5	0.1	0.7	0.5	0.2	1.0	0.7012	0.0485	0.00

Table G.1: The w_2 is the actual value of player 2's outside option and w_2' is another possible value of player 2's outside option in player 1's initial belief. ω_0' is the possibility of player 1's initial belief of $w_2 = MIX(w_2, w_2')$. The experimental results $\bar{x_1}$s and their σs and $\bar{t}$s.

Exp	Game Setting						*Experimental Results*		
No.	δ_1	δ_2	$w1$	ω_0'	w_2	w_2'	$\bar{x_1}$	σ	$\bar{t}$
# 17	0.5	0.5	0.2	0.5	0.7	0.5	0.2811	0.0072	0.00
# 18	0.5	0.5	0.7	0.5	0.2	0.5	0.7238	0.0258	0.03
# 19	0.5	0.5	0.2	0.5	0.7	0.0	0.2465	0.0335	0.00
# 20	0.5	0.5	0.7	0.5	0.2	1.0	0.7060	0.0441	0.02
# 21	0.5	0.9	0.2	0.5	0.7	0.5	0.2588	0.0081	0.04
# 22	0.5	0.9	0.7	0.5	0.2	0.5	0.6699	0.0168	0.01
# 23	0.5	0.9	0.2	0.5	0.7	0.0	0.2594	0.0083	0.04
# 24	0.5	0.9	0.7	0.5	0.2	1.0	0.6602	0.0241	0.01
# 25	0.9	0.1	0.2	0.5	0.7	0.5	0.2826	0.0064	0.00
# 26	0.9	0.1	0.7	0.5	0.2	0.5	0.7372	0.0227	0.00
# 27	0.9	0.1	0.2	0.5	0.7	0.0	0.2789	0.0072	0.00
# 28	0.9	0.1	0.7	0.5	0.2	1.0	0.7110	0.0499	0.00
# 29	0.9	0.5	0.2	0.5	0.7	0.5	0.2832	0.0064	0.00
# 30	0.9	0.5	0.7	0.5	0.2	0.5	0.7312	0.0221	0.03
# 31	0.9	0.5	0.2	0.5	0.7	0.0	0.2385	0.0344	0.00
# 32	0.9	0.5	0.7	0.5	0.2	1.0	0.7070	0.0538	0.03
# 33	0.9	0.9	0.2	0.5	0.7	0.5	0.2551	0.0262	0.08
# 34	0.9	0.9	0.7	0.5	0.2	0.5	0.6823	0.0197	0.08
# 35	0.9	0.9	0.2	0.5	0.7	0.0	0.2378	0.0305	0.07
# 36	0.9	0.9	0.7	0.5	0.2	1.0	0.6650	0.0329	0.08

Table G.2: The w_2 is the actual value of player 2's outside option and w_2' is another possible value of player 2's outside option in player 1's initial belief. ω_0' is the possibility of player 1's initial belief of $w_2 = MIX(w_2, w_2')$. The experimental results $\bar{x_1}$s and their σs and $\bar{t}$s.

Exp No.	Game Setting δ_1	δ_2	$w1$	ω'_0	w_2	w'_2	*ICOO* $\bar{x_1}$	COO $\bar{x_1}$
# 1	0.1	0.1	0.2	0.5	0.7	0.5	0.2800	0.2789
# 2	0.1	0.1	0.7	0.5	0.2	0.5	0.7291	0.7355
# 3	0.1	0.1	0.2	0.5	0.7	0.0	0.2763	0.2789
# 4	0.1	0.1	0.7	0.5	0.2	1.0	0.6874	0.7355
# 5	0.1	0.5	0.2	0.5	0.7	0.5	0.2814	0.2816
# 6	0.1	0.5	0.7	0.5	0.2	0.5	0.7303	0.7420
# 7	0.1	0.5	0.2	0.5	0.7	0.0	0.2391	0.2816
# 8	0.1	0.5	0.7	0.5	0.2	1.0	0.7033	0.7420
# 9	0.1	0.9	0.2	0.5	0.7	0.5	0.2526	0.2563
# 10	0.1	0.9	0.7	0.5	0.2	0.5	0.6684	0.6703
# 11	0.1	0.9	0.2	0.5	0.7	0.0	0.2556	0.2563
# 12	0.1	0.9	0.7	0.5	0.2	1.0	0.6535	0.6703
# 13	0.5	0.1	0.2	0.5	0.7	0.5	0.2798	0.2823
# 14	0.5	0.1	0.7	0.5	0.2	0.5	0.7360	0.7382
# 15	0.5	0.1	0.2	0.5	0.7	0.0	0.2783	0.2823
# 16	0.5	0.1	0.7	0.5	0.2	1.0	0.7012	0.7382
# 17	0.5	0.5	0.2	0.5	0.7	0.5	0.2811	0.2822
# 18	0.5	0.5	0.7	0.5	0.2	0.5	0.7238	0.7391
# 19	0.5	0.5	0.2	0.5	0.7	0.0	0.2465	0.2822
# 20	0.5	0.5	0.7	0.5	0.2	1.0	0.7060	0.7391
# 21	0.5	0.9	0.2	0.5	0.7	0.5	0.2588	0.2574
# 22	0.5	0.9	0.7	0.5	0.2	0.5	0.6699	0.6763
# 23	0.5	0.9	0.2	0.5	0.7	0.0	0.2594	0.2574
# 24	0.5	0.9	0.7	0.5	0.2	1.0	0.6602	0.6763
# 25	0.9	0.1	0.2	0.5	0.7	0.5	0.2826	0.2807
# 26	0.9	0.1	0.7	0.5	0.2	0.5	0.7372	0.7438
# 27	0.9	0.1	0.2	0.5	0.7	0.0	0.2789	0.2807
# 28	0.9	0.1	0.7	0.5	0.2	1.0	0.7110	0.7438
# 29	0.9	0.5	0.2	0.5	0.7	0.5	0.2832	0.2834
# 30	0.9	0.5	0.7	0.5	0.2	0.5	0.7312	0.7457
# 31	0.9	0.5	0.2	0.5	0.7	0.0	0.2385	0.2834
# 32	0.9	0.5	0.7	0.5	0.2	1.0	0.7070	0.7457
# 33	0.9	0.9	0.2	0.5	0.7	0.5	0.2551	0.2695
# 34	0.9	0.9	0.7	0.5	0.2	0.5	0.6823	0.6880
# 35	0.9	0.9	0.2	0.5	0.7	0.0	0.2378	0.2695
# 36	0.9	0.9	0.7	0.5	0.2	1.0	0.6650	0.6880

Table G.3: The experimental results $\bar{x_1}$s of ICCO and experimental results $\bar{x_1}$s of CCO under the same $(\delta_1, \delta_2, w1, w2)$

Appendix H

Evolve Genetic Programs for 25 Game Settings

We design a co-evolutionary system in which genetic programs evolve by playing 25 instances of CRub82 bargaining games. We attempted to find the exact game-theoretic solutions. This appendix reports the experimental results of this design.

H.1 Experimental Design

This set of experiments is designed to evolve genetic programs which adapt to many different game settings. We hope under the evolutionary pressure, the identical genetic program as the game-theoretic solution will be found.

The design of these experiments is illustrated in Figure H.1.

H.2 Experimental Set-up

We use the same computational resource as that for previous experiments in Chapter 3. The GP set-up is the same as in Table 3.3:

- two-population co-evolution;
- one population for one player;

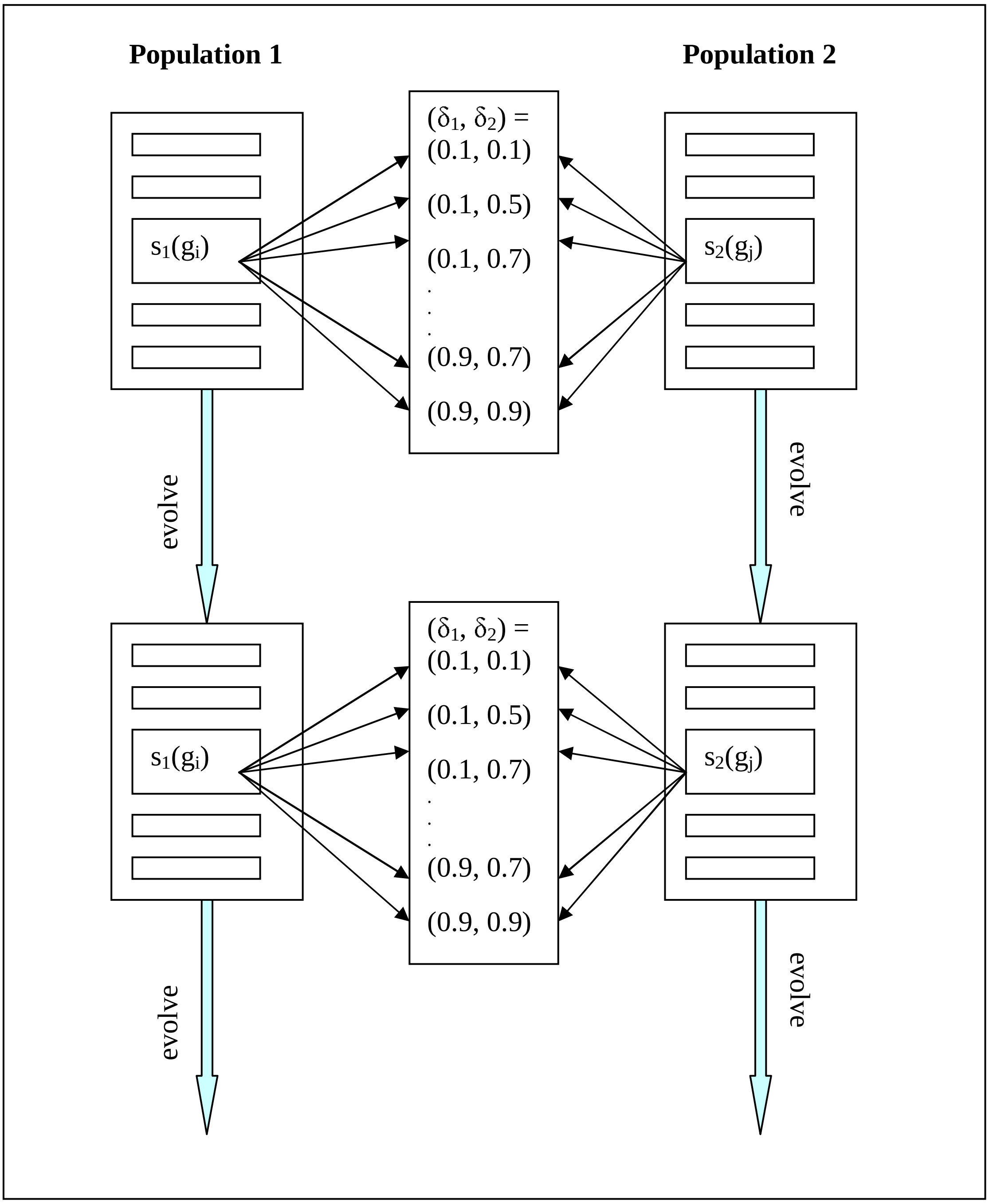


Figure H.1: The design

- each population has 100 individuals (genetic programs);

- The terminal set is $\{\delta_1, \delta_2, 1, -1\}$;

- The functional set is $\{+, -, \times, \div\}$ ($\div$ is Protected).

The difference is the fitness function. The fitness function sums up the game fitness from playing 25 instances of bargaining games. These 25 game settings are listed in Table H.2 and H.1, $\delta_1 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and $\delta_2 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

H.3 Experimental Results

Each run starts with a different random sequence. After 300 generations of evolution, we check the individuals in the final generation of two populations. For each individual (genetic program), we test it with 25 game settings. We give five examples in Table H.1. It lists first offers x_1s of the five genetic programs for the 25 game settings.

Totally 300 individuals in the final generations of the population for player 1, are checked. Only 20 of them satisfy the hard constraint $x_1 \in (0, 1)$.

Totally 300 individuals in the final generations of the population for player 2 are checked. Only 54 of them satisfy the hard constraint $x_2 \in (0, 1)$.

We haven't found the theoretic solutions $x_1^* = \frac{1-\delta_2}{1-\delta_1\delta_2}$ or $x_2^* = \frac{1-\delta_1}{1-\delta_1\delta_2}$ from the total 600 evolved genetic programs yet.

Experimental results from executing the co-evolutionary system (CCGP) with the input of a specific pair of (δ_1, δ_2) are reported in Table 4.2 in the thesis. In order to compare the experimental results, we give another five examples. In Table H.2, genetic programs are the individuals in player 1' final population given (δ_1, δ_2). Its design is detailed in Chapter 3. The highlighted values are the x_1s for the (δ_1, δ_2) to which the genetic programs evolve. For example, $g_1' = \delta_1 - \delta_2 + 1$ is one resulting genetic program by inputting $(\delta_1 = 0.1, \delta_2 = 0.1)$.

Game setting		*SPE*	g_1	g_2	g_3	g_4	g_5
δ_1	δ_2	x_1^*	x_1	x_1	x_1	x_1	x_1
0.1	0.1	0.91	0.10	-0.20	-1.00	0.20	0.01
0.1	0.3	0.72	0.10	-0.40	-0.80	0.60	0.09
0.1	0.5	0.53	0.10	-0.60	-0.60	1.00	0.25
0.1	0.7	0.32	0.10	-0.80	-0.40	1.40	0.49
0.1	0.9	0.11	0.10	-1.00	-0.20	1.80	0.81
0.3	0.1	0.93	0.30	-0.40	-1.20	0.20	0.01
0.3	0.3	0.77	0.30	-0.60	-1.00	0.60	0.09
0.3	0.5	0.59	0.30	-0.80	-0.80	1.00	0.25
0.3	0.7	0.38	0.30	-1.00	-0.60	1.40	0.49
0.3	0.9	0.14	0.30	-1.20	-0.40	1.80	0.81
0.5	0.1	0.95	0.50	-0.60	-1.40	0.20	0.01
0.5	0.3	0.82	0.50	-0.80	-1.20	0.60	0.09
0.5	0.5	0.67	0.50	-1.00	-1.00	1.00	0.25
0.5	0.7	0.46	0.50	-1.20	-0.80	1.40	0.49
0.5	0.9	0.18	0.50	-1.40	-0.60	1.80	0.81
0.7	0.1	0.97	0.70	-0.80	-1.60	0.20	0.01
0.7	0.3	0.89	0.70	-1.00	-1.40	0.60	0.09
0.7	0.5	0.77	0.70	-1.20	-1.20	1.00	0.25
0.7	0.7	0.59	0.70	-1.40	-1.00	1.40	0.49
0.7	0.9	0.27	0.70	-1.60	-0.80	1.80	0.81
0.9	0.1	0.99	0.90	-1.00	-1.80	0.20	0.01
0.9	0.3	0.96	0.90	-1.20	-1.60	0.60	0.09
0.9	0.5	0.91	0.90	-1.40	-1.40	1.00	0.25
0.9	0.7	0.81	0.90	-1.60	-1.20	1.40	0.49
0.9	0.9	0.53	0.90	-1.80	-1.00	1.80	0.81

Table H.1: Experimental results of the co-evolutionary system as designed in Figure H.1. We give five examples here. The values of x_1s of five genetic programs, i.e. g_1, g_2, g_3, g_4, and g_5 are reported.

Its x_1 is 1. The rest values under g_1' are the x_1s given other game settings.

H.4 Discussion

From the above two tables and experiments which we have done, we can not make a definite conclusion that which design generates better results.

The experiments to evolve the game-theoretic solutions probably need more computational resources in order to find out the game-theoretic solutions. It is because that the search for the genetic program(s) which is (are) ideal for 25 game settings is much harder than the search for genetic programs good at a specific pair of (δ_1, δ_2).

Even the game-theoretic genetic program appears during the evolutionary process. It may not be found at the end of evolutionary process, because any crossover or mutation may change it.

Game setting		*SPE*	g'_1	g'_2	g'_3	g'_4	g'_5
δ_1	δ_2	x_1^*	x_1	x_1	x_1	x_1	x_1
0.1	0.1	0.91	**1.00**	1.00	0.91	1.90	1.00
0.1	0.3	0.72	0.80	0.80	0.26	3.70	0.33
0.1	0.5	0.53	0.60	0.60	0.13	5.50	0.20
0.1	0.7	0.32	0.40	0.40	0.08	7.30	0.14
0.1	0.9	0.11	0.20	**0.20**	0.06	9.10	0.11
0.3	0.1	0.93	1.20	1.20	2.73	1.23	3.00
0.3	0.3	0.77	1.00	1.00	0.77	1.70	1.00
0.3	0.5	0.59	0.80	0.80	0.40	2.17	0.60
0.3	0.7	0.38	0.60	0.60	0.25	2.63	0.43
0.3	0.9	0.14	0.40	0.40	0.18	3.10	0.33
0.5	0.1	0.95	1.40	1.40	4.55	1.10	5.00
0.5	0.3	0.82	1.20	1.20	1.28	1.30	1.67
0.5	0.5	0.67	1.00	1.00	**0.67**	1.50	1.00
0.5	0.7	0.46	0.80	0.80	0.42	1.70	0.71
0.5	0.9	0.18	0.60	0.60	0.29	1.90	0.56
0.7	0.1	0.97	1.60	1.60	6.36	1.04	7.00
0.7	0.3	0.89	1.40	1.40	1.79	1.13	2.33
0.7	0.5	0.77	1.20	1.20	0.93	1.21	1.40
0.7	0.7	0.59	1.00	1.00	0.59	1.30	1.00
0.7	0.9	0.27	0.80	0.80	0.41	1.39	0.78
0.9	0.1	0.99	1.80	1.80	8.18	**1.01**	9.00
0.9	0.3	0.96	1.60	1.60	2.31	1.03	3.00
0.9	0.5	0.91	1.40	1.40	1.20	1.06	1.80
0.9	0.7	0.81	1.20	1.20	0.76	1.08	1.29
0.9	0.9	0.53	1.00	1.00	0.53	1.10	**1.00**

Table H.2: Experimental results of the co-evolutionary system as designed in Chapter 3. We give five examples here. The values are x_1s of five genetic programs, i.e. $g'_1 = \delta_1 - \delta_2 + 1$, $g'_2 = 1 - \delta_2 + \delta_1$, $g'_3 = \delta_1/(\delta_2 \times (1 + \delta_2))$, $g'_4 = 1 + \delta_2/\delta_1 - \delta_2$, and $g'_5 = \delta_1/\delta_2$. Those x_1s which are highlighted are experimental results from the co-evolutionary system given the corresponding (δ_1, δ_2).

THE END OF APPENDIX