TWO MINDED PRISONERS:
A SKEPTIC APPROACH TO ROUND ROBIN TOURNAMENTS

by
Orcan Alpar
B.S. in Systems Engineering, Yeditepe University, 2001

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Systems and Control Engineering
Boğaziçi University
2005

ACKNOWLEDGEMENTS

I wish to thank the following individuals for making this thesis possible.

First of all I wish to thank my supervisor Birgül Egeli for her bright advices regarding my thesis and, further, for her comradeship and time that has been spent for me.

I wish to thank Eşref Akpınar for his genius mind and for our incessant brainstorming. I am sure that he should be as happy as I can be for my thesis completion, there won't be any quantum conversation anymore.

I wish to thank my managers Cenk Bıçakçı and Ekrem T. Gençarslan for their support and permission through my master days.

I also wish to thank my general manager, Sir Lucian Arkas for his helpful doctrines and for trusting in me. I hope that I haven't devastated him anytime.

Finally, I wish to thank Nurcan Alkan, the most beautiful and sentient entity that I've ever seen in my life, for her congeniality and understanding while I am working on my project. Also I wish to thank Bülban Alpar, my mom, for her corporeal and incorporeal support to my project, to my life for everlasting sponsorship contract.

ABSTRACT

TWO MINDED PRISONERS: A SKEPTIC APPROACH TO ROUND ROBIN TOURNAMENTS

This thesis mainly focuses on human brain strategies while grading in a round robin tournament and strategies that a grader can have to achieve goals and increase in rankings. In the round robin tournament, all participants grade the rest sequentially and iteratively with purpose of gaining higher grades and rising in rankings.

When investigating round robin tournaments, the main bug to cast the doubt on is graders giving points to the others and receiving points from them. It is obvious that, in human thinking, being a single minded machine is impossible so the graders are certainly subjective. This situation, graders and grades, causes a skeptic approach to round robin tournaments.

The main objective of this project is to rationalize the human thinking when giving a grading decision while the grades are directly correlated to other participants grades. The project includes a random network generation and a grading process in the random network. After grades are generated, the plane is turned into prisoner's dilemma game, and payoff points are calculated. The worst player is chosen and several methods, methods like genetic algorithm, fuzzy logic and quantum inspired genetics, are applied to the current system to lift the loser up in the rankings.

Eventually, these methods are compared to find out which method is better to explain human thinking, when decision making in prisoner's dilemma game, to achieve better results.

ÖZET

ÇİFT ODAKLI MAHKUMLAR: ROUND ROBIN TURNUVALARINA ŞÜPHECİ YAKLAŞIM

Bu çalışma temel olarak Round Robin turnuvalarında insan beyni stratejilerine ve oyuncunun sıralamada yükselip, amaçlarına ulaşması için gerekli stratejilere odaklanmaktadır. Round Robin turnuvalarında, bütün katılımcılar yüksek puan alıp en başta olabilmek için birbirlerine puan vermektedirler.

Round Robin turnuvalarını araştırırken, süphe duyulan esas yanlışın, puan veren oyuncunun puan verdiği oyuncu tarafından da puan almasıdır. İnsan beyninin tek odaklı olması şüphesiz ki mümkün olamayacağından, bütün oyuncular sübjektif kararlar vermektedirler. Bu durum turnuvalara oyunculara ve puanlara şüpheci yaklaşımı beraberinde getirmektedir.

Bu projenin asıl amacı, bağımlı puanlar verirken insanın düşünce sistemini rasyonalize etmektir. Proje, rastgele oyuncu ağı yaratılmasını ve oyuncuların rastgele birbirlerini notlamasını kapsamaktadır. Notlar yaratıldıktan sonra düzlem, mahkum ikilemi düzlemine çevirilip, oyun puanları hesaplanmaktadır. En kötü oyuncu tesbit edilip, bu oyuncunun puanlarını yükseltmek için genetik algoritmaları, bulanık mantık ve kuantum genetik metodları uygulanmaktadır.

Son olarak, bu metodlarla bulunan sonuçlar karşılaştırılıp, insan beyninin mahkum ikilemi oyunundaki çalışma prensiplerini hangi metodun daha iyi açıkladığı bulunmaya çalışılmaktadır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

C	Cooperation
CHO	Choice of Cooperation or Defection as Output
D	Defection
GA	Genetic Algorithms
GF	Gaming Factor (a.k.a. Deviance)
GR	Grade
IPD	Iterated Prisoner's Dilemma Game
MF	Membership Function
OPS	Opponent's Score
OPT	Opponent's Tactic
OWS	Own Score
OWT	Own Tactic
QGA	Quantum inspired Genetic Algorithms
PD	Prisoner's Dilemma Game

1. INTRODUCTION

1.1. Background

There is a dramatic story beneath the origination of this project:

One day, a student takes an elective course in his university. In this course, great majority belongs to another department that has studied together for many years so they are close friends at first sight. In this course, the main duty of the participants is to present a paper. No problem takes place until the presentations have ended and the instructor claims about grading. The request is so simple: Everyone will grade every other one, like a round robin tournament, and will give points from 0 to 5 in 4 categories.

After the grading process finishes, the instructor declares the grades, so the student finds himself at the bottom of the curve of the lesson dramatically.

There must be a hidden problem in the tournament. It is obvious that nobody can be a single minded machine that can grade objectively without thinking other factors, factors like closeness. So this method cannot be used without tuning, because it is anyhow a GAME, "Prisoner's Dilemma Game in a Classroom".

This project will show the tuning methods, to elect the disregarded increase and decrease of points in round robin tournament grading. These methods are social network analysis, fuzzy logic approach, genetic algorithms and quantum inspiration on genetics.

1.2. Definitions

1.2.1. Game Theory

Game theory (Rasmusen, 1989) is a branch of mathematics that presents formal analysis of the interaction among a group of rational players. The players have choices available to them, so as to select particular course of action. They are supposed to behave strategically and are motivated to increase their utilities that depend on the collective course of action.

Modern game theory (Rasmusen, 1989), which started with the work of John von Neumann and Oskar Morgenstern (Neumann and Morgenstern, 1953) in 1930s, offers solution concepts that are relevant to certain types of games. The notions of value of coalition, backwards-induction outcome and sub-game perfect outcome present a few examples. The types of games for which these concepts are appropriate are known to be the cooperative, sequential (with moves made in order) and repeated (with moves made simultaneously in one stage) games, respectively.

Many decision making problems in sociology, politics and economics deal with situations in which the results depend not only on the action of one individual but also on the actions of others. Game theory is used in modeling situations in which many individuals with conflicting interests interact, such that the results depend on the actions of all the participants. It is considered a formal way to analyze interaction among a group of individuals who behave rationally and strategically. The participants in a game strive to maximize their (expected) utilities by choosing particular courses of action. Because the actions of the others matter, a player's final utility depends on the profile of courses of action chosen by all the individuals (Iqbal, 1995).

A game deals with the following concepts:

- *Players.* These are the individuals who compete in the game. A player can be an individual or a set of individuals.
- A *move* will be a player's action.
- A player's (pure*) strategy* will be a rule (or function) that associates a player's move with the information available to player at the time when the player decides which move to choose.
- A player's *mixed strategy* is a probability measure on the player's space of pure strategies.
- *Payoffs* are real numbers representing the players' utilities.

Although first attempts to analyze such problems are apparently rather old (Cournot, 1897), modern game theory started with the work of John von Neumann and Oskar Morgenstern who wrote the book Theory of Games and Economic Behaviour (Neumann and Morgenstern, 1953). Game theory is now widely used in research in diverse areas ranging from economics, social science, to evolutionary biology and population dynamics.

The information at the disposal of a player, when he/she has to select a move, is described by the information structure in the game. Based on this structure, games can usually be put in either one of the following two broad classes, which also form the two main branches of game theory:

Cooperative games: In cooperative games the players are allowed to form binding agreements. These are restrictions on the possible actions decided by two or more players. Binding an agreement usually requires an outside authority that can monitor the agreement at no cost and impose on violators' sanctions so severe that cheating is prevented. For players in a binding agreement there is a strong incentive to work together to receive the largest total payoff. The agreements may include, for example, commitments and threats (Iqbal, 1995).

Non-cooperative games: In non-cooperative games the players may not form binding agreements. Neither do the players cooperate nor do they enter into negotiation for achieving a common course of action. However the players know how the actions, the actions of the other players and their own, will determine the payoffs of every player.

Matrix games:

One way to describe a game is to list the players participating in the game and to list the alternative choices or moves available to each player. In the case of a two-player game, the moves of the first player form the rows, and the moves of the second player from the columns of a matrix. The entries in the matrix are two numbers representing the payoff to the first and second player, respectively. Such a description of a game makes possible to completely represent the players' payoffs by a matrix. In game theory these games are recognized as matrix games (Osborne and Rubinstein, 1994).

Bi-matrix games:

Classes of games that have attracted much attention because of the relative simplicity of their mathematical analysis involve two players Alice and Bob. Games of this kind are called bi-matrix games (Flitney and Abbott, 2002). The most popular bi-matrix game is the so-called the Prisoner's Dilemma (PD).

1.2.2. Prisoner' Dilemma Game

The PD got its name from the following hypothetical situation:

Two criminals were arrested under the suspicion of having committed a crime together. However, the police do not have sufficient proof in order to have them convicted. The two prisoners are isolated from each other, and the police visit each of them and offer a deal: the one who offers evidence against the other one will be freed. If none of them accepts the offer, they are in fact cooperating against the police, and both of them will get only a small punishment because of lack of proof. They both gain. However, if one of them betrays the other one, by confessing to the police, the defector will gain more, since he is freed; the one who remained silent, on the other hand, will receive the full punishment, since he did not help the police, and there is sufficient proof. If both betray, both will be punished, but less severely than if they had refused to talk. The dilemma resides in the fact that each prisoner has a choice between only two options, but cannot make a good decision without knowing what the other one will do (Axelrod, 1984).

Each suspect may choose between two strategies namely confessing (D) and not confessing (C), where C and D stand for cooperation and defection. There are 3 situations and 3 outcomes (Iqbal, 1995):

.

- If neither suspect confesses, i.e. (C, C), they go free, and split the proceeds of their crime which we represent by 3 units of payoff for each suspect.
- However, if one prisoner confesses (D) and the other does not (C), the prisoner who confesses testifies against the other in exchange for going free and gets the entire 5 units of payoff, while the prisoner who did not confess goes to prison and gets nothing.
- If both prisoners confess, i.e. (D, D), then both are given a reduced term, but both are convicted, which we represent by giving each 1 unit of payoff: better than having the other prisoner confess, but not as good as going free.

The whole game situation and its different outcomes can be summarized by Table 1.1 where hypothetical ”points” are given as an example of how the differences in result might be quantified.

Table 1.1 Outcomes of a Standard PD Game

		Player A C	Player A D
Player B	C	(3,3)	(0,5)
Player B	D	(5,0)	(1,1)

Yet the pursuit of individually sensible behavior results in each player getting only 1 unit of payoff, much less than the 3 units each that they would get if neither confessed (C, C). This conflict between the pursuit of individual goals and the common good is at the heart of many theoretical game problems. For PD, the rational choice for both players is to defect (Hofbauer and Sigmund, 1988; Nowak and Sigmund, 1989; Nowak and Sigmund, 1990).

A spatially extended Prisoner's Dilemma was first proposed by Axelrod who concluded that territoriality strongly influences the evolution of cooperation (Axelrod, 1984). Extensive work on the spatial Prisoner's Dilemma started in 1992 when Nowak and May explored a cellular automaton based on this game on regular lattices. They and others found complex spatiotemporal dynamics and emergence of cooperation for strategy spaces confined to the strategies defecting and cooperating (Herz and Theor, 1994; Huberman and Glance, 1993; Nowak, Bonhoeffer and May *et al.*, 1994; Nowak and May, 1992; Nowak and May, 1993).

Iterated Prisoner's Dilemma (IPD):

For a fixed number of iterations, which is known in advance to the players, the dominant strategy is Defect since it maximizes minimal payoff for each player. However, in an iterated prisoner's dilemma where the number of iterations is not initially known to the players, cooperative outcomes are possible (Axelrod, 1984). Software development projects typically involve repeated interactions of the participants during their lifecycle. Due to changes of schedule and other factors, the number of interactions is not known initially. Therefore, the iterated prisoner's dilemma is a plausible model of the development process.

Individual Behaviors in IPD:

In an iterated game cooperative and non-cooperative behavior will typically be reciprocated to a certain extend. The players are influenced in their choice of strategy by the opponent's previous move, if they are able to perceive it. Such reactive strategies in the iterated prisoner's dilemma can be modeled stochastically.

The whole population has individuals to play the game with their own strategies. Examples of these strategies are listed below (Kraines and Kraines, 1993);

- Tit For Tat: Repeat opponent's last choice.
- Tit For Tat and Random: Repeat opponent's last choice skewed by random setting.
- Tit For Two Tats and Random: Like Tit For Tat except that opponent must make the same choice twice in a row before it is reciprocated. Choice is skewed by random setting.
- Tit For Two Tats: Like Tit For Tat except that opponent must make the same choice twice in row before it is reciprocated.
- Naive Prober: Repeat opponent's last choice, but sometimes probe by defecting in lieu of cooperating.
- Remorseful Prober: Repeat opponent's last choice, but sometimes probe by defecting in lieu of cooperating. If the opponent defects in response to probing, show remorse by cooperating once.
- Naive Peace Maker: Repeat opponent's last choice, but sometimes make peace by cooperating in lieu of defecting.
- True Peace Maker: Cooperate unless opponent defects twice in a row, then defect once, but sometimes make peace by cooperating in lieu of defecting.
- Random: Always set at 50% probability.
- Always Defect: Defector
- Always Cooperate: Cooperator
- Grudger: Cooperate until the opponent defects. Then always defect without forgiving.
- Pavlov: If 5 or 3 points scored in the last round then repeat last choice.
- Pavlov/Random: If 5 or 3 points scored in the last round then repeat last choice, but sometimes make random choices.

- Adaptive: Start with c,c,c,c,c,c,d,d,d,d,d and then take choices which have given the best average score re-calculated after every move.
- Gradual: Cooperate until the opponent defects, in such case defect the total number of times the opponent has defected during the game. Followed up by two cooperations.
- Suspicious Tit For Tat: As for Tit For Tat except begin by defecting.
- Soft Grudger: Cooperate until the opponent defects, in such case opponent is punished with d,d,d,d,c,c.

These strategies are often used in Iterated Prisoner's Dilemma Game. The main objective is payoff maximization nevertheless, in experimental tests, people often cooperate to garner a greater payoff indicating either that modeling in game theory is somehow incomplete or that people behave irrationally (Nemoto and Gagen, 2004).

Population Behaviors in IPD:

A large number of population structures occur in Prisoner's Dilemma Game. Generally no single behavior emerges and persists forever, indicating that populations which are stable against mutation and crossover probably don't exist. However, there do appear to be general population structures which can be thought of as metastable in the sense that they emerge frequently and persist for many generations (Smucker, Stanley and Ashlock, 1994).

The populations often make rapid transitions from one of these metastable behaviors to another, sometimes after spending a number of generations with a very low average fitness.

Population behaviors discussed above are listed below:

- Full Nice Cooperation: Everyone likes each other.
- Latching: Individuals like only a few other individuals, but don't mind the others.
- Raquel and the Bobs: Bobs like all Raquels. Raquels like each other and don't mind Bobs.

- Disconnected Stars: Hubs dislike one another. Spokes like hubs. Hubs don't spokes and like the spokes they are connected to in sequence.
- Connected Centers: Nice guys like each other and don't mind the thugs. Thugs like and latch onto a center nice guy.
- Wallflower: Everyone dislikes each other.

1.2.3. Quantum Theory

History of Quantum Theory:

In classical physics, some aspects of nature are described using particles, others using waves. Light, in particular, was described by waves. In 1900, Max Planck showed that the spectrum of blackbody radiation, which could not be explained in terms of waves, could be explained by the assumption that light is emitted in discrete quanta of energy (Schreiber, 1994).

Later, in 1905, Albert Einstein showed that the photoelectric effect, which also did not fit the wave model, could be explained by the assumption that light is always quantized (Schreiber, 1994). This controversial suggestion was not accepted until after the famous experiments of Arthur Holly Compton in 1923 (Compton, 1923).

In 1923, Louis de Broglie suggested that, conversely, 'particles' might display wavelike properties (Broglie, 1924). This was later confirmed most strikingly by the Davisson-Germer experiment of 1927 (Clinton and Germer, 1927).

In 1925, Werner Heisenberg introduced matrix mechanics (Schreiber, 1994). This formalism was able to predict the energy levels of quantum systems. It was cast in the form of a wave function and differential equation by Erwin Schrödinger in the following year (Schreiber, 1994).

The efforts of Paul Dirac (Paul , 1926), Pascual Jordan (Schreiber, 1994) and others at synthesizing the two approaches (the so called transformation theory) led to the more general formalism of quantum mechanics (Neumann, 1955)

These developments marked a turning point in physics. For the first time, physical results were being derived, not from a model of the universe, but from abstract mathematical constructs such as the Schrödinger wave function. There was an unprecedented need for interpretation.

In 1926, Max Born suggested that the values in a particle's wave function gave the probabilities of finding the particle in a given place (Schreiber, 1994). This really set the cat amongst the pigeons. The suggestion that the laws of physics were non-deterministic flew in the face of everything achieved since Newton.

In retrospect, the work of Born precipitated an even more radical revolution: a switch in physics from ontology (discussion of what is) to epistemology (discussion of what is known). According to Born, an experiment to determine the position of a quantum system would give results with probabilities dictated by the wave function. This statement talks about what results are obtained instead of modeling the process. It involves an artificial division of the world into system and observer.

The following interpretational questions were raised:

- Accepting quantum mechanics as an epistemological theory, is there a specific place at which the observer-system cut should be located? In more concrete terms, are there certain systems which may be described by a wave function and others which may not?
- If there is no such fixed observer-system cut, is it consistent to allow this cut to be made at different places according to convenience? In other words, may we always choose how much of a system to describe by a wave function?
- Is there an ontological theory underlying the epistemological formulation of quantum mechanics? In other words, is there a model of the universe from which we can derive the fact that certain systems may be treated using quantum mechanics?

If such a model is proposed, does it explain the non-determinism of quantum mechanics in terms of an underlying deterministic mechanism? In response to these questions, two main schools of thought arose. Bohr and his followers (notably Heisenberg) embraced the epistemological viewpoint and argued that it cannot be possible to find an underlying ontological theory.

Einstein and his followers (notably Schrödinger) felt that a deterministic ontological physics must underlie the non-deterministic epistemological predictions of quantum theory. The states in such underlying theories were dubbed hidden variables. A critical twist came in 1935 when Einstein, Podolsky and Rosen (EPR) showed that the formalism of quantum mechanics implied non-locality: an action at one point may have immediate consequences at a distant point without any apparent intervening mechanism (Einstein, Podolsky and Rosen, 1935). They inferred that quantum mechanics cannot be complete. There must be an underlying theory which is local. This highlighted a division amongst those who accepted quantum mechanics as a fundamental theory.

Bohr himself was a positivist. He insisted on describing things through macroscopic observations only and rejected the idea that the Schrödinger wave function represented the state of a particle. At best, the wave function was a useful way of predicting the macroscopic results of an experiment on the particle. As such, it was essentially physically meaningless to discuss whether the wave function behaved non-locally. John von Neumann and Paul Dirac, on the other hand, were perfectly happy to talk about the state of a particle. However, rather than accepting EPR's argument for an underlying theory, they simply accepted that the world is indeed non-local.

Later, in 1964, the controversy was to take another turn when John Bell showed that it was the predictions of quantum theory and not merely the formalism that implied non-locality (Bell, 1964). Even if an underlying deterministic theory were found, it would not be local!

In the mean time, there had been two important developments in the interpretation of quantum mechanics. In 1952, David Bohm proposed a hidden variables theory (Bohm, 1952). The core of Bohm's idea had earlier been proposed by de Broglie and abandoned (Wheeler and Wojciech, 1983). Although it had some strange features including, of course, non-locality, this was the first real candidate for a deterministic theory underlying quantum mechanics.

In 1957, Hugh Everett III proposed a radical interpretation. He suggested that the Schrödinger wave function describes not one world but an infinite and growing collection of realities (Everett, 1957). When the position of a particle is measured, rather than saying it may turn out to be here or there, Everett suggested that it will be both here and there, in parallel realities. As well as being bizarre, Everett's interpretation was rather vague: What constitutes a measurement for the purposes of causing reality to split? Considerable progress was made on this question through study of the phenomenon of decoherence (Caldeira and Legget, 1983; Feynman and Vernon, 1963; Joos and Zeh, 1985; Zurek, 1981; Zurek, 1982; Zurek, 1986). This study abandoned simplified models of isolated laboratory equipment and started to consider the effects of the environment. This led to a number of "post-Everett" interpretations, some building closely on Everett's ideas, others not, but all a little more sophisticated and a little less vague. Histories have emerged as the favorite formalism for this work.

Qubits:

Quantum bits (qubits) are the elementary units of information that are used to represent quantum data (Schumacher, 1996). Thus, the idea of a qubit underlies all investigations in the rapidly growing science of quantum information – including quantum information theory, quantum communication, quantum computation, quantum complexity, and quantum game theory (Bennett and Vincenzo, 2000; Hey, 1999; Nielsen and Chuang, 2000). In particular, qubits are the basic building blocks for defining the standard model of quantum computation as introduced by Deutsch (Deutsch, 1985), which has so far provided the appropriate representation for identifying and understanding efficient ways of processing information using quantum mechanics. His investigation resulted in feasible algorithms for factoring large integers (Shor, 1994) and for simulating many-particle quantum systems (Lloyd, 1996), two problems not known to be efficiently solvable with classical computers.

A qubit can be thought of as the extension of a classical bit obtained by applying the superposition principle. When quantum superposition states of many qubits are constructed in the tensor product state space that quantum mechanics prescribes for a composite system, quantum entanglement arises as an additional information resource with no classical counterpart. However, qubits share with classical bits the fundamental property of being a fungible information resource (Bennett and Hey, 1999; Toffoli and Hey, 1999): While both classical and quantum information is intended, in fact required, to be physically realized, it is abstractly defined and therefore independent of the details of the underlying physical realization.

The fungibility property is essential to quantum information in two respects. First, by defining quantum information independent of the details of specific physical devices and their complex physics, it has been possible to study qubit properties at the abstract level and thus to obtain a deeper understanding of the distinctive features that qubits inherit from their intrinsic quantum-mechanical nature. Examples of fundamental results following from the basic properties of superposition and of the randomness associated with quantum measurements include the fact that qubits in an unknown quantum state cannot be perfectly copied (no-cloning theorem (Dieks, D., 1982; Wooters and Zurek, 1982)) and they cannot be broadcast (no-broadcasting theorem (Barnum, Caves, Fuchs, Jozsa and Schumacher, 1966)). On the other hand they can be reliably communicated by means of the quantum teleportation protocol (Bennett, Brassard, Crepeau, Jozsa, Peres and Wooters, 1993), an extremely useful protocol with many applications (Gottesman and Chuang, 1999). Second, from a practical standpoint, the arbitrariness of the physical realization implied by fungibility allows for greater flexibility in the identification and design of quantum information processors. This is reflected in the amazing variety of representations of qubits that have appeared in recent proposals for physical realizations of quantum computers (Holland, 1975; Viola, Knill and Laflamme, 2001).

1.2.4. Genetic Algorithms

Genetic Algorithms (GAs) were invented by John Holland in the 1960s (Holland, 1975). Holland's original goal was to formally study the phenomenon of adaptation as it occurs in nature and to develop ways in which the mechanism of natural adaptation might be imported into computer systems. In Holland's work, GAs are presented as an abstraction of biological evolution and a theoretical framework for adaptation under the GA is given. Holland's GA is a method for moving from one population of chromosomes to a new one by using a kind of natural selection together with the genetic-inspired operators of crossover and mutation.

Each chromosome consists of genes (bits in computer representation), each gene being an instance of a particular allele (0 or 1). Traditionally, these crossover and mutations are implemented as follows (Holland, 1975; Mitchell, 1996):

Crossover: Two parent chromosomes are taken to produce two child chromosomes. Both parent chromosomes are split into left and right sub chromosomes. The split position (crossover point) is the same for both parents. Then each child gets the left sub chromosome of one parent and the right sub chromosome of the other parent. For example, if the parent chromosomes are 011 10010 and 100 11110 and the crossover point is between bits 3 and 4 (where bits are numbered from left to right starting at 1), then the children are 011 11110 and 100 10010.

Mutation: When a chromosome is taken for mutation, some genes are randomly chosen to be modified. The corresponding bits are flipped from 0 to 1 or from 1 to 0. These operations reveal the fact that GAs are inherently parallel algorithms. GAs work by discovering the most adapted chromosomes, emphasizing, and recombining their good "building blocks" through operations that can be easily performed in parallel. This has been explored in many works over the GA literature (Han, Park, Lee and Kim, 2001; Theodore, 1995).

Genetic Optimization Algorithms are stochastic search algorithms which are used to search large, non-linear spaces where expert knowledge is lacking or difficult to encode and where traditional optimization techniques fall short (Goldberg, 1989).

To design a standard genetic optimization algorithm, the following elements are needed:

- A method for choosing the initial population;
- A "scaling" function that converts the objective function into a non-negative fitness function;
- A selection function that computes the "target sampling rate" for each individual. The target sampling rate of an individual is the desired expected number of children for that individual;
- A sampling algorithm that uses the target sampling rates to choose which individuals are allowed to reproduce;
- Reproduction operators that produce new individuals from old ones;
- A method for choosing the sequence in which reproduction operators will be applied.

Consequently, alleles are allowed to be real parameters. Thus, some care should be taken to define these operators. Mutations can be implemented as a perturbation of the chromosome. In (Wright, 1991), the authors chosen to make mutations only in coordinate directions due to the difficulty to perform global mutations compatible with the schemata theorem (it is a fundamental result for GAs (Goldberg, 1989; Holland, 1975)).

1.2.5. Fuzzy Logic

Fuzzy logic has rapidly become one of the most successful of today's technologies for developing sophisticated control systems. The reason for which is very simple. Fuzzy logic addresses such applications perfectly as it resembles human decision making with an ability to generate precise solutions from certain or approximate information. It fills an important gap in engineering design methods left vacant by purely mathematical approaches (e.g. linear control design), and purely logic-based approaches (e.g. expert systems) in system design. While other approaches require accurate equations to model real-world behaviors, fuzzy design can accommodate the ambiguities of real-world human language and logic. It provides both an intuitive method for describing systems in human terms and automates the conversion of those system specifications into effective models.

The first applications of fuzzy theory were primarily industrial, such as process control for cement kilns. However, as the technology was further embraced, fuzzy logic was used in more useful applications. In 1987, the first fuzzy logic-controlled subway was opened in Sendai in northern Japan. Here, fuzzy-logic controllers make subway journeys more comfortable with smooth braking and acceleration. Best of all, all the driver has to do is push the start button! Fuzzy logic was also put to work in elevators to reduce waiting time. Since then, the applications of Fuzzy Logic technology have virtually exploded, affecting things we use everyday.

Take for example, the fuzzy washing machine. There is a load of clothes in it, you press start and the machine begins to churn, automatically choosing the best cycle. For the fuzzy microwave, you place chili, potatoes, or etc. in it and push single button, then it cooks for the right time at the proper temperature. The fuzzy car maneuvers by following simple verbal instructions from its driver. It can even stop itself when there is an obstacle immediately ahead using sensors. But, practically the most exciting thing about it is the simplicity involved in operating it (Bezdek, 1993).

Fuzzy Sets:

Fuzzy Set Theory was formalized by Professor Lofti Zadeh at the University of California in 1965. What Zadeh proposed is very much a paradigm shift that first gained acceptance in the Far East and its successful application has ensured its adoption around the world.

A paradigm is a set of rules and regulations which defines boundaries and tells us what to do to be successful in solving problems within these boundaries. For example the use of transistors instead of vacuum tubes is a paradigm shift - likewise the development of Fuzzy Set Theory from conventional bivalent set theory is a paradigm shift. Bivalent Set Theory can be somewhat limiting if we wish to describe a 'humanistic' problem mathematically. For example, Fig 1.1 below illustrates bivalent sets to characterize the temperature of a room (Bandler and Kohout, 1980).

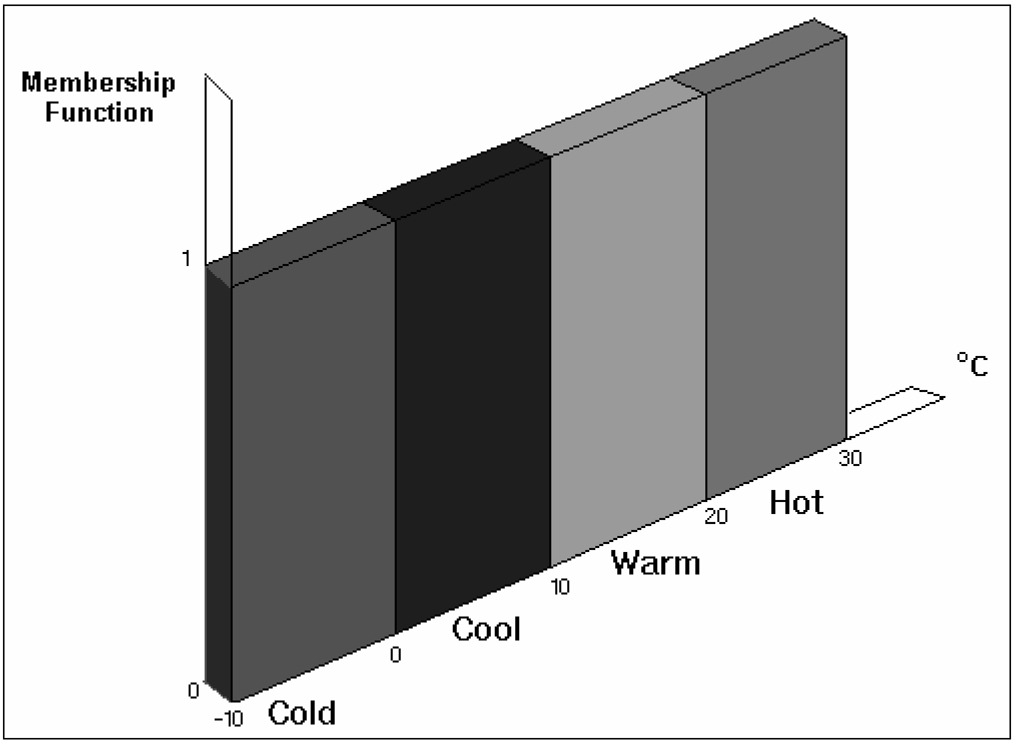

Figure 1.1. Bivalent Sets to Characterize the Temperature of a Room

The most obvious limiting feature of bivalent sets that can be seen clearly from the diagram is that they are mutually exclusive - it is not possible to have membership of more than one set (opinion would widely vary as to whether 0 degrees Celsius is 'cold' or 'cool' hence the expert knowledge we need to define our system is mathematically at odds with the humanistic world). Clearly, it is not accurate to define a transition from a quantity such as 'warm' to 'hot' by the application of one degree Celsius of heat. In the real world a smooth (unnoticeable) drift from warm to hot would occur (Bandler and Kohout, 1980).

This natural phenomenon can be described more accurately by Fuzzy Set Theory. Fig.1.2 below shows how fuzzy sets quantify the same information can describe this natural drift (Bandler and Kohout, 1980).

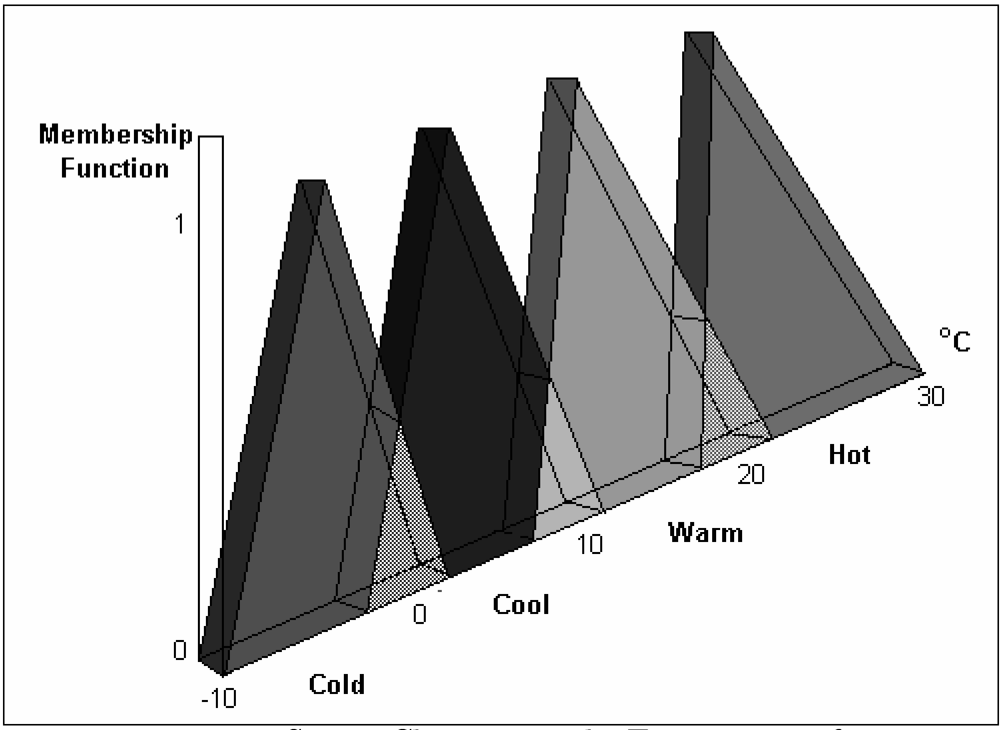

Figure 1.2. Fuzzy Sets to Characterize the Temperature of a Room

Fuzzy Rules:

Human beings make decisions based on rules. Although, we may not be aware of it, all the decisions we make are all based on computer like if-then statements. If the weather is fine, then we may decide to go out. If the forecast says the weather will be bad today, but fine tomorrow, then we make a decision not to go today, and postpone it till tomorrow. Rules associate ideas and relate one event to another. Fuzzy machines, which always tend to mimic the behavior of man, work the same way. However, the decision and the means of choosing that decision are replaced by fuzzy sets and the rules are replaced by fuzzy rules. Fuzzy rules also operate using a series of if-then statements. For instance, if X then A, if Y then B, where A and B are all sets of X and Y. Fuzzy rules define fuzzy patches, which is the key idea in fuzzy logic.

A machine is made smarter using a concept designed by Bart Kosko called the Fuzzy Approximation Theorem (FAT). The FAT theorem generally states that a finite number of patches can cover a curve. If the patches are large, then the rules are sloppy. If the patches are small then the rules are fine.

In a fuzzy system this simply means that all our rules can be seen as patches and the input and output of the machine can be associated together using these patches. Graphically, if the rule patches shrink, our fuzzy subset triangles get narrower. Simple enough? Yes, because even novices can build control systems that beat the best math models of control theory. Naturally, it is math-free system (Dubois and Prade, 1980).

Fuzzy Control:

Fuzzy control, which directly uses fuzzy rules, is the most important application in fuzzy theory. Using a procedure originated by Ebrahim Mamdani in the late 70s, three steps are taken to create a fuzzy controlled machine: fuzzification, rule evaluation and defuzzification (Mamdani and Assilian, 1975).

Step 1: Define inputs and output. The different levels of input and output of the platform is defined by specifying the membership functions for the fuzzy sets are shown in Figure 1.3.

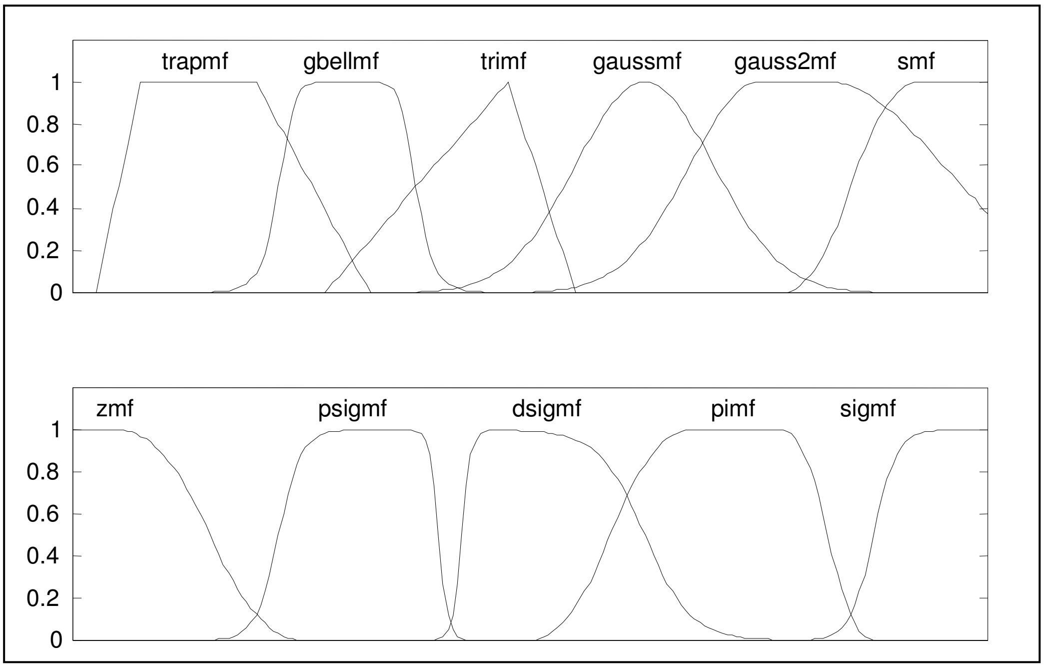

Figure 1.3. Membership Functions

Step 2: The next step is to define the fuzzy rules. The fuzzy rules are merely a series of if-then statements. These statements are usually derived by an expert to achieve optimum results.

Step 3: The result of the fuzzy controller is a fuzzy set. In order to choose an appropriate representative value as the final output (crisp values), defuzzification must be done. There are numerous defuzzification methods, some of them are illustrated in Figure 1.4.

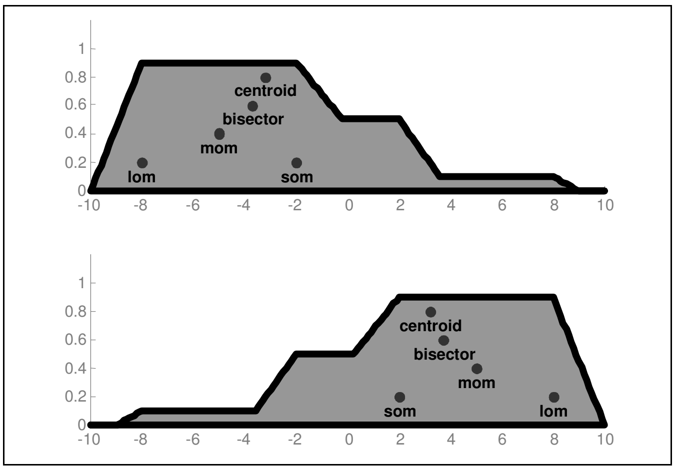

Figure 1.4. Methods of Defuzzification

1.2.6. Social Networks

A social network is a set of people or groups of people, "actors" in the jargon of the field, with some pattern of interactions or "ties" between them (Scott, 2000; Wasserman and Faust, 1994). Friendships among a group of individuals, business relationships between companies, and intermarriages between families are all examples of networks that have been studied in the past. Network analysis has a long history in sociology, the literature on the topic stretching back at least half a century to the pioneering work of Rapoport, Harary, and others in the 1940s and 1950s. Typically, network studies in sociology have been data-oriented, involving empirical investigation of real-world networks followed, usually, by graph theory analysis often aimed at determining the centrality or influence of the various actors.

Most recently, after a surge in interest in network structure among mathematicians and physicists, partly as a result of research on the Internet and the World Wide Web, another body of research has investigated the statistical properties of networks and methods for modeling networks either analytically or numerically (Albert and Barabasi, 2001; Strogatz, 2001). One important and fundamental result that has emerged from these studies concerns the numbers of ties that actors have to other actors, their so-called "degrees". It has been found that in many networks, the distribution of actors' degrees is highly skewed, with a small number of actors having an unusually large number of ties. Simulations and analytic work have suggested that this skewness could have an impact on the way in which communities operate, including the way information travels through the network and the robustness of networks to removal of actors (Albert, Jeong, and Barabasi, 2000; Callaway, Newman, Strogatz and Watts, 2000; Cohen, Erez, Ben-Avraham and Havlin, 2000).

1.2.7. Network Analysis

The structure of networks has been studied by mathematical graph theory (Bollobas, 1985; Bollobas, 1998; Janson, Luczak and Rucinski, 2000). Some basic ideas, used later by physicists, were proposed long ago by the incredibly prolific and outstanding Hungarian mathematician Paul Erdös and his collaborator Renyi (Erdös and Renyi, 1959; Erdös and Renyi, 1960). Nevertheless, the most intriguing type of growing networks, which evolve into scale-free structures, hasn't been studied by graph theory. Most of the results of graph theory (Flajolet, Knuth and Pittel, 1989; Janson, Knuth, Luczak and Pittel, 1993) are related to the simplest random graphs with Poisson distribution of connections (classical random graph) (Erdös and Renyi, 1959; Erdös and Renyi, 1960). Moreover, in graph theory, by definition, random graphs are graphs with Poisson distribution of connections (this term is used in a much more wide sense). Nevertheless, one should note very important results were obtained recently by mathematicians for graphs with arbitrary distribution of connections (Molloy and Reed, 1995; Molloy and Reed, 1998). The mostly empirical study of specific large random networks such as nets of citations in scientific literature has a long history (Garfield, 1972; Garfield, 1979; Lotka, 1926; Shockley, 1957). Unfortunately, their limited sizes did not allow getting reliable data and describing their structure until recently.

Fundamental concepts such as functioning and practical organization of large communications networks were elaborated by the "father" of the Internet, Paul Baran, (Baran, 1964). Actually, many present studies are based on his original ideas and use his terminology. What is the optimal design of communications networks? How may one ensure their stability and safety? These and many other vital problems were first studied by P. Baran in a practical context.

By the middle of 90's, the Internet and the World Wide Web (WWW) had reached very large sizes and continued to grow so rapidly that intensively developed search engines failed to cover a great part of the WWW (Claffy, Monk and McRobb, 1999; Lawrence and Giles, 1998; Lawrence and Giles, 1999). A clear knowledge of the structure of the WWW has become vitally important for its effective operation. The first experimental data, mostly for the simplest structural characteristics of the communications networks, were obtained in 1997-1999 (Albert, Jeong and Barabasi, 1999; Huberman and Adamic, 1999; Huberman, Pirolli, Pitkow and Lukose., 1998). Distributions of the number of connections in the networks and their surprisingly small average shortest-path lengths were measured. A special role of long-tailed, power-law distributions was revealed. After these findings, physicists started intensive study of evolving networks in various areas, from communications to biology and public relations.

The networks are graphs consisting of vertices (nodes) connected by edges (links). Edges may be directed or undirected (leading to directed and undirected networks, relatively). For definition of distances in a network, one sets lengths of all edges to be one.

Here we do not consider networks with unit loops (edges started and terminated at the same vertex) and multiple edges, i.e., we assume that only one edge may connect two vertices (One should note that multiple edges are encountered in some collaboration networks (Newman, 2001 *et al.*). Pairs of opposing edges connect some vertices in the WWW, in networks of protein-protein interactions, and in food webs. Also, protein-protein interaction nets and food webs contain unit loops.

The structure of a network is described by its adjacency matrix, whose elements consist of zeros and ones. An element of the adjacency matrix of a network with undirected edges, $b_{\mu\nu}$, is 1 if vertices μ and ν are connected, and is 0 otherwise. Therefore, the adjacency matrix of a network with undirected edges is symmetrical.

For a network with directed edges, an element of the adjacency matrix, $b_{\mu\nu}$, equals 1 if there is an edge from the vertex μ to the vertex ν, and equals 0 otherwise. In the case of a random network, an adjacency matrix describes only a particular member of the entire statistical ensemble of random graphs. Hence, what one observes is only a particular realization of this statistical ensemble and the adjacency matrix of this graph is only a particular member of the corresponding ensemble of matrices.

The statistics of the adjacency matrix of a random network contains complete information about the structure of the net, and, in principle, one has to study just the adjacency matrix. Generally, this is not an easy task, so that, instead of this, only a very restricted set of structural characteristics is usually considered.

The simplest and the most intensively studied one vertex characteristic is degree. Degree, k, of a vertex is the total number of its connections. In physical literature, this quantity is often called "connectivity" that has a quite different meaning in graph theory (Newman, 2000; Watts, 1999; Watts and Strogatz, 1998).

1.3. Problem Statement

The search for models that account for the complex behavior of biological, social and economic systems has been the motivation of much interdisciplinary work in the last decade (Poston and Stewart, 1978). In particular, the emergence of altruistic or cooperative behavior is a favorite problem of game theory approaches (Smith, 1982). In this context, the Prisoner's Dilemma game has been widely studied in different versions, as a standard model for the confrontation between cooperative and selfish behaviors, the later manifested by a defecting attitude, aspiring to obtain the greatest benefit from the interaction with another individual (Abramson and Kuperman, 2001).

Here, we will consider random networks of players which face a social dilemma or, in physical terms, a frustrated interaction (Ebel and Bornholdt, 2002). Imagine a course that the grades will be determined by round robin tournament (Smucker, Stanley and Ashlock, 1994) that every student gives grade to each other. Beyond the naive approach regarding the performance criteria, the grading network becomes an acquaintance network and begins to regard the acquaintanceship criteria. This hypothesis that people are composed of more than one "player" may explain some of the anomalies that occur in human experiments exploring game theory (Roberts, 2003).

In round robin grading process (Smucker, Stanley and Ashlock, 1994), human brain divides into two minds, and the grades are given by superposition of this duality. The strong anxiety of not being number one, instinct of close friend preservation, desire to curb the leaders turn the grader to a subjective unconscious prisoner's dilemma game player. This player does not give grades but makes cooperation or defection as a collapse of two minds as a result. These two minds are:

- First Mind: (Objective Grader) Grades the participants objectively without thinking other factors.
- Second Mind: (Prisoner) Grades the participants with consideration of closeness of the participant, position of the participant and self on the curve of the course grades.

In iterated prisoner's dilemma game, two single chromosomes are created as a result of these minds' decisions in every iteration with C (Cooperate) and D (Defect) bits. The 10-bit chromosomes have identical genes in both (like CC or DD); however, the chromosomes can have opposite genes (like CD or DC) that create two distinct alternatives to decide. There are two kinds of duality formed with these two minds; iterative duality and elemental duality (Roberts, 2003).

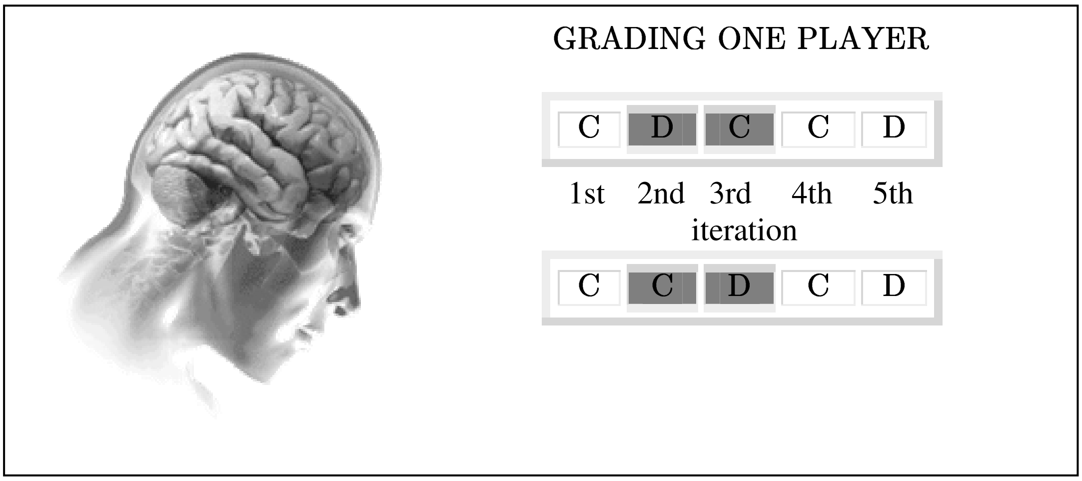

Figure 1.5. Iterative Duality (Points given to ONE player in FIVE iterations)

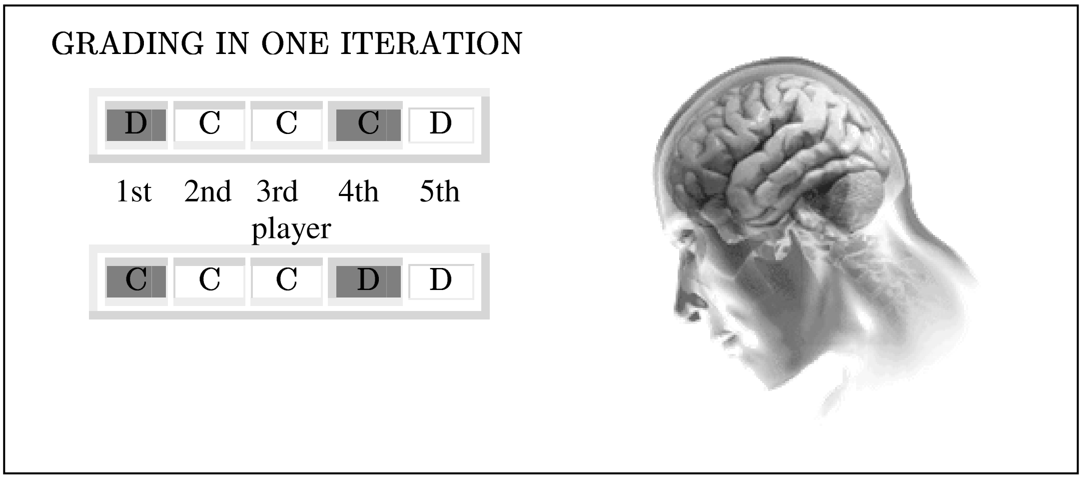

Figure 1.6. Elemental Duality (Points given to FIVE players in ONE iteration)

The chromosomes collapse in one master chromosome, when a human brain makes a decision. In this research, several collapsing methods will be investigated for surviving the iterated prisoner's game. These methods are fuzzy decision making, genetic algorithms and quantum inspiration on genetic in order to achieve best results in the game.

2. METHODOLOGY

2.1. The Case

In the case, the game has 10 participants and every participant will give a grade for all other participants in 10 iterations. Grades are from 0 to 5 that 0 is the worst and 5 is the best case. At this point of time, prisoner's dilemma game is started with complex dynamics and emergence of cooperation for strategy spaces confined to the strategies defecting and cooperating (Axelrod, 1984).

In the round robin tournament, players give each other a random grade. The mean must have a constant value of 2.5 as expected after 10 iterations and this will show the experiment is perfectly random.

2.1.1. Game Simulation

In the simulation, all points are given to players randomly. After 10 iterations, points given to each player are ranked, and loser with the least value is chosen. The main purpose is to enhance the loser's score and elevate it in the rank. All values have an average of 2.5. The raw points matrix is illustrated in Table 2.1 below.

Table 2.1 Raw Points Matrix

	A	B	C	D	E	F	G	H	I	J
A		3	2	3	1	2	2	0	3	4
B	5		3	3	1	1	4	0	5	3
C	3	5		2	5	0	0	1	3	4
D	2	3	0		5	0	3	4	4	3
E	2	3	2	1		5	3	5	5	0
F	2	3	3	1	3		1	1	1	2
G	5	3	3	0	4	5		3	3	3
H	2	1	5	1	2	3	0		0	2
I	2	5	4	2	0	0	4	0		5
J	3	1	0	3	3	0	4	3	0	

ITERATION 1

	A	B	C	D	E	F	G	H	I	J
A		4	5	2	3	1	2	2	4	0
B	4		0	4	2	2	4	0	3	5
C	3	0		2	0	4	3	0	4	4
D	1	5	2		5	4	1	3	4	0
E	3	2	4	3		4	2	0	1	2
F	5	5	4	1	5		3	5	4	5
G	2	4	5	0	5	3		1	5	5
H	3	5	3	5	0	5	0		1	3
I	5	0	5	5	0	0	5	5		4
J	3	1	5	1	5	0	3	3	4	

ITERATION 2

	A	B	C	D	E	F	G	H	I	J
A		4	1	2	5	1	0	2	1	0
B	5		3	2	0	4	3	5	0	4
C	3	4		1	2	0	0	5	3	2
D	5	0	3		3	3	2	2	0	1
E	5	1	3	1		0	1	5	1	4
F	0	5	3	3	4		1	2	1	1
G	0	2	3	0	2	3		3	4	4
H	3	5	3	0	4	3	3		0	2
I	5	1	5	2	2	2	5	5		5
J	2	0	1	3	0	1	0	2	4	

ITERATION 3

	A	B	C	D	E	F	G	H	I	J
A		5	0	5	4	1	2	1	2	0
B	0		1	4	0	5	4	0	1	5
C	1	2		0	3	4	2	3	5	4
D	2	4	4		1	4	3	3	3	4
E	4	5	3	5		4	5	0	5	2
F	1	0	5	4	0		5	4	4	4
G	4	2	1	2	2	1		4	1	3
H	0	3	3	3	4	5	4		1	3
I	5	3	1	0	4	0	5	1		1
J	2	3	5	2	4	5	2	0	3	

ITERATION 4

	A	B	C	D	E	F	G	H	I	J
A		4	5	2	4	3	2	1	3	0
B	0		5	4	5	2	3	3	3	0
C	2	3		0	4	5	0	0	3	2
D	5	4	0		2	2	5	1	3	3
E	0	1	3	3		2	4	3	4	5
F	2	5	1	1	0		0	3	3	3
G	0	3	2	5	3	4		4	4	1
H	1	4	3	2	1	5	3		0	4
I	3	2	0	4	3	3	0	4		4
J	5	4	3	3	2	3	1	2	5	

ITERATION 5

	A	B	C	D	E	F	G	H	I	J
A		5	4	1	1	2	2	0	4	3
B	4		5	0	3	1	3	4	5	5
C	3	1		1	5	1	4	2	0	5
D	1	4	1		4	2	3	3	5	0
E	0	0	1	2		2	2	2	3	4
F	4	0	3	3	1		0	4	1	0
G	5	2	5	0	2	2		2	4	0
H	3	3	2	0	3	0	0		1	5
I	5	0	1	4	1	0	5	2		3
J	5	3	1	1	3	2	4	3	3	

ITERATION 6

	A	B	C	D	E	F	G	H	I	J
A		1	1	3	4	1	0	4	4	0
B	1		2	4	5	0	2	5	4	1
C	3	2		0	2	4	0	1	0	0
D	3	1	1		3	2	4	1	0	0
E	4	2	2	5		5	3	2	2	5
F	0	5	1	5	1		0	0	1	1
G	4	3	5	4	5	4		1	0	2
H	1	3	2	0	3	5	3		4	5
I	2	4	1	0	5	0	3	2		1
J	4	1	3	3	5	5	4	3	4	

ITERATION 7

	A	B	C	D	E	F	G	H	I	J
A		2	4	5	3	3	1	4	3	4
B	3		0	3	2	5	3	3	3	0
C	0	1		3	0	0	1	4	5	3
D	3	4	4		1	4	4	5	5	2
E	5	2	5	3		4	2	4	5	0
F	3	2	5	3	5		5	5	2	4
G	1	0	1	0	3	4		0	0	0
H	1	0	5	1	4	0	3		0	4
I	1	4	2	1	1	4	2	1		0
J	3	4	5	5	5	4	2	1	4	

ITERATION 8

	A	B	C	D	E	F	G	H	I	J
A		1	3	1	4	0	1	3	1	0
B	0		3	5	0	1	5	0	5	0
C	5	0		3	2	2	3	4	4	5
D	5	2	2		5	2	2	1	2	2
E	0	5	1	0		0	0	1	1	0
F	2	4	1	5	1		2	4	1	3
G	2	5	0	1	4	5		5	4	1
H	0	0	3	0	1	3	3		5	0
I	1	3	1	0	1	4	2	0		1
J	2	3	1	5	4	1	4	2	4	

ITERATION 9

	A	B	C	D	E	F	G	H	I	J
A		4	4	0	0	0	2	2	5	5
B	3		3	5	5	2	0	0	1	5
C	0	0		0	5	1	0	1	4	0
D	5	4	4		1	0	3	4	5	3
E	1	3	0	0		3	1	2	1	3
F	3	5	0	2	4		0	1	5	2
G	1	2	0	2	3	4		0	0	3
H	4	4	3	4	2	2	0		3	1
I	5	2	2	1	4	2	4	2		4
J	4	2	5	4	3	2	4	0	3	

ITERATION 10

2.1.2. Converting to Prisoner's Dilemma Game

With the results of the round robin process, the grading plane is converted to prisoner's dilemma game plane, by changing grades into cooperation or defection. The grades which are between 0 and 2 indicate the defection, and grades between 3 and 5, indicate the cooperation. It will be easier to see prearranged cooperation and defection in the social network. Table 2.2 below shows the IPD matrix.

Table 2.2 IPD Matrix

	A	B	C	D	E	F	G	H	I	J
A		C	D	C	D	D	D	D	C	C
B	C		C	C	D	D	C	D	C	C
C	C	C		D	C	D	D	D	C	C
D	D	C	D		C	D	C	C	C	C
E	D	C	D	D		C	C	C	C	D
F	D	C	C	D	C		D	D	D	D
G	C	C	C	D	C	C		C	C	C
H	D	D	C	D	D	C	D		D	D
I	D	C	C	D	D	D	C	D		C
J	C	D	D	C	C	D	C	C	D	

ITERATION 1

	A	B	C	D	E	F	G	H	I	J
A		C	C	D	C	D	D	D	C	D
B	C		D	C	D	D	C	D	C	C
C	C	D		D	D	C	C	D	C	C
D	D	C	D		C	C	D	C	C	D
E	C	D	C	C		C	D	D	D	D
F	C	C	C	D	C		C	C	C	C
G	D	C	C	D	C	C		D	C	C
H	C	C	C	C	D	C	D		D	C
I	C	D	C	C	D	D	C	C		C
J	C	D	C	D	C	D	C	C	C	

ITERATION 2

	A	B	C	D	E	F	G	H	I	J
A		C	D	D	C	D	D	D	D	D
B	C		C	D	D	C	C	C	D	C
C	C	C		D	D	D	D	C	C	D
D	C	D	C		C	C	D	D	D	D
E	C	D	C	D		D	D	C	D	C
F	D	C	C	C	C		D	D	D	D
G	D	D	C	D	D	C		C	C	C
H	C	C	C	D	C	C	C		D	D
I	C	D	C	D	D	D	C	C		C
J	D	D	D	C	D	D	D	D	C	

ITERATION 3

	A	B	C	D	E	F	G	H	I	J
A		C	D	C	C	D	D	D	D	D
B	D		D	C	D	C	C	D	D	C
C	D	D		D	C	C	D	C	C	C
D	D	C	C		D	C	C	C	C	C
E	C	C	C	C		C	C	D	C	D
F	D	D	C	C	D		C	C	C	C
G	C	D	D	D	D	D		C	D	C
H	D	C	C	C	C	C	C		D	C
I	C	C	D	D	C	D	C	D		D
J	D	C	C	D	C	C	D	D	C	

ITERATION 4

	A	B	C	D	E	F	G	H	I	J
A		C	C	D	C	C	D	D	C	D
B	D		C	C	C	D	C	C	C	D
C	D	C		D	C	C	D	D	C	D
D	C	C	D		D	D	C	D	C	C
E	D	D	C	C		D	C	C	C	C
F	D	C	D	D	D		D	C	C	C
G	D	C	D	C	C	C		C	C	D
H	D	C	C	D	D	C	C		D	C
I	C	D	D	C	C	C	D	C		C
J	C	C	C	C	D	C	D	D	C	

ITERATION 5

	A	B	C	D	E	F	G	H	I	J
A		C	C	D	D	D	D	D	C	C
B	C		C	D	C	D	C	C	C	C
C	C	D		D	C	D	C	D	D	C
D	D	C	D		C	D	C	C	C	D
E	D	D	D	D		D	D	D	C	C
F	C	D	C	C	D		D	C	D	D
G	C	D	C	D	D	D		D	C	D
H	C	C	D	D	C	D	D		D	C
I	C	D	D	C	D	D	C	D		C
J	C	C	D	D	C	D	C	C	C	

ITERATION 6

	A	B	C	D	E	F	G	H	I	J
A		D	D	C	C	D	D	C	C	D
B	D		D	C	C	D	D	C	C	D
C	C	D		D	D	C	D	D	D	D
D	C	D	D		C	D	C	D	D	D
E	C	D	D	C		C	C	D	D	C
F	D	C	D	C	D		D	D	D	D
G	C	C	C	C	C	C		D	D	D
H	D	C	D	D	C	C	C		C	C
I	D	C	D	D	C	D	C	D		D
J	C	D	C	C	C	C	C	C	C	

ITERATION 7

	A	B	C	D	E	F	G	H	I	J
A		D	C	C	C	C	D	C	C	C
B	C		D	C	D	C	C	C	C	D
C	D	D		C	D	D	D	C	C	C
D	C	C	C		D	C	C	C	C	D
E	C	D	C	C		C	D	C	C	D
F	C	D	C	C	C		C	C	D	C
G	D	D	D	D	C	C		D	D	D
H	D	D	C	D	C	D	C		D	C
I	D	C	D	D	D	C	D	D		D
J	C	C	C	C	C	C	D	D	C	

ITERATION 8

	A	B	C	D	E	F	G	H	I	J
A		D	C	D	C	D	D	C	D	D
B	D		C	C	D	D	C	D	C	D
C	C	D		C	D	D	C	C	C	C
D	C	D	D		C	D	D	D	D	D
E	D	C	D	D		D	D	D	D	D
F	D	C	D	C	D		D	C	D	C
G	D	C	D	D	C	C		C	C	D
H	D	D	C	D	D	C	C		C	D
I	D	C	D	D	D	C	D	D		D
J	D	C	D	C	C	D	C	D	C	

ITERATION 9

	A	B	C	D	E	F	G	H	I	J
A		C	C	D	D	D	D	D	C	C
B	C		C	C	C	D	D	D	D	C
C	D	D		D	C	D	D	D	C	D
D	C	C	C		D	D	C	C	C	C
E	D	C	D	D		C	D	D	D	C
F	C	C	D	D	C		D	D	C	D
G	D	D	D	D	C	C		D	D	C
H	C	C	C	C	D	D	D		C	D
I	C	D	D	D	C	D	C	D		C
J	C	D	C	C	C	D	C	D	C	

ITERATION 10

2.1.3. Analysis

In simulation process, the grades are added up and the averages are calculated for determining the ranking. To calculate the deviations, mean is subtracted from the grades and this difference is divided by the grades. This method shows the percent deviation (gaming factor) between the grades and the average.

The IPD points are calculated by payoff matrix (Table 1.1) and these points are ranked. The IPD percent deviations are calculated by same method used in grades gaming factor (GF) evaluation. Also deviation ranks that are results of difference of percentages are calculated. The detailed table of data analysis is shown below in Table 2.3.

Table 2.3 Analysis of Players

	Grades	Grades Rank	Grade Percentage	IPD Points	IPD Rank	IPD Percentage	Deviance	Deviation Rank
A	2,61	4	4,30%	218	4	6,88%	2,58%	4
B	2,67	3	6,29%	210	5	3,33%	-2,96%	6
C	2,59	5	3,48%	219	3	7,31%	3,83%	2
D	2,22	10	-12,45%	168	10	-20,83%	-8,38%	10
E	2,73	2	8,58%	227	2	10,57%	2,00%	5
F	2,38	7	-5,09%	186	8	-9,14%	-4,05%	7
G	2,33	8	-7,10%	195	6	-4,10%	2,99%	3
H	2,27	9	-10,25%	176	9	-15,34%	-5,10%	9
I	2,74	1	8,95%	240	1	15,42%	6,47%	1
J	2,44	6	-2,23%	191	7	-6,28%	-4,06%	8

The data analysis helps us to find the "loser" of the game easily. Deviance (GF) column values indicate that the players are either a grader or a prisoner. A grader is a player for whom to grade is more important than to play the PD game. The grader has a low GF and very close to the objective grading however a prisoner is very subjective and has a high GF.

The Player D is the loser in all statistics whereas Player I is the winner. -8,38 value in Deviance column shows the player D is an objective grader more than a prisoner so player D seems to be the best player to choose for further investigation.

2.1.4. Cooperation and Defection

Two way cooperation/defection is the situation, that both players cooperate/defect each other respectively. Number of both cooperation and defection choices are counted to identify pre-arranged activities on the social network. Table 2.4 below shows the sum of cooperation and defection choices and differences between the sums.

Table 2.4 Two Way Matrices

COOPERATION

	A	B	C	D	E	F	G	H	I	J
A		5	3	2	5	1	0	0	4	4
B	5		3	6	1	1	4	4	4	2
C	3	3		1	2	2	2	4	3	3
D	2	6	1		2	3	2	3	3	3
E	5	1	2	2		4	3	2	2	3
F	1	1	2	3	4		2	4	1	3
G	0	4	2	2	3	2		4	4	3
H	0	4	4	3	2	4	4		1	3
I	4	4	3	3	2	1	4	1		5
J	4	2	3	3	3	3	3	3	5	

DEFECTION

	A	B	C	D	E	F	G	H	I	J
A		2	1	2	3	5	6	4	1	3
B	2		4	1	3	1	1	1	2	1
C	1	4		5	2	2	4	2	2	1
D	2	1	5		1	3	3	4	3	2
E	3	3	2	1		3	2	3	3	0
F	5	1	2	3	3		1	1	4	4
G	6	1	4	3	2	1		3	1	2
H	4	1	2	4	3	1	3		4	3
I	1	2	2	3	3	4	1	4		0
J	3	1	1	2	0	4	2	3	0	

DIFFERENCE

	A	B	C	D	E	F	G	H	I	J
A		3	2	0	2	**-4**	**-6**	**-4**	3	1
B	3		-1	**5**	-2	0	3	3	2	1
C	2	-1		**-4**	0	0	-2	2	1	2
D	0	5	-4		1	0	-1	-1	0	1
E	2	-2	0	1		1	1	-1	-1	3
F	-4	0	0	0	1		1	3	-3	-1
G	-6	3	-2	-1	1	1		1	3	1
H	-4	3	2	-1	-1	3	1		**-3**	0
I	3	2	1	0	-1	-3	3	-3		**5**
J	1	1	2	1	3	-1	1	0	5	

In difference two way matrices, black values with gray borders adumbrate pre-arranged defection while white values with dark gray borders adumbrate prearranged cooperation. This matrix also gives idea of closeness in the acquaintanceship network.

2.1.5. Social Network Analysis for the Case

The social network analysis for the case has two sections, cooperation analysis and defection analysis. The analysis is realized in Agna 2.11, to find out prearranged cooperation and defection in the network for further analysis of closeness in Fuzzy Logic Toolbox.

Cooperation Analysis:

For all nodes of the network, two-side cooperation is expurgated and cooperation network is constructed as Figure 2.1 below.

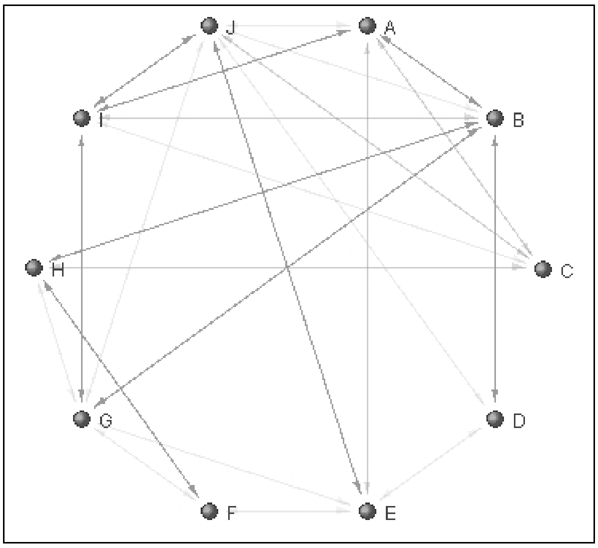

Figure 2.1. Cooperation Network

The thickness of the edges between the nodes indicates the recurrence of cooperation. The cooperation network diagram has unbalanced cooperation, which can be easily clustered. The degree analysis of the cooperation network of for each node is shown below in Table 2.5.

Table 2.5 The Cooperation Network Degree Analysis

Node	*Degree*	*Relative Degree*	*Weighted Degree*
A	5.0	0.5	11.0
B	6.0	0.6	17.0
C	4.0	0.4	7.0
D	3.0	0.3	7.0
E	5.0	0.5	8.0
F	3.0	0.3	5.0
G	6.0	0.6	10.0
H	4.0	0.4	9.0
I	5.0	0.5	14.0
J	7.0	0.7	14.0

In Table 2.5, the degrees indicate connectivity to other nodes and the weighted degrees of the nodes exposes the grader's cooperative characteristics. If weighted degree is large, the node is more cooperative. The most cooperative node is B while F is less.

Network analysis also contains basic concepts, like diameter, density and cohesion factor. The analysis of the networks with these factors is shown below in Table 2.6.

Table 2.6 The Cooperation Network Factor Analysis

Number of Nodes	10 inside, 0 outside
Number of Edges	24
Diameter	2.0
Density	0.533
Non-directed Density	0.266
Weighted Density	1.133
Cohesion	0.533

All nodes are inside the network with density of 0.266. Then network has 24 non-directed edges out of 90 possible edges. These edges are non-uniformly distributed over the nodes.

Defection Analysis:

For all nodes of the network, two-side defection is expurgated and defection network is constructed as Figure 2.2 below.

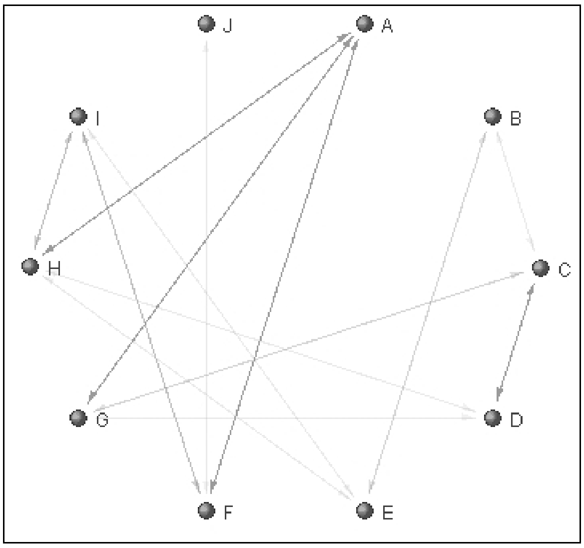

Figure 2.2. Defection Network

It is clearly seen that defection network has less edges than cooperation network. Also in the defection network, bold edges indicate more defection between nodes. The degree analysis of the defection network is shown in Table 2.7.

Table 2.7 The Defection Network Degree Analysis

Node	*Degree*	*Relative Degree*	*Weighted Degree*
A	3.0	0.3	14.0
B	2.0	0.2	3.0
C	3.0	0.3	7.0
D	3.0	0.3	6.0
E	3.0	0.3	4.0
F	3.0	0.3	8.0
G	3.0	0.3	9.0
H	4.0	0.4	9.0
I	3.0	0.3	7.0
J	1.0	0.1	1.0

The player A is more defecting while the player J is less. The network factor analysis is shown in Table 2.8.

Table 2.8 The Defection Network Factor Analysis

Number of Nodes	10 inside, 0 outside
Number of Edges	14
Diameter	4.0
Density	0.311
Non-directed Density	0.155
Weighted Density	0.755
Cohesion	0.311

Both network analysis models give a raw idea of acquaintanceship clusters in the network. The results of the analysis will be used in fuzzy logic approach.

Beneath several analysis, player D is chosen for further research because of the least number of defection, cooperation, and the least grade average and payoff points. The main idea is to turn player D from a grader to a prisoner with tuning the choices of D.

2.2. Preparation

The player D is chosen for further analysis, because of the player's having more focus on objective evaluating, less grades and points. The main aim is to simulate the chromosomes that are generated by the player, thus the first chromosome is the cooperation of defection choice of the player and second chromosome will be generated by fuzzy decision making.

The chromosomes are expurgated from the data mess, and classified into two parts. The first chromosome pack is the C or D choices of the Player D for all iterations regarding the other players separately. These matrices can be seen in Table 2.9.

Table 2.9 Player Matrices of Player D

D	D	D	C	D	C	D	C	C	C	C
A	C	D	D	C	D	D	C	C	D	D
D	C	C	D	C	C	C	D	C	D	C
B	C	C	D	C	C	D	C	C	C	C
D	D	D	C	C	D	D	D	C	D	C
C	D	D	D	D	D	D	D	C	C	D
D	C	C	C	D	D	C	C	D	C	D
E	D	C	D	C	C	D	C	C	D	D
D	D	C	C	C	D	D	D	C	D	D
F	D	D	C	C	D	C	C	C	C	D
D	C	D	D	C	C	C	C	C	D	C
G	D	D	D	D	C	D	C	D	D	D
D	C	C	D	C	D	C	D	C	D	C
H	D	C	D	C	D	D	D	D	D	C
D	C	C	D	C	C	C	D	C	D	C
I	D	C	D	D	C	C	D	D	D	D
D	C	D	D	C	C	D	D	D	D	C
J	C	D	C	D	C	D	C	C	C	C

The transpose of the player matrices gives us iteration matrices shown in Table 2.10 below.

Table 2.10 Iteration Matrices of the Player D

1	A	B	C	E	F	G	H	I	J
	D	C	D	C	D	C	C	C	C
	C	C	D	D	D	D	D	D	C

2	A	B	C	E	F	G	H	I	J
	D	C	D	C	C	D	C	C	D
	D	C	D	C	D	D	C	C	D

3	A	B	C	E	F	G	H	I	J
	C	D	C	C	C	D	D	D	D
	D	D	D	D	C	D	D	D	C

4	A	B	C	E	F	G	H	I	J
	D	C	C	D	C	C	C	C	C
	C	C	D	C	C	D	C	D	D

5	A	B	C	E	F	G	H	I	J
	C	C	D	D	D	C	D	C	C
	D	C	D	C	D	C	D	C	C

6	A	B	C	E	F	G	H	I	J
	D	C	D	C	D	C	C	C	D
	D	D	D	D	C	D	D	C	D

7	A	B	C	E	F	G	H	I	J
	C	D	D	C	D	C	D	D	D
	C	C	D	C	C	C	D	D	C

8	A	B	C	E	F	G	H	I	J
	C	C	C	D	C	C	C	C	D
	C	C	C	C	C	D	D	D	C

9	A	B	C	E	F	G	H	I	J
	C	D	D	C	D	D	D	D	D
	D	C	C	D	C	D	D	D	C

10	A	B	C	E	F	G	H	I	J
	C	C	C	D	D	C	C	C	C
	D	C	D	D	D	D	C	D	C

These iteration matrices will be used for creating duality for the first chromosome in genetic algorithms also in fuzzy decision making.

The first approach is generation of the output by fuzzy logic decision making model that is introduced in Section 2.3. Fuzzy logic chromosome generation is the second approach that is the second chromosome generation tool, with the factors that affect the outcome.

2.3. Fuzzy Decision Making

The second approach is generation of the output by fuzzy logic decision making model. To rationalize the use of fuzzy logic as the tool for modeling the iterated prisoner's dilemma game output, general block diagram of the decision system of the human brain is simulated in Figure 2.3.

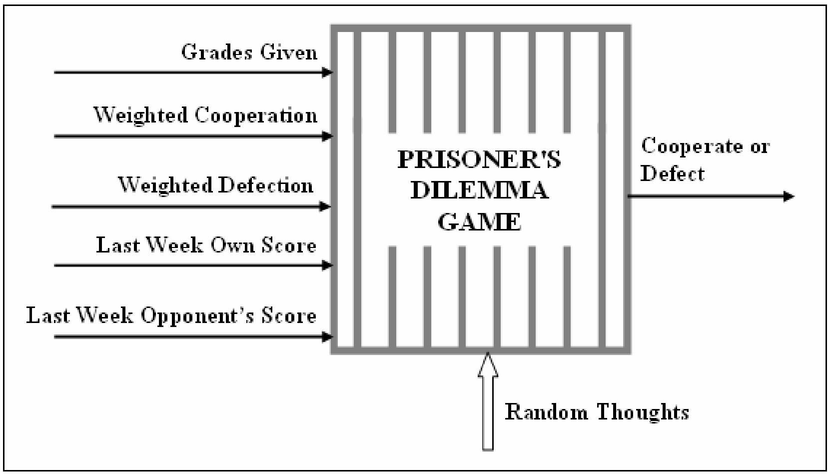

Figure 2.3. Human Thinking System in IPD

The output variable from the system is the choice of cooperate or defect (CHO). The primary variables that affect the choice are Grades Given (GR), Weighted Cooperation, Weighted Defection, Last Week's Opponent's Score (OPS) and Last Week's Own Score (OWS). The difference in weighted cooperation and defection gives game tactic for both own and opponent's side. These factors are the remaining inputs of the system, Own Tactic (OWT) and Opponent's Tactic (OPT). Also there is randomness in the system, when a human makes a decision that cannot be neglected.

There is very little (if any) objective knowledge of the system, which would take the form of mathematical equations expressing the output as a function of the input variables. There is, on the other hand, considerable subjective knowledge in the form of linguistic information about the effects of the input variables on the output variable. Some examples of the subjective knowledge are provided by the following statements: "if OWT is defecting then CHO is defect" as one task statement or "if OPS is low and GR is high then CHO is cooperate".

Another characteristic of this system that makes it suitable for a fuzzy logic approach is the "fuzziness" of all the system variables. These variables are fuzzified in Matlab Fuzzy Logic Toolbox and the following membership function (MF) diagrams are attained.

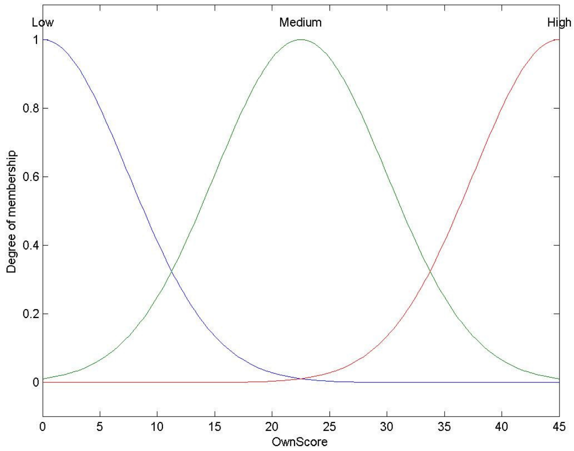

Figure 2.4. Own Score MF

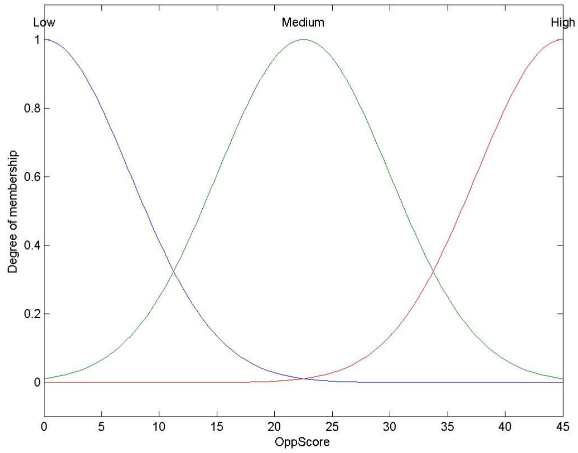

Figure 2.5. Opponent's Score MF

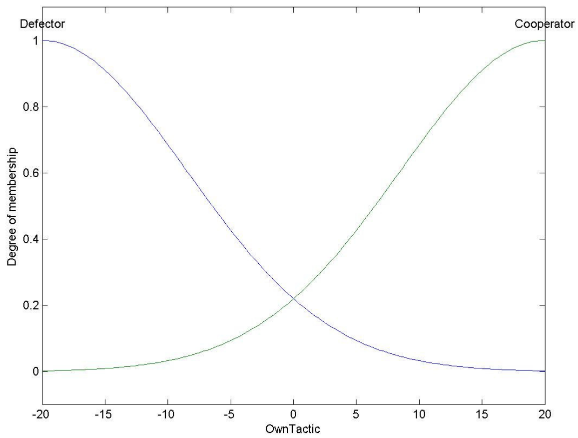

Figure 2.6. Own Tactic MF

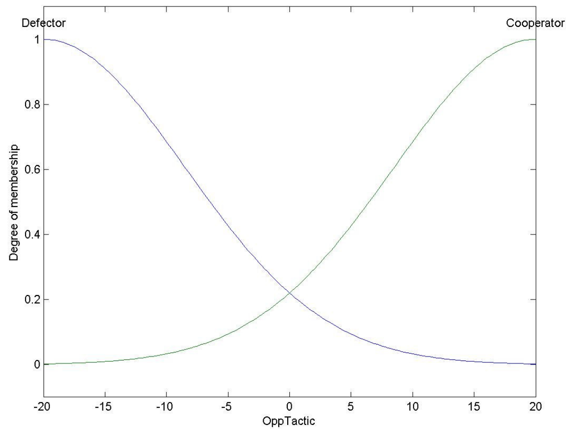

Figure 2.7. Opponent's Tactic MF

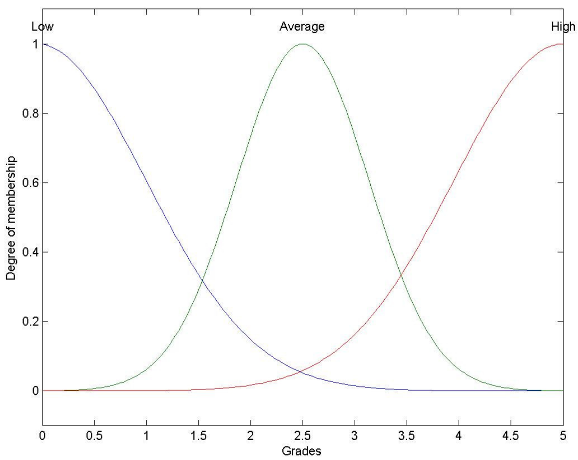

Figure 2.8. Grades MF

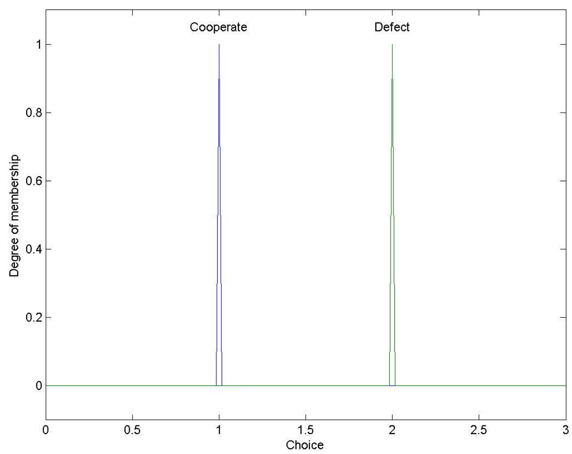

Figure 2.9. Choice MF

Scores are representing the last week scores for both players, and calculated dynamically after every iteration. The OWS and OPS inputs are divided in three sets, low, medium and high. Tactics are static variables that vary from player to player and represent difference between weighted cooperation and defection values that is attained from social network analysis. The OWT and OPT are divided into two sets, cooperator and defector. Grades are taken directly from the case and fuzzified into three sets, low, average and high.

The output CHO is delineated as crisp binary values C and D however, constructed as triangles. Projection of the intersection point is 1.5 where C corresponds to 1 and D corresponds to 2.

Usually, fuzzy control rules can be designed on the basis of an expert's experience and an operator's control actions. On the other hand, it can also be derived by the self-organizing or clustering method (Bezdek, 1981). The procedure for deriving a fuzzy control rule is as follows. For the Ith iteration, a rule is constructed and the weight of this rule is:

$$w_i = \mu_{GR} \wedge \mu_{OWT} \wedge \mu_{OPT} \wedge \mu_{OWS} \wedge \mu_{OPS}$$

where:

w_i is the weight of the Ith rule, based on the data of the Ith test iteration.

$\wedge$ is minimum operator, which refers to the minimum values.

μ_{GR} is the highest degree of membership of given grades among all fuzzy sets at the Ith test iteration.

μ_{OWT} is the highest degree of membership of own tactic among all fuzzy sets at the Ith test iteration.

μ_{OPT} is the highest degree of membership of opponent's tactic among all fuzzy sets at the Ith test iteration.

μ_{OWS} is the highest degree of membership of own score with precipitation among all fuzzy sets at the Ith test iteration.

μ_{OPS} is the highest degree of membership of opponent's score among all fuzzy sets at the Ith test iteration.

After reviewing all data from all players and iterations, a total of 105 rules are obtained. These rules not only express the relationship between the input and output, but also clearly show the mechanism decision making in IPD. Figure 2.10 below represents the block diagram of the system.

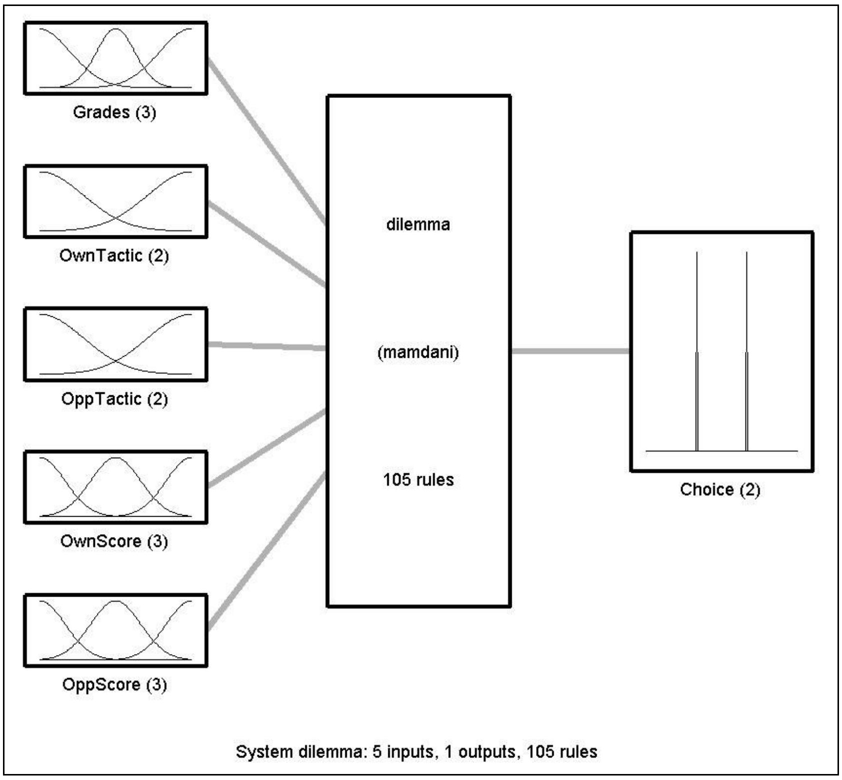

Figure 2.10. Block Diagram of the Decision Making Model

Mutational randomization is applied to the results for simulating random thoughts after the results are achieved. For all cells in the iterative matrices, possibility of mutational randomization is set to 0.10, which converts C to D and D to C randomly.

2.4. Chromosome Generation with Fuzzy Logic

Two minded human brain can be simulated with one pair of chromosomes for IPD game. The first chromosome states what is seen, while second chromosome states what is felt. The genes of first chromosome are filled with the randomized data of the real case, and outputs of modified fuzzy logic model for second chromosome. The fuzzy logic model mentioned in Section 2.3 is modified for chromosome generation so grades (GR) are excluded from the inputs. The chromosomes as the output of this model represent an emotional manner, and express the situation of "What is the choice of the player without anything to be graded?". The inputs, OWT, OPT, OWS, OPS and the output, CHO, stay same with identical membership functions and rules are reduced to 54. The human feeling system and block diagram are in Figure 2.11 and 2.12 below.

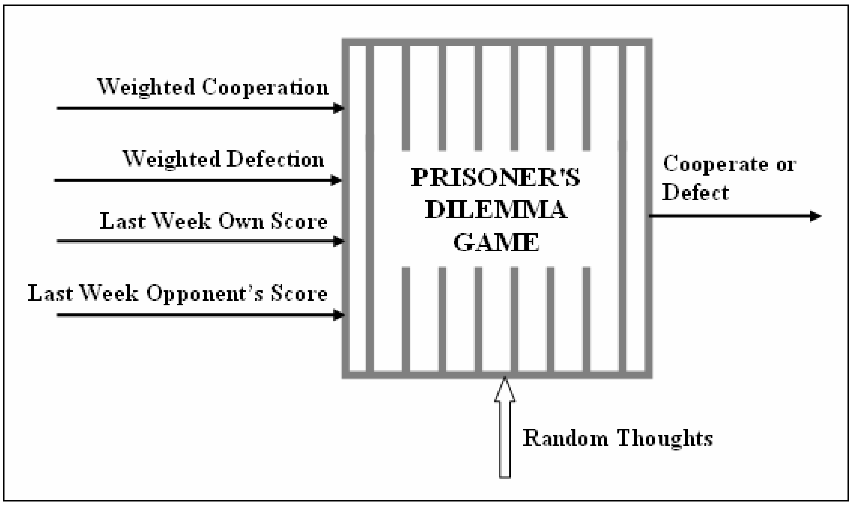

Figure 2.11. Human Feeling System in IPD

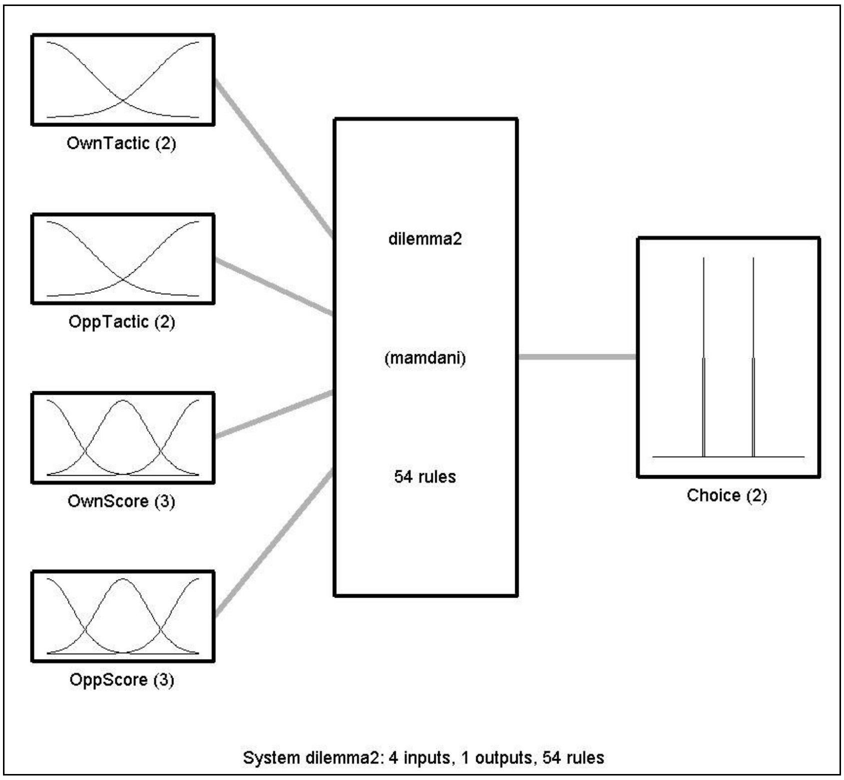

Figure 2.12. Block Diagram of the Chromosome Generation Model

2.5. Genetic Optimization

Simple breeding methods are used in genetic optimization without any extra crossovers or mutations. First chromosome comes from choices in the case and second from fuzzy chromosome generation tool.

Every iteration, children of these chromosomes are created by one point crossover and all children's fitness values are calculated. Fitness calculation process belongs to last iteration's chromosome; the child with best fitness is inserted to the grading vector of player D for all iterations. The presentation of the genetic optimization model is shown below in Figure 2.13.

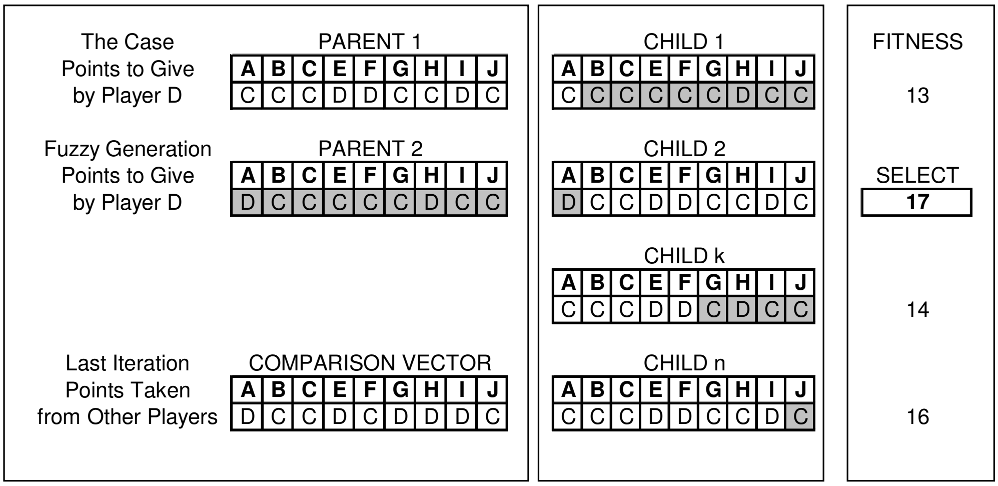

Figure 2.13. Genetic Optimization Process

The selected child is substituted to the relevant vector of the case and for the first iteration "last iteration" chromosome is fixed to the grades given by Player D of the first iteration.

2.6. Quantum Inspired Genetic Optimization

All procedures are the similar to Genetic Optimization section except the genes that the player has conflict on them. Quantum inspiration is about the superposition of the genes and taken from the concept of qubits.

Matching genes are operated directly, while these conflicting genes are considered as qubits and expressed with * character. These * genes are generated with combinatorial generation and the best fitness is derived. The child chromosome with best fitness is inserted to the grading vector of player D. The presentation of the quantum inspired genetic optimization model is shown below in Figure 2.14.

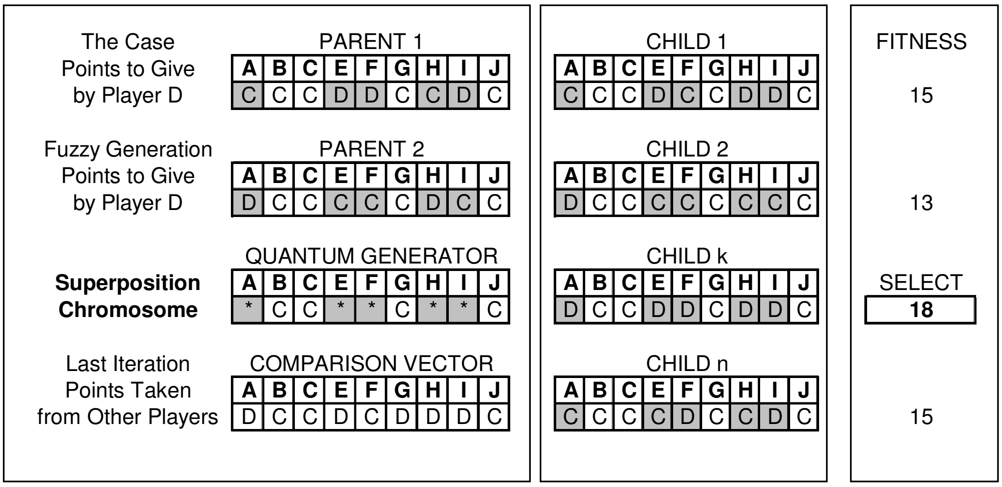

Figure 2.14. Quantum Inspired Genetic Optimization Process

3. RESULTS

3.1. Results of Fuzzy Decision Making

The first fuzzy model is absolutely dependent of the player choices made in the case so little deviation takes place. Results are collected from fuzzy decision making model and mutation is applied to all cells with 0.1 probability value. The IPD matrices of player D are shown below in Figure 3.1.

1	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	C	C	C	C
RECEIVE	C	C	D	D	D	D	D	D	C

2	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	C	D	C	C	D
RECEIVE	D	C	D	C	D	D	C	C	D

3	A	B	C	E	F	G	H	I	J
GIVE	C	D	C	C	C	D	D	D	D
RECEIVE	D	D	D	D	C	D	D	D	C

4	A	B	C	E	F	G	H	I	J
GIVE	D	C	C	D	C	C	C	C	C
RECEIVE	C	C	D	C	C	D	C	D	D

5	A	B	C	E	F	G	H	I	J
GIVE	C	C	D	D	D	C	D	C	C
RECEIVE	D	C	D	C	D	C	D	C	C

6	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	D	C	C	D
RECEIVE	D	D	D	D	C	D	D	C	D

7	A	B	C	E	F	G	H	I	J
GIVE	C	D	D	C	D	C	D	D	C
RECEIVE	C	C	D	C	C	C	D	D	C

8	A	B	C	E	F	G	H	I	J
GIVE	C	C	C	D	C	C	C	D	C
RECEIVE	C	C	C	C	C	D	D	D	C

9	A	B	C	E	F	G	H	I	J
GIVE	C	D	D	C	D	D	D	D	C
RECEIVE	D	C	C	D	C	D	D	D	C

10	A	B	C	E	F	G	H	I	J
GIVE	C	D	C	D	D	C	C	D	C
RECEIVE	D	C	D	D	D	D	C	D	C

Figure 3.1. IPD Matrices of the Fuzzy Decision Making Model

Points of the players, new ranking information, difference between points and shifting degrees are listed below in Table 3.1.

Table 3.1 Results of the Fuzzy Decision Making Model

	Old IPD Points	Old IPD RANK	Old IPD Percentage	New IPD Points	New IPD RANK	New IPD Percentage	Difference	Shifting
A	218	4	6,88%	199	7	7,44%	-19	-3
B	210	5	3,33%	178	4	-3,48%	-32	+1
C	219	3	7,31%	200	3	7,90%	-19	0
D	168	10	-20,83%	173	8	-6,47%	5	+2
E	227	2	10,57%	205	2	10,15%	-22	0
F	186	8	-9,14%	170	9	-8,35%	-16	-1
G	195	6	-4,10%	174	5	-5,86%	-21	+1
H	176	9	-15,34%	154	10	-19,61%	-22	-1
I	240	1	15,42%	214	1	13,93%	-26	0
J	191	7	-6,28%	175	6	-5,26%	-16	+1

With the model, the rank of our first time loser player D has become 8th; however it was 10th in the original case. Player D has 11 points advance in the game with 173 points consequently.

3.2. Results of Fuzzy Derived Genetic Optimization

The second fuzzy model is independent to the choices made in the case, so results declare the strategies of player D and also the behavior to the players. The IPD matrices of fuzzy chromosome generation tool are illustrated in Figure 3.2 below.

1	A	B	C	E	F	G	H	I	J
GIVE	D	C	C	C	D	C	C	C	C
RECEIVE	C	C	D	D	D	D	D	D	C

2	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	C	D	C	C
RECEIVE	D	C	D	C	D	D	C	C	D

3	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	C	D	C	C
RECEIVE	D	D	D	D	C	D	D	D	C

4	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	C	C	C	C
RECEIVE	C	C	D	C	C	D	C	D	D

5	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	C	D	C	C
RECEIVE	D	C	D	C	D	C	D	C	C

6	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	C	D	C	C
RECEIVE	D	D	D	D	C	D	D	C	D

7	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	C	D	C	C
RECEIVE	C	C	D	C	C	C	D	D	C

8	A	B	C	E	F	G	H	I	J
GIVE	D	C	C	C	D	C	D	C	C
RECEIVE	C	C	C	C	C	D	D	D	C

9	A	B	C	E	F	G	H	I	J
GIVE	D	C	C	C	D	C	D	C	C
RECEIVE	D	C	C	D	C	D	D	D	C

10	A	B	C	E	F	G	H	I	J
GIVE	D	C	C	C	D	C	D	C	C
RECEIVE	D	C	D	D	D	D	C	D	C

Figure 3.2. IPD Matrices of the Fuzzy Chromosome Generation

These choices are taken as the first parent chromosome and the real values in the data set of the case are taken as second. The child chromosomes are propagated with combination of the parents and one child chromosome is chosen regarding its fitness. The fitness is calculated by comparing with the previous iteration choices, for predicting next iteration. This method shows superior gain for previous observation data set, however somehow does not fully fit current data set for best payoff.

The combined chromosomes for each iteration are shown below in Figure 3.3.

1	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	C	C	C	C
RECEIVE	C	C	D	D	D	D	D	D	C

2	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	C	D	C	D
RECEIVE	D	C	D	C	D	D	C	C	D

3	A	B	C	E	F	G	H	I	J
GIVE	D	D	C	C	C	D	D	D	D
RECEIVE	D	D	D	D	C	D	D	D	C

4	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	C	C	C	C
RECEIVE	C	C	D	C	C	D	C	D	D

5	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	D	D	C	D	C	C
RECEIVE	D	C	D	C	D	C	D	C	C

6	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	C	D	C	D
RECEIVE	D	D	D	D	C	D	D	C	D

7	A	B	C	E	F	G	H	I	J
GIVE	D	D	D	C	D	C	D	D	D
RECEIVE	C	C	D	C	C	C	D	D	C

8	A	B	C	E	F	G	H	I	J
GIVE	D	C	C	C	D	D	D	C	D
RECEIVE	C	C	C	C	C	D	D	D	C

9	A	B	C	E	F	G	H	I	J
GIVE	D	C	C	D	D	D	D	D	C
RECEIVE	D	C	C	D	C	D	D	D	C

10	A	B	C	E	F	G	H	I	J
GIVE	D	C	C	D	D	C	C	C	C
RECEIVE	D	C	D	D	D	D	C	D	C

Figure 3.3. IPD Matrices of the Genetic Optimization Model

The payoff matrices are calculated and illustrated in Table 3.2 below.

Table 3.2 Results of the Genetic Optimization Model

	Old IPD Points	Old IPD RANK	Old IPD Percentage	New IPD Points	New IPD RANK	New IPD Percentage	Difference	Shifting
A	218	4	6,88%	177	7	-2,94%	-41	-3
B	210	5	3,33%	184	4	0,98%	-26	+1
C	219	3	7,31%	199	3	8,44%	-20	0
D	168	10	-20,83%	183	5	0,44%	15	+5
E	227	2	10,57%	207	2	11,98%	-20	0
F	186	8	-9,14%	160	9	-13,88%	-26	-1
G	195	6	-4,10%	178	6	-2,36%	-17	0
H	176	9	-15,34%	143	10	-27,41%	-33	-1
I	240	1	15,42%	222	1	17,93%	-18	0
J	191	7	-6,28%	169	8	-7,81%	-22	-1

With the model, the rank of Player D has become 5th and Player D has 11 points advance in the game with 183 points.

3.3. Results of Fuzzy Derived Quantum Genetic Optimization

The values obtained from fuzzy chromosome generation tool are inserted to quantum inspired genetic algorithm and following IPD matrices are obtained in Figure 3.4.

1	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	C	C	C	C
RECEIVE	C	C	D	D	D	D	D	D	C

2	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	D	D	C	D
RECEIVE	D	C	D	C	D	D	C	C	D

3	A	B	C	E	F	G	H	I	J
GIVE	D	D	D	C	D	D	D	D	D
RECEIVE	D	D	D	D	C	D	D	D	C

4	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	D	D	C	C	C	C
RECEIVE	C	C	D	C	C	D	C	D	D

5	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	D	D	C	D	C	C
RECEIVE	D	C	D	C	D	C	D	C	C

6	A	B	C	E	F	G	H	I	J
GIVE	D	C	D	C	D	C	D	C	D
RECEIVE	D	D	D	D	C	D	D	C	D

7	A	B	C	E	F	G	H	I	J
GIVE	D	D	D	C	D	C	D	D	D
RECEIVE	C	C	D	C	C	C	D	D	C

8	A	B	C	E	F	G	H	I	J
GIVE	D	C	C	D	D	C	D	C	D
RECEIVE	C	C	C	C	C	D	D	D	C

9	A	B	C	E	F	G	H	I	J
GIVE	D	D	D	C	D	D	D	D	D
RECEIVE	D	C	C	D	C	D	D	D	C

10	A	B	C	E	F	G	H	I	J
GIVE	D	C	C	D	D	C	D	C	C
RECEIVE	D	C	D	D	D	D	C	D	C

Figure 3.4. IPD Matrices of the Quantum Inspired Genetic Optimization Model

The payoff matrices are calculated and illustrated in Table 3.3 below.

Table 3.3 Results of the Quantum Inspired Genetic Optimization Model

	Old IPD Points	Old IPD RANK	Old IPD Percentage	New IPD Points	New IPD RANK	New IPD Percentage	Difference	Shifting
A	218	4	6,88%	177	7	-2,54%	-41	-3
B	210	5	3,33%	181	5	-0,28%	-29	0
C	219	3	7,31%	192	4	5,47%	-27	-1
D	168	10	-20,83%	197	3	7,87%	29	+7
E	227	2	10,57%	205	2	11,46%	-22	0
F	186	8	-9,14%	157	9	-15,61%	-29	-1
G	195	6	-4,10%	178	6	-1,97%	-17	0
H	176	9	-15,34%	140	10	-29,64%	-36	-1
I	240	1	15,42%	222	1	18,24%	-18	0
J	191	7	-6,28%	166	8	-9,34%	-25	-1

With the model, the rank of Player D has become 3rd and Player D has 29 points advance in the game with 197 points.

3.4. Corporative Returns of the Models

The outcomes are collected from the models and compared in Table 3.4 below.

Table 3.4 Comparison Table of the Models

	FUZZY POINTS	FUZZY RANK	FUZZY DIF.	FUZZY SHIFTING	GA POINTS	GA RANK	GA DIF.	GA SHIFTING	QGA POINTS	QGA RANK	QGA DIF.	QGA SHIFTING
A	199	7	-19	-3	177	7	-41	-3	177	7	-41	-3
B	178	4	-32	+1	184	4	-26	+1	181	5	-29	0
C	200	3	-19	0	199	3	-20	0	192	4	-27	-1
D	173	8	5	+2	183	5	15	+5	197	3	29	+7
E	205	2	-22	0	207	2	-20	0	205	2	-22	0
F	170	9	-16	-1	160	9	-26	-1	157	9	-29	-1
G	174	5	-21	+1	178	6	-17	0	178	6	-17	0
H	154	10	-22	-1	143	10	-33	-1	140	10	-36	-1
I	214	1	-26	0	222	1	-18	0	222	1	-18	0
J	175	6	-16	+1	169	8	-22	-1	166	8	-25	-1

Although all models ensure advance of the points and increase of the rank for Player D, the quantum inspired genetic optimization model shows superiority over other models. These result sets grant evidence of game in round robin tournaments. Strategic behaviors strongly affect the results and all results can be changed by concentrating on one player choices. The changes in the results can be easily seen in Figure 3.5 and Figure 3.6 below.

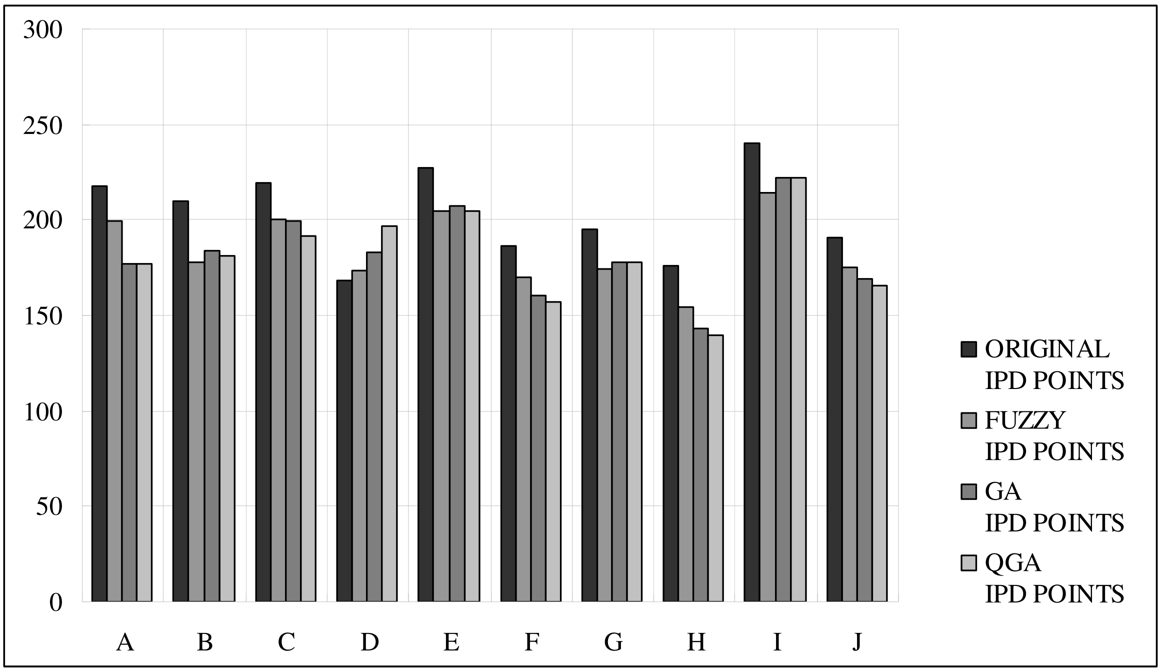

Figure 3.5. Comparison of the Players' Points for each Method

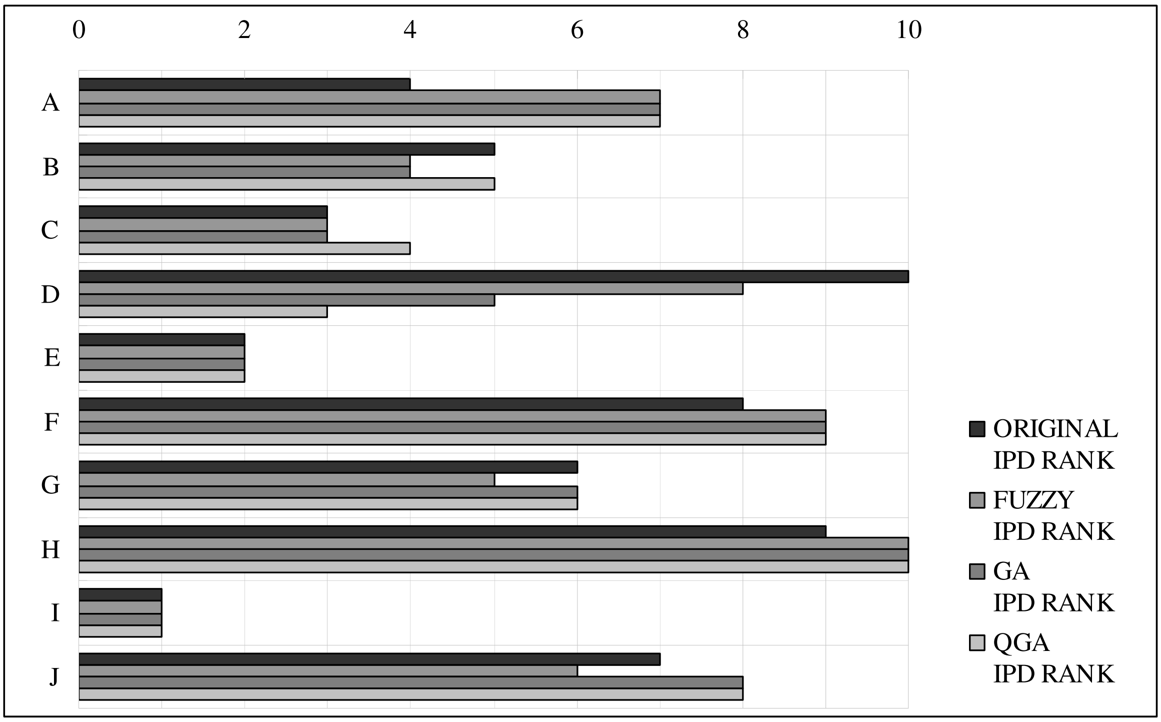

Figure 3.6. Comparison of the Players' Ranks for each Method

The first experiment shows the superiority of quantum inspired genetic algorithms; however, 30 additional experiments are made to support this superiority. The rule bases and the algorithms are remained basically identical, while the grades are generated randomly for each experiment. The networks that the player D is the loser in grades are chosen and points for each player are calculated. The ones that have the least gaming factors for player D are selected for further analysis.

The results of the experiments are shown in Table 3.5. The increase in the points of player D can be easily seen in the table. The points are combined together and the clusters as the point ranges for each method are illustrated in Figure 3.7. The groups in the diagram support the main idea of fine tuning of fuzzy logic results. Also, in Figure 3.8, the point curves are plotted for better visibility of the navigation.

Table 3.5 Results of the Experiments

	CASE		FUZZY		GA		QGA	
EXPERIMENTS	**RANK**	**POINT**	**RANK**	**POINT**	**RANK**	**POINT**	**RANK**	**POINT**
1	10	174	8	191	7	192	6	198
2	10	173	9	178	3	205	1	214
3	10	174	10	185	7	195	2	208
4	10	161	9	182	6	194	6	197
5	10	158	10	170	7	180	5	200
6	10	165	9	178	6	194	3	204
7	10	183	7	191	6	200	3	212
8	10	172	9	189	4	202	3	204
9	10	180	8	192	6	197	4	207
10	10	177	8	193	3	214	3	218
11	10	167	9	180	7	194	2	206
12	10	179	8	192	7	195	5	201
13	10	179	6	194	4	207	1	215
14	10	168	9	179	4	201	2	208
15	10	174	6	201	3	214	1	216
16	10	181	8	188	7	200	2	213
17	10	168	9	184	5	200	2	210
18	10	166	8	180	5	202	3	211
19	10	168	7	197	4	208	3	212
20	10	176	9	183	7	196	6	200
21	10	183	4	205	2	216	1	228
22	10	168	8	189	7	195	4	203
23	10	177	3	201	2	210	2	213
24	10	162	8	183	5	205	2	214
25	10	168	9	186	6	193	5	197
26	10	170	9	182	6	192	5	198
27	10	154	8	180	6	182	6	194
28	10	162	7	193	4	198	3	208
29	10	158	8	189	7	194	4	199
30	10	167	4	200	3	212	3	218

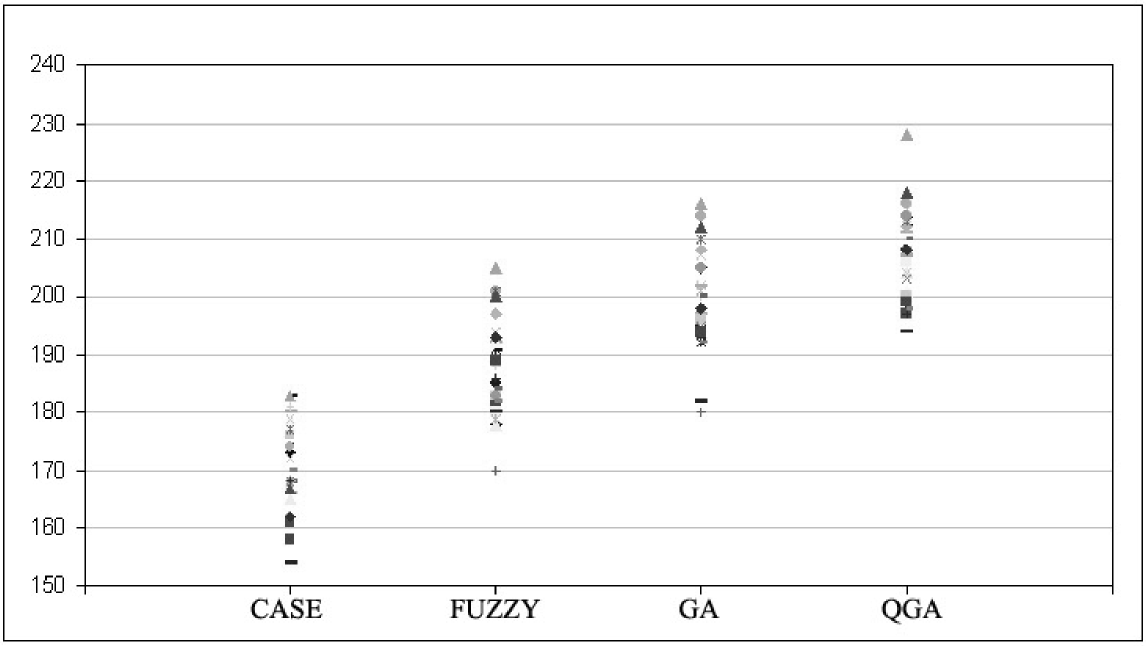

Figure 3.7. Point Ranges of the Experiments

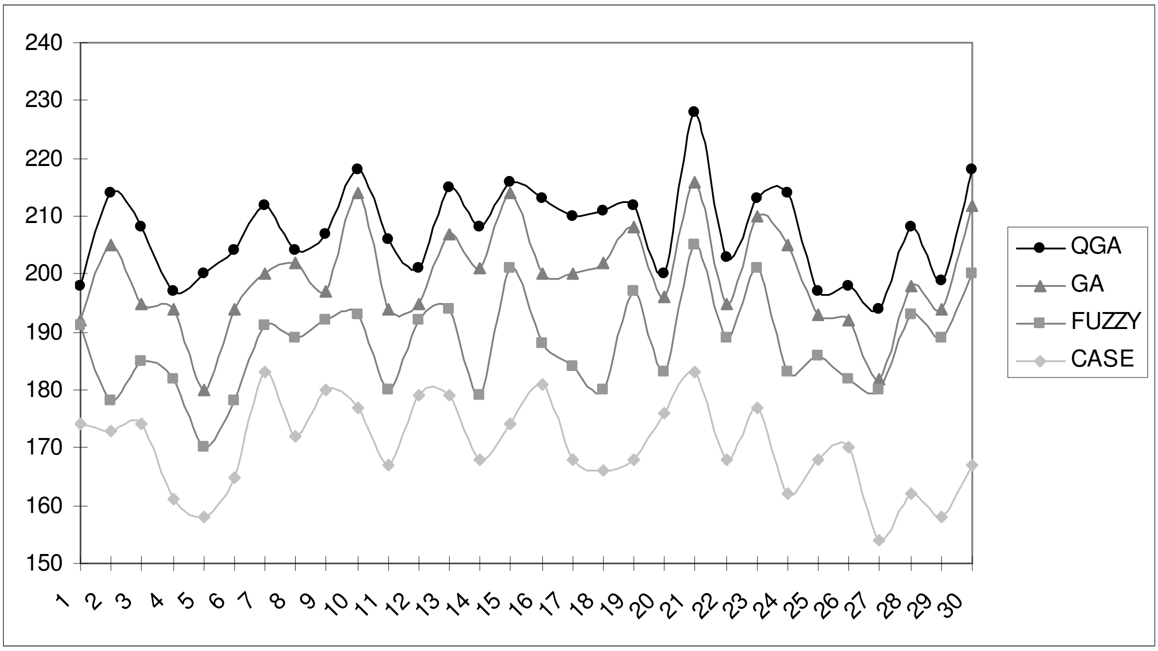

Figure 3.8. Points Navigation of the Experiments

The results of the experiments can be summarized that;

- The payoff of the methods is independent of the random networks created for the loser player with the least GF.
- Fuzzy decision making method increases the points and rank while genetic and quantum inspired genetic tools have better results.
- All methods have increase in points for each experiment; however, rarely these increases are not adequate for increase in ranking.

4. DISCUSSION AND CONCLUSIONS

Human brain activities are completely cryptic, however somehow it can be possible to formulate. The formulation of the human brain in decision making process is the main task of this project for identifying social network circumstances and prearranged activities in the network. Fuzzy logic emerged to formulate the process with linguistic information in human brain strategies.

Realizing prearranged cooperation or defection in a round robin tournament has proven to be a very difficult task with great number of grades. It would be merely possible if the grades change to prisoner's dilemma game space. Prisoner's dilemma game space has binary values C and D to specify the cooperation and the defection.

The concepts in human brain strategies and prisoner's dilemma game combined to analyze complex interactions of decision making process. These interactions are specified as inputs of the fuzzy logic model for loser player and choices of the player are added to the system in order to examine the differences.

Only the interactions are added in fuzzy chromosome generation and the results are compared with real case data with genetic algorithms for best fitness. Best fit values are selected as the vector to play for the loser player.

Finally, the chromosomes created by fuzzy chromosome generation tool, compared with real case values and the genes that have conflict turned to a choice in quantum inspired genetic optimization method to achieve best fitness constraints.

By comparing the quantum genetic optimization model with the genetic optimization model and the fuzzy decision making model, it can be concluded that:

- The quantum genetic model appears to be more accurate and reliable for revealing the increase of ranking and points.
- The quantum genetic model is superior in advocating the most efficient strategic method in human brain decision making.

Statistically; the chosen player's rank and points increase with the quantum genetic model as illustrated in Figure 3.7.

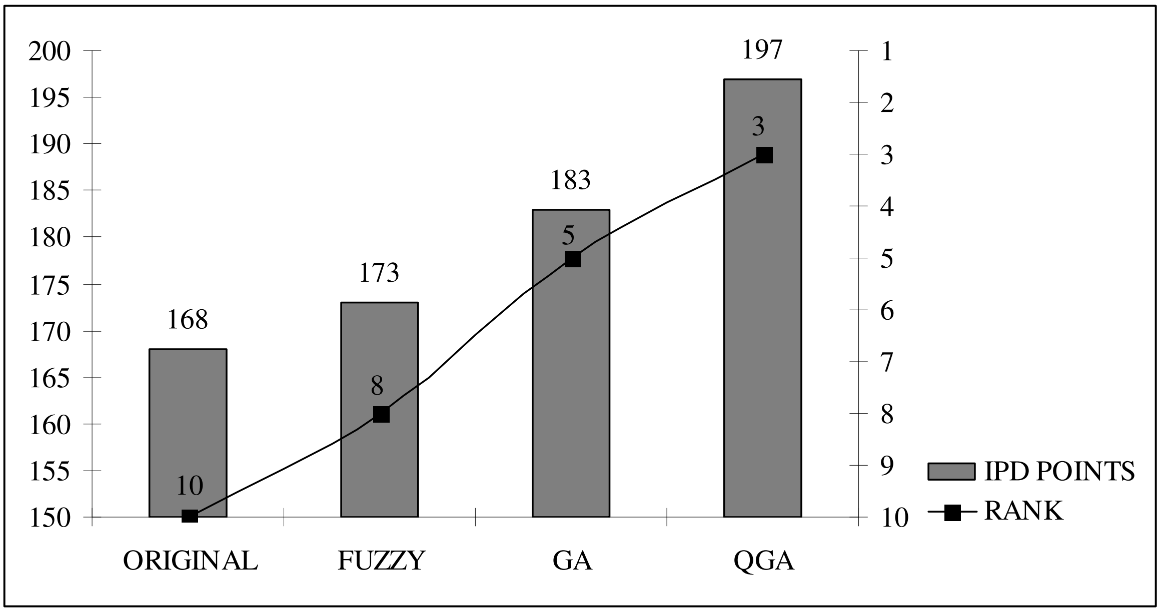

Figure 3.9. Comparison of the Selected Player Points and Ranks for each Method

Eventually; fuzzy logic model is helpful to formulate the human decision making strategies, while genetic algorithms is the useful tool to optimize the fuzzy results. Quantum inspired genetic optimization is like the standard genetics with many point random crossovers that is extremely assisting for optimization in prisoner's dilemma game.

5. FUTURE RESEARCH

The following cases should be considered as future research:

- The game system can be considered as a dynamic system, so all players must be in the game and decide to cooperate or defect regarding the situations that the Player D faces.
- The number of players and iterations can be increased.
- All of the nodes of the system can be real players, for better simulating the human thinking.
- All of the nodes except one can be real players, and the one can be a computer for comparing the strategies.

APPENDIX A. SCORE TABLES

Table A.1 Scores of Fuzzy Decision Making Model

		1	2	3	4	5	6	7	8	9	10	
4	**A**	19	29	29	17	15	29	23	18	13	26	199
5	**B**	26	21	15	27	27	15	19	13	23	18	178
3	**C**	22	27	30	20	23	19	15	26	13	27	200
2	**E**	22	22	19	19	20	26	28	20	24	27	205
9	**F**	17	17	19	23	26	5	27	25	16	12	170
7	**G**	17	18	14	29	15	20	21	22	18	17	174
10	**H**	23	16	19	14	17	19	13	27	22	7	154
1	**I**	26	26	13	24	30	25	21	26	26	23	214
6	**J**	25	23	22	25	12	22	10	18	21	22	175
8	**D**	13	16	12	19	20	12	25	21	21	14	173

Table A.2 Scores of Genetic Generation Model

		1	2	3	4	5	6	7	8	9	10	
7	**A**	19	29	25	17	11	29	20	15	9	22	177
4	**B**	26	21	15	27	27	15	19	13	26	21	184
3	**C**	22	27	30	16	23	19	15	26	16	27	199
2	**E**	22	22	19	22	20	26	28	23	20	27	207
9	**F**	17	13	19	20	26	5	27	22	16	12	160
5	**G**	17	22	14	29	15	24	21	22	18	17	182
10	**H**	23	13	19	14	17	15	13	23	22	7	143
1	**I**	26	26	13	24	30	25	21	30	26	27	222
8	**J**	25	23	22	25	12	22	7	15	21	22	169
5	**D**	13	18	13	20	21	12	29	25	19	12	182

Table A.3 Scores of Quantum Inspired Genetic Model

		1	2	3	4	5	6	7	8	9	10	
7	**A**	19	29	25	17	11	29	20	15	9	22	177
5	**B**	26	21	15	27	27	15	19	13	23	21	181
4	**C**	22	27	26	16	23	19	15	26	13	27	192
2	**E**	22	22	19	19	20	26	28	20	24	27	205
9	**F**	17	13	16	20	26	5	27	22	16	12	157
6	**G**	17	18	14	29	15	24	21	22	18	17	178
10	**H**	23	13	19	14	17	15	13	23	22	4	140
1	**I**	26	26	13	24	30	25	21	30	26	27	222
8	**J**	25	23	22	25	12	22	7	15	18	22	166
3	**D**	13	19	16	22	21	12	29	27	24	14	197

APPENDIX B. FUZZY DECISION MAKING RULES

1. If (Grades is High) then (Choice is Cooperate) (0.5)
2. If (Grades is Low) then (Choice is Defect) (0.5)
3. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) then (Choice is Cooperate) (0.4)
4. If (OwnTactic is Defector) and (OppTactic is Defector) then (Choice is Defect) (0.4)
5. If (OwnTactic is Defector) and (OppTactic is Cooperator) then (Choice is Cooperate) (0.4)
6. If (OwnTactic is Cooperator) and (OppTactic is Defector) then (Choice is Defect) (0.4)
7. If (OwnScore is Low) then (Choice is Defect) (0.3)
8. If (OwnScore is High) then (Choice is Cooperate) (0.3)
9. If (OppScore is High) then (Choice is Defect) (0.3)
10. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OppScore is Low) then (Choice is Cooperate) (0.7)
11. If (OwnTactic is Cooperator) and (OppTactic is Defector) and (OppScore is Low) then (Choice is Defect) (0.7)
12. If (OwnTactic is Defector) and (OppTactic is Cooperator) and (OppScore is Low) then (Choice is Cooperate) (0.7)
13. If (OwnTactic is Defector) and (OppTactic is Defector) and (OppScore is Low) then (Choice is Defect) (0.7)
14. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OppScore is Medium) then (Choice is Cooperate) (0.8)
15. If (OwnTactic is Defector) and (OppTactic is Defector) and (OppScore is Medium) then (Choice is Defect) (0.8)
16. If (OwnTactic is Defector) and (OppTactic is Cooperator) and (OppScore is Medium) then (Choice is Cooperate) (0.4)
17. If (OwnTactic is Cooperator) and (OppTactic is Defector) and (OppScore is Medium) then (Choice is Defect) (0.4)
18. If (OwnTactic is Defector) then (Choice is Defect) (0.2)
19. If (OwnTactic is Cooperator) then (Choice is Cooperate) (0.2)
20. If (OppTactic is Cooperator) then (Choice is Cooperate) (0.1)
21. If (OppTactic is Defector) then (Choice is Defect) (0.1)
22. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is Low) then (Choice is Cooperate) (1)
23. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is Medium) then (Choice is Cooperate) (0.8)
24. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is High) then (Choice is Cooperate) (0.6)
25. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is High) then (Choice is Cooperate) (0.3)
26. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Medium) then (Choice is Cooperate) (0.4)
27. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Cooperate) (0.5)
28. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (0.2)
29. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.1)
30. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is Low) then (Choice is Cooperate) (0.2)
31. If (Grades is Average) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is Medium) then (Choice is Cooperate) (0.6)
32. If (Grades is Average) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is High) then (Choice is Cooperate) (0.3)
33. If (Grades is Average) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is High) then (Choice is Defect) (0.1)
34. If (Grades is Average) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Medium) then (Choice is Cooperate) (0.2)
35. If (Grades is Average) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Cooperate) (0.3)
36. If (Grades is Average) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (0.4)
37. If (Grades is Average) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.2)
38. If (Grades is Low) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is Medium) then (Choice is Defect) (0.1)
39. If (Grades is Low) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is High) then (Choice is Cooperate) (0.1)
40. If (Grades is Low) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is High) then (Choice is Defect) (0.3)
41. If (Grades is Low) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Medium) then (Choice is Defect) (0.2)
42. If (Grades is Low) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Defect) (0.1)
43. If (Grades is Low) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (0.6)
44. If (Grades is Low) and (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.4)
45. If (Grades is Low) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (1)
46. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is Medium) then (Choice is Cooperate) (0.1)
47. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is High) then (Choice is Cooperate) (0.2)
48. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is High) then (Choice is Cooperate) (0.3)
49. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Medium) then (Choice is Defect) (0.2)
50. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Defect) (0.3)
51. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (0.3)
52. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.2)
53. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is Low) then (Choice is Cooperate) (0.2)
54. If (Grades is Average) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is Medium) then (Choice is Cooperate) (0.3)
55. If (Grades is Average) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is High) then (Choice is Defect) (0.2)
56. If (Grades is Average) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is High) then (Choice is Defect) (0.3)
57. If (Grades is Average) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Medium) then (Choice is Defect) (0.4)
58. If (Grades is Average) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Defect) (0.3)
59. If (Grades is Average) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (0.4)
60. If (Grades is Average) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.2)
61. If (Grades is Low) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is Medium) then (Choice is Defect) (0.4)
62. If (Grades is Low) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is High) then (Choice is Cooperate) (0.1)
63. If (Grades is Low) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is High) then (Choice is Defect) (0.3)
64. If (Grades is Low) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Defect) (0.1)
65. If (Grades is Low) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (0.3)
66. If (Grades is Low) and (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.3)
67. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is Medium) then (Choice is Defect) (0.6)
68. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is High) then (Choice is Defect) (0.5)
69. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is High) then (Choice is Defect) (0.7)
70. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is Medium) then (Choice is Defect) (0.6)
71. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Defect) (0.7)
72. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.9)
73. If (Grades is High) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is Low) then (Choice is Defect) (0.7)
74. If (Grades is Average) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is Medium) then (Choice is Defect) (0.7)
75. If (Grades is Average) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is High) then (Choice is Defect) (0.8)
76. If (Grades is Average) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is High) then (Choice is Defect) (0.8)
77. If (Grades is Average) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is Medium) then (Choice is Defect) (0.9)
78. If (Grades is Average) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Defect) (0.7)
79. If (Grades is Average) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (0.9)
80. If (Grades is Average) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.8)
81. If (Grades is Low) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is Medium) then (Choice is Defect) (0.8)
82. If (Grades is Low) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is High) then (Choice is Defect) (0.1)
83. If (Grades is Low) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is High) then (Choice is Defect) (0.5)
84. If (Grades is Low) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Defect) (0.2)
85. If (Grades is Low) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (0.5)
86. If (Grades is Low) and (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.6)
87. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is Medium) then (Choice is Defect) (0.8)
88. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is High) then (Choice is Defect) (0.9)
89. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is High) then (Choice is Defect) (0.9)
90. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is Medium) then (Choice is Defect) (0.8)
91. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Defect) (0.9)
92. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.7)
93. If (Grades is High) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is Low) then (Choice is Defect) (0.7)
94. If (Grades is Average) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is Medium) then (Choice is Defect) (0.5)
95. If (Grades is Average) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is High) then (Choice is Defect) (0.4)
96. If (Grades is Average) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is High) then (Choice is Defect) (0.6)
97. If (Grades is Average) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is Medium) then (Choice is Defect) (0.6)
98. If (Grades is Average) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Defect) (0.5)
99. If (Grades is Average) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.5)
100. If (Grades is Low) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is Medium) then (Choice is Defect) (0.9)
101. If (Grades is Low) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is High) then (Choice is Defect) (0.3)
102. If (Grades is Low) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is High) then (Choice is Defect) (0.6)
103. If (Grades is Low) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Defect) (0.6)
104. If (Grades is Low) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (0.7)
105. If (Grades is Low) and (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.8)

Figure B.1. Fuzzy Decision Making Rules

APPENDIX C. FUZZY CHROMOSOME GENERATION RULES

1. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) then (Choice is Cooperate) (0.4)
2. If (OwnTactic is Defector) and (OppTactic is Defector) then (Choice is Defect) (0.4)
3. If (OwnTactic is Defector) and (OppTactic is Cooperator) then (Choice is Cooperate) (0.4)
4. If (OwnTactic is Cooperator) and (OppTactic is Defector) then (Choice is Defect) (0.4)
5. If (OwnScore is Low) then (Choice is Defect) (0.3)
6. If (OwnScore is High) then (Choice is Cooperate) (0.3)
7. If (OppScore is High) then (Choice is Defect) (0.3)
8. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OppScore is Low) then (Choice is Cooperate) (0.7)
9. If (OwnTactic is Cooperator) and (OppTactic is Defector) and (OppScore is Low) then (Choice is Defect) (0.7)
10. If (OwnTactic is Defector) and (OppTactic is Cooperator) and (OppScore is Low) then (Choice is Cooperate) (0.7)
11. If (OwnTactic is Defector) and (OppTactic is Defector) and (OppScore is Low) then (Choice is Defect) (0.7)
12. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OppScore is Medium) then (Choice is Cooperate) (0.8)
13. If (OwnTactic is Defector) and (OppTactic is Defector) and (OppScore is Medium) then (Choice is Defect) (0.8)
14. If (OwnTactic is Defector) and (OppTactic is Cooperator) and (OppScore is Medium) then (Choice is Cooperate) (0.4)
15. If (OwnTactic is Cooperator) and (OppTactic is Defector) and (OppScore is Medium) then (Choice is Defect) (0.4)
16. If (OwnTactic is Defector) then (Choice is Defect) (0.2)
17. If (OwnTactic is Cooperator) then (Choice is Cooperate) (0.2)
18. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is Low) then (Choice is Cooperate) (1)
19. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is Medium) then (Choice is Cooperate) (0.8)
20. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is High) then (Choice is Cooperate) (0.6)
21. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is High) then (Choice is Cooperate) (0.3)
22. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Medium) then (Choice is Cooperate) (0.4)
23. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Cooperate) (0.5)
24. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (0.2)
25. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.1)
26. If (OwnTactic is Cooperator) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is Low) then (Choice is Cooperate) (0.2)
27. If (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (1)
28. If (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is Medium) then (Choice is Cooperate) (0.1)
29. If (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is High) then (Choice is Cooperate) (0.2)
30. If (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is High) then (Choice is Cooperate) (0.3)
31. If (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Medium) then (Choice is Defect) (0.2)
32. If (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Defect) (0.3)
33. If (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (0.3)
34. If (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.2)
35. If (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is Low) and (OppScore is Low) then (Choice is Cooperate) (0.2)
36. If (OwnTactic is Defector) and (OppTactic is Cooperator) and (OwnScore is High) and (OppScore is Low) then (Choice is Cooperate) (0.3)
37. If (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is Medium) then (Choice is Defect) (0.6)
38. If (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is High) then (Choice is Defect) (0.5)
39. If (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is High) then (Choice is Defect) (0.7)
40. If (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is Medium) then (Choice is Defect) (0.6)
41. If (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Defect) (0.7)
42. If (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.9)
43. If (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is Low) then (Choice is Defect) (0.5)
44. If (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is Low) then (Choice is Defect) (0.3)
45. If (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is Medium) then (Choice is Defect) (0.8)
46. If (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is High) then (Choice is Defect) (0.9)
47. If (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is High) then (Choice is Defect) (0.9)
48. If (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is Medium) then (Choice is Defect) (0.8)
49. If (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Medium) and (OppScore is Low) then (Choice is Defect) (0.9)
50. If (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is Medium) then (Choice is Defect) (0.7)
51. If (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is Low) then (Choice is Defect) (0.7)
52. If (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is High) and (OppScore is Low) then (Choice is Defect) (0.4)
53. If (OwnTactic is Cooperator) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (0.8)
54. If (OwnTactic is Defector) and (OppTactic is Defector) and (OwnScore is Low) and (OppScore is High) then (Choice is Defect) (0.8)

Figure C.1. Fuzzy Chromosome Generation Rules

APPENDIX D. GENETIC GENERATION

	A	B	C	E	F	G	H	I	J	
1	C	C	C	D	D	C	C	D	C	GIVEN
2	D	C	C	C	C	C	D	C	C	DESIRE TO GIVE
	Q			Q	Q		Q	Q		
	d	d	d	c	d	c	d	d	d	
	1	5	5	0	5	0	1	1	5	23

1

	1	2	3	4	5	6	7	8	9		
r	**D**	**C**	**C**	**D**	**C**	**D**	**D**	**D**	**C**	REFERENCE VECTOR	
1	C	C	C	C	C	C	D	C	C		1
s1	0	3	3	0	3	0	1	0	3	13	s1
2	C	C	C	C	C	C	D	C	C		2
s2	0	3	3	0	3	0	1	0	3	13	s2
3	C	C	C	C	C	C	D	C	C		3
s3	0	3	3	0	3	0	1	0	3	13	s3
4	C	C	C	D	C	C	D	C	C		4
s4	0	3	3	1	3	0	1	0	3	14	s4
5	C	C	C	D	D	C	D	C	C		5
s5	0	3	3	1	5	0	1	0	3	16	s5
6	C	C	C	D	D	C	D	C	C		6
s6	0	3	3	1	5	0	1	0	3	16	s6
7	C	C	C	D	D	C	C	C	C		7
s7	0	3	3	1	5	0	0	0	3	15	s7
8	C	C	C	D	D	C	C	D	C		8
s8	0	3	3	1	5	0	0	1	3	16	s8

2

	1	2	3	4	5	6	7	8	9		
r	**D**	**C**	**C**	**D**	**C**	**D**	**D**	**D**	**C**	REFERENCE VECTOR	
1	D	C	C	D	D	C	C	D	C		1
s1	1	3	3	1	5	0	0	1	3	17	s1
2	D	C	C	D	D	C	C	D	C		2
s2	1	3	3	1	5	0	0	1	3	17	s2
3	D	C	C	D	D	C	C	D	C		3
s3	1	3	3	1	5	0	0	1	3	17	s3
4	D	C	C	C	D	C	C	D	C		4
s4	1	3	3	0	5	0	0	1	3	16	s4
5	D	C	C	C	C	C	C	D	C		5
s5	1	3	3	0	3	0	0	1	3	14	s5
6	D	C	C	C	C	C	C	D	C		6
s6	1	3	3	0	3	0	0	1	3	14	s6
7	D	C	C	C	C	C	D	D	C		7
s7	1	3	3	0	3	0	1	1	3	15	s7
8	D	C	C	C	C	C	D	C	C		8
s8	1	3	3	0	3	0	1	0	3	14	s8

Figure D.1. Genetic Chromosome Generation and Best Fitness Method

REFERENCES

Abramson, G., and M., Kuperman, 2001, “Social Games in a Social Network”, *Physical Review E 63, 030901(R).*

Albert, R., and A. L., Barabasi, 2001, *Rev. Mod. Phys.*

Albert, R., H. Jeong, and A. L., Barabasi, 1999, “The Diameter of the World Wide Web”, *Nature*, pp 401, 130.

Albert, R., H., Jeong, and A. L., Barabasi, 2000, *Nature*, pp 406: 378–382.

Axelrod, R., 1984, “The Evolution of Cooperation”, *London Penguin Books.*

Bandler, W., and L.J., Kohout, 1980, “Fuzzy Power Sets and Fuzzy Implication Operators”, *Fuzzy Sets and Systems*, pp 4:13-30.

Baran, P., 1964, “Introduction to Distributed Communications Networks”, *RM-3420-PR.*

Barnum, H., C. M., Caves, C. A., Fuchs, R. Jozsa, and B., Schumacher, 1966, *Phys. Rev. Lett.*, pp 76: 2818.

Bell, S. J., 1964, “On the Einstein-Podolsky-Rosen Paradox”, *Physics*, 1:195–200.

Bennett, C. H, and A. J. G., Hey, 1999, *Feynman and Computation*, p 177.

Bennett, C. H., and D. P. Vincenzo, 2000, *Nature,* pp 406:247.

Bennett, C. H., D., Brassard, C., Crepeau, R., Jozsa, A., Peres, and W. K., Wooters, 1993 *Phys. Rev. Lett.*, pp 70: 1895.

Bezdek, J. C., 1981, “Pattern Recognition with Fuzzy Objective Function Algorithms”. *Plenum Press, New York.*

Bezdek, J. C., 1993, "Fuzzy Models: What Are They, and Why?", *IEEE Transactions on Fuzzy Systems*, pp 1-6.

Bohm, D., 1952, "A Suggested Interpretation of the Quantum Theory in Terms of Hidden Variables I and II", *Physical Review*, pp 85:166–193.

Bollobas, B., 1985, "Random Graphs", *London Academic Press.*

Bollobas, B., 1998, "Modern Graph Theory", *New York: Springer.*

Broglie, L. D, 1924, "A Tentative Theory of Light Quanta", *Philosophical Magazine*, pp 47:446–458.

Caldeira, O., and A. J., Legget, 1983, "Quantum Tunnelling in a Dissipative System. *Annals of Physics*, pp 149:374–456.

Callaway, D. S., M. E. J, Newman, S. H., Strogatz, and D. J., Watts, 2000, *Phys. Rev. Lett.*, pp 85: 5468–5471.

Claffy, K., T. E., Monk, and D., McRobb, 1999, "Internet Tomography", *Nature Web Matters.*

Clinton, J. D, and L. H., Germer, 1927, "Diffraction of Electrons by a Crystal of Nickel", *Physical Review*, pp 30:705–740.

Cohen, R., K., Erez, D., Ben-Avraham, and S. Havlin, 2000, *Phys. Rev. Lett.*, pp 85: 4626–4628.

Compton, A. H., 1923, "Wave-length Measurements of Scattered X-rays". *Physical Review*, pp 21:715.
Cournot, 1897, "Researches into the Mathematical Principles of the Theory of Wealth", Edited by N. Bacon. *New York, Macmillan.*

Deutsch, D., 1985, *Proc. R. Soc. Lond.*, pp 400: 97.

Dieks, D., 1982, *Phys. Lett. A* , pp 92: 271.

Dubois, D., and H., Prade, 1980, "A Class of Fuzzy Measures Based on Triangle Inequalities"*, Int. J. Gen. Sys. 8.*

Ebel H. and S. Bornholdt, 2002, "Co-evolutionary Games on Networks", *Phys. Rev. E 66.*

Einstein, A., B, Podolsky, and N., Rosen, 1935, "Can Quantum-mechanical Description of Physical Reality be Considered Complete?" *Physical Review*, pp 47:777–780.

Erdös, P., and A., Renyi, 1959, "On Random Graphs", *Publications Mathematicae*, pp 6-290.

Erdös, P., and A., Renyi, 1960, "On the Evolution of Random Graphs", *Publ. Math. Inst. Hung. Acad. Sci.,* pp 5-17.

Everett, H., 1957, "Relative State Formulation of Quantum Mechanics." *Reviews of Modern Physics*, pp 29(3):454–462.

Feynman, R. P., and F. L., Vernon, 1963, "The Theory of a General Quantum System Interactiong with a Linear Dissipative System", *Annals of Physics*, pp 24:118–173.

Flajolet, P., D. E., Knuth, and B., Pittel, 1989, "The First Cycles in an Evolving Graph", *Discrete Mathematics*, pp 75: 167.

Flitney, A.P., and D., Abbott, 2002, "An Introduction to Quantum Game Theory", *The University of Adelaide.*

Garfield, E., 1972, "Citation Analysis as a Tool in Journal Evaluation", *Science*, pp 178: 471.

Garfield, E., 1979, "Citation Indexing: Its Theory and Application in Science", *New York: Wiley.*

Goldberg, E., 1989, "Genetic Algorithms in Search, Optimization, and Machine Learning". *Addison-Wesley.*

Gottesman, D., and I. L., Chuang, 1999, *Nature*, pp 402: 390.

Han, K. H., K. H., Park, C. H., Lee, and J. H., Kim. 2001, "Parallel Quantum-inspired Genetic Algorithm for Combinatorial Optimization Problem" *In Proc. of the 2001 IEEE Congress on Evolutionary Computation.*

Herz, V. M., and J., Theor, 1994, *Biology*, pp 169: 65.

Hey, A. J. G., 1999, *Feynman and Computation*

Hofbauer, J., and K., Sigmund, 1988, "The Theory of Evolution and Dynamical Systems", *Cambridge University Press.*

Holland, J. H., 1975, "Adaptation in Natural and Artificial Systems", *MIT Press, Cambridge.*

Huberman, A., and N. S. Glance, 1993, *Proc. Natl. Acad. Sci.USA* pp 90: 7716.

Huberman, B. A. and L. A., Adamic, 1999, "Growth Dynamics of the World-Wide Web", *Nature*, pp 401: 131.

Huberman, B. A., P. L. T., Pirolli, J. E., Pitkow, and , R. J., Lukose., 1998, "Strong Regularities in World Wide Web Surfing", *Science*, pp 280: 95.

Iqbal, A., 1995, "Studies in the Theory of Quantum Games", *University of Sheffield*, UK.

Janson, S., D. E., Knuth, T., Luczak, and B. Pittel, 1993, "The Birth of the Giant Component", *Random Structures and Algorithms,* pp 4: 233.

Janson, S., T., Luczak, and A., Rucinski, 2000, "Random Graphs", *New York: Wiley & Sons.*

Joos, E., and H. D., Zeh, 1985, "The Emergence of Classical Properties through Interaction with the Environment", *Zeitschrift fur Physik B*, pp 59:223–243.

Kraines, D., and V., Kraines, 1993, "Evolution of Learning among Pavlov Strategies in a Competitive Environment with Noise", *Theory and Decision,* pp 35: 107-150.

Lawrence, S., and C. L., Giles, 1998, "Searching the World Wide Web", *Science*, pp 280: 98.

Lawrence, S., and C. L., Giles, 1999, "Accessibility of Information on the Web", *Nature*, pp 400: 107.

Lloyd, S., 1996, *Science*, pp 273: 1073.

Lotka, A. J., 1926, "The Frequency Distribution of Scientific Productivity", *J. Washington Acad. Sci.*, pp 16: 317.

Mamdani, E. H., and S., Assilian, 1975, "An Experiment in Linguistic Synthesis with A Fuzzy Logic Controller", *International Journal of Man-Machine Studies, Vol. 7, No. 1*, pp. 1-31.

Mitchell, M., 1996, "An Introduction to Genetic Algorithms", *MIT Press*.

Molloy M. and B., Reed, 1995, "A Critical Point for Random Graphs with a Given Degree Sequence", *Random Structures and Algorithms*, pp 6: 161.

Molloy M. and B., Reed, 1998, "The Size of the Giant Component of a Random Graph with a Given Degree Sequence", *Combinatorics, Probability and Computing*, pp 7: 295.

Nemoto K., and M. J., Gagen, 2004, "Cooperative Equilibria in the Finite Iterated Prisoner's Dilemma"*, EconWPA.*

Neumann, V. J., 1955, "Mathematical Foundations of Quantum Mechanics", *Princeton University Press*.

Neumann, V. J., and O., Morgenstern, 1953, "Theory of Games and Economic Behaviour", *Princeton.*

Newman, M. E. J., 2000, "Models of the Small World", J. *Stat. Phys.*, pp 101: 819.

Newman, M. E. J., 2001, "Scientific Collaboration Networks: I. Network Construction and Fundamental Results", *Phys. Rev. E*, pp 64: 016131.

Newman, M. E. J., 2001, "Scientific Collaboration Networks: II. Shortest Paths, Weighted Networks, and Centrality", *Phys. Rev. E*, pp 64: 016132.

Nielsen, M. A., and I. L., Chuang, 2000, "Quantum Computation and Quantum Information", Cambridge University Press.

Nowak, M., and K., Sigmund, 1989, "Game Dynamical Aspects of the Prisoner's Dilemma", *J. Appl. Math. Comp.*, vol. 30, pp. 191-213.

Nowak, M., and K., Sigmund, 1990, "The evolution of Stochastic Strategies in the Prisoner's Dilemma", *Acta Applicandae Mathematicae*, vol. 20, pp. 247-265.

Nowak, M., and R. M., May, 1992, *Nature*, pp 359.

Nowak, M., and R. M., May, 1993, *Int. J. Bifurcation Chaos*, pp 3: 35.

Nowak, M., S., Bonhoeffer, and R. M., May, 1994, *Int. J. Bifurcation Chaos* pp 4: 33.

Nowak, M., S., Bonhoeffer, and R. M., May, 1994, *Proc. Natl. Acad. Sci. USA* pp 91: 4877.

Opatrny, T., and G., Kurizki, 2000, "On the possibility of quantum computation based on photon exchange interactions", *Special focus issue on Experimental Proposals for Quantum Computation 2000 Fort. Phys.,* 48 nos 9–11.

Osborne, M. J., and A., Rubinstein, 1994, "A Course in Game Theory", *MIT Press.*

Paul, A.M.D., 1926, "The Physical Interpretation of the Quantum Dynamics" *Proceedings of the Royal Society of London (A)*, pp 113:621–641.

Poston, T., and I., Stewart, 1978, "Catastrophe Theory and Its Applications", *London, Pitman.*

Rasmusen, E., 1989, "Games and Information", *Blackwell Cambridge MA.*

Roberts, A.J., 2003, "The Prisoners may be in Two Minds", *General Mathematics.*

Schreiber, Z., 1994, "The Nine Lives of Schrödinger's Cat", *Master of Science of the University of London.*

Schumacher, B., 1996, *Phys. Rev. A,* pp 54:2614.

Scott, J., 2000, "Social Network Analysis: A Handbook ", *Sage Publications, London, 2nd Ed.*

Shockley, W., 1957, "On the Statistics of Individual Variations of Productivity in Research Laboratories", *Proceedings of the IRE*, pp 45: 279.

Shor, P. W., 1994, *Proc. 35th Ann. Symp. on the Foundations of Computer Science Los Alamitos CA: IEEE Computer Society Press*, p 124.

Smith, J. M., 1982, "Evolution and the Theory of Games", *Cambridge University Press.*

Smucker, M. D., E. A., Stanley, and D., Ashlock, 1994, "Anaylzing Social Network Structures in the Iterated Prisoner's Dilemma with Choice and Refusal", *University of Wisconsin-Madison, Department of Computer Sciences.*

Strogatz, S. H., 2001, *Nature*, pp 410: 268–276.

Theodore, C. B., 1995, "The Distributed Genetic Algorithm Revisited.", *In Proceedings of the Sixth International Conference on Genetic Algorithms.*

Toffoli, T, and A. J. G., Hey, 1999, *Feynman and Computation*, p 349.

Viola, L., E., Knill, and R., Laflamme, 2001, "Constructing Qubits in Physical Systems", *J. Phys. A*, pp 34: 7067

Wasserman, S., and K., Faust, 1994, "Social Network Analysis", *Cambridge Univ. Press.*

Watts, D. J., 1999, "Small Worlds", *Princeton University Press.*

Watts, D. J., and S. H., Strogatz, 1998, "Collective Dynamics of Small-world Networks", *Nature*, 393: 440.

Wheeler, J. A., and H. Z., Wojciech, 1983, "Quantum Theory and Measurement" *Princeton University Press.*

Wooters, W. K., and W. H., Zurek, 1982, *Nature*, pp 299: 802.

Wright, A. H., 1991, "Genetic Algorithms for Real Parameter Optimization", *Foundations of Genetic Algorithms*, pp 205–218.

Zurek, W. H., 1981, "Pointer Basis of Quantum Apparatus: Into What Mixture does the Wave Packet Collapse?", *Physical Review D*, pp 24(6):1516–1525.

Zurek, W. H., 1982, "Environment-induced Superselection Rules", *Physical Review D*, pp 26(8):1862–1880.

Zurek, W. H., 1986, *Annals of the New York Academy of Science*, pp 480:89.