NATURE INSPIRED COMPUTATIONAL INTELLIGENCE FOR FINANCIAL CONTAGION MODELLING

By

Fang Liu

SUBMITTED IN FULFILLMENT OF THE
REQUIREMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
BRUNEL UNIVERSITY
UXBRIDGE, WEST LONDON, UNITED KINGDOM
22TH FEBRUARY 2014

BRUNEL UNIVERSITY

BRUNEL BUSINESS SCHOOL

The undersigned hereby certify that they have read and recommend to the Brunel Business School for acceptance a thesis entitled "**Financial Contagion Analysis Using Computational Intelligence**" by **Fang Liu** in fulfillment of the requirement for the degree of **Doctor of Philosophy.**

Dated: 22th February 2014

External Examiner: ______________________________

Internal Examiner: ______________________________

First Supervisor: Dr Antoaneta Serguieva

Second Supervisor: Dr Paresh Date

Declaration of Originality

I hereby declare that this thesis is composed entirely by myself. The notions and conclusions included herein originate from my work, if not else acknowledged in the text. The work described in the thesis has not been previously submitted for a degree at this or any other university.

The thesis is completed on 22th February 2014 under a supervised PhD program at Brunel University. Developed measures, techniques and algorithms, as well as empirical results, have been published as follows:

- Chapter 2 in [P1,P2]
- Chapter 3 in [P1]
- Chapter 4 in [P1]
- Chapter 5 in [P3]
- Chapter 6 in [P1,P3]

Statement of Copyright

Table of Contents

List of Figures

List of Tables

List of Acronyms

GARCH	Generalized Autoregressive Conditional Heteroscedasticity
MG	Minority Game
iid	Independent and Identically Distributed
EC	Evolutionary Computation
EP	Evolutionary Programming
GA	Genetic Algorithms
GP	Genetic Programming
IMF	International Monetary Fund
PSO	Particle Swarm Optimization
CSA	Clonal Selection Algorithms

Abstract

Financial contagion refers to a scenario in which small shocks, which initially affect only a few financial institutions or a particular region of the economy, spread to the rest of the financial sector and other countries whose economies were previously healthy. This resembles the "transmission" of a medical disease. Financial contagion happens both at domestic level and international level. At domestic level, usually the failure of a domestic bank or financial intermediary triggers transmission by defaulting on inter-bank liabilities, selling assets in a fire sale, and undermining confidence in similar banks. An example of this phenomenon is the failure of Lehman Brothers and the subsequent turmoil in the US financial markets. International financial contagion happens in both advanced economies and developing economies, and is the transmission of financial crises across financial markets. Within the current globalise financial system, with large volumes of cash flow and cross-regional operations of large banks and hedge funds, financial contagion usually happens simultaneously among both domestic institutions and across countries.

There is no conclusive definition of financial contagion, most research papers study contagion by analyzing the change in the variance-covariance matrix during the period of market turmoil. King and Wadhwani (1990) first test the correlations

between the US, UK and Japan, during the US stock market crash of 1987. Boyer (1997) finds significant increases in correlation during financial crises, and reinforces a definition of financial contagion as a correlation changing during the crash period. Forbes and Rigobon (2002) give a definition of financial contagion. In their work, the term interdependence is used as the alternative to contagion. They claim that for the period they study, there is no contagion but only interdependence. Interdependence leads to common price movements during periods both of stability and turmoil.

In the past two decades, many studies (e.g. Kaminsky et at., 1998; Kaminsky 1999) have developed early warning systems focused on the origins of financial crises rather than on financial contagion. Further authors (e.g. Forbes and Rigobon, 2002; Caporale et al, 2005), on the other hand, have focused on studying contagion or interdependence.

In this thesis, an overall mechanism is proposed that simulates characteristics of propagating crisis through contagion. Within that scope, a new co-evolutionary market model is developed, where some of the technical traders change their behaviour during crisis to transform into herd traders making their decisions based on market sentiment rather than underlying strategies or factors. The thesis focuses on the transformation of market interdependence into contagion and on the contagion effects. The author first build a multi-national platform to allow different type of players to trade implementing their own rules and considering information from the domestic and a foreign market. Traders' strategies and the performance of the simulated domestic market are trained using historical prices on both markets, and optimizing artificial market's parameters through immune - particle swarm optimization techniques (I-PSO). The author also introduces a mechanism contributing to the transformation of technical into herd traders. A generalized auto-regressive conditional heteroscedasticity - copula (GARCH-copula) is further applied to calculate the tail dependence between the affected market and the origin of the crisis, and that parameter is used in the fitness function for selecting the best solutions within the evolving population of possible model parameters, and therefore in the optimization criteria for contagion simulation. The overall model is also applied in predictive mode, where the author optimize in the pre-crisis period using data from the domestic market and the crisis-origin foreign market, and predict in the crisis period using data from the foreign market and predicting the affected domestic market.

Chapter 1 : Introduction

1.1. Background and Motivation

A series of financial crises, such as the Mexican crisis of 1987, the Asian turmoil of 1997, and the Russian instability of 1998, all share a common feature – problems spread from one country to neighbouring countries, and even regionally or globally. The spread is due to the cross-market linkages. If the cross-market linkages stay stable then the crisis is transferred through interdependence, and the recovery follows the recovery of the underlying economic reasons in the country of origin. When the cross-market linkages get destabilized due to the crisis, then the crisis starts "feeding on itself" and the recovery of the underlying economic reason is not sufficient to get control of the crisis; a more comprehensive strategy with international involvement is required. The second type of crisis exhibits the phenomenon called 'financial contagion'.

There is no conclusive definition of financial contagion, most research papers study contagion by analyzing the change in the variance-covariance matrix during the period of market turmoil. King and Wadhwani (1990) first test the correlations between the US, UK and Japan, during the US stock market crash in 1987. Boyer (1997) finds significant increases in correlation during financial crises, and reinforces a definition of financial contagion as a correlation breakdown during the crash period. Forbes and Rigobon (2002) define financial contagion as “a significant increase in cross-market linkages after a shock to a group of countries”. They claim that for the period they study, there is no contagion but only interdependence. Interdependence leads to common price movements during periods both of stability and turmoil.

In the past two decades, many studies (e.g. Kaminsky et al., 1998; Kaminsky 1999) have developed early warning systems focused on the origins of financial crises rather than on financial contagion. Further authors (e.g. Forbes and Rigobon, 2002; Caporale et al, 2005), on the other hand, have focused on studying contagion or interdependence. In this thesis, the author simulates the transmission of financial crises, modelling through computational intelligence the behaviour of market players and their various strategies.

Computational intelligence combines elements of learning, adaptation, and evolution. In this thesis the author choose to use a hybrid computational approach involving artificial immune–particle swarm intelligence, genetic programming, and a mixed-game, to implement the challenging task of developing a system capable of simulating realistic market behaviour and the contagion phenomenon. In the next section, the author will discuss the application of computational intelligence approaches in the area of finance.

1.2. Structure of Thesis

In Chapter 2, the author introduce the origin, development, and applications to finance of different evolutionary computing approaches. The reason for considering these in detail is the conclusion the author reach in the previous section here that evolutionary computing is the most representative computational intelligence area with financial applications. The model the author develops later in the thesis is also based on a hybrid evolutionary approach.

In chapter 3, the author will briefly discuss the correlation coefficient as a measure of dependence between two random variables, and the limitations of this measure. Then, the author will introduce the copula as an alternative measure, together with a discussion of copula types and how their parameters are estimated. The reason for focusing on an effective measure of market interdependence is to be able to introduce that in the objective function of the optimization approach. An effective measure of interdependence also contributes to recognizing the shift towards contagion between markets.

In chapter 4, the author develop a comprehensive market model comprising four types of traders: technical, game, herd, and noise traders, respectively. Then, the author extend this one-market model to an international two-market model, in order to explore how financial contagion happens. This is achieved by evolving the two-market model and observing the interactions between the markets. The two-market model is extendible to a multi-national market, i.e. a multiple-market model.

In chapter 5, the author propose an Immune Particle Swarm Optimization (Immune-PSO) algorithm, which is combined with an Immune Clone Selection algorithm. The reason for developing these algorithms is to improve the optimization technique applied to the two-market (multiple-market) model. Within the new approach, several operators are performed – a clone copy, a clone hyper-mutation and a clone selection - during the evolutionary steps of the model. The author also compare, on a set of test functions, the performance of the new approach with the genetic algorithm used in the previous chapter. Finally, the Immune-PSO is implemented to estimate the parameters of our agent-based multinational market model.

In chapter 6, the author further modifies the co-evolutionary market model by introducing a mechanism allowing for some of the technical and game traders to transform their behaviour during crisis periods. Thus, during crises they will make their decisions based on market sentiment rather than following their usual trading strategies.

In chapter 7, the author draw up and summarize the conclusions of this study. The main contributions are highlighted, and directions for further research are proposed.

.

Chapter 2 : Nature-inspired Computational Approaches: Origin, Development and Applications to Finance

Nature-inspired Computing is a collection of nature-inspired analytical and optimization tools. The systems resulting from the implementation of these techniques are better able to cope with complex problems.

Evolutionary computing (EC) is the most representative computational intelligence area with financial applications. The model the author develop later in the thesis is also based on a hybrid evolutionary approach. According to Isasi et al. (2007), EC is divided into four main areas: evolutionary programming (EP), evolutionary strategies (ES), genetic algorithms (GA) and genetic programming (GP). They have been developed independently, where EP focuses on optimizing continuous functions without recombination, ES focus on optimizing continuous functions with recombination, GA focus on optimizing general combinatorial problems, and GP evolves programs, (Forrest 1993, Holland 1975, Michalewicz 1996).

Along with the earlier established areas of EP, ES, GA and GP, the author also review here the recently developed artificial immune systems (AIS) and swarm intelligence (SI) approaches and their applications to finance. These latter techniques are also nature-inspired, and particularly relevant to the algorithm the author design and implement. Artificial Immune is adaptive systems, which are inspired by theoretical immunology and observed immune functions, principles and models, and then applied to problem solving. The techniques are inspired by specific immunological theories that explain the function and behaviour of the mammalian adaptive. (Castro et al., 2002) Swarm intelligence describes the collective behaviour of decentralized, self-organized systems, where interactions between agents lead to the emergence of "intelligent" global behaviour, (Beni and Wang, 1989). Examples of SI include ant colony optimization (ACO) and particle swarm optimization (PSO), (Dorigo, 1992; Eberhart& Kennedy, 1995).

All of these approaches are widely used in various areas such as chemical industry, power system, machine design, robotic design, signal processing, biology, operational research, system identification, optimal control, learning, prediction, and fault diagnosis. Each of them has been also successfully applied to the area of finance and economics. The author will focus on four techniques in this chapter, directly relevant to our research, namely, GA, GP, PSO and AIS.

2.0. Overview of Applications of Computational Intelligence Approaches to Financial Problems

Financial markets, as highly nonlinear dynamic systems, are affected by many fundamental and sentiment factors, including interest rates, inflation rates, and political issues. Having the interdependency between the factors, it is difficult to model stock price movements with traditional methods. Computational intelligence approaches (CI), as a more powerful framework for dealing with complex problems, are being introduced to financial analysis. Currently, applications of CI are increasingly covering various aspects of finance and economics. The range of techniques includes the main CI areas - artificial neural networks (ANN), evolutionary computing (EC), and fuzzy logic -as well as the more specific swarm intelligence technologies, support vector machines, and simulated annealing (SA), to name a few.

Evolutionary approaches have been applied to problems in finance for a considerable time now. Bauer (1994) uses genetic algorithms (GA) based intelligent systems to find out effective market timing strategies. Allen and Karjalainen (1995) use GAs to identify profitable trading rules for the S&P 500 index using daily prices from 1928 to 1995. Chen and Yeh (1996) use genetic programming (GP) to prove the efficient market hypothesis, namely, to formalize the notion that stock price is unpredictable. Mahfoud and Mani (1996) introduce a GA-based system to forecast share price. Neely et al. (1997) apply a GP-based model to predict foreign exchange rates and the result is reported as a success. Li and Tsang (1999) develop a financial genetic programming technique (FGP), and the result indicates that FGP outperforms random walk.

Regarding the application of evolutionary strategies (ES), Streichert (2002) uses ES to discover technical trading rules. In recent years, evolutionary strategies have been used, in combination with other approaches. Hong (2007) proposes an integrated model of support vector regression (SVR) and ES. With ES, the problem of determining the parameters for SVR is resolved. The model overcomes the disadvantage of traditional time series forecasting, which has problems capturing the nonlinear patterns. Experimental results show that the proposed model outperforms other approaches that have been applied for exchange rate forecasting. Mora (2008) uses a self-organizing map (SOM) to reduce the dimensions of the prediction problem. In this model, the capability of GP is merged with ES, to generate classification trees. The result indicates that the model outperforms an evolutionary ANN method. Fan et al (2008) propose a real-valued quantum-inspired evolutionary strategy (QIES). This model is similar to estimation of distribution algorithms (EDAs). They also compare the results with those from a canonical GA. The results are reported to be robust and also suggest a good potential for high-dimensional optimization.

ANNs, nearly as popular as EC, are also powerful techniques to employ and are now widely used in the area of finance. Kimoto and Asakawa (1990) produce a timing prediction system for sale and purchase, based on modular neural networks for stocks on the Tokyo stock exchange. The authors claim that their model achieves accurate predictions. Yoon and Swales (1991) examine the capability of ANN, and compare it with other techniques such as multiple discriminate analyses. Results indicate that ANN enhances investors' forecasting ability. Aiken and Bsat (1994) apply ANNs to the area of real estate. Yao and Poh (1995) use back-propagation neural networks to predict movements in the Kuala Lumpur Stock Exchange (KLSE), and report a significant profit. Harrald and Kamstra (1997) conduct GP experiments to evolve ANNs to forecast stock price volatility. Lee (2001) adopts ANN to identify stock price trends. Chang (2004) proposes an integrated ANN and auto-regressive integrated moving average model to forecast the future fluctuation of the stock market index. The author concludes, however, that due to the noise and complexity, and the dimensionality of stock price, all those models have their inherent limitations. In other words, the input variables interfere with each other; hence the results are not convincing.

Furthermore, fuzzy logic is also popular in the financial arena but is often combined with other techniques. Wang et al (1998) propose an integrated model which combines GA with fuzzy knowledge. This model can integrate multiple fuzzy rule sets and their membership functions. Results show that their approach outperforms every individual knowledge base. Larsen and Yager (2000) present a hybrid soft computing technique for automated stock market forecasting and trend analysis. Firstly, they use principal component analysis to initiate the input data. Then, they use a neuro-fuzzy system to analyze the trend of stock prices. Lee (2001) suggests a Takagi-Sugeno-Kang (TSK) fuzzy rule-based system. In this model, a technical index is used as the input variable. The result, which is a linear combination of input variables, is tested on Taiwanese electronic shares from the Taiwan Stock Exchange (TSE). It is reported that the model can successfully forecast the price variation for stocks. Serguieva and Kalganova (2002) build a fuzzu-neuro-evolutionary classifier of risky investments. Due to the complexity of the problem they tackle, an evolutionary strategy is applied using bidirectional incremental evolution (BIE) to evolve a fuzzy network. The model is tested with data on UK companies traded on the London Stock Exchange. Kuzemin and Lyashenko (2007) propose a fuzzy set theory approach which is used as the basis for analysis of financial flows in the economic security system. Qin and Li (2008) formulate a

European option-pricing formula for fuzzy financial markets, and discuss some of their mathematical properties.

From the above review of literature, the author can see that application of computational intelligence approaches in the financial area mainly include processing and analyzing financial data (financial forecasting), pricing complex financial products (securities pricing), and analyzing market behaviour (trading strategy generation). Evolutionary computing is the most representative of the computational intelligence approaches, and can successfully provide agents with learning capability. In the next chapter, the author will focus on the application of evolutionary computing in the area of finance.

2.1. Genetic Algorithms

Holland called his method a genetic algorithm (Holland, 1975). Genetic algorithms are categorized as global search heuristics. They are implemented in computer simulations where a population of abstract representations, called chromosomes, of candidate solutions to an optimization problem evolves towards better solutions. Traditionally, solutions are represented binary as strings of 0s and 1s, but other encodings are also possible. (Vajedia 2003) In each generation, the fitness of every individual in the population is evaluated and multiple individuals are selected from the current population based on their fitness. The best chromosomes are picked to be parents for the next generation. Then new child chromosomes are generated, e.g. representing new trading strategies, by crossover (setting points and exchanging the genes in the chromosomes between the points) and mutation (randomly changing one gene at a given point). Finally, the poorest strategies are replaced by the new child strategies. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. The process is illustrated in Figure 2-1 below.

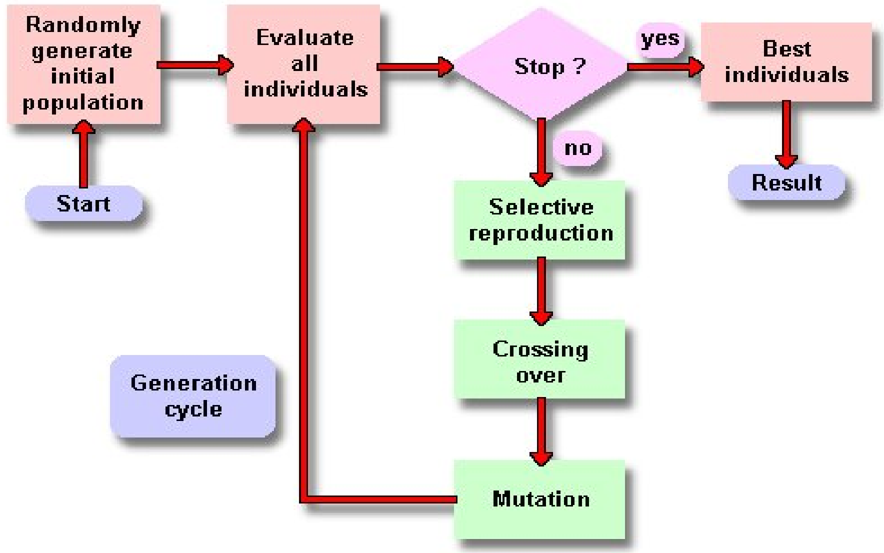

Figure 2-1: GA flowchart

2.1.1. Population Representation and Initialization

GAs operates on a number of potential solutions, called a population, consisting of some encoding of the parameter set simultaneously. Typically, a population is composed of between 30 and 100 individuals, although, a variant called the micro GA uses very small populations, less than 10 individuals, with a restrictive reproduction and replacement strategy in an attempt to reach real-time execution (Karr, 1991). In this thesis, micro GA is used due to the complexity of our agent based model .The most commonly used representation of chromosomes in the GA is that of the single-level binary string. Here, each decision variable in the parameter set is encoded as a binary string and these are concatenated to form a chromosome. Whilst binary-coded

GAs are most commonly used, there is an increasing interest in alternative encoding strategies, such as integer and real-valued representations.

The use of real-valued genes in GAs is claimed by Wright (1991) to offer a number of advantages in numerical function optimization over binary encodings. Efficiency of the GA is increased as there is no need to convert chromosomes to phenotypes before each function evaluation; less memory is required as efficient floating-point internal computer representations can be used directly; there is no loss in precision by discretisation to binary or other values; and there is greater freedom to use different genetic operators. Having decided on the representation, the first step in the SGA is to create an initial population. This is usually achieved by generating the required number of individuals using a random number generator that uniformly distributes numbers in the desired range. For example, with a binary population of Nind individuals whose chromosomes are Lind bits long, Nind× Lind random numbers uniformly distributed from the set {0, 1} would be produced.

2.1.2. Objective and Fitness Functions

The objective function is used to provide a measure of how individuals have performed in the problem domain. In the case of a minimization problem, the fittest individuals will have the lowest numerical value of the associated objective function.

This raw measure of fitness is usually only used as an intermediate stage in determining the relative performance of individuals in a GA. Another function, the fitness function, is normally used to transform the objective function value into a measure of relative fitness (De Jong, 1975), thus

$$F(x) = g(f(x)) \quad (2.1)$$

where f is the objective function, g transforms the value of the objective function to a non-negative number and F is the resulting relative fitness.

2.1.3. Fitness Calculation

(i). Proportional fitness assignment

A commonly used transformation is that of proportional fitness assignment. The individual fitness, $F(x_i)$ of each individual is computed as the individual's raw performance, $f(x_i)$, relative to the whole population, i.e.

$$F(x_i) = \frac{f(x_i)}{\sum_{i=1}^{N} f(x_i)} \quad , \quad (2.2)$$

where Nis the population size and xi is the phenotypic value of individual i. Whilst this fitness assignment ensures that each individual has a probability of reproducing according to its relative fitness, it fails to account for negative objective function values.

(ii). Rank-based fitness assignment

In rank-based fitness assignment, the population is sorted according to the objective values. The fitness assigned to each individual depends only on its position in the individuals rank and not on the actual objective value. Rank-based fitness assignment overcomes the scaling problems of the proportional fitness assignment. The reproductive range is limited, so that no individuals generate an excessive number of offspring. Ranking introduces a uniform scaling across the population and provides a simple and effective way of controlling selective pressure. Rank-based fitness assignment behaves in a more robust manner than proportional fitness assignment and, thus, is the method of choice. (Bäck, T. &Hoffmeister, 1991)

(iii). Linear ranking

Consider Nind as the number of individuals in the population, Pos as the position of an individual in this population (the least fit individual has Pos=1, the fittest individual Pos=Nind) and SPas the selective pressure. The fitness value for an individual is calculated as:

$$Fitness(pos) = 2 - SP + 2 * (SP - 1) * \frac{(Pos - 1)}{(Nind - 1)} \quad (2.3)$$

Linear ranking allows values of selective pressure in [1.0, 2.0].

Riechmann (2000) linked the theory of genetic algorithm learning to evolutionary game theory and showed that economic learning via genetic algorithms can be described as a specific form of an evolutionary game. In that paper the fitness is defined as:

$$\text{Fitness} = \text{agent's quantity} \cdot (\text{price-unit costs}) \qquad (2.4)$$

The quantity the agent supplies reflects their own strategy, the market price reflects the state of the whole population, which means, given total demand, it reflects aggregate supply, i.e. the sum of all individual supply strategies. The total supply has an important influence on each agent's fitness.

2.1.4. Selection

Selection is the process of determining the number of times, or trials, a particular individual are chosen for reproduction and, thus, the number of offspring that an individual will produce. (Baker, 1987)

(i). Roulette wheel selection

The simplest selection scheme is roulette-wheel selection, also called stochastic sampling with replacement (Baker, 1987). This is a stochastic algorithm and involves the following technique. The individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness. A random number is generated and the individual whose segment spans the random number is selected. The process is repeated until the desired number of individuals is obtained (called mating population). This technique is analogous to a roulette wheel with each slice proportional in size to the fitness.

Table 2-1 shows the selection probability for 11 individuals, linear ranking and selective pressure together with the fitness value. Individual 1 is the fittest individual and occupies the largest interval, whereas individual 10 as the second least fit individual has the smallest interval on the line. Individual 11, the least fit interval, has a fitness value of 0 and get no chance for reproduction.

Table 2-1: Selection probability and fitness value

Number of individual	1	2	3	4	5	6	7	8	9	10	11
fitness value	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2	0.0
selection probability	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03	0.02	0.0

For selecting the mating population, the appropriate number of uniformly distributed random numbers (uniformly distributed between 0.0 and 1.0) is independently generated. For example, for a sample of six random numbers:

0.81, 0.32, 0.96, 0.01, 0.65, 0.42,

Figure 2-2 shows the selection process of the individuals for the example in Table together with the above sample trials.

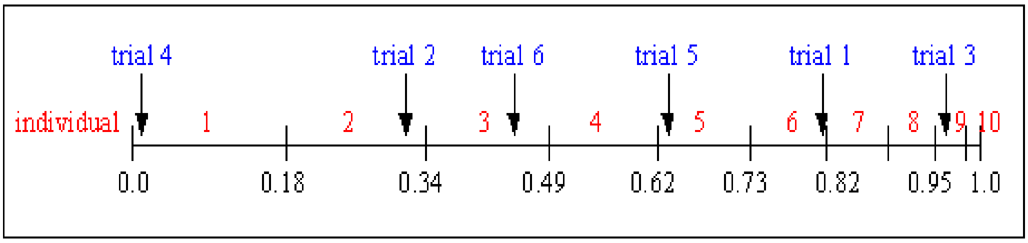

Figure 2-2: Roulette-wheel selection

After the selection, the mating population consists of the following individuals:

1, 2, 3, 5, 6, 9.

The roulette-wheel selection algorithm provides a zero bias but does not guarantee minimum spread.

(ii). Stochastic universal selection

Stochastic universal sampling (Baker, 1987) provides zero bias and minimum spread. The individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness exactly as in roulette-wheel selection. Here equally spaced pointers are placed over the line as many as there are individuals to be selected. Consider NPointer as the number of individuals to be selected, then the distance between the pointers is1/NPointer and the position of the first pointer is given by a randomly generated number in the range [0, 1/NPointer].

Thus for 6 individuals to be selected, the distance between the pointers is 1/6=0.167. Figure 2-3 shows the selection for the above example, where 1 random number in the range [0, 0.167] is used:

:

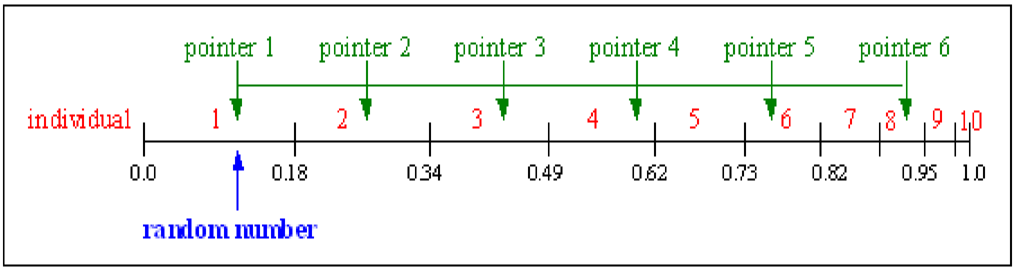

Figure 2-3: Stochastic universal sampling

After the selection, the mating population consists of the following individuals:

1, 2, 3, 4, 6, 8.

Stochastic universal sampling ensures a selection of offspring which is closer to what is deserved then roulette wheel selection.

2.1.5. Crossover

The basic operator for producing new chromosomes in GA is that of crossover. Like its counterpart in nature, crossover produces new individuals that have some parts of both parent's genetic material.

(i).Single-point crossover

The simplest form of crossover is that of single-point crossover, see below:

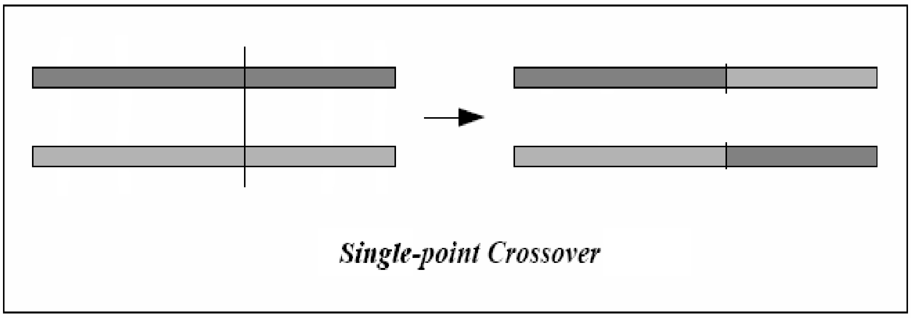

Figure 2-4: Single-point crossover

(ii). Multi-point crossover

For multi-point crossover, the bits between successive crossover points are exchanged between the two parents to produce two new offspring. The section between the first allele position and the first crossover point is not exchanged between individuals. This process is illustrated in Figure 2-5 bellow:

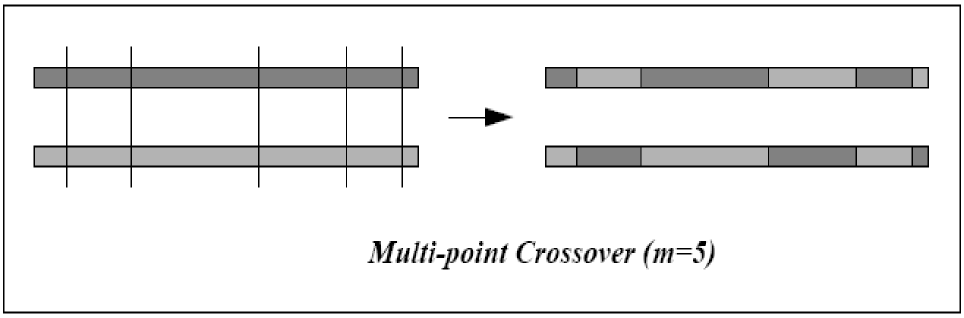

Figure 2-5: Multi-point Crossover

The idea behind multi-point, and indeed many of the variations on the crossover operator, is that the parts of the chromosome representation that contribute most to the performance of a particular individual may not necessarily be contained in adjacent substrings (Booker, 1987). Further, the disruptive nature of multi-point crossover appears to encourage the exploration of the search space, rather than favoring the convergence to highly fit individuals early in the search, thus making the search more robust (Spears & De Jong, 1991).

(iii). Uniform Crossover

Single and multi-point crossover defines cross points as places between loci where a chromosome can be split. Uniform crossover (Syswerda, 1989) generalises this scheme to make every locus a potential crossover point. A crossover mask, the same length as the chromosome structures is created at random and the parity of the bits in the mask indicates which parent will supply the offspring with which bits. Consider the following two parents, crossover mask and resulting offspring:

P1 = 1 0 1 1 0 0 0 1 1 1
P2 = 0 0 0 1 1 1 1 0 0 0
Mask = 0 0 1 1 0 0 1 1 0 0
O1 = 0 0 1 1 1 1 0 1 0 0
O2 = 1 0 0 1 0 0 1 0 1 1

Here, the first offspring, O1, is produced by taking the bit from P1 if the corresponding mask bit is 1 or the bit from P2 if the corresponding mask bit is 0. Offspring O2 is created using the inverse of the mask or, equivalently, swapping P1 and P2. Uniform crossover, like multi-point crossover, has been claimed to reduce the bias associated with the length of the binary representation used and the particular coding for a given parameter set.

(iv). Intermediate Recombination

Given a real-valued encoding of the chromosome structure, intermediate recombination is a method of producing new phenotypes around and between the values of the parents' phenotypes (Mühlenbein & Schlierkamp-Voosen, 1993).

Offspring are produced according to the rule, where α is a scaling factor chosen uniformly at random over some interval, typically [-0.25, 1.25] and P1 and P2 are the parent chromosomes. Each variable in the offspring is the result of combining the variables in the parents according to the following expression with a new α chosen for each pair of parent genes.

$$O1 = P1 \times \alpha\,(P2 - P1) \qquad (2.5)$$

(v). Linear Order Crossover

Linear Order Crossover was first proposed by Falkenauer & Bouffouix (1991). It is implemented as follows:

Step 1: Randomly select a subsequence of genes from one of the two parent chromosomes and copy it into a new offspring maintaining the position of the subsequence.

Step 2: Cross out the genes in the selected subsequence of Step 1 from the second parent. Then place the remaining genes of the second parent from left to right in the child's chromosome around the already inserted subsequence. This completes one offspring.

Step 3: Repeat Steps1and2, but reverse the roles of the two parents.

Matta (2009) uses GA to solve multiprocessor open shop scheduling problem which is categorized as a hard combinatorial optimization problem. In that paper he uses Linear Order Crossover.

Parent 1:	[(1,1),	(3,2),	(5,2),	(4,1),	(5,3)	(4,2),	(2,3),	(2,1),	(1,3),	(1,2),	(3,3),	(3,1),	(5,1),	(2,2),	(4,3)]
Parent 2:	[(3,1),	(3,2),	(1,2),	(4,1),	(3,3)	(4,2),	(5,3),	(2,2),	(1,3),	(2,1),	(2,3),	(1,1),	(4,3),	(5,2),	(5,1)]

Figure 2-6: Example of two parent chromosomes

Parent 1:	[(1,1),	(3,2),	(5,2),	(4,1),	(5,3)	(4,2),	(2,3),	(2,1),	(1,3),	(1,2),	(3,3),	(3,1),	(5,1),	(2,2),	(4,3)]
Parent 2:	[(3,1),	(3,2),						(1,2),	(3,3)	(2,2),	(1,3),	(2,1),	(1,1),	(4,3),	(5,1)]
Offspring 1	[(3,1),	(3,2),	(5,2),	(4,1),	(5,3)	(4,2),	(2,3),	(1,2),	(3,3)	(2,2),	(1,3),	(2,1),	(1,1),	(4,3),	(5,1)]

Figure 2-7: Illustrative example of crossover

Consider the two parent chromosomes shown in Figure 2-6. The sub-sequence from (5, 2) to (2, 3) of parent1 was randomly selected to be copied directly into the offspring. Figure 2-7 illustrates the creation of an offspring. The selected subsequence of parent1 is directly placed into the same position in the offspring and the other genes of parent2 are placed around this subsequence, forming a feasible offspring. This process is repeated with the parent roles reversed to create a second offspring.

2.1.6. Mutation

In natural evolution, mutation is a random process where one allele of a gene is replaced by another to produce a new genetic structure. In GAs, mutation is randomly applied with low probability, typically in the range 0.001 to0.01, and modifies elements in the chromosomes. Usually considered as a background operator, the role of mutation is often seen as providing a guarantee that the probability of searching any given string will never be zero and acting as a safety net to recover good genetic material that may be lost through the action of selection and crossover (Goldberg, 1989). The effect of mutation on a binary string is illustrated bellow:

0001 100000

0000 100000

Figure 2-8: Single point mutation

Many variations on the mutation operator have been proposed. For example, biasing the mutation towards individuals with lower fitness values to increase the exploration in the search without losing information from the fitter individuals (Davis, 1989) or parameterising the mutation such that the mutation rate decreases with the population convergence (Fogarty,1989). Mühlenbein & Schlierkamp-Voosenhas (1993) introduced a mutation operator for the real-coded GA that uses a non-linear term for the distribution of the range of mutation applied to gene values. It is claimed that by biasing mutation towards smaller changes in gene values, mutation can be used in conjunction with recombination as a foreground search process.

2.1.7. Termination of GA

Because the GA is a stochastic search method, it is difficult to formally specify convergence criteria. As the fitness of a population may remain static for a number of generations before a superior individual is found, the application of conventional termination criteria becomes problematic. A common practice is to terminate the GA after a pre-specified number of generations and then test the quality of the best members of the population against the problem definition. If no acceptable solutions are found, the GA may be restarted or a fresh search initiated.

2.1.8. Applications

Genetic algorithms have been widely applied in various areas. For example, Charbonneau (1995) suggests that GA are useful for problems in astrophysics and applies them to three specific problems: fitting the rotation curve of a galaxy based on observed rotational velocities of its components, determining the pulsation period of a variable star based on time-series data, and deriving the critical parameters in a magneto hydrodynamic model of the solar wind. Obayashi et al. (2000) uses a multiple-objective genetic algorithm to design the wing shape for a supersonic aircraft.

Sato et al. (2002) use genetic algorithms to design a concert hall with optimal acoustic properties, maximizing the sound quality for the audience, for the conductor, and for the musicians on stage. The authors state that these solutions have proportions similar to Vienna's Grosser Musikvereinsaal, whose acoustic properties are considered to be one of the best in the world.

GA hasfurther been applied to financial problems. LeBaron (1999) proves that GA is a powerful method to locate improvement in complicated higher-dimension space, after a long discussion of the time series properties of an artificial stock market. Allen et al. (1999) use GA to learn technical trading rules for the S&P 500 index using daily price data from 1928-1995; the rules are able to identify the index when daily returns are positive and volatility is low.Mahfoud and Mani (1996) use a genetic algorithm to predict the future performance of 1600 publicly traded stocks. In their paper, GA are used to evolve a set of if/then rules to classify each stock and to provide, as output, both a recommendation on what to do with regards to that stock (buy, sell, or no prediction) and a numerical forecast of the relative return. The results are compared to those of an established neural net-based system, used to forecast stock prices and manage portfolios for three years. Overall, the genetic algorithm significantly outperforms the neural network. Similar success was achieved by Andreou, Georgopoulos and Likothanassis(2002), who used hybrid genetic algorithms to evolve neural networks that predicted the exchange rates of foreign currencies up to one month ahead. GA clearly outperforms the other methods. More recent applications of GA include Chun-Teck Lye (2011) which presents a hybrid approach by associating GA and Sequential Quadratic Programming (SQP) to improve the Stutzer Index optimization.

2.2. Genetic Programming

Genetic programming is another of the EC approaches and focuses on finding computer programs that perform a user-defined task (Banzhaf 1998). The first statement of modern "tree-based" GP, that is procedural languages organized in tree-based structures and operated on by suitably defined genetic operators, is given by Michael Cramer (1985). That work is later greatly expanded by John Koza, a main proponent of GP, who has pioneered the application of genetic programming in various complex optimization and search problems (Koza, 1990). GP is a specialization of GA where each individual is a computer program. The fitness landscape of GP is determined by a program’s ability to perform a given computational task. The population of computer programs is optimized according to fitness. Thus GP evolves computer programs represented in memory as tree structures. Trees can be easily evaluated in a recursive manner. Every tree node has an operator function and every terminal node has an operand, making mathematical expressions easy to evolve and evaluate.

2.2.1. Solution Initialization

The innovation of GP lies in the variable sized solution representation which requires efficient initial population construction, and this feature makes it different from other evolutionary algorithms. Individuals are represented as trees constructed randomly from a primitive set. This primitive set contains functions and terminals. A tree's internal nodes are selected from the functions and leaf nodes are selected from the terminals. GP allows variety in composition of solution structures using same primitive set. Initialization plays an important role in success of an evolutionary algorithm. A poor initial population can cause any good algorithm to get stuck in local optima. On the other hand a good initialization can make most of the algorithms work sufficiently well. There are a few initialization techniques popular in tree based GP.

(i). Full method

The full method is very similar to the grow method except the terminals are guaranteed to be of a certain depth. This guarantee does not specify the number of nodes in an individual. This method requires a final depth, d.

1. Every node, starting from the root, with a depth less than d, is made of a

randomly selected function. If the node has a depth equal to d, the node is made of a randomly selected terminal.

2. All functions have a number (equal to the arity of the function) of child nodes appended, and the algorithm starts again. Thus, only if d is specified as one, could this method produce a one-node tree.

(ii). Grow method

With this technique the entire population is created by using the grow method which creates one individual at a time. An individual created with this method may be a tree of any depth up to a specified maximum, m.

1. Starting from the root of the tree every node is randomly chosen as either a function or terminal.
2. If the node is a terminal, a random terminal is chosen.
3. If the node is a function, a random function is chosen, and that node is given a number of children equal to the arity (number of arguments) of the function. For every one of the function's children the algorithm starts again, unless the child is at depth m, in which case the child is made of a randomly selected terminal. This method does not guarantee individuals of a certain depth (although they will be no deeper than m). Instead it provides a range of structures throughout the population.

2.2.2. Selection

The evolutionary operators are applied on individuals particularly selected for that operation. The individuals are selected using a particular selection mechanism. Two of such mechanisms are defined as follows.

(i).Tournament selection

In this type of selection, a tournament is conducted among few individuals chosen randomly from the population. The winner or best member is selected as a result of a tournament. The tournament size determines how many random members are selected for the tournament. Tournament size determines the selective pressure; large tournament size favours fitter solutions for selection. (Fang &Li, 2010)

(ii). Fitness proportionate selection

All the trees have probability of selection based upon their fitness. The probability of selection for a population of size 'N' is calculated as

$$F(x_i) = \frac{f(x_i)}{\sum_{i=1}^{N} f(x_i)} \qquad (2.6)$$

This is also called Roulette Wheel Selection mechanism. Several other selection mechanisms also exist in the literature like Rank Based Selection and Stochastic Universal Sampling. (Koza, 1997)

2.2.3. Crossover

Crossover operator works by selecting two parents from the population. Two random subtrees are selected from each parent and swapped to create children. Advancements have been made to pure random crossover operator in order to make it more efficient and propagate good building blocks among generations. The information regarding size (Langdon, 2000), depth (Ito et al., 1998) or homogeneity (Langdon, 2000) of subtrees is also exploited while performing this operation.

2.2.4. Mutation

Mutation used in GP is of three types. In point mutation, a single node in a parent tree is selected and replaced with a random node of the same type. E.g. a function node is replaced by a function node of the same arity, and a terminal node is replaced by a randomly selected terminal node. (Koza, 1997)

(i)Shrink mutation selects a node randomly and the subtree rooted at that node is replaced by a single terminal node. (Koza, 1997)

(ii)Grow mutation selects a random node and a randomly generated subtree is replaced by the subtree rooted at that node. (Koza, 1997)

2.2.5. Reproduction

In this operator an individual is selected and copied directly to the new generation without any changes or modifications to it. Koza (1997) allowed 10% of the population to reproduce. If the fitness test does not change, reproduction can have a significant effect on the total time required for GP because a reproduced individual will have an identical fitness score to that of its parent. Thus a reproduced individual does not need to be tested, as the result is already known. For Koza, this represented a 10% reduction in the required time to fitness test a population. However, a fitness test that has a random component, which is effectively a test that does not initialise to exactly the same starting scenario, would not apply for this increase in efficiency. The selection of an individual to undergo reproduction is the responsibility of the selection function.

2.2.6. Solution Fitness

Fitness is the performance of an individual corresponding to the problem it is aimed to solve. It tells which elements or the regions of the search space are good. The fitness measure steers the evolutionary process towards better approximate solutions to the problem. Fitness of individuals in a population can be measured in many ways. It can be measure of error between the original and desired output of a solution. It can be compliance of the structure to the task based on a user specified criteria. The difference between fitness evaluation in GP and other evolutionary algorithms is that each individual of GP is a program which needs recursive execution of the nodes of the tree in a precise manner. This adds an overhead to the algorithm, increasing its evolution time and required computational sources.

2.2.7. Applications

In 1990s, GP was mainly used to solve relatively simple problems because it is very computationally intensive. Recently it has produced many novel and outstanding results in areas such as quantum computing, electronic design, game playing, sorting, and searching, due to improvements in GP technology and the exponential growth in CPU power. For example, it has been used for novel designs such as patented antenna designs (Lohn et al., 2005), patented analogue electronic circuits (Koza et al., 2004), and a small molecule design (Nachbar, 2000). It is also being used commercially to characterize dynamic processes such as chemical processes (Hinchliffe et al., 1999), and image processing (Zhang, 2007). Further, Kishore et al. (2000) explore the feasibility of applying genetic GP to multi-category pattern classification problem, for the first time. GP can discover relationships among observed data and express them mathematically. In their paper, a methodology for GP-based n-class pattern classification is developed, and the reported results indicate a very good performance.

Genetic programming also has a rich history of applications in the area of finance. Chen and Yeh (1996) use GP to prove the efficient market hypothesis (EMH), namely, to formalize the notion that stock price is unpredictable. Chen (2000) proposes a new GP-based architecture to study artificial stock markets. In his model, a new concept, “business school”, is introduced. In essence, business school is a procedure to map the phenotype and genotype, and in the author’s words, “school is for discovering the secret of success”. Furthermore, traders’ search behaviour is also considered, and the result indicates that the return series is independently and identically distributed (iid), conforming to EMH. However, the authors claim that many of their traders are able quite often to find useful signals from the business school. Wilson and Banzhaf (2009) compare two GP approaches: a co-evolutionary genetic programming approach (PAM DGP) and a standard linear genetic programming (LGP), implemented for trading of stocks across market sectors. Both implementations are found to be impressively robust to market fluctuations while reacting efficiently to opportunities for profit, where PAM DGP proved slightly more reactive to market changes than LGP.

Furthermore, Tsang et al. (2009) develop an artificial financial market and use it to model stock markets’ behaviour. The model introduces technical, fundamental and noise traders. Technical traders are sophisticated GP-based agents that co-evolve by forecasting investment opportunities using technical analysis. By identifying the statistical properties of price series and introducing the “red queen principle” in evolution, they demonstrate that GP could play a key role in studying stock markets. Almanza et al. (2007) use a GP based repository method (RM) to find out significant movements in financial stock prices. Results show that the contribution of GP processing is very valuable to the performance of RM. Markose et al. (2001) develop and implement a financial genetic programming model (FGP) on intraday tick data for stock index options and futures arbitrage. This model is suitable for online trading when profitable window arbitrage opportunities exist, which range from one to ten minutes. This application indicates that FGP, in its interactive capacity, allows experts to channel their knowledge into machine discovery.

2.3. Particle Swarm Optimization

Particle swarm optimization is a population-based stochastic optimization technique developed by Eberhart and Kennedy in 1995, and inspired the by social behaviour of bird flocking or fish schooling. (Eberhart& Kennedy, 1995) PSO shares similarities with evolutionary techniques such as GA. The system is initialized with a population of random solutions, and searches for optima by updating successive generations. However unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles (Jaco&Schutte, 2005). Figure 2-9 below presents the flowchart . pseudo code for the overall PSO algorithm see appendix A

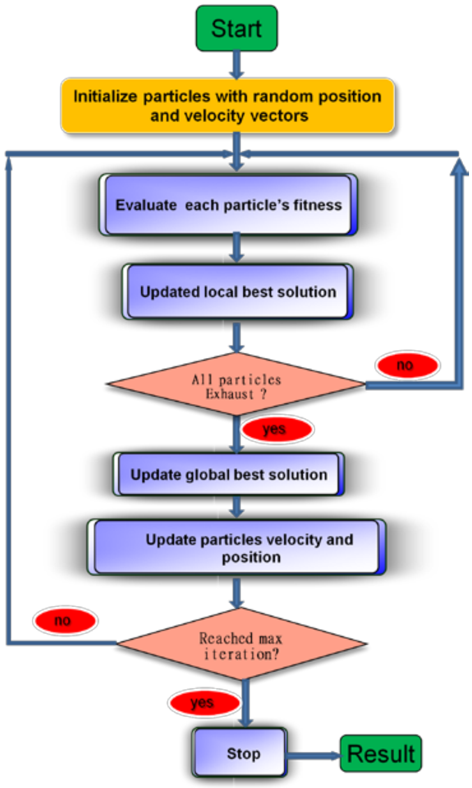

Figure 2-9: PSO flowchart

Each particle keeps track of its coordinates in the problem space, which are associated with the best solution or fitness it has achieved so far, and the fitness value is also stored and called p_{best} (particle best). Another "best" value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle within the neighbours of the particle, and this location is called l_{best} (local best). When a particle takes all the population as its topological neighbours, the best value is global and is called g_{best} (global best). The particle swarm optimization concept consists of, at each time step, changing the velocity of or accelerating randomly each particle toward its p_{best} and l_{best} locations, with separate random numbers being generated for acceleration toward the p_{best} and l_{best} locations (Jaco&Schutte, 2005).

Standard Particle Swarm Optimization (SPSO) achieves optimization by means of cooperation and competition between individual members of the population. Each particle represents a possible solution to the problem. SPSO starts by initializing a group of solutions, and then finds the optimum solution through iteration. Each particle i updates itself by tracing two “best values”: one is the best solution pbest found by the particle itself and denoted as p_i , and the other is the best solution gbest found by its neighbours and denoted as p_g . This process could be described mathematically as follows. In an n dimensional searching space, a population contains m particles, $X = \{x_1, x_2. \cdots x_m\}$, where the position of particlei is $x_i = \{x_{i1}, x_{i2}. \cdots x_{in}\}$and its velocity is$v_i = \{v_{i1}, v_{i2}. \cdots v_{in}\}$, while the local best solution is $p_i = \{p_{i1}, p_{i2}, \cdots p_{in}\}$ and the global best solution is $p_g = \{p_{g1}, p_{g2}, \cdots p_{gn}\}$. A particle's velocity and position are updated through the following formulas:

$$v_{id}^{j+1} = wv_{id}^{j} + c_1 R_1 (p_{id}^{j} - x_{id}^{j}) + c_2 R_2 (p_{gd}^{j} - x_{gd}^{j}), \quad (2.7)$$

$$x_{id}^{j+1} = x_{id}^{j} + v_{id}^{j+1} \quad , \quad (2.8)$$

where $d = 1, 2 \ldots . n$; $i = 1, 2 \ldots . m$. Here n is the dimension of the search space, m is the population size, j is the current generation, while c_1, c_2 are the acceleration constants and R_1 , R_2 are uniformly distributed random numbers in the range from 0 to 1. Also w is the weight given to the extent to which the previous velocity affects the current velocity. The velocity is normally restricted within the interval$[-v_{max} \, , v_{max}]$, and $v_{max} = k \times x_{max}$ where $0.1 \leq k \leq 1$.

2.3.1. Algorithms Refinement

Clerc and Kennedy (2002) apply a constriction factor, v, to the new velocity

$$v_{id}^{j+1} = \chi\{v_{id}^{j} + c_1 R_1 (p_{id}^{j} - x_{id}^{j}) + c_2 R_2 (p_{gd}^{j} - x_{gd}^{j})\} \quad , \quad (2.9)$$

$$\chi = \frac{2}{|2 - c - \sqrt{c^2 - 4c}|} \text{ where } c = c_1 + c_2,\ c > 4 \quad (2.10)$$

With this formulation, the velocity limit,v_{max} , is no longer necessary. A choice need not be made between constriction and inertia (w); Eberhart and Shi (2000) show that with judicious parameter settings (w set to χ, and $c_1 + c_2$>4) the two approaches arealgebraically equivalent and improved performance could be achieved across a wide range of problems.

Parsopoulos and Vrahatis (2004) modified the constricted algorithm to harness the explorative behaviour of global search and exploitative nature of a local neighbourhood scheme. To combine the two, two velocity updates are initially calculated:

$$G_{t+1} = \chi\{v_{id}^{j} + c_1 R_1 \left(p_{id}^{j} - x_{id}^{j}\right) + \ c_2 R_2 \left(p_{gd}^{j} - x_{gd}^{j}\right)\} \quad (2.11)$$

$$L_{t+1} = \chi\{v_{id}^{j} + c_1 R_1^{'} \left(p_{id}^{j} - x_{id}^{j}\right) + \ c_2 R_2^{'} \left(p_{ld}^{j} - x_{ld}^{j}\right)\} \quad (2.12)$$

where G and L are the global and local velocity updates respectively, p_g is the global best particle position and p_l is the particle's local neighbourhood best particle position. These two updates are then combined to form a unified velocity update (U), which is then applied to the current position:

$$U_{t+1} = (1 - u)L_{t+1} + uG_{t+1} u \in [0,1] \qquad (2.13)$$

$$x_{t+1} = x_1 + U_{t+1} (2.14)$$

where u is a unification factor that balances the global and local aspects of the search and suggestions are given to add mutation style influences to each in turn. Experimentation was promising and further work has been carried out since for dynamic environments (Parsopoulos and Vrahatis 2005a, 2005b).

2.3.2. Applications

PSO has been applied to a number of areas. Yuan et al. (2004) apply PSO topower systems, such as distribution system expansion planning, generator maintenance scheduling, unit commitment, load dispatch, optimal power flow calculation and optimal control of reactive power, harmonic analysis and capacitor configuration. They claim, through a thorough study of PSO, that its great latent capacity will be brought into play in the electricity market auction, bidding strategy, and electricity market simulation. Kannana et al. (2003) present the application of variants of the PSO technique to the expansion planning problem. The results obtained are compared with dynamic programming (DP) and show that PSO outperforms DP in terms of both speed and efficiency.

PSO is also increasingly used in the area of finance. Nenortaite and Simutis (2004) present a decision-making method based on the application of neural networks and particle swarm techniques, which is used to generate one-step-ahead investment decisions. The experiments presented in the paper show that the application of their proposed method achieves better results than the market average. Kendall and Su (2005) apply PSO to the construction of optimal risky portfolios for financial investments. A particle swarm solver is developed and tested on various restricted and unrestricted risky investment portfolios. The particle swarm solver demonstrates high computational efficiency in constructing optimal risky portfolios of less than fifteen assets. Kwok et al (2009) propose the use of PSO to determine the appropriate long/short durations, when optimizing the rules generated by technical traders to maximize trading profit. The results are verified as effective. Jha et al., (2009) apply PSO to pricing options. The results are compared with the classical Black-Scholes model for simple European options and indicate that PSO is the superior method.

2.4. Artificial Immune Systems

Artificial Immune Systems are adaptive systems, which are inspired by theoretical immunology and observed immune functions, principles and models, and then applied to problem solving (Castro et al., 2002). AIS is distinct from computational immunology and theoretical biology. These are rather concerned with simulating immunology using computational and mathematical models, in order to gain a better understanding of the immune system. Such models, though, first led to establishing AIS as a new field of study, and they continue to provide a fertile ground for inspiration. Finally, the field of AIS is not concerned with the investigation of the immune system as a substrate computation, such as DNA computing.

AIS begin in the mid 1980s with Farmer, Packard and Perelson (1986) and

Bersini and Varela's papers on immune networks (1990). However, it is only in the mid 90s that AIS become a subject area in its own right. The common techniques are inspired by specific immunological theories that explain the function and behaviour of the mammalian adaptive.

2.4.1. Clonal Selection Algorithm

Clonal Selection Algorithm (CSA) is a class of algorithms inspired by the clonal selection theory of acquired immunity, which explains how B and T lymphocytes improve their response to antigens over time. This is called “affinity maturation”. These algorithms focus on the Darwinian attributes of the theory where selection is inspired by the affinity of antigen-antibody interactions, reproduction is inspired by cell division, and variation is inspired by somatic hyper-mutation. (deCastro, Von, 2002) Figure 2-4 below presents the flowchart and pseudo code for the overall CSA algorithm.

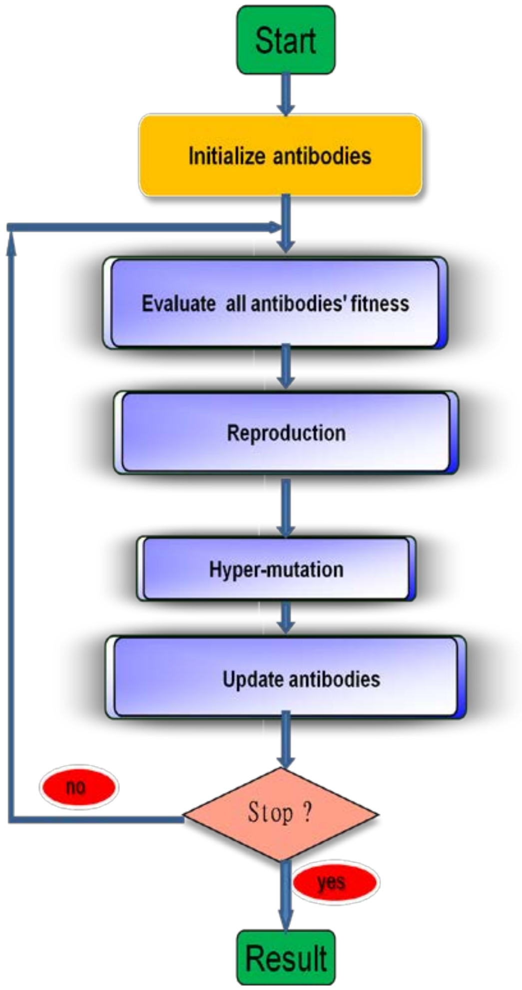

Figure 2-10: CSA flowchart

2.4.2 Applications

Clonal selection algorithms are most commonly applied to optimization and pattern recognition domains, some of which resemble parallel hill climbing and a genetic algorithm without the recombination operator, (deCastro and VonZuben, 2002). deCastro and VonZuben propose a powerful computational implementation of the clonal selection principle that explicitly takes into account the affinity maturation of the immune response. The algorithm is shown to be an evolutionary strategy capable of solving complex machine learning tasks, like pattern recognition and multimodal optimization. White and Garrett (2003) examine the clonal selection algorithm CLONALG and the suggestion that it is suitable for pattern recognition. CLONALG is tested over a series of binary character recognition tasks and its performance compared to a set of benchmark binary matching algorithms. A number of enhancements are made to the algorithm to improve its performance and the classification tests are repeated. Results show that given enough data, CLONALG can successfully classify previously unseen patterns and that adjustment to the existing algorithm can improve performance. Clonal selection algorithms haven't been used in the area of finance up to date. But as an optimization technique, it could still be applied to optimise the parameters of finance models in theory.

2.5. Game Theory

Game theory is the study of strategic decision making. More formally, it is "the study of mathematical models of conflict and cooperation between intelligent rational decision-makers."(Roger 1991).

2.5.1. Types of Games

Nash equilibrium

In game theory, the Nash equilibrium (named after John Forbes Nash, who proposed it) is a solution concept of a game involving two or more players, in which each player is assumed to know the equilibrium strategies of the other players, and no player has anything to gain by changing only his own strategy unilaterally. If each player has chosen a strategy and no player can benefit by changing his or her strategy while the other players keep theirs unchanged, then the current set of strategy choices and the corresponding payoffs constitute Nash equilibrium. (Osborne & Ariel, 1994)

Non-cooperative game and cooperative game

In game theory, a non-cooperative game is one in which players make decisions independently. Thus, while they may be able to cooperate, any cooperation must be self-enforcing. A game in which players can enforce contracts through third parties is a cooperative game. (Harsanyi, John, 1974)

Zero-sum game

In game theory and economic theory, a zero-sum game is a mathematical representation of a situation in which a participant's gain (or loss) of utility is exactly balanced by the losses (or gains) of the utility of the other participant(s). If the total gains of the participants are added up, and the total losses are subtracted, they will sum to zero. Thus cutting a cake, where taking a larger piece reduces the amount of cake available for others, is a zero-sum game if all participants value each unit of cake equally (see marginal utility). In contrast, non-zero-sum describes a situation in which the interacting parties' aggregate gains and losses are either less than or more than zero. A zero-sum game is also called a strictly competitive game while non-zero-sum games can be either competitive or non-competitive. (Ken 2007).

The prisoner's dilemma is a canonical example of a game analyzed in game theory that shows why two individuals might not cooperate, even if it appears that it is in their best interest to do so. It was originally framed by Merrill Flood and Melvin Dresher working at RAND in 1950. Albert W. Tucker formalized the game with prison sentence payoffs and gave it the "prisoner's dilemma" name (Poundstone, 1992). A classic example of the prisoner's dilemma (PD) is presented as follows:

Two men are arrested, but the police do not possess enough information for a conviction. Following the separation of the two men, the police offer both a similar deal—if one testifies against his partner (defects/betrays), and the other remains silent (cooperates/assists), the betrayer goes free and the one that remains silent receives the full one-year sentence. If both remain silent, both are sentenced to only one month in jail for a minor charge. If each testifies against the other, the other, each receives a three-month sentence. Each prisoner must choose either to betray or remain silent; the decision of each is kept quiet. What should they do? If it is supposed here that each player is only concerned with lessening his time in jail, the game becomes a non-zero sum game where the two players may either assists or betrays the other. In the game, the sole worry of the prisoners seems to be increasing his reward. The interesting symmetry of this problem is that the logical decision leads each to betray the other, even though their individual ‘prize’ would be greater if they cooperated.(Osborne & Ariel 1994)

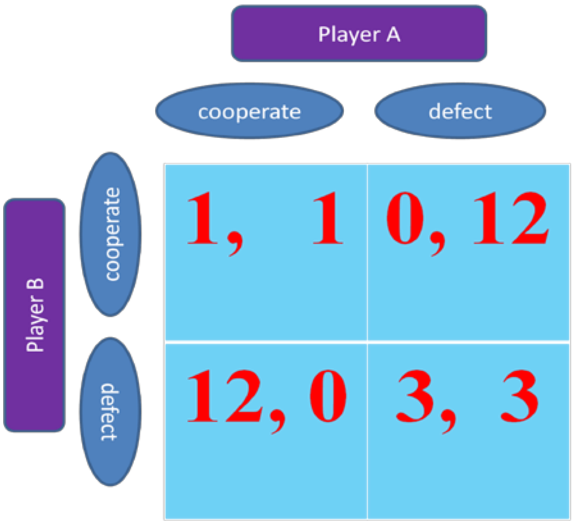

Figure 2-11: Prisoners’ dilemma

In the regular version of this game, collaboration is dominated by betrayal, and as a result, the only possible outcome of the game is for both prisoners to betray the other. Regardless of what the other prisoner chooses, one will always gain a greater payoff by betraying the other. Because betrayal is always more beneficial than cooperation, all objective prisoners would seemingly betray the other if operating purely rationally. However, in reality humans display a systematic bias towards cooperative behavior in Prisoner's dilemma and similar games, much more so than predicted by a theory based only on rational self interested action. (Ahn et al 2003)

2.5.2 Applications

Shafer and Vovk（2008）use a game-theoretic framework for probability to derive a capital asset pricing model from an efficient market hypothesis, with no assumptions about the beliefs or preferences of investors. While efficient market hypothesis says that a speculator with limited means cannot beat a particular index by a substantial factor, the model they derive says that the difference between the average returns of a portfolio and the index should approximate the difference between the portfolio's covariance with the index and the index's variance.

ItayandPauzner(2004) explore a model with two countries which might be subject to a self-fulfilling crisis, induced by agents withdrawing their investments in the fear that others will do so. While the fundamentals of the two countries are independent, the fact that they share the same group of investors may generate a contagion of crises. The realization of a crisis in one country reduces agents' wealth and thus makes them more risk averse (they assume decreasing absolute risk aversion). This reduces their incentive to maintain their investments in the second country since doing so exposes them to the strategic risk associated with the unknown behaviour of other agents. Consequently, the probability of a crisis in the second country increases. This yields a positive correlation between the returns on investments in the two countries even though they are completely independent in terms of fundamentals.

Dasgupt (2004) models financial contagion as an equilibrium phenomenon in a dynamic setting with coordinating game with incomplete information and multiple banks. The equilibrium probability of bank failure is uniquely determined. They explore how the cross holding of deposits motivated by imperfectly correlated regional liquidity shocks can lead to contagious effects conditional on the failure of a financial institution. They show that contagious bank failure occurs with positive probability in the unique equilibrium of the economy, and demonstrate that the presence of such contagion risk can prevent banks from perfectly insuring each other against liquidity shocks via the cross-holding of deposits.

Frankela et al (2008) study games with strategic complementarities, arbitrary numbers of players and actions, and slightly noisy payoff signals. They prove limit uniqueness: as the signal noise vanishes, the game has a unique strategy profile that survives iterative dominance. This generalizes a result of Carlsson and van Damme (1993) for two players, two action games. The surviving profile, however, may depend on fine details of the structure of the noise. They provide sufficient conditions on payoffs for there to be noise-independent selection.

Angeletos et al., (2007a) use coordination games of incomplete information to study the regime change. They extend the static benchmark examined in the literature by allowing agents to take actions in many periods and to learn about the underlying fundamentals over time. They first provide a simple recursive algorithm for the characterization of monotone equilibria. They then show how the interaction of the knowledge that the regime survived past attacks with the arrival of information over time, or with changes in fundamentals, leads to interesting equilibrium properties.

First, multiplicity may obtain under the same conditions on exogenous information that guarantee uniqueness in the static benchmark. Second, fundamentals may predict the eventual regime outcome but not the timing or the number of attacks. Finally, equilibrium dynamics can alternate between phases of tranquillity–where no attack is possible–and phases of distress–where a large attack can occur–even without changes in fundamentals

Angeletos et al (2007b) studies defence policies in a global-game model of speculative currency attacks. Although the signalling role of policy interventions sustains multiple equilibria, a number of novel predictions emerge which are robust across all equilibria. (i) The central bank intervenes by raising domestic interest rates, or otherwise raising the cost of speculation, only when the value it assigns to defending the peg - its "type" is intermediate. (ii) Devaluation occurs only for low types. (iii) The set of types who intervene shrinks with the precision of market information. (iv) A unique equilibrium policy survives in the limit as the noise in market information vanishes, whereas the devaluation outcome remains indeterminate. (v) The payoff of the central bank is monotonic in its type. (vi) The option to intervene can be harmful only for sufficiently strong types; and when this happens, weak types are necessarily better off. While these predictions seem reasonable, none of them would have been possible in the common-knowledge version of the model. Combined, these results illustrate the broader methodological point of the paper: global games can retain significant selection power and deliver useful predictions even when the endogeneity of information sustains multiple equilibria.

Bobtcheff and Mariotti (2012) study a preemption game in which two potential competitors come into play at some random secret times. The presence of a competitor is revealed to her opponent only when the former moves, which terminates the game. They show that all perfect Bayesian equilibria give rise to the same distribution of players' moving times, and the author explicitly construct such equilibrium. The intensity of competition is non-monotonic over time, and private information tends to alleviate rent dissipation. Our results have a natural interpretation in terms of eroding reputations.

2.6. Summary

Nature-inspired computational approaches are often viewed as global optimization methods, although convergence to a global optimum is only guaranteed in a weak probabilistic sense. However, one of the strengths of nature-inspired techniques is that they perform well on "noisy" functions where there may be multiple local optima. They tend not to get "stuck" on local minima and can often find globally optimal solutions.

Evolutionary computing is becoming a successful methodology in approaching problems in the area of finance. It has already been proven to be a powerful tool in domains where more conventional analytical solutions may not be a good alternative. The choice of optimization techniques it offers have the advantages of speed of convergence, along with not getting easily stuck in local optima. Characteristics that answer well our criteria in choosing parameter exploration techniques for the hybrid computational model. The author will particularly use genetic programming in developing the technical traders' strategies in our multinational-market model.

Swarm intelligence and artificial immune systems are relatively new nature-inspired approaches, compared to evolutionary computing, but have been already widely used in a range of areas. Swarm intelligence is also being increasingly applied to problems in finance, while financial applications of artificial immune systems are still rare and the author are yet to see further development there. Our work is a step in that direction, and the author will implement an algorithm for optimising the parameters of the overall multiple-market model with its different types of traders, where the algorithm is a combination of SI and AIS techniques. Particularly, it is a particle swarm optimization with a clonal selection algorithm (Immune-PSO).

Co-evolutionary algorithms (CoE), as an enhancement of nature-inspired algorithms, are a corollary of the fact that in complex domains it is difficult to assess an objective fitness measure for the problem. Co-evolution is also more realistic, reflecting similarity to real biological evolution. In CoE, fitness itself is a measurement of interacting individuals, which allows the potential for evolving greater complexity by allowing pieces of a problem to evolve in tandem. Thus, individual fitness is subjective, and is a function of its interaction with other individuals. For that reason, the author chooses to apply a co-evolutionary approach to the overall model and co-evolve the behaviours of different traders within the multi-national markets. This leads to co-evolving the behaviours of the markets, which is the more realistic option within the global financial system.

Finally, Game theory is the study of the ways in which strategic interactions among economic agents produce outcomes with respect to the preferences (or utilities) of those agents, where the outcomes in question might have been intended by none of the agents. It is easily understood and more suitable to be applied to our agent based model.

Chapter 3 : Measures of Interdependence for Random Variables

3.1 Introduction

Interdependence measures capture how two uncertain variables, e.g. two markets indices, are related to each other. In our agent-based model, described in details in Chapters 4 and 6, interdependence measures are used in the fitness function for the evolving model and its parameters. In this chapter, the author briefly discusses the correlation coefficient as a measure of interdependence between two random variables, and the limitations of this measure. Then, the author introduces the copula as an alternative measure, together with a discussion of types of copula and how their parameters are estimated.

3.2 Interdependence

3.2.1 Linear Correlation Coefficient

The linear correlation coefficient (r) between two random variables is a number between -1 and 1 which measures how close to a straight line a set of points falls in a plane, with the two coordinates of points given by corresponding realizations of two random variables. The closer to zero the correlation coefficient is, for a given set of points, the further away from a straight line they fall (hence the term "linear" correlation coefficient)(Black, 2006).The sign and the absolute value of a correlation coefficient respectively describe the direction, and the magnitude of the relationship between two variables.

- The greater the absolute value of a correlation coefficient, the stronger the linear relationship.
- The strongest linear relationship is indicated by a correlation coefficient of -1 or 1.
- The weakest linear relationship or the luck of relationship is indicated by a correlation coefficient equal to 0.
- A positive correlation means that if one variable gets larger, the other variable tends to get larger.
- A negative correlation means that if one variable gets larger, the other variable tends to get smaller.

The formula for linear correlation coefficient is

$$\rho(X,Y) = \frac{Cov\,(X,Y)}{\sqrt{Var\,(X)Var\,(Y)}} \quad , \qquad (3.1)$$

where $Cov(X,Y)$ is the co-variance of X, Y. $Var(X)$ and $Var(Y)$ are the variances of X or Y, respectively.

For random variables which are normally distributed, linear correlation completely characterizes the interdependence between them. However, the linear correlation coefficient is far less meaningful for non-normal distributions. Unfortunately, researchers find the data in the area of financial markets do not follow a normal distribution. Given this limitation of the linear correlation coefficient, the rank correlation coefficient has been given more weight in the area of financial markets.

3.2.2 Rank Correlation Coefficients

In statistics, rank correlation is the study of relationships between different rankings on the same set of items. A rank correlation coefficient measures the correspondence between two rankings and assesses its significance. Two of the more popular rank correlation statistics are Kendall's tau rank correlation coefficient (Kendall's $\mathfrak{H}$) and Spearman's rank correlation coefficient (Spearman's ρ). Rank correlation coefficient has at least four advantages. First of all, it is less sensitive to bias due to the effect of outliers. Secondly, it can be used to reduce the weighting of outliers, as large distances get treated as a one-rank difference. Thirdly, it does not require an assumption of data being distributed normally. Lastly, when one or more outlier exist, it is advisable to study the rankings rather than the actual values (Asuero et al, 2006).

3.2.2.1 Kendall's rank correlation coefficient

In statistics, the Kendall rank correlation coefficient, commonly referred to as Kendall's tau ($\mathfrak{H}$) coefficient, is used to measure the association between two measured quantities. A tau test is a non-parametric hypothesis test which uses the coefficient to test for statistical dependence. Specifically, it is a measure of rank correlation: that is, the similarity of the orderings of the data when ranked by each of the quantities. It is named after Maurice Kendall, who developed it in Kendall (1938), though Gustav Fechner had proposed a similar measure in the context of time series in 1897 (Kruskal, 1958).

Definition: Let $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ be a set of joint observations from two random variables X and Y respectively. Any pair of observations (x_i, y_i) and (x_j, y_j) are said to be concordant if $x_i, x_j\, and y_i, y_j$ moves in the same direction, otherwise they are said to be discordant. (Nelsen, 2001) In other words, if $x_i > x_j$ and$y_i > y_j$, or $x_i < x_j$ and$y_i < y_j\, then pair$ (x_i, y_i) and (x_j, y_j) are said to be concordant, otherwise discordant. The Kendall τ coefficient is defined as:

$$\mathfrak{H} = \frac{(number of concordant pairs)-(number of discordant pairs)}{\frac{1}{2}n(n-1)} . \qquad (3.2)$$

Properties: The denominator value is the total number of pairs, so the coefficient must be in the range $-1 \leq \mathfrak{H} \leq 1$. If the agreement between the two rankings is perfect, i.e. the two rankings are the same, the coefficient has value of 1.

(a) If the divergence between the two rankings is perfect, i.e. one ranking is the reverse of the other, the coefficient has value of -1.

(b) If X and Y are independent variables, then the author would expect the coefficient to be approximately zero.

3.2.3 Copula - A General Measure of Interdependence

A copula is a multivariate joint distribution defined on the n-dimensional unit cube $[0,1]^n$, such that every marginal distribution is uniform on the interval $[0,1]$. Specifically, $C: [0,1]^n \rightarrow [0,1]$ is an n-dimensional copula, or briefly n-copula, if:

$C(u) = 0$ whenever $u \in [0,1]^n$ has at least one component equal to 0;

$C(u) = u_i$ whenever $u \in [0,1]^n$ has all the components equal to 1 except the ith one, which is equal to u_i;

$C(u)$ is n-increasing, for each hyper-rectangle

$$B = [x, y] = [x_1, y_1] \times [x_2, y_2] \times \cdots [x_n, y_n] \in [0,1]^n. \qquad (3.3)$$

3.2.3.1 Sklar's Theorem

The theorem proposed by Sklar (1959) underlies most applications of the copula. Sklar's theorem states that given a joint distribution function H for p variables, and respective marginal distribution functions, there exists a copula C such that the copula binds the margins to give the joint distribution.

For the bivariate case, Sklar's theorem can be stated as follows. For any bivariate distribution function $H(x, y)$, let $F(x) = H(x, \infty)$ and $G(y) = H(\infty, y)$ be the univariate marginal probability distribution functions. Then there exists a copula C such that

$$H(x, y) = C(F(x), G(y)) \ , \qquad (3.4)$$

where the distribution C is identified with its cumulative distribution function. If the marginal distributions $F(x)$ and $G(y)$ are continuous, the copula function C is unique. Otherwise, the copula C is unique on the range of values of the marginal distributions.

Sklar's Theorem is very important because $F(x), G(y)$ could be any distribution, and even if they are of different distribution, Sklar's Theorem still holds. That means the author can find a unique function to link two distribution functions. Further, this function can be parameterized independently from the underlying univariate distributions.

3.2.3.2 Copula Family

A: Gaussian copula

One example of a copula often used for modelling in finance as introduced by Li (2000) is the Gaussian copula, which is constructed from the bi-variate normal distribution via Sklar's theorem. With Φ_ρ being the standard bivariate normal cumulative distribution function with correlationρ, the Gaussian copula function is

$$C_\rho(u,v) = \Phi_\rho\left(\Phi^{-1}(u), \Phi^{-1}(v)\right) \quad , \qquad (3.5)$$

where $u\,, v \in [0,1]$ and Φ denotes the standard normal cumulative distribution function.

Differentiating C yields the copula density function:

$$c_\rho(\text{u,v}) = \frac{\varphi_{XY\rho}\left(\Phi^{-1}(u), \Phi^{-1}(v)\right)}{\varphi\left(\Phi^{-1}(u), \Phi^{-1}(v)\right)} \quad , \qquad (3.6)$$

where

$$\varphi_{XY\rho}(x,y) = \frac{1}{2\pi\sqrt{1-\rho^2}} exp\left(-\frac{1}{2(1-\rho^2)}[x^2 + y^2 - 2\rho xy]\right) \qquad (3.7)$$

is the density function for the standard bivariate Gaussian, φis the standard normal density.

B. Archimedean copulas

One particularly simple form of an n-dimensional copula is

$$H\left(x_{1\,,}x_{2\,,}x_{3\,,} \cdots x_n\right) = \Psi^{-1}\left(\sum_{i=1}^{n} \Psi(F_i\,(x_i))\right) \;, \qquad (3.8)$$

where Ψ is known as a generator function. Such copulas are known as Archimedean. Any generator function which satisfies the properties below is a basis for a valid copula:

$$\Psi(1) = 0\ ;\ \ lim_{x\to 0}\,\Psi(x) = \infty;\ \Psi'(x) < 0\ ;\ \Psi''(x) > 0\ \ . \qquad (3.9)$$

Archimedean copulas are an important family of copulas, which have a wide range of applications. There are a number of reasons for this. They are easy to construct, there is a great variety of copula families belonging to this class (Roger, 2006).

Three common Archimedean copulas

(i) Clayton copula(Nelsen,1999):

$C_\theta(x, y) = max([x^{-\theta} + y^{-\theta} - 1]^{-1/\theta}, 0)$ (3.10)
and its generator is $\Psi_\theta(t) = (t^{-\theta} - 1)$, where $\theta \in [-1, \infty)\backslash\{0\}$

For $\theta = 0$ in the Clayton copula, the random variables are statistically independent. The generator function approach can be extended to create multivariate copulas, by simply including more additive terms.

(ii) Gumbel copula(Nelsen,1999)::

$$C_\alpha(x, y) = exp\{-[(-lnx)^\alpha + (-lny)^\alpha]^{1/\alpha}\} \qquad (3.11)$$

and its generator is:

$$\Psi_\alpha(t) = (-ln(t))^\alpha,\ \text{where}\ \alpha \in [1, \infty)$$

(iii) Frank copula(Nelsen,1999):

$$C_\alpha(x, y) = -\frac{1}{\alpha} l\,n\left(1 + \frac{(e^{-\alpha x} - 1)(e^{-\alpha y} - 1)}{e^{-\alpha} - 1}\right) \qquad (3.12)$$

and its generator is:

$$\Psi_\alpha(t) = -ln(\frac{e^{-\alpha t} - 1}{e^{-\alpha} - 1}),\ \text{where}\ \alpha \in [-\infty, \infty)\backslash\{0\}$$

3.3. GARCH Model and Clayton-Copula for Return Simulation

3.3.1 Selecting the Copula Function

In our agent-based model, an interdependence measure is used in the fitness function, which further contains a group of parameters to be optimized. Previous models for contagion simulation use the linear correlation coefficient in the optimisation criterion. Given the limitation of that measure of interdependence, as described earlier in this chapter, the use of a copula function is suggested here. Copulas are becoming popular in the area of financial mathematics, overall. Given two random variables X, Y, with univariate distributions Fx, Fy, suppose that a joint distribution $H(x, y)$ also exists. Then there exists a copula, which maps the pair (Fx, Fy) into $H(x, y)$. Crucially, such function can be parameterized independently from the underlying univariate distributions. Different choices of copula functions focus on different aspects of interdependence (Nelsen, 1999; Cherubini, 2004).

In this work, the author use the Clayton copula to model the interdependence between two markets, as the author focus on the left tail risk. One of the qualities of the Clayton copula consists of capturing well tail dependence. The Clayton copula is defined in equation (3.11)

3.3.2 Calculating Tail Dependence

Tail dependence describes the conditional probability that one margin exceeds a certain threshold given that the other margin has already exceeded that threshold. Let X and Y be random variables with distribution functions F and G, respectively. The upper tail dependence parameter λ_U is the limit, if it exists, of the conditional probability that Y is greater than the 100ζ-th percentile of G given that X is greater than the 100ζ-th percentile of F as the parameter ζ approaches 1 (Bickel et al., 2006). Therefore,

$$\lambda_U = lim_{\zeta \to 1^-} P[Y > G^{-1}(\zeta) \,|\, X > F^{-1}(\zeta)] \ . \qquad (3.13a)$$

Similarly, the lower tail dependence parameter λ_L is the limit, if it exists, of the conditional probability that Y is less than or equal to the 100ζ-th percentile of G given that X is less than or equal to the 100ζ-th percentile of F as ζ approaches 0:

$$\lambda_L = lim_{\zeta \to 1^-} P[Y \leq G^{-1}(\zeta) \,|\, X \leq F^{-1}(\zeta)] \ . \qquad (3.13b)$$

The author are interested in the lower tail dependence coefficient of the Clayton copula, in particular. According to Nelsen (1999) it is evaluated as:

$$\lambda_{L,Clayton} = 2^{-\frac{1}{\theta}} \ . \qquad (3.14)$$

Furthermore, there is a relationship between parameter θ from the Clayton copula and the Kendal $\mathfrak{H}$ coefficient, and therefore between tail depencence and $\mathfrak{H}$. The relationship is in the following form:

$$\theta = \frac{2\mathfrak{H}}{1-\mathfrak{H}} \ , \qquad (3.15a)$$

$$\lambda_{L,Clayton} = 2^{\frac{\mathfrak{H}-1}{2\mathfrak{H}}} \ . \qquad (3.15b)$$

Therefore, in order to use tail dependence as a risk-relevant measure of interdependence between financial market series, the author need to evaluate parameter θ of the Clayton copula, and it is possible to do that by first calculating Kendal's $\mathfrak{H}$. As defined in formula (3.2), $\mathfrak{H} = \frac{(number of concordant pairs)-(number of discordant pairs)}{\frac{1}{2}n(n-1)}$, where any pair of observations $(x_i, y_i),(x_j, y_j)$ is concordant if $(x_i - x_j)(y_i - y_j) > 0$ while otherwise pairs are discordant. Furthermore, to be able to link $\mathfrak{H}$ to θ and $\lambda_{L,Clayton}$ in formula (3.15a, b), the observations (x_i, y_i) should be drawn from uniform distributions. Thus it is necessary to transform first the raw data, which may come from different underlying distributions. The author next applies CARCH modelling to transform the return series data.

3.3.3 Mapping to Uniform Distribution

A large number of references (e.g. Engel, 1982) reveal that Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) models can successfully capture market dynamics and volatility. Here, the author chooses GARCH (1,1) to model both the return data series in the domestic market and in the foreign market. Let us consider normal GARCH (1,1) model:

$$R_t = \mu + \epsilon_t \quad (3.16a)$$

$$\epsilon_t = \sigma_t * \varepsilon_t\ ,\ \varepsilon_t \backsim N(0,1) \qquad (3.16b)$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta \sigma_{t-1}^2 \quad , \qquad (3.16c)$$

where R_t models the real return seriesr_t, μ is the mean value of r_t, σ_t is the standard deviation, and ε_t is a random variable with standard normal distribution, while $\alpha_0, \alpha_1, \beta$ are model parameters. (Bollerslev, 1986) The returns are modelled as conditionally normal, and a maximum likelihood aproach can be used to evaluate the parameters.

Next, the author transforms the stock return series $r_t$into the uniform distribution with the following formula:

$$Prob(R_{t+1} \le r_{t+1}) = Prob(R_{t+1} - \mu \le r_{t+1} - \mu) = Prob(\epsilon_{t+1} \le r_{t+1} - \mu)$$

$$=Prob(\sigma_{t+1} * \varepsilon_{t+1} \leq r_{t+1} - \mu)=Prob(\sqrt{\alpha_0 + \alpha_1\epsilon_t^2 + \beta\sigma_t^2}\varepsilon_{\mathrm{t}+1} \leq r_{t+1} - \mu)$$

$$=Prob\left(\varepsilon_{t+1} \leq \frac{r_{t+1}-\mu}{\sqrt{\alpha_0+\alpha_1\epsilon_t^2+\beta\sigma_t^2}}\right) = N\left(\frac{r_{t+1}-\mu}{\sqrt{\alpha_0+\alpha_1\epsilon_t^2+\beta\sigma_t^2}}\right) \qquad , (3.17)$$

where R_t satisfies (3.16) above. Thus the author can uniquely map the real time series r_t to a time series that has uniform distribution and takes values between 0 and 1. Applying that mapping to two stock return series, the author then can estimate Kendal's $\mathfrak{H}$ with formula (3.2) where the pairs of observations$\left(x_i^{map}, x_i^{map}\right)$,$\left(x_j^{map}, x_j^{map}\right)$ are taken from the mapped series. Finally, Clayton copula's parameter θ and the tail dependence $\lambda_{L,Clayton}$ are evaluated using formula (3.15).

3.4 Conclusions

In this chapter, the author briefly discusses the correlation coefficient as a measure of interdependence between two random variables, and the limitations of this measure. Then, the author introduces the copula as an alternative. Specifically, the author choose the Clayton copula to model the interdependence between two markets, as the author focus on the left tail risk and one of the qualities of the Clayton copula consists of capturing well lower tail dependence. The author also introduce a GARCH (1, 1) approach to help to transform the share price series into uniform distribution which are used for estimating Kendal's $\mathfrak{H}$. Finally, Clayton copula's parameter θ and the tail dependence $\lambda_{L,Clayton}$ are evaluated. The tail dependence will be used in the next chapter as the optimization criteria in the fitness function and will contribute as a more reliable measure of interdependence than the linear correlation coefficient. The comprehensive model GARCH Clayton copula is the contribution of the author and the components GARCH (1, 1), and Clayton copula are developed by mathematicians before the author.

Chapter 4 : An Agent-Based Model of Financial Contagion

4.1 Introduction

A series of historical financial crises, such as Mexico in 1987, Asia in 1997, and Russia in 1998, all share a common feature – problems spread from one country to others. The spread is due to the cross-market linkages. If the cross-market linkages stay stable then the crises is transferred through interdependence, and the recovery follows the recovery of the underlying economic reason in the country of origin. When the cross-market linkages get destabilized due to the crisis, then the crisis starts "feeding on it self" and the recovery of the underlying economic reason is not sufficient to get control of the crisis; a more comprehensive strategy with international involvement is required. The second type of crisis exhibits the phenomenon called 'financial contagion'.

There is no conclusive definition of the financial contagion phenomenon described above. Most research papers identify contagion by analyzing the change in the variance-covariance matrix during the period of market turmoil. King and Wadhwani (1990) first test the correlations between the US, UK and Japan, during the US stock market crash in 1987. Boyer (1997) finds significant increases in correlation during financial crises, and reinforces a technical definition of financial contagion as a correlation breakdown during the crash period. Forbes and Rigobon (2002) define financial contagion as "a significant increase in cross-market linkages after a shock to a group of countries". In their work, the term "interdependence" is used as the alternative to "contagion". Interdependence leads to common price movements during periods both of stability and turmoil.

In the past two decades, many studies (e.g. Kaminsky et at., 1998; Kaminsky 1999) developed early warning systems focused on the origins of financial crisis rather than on financial contagion. Other works (e.g. Forbes and Rigobon, 2002; Caporale et al., 2005) focused on studying contagion compared to interdependence. In this chapter, the author model and simulate the transmission of financial crises, through the behavior of market players and their various strategies, using an integrated approach that involves a mixed-game (Game), co-evolutionary genetic programming (GP), a multinational agent-based model, and Clayton Copula. Our multinational model is developed to suit analyzing financial contagion; it is composed of four types of traders - Technical-GP, Technical-Game, Herd and Noise traders. A Technical-GP trader is a trader who makes decisions based on the technical analysis of price charts. Technical-GP traders analyze price charts to develop theories about the direction in which the market is likely to move. Technical-GP traders are modelled in the artificial market here through co-evolutionary genetic programming. Technical-Game traders make decision based on their decision tables. Game theory is a branch of applied mathematics and economics. The so-called "minority game", as a further development of game theory, is especially useful to simulate real financial markets (Lebaron, 2006). Herd traders, as important as Technical-GP and Technical-

Game traders when describing market behaviour, are the ones who make buy and sell decisions following the prevailing behaviour, regardless of other factors and market fundamentals. Herd behaviour has been identified as a major factor behind contagion (Cont and Bouchaud, 2000). Finally, Noise traders are stock traders whose decisions to buy, sell, or hold are irrational and erratic: their presence in financial markets can cause prices and risk levels to diverge from expected levels even if all other traders are rational (De Long et al., 1990).

In this chapter, the author develops a comprehensive model comprising the four types of traders: technical-GP, technical-Game, herd, and noise traders, respectively. Furthermore, the one-market model is extended to a two-market model, in order to explore how financial contagion happens, by observing the evolution of the interactions between the two markets.

4.2 Price Formation and Assets Allocation

Real financial markets are composed of different types of participants who interact through asset trading. A market player i generally holds two types of assets:

a risky asset, denoted by $h_i(t)$;

cash, denoted by $c_i(t)$.

The proportions of technical-GP, technical-Game, herd and noise traders, are denoted as $N_{GP}, N_{Game}, N_{Herd}, N_{Noise}$, respectively. The notation $P(t)$ stands for the share price at time t. The initial conditions include 10 shares, £1000 each and £10,000 cash available to each player. At any step in time, a trader buys or sells certain number of assets according to their own trading rules.

The price formation mechanism that the author uses here is similar to the one used in Giardina and Bouchaud (2003). A player i, takes a decision $d_i(t)$, at each time step, where a decision to buy is denoted with $d_i(t) = 1$, to sell with $d_i(t) = -1$, and to do nothing $d_i(t) = 0$. Moreover, they will make a bid or offer of a fraction $q_i(t)$ of their current holdings, where

$$q_i(t) = \begin{cases} g\frac{c_i(t)}{P_i(t)} if d_i(t) = 1 \\ -gh_i(t) if d_i(t) = -1 \\ 0 \qquad\qquad if d_i(t) = 0 \end{cases} \qquad (4.1)$$

and g denotes the fraction of the maximum change of an agent's holdings. It is an important parameter related to the cautiousness of the agents. In a market with heterogeneous agents, g should be deferent for each agent. However, to make the simulation simple, the author assumes all agents are risk neutral and all have the same cautiousness coefficient. Namely, g is constant. $B(t)$ stands for the aggregated volume of bids, and $O(t)$ for the aggregated volume of offers. These functions are used to calculate the excess demand $D(t) = B(t) - O(t)$, and $D(t)$ is used in a price determination equation similar to the ones proposed in Cont (2003). Thus price is calculated with the following formula:

$$P(t) = P(t-1) + D(t)/\lambda \qquad (4.2)$$

where λ is an important parameter representing the market sensitivity to the order imbalance.

The fraction of fulfilled orders is similar to the one introduced in Giardina and Bouchaud (2003). The total number of shares that can be bought at the new price is calculated as:

$$\widetilde{B}(t) = B(t)\frac{P(t-1)}{P(t)} \quad (4.3)$$

From formula (4.3), the author can see that if price goes up at time point t, the actual number of shares that can be bought $\widetilde{B}(t)$ is less than the original order $\mathrm{B}(t)$, and vice versa. The fraction of fulfilled buy σ_+ and sell orders σ_- can be described as follows:

$$\sigma_+ = min(1, \ \frac{O(t)}{\widetilde{B}(t)}) \quad \text{and} \ \sigma_- = min(1, \ \frac{\widetilde{B}(t)}{O(t)}) \qquad (4.4)$$

If global amount of sell orders $O(t)$ is bigger than the actual number of shares that can be bought $\widetilde{B}(t)$, then the fulfilled buy orders are still $\widetilde{B}(t)$, namely, the fraction of fulfilled buy σ_+ is 1. Similarly, if global amount of sell orders $O(t)$ is smaller than the actual number of shares that can be bought $\widetilde{B}(t)$, then only $O(t)$ shares could be bought. That is to say, the fraction of fulfilled buy σ_+ is $\frac{O(t)}{\widetilde{B}(t)}$. The same rule applies to the fraction of sell orders σ_-.

Having established this, the author can now calculate the amount of shares $\rho_i(t)$ that the agent i will buy or sell,

$$\rho_i(t) = \begin{cases} g\sigma_+ \frac{c_i(t)}{P_i(t)} if d_i(t) = 1 \\ -g\sigma_- h_i(t) if d_i(t) = -1 \\ 0 \qquad\qquad if d_i(t) = 0 \end{cases} \quad (4.5)$$

Finally, the author can update the traders' holdings of cash and the risky asset:

$$h_i(t) = h_i(t-1) + \rho_i(t) \quad (4.6a)$$

$$c_i(t) = c_i(t-1) + \rho_i(t)P(t) \quad (4.b)$$

4.3 Single Market Model

Noted above, in our model, the author classifies the market players into four categories:

a) Noise traders: these make decisions to buy, sell or do nothing with different probabilities. These probabilities are different for each player and are randomly predefined at the beginning and remain constant during simulation.

b) Herd traders: these tend to follow the trend of price movements, and the probability of a 'hold' transaction at time t is denoted with $p_{0,t}^{Herd}$ and calculated as follows:

$$p_{0,t}^{Herd}(i) = \frac{1}{1+d|\zeta_{t-1}|} \quad , \quad (4.7)$$

where parameter d controls the sensitivity to price change. This formula ensures that when the price movement $\zeta_{t-1}$is big, then the probability of holding a share is small and vice versa.

The probability of a "buy" decision is correspondingly:

$$p_{1,t}^{Herd}(i) = (1 - p_{0,t}^{Herd}(i))\frac{exp(\zeta_{t-1})}{exp(\zeta_{t-1}) + exp(-\zeta_{t-1})} \quad . \qquad (4.8)$$

Here, ζ_t is the overall price change at time t. Correspondingly, the probability of a "sell" decision is:

$$p_{-1,t}^{Herd}(i) = 1 - p_{0,t}^{Herd}(i) - p_{1,t}^{Herd}(i) \quad (4.9)$$

c) Technical-Game traders: in our model, these play a mixed game, which is to say that technical-Game traders are divided into two groups, one of which plays a minority game and the other play a majority game. Tanaka-Yamawaki and Tokuka (2006) propose a minority game, where traders take one of two possible actions: buy (1) or sell (0). If the minority side is defined to mean the decision made by a minority of traders, those who end up on the minority side win the game – the price will move in their favour. After each trade is executed, all the traders know, by the way the price has moved, whether the right choice would have been to buy or sell. Also in the model, all agents have their own decision table.

In our model, the author add one more choice "hold" (do nothing) to the model, to make it more realistic. Now the buy, hold and sell decision are denoted by 1, 0 and -1, respectively.

Table 4-1: An exemplary decision table

Historical string	Strategy(1)	Strategy(2)	Strategy(3)
-1,-1	+1	0	0
-1,0	+1	+1	0
-1,+1	0	-1	+1
0,-1	0	0	0
0, 0	-1	0	-1
0,+1	+1	+1	-1

+1,-1	0	+1	-1
+1,0	-1	+1	-1
+1,+1	0	-1	0

Elements of the decision table:

- Memory size, m
- Number of strategies, k, included in a decision table
- Binary descriptions (-1 sell, 0 hold, 1 buy)
- Huge pool (3^{3^m}) of possible strategies

Table 4-1 gives an example of a decision table, with $m = 2$, and $k = 3$. There are $K = 19{,}683$ possible strategies for $m = 2$. The decision table of a single agent includes only a few strategies out of these, in our case $k = 3$ strategies. The strategy table becomes a baseline for a trader to make decisions. For example, if the historical string "$-1 - 1$" happened, which means the correct decision for the past two trade days would have been "sell", then strategy one recommends to choose 1 in the current period, which means "buy", but strategy two recommends to hold choosing 0. To select a strategy and evolve decision tables after each trade, traders re-evaluate all strategies; increasing the score for each strategy that produced the right decision and reducing the score for the strategies that gave wrong decisions. During the next trading period, each trader makes decision following the strategy with the highest score that is available to him or her. Importantly, all the traders have their own decision tables, each trader works with different k strategies out of the large strategy pool K. The example in Table 4-1 presents one particular trader. The score w_{s_i} for each strategy is calculated as follows:

$$w_{s_{i},t} = w_{s_{i},t-1} + \alpha_t \quad , \ \alpha_t = \begin{cases} 1 \ rightdicision \\ 0 \qquad\qquad hold \\ -1 \ wrongdecision \end{cases} \quad , \qquad (4.10)$$

where α_t is the decision made in time period t.

d) Technical-GP traders (Martinez-Jaramillo and Tsang, 2009)**:** technical-GP analysis is a key feature of our model. This group of traders presents the richest range of behaviours. Technical-GP traders use GP to develop trading rules, and each individual technical-GP trader is represented by a different decision tree. The basic elements of such decision trees are rules and forecast values. A single rule is made up with a combination of three technical indicators, one rational operator such as "greater than" or "less than or equal to", and a real-value threshold. The three technical indicators are moving average (MA), trading breakout (TRB), and volatility (VOL). A single rule interacts with other rules in one decision tree through logical operators such as "or", "and", "not" and "if-then-else", as shown in Figure 4-1 presenting an example of a decision tree. The root node is always an "if-then-else" node (ITE); an ITE node has two children, each of which could be either a decision node or another "if-then-else" node. The code following Figure 4-1 shows how the decision-rule logic is derived from the decision tree.

For each technical GP player at time t, after the crossover and mutation of its group of decision trees, test its decision trees by comparing the right decisions with the decisions made by its decision trees from the start time to time t-1. Calculate the number of right decisions made by each tree and rank them by the fitness according to the following contingent Table 4-2.The best tree will be selected and used to make decision at time t.

Table 4-2: Contingent table

	Predicted price rise (PBs) **BUY**	**Predicted no inf.** (PHs) **HOLD**	**Predicted price drop** (PSs) **SELL**
Actual price rise (ABs) **BUY**	# of True Buys (TB)	# of Actual Buy Predicted Hold (BH)	# Actual Buy Predicted Sell (BS)
Actual no inf. (AHs) **HOLD**	# of Actual Hold Predicted Buy (HB)	# of True Holds (TH)	# of Actual Hold Predicted Sell (HS)
Actual price drop (ASs) **SELL**	# of Actual Sell Predicted Buy (SB)	# of Actual Sell Predicted Hold (SH)	# of True Sells (TS)

$$\mathrm{RC} = \frac{TB + TH + TS}{ABs + AHs + ASs}$$

RC is the fitness of each decision tree.

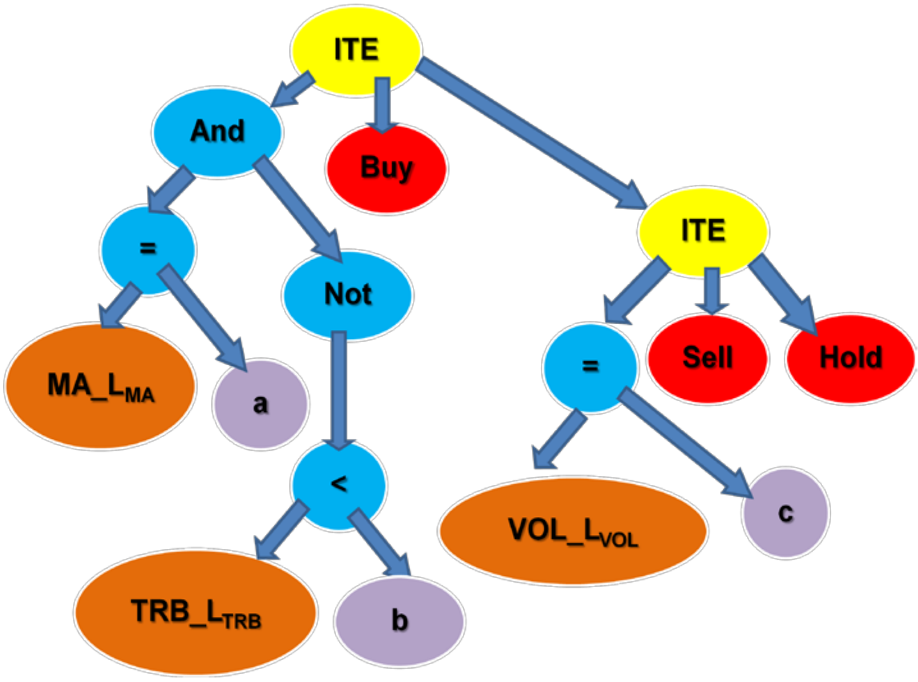

Figure 4-1: An exemplary decision tree

In the above algorithm, parameters a, b, c are different for each technical-GP trader, which allows for any trading preferences. The parameters are initially drawn randomly for each trader, from a standard normal distribution, and then remain constant during the evolution of the decision trees and the optimisation of the overall model. The technical indicators are calculated based on the periods L_{MA}, L_{TRB}, L_{VOL}, correspondingly, and based on the data used to evaluate the model. The Moving Average (MA) indicator is defined as:

$$MA(L_{MA}, t) = \frac{Price\ (t) - \frac{1}{L}\sum_{i=1}^{L_{MA}} Price\ (t-i)}{\frac{1}{L}\sum_{i=1}^{L_{MA}} Price\ (t-i)} \quad , \ (4.11a)$$

the Trading Breakout (TRB) indicator is defined as:

$$TRB(L_{TRB}, t) = \frac{Price\ (t) - \max(Price\ (t-1), \cdots Price\ (t-L_{TRB}))}{\max(Price\ (t-1), \cdots Price\ (t-L_{TRB}))} \quad , \quad (4.11b)$$

and the Volatility (VOL) indicator is defined as:

$$VOL(L_{VOL}, t) = \frac{\sigma(Price\ (t-1), \cdots Price\ (t-L_{VOL})}{\frac{1}{L_{VOL}} \sum_{i=1}^{L_{VOL}} Price\ (t-i)} \quad (4.11c)$$

d.1) Co-evolutionary GP setup for technical-GP traders: the basic decision tree for a technical-GP trader is introduced above and the author can now build on this to a co-evolutionary nature following a “red queen” principle. The “red queen” principle was originally proposed by Leigh Van Valen (1973) as a metaphor of a co-evolutionary arms race between species. The result is that inevitably, when competing for scarce resources, one party is going to end up the winner controlling the majority of those resources. In our setting, the red queen principle is applied in the form of “red queen retraining”, similar to Martinez-Jaramillo and Tsang (2009). When a trader’s wealth falls below the average wealth, he/she launches a GP mechanism to evolve the population of rules by retaining half of his/her current population and initializing the other half randomly.

4.4 Traders in the Multinational Market Model

The author considers a two-nation market for notational and computational simplicity. The results can be generalized to involve more than two countries.

a) **Noise traders in a multinational market model:** within a single market, noise traders make decisions to buy, sell or do nothing, with different probabilitiesp^b, $p^s andp^h$, respectively. The probabilities are predefined and remain constant during the whole process of simulation. These settings remain generally the same in the multinational market model; however, the author now has noise traders associated with each of the markets. For example, $p_b^{Noise,A}$, $p_s^{Noise,A}$, $p_h^{Noise,A}$, are the probabilities for noise traders' decisions in market A. For market B, the corresponding probabilities are $p_b^{Noise,B}$, $p_s^{Noise,B}$, $p_h^{Noise,B}$.

b) **Herd traders in a multinational market model:** in the multinational market model, herd traders summarize the latest price changes in all markets, and have a tendency to follow the overall market trend (price change) ζ with sensitivity τ^{Market} . This means that herd traders in market A will tend to follow the overall price change in all markets ζ with sensitivity τ^{A} towards that change. For example, in the case of two markets, market A and market B, the probability $p_{1,t}^{Herd,A}(i)$ at time t of herd traders in market A to make a buy (1) decision is given as follows:

$$p_{1,t}^{Herd,A}(i)=\frac{exp(\zeta_{t-1}^{A})}{exp(\zeta_{t-1}^{A})+exp(-\zeta_{t-1}^{A})} \quad , \qquad (4.12a)$$

$$\zeta_{t-1}^{A}=\zeta*\tau^{A}=\sum_{Market=A}^{B}\frac{Price_{t-1}^{Market}-Price_{t-2}^{Market}}{Price_{t-2}^{Market}}\tau^{A} \quad . \qquad (4.12b)$$

Higher sensitivity will lead to a higher probability of following the overall price change. The probability to make a sell decision (-1) is $p_{-1,t}^{Herd,A}(i) = (1-p_{1,t}^{Herd,A}(i))$, if the author only consider buy and sell decisions (Caporale et al., 2009). However, in order to add one more strategy "hold", the author have further introduced into the above formula, the probability of a hold strategy calculated as follows:

$$p_{0,t}^{Herd,A}(i)=\frac{1}{1+d|\zeta_{t-1}^{A}|} \qquad (4.13)$$

by analogy with formula (4.7). The probability of a "buy" decision is revised then to:

$$p_{1,t}^{Herd,A}(i) = (1 - p_{0,t}^{Herd,A}) \frac{exp(\zeta_{t-1}^{A})}{exp(\zeta_{t-1}^{A}) + exp(-\zeta_{t-1}^{A})} \quad . \qquad (4.14)$$

Correspondingly, the probability of a "sell" decision becomes:

$$p_{-1,t}^{Herd,A}(\mathrm{i}) = 1 - p_{0,t}^{Herd,A}(\mathrm{i}) - p_{1,t}^{Herd,A}(i) \quad . \qquad (4.15)$$

c) Technical-Game players in a multinational market model: in order to simulate the linkage between two markets, the mixed-game-based players are extended by allowing players to make investment decisions based on information from both domestic and foreign markets, while investing in the domestic market (Caporale et al. 2009). The author again consider two markets, A and B, where A is the domestic market and B is the foreign market. An agent i $(i = 1, \cdots N^{A})$ operating in market A, has a probability $P_{i}^{DOM,A}(i)$ of choosing an action at time t based on the domestic market, described as follows:

$$P_{i}^{DOM,A}(i) = \frac{exp(\gamma_{Game}^{A} \ w_{t}^{A}(\mathrm{i}))}{exp(\gamma_{Game}^{A} \ w_{t}^{A}(i)) + exp(-\gamma_{Game}^{A} \ w_{t}^{A}(i))} \quad , \qquad (4.16)$$

where γ_{Game}^{A} is a scale factor. Agents will take actions based on either the domestic or the foreign market. Thus the probability $P_{i}^{FOR,A}(i)$ of an agent in market A choosing an action based on the foreign market is $1 - P_{i}^{DOM,A}(i)$. Parameter $w_{i}^{A}(i)$ is updated after each period. If a decision based on the foreign market history loses the game, then $w_{t}^{A}(i) = w_{t-1}^{A}(i) + 1$. If the action based on the foreign market history wins the game, and the decision based on the domestic market history loses, then $w_{t}^{A}(i) = w_{t-1}^{A}(i) - 1$. Therefore, if the decisions based on the domestic and foreign markets are different, $w_t^A(i)$ is updated (Caporale et al. 2009). By analogy, the probabilities and score update can be evaluated for the case when an agent i ($i = 1, \cdots N^B$) operates in market B, and therefore market B is his/her domestic market. Once a trader chooses a market to make a decision, for example, market B, he/she then makes the decision using the historical price of market B, and the decision table for market B. After the trade, he/she updates the score for each strategy in decision table B.

Table 4-3: A trader's decision table involving markets A and B

Historical string	s_1^A	s_2^A	s_3^A	s_1^B	s_2^B	s_3^B
-1,-1	1	0	0	0	1	1
-1,0	1	1	0	1	1	0
-1,+1	0	-1	1	1	1	1
0,-1	0	0	0	0	0	0
0,0	-1	0	1	1	0	1
0,+1	1	1	1	1	1	1
+1,-1	0	1	1	0	1	1
+1,0	-1	1	1	1	1	1
+1,+1	0	-1	0	0	1	0

d) Technical-GP traders in a multinational market model: the process by which the technical-GP traders choose the market - domestic or foreign - on the information of which to base their decision to trade in the domestic market, is similar to the process followed by the technical-Game players. All Technical-GP players also co-evolve their decision trees.

4.5 Evaluating the Integrated Model

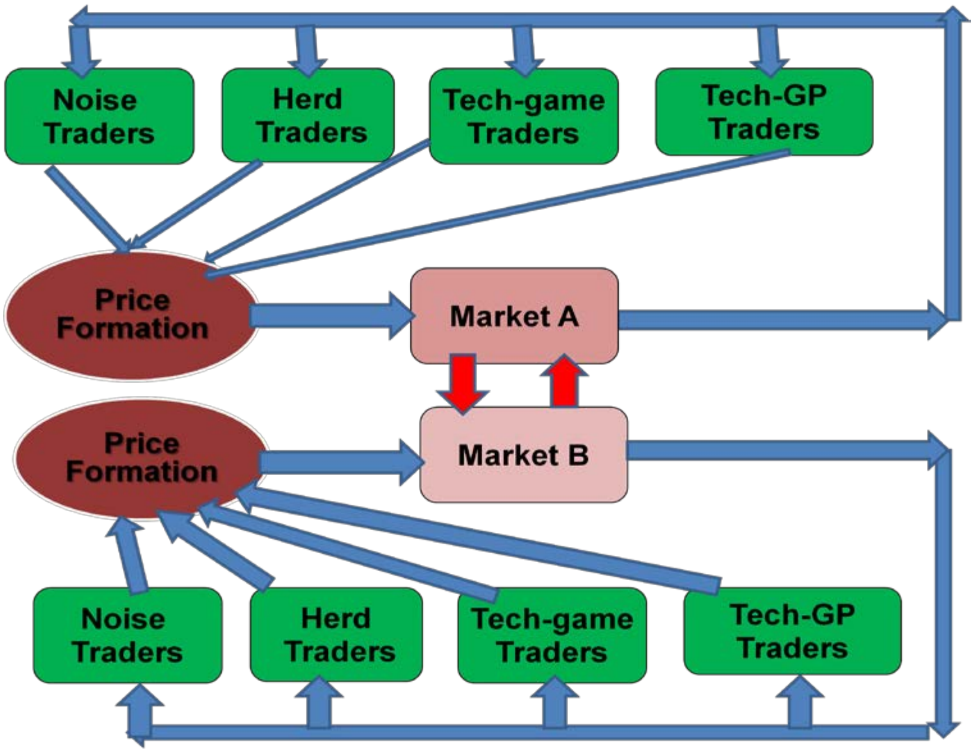

Figure 4-2: Multinational agent-based model

In the artificial multinational market, as pictured in Figure 4-2, each type of player makes decisions based on the information from all markets. For example, in market A, during each trade, after decisions are made, a new market price is calculated with the price formation formula. The price information of both markets, A and B, becomes public information and each player re-evaluates their strategies. If a decision made based on market$B$, a foreign market, is more accurate than that based on market A, the domestic market, the player will have a higher probability of choosing information from market B to make a decision in the next trading period. The same rules apply to players in market$B$.

4.5.1 Model Parameters

The aim of our model is to simulate shifts from interdependence to contagion. In particular, the author target simulation of the effects of the 1997 regional crisis, originating in Thailand, on the South Korea's market. To achieve this purpose, the author needs to optimize the parameters in the multiple-market model. The model is capable of simulating multiple markets. However, the author will now consider two markets and use the real time series for one of the markets to optimise the model parameters while simulating the second market. The second market is simulated in such way that the tail dependence between the simulated time series and the real time series from the first market， is as close as possible to the tail dependence between the real markets. Let us assume that the real market is B and the simulated market is A.

The model parameters then include the proportions of each type of trader - technical-GP, technical-Game, herd, and noise traders - denoted as $N_{GP,}$ $N_{Game}, N_{Herd}, N_{Noise}$, correspondingly. The heterogeneous model's parameters further include the sensitivity of market A towards the order imbalance λ^A in the price formation formula (4.2), and $p_b^{Noise,A}, p_s^{Noise,A}$, $p_h^{Noise,A}$ corresponding to the behaviour of noise traders in market$A$, as well as the sensitivity τ^A in formula (4.12b) of herd traders in market A towards the overall price change on both markets, and the scale factor d in formula (4.13). The other parameters are the memory sizes m_1,$m_2$for the strategies of minority and majority technical-Game players in market A, the number of strategies k_1,k_2 included in a decision table of minority and majority Game players, the scale factor γ^A_{Game} in formula (4.16) relevant to the choice of information market, the time periods L_{MA}, L_{TRB}, L_{VOL} in calculating the technical indicators for GP traders in market A, and a scale factor γ^A_{GP} corresponding to γ^A_{Game} but now relevant to the choice of information market by GP traders. The model parameters are summarised in Table 4-3.

Table 4-4: Symbols and parameters in the artificial market simulating contagion

Symbol	Parameter
N_{GP}	Technical-GP traders proportion
N_{Game}	Technical-Game traders proportion
N_{Herd}	Herding traders proportion
N_{Noise}	Noise traders proportion
$p_b^{Noise,A}$	Probability to buy for noise traders
$p_s^{Noise,A}$	Probability to sell
$p_h^{Noise,A}$	Probability for hold
k_1	strategies for a minority technical-Game player
k_2	strategies for a majority technical-Game player
L_{MA}	Time period for calculating the MA indicators
L_{TRB}	Time period for calculating the TRB indicators
L_{VOL}	Time period for calculating the VOL indicators
γ^A_{GP}	Scale factor for Tech-GP market choosing
γ^A_{Game}	Scale factor for Tech-Game market choosing
m_1	Memory size of minority Technical-Game players
m_2	Memory size of majority Technical-Game players
τ^A	sensitivity to price change for herd traders
λ^A	sensitivity of the market, in price formation, towards the order imbalance

4.5.2 Parameter Optimisation

In order to measure the performances of parameter configurations, the author compare the tail dependence between the two real markets, with the tail dependence generated by the artificial financial market, i.e. the dependence between the simulated market A and real market B. The tail dependence coefficient, as introduced in Chapter 3, is based on the Clayton copula. The fitness function f is then formulated in such way that the fitness of parameter configurations could improve between 0 and 1 , i.e. fitness of 0 to 100%.

$$f(\lambda_{L,Clayton}) = \frac{1}{10}\sum_{j=1}^{10}\left(2\,\frac{e^{-\left|\lambda_{L,Clayton}^{real} - \lambda_{L,Clayton}^{sim,j}\right|}}{e^{-\left|\lambda_{L,Clayton}^{real} - \lambda_{L,Clayton}^{sim,j}\right|}+1}\right) \quad , \qquad (4.17)$$

where the left tail dependence between the real markets is denoted as $\lambda_{L,Clayton}^{real}$, while the tail dependence coefficients generated through simulations of the artfifcial market model is denoted as $\lambda_{L,Clayton}^{sim,j}$. At each step of the optimization procedure, the author run 10 simulations with the same set of parameters, i.e. $(j = 1, \ldots, 10)$ as indicated in formula (4.17). The average fitness over the 10 simulations is then assigned as the fitness of that parameter configuration. The author then modify the parameters, and re-evaluate the fitness function. The process stops if the fitness function approaches 1 or the maximum number of iterations is reached. Next in Chapter 5, the author develops a hybrid evolutionary algorithm capable of optimizing the artificial financial market introduced here.

4.6. Conclusion

In this chapter the author develop an agent-based financial market model comprising four types of traders - technical-GP, technical-Game, herd and noise traders – who make buy, hold and sell decisions, based on different characterising strategies. The developed model also represents multiple markets rather than a single market, in order to simulate how financial contagion happens and observe the evolution of interactions between markets. Several features of our model are expected to contribute to the capability of simulating real markets and even the complex phenomenon of contagion. First, the model involves agents with different type of behaviour, and even different strategies within the same type of behaviour, which provides for a richer, complex, and more realistic simulated system. Second, the strategies incorporate information from two (multiple) markets rather than a single market; thus the simulated market's links emerge through the underlying behaviours of agents. Third, instead of following previous studies and using a linear measure of how markets are co-related with each other, the author introduce the non-linear Clayton copula function and the Clayton tail-dependence coefficient to measure non-linear interdependence between non-normally distributed financial data. That measure captures more realistic cross-market links, and it is used in the fitness function for optimizing our model parameters. In the next chapter, the author will develop hybrid optimization techniques to explore the parameters used in our model.

Chapter 5 : Immune-Particle Swarm Optimization Algorithm

The agent-based artificial market the author have introduced in the Chapter 4 involves different types of traders, information from more than one market, and a set of parameters to be optimised for the overall problem, while the technical traders also optimise their strategies. The complexity of the model requires developing an optimisation algorithm capable of handling the task and achieving good convergence. As the author discussed in Chapter 2, different optimization techniques have their advantages and disadvantages. The author propose here an Immune Particle Swarm Optimization (Immune-PSO) algorithm, which is combined with an Immune Clone Selection algorithm. Clone copy, clone hyper-mutation and clone selection operations are performed during the evolutionary steps of the model. Cloning individual particles in proportion to their affinity can protect high fitness individuals and speed up convergence. Clone hyper-mutation provides a mechanism producing new particles and maintaining diversity. Clone selection, which selects the best individuals, avoids degenerating algorithm’s effectiveness.

5.1 Immune-PSO

During the PSO search process, if one particle finds a temporary best solution, all the other particles will tend to draw close to it, thus causing the phenomenon of clustering, and leading to a decrease in the diversity of the population. If the temporary best solution is a local best solution, searching in other spaces becomes less possible, and premature convergence to local optima occurs. The author introduce an immune clone algorithm which overcomes this drawback of the standard PSO algorithm.

Figure 5-1 presents the flowchart of the Immune-PSO. First, a group of particles are initialized and the algorithm calculates the fitness of each particle, picking the best solution pbest of each individual as well as identifying the global best solution gbest. After that, the immune clonal algorithm runs. The author views the particles as antibodies and calculates their affinity. This is followed by clone copy, hyper-mutation, and clone selection operations. Finally, the author updates the particles' (antibodies) speed and position. If the results meet the terminal condition, the author displays the result and end processing.

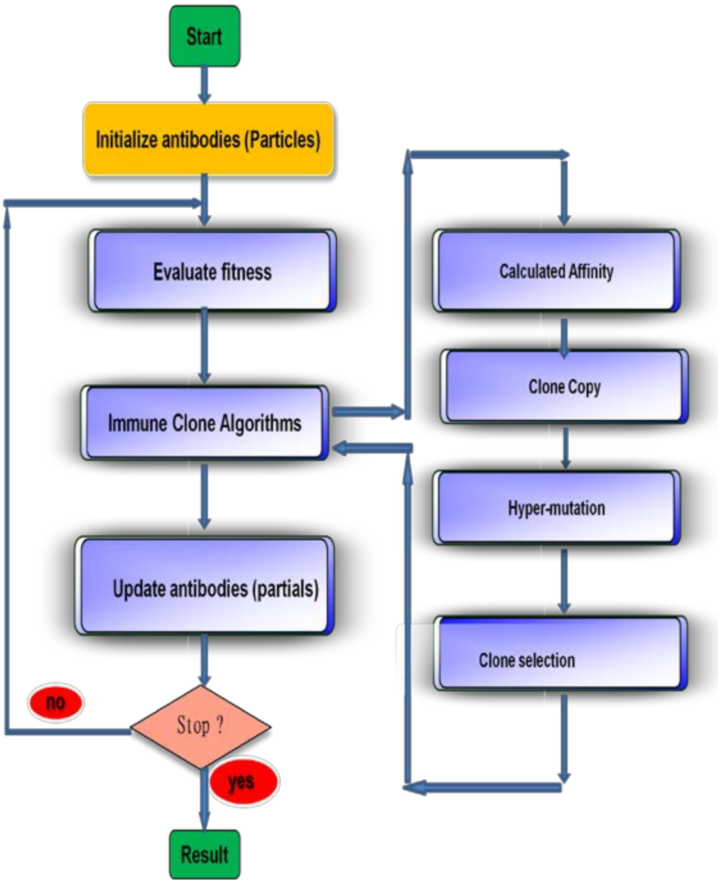

Figure 5-1: Flowchart of Immune-PSO

5.2 Implementation of the Immune-PSO Algorithm

5.2.1 Calculating the Affinity of Particles

Affinity is the criterion used to measure the goodness of each antibody in the population. If the author views particles as antibodies, then to calculate an antibodies' affinity means to calculate a particles' affinity. The affinity takes both the fitness and particle's position into consideration, and its formula is as follows:

$$affinity_i = \frac{fitness_i}{dis_i + 1} \quad , \qquad (5.1)$$

where dis_i is the distance between particle i and the global best solution gbest

$$dis_i = \sqrt{\sum_{j=1}^{DIM} \left(p_{ij} - gbest_j\right)^2} \quad . \qquad (5.2)$$

Here, p_{ij} and $pbest_j$ are the position of particle i and the global best particle in dimension j, correspondingly. The author can see from formula (5.2) that the closer the particle to global best, and the higher the fitness, the larger the affinity.

5.2.2 Clone Copy

The number of copies of each individual is calculated in proportion to its affinity. The number of copies of the ith individual is:

$$num_i = \left| \frac{affinity_{\ i}}{\sum_{i=1}^{n} affinity_{\ i}} \right| .m \quad , \qquad (5.3)$$

where m is the size of the population. The larger the affinity, the better the individual, and thus the more offspring it will clone. Therefore, the superior individual is preserved and propagated, and this accelerates the speed of convergence.

5.2.3 Clone Hyper-mutation

For each copy of an individual, a probability is assigned to determine whether to execute hyper-mutation. During the course of evolution, the diversity of the population will decrease rapidly, as all the particles gather around the best one. If the best solution is a local best solution, the result will converge to a local optimum. Hyper-mutation can help to avoid getting stuck on the local best value.

Hyper-mutation, namely Gaussian Mutation (GM), together with Cauchy Mutation (CM) has already been successfully applied to Evolutionary algorithms (e.g.Hinterding, 1996; Radha,et al., 2009). This chapter uses both approaches. GM is used for small length mutation; CM is used for large length mutation, and the decision on which one to choose depends on the individual's fitness. GM helps to promote accuracy, and CM helps to avoid local best value.

(1)Gaussian Mutation (GM(Hinterding, 1996)): following GM, p_i will be replaced by$p_i^{'}$

$$p_i^{'} = p_i + N(0,1) \quad , \quad (5.4)$$

where $N(0,1)$ is the standard normal distribution.

(2)Cauchy Mutation (CM)(Radha,et al., 2009): Cauchy density function is defined as

$$f_{Cauchy}(x) = \frac{t}{\pi(t^2+x^2)} \quad , \quad (5.5)$$

where $t > 0$ is a scale factor. Then,

$$p_i^{'} = p_i + \rho\sigma_k \quad , \quad (5.6)$$

where $\sigma_k$is generated by Cauchy function, and ρ is an adjusting factor.

5.2.4 Clone Selection

After clone copy, clone hyper-mutation is carried out, and the author then chooses the best individuals to constitute the next generation. The parent solutions, and the offspring solutions following copy and mutation, are both considered at the selection stage. Figure 5-2 presents the process, where $p_1, p_2,\ldots,p_m$ correspond to the parent population, and p_1^*, $p_2^*,\ldots,p_m^*$ present the new population.

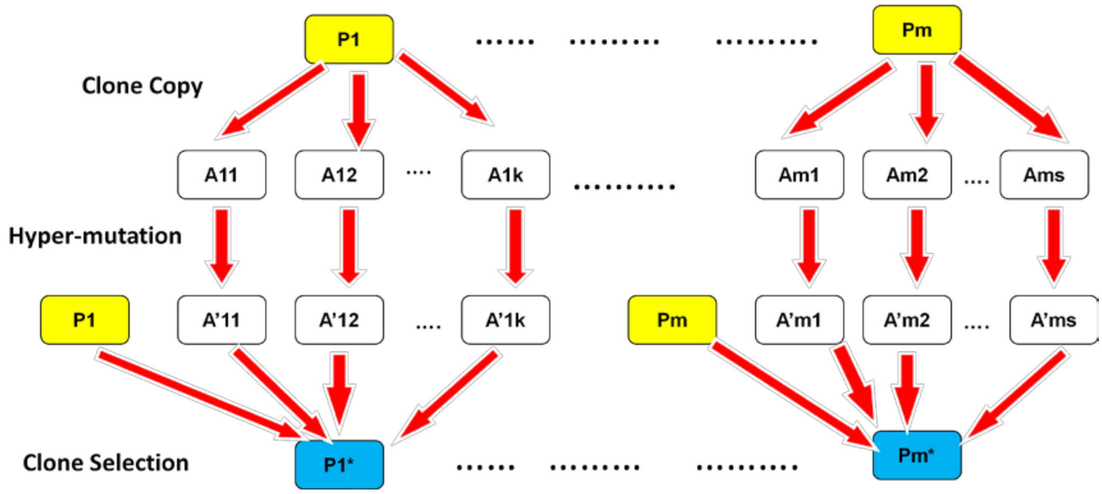

Figure 5-2: Main operations of the clone selection algorithm

5.3 Performance Assessment

The author will test our algorithm on a set of benchmarks, including the following functions: Branin, Shekel's Foxholes, Goldstein-Price, generalized Rosenbrock's, Schwefel's, and Six-Hump Camel-Back function.

5.3.1 Testing Functions

（1）Branin Function

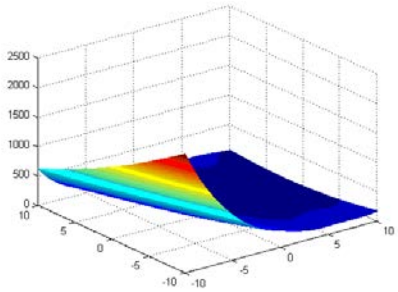

Figure 5-3: Branin function

$$f(X)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10,$$
$$-5\le x_1\le 10, 0\le x_2\le 15$$

$$\min(f(X^*))=f(-3.142,2.275)=f(3.142,2.275)$$
$$=f(9.425,2.425)=0.398$$

（2）Shekel's Foxholes Function

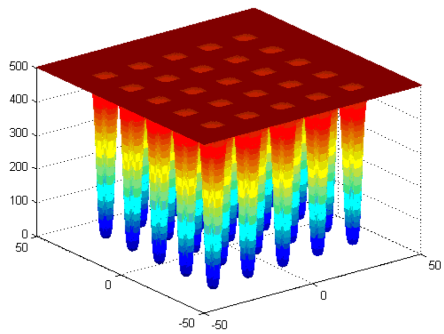

Figure 5-4: Shekel's Foxholes function

$$f(X) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right]^{-1}, \qquad |x_i| \le 65.56$$

$$(a_{ij}) = \begin{pmatrix} -32, -16, & 0, & 16, & 32, & -32, -16, & 0, & 16, & 32, \\ -32, -32, -32, -32, -32, -16, -16, -16, -16, -16, \end{pmatrix.$$

$$\begin{matrix} -32, -16, 0, 16, 32, -32, -16, 0, 16, 32, -32, -16, 0, 16, 32 \\ 0, \quad 0, \; 0, 0, \; 0, \; 16, \; 16, 16, 16, 16, \; 32, \; 32, 32, 32, 32 \end{matrix}\right)$$

$$\min(f(X^*)) = f(-32, -32) \approx 1$$

（3）Goldstein-Price Function

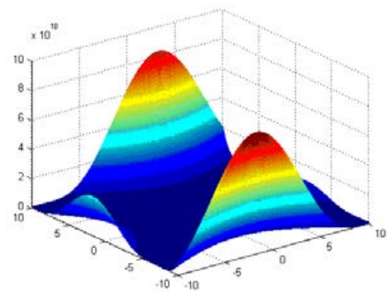

Figure 5-5: Goldstein-Price function

$$\begin{aligned} f(X) = {} & [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \\ & \times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)], \\ & |x_i| \le 2 \end{aligned}$$

$$\min(f(X^*)) = f(0, -1) = 3$$

（4）Generalized Rosenbrock's Function

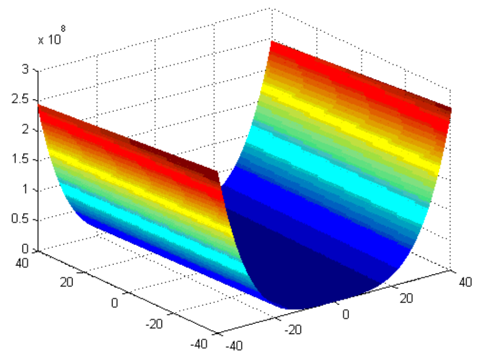

Figure 5-6: Generalized Rosenbrock's function

$$f(X)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(1-x_i)^2], \qquad |x_i|\le 30$$

$$\min(f(X^*))=f(1,1,\cdots,1)=0$$

（5）Schwefel's function

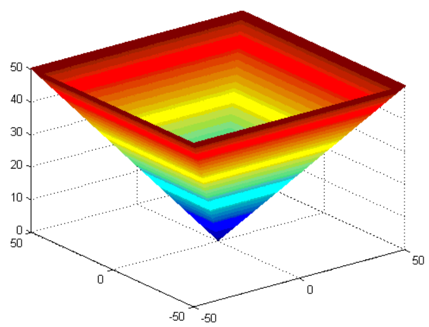

Figure 5-7: Schwefel's function

$$f(X)=\max_{i=1}^{n}\{|x_i|\}, \qquad |x_i|\le 100$$

$$\min(f(X^*))=f(0,0,\cdots,0)=0$$

（6）Six-Hump Camel-Back Function

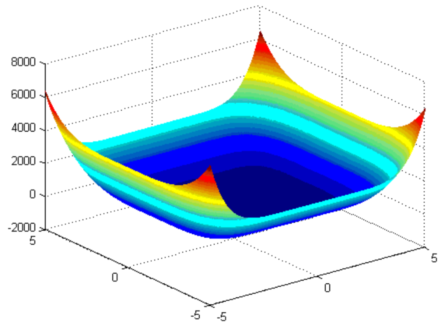

Figure 5-8: Six-Hump Camel-Back function

$$f(X) = 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1x_2 - 4x_2^2 + 4x_2^4, \qquad |x_i| \leq 5$$

$$\begin{aligned} \min(f(X^*)) &= f(0.08983, -0.7126) \\ &= f(-0.08983, 0.7126) \\ &= -1.0316285 \end{aligned}$$

5.3.2 PSO compared to Immune-PSO

Settings for a generic PSO

Learning factor c1=1.4962
Learning factor c2=1.4962
Maximum iteration time =1000;
Population size =40;
Stopping criteria: eps=10^ (-6);

Table 5-1: Performance of standard PSO

Function	Branin	Foxhole	GoldsteinPrice	Rosenbrock	Schwefel	SixHump
Actual	0.398	0.998	3	0	0	-1.031
PSO	0.4	1.005	3	0	0.0001	-1.031
MSE	4.00E-06	4.90E-05	0	0	0	0

Table 5-2 Performance of Immune-PSO

Function	Branin	Foxhole	GoldsteinPrice	Rosenbrock	Schwefel	SixHump

Actual	0.3979	0.998	3	0	0	-1.031
I-PSO	0.3979	0.998	3	0	0.0001	-1.031
MSE	0	0	0	0	1E-08	0

The author can see from the above comparison that our Immune-PSO outperforms the standard PSO in function Branin and Foxhole. The standard PSO gets stuck in the local optimum while our Immune PSO find out the global optimum solution successfully.

5.3.3 Experiment setting for GA

The author compares the performance of our Immune-PSO algorithm in the benchmark functions to that of the GA toolbox in MATLAB 7.8. In order to choose the best GA setting for the test functions, the author designed a procedure considering three key operators - mutation, crossover, selection – and their combinations. Detailed results for the 18 groups of settings are provided in the appendix.

Mutation: Gaussian mutation, uniform mutation, adapt feasible mutation
Crossover: one point crossover, two point crossover, intermediate crossover
Selection: stochastic universal selection, roulette wheel selection

Table 5-3: Ranked performance for each group

Group Number	Running Time	Error
3	38.9627	0
6	39.7517	0
9	44.8487	0
18	45.8885	0
17	47.5795	1E-04
15	43.3207	0.0001
12	41.5987	0.0001
8	47.009	0.0001

7	43.9643	0.0002
16	48.0341	0.0005
10	44.9437	1.0814
1	51.2532	1.1941
2	23.4252	21.6722
4	37.8866	28.6966
11	42.7292	29.3478
13	50.4376	35.0674
14	40.6572	90.8184
5	39.4206	158.219

The author has two criteria to measure the performance of an optimization technique, running time and accuracy. In order to find the best experiment setting, the author give priority to accuracy and the above table is sorted by accuracy in terms of absolute error. The running time corresponds to the total average running time for each test function. The author can see from the table that groups 3, 6,9,18 all perfectly captured the global optimum, but group 3 achieves this in minimum time (38.9627). The setting of group 3 will be used when comparing GA and Immune-PSO.

Setting of group 3:
Stochastic Universal Selection
Gaussian Mutation
Stall Generation Limit: 30
Population Size: 1000
Intermediate Crossover

Table 5-4: Performance of Group 3

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.3979	6.869	0
Foxhole	0.998	0.998	7.7103	0
GoldsteinPrice	3	3	6.1596	0
Rosenbrock	0	0	6.052	0
Schwefel	0	0	5.9968	0
SixHump	-1.0316	-1.0316	6.175	0

	38.9627	0

When the author compare the performance of each group of setting for an individual function, the author find that group 6(Stochastic Universal Selection, Uniform Mutation, Intermediate Crossover) is suitable for functionsBranin, Rosenbrock and SixHump while group 3 (Stochastic Universal Selection, Gaussian Mutation, Intermediate Crossover) is suitable for functions Foxhole, GoldsteinPrice and Schwefel.

Table 5-5 is a summary of appendix B. It indicates that Stochastic Universal Selection and Intermediate Crossover can help improve the performance of GA, since they appear in groups 3, 6 and 9. Intermediate Crossover appears in all four best groups (3, 6, 9, and 18).

Table 5-5: Best player for each testing function

Function	Group Number	Running Time	Error
Branin	6	5.9438	0
Foxhole	3	7.7103	0
GoldsteinPrice	3	6.1596	0
Rosenbrock	6	6.0184	0
Schwefel	3	5.9968	0
SixHump	6	6.1273	0

5.3.4 Immune-PSO compared to GA

Table 5-6: Performance of Immune-PSO (population size 10000)

Function	Branin	Foxhole	GoldsteinPrice	Rosenbrock	Schwefel	SixHump
Actual	0.3979	0.998	3	0	0	-1.031
PSO	0.3979	0.998	3	0	0	-1.031

MSE	0	0	0	0	0	0
Running Time	84.352	110.856	90.358	96.658	88.534	107.851

Table 5-7: Performance of Immune-PSO (populaion size 1000)

Function	Branin	Foxhole	GoldsteinPrice	Rosenbrock	Schwefel	SixHump
Actual	0.3979	0.998	3	0	0	-1.031
PSO	0.3979	0.998	3	0	0	-1.031
MSE	0	0	0	0	0	0
Running Time	7.615	8.964	7.113	7.542	7.634	6.997

Table 5-8: Performance of Immune-PSO (population size 100)

Function	Branin	Foxhole	GoldsteinPrice	Rosenbrock	Schwefel	SixHump
Actual	0.3979	0.998	3	0	0	-1.031
PSO	0.3979	0.998	3	0	0	-1.031
MSE	0	0	0	0	0	0
Running Time	7.615	8.964	7.113	7.542	7.634	6.997

Table 5-9: Performance of Immune-PSO (population size 10)

Function	Branin	Foxhole	GoldsteinPrice	Rosenbrock	Schwefel	SixHump
Actual	0.3979	0.998	3	0	0	-1.031
PSO	0.3979	0.998	3	0	0	-1.031
MSE	0	0	0	0	0	0
Running Time	5.615	6.644	6.128	7.542	5.654	6.319

Table 5-10: Performance of GA (population size 10000)

Function	Branin	Foxhole	GoldsteinPrice	Rosenbrock	Schwefel	SixHump
Actual	0.3979	0.998	3	0	0	-1.031
GA	0.3979	0.998	3	0	0	-1.031
MSE	0	0	0	0	0	0
Running Time	78.869	70.7103	76.1596	76.852	75.9968	76.175

Table 5-11: Performance of GA (population size 1000)

Function	Branin	Foxhole	GoldsteinPrice	Rosenbrock	Schwefel	SixHump
Actual	0.3979	0.998	3	0	0	-1.031

GA	0.3979	0.998	3	0	0	-1.031
MSE	0.0999824	136.24726	2.4542356	0.039442	2.131016	1E-08
Running Time	7.653	8.124	6.587	7.845	8.457	6.954

Table 5-12: Performance of GA (population size 100)

Function	Branin	Foxhole	GoldsteinPrice	Rosenbrock	Schwefel	SixHump
Actual	0.3979	0.998	3	0	0	-1.031
GA	0.3979	1.992	3	0.0082	0	-1.0316
MSE	0	0.988036	0	6.724E-05	0	3.6E-07
Running Time	6.869	7.7103	6.1596	6.052	5.9968	6.175

Table 5-13: Performance of GA (population size 10)

Function	Branin	Foxhole	GoldsteinPrice	Rosenbrock	Schwefel	SixHump
Actual	0.3979	0.998	3	0	0	-1.031
GA	0.7141	12.6705	4.5666	0.1986	1.4598	-1.0311
MSE	0.0999824	136.24726	2.4542356	0.039442	2.131016	1E-08
Running Time	5.457	4.689	5.004	6.398	6.154	5.954

Table 5-14: Summary of the performance of I-PSO with different Population size

Population	10000	1000	100	10
MSE	0	0	0	0
Running Time	578.609	45.865	45.865	37.902

Table 5-15: Summary of the performance of GA with different Population size

Population	10000	1000	100	10
MSE	0	0	0.9881036	140.97193
Running Time	454.7627	45.62	38.9627	33.656

All results in the tables are the average of 100 runs, for each test function. The true minimum value of each function is indicated in the first row. MSE stands for the mean squared error of the value obtained through each algorithm. When use GA on the complex problem, GA is tuned. GA in matlab is by no means the original basic GA, it has been revised by thousands of people, and the author can guess from the size of GA toolbox that it is a hybrid algorithm to certain extent despite the fact that it still called GA. that is why it is no worse than our I-pso. As I calculated in the viva, it takes i-pso about four days to produce a result and about 3 days for i-pso with population size of 10. You may not surprise as your previous student takes several days to get a result and our model is more complex than that one. It takes 6-9 seconds for GA with a population size of 1000 to produce a result. Intuitively, this is incredible as the author assure it takes 100 times as long as a population size of 10. However, further experiment show that it takes 70-90 seconds to produce a result with a population size of 10000 which takes ten times as long as size of 1000. The conclusion got is there is a non-linear relationship between population and running time, so does the complexity of test function. And there may be a joint effect of complexity and population size. That means it is possible that the running time will significantly increase when the population size is less 1000, say 100 or even less. Although the author cannot test the second assertion as it takes unreasonably lone time to test it (may be years), the author could still come to the conclusion that GA with population size of 1000 takes much longer and is not suitable for our model.

5.3.5 Optimization of the Market Model

Having tested the Immune-PSO on benchmark functions and confirmed its general performance, the author now apply the algorithm to the complex artificial market model. The optimisation is performed using real data for 222 trading days during the

Asian crisis of 1997, divided in equal trading periods around the crisis point. Table 5-16 presents the optimum parameter configurations achieved through I-PSO.

Table 5-16: Optimum parameter values for the simulation of South Korea's market

Symbol	Represents	Parameters, I-PSO
N_{GP}	Technical-GP traders proportion	0.14
N_{Game}	Technical-Game traders proportion	0.34
N_{Herd}	Herding traders proportion	0.32
N_{Noise}	Noise traders proportion	0.20
$p_b^{Noise,A}$	Probability to buy for noise traders	0.33
$p_s^{Noise,A}$	Probability to sell for noise traders	0.29
$p_h^{Noise,A}$	Probability for hold for noise traders	0.38
k_1	strategies for a minority technical-Game player	30
k_2	strategies for a majority technical-Game player	52
L_{MA}	Time period for calculating the MA indicators	8
L_{TRB}	Time period for calculating the TRB indicators	14
L_{VOL}	Time period for calculating the VOL indicators	20
γ_{GP}^{A}	Scale factor for Tech-GP market choosing	24
γ_{Game}^{A}	Scale factor for Tech-Game market choosing	31
m_1	Memory size of minority Technical-Game players	30
m_2	Memory size of majority Technical-Game players	56
τ^A	sensitivity to price change for herd traders	34
λ^A	sensitivity of the market, in price formation, towards the order imbalance	4.3

Table 5-17 also presents the characteristics of the real and simulated South Korea's index return distribution, as well as the real and simulated dependence with Thailand's market. The dependence is measured through Kendal's tau rather than the correlation coefficient, following the argument in Chapter 3 for better capturing market dependence.

Table 5-17: Real and simulated dependence between South Korea's and Thailand's

Target Value	Real	I-PSO
Kurtosis of daily return distribution	3.08	7.54
Volatility	63.7	47.6
Kendal's tau for the pre-crisis phase	-0.4334	-0.2143
Kendal's tau during the crisis phase	0.7328	0.2314

The real and simulated market indices of South Korea, along with the real Thai index, are compared in Figure 5-9.

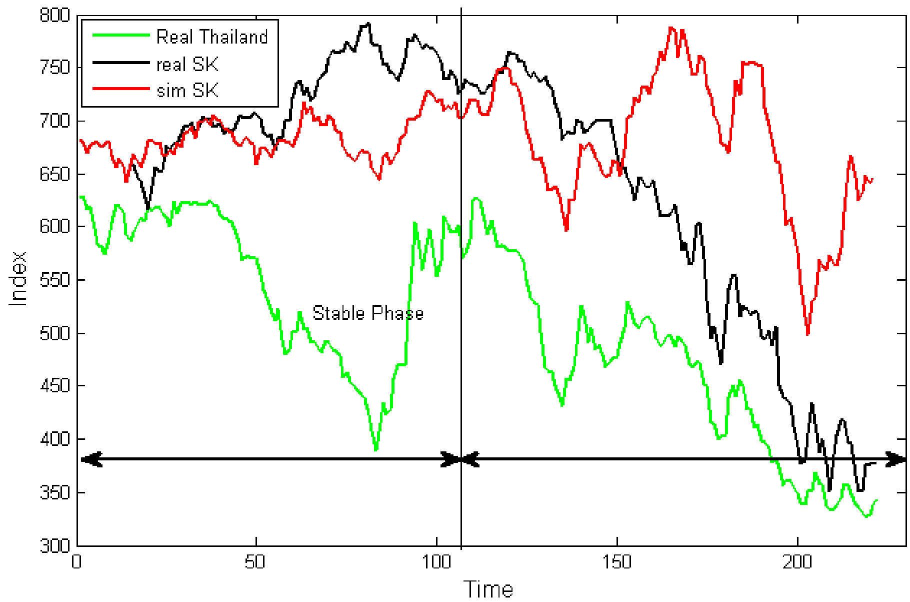

Figure 5-9: Immune-PSO - a comparison of the simulated and real market indices of South Korea and the real Thai index, from 25/02/1997 to 31/12/1997

Analysing the chart, the author find that in the pre-crisis phase, the simulated index relatively well approximates the real time series. The pattern however differs in the crisis phase. An issue here may be that the model assumes that the number of each type of traders remains the same throughout the experiment. This may not be the case. When a crisis happens, more and more rational traders become herd traders, e.g. sell all their shares to avoid loss and push the price further down.

5.4 Conclusion

In this chapter, the author have developed a sophisticated optimization technique, which is accurate on benchmark functions, and capable to approach the complex market model introduced in Chapter 4. The proposed technique is a hybrid algorithm, namely, an Immune Particle Swarm Optimization (Immune-PSO) algorithm, which includes Immune Clone Selection. Thus, clone copy, clone hyper-mutation and clone selection operations are performed during the evolutionary steps in optimising the model. Cloning individual particles in proportion to their affinity can protect high fitness individuals and speed up convergence. Clone hyper-mutation provides a new mechanism producing new particles and maintaining diversity. Clone selection, which selects the best individuals, can avoids degenerating algorithm's effectiveness. The typical benchmark functions are performed and the result was compared with GA using MATLAB GA Toolbox with appropriate setting. The results indicate that our technique performs at least as good as GA, and can be a reliable technique for the optimization of the agent based model. The optimisation and simulation results, however, reveal that the agent model follows reasonably well the market in the pre-crisis period, but fails to capture financial contagion during the crisis phase. The author consider as a reason that the number of each type of traders in the model never change, which is not the case in reality. To address the issue, the author introduce a mechanism allowing that technical traders could change their status during the simulation experiment, and relate that model feature to the observation that when a crisis happens, more and more rational traders become herd traders.

Chapter 6 : Financial Contagion: A Propagation Mechanism

6.1 Background of the Asian crisis of 1997

There are two views towards the cause of the 1997 Asian crisis. One is that the panicand inadequate policy responses triggered a region-wide financial crisis and the economic disruption that followed (Sachs and Radelet, 1998).An alternative view is that weaknesses in the Asian financial systems were at the root of the crisis (e.g. Moreno, Pasadilla, and Remolona, 1998). Although the two implications vary greatly, the two views are not mutually exclusive. Both causes contributed to the crisis.

The economic shocks affecting East Asia at the time were followed by "runs" on the financial systems and currencies. Even well-managed banks or financial intermediaries are vulnerable to panics, because they traditionally engage in maturity transformation. That is, banks accept deposits with short maturities (say, three months) to finance loans with longer maturities (say, a year or longer). Maturity transformation is beneficial because it can make more funds available to productive long-term investors than they would otherwise receive. Outside crisis periods, banks have no problem managing their portfolios to meet expected withdrawals. However, if all depositors in panic decided to withdraw their funds from a given bank at the same time, the bank would not have enough liquid assets to meet its obligations, threatening the viability of an otherwise solvent financial institution. As pointed out by Radelet and Sachs (1998), East Asian financial institutions had incurred a significant amount of external liquid liabilities that were not entirely backed by liquid assets, making them vulnerable to panics. As a result of the maturity transformation, some otherwise solvent financial institutions may indeed have been rendered insolvent because they were unable to deal with the sudden interruption in the international flow of funds.

As investors tested currency pegs and financial systems in the region, those economies with the most vulnerable financial sectors (Indonesia, South Korea, and Thailand) experienced the most severe crises. In contrast, economies with more robust and well-capitalized financial institutions (such as Singapore) did not experience similar disruptions, in spite of slowing economic activity and declining asset values. Firstly, financial intermediaries were not always free to use business criteria in allocating credit. In some cases, well-connected borrowers could not be refused credit; in others, poorly managed firms could obtain loans to meet some government policy objective. Hindsight reveals that the cumulative effect of such type of credit allocation can produce massive losses. Second, financial intermediaries or their owners were not expected to bear the full costs of failure, reducing the incentive to manage risk effectively. In particular, financial intermediaries were protected by implicit or explicit government guarantees against losses, because governments could not bear the costs of large shocks to the payments system (McKinnon and Pill, 1997). The importance of implicit government guarantees in the most affected economies was highlighted by the generous support given to financial institutions experiencing difficulties. For example, in South Korea, the very high overall debt ratios of corporate conglomerates (400% or higher) suggested that these borrowers were ultimately counting on government support in case of adverse outcomes. That was confirmed by events in 1997, when the government encouraged banks to extend emergency loans to some troubled conglomerates which were having difficulties servicing their debts, and supplied special loans to weak banks. Those responses further weakened the financial position of lenders and contributed to the uncertainty that triggered the financial crisis towards the end of 1997.Since weaknesses in East Asian financial systems had existed for decades and were not unique to the region, why did Asia not experience crises of this magnitude before? Two explanations are likely. First, rapid growth disguised the extent of risky lending. For many years, such growth allowed financial policies shielding firms that incurred losses from the adverse effects of their decisions. However, such policies would make economies highly vulnerable during periods of uncertainty. Second, innovations in information and transactions technologies had linked those countries more closely to the world financial markets in the 1990s, thus increasing their vulnerability to changes in market sentiment.

Closer integration with the world financial markets adds dimensions of vulnerability that are not present in a closed economy. In a closed economy, bad loans caused by risky lending may not lead to a run because depositors know that the government can supply enough liquidity to financial institutions to prevent any losses to depositors. In an open economy, that same injection of liquidity can destabilize the exchange rate. As a result, during periods of uncertainty, runs or speculative attacks on a currency can be avoided only if the holders of domestic assets are assured that the government can meet the demand for foreign currency. Those East Asian economies where foreign exchange reserves were large relative to their short-term borrowing (Philippines, Malaysia, and Taiwan) were in a better position to provide such assurances than those economies where such reserves were relatively low (South Korea, Indonesia, and Thailand).

6.2. Methodology

6.2.1. Introduction

During crisis periods, some of the technical traders in real markets would give up their original trading strategies and become herd traders. In this chapter the author develop further the artificial market introduced earlier into a co-evolutionary market model where technical traders can change their behaviour during crisis periods and make their decisions based on the latest market sentiment rather than their usual criteria.

The strategy-changing process applied here is based on the reasoning in game theory, though the author does not formally apply game theory and consider this as a direction for future research. Let us consider the trading process just before the outset of a crisis, and compare it with the prisoner’s dilemma. If all traders maintain their approach to decision making and strategy choice, then they are all better off, and the author will refer to this as the cooperative setting of the trading process. If all become herd traders then all are worse off and suffer larger losses, due to pushing the prices further down than they would have otherwise gone. The author will refer to that as the non-cooperative setting of the trading process. Nuances here are the mostly-cooperative setting and the mostly non-cooperative setting. In the former, most traders maintain their approach to decision making and strategy choices; while in the latter, most traders follow the latest sentiment. A trader in the mostly non-cooperative setting is on average worse off than a trader in the mostly cooperative setting, again for the reason of pushing the prices further down though not to the limit.

The author can see that in the 1997 Asian crisis, the market portfolio, as represented by the stock market index, lost almost 70% of its assets. The detail here is that a technical trader may not necessarily change his status and follow the market sentiment right after a shock. He would keep observing and only when the long term adverse price change exceeds what he can bear, then he may choose to give up his trading strategy and become a herd trader. As the number of herd traders increases, the depth of the crisis may worsen and affect the recovery. As the herd traders follow the downward trend in the market where the crisis originates, and as our model provides a mechanism linking with other markets and transferring the sentiment, the traders in linked markets are gradually conditioned in their activity by the crisis in the original market. Thus the downward trend spreads to linked markets, leading to a significant increase in the correlation coefficient between markets. This behaviour meets the definition given in Forbes and Rigobon(2002), and contributes to the mechanism causing financial contagion.

6.2.2A Co-evolutionary Mechanism

The author define the probability of status change as follows:

$$Prob_{Tech}(X_t) = \frac{1}{1+e^{X_t}} \tag{6.1a}$$

$$X_t = -PC + resistance = -\left|\Delta price_{1,t} + \Delta price_{m,t}\right| + resistance \tag{6.1b}$$

$$\Delta price_{1,t} = a_1\left(\Delta price_{1,t}^{Domestic} + \Delta price_{1,t}^{Foreign}\right) \tag{6.2}$$

$$\Delta price_{m,t} = a_2\left(\Delta \mathrm{price}_{m,t}^{Domestic} + \Delta price_{m,t}^{Foreign}\right) \tag{6.3}$$

$$\Delta price_{1,t}^{Domestic} = price_t^{Domestic} - price_{\mathrm{t-1}}^{\mathrm{Domestic}} \tag{6.4}$$

$$\Delta \mathrm{price}_{1,t}^{\mathrm{Foreign}} = \mathrm{price}_{t}^{\mathrm{Foreign}} - \mathrm{price}_{t-1}^{\mathrm{Foreign}} \tag{6.5}$$

$$\Delta \mathrm{price}_{m,t}^{\mathrm{Domestic}} = \sum_{i=1}^{m-1} \Delta \mathrm{price}_{t-i}^{\mathrm{Domestic}} =$$

$$= \sum_{i=1}^{m-1} \left(\mathrm{price}_{t-i}^{\mathrm{Domestic}} - \mathrm{price}_{t-i-1}^{\mathrm{Domestic}}\right) \tag{6.6}$$

$$\Delta \mathrm{price}_{m,t}^{\mathrm{Foreign}} = \sum_{i=1}^{m-1} \Delta \mathrm{price}_{t-i}^{\mathrm{Foreign}} =$$

$$= \sum_{i=1}^{m-1} \left(\mathrm{price}_{t-i}^{\mathrm{Foreign}} - \mathrm{price}_{t-i-1}^{\mathrm{Foreign}}\right) \tag{6.7}$$

Formulas (6.1), whereX_t is the composite force of the resistance and price change (PC), meets two criteria:

(a) When the overall price change ($\Delta \mathrm{price}_{1,t} + \Delta \mathrm{price}_{m,t}$) is within limits, a trader has a high probability of his status or strategy-selection remaining unchanged.

(b) When the overall price change ($\Delta\mathrm{price}_{1,t} + \Delta\mathrm{price}_{m,t}$) in absolute value is large enough to exceed the positive constant resistance, a trader has a high probability of changing his status to a herd trader. Here, the author particularly considers large negative price changes corresponding to a crisis period.

In formulas (6.2) and (6.3), a_1 and a_2 are scale factors, $\Delta\mathrm{price}_{1,t}$ is the last price change and $\Delta\mathrm{price}_{m,t}$ is the long-term bias. A trader will factor in his previous memories, which will persist for a while, but gradually fade to be replaced by recent memories. In our model setting, some technical traders will change their behaviour under certain circumstances and join the troop of herd traders. Converting back to technical traders may require an external intervention.

6.2.3. Summary

To simplify the setup - financial contagion occurs when a crisis happens in a foreign market, which causes panic in the domestic market. Then traders start selling stocks to reduce their potential loss, which pushes the price down to levels that trigger a financial crisis in the domestic market. The model gives an initial insight into the financial contagion phenomenon.

6.3 Results and Analysis

The author start the optimization using as initial parameter configuration, the values obtained by the I-PSO in Chapter 5 and presented in Table 5-8. Then the author optimize further, introducing to the set of parameters a_1 and a_2 from formulas (6.2) and (6.3). Notice that now the proportions of different types of technical traders and herd traders are not part of the parameter configuration, as they change throughout a simulation. The proportion of noise traders is still part of the sets of parameters, however. A constant resistance is allocated randomly, as an integer number between 0 and 100, to each technical trader i. The new optimised set of parameters is presented in Table 6-1.

Table 6-1: Optimum parameter values for the simulated South Korean market

Symbol	Represents	Parameters, I-PSO
N_{Noise}	Noise traders proportion	0.11
$p_b^{Noise,A}$	Probability to buy for noise traders	0.33
$p_s^{Noise,A}$	Probability to sell for noise traders	0.26
$p_h^{Noise,A}$	Probability for hold for noise traders	0.30
k_1	Strategies for a minority technical-Game player	28
k_2	Strategies for a majority technical-Game player	49
L_{MA}	Time period for calculating the MA indicators	7
L_{TRB}	Time period for calculating the TRB indicators	15
L_{VOL}	Time period for calculating the VOL indicators	22
γ_{GP}^{A}	Scale factor for Tech-GP market choosing	17
γ_{Game}^{A}	Scale factor for Tech-Game market choosing	24
m_1	Memory size of minority Technical-Game players	24
m_2	Memory size of majority Technical-Game players	51
τ^A	Sensitivity to price change for herd traders	25
λ^A	Sensitivity of the market, in price formation, towards the order imbalance	3.8
a_1	Scale factor for short memory	31
a_2	Scale factor for long memory	42

Next, Figure 6-1 compares the real and simulated market indices of South Korea, using the optimum parameter configuration, along with the real Thai index. The characteristics of the real and simulated South Korea's market are presented in Table 6-2, where Kendal's tau uniquely corresponds to Clayton copula's tail dependence.

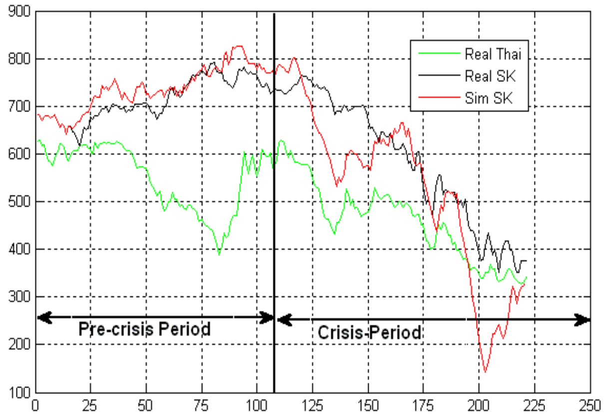

Figure 6-1: Co-evolutionary market - a comparison of the simulated and real market indices of South Korea and the real Thai index, from 25/02/1997 to 31/12/1997

Table 6-2: Real and simulated dependence between South Korea's and Thailand's markets

Target Value	Real	I-PSO
Kurtosis of daily return distribution	3.08	5.43
Volatility	63.7	52.6
Kendal's tau for the pre-crisis phase	-0.4334	-0.4133
Kendal's tau during the crisis phase	0.7328	0.6512

The change is brought by the variable status of traders, which can be observed in Figures 6-2, 6-3 and 6-4. The status profiles for technical-GP traders, technical-Game traders, and herd traders are shown in Figure 6-2. Figure 6-3 is particularly focused on the daily increment of herd traders. Finally, Figure 6-4 presents the net order of the four types of traders, including noise traders.

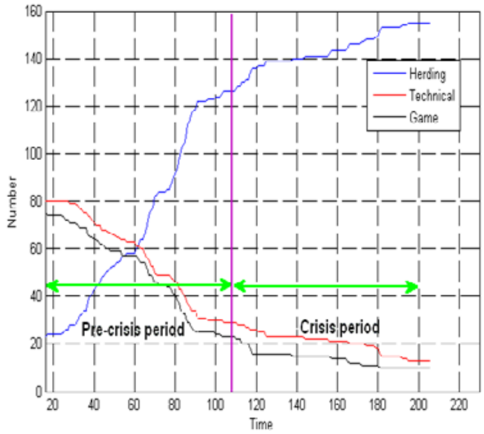

Figure 6-2: South Korea simulation: changes in trader status

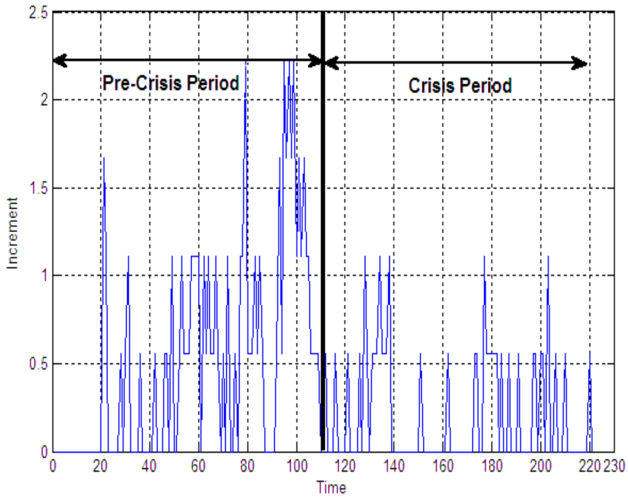

Figure 6-3: Daily increment of herd traders

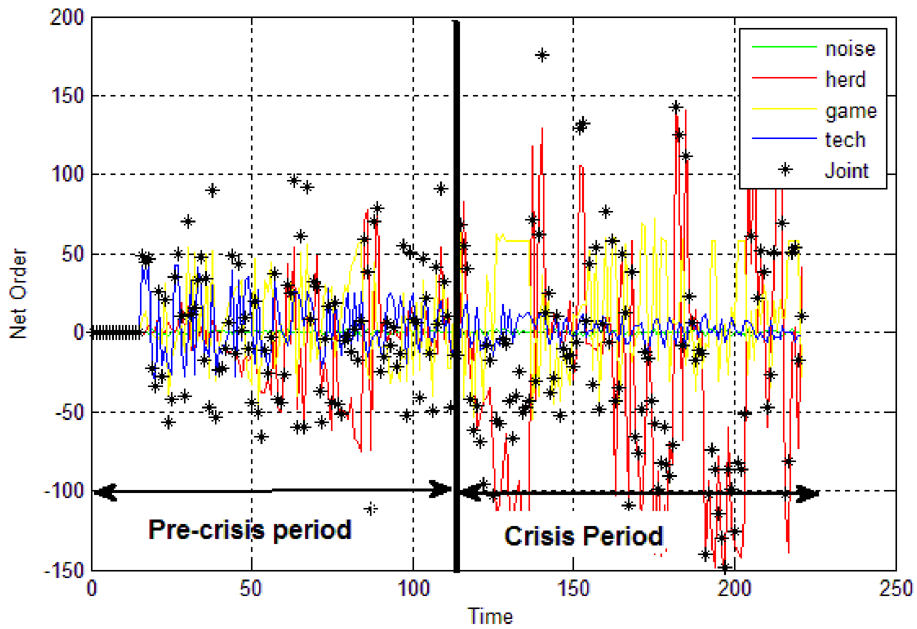

Figure 6-4: Net order of the four types of traders

The simulated price index now follows closer the real one, and the simulated dependence coefficient, in both pre-crisis and crisis period, approximates well the real dependence. Figure 6-3 further demonstrates the daily increase in herd traders. The author can see that the number of herd traders increases from about 1% to almost 2.5%, 10 days before the crisis, then continues to almost 1% increase each day. Figure 6-4 next shows the net orders, i.e. buy order less sell orders, for the four types of traders. Since the net order affects the market price in the model, the author can follow the trend of price change by observing the net order total across all traders. The net order total is represented with black stars in the graph. Initially, the market price is affected by the joint impact of all four types of traders. As the crisis progresses, the red line (herd traders) gets closer to the black star (net order total). Therefore, mostly herd traders' behaviour contributes to market price levels during crisis, though a significant number of other types of traders remain on the market.

Finally, traders' assets, whether noise, herd, technical-Game or technical-GP traders, all suffered a big loss. Total asset value decreased from 10,000 to 3,000, losing almost 70% percent of their value. Since the crisis originated in Thailand, the situation could be worse there. The above analysis reveals that herd traders are a factor in the mechanism of financial contagion. A question to raise here is as follows: since the crisis is caused by market conditions, mainly caused by international currency speculators, beyond the control of individual traders and even their governments, how then can the author recover or better still prevent the crisis from happening? This is a difficult issue, and requires the co-ordination in action and the shared responsibilities of governments.

6.4. Comparing the Results from I-PSO and GA

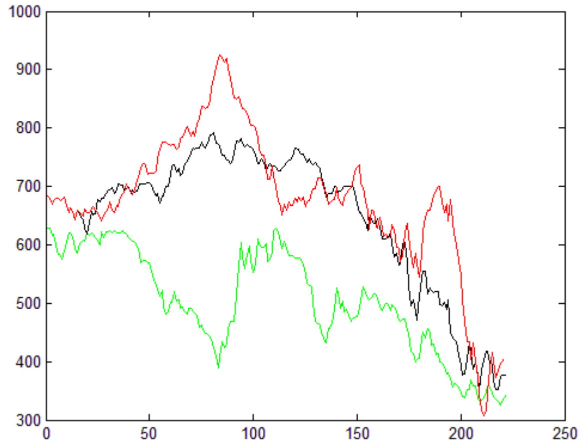

Figure 6-5: GA for SK and Thailand

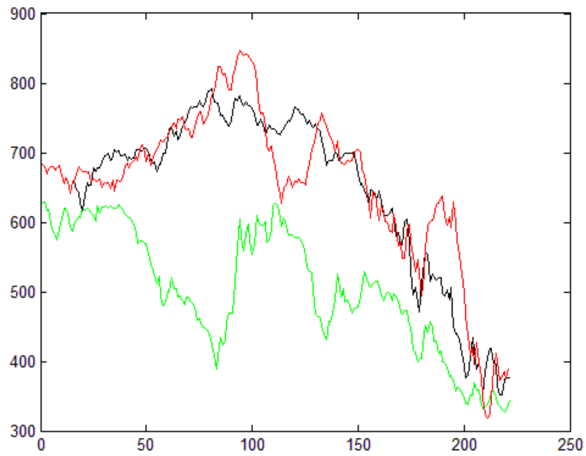

Figure 6-6: I-PSO for SK and Thailand

Table 6-3: Real and simulated dependence between Thailand and SK markets

Target Value	Real	I-PSO	GA

Kurtosis of daily return distribution	3.08	7.54	8.9
Volatility	63.7	47.6	35.6
Kendal's tau for the pre-crisis phase	-0.4334	-0.2143	-02349
Kendal's tau during the crisis phase	0.7328	0.4714	0.4612

The author can see from the results above that I-PSO is better overall.

6.5 Simulated Prediction of Contagion from Thailand to South Korea

As the main goal of this thesis is to model and predict financial contagion, the author optimize in the pre-crisis period using data from the domestic market (South Korea) and the crisis-origin foreign market (Thailand), and predict in the crisis period using data from the foreign market and predicting the affected domestic market.

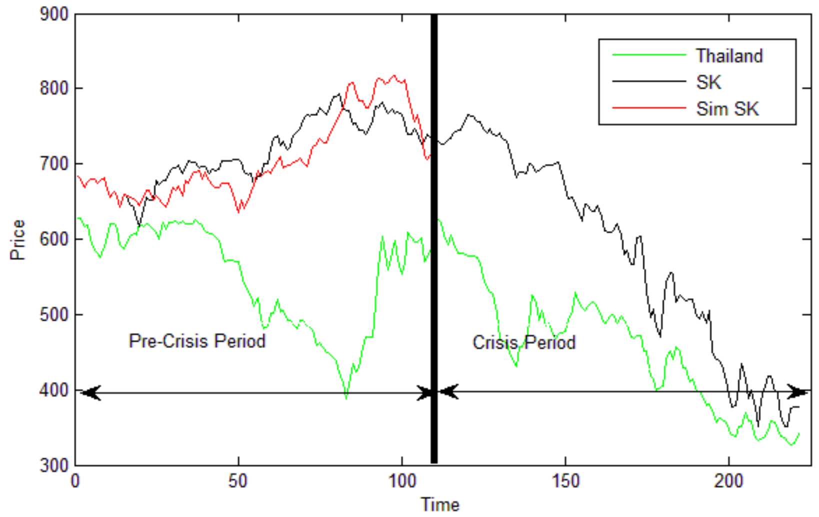

Figure 6-7:Pre-crisis period optimisation-simulation for SK and Thailand

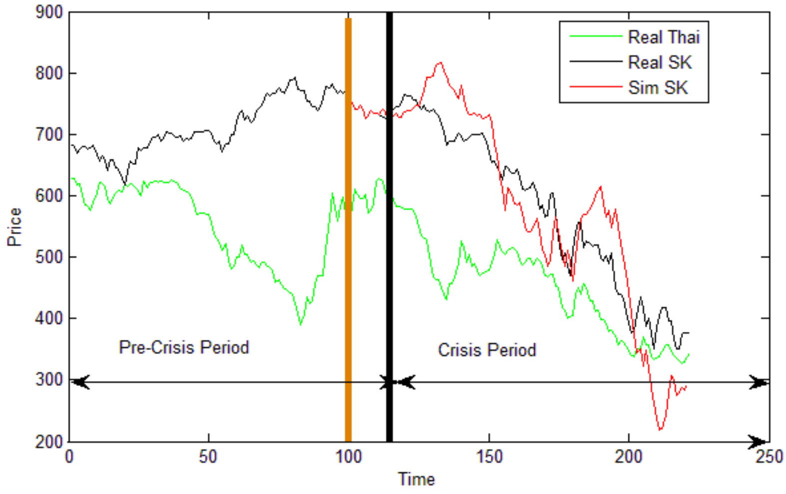

Figure 6-8: Crisis period predictive-simulation for SK and Thailand

Using the model parameters optimised during the pre-crisis period, the author simulate the post-crisis period, and can see from Figure 6-8above that the predictive simulation approaches the real contagion behaviour well.

6.6. Application to the Russian crisis of 1998

6.6.1. Background

The financial crisis was caused by the high fixed exchange rates between the Ruble, the falling productivity and foreign currencies intervention as well as the chronic fiscal deficit. The economic cost of the World War One also contributed to the crisis. Russian economy showed some signs of improvement in the first half of 1997, however, soon after this, problems began to get serious gradually. Two external shocks, which were the Asian financial crisis that began in 1997 and the following dropping demand (and as a result the price also dropped) for crude and non-ferrous metals, had severely influences on Russian's foreign (IMF, 2012). When the East Asian financial crisis broke out in 1997, prices of energy and metals, which were Russia's two most valuable sources of capital flows, plummeted. In view of the fragile economy in Russia, the rapid decline in the value of those two capital sources resulted in an economic chaos in the country where GDP per capita fell, unemployment soared, as well as global investors liquidate their assets in Russia. At the time, Russia employed a "floating peg" policy toward the Ruble, meaning that the Central Bank decided that at any given time the ruble-to-dollar exchange rates would stay within a particular range. If the Ruble threatened to devalue outside of that range or "band", the Central Bank would intervene by spending foreign reserves buying Rubles (Joseph, 2003)

The Russian government has no ability to implement a coherent set of economic reforms has led to the falling confidence of investors and can be compared to the chain reaction severe erosion at the central bank. Investors sell the Rouble and Russia's assets (such as securities); it also brings pressure on the Ruble downward to flee market. Forcing central Banks to use foreign exchange reserves to defend Russia's currency, which in turn further blow to investor confidence, weaken the Ruble. It is estimated that between October 1, 1997 and August 17, 1998, the Central Bank spent approximately $27 billion of its U.S. dollar reserves in order to maintain the floating.

The Russian stock, bonds, and currency markets collapsed On August 13, 1998, as a result of investors fearing that the government would devalue the Ruble or default on domestic debt, or both of the results. Annual production in the Ruble denominated bonds doubled. The stock market had to be closed for half an hour as prices plummeted. When this happened, it was down 65 percent with a small number of shares actually traded. From January to August 1998, the stock market had lost more than 75 percent of its value, 39 percent in the month of May alone. (Kotz, 1998) The nearby economies were also affected, including Ukraine, which we will use in the contagion simulation next.

6.6.2. Results and Analysis

Figure 6-7 compares the real and simulated market indices of Ukraine, using the optimum parameter configuration, along with the real Russian index. The characteristics of the real and simulated Ukraine's market are presented in Table 6-5, where Kendal's tau uniquely corresponds to Clayton copula's tail dependence.

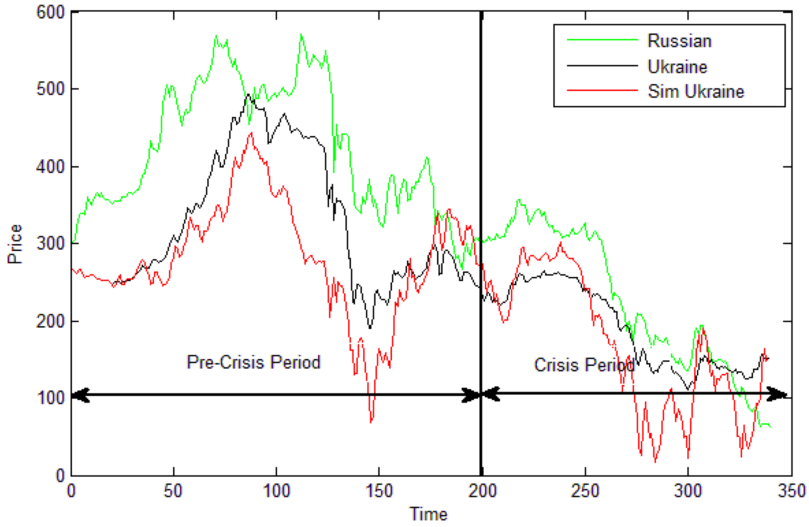

Figure 6-9: Co-evolutionary market - a comparison of the simulated and real market indices of Ukraine, along with the real Russian index, from 28/04/1997 to 04/09/1998

Table 6-4: Optimum parameter values for the simulated Ukraine market

Symbol	Represents	Parameters, I-PSO
N_{Noise}	Noise traders proportion	0.12
$p_b^{Noise,A}$	Probability to buy for noise traders	0.31
$p_s^{Noise,A}$	Probability to sell for noise traders	0.22
$p_h^{Noise,A}$	Probability for hold for noise traders	0.35
k_1	Strategies for a minority technical-Game player	28
k_2	Strategies for a majority technical-Game player	49
L_{MA}	Time period for calculating the MA indicators	7
L_{TRB}	Time period for calculating the TRB indicators	15
L_{VOL}	Time period for calculating the VOL indicators	27
γ_{GP}^{A}	Scale factor for Tech-GP market choosing	17
γ_{Game}^{A}	Scale factor for Tech-Game market choosing	34
m_1	Memory size of minority Technical-Game players	24
m_2	Memory size of majority Technical-Game players	51
τ^A	Sensitivity to price change for herd traders	22
λ^A	Sensitivity of the market, in price formation, towards the order imbalance	4.1
a_1	Scale factor for short memory	41
a_2	Scale factor for long memory	42

Table 6-5: Real and simulated dependence between Ukraine's and Russian's markets

Target Value	Real	I-PSO
Kurtosis of daily return distribution	3.98	4.23
Volatility	43.7	58.6
Kendal's tau for the pre-crisis phase	-0.3314	-0.4133
Kendal's tau during the crisis phase	0.7328	0.6322

The change is brought by the variable status of traders, which can be observed in Figures 6-8, 6-9 and 6-10. The status profiles for technical-GP traders, technical-Game traders, and herd traders are shown in Figure 6-8. Figure 6-9 is focused on the daily increment of herd traders. Figure 6-10 presents the net order of the four types of traders, including noise traders.

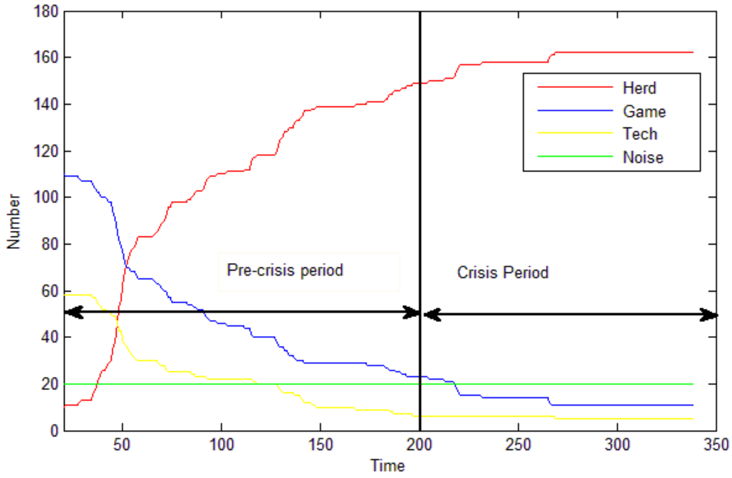

Figure 6-10: Ukraine simulation: changes in trader status

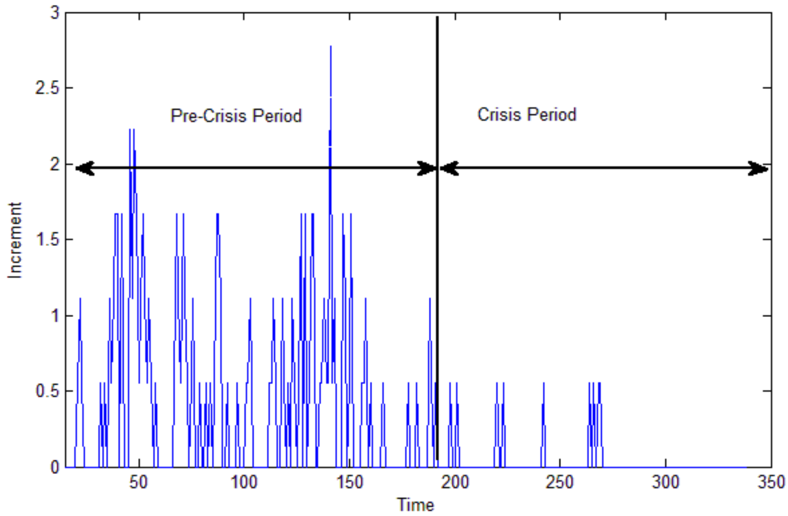

Figure 6-11: Daily increment of herd traders

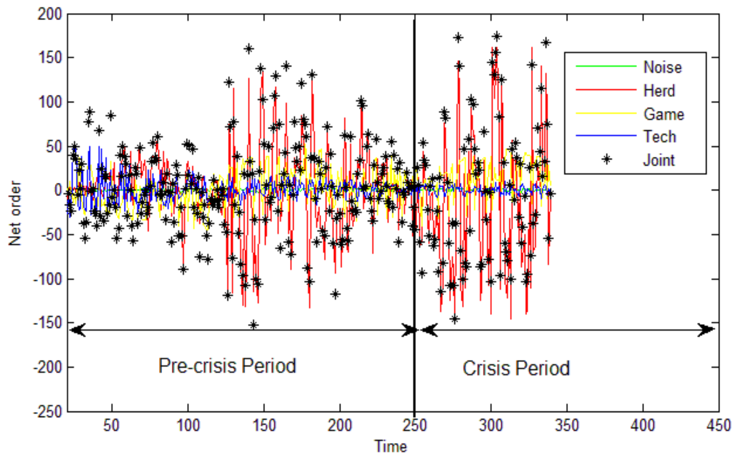

Figure 6-12: Net order of the four types of traders

The simulated price index follows closely the real one, and the simulated dependence coefficient, in both pre-crisis and crisis period, approximates the real dependence.

6.7. Simulated Prediction of Contagion from Russia to Ukraine

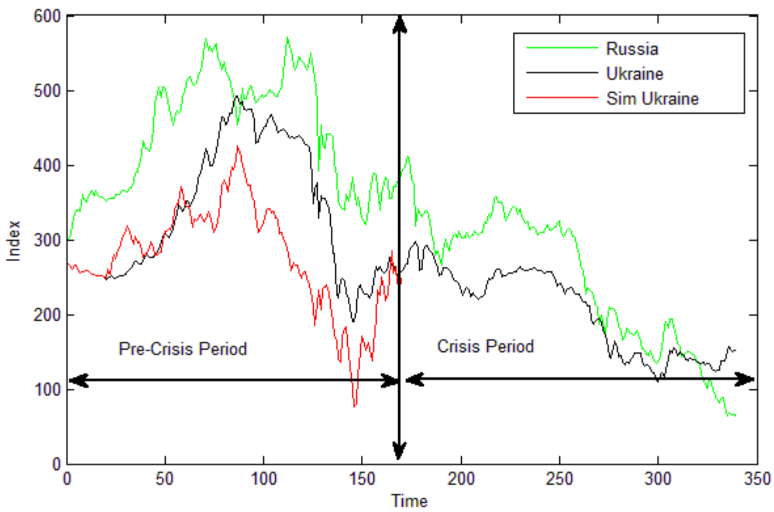

Figure 6-13: Pre-crisis period optimisation-simulation for Russia and Ukraine

The author optimize in the pre-crisis period using data from the domestic market (Ukraine) and the crisis-origin foreign market (Russia), and predict in the crisis period using data from the foreign market and predicting the affected domestic market.

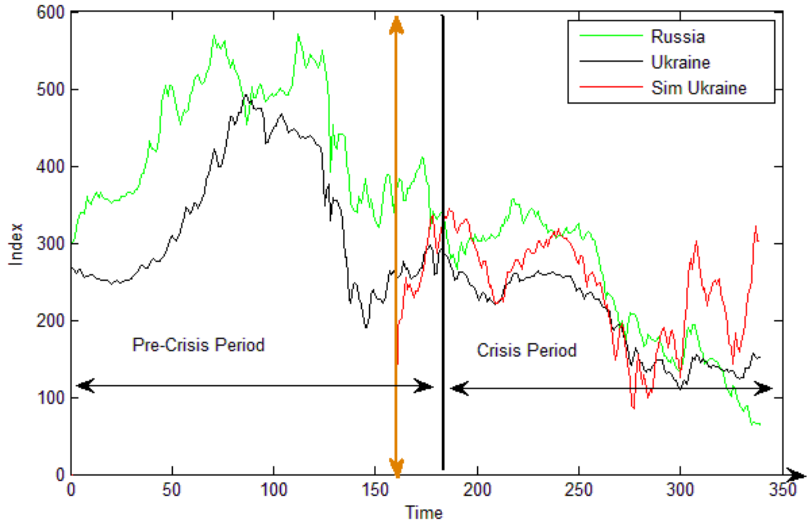

Figure 6-14: Crisis period predictive-simulation for Russia and Ukraine

Using the model parameters optimised during the pre-crisis period, the author simulate the post-crisis period, and as Figure 6-14 above shows, the simulated result captures the pattern of the real contagion behaviour relatively well.

6.8. Conclusion

In this chapter, an overall mechanism is proposed of propagating crisis through contagion. Within that scope, a new co-evolutionary market model is discussed, where some of the technical traders change their behaviour during crisis and rather make their decisions based on market sentiment than on underlying strategies and factors. Thus psychological elements are contributed to the model. After analyzing the interactive behaviour of agents, the author observes that the herd mentality intensifies during crisis.

This chapter is focused on the transformation of market interdependence into contagion, and on the contagion effects. The author first build a multi-national platform to allow different type of players to trade implementing their own rules and considering information from the domestic and a foreign market. Traders' strategies and the performance of the simulated domestic market is trained using historical prices on both markets, and optimizing artificial market's parameters through immune-PSO techniques. The author also introduces psychological elements contributing to the transformation of technical into herd traders. A GARCH-copula is further applied to calculate the tail dependence between the affected market and the origin of the crisis, and that parameter is used in the fitness function for selecting the best solutions within the evolving population of possible model parameters, and therefore in the optimization criteria for contagion simulation.

Our results show that the proportion of herd traders and their decisions increases in the net market order, for optimum contagion simulations. While technical traders 'trading behaviour corresponds to propagating a crisis through interdependence, herd behaviour corresponds to propagating through contagion. If contagion could be avoided or transformed back to interdependence with the effort of national governments and international bodies, a crisis would be more manageable. In that respect, a future focus of research would be to introduce a recovery mechanism into the model and modelling government and international intervention, so that the overall effect is either avoiding the transformation of interdependence into contagion or a recovery from contagion within a manageable time.

Chapter 7 : Conclusions and Outlook

7.1 Summary of Work

The objective of this thesis is to develop a co-evolutionary artificial market, with the purpose of simulating financial contagion between markets, occurring during financial crises. Our work focuses on understanding the characteristics and warning signs of contagion, which will facilitate developing early warning systems. Such warning systems for contagion will help the authorities to implement appropriate management actions faster and therefore more effectively.

The author develops an agent-based model for predictive simulation of financial contagion, and applies to two crisis cases. This approach can be next applied to current data rather than historic data, optimising the model up to the current time and then exploring different scenarios forward for the market the author consider as potential crisis origin, which will produce responding predictive simulation of the domestic market(s) that the author are concerned about being affected through contagion. Scenarios leading to contagion can be identified, as part of continuous monitoring for contagion. Therefore, our model acts as the first step in developing an early warning system for financial contagion.

The way that different types of traders change their behaviour in the model in response to a crisis, allows us to gain an insight into the way contagion develops. The author aims to simulate the spread of financial difficulties from the original market experiencing a crisis, to other markets outside the original crisis zone. The author also aim to optimize the parameters in our model, so that characteristics of markets interactive behaviour are better captured.

The contribution of the work has several aspects, both at level of developing the methodology and at level of empirical implementation. The author develop an artificial co-evolutionary international, instead of national, financial market. The author also suggests how the transfer mechanism operates to propagate the crisis through the market. The author introduces qualitatively different types of traders: technical-GP, technical-Game, herd and noise traders. Each technical-GP trader's decision tree is evolved based on a technical analysis of market data. Each technical-Game player has a distinct set of strategies, and re-evaluates their score according to their success on the market. Both types of technical traders may select to make a particular decision based on the information from the domestic or the foreign markets. Technical traders may further transform into herd traders. Each herd trader in a particular market has a propensity to follow the last market change in the interlinked markets. Each noise trader makes buy, hold or sell decisions randomly, without factoring in any market information

The author also explores a more general mechanism to measure the interdependence between two markets and choose the Clayton copula function and tail-dependence coefficient. The author investigates the relation between the Claton copula's tail-dependence coefficient and Kendal's tau coefficient. Then the author use a GARCH model to map the index return time-series into a distribution allowing the calculation of tau and thus of tail-dependence. Thus the author successfully captures dependence between markets and its changes from stable towards crisis periods.

Next the author evaluates the artificial evolutionary international market to study the characteristics of financial contagion empirically. The model is estimated on real data for Thailand, where the Asian crisis of 1997 originated, and for South Korea, one of the most affected countries where the crisis transferred to. The objective is to simulate the movements of the South Korean market, in relation to the Thailand's market. Before evaluating and running the simulation, the author begins by examining the available parameters. The dependence coefficient from the Clayton copula is further included in the formulation of the optimization criteria, i.e. the fitness function of the evolutionary optimisation technique. The author also applies the overall approach to the Russian crisis of 1998 and to modelling the contagion between the Russian and Ukrainian markets.

The author developed a new hybrid optimization technique, namely immune particle swarm optimization (Immune-PSO), which maintains the good characteristics of both Immune clonal optimization and Particle Swarm Optimization while overcoming their drawbacks, and is capable of approaching the complexity of the model,. The author first benchmarks the Immune-PSO and then successfully applies it to optimise our model. The simulations reveal, however, that the results could be improved from that point on by modifying the model itself rather than by improving further the optimisation algorithm. A changing behaviour of traders during the crisis period is introduced, in order to reflect real market observations. Thus the overall mechanism of the co-evolutionary international market enables us to gain an insight to the phenomenon of financial contagion.

7.2 Contribution

7.2.1 A GARCH-tau Approach to Clayton Tail Dependence

The author briefly discusses the correlation coefficient as a measure of dependence between two random variables, and the limitations of this measure. Then, the author discusses different copula types and how their parameters are estimated. Finally, the author developed an approach to calculating the tail-dependence coefficient, as related to the Clayton copula function. The approach is based on estimating a GARCH model and then using it to map the time-series of the stock indices into distributions allowing the calculation of Kendal's tau coefficient. Tau is then uniquely related to the left tail-dependence coefficient of the Clayton copula function. Thus the author is able to better measure the interdependence between two markets. Tail-interdependence is also used in the formulation of the fitness function for our Immune-PSO optimization algorithm.

7.2.2Immune Particle Swarm Optimization Algorithm

The author developed a new optimization technique, an Immune-PSO algorithm, which maintains the good characteristics of immune clonal optimization and particle swarm optimization. The author benchmarks the Immune-PSO algorithm against genetic algorithms, and then applies the Immune-PSO to estimate the artificial international market parameters based on empirical data for real markets. The Asian financial crisis of 1997 and the Russian crisis of 1998are selected as case studies. The artificial international market is optimised and simulated, based on the data for Thailand as origin of a crisis and South Korea as an affected market. The results of the experiments indicate that the Immune-PSO is capable of optimising the model. They also indicate that the model can be further improved.

7.2.3A Co-Evolutionary Artificial Financial Market Modelling Financial Contagion

The author have already implemented in the model, a three-option mixed-game as an extension of the two option mixed-game for the technical-Game traders, as well technical-GP traders as a new type of traders evolving decision trees based on market performance. Next the author give a tentative explanation of factors contributing to the phenomenon of financial contagion allowing for changing traders' behaviour from technical to herd traders. Thus the price is pushed even lower, leading to all traders being worse off. As the herd traders follow the downward trend in the market originating the crisis, the weighted sum of the price change in both that market and the trader's domestic market conditions their trading in the domestic market. Gradually, the downward trend in the price series will spread to the domestic market and it will become affected by the crisis in the origin market, leading to a significant increase in the dependence between the two markets. This behaviour meets the definition given in Forbes (2002), and contributes to the circle that causes financial contagion. The author now optimises the model for two case studies, the Asian crisis of 1997 and then the Russian crisis. In the latter, Russia is the origin market in the contagion process and Ukraine is the affected market.

7.3 Risk Management Implications

The financial crisis is a long-standing theme in the economic literature. The development of the world economy has been accompanied by financial crises, and they leave economists with unresolved questions.

Financial contagion, as an extension of a financial crisis, has attracted more attention, ever since the global economic crisis, including the Asian financial crisis and its impact on many emerging markets (Claessens, 2000). Claessens and Forbes (2004) recommend improving national policies, market/investor strategies and the global financial regulatory framework to deal with possible future financial contagion. One motive for this thesis on contagion is to better understand how to reduce the impact of herding traders, particularly in a time of crisis.

Despite there being a number of sophisticated methods for predicting future crises that have been developed in the recent literature, they focus on predicting the outbreak of a crisis rather than contagion, and a non-contagious crisis can be contained without the need for international intervention to mitigate risk. Secondly, whilst those models are based on factors which have been observed historically, more and more crises are now caused by factors which have not been significant in the past (Claessens and Forbes, 2004). Our co-evolutionary agent based model can be helpful, as part of an early warning system for financial contagion, in detecting contagion at an early stage, which will make crisis management more effective.

In this thesis, the author develops a model capable of simulating financial contagion, and identifies its characteristics and parameters. The analysis of the results of our model indicates that herding behaviour is a contributing factor to crisis transitions. Investors should improve their investment strategies through applying newly developed risk assessment models. It is also worth mentioning, however, as pointed out by Claessens and Forbes (2004), that when a number of institutions use similar models, the risk of contagion will be increased if these institutions all initiate similar behaviour after a crisis shock.

Governments can also take effectives measure to halt a crisis. For example, the Financial Services Authority (FSA) in the UK acted to ban short-selling during the financial crisis of 2008. During the recent crisis, Pakistan has also imposed emergency stock limits to halt the slide, through narrowing the limit on losses from 5% to 1% and doubling the cap on gains to 10%, as of June 2008. These emergency policies limit the herding traders' movements during the crisis period, and have appeared to be effective.

In the model, the large price change factors as well as a sensitivity factor controlling to what extent herd traders are allowed to follow large price changes in foreign markets. One of the conclusions is that the focus should be on improving domestic policy first. Countries with effective macro-economic management, such as appropriate debt management, controllable exchange rates and strong financial systems are less affected by the knock-on effects from neighbouring countries' trading, Claessens and Forbes (2004). A timely bailout plan by the government can also help improve the investor confidence in domestic markets, thus reducing the effects of the shock from other markets. For example, the Federal Reserve of the U.S. cut its benchmark interest rate from 5.25% to 1%, from 2006 to 2008, in an effort to halt the worst crisis of the last decade.

7.4 Outlook

In this thesis the author has developed a co-evolutionary agent-based model with the purpose of simulating contagion occurring during financial crises. Although the author have investigated and simulated financial contagion, the author considers further directions of research. The technical-GP traders evolve their strategies, the technical-Game traders work with fixed decision tables -modelling an evolution of their strategies will further improve the realism of our artificial financial markets. Modelling agents with different levels of learning ability and memory window will also be more realistic. Introducing further game theory formally, with cooperative and non-cooperative games as different stages in crisis propagation can be another development. One more direction can be improving the optimization criterion used to measure the interdependence between two markets, rather than using a single-parameter Clayton-copula. Finally, as our model is now focused on financial contagion, future work could go further and focus on the recovery mechanism that is to transform the contagion back into interdependence.

The author believes that this work has contributed value to the analysis of the financial contagion phenomenon using agent-based simulation techniques. The investigation of market characteristics leading to financial contagion will contribute in future research to enabling regulators and other market analysts to recognise at an earlier stage an emerging financial crisis of the type that is only manageable through co-ordinated international effort. This information will be valuable for managers and governments to launch effective risk management strategies.

List of Publications

[P1] Liu F, Serguieva A, Date P., (2010) “A Mixed-Game and Co-Evolutionary Genetic Programming Agent-Based Model of Financial Contagion”, Proceeding Of 2010 IEEE World Congress of Evolutionary Computation, Barcelona. vol 3, pp: 219-223

[P2] Serguieva A, Liu F, Date P., (2011) “Financial Contagion Simulation through Modelling Behavioural Characteristics of Market Participants and Capturing Cross-Market Linkages”, Proceeding Of IEEE Symposium Series on Computational Intelligence - SSCI 2011, Paris. vol 4, pp: 168-172

[P3] Liu F, Peng B., (2010) “An Immune-Particle Swarm Optimization Beats Genetic Algorithms”, Proceeding Of 2010 IEEE Global Congress of Information Systems, Wu Han, China. vol 5, pp: 319-322

Appendix

A. Pseudo code for PSO

A) For each particle
  Initialize particle
 End
B) Do
 a) For each particle
   Calculate fitness value
   If the fitness value is better than the
    best fitness value p_{Best} in history
   Set the current value as the new p_{Best}
  End
 b) For each particle
   $v[]=v[]+c_1*rand()*(p_{best}[]-present[])+$
     $+c_2*rand()*(g_{best}[]-p_{resent}[])$
   present[]=present[]+v[]
  End
While the maximum iterations or minimum
    error criteria is not attained

B. Pseudo code for an example decision tree

If (($MA\_L_{MA} = a$) AND (NOT (TRB_L_{TRB}<b))) Then
 Buy
Else
 If ($VOL\_L_{VOL} = c$) Then
  Sell
 Else
  Hold
 End if
End if

C. Pseudo code for the agent based model.

// Initialize technical-game trader
For i=1:num_technical_game
[Decision_ table_game, score_table_game] =Initialize technical-game trader;
End

```
////////////////////////////////////////////////////////////////////////////////// Initialize technical-GP trader
For i=1:n_technical
[decision_tree_GP,score_table_GP] =Initialize technical GP trader;
End
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////


For t=start: 222 /////////////////////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////////////////////////Noise trader's decision making
/////////////////////////////////////////////////// prob_sell_noise, prob_hold_noise, prob_sell_noise are predefined
For i=1:num_noise
   Probability=rand ()
   If   0<probability <prob_buy_noise
         Noise_deci(i)=1;
     Else if prob_buy_noise<probability<prob_sell_noise+prob_hold_noise
        Noise_deci(i)=0;
     Else
         Noise_deci(i)=-1;
   End
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////



/////////////////////////////////////////////////////////////////////////////////////////Herd trader's decision making

For i=1:num_herd
   Delta _A (t) =price_A (t)-price_A (t-1);
    Delta _B (t) =price_B (t)-price_B (t-1);
   Zeta (t) =sensitivity_factor_Herd*(Delta _A (t) +Delta _B (t));
   Prob_hold_herd=1/ (1+d*zeta (t-1));
   Prob_buy_herd=(1-prob_hold)*exp(zeta(t-1))/( exp(zeta(t-1))+ exp(-zeta(t-1)));
   Prob_sell_herd=1-prob_hold-prob_buy;
   Probability=rand ()
   If   0<probability <prob_buy_herd
         Herd_deci(i)=1;
     Else if prob_buy_herd<probability<prob_sell_herd+prob_hold_herd
     Herd_deci(i)=0;
     Else
         Herd_deci(i)=-1;
   End
End
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////


/////////////////////////////////////////////////////////////////////////////Technical Game traders decision making
For i=1:num_tech_Game
   For j=start: t-1
   game_Deci_ A(j)=decode_game(decision_table_A(i));
   game_Deci__B(j)=decode_game(decision_table_B(i));
   End
   Num_right_deci_A=count (find (deci_Game_A(i)==win));  // number of right decision using  A
   Num_right_deci_B=count (find (deci_Game_B(i)==win));  // number of right decision using  B
```

```
   score_game_A=update_score_table(num_right_deci);
   score_game_B=update_score_table(num_right_deci);
   Best_ strategy_A =selection_Game(score _game_A);
   Best_ strategy_B =selection_Game(score_game_B));
   (Market,Best_strategy)=market_selection_Game(score_market_A, score_market_B); ///select market
   Game_Deci (i) =decode (Best_ strategy) /////best strategy is of the selected market
    End
///////////////////////////////////////////////////////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////////////////////technical GP traders decision making
For i=1:num_tech_GP
    If wealth_GP (i) <average_wealth //////red queen principal, re-initialize half of the population when////////////////////////////////////////////////////////////wealth fall below average
        Population_GP (1:population_size_GP/2) =initiate_population_GP (population_size_GP/2);
    End
  Population_GP=crossover (population_GP);
   Population_GP=mutation (population_GP);
   For j=1: t-1
       GP_deci(j)=decode_GP(population_GP);
   End
   Num_right_deci=count (find (GP_ deci (i)==win));   //calculate the number of right decision/////////////////////////////////////////////////     /for each decision tree  of player i up to time t;
   Fitness=num_right_deci/ (t-start);
   Best_tree=selection_GP (fitness);
   Market=market_selection_GP(score_GP_market_A, score_GP_market_B); ///select market
   GP_deci(i)=decode_GP(Best_tree, market_info(market)); ////////////decision of GP trader i at time t
   End
   ////////////////////////////////////////////////////////////////////


/////////////////////////////////////////////////////////////////////////////////////////////////////// price formation
Total_decision= [noise_deci,herd_deci,game_deci,GP_deci];

     [num_Buy (t), num_Sell (t), num_Hold (t)]=classification (noise_deci,herd_deci,game_deci,GP_deci);
     D (t) =sum (decision);
     Price (t) =Price (t-1) +D (t)/lamna;
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

//////////////////////////////////////////////////////////////////////////////////////////////////////// Update wealth
Num_Buy_new(t)=num_Buy(t)*Price(t-1)/Price(t);
  tou_plus=min(1,O(t)/num_Buy_new(t));
  tou_minus=min(1,num_Buy_new(t)/O(t));


for i=1:total_players
      if  total_decision(i)==1
        rou(i,t)=g*tou_plus*cash(i,t)/Price(t);
else if
       total_decision(i)==-1
        rou(i,t)=-g*tou_minus*num_shareholding(i,t);
else
        rou(i,t)=0;
end
```

```
num_shareholding(i,t)= num_shareholding(i,t-1)+rou(i,t);  /////////// new number of  share holding
cash(i,t)= cash(i,t-1)+ rou(i,t)*P(t);  //////////////////////////////////////////////////new cash holding
end

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////// Update technical game traders
Score _game_A =Update_game_score(score _game_A);
Score _game_B =Update_game_score(score _game_B);
Score _market_game_ A =Update_game_market_score(score_market _game_A);
Score _market_game_ B =Update_game_market_score(score_market _game_B);

////////////////////////////////////////////////////////////////////////////////////////////////////Update technical GP traders
Score _market_GP_ A =Update_GP_market_score(score_market _GP_A);
Score _market_GP_ B =Update_GP_market_score(score_market _GP_B);
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
End
```

D. Pseudo code for the Immune-PSO Algorithm

```
%------experiment setup---------------------------------------------
c1=1.4962;          %learning factor 1
c2=1.4962;          %learning factor 2
w=0.7298;           %inertia coefficient
MaxDT=200;          %maximum iteration time
D=14;               %dimension
N=10;               %size of population
eps=10^(-20);       %stopping criteria
replaceP=0.6;       %replacement probability

%------initialize individuals ------------
fori=1:N
position (i)=Initialize_ position()   %%%% position(i) is a set of parameters to be optimized
velocity(i)=Initialize_velocity()
End

%------calculate the fitness of each particle and initialize local_best(i)andpg (global best) - --

Fori=1:N
   fitness (i)=Fitness_function(position(i))%%%%%% see the pseudo code for artificial
financial market
   local_best(i)=position(i);
```

```
End

pg=position (1);            %set an initial value for pg

Fori=2: N% find global best (pg) before iteration
   If Fitness_function(position(i))<Fitness_function(pg)
   pg=position (i);
End
End

%------iterating till the stopping criteria met------------------------------------------------------------
-----
For t=1:MaxDT
Fori=1:N
Velocity (i) =update_velocity(position(i),local_best(i),pg)
position (i)=position(i)+velocity (i);

       If Fitness_function(position(i))<fitness(i)
fitness(i)=fitness_function(position(i))
local_best(i)=position(i);
End

  If position (i)<Fitness_function(pg)
pg=local_best(i);
End
Pbest(t)=Fitness_function(pg);

  %-----------proceed immune process—————————————————
affinity=calc_affinicty(position,pg);
position=clone_copy(affinity,N);
position=hyper_mutaion(position);
pg=selection (position);
end
```

Stochastic Universal Selection 1

Gaussian Mutation

Stall Generation Limit:30

Population Size:1000

Singlepoint Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.4094	13.5018	0.0115
Foxhole	0.998	0.998	10.2794	0
GoldsteinPrice	3	3.991	7.0003	0.991
Rosenbrock	0	0.1247	6.0908	0.1247
Schwefel	0	0.0637	7.1027	0.0637
SixHump	-1.0316	-1.0284	7.2782	0.0032
			51.2532	**1.1941**

Stochastic Universal Selection 2

Gaussian Mutation

Stall Generation Limit:30

Population Size:1000

Two point Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.4206	4.5658	0.0227
Foxhole	0.998	0.9989	4.3474	0.0009
GoldsteinPrice	3	24.3153	3.3358	21.3153
Rosenbrock	0	0.0362	3.7151	0.0362
Schwefel	0	0.274	4.2414	0.274
SixHump	-1.0316	-1.0085	3.2197	0.0231
			23.4252	**21.6722**

Stochastic Universal Selection 3

Gaussian Mutation

Stall Generation Limit:30

Population Size:1000

Intermediate Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.3979	6.869	0
Foxhole	0.998	0.998	7.7103	0
GoldsteinPrice	3	3	6.1596	0
Rosenbrock	0	0	6.052	0
Schwefel	0	0	5.9968	0
SixHump	-1.0316	-1.0316	6.175	0
			38.9627	**0**

Stochastic Universal Selection 4

Uniform Mutation

Stall Generation Limit:30

Population Size:1000

Single point Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.4251	6.1832	0.0272
Foxhole	0.998	0.998	7.7954	0
GoldsteinPrice	3	31.2417	6.1035	28.2417
Rosenbrock	0	0.2564	5.8863	0.2564
Schwefel	0	0.0843	5.9127	0.0843
SixHump	-1.0316	-0.9446	6.0055	0.087
			37.8866	**28.6966**

Stochastic Universal Selection 5

Uniform Mutation

Stall Generation Limit:30

Population Size:1000

Two point Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	1.1818	5.7984	0.7839
Foxhole	0.998	0.9981	7.879	1E-04
GoldsteinPrice	3	159.8011	6.2036	156.8011
Rosenbrock	0	0.5681	6.0293	0.5681
Schwefel	0	0.0345	6.3845	0.0345
SixHump	-1.0316	-1.0002	7.1258	0.0314

			39.4206	**158.2191**

Stochastic Universal Selection 6

Uniform Mutation

Stall Generation Limit:30

Population Size:1000

Intermediate Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.3979	5.9438	0
Foxhole	0.998	0.998	9.2044	0
GoldsteinPrice	3	3	6.2838	0
Rosenbrock	0	0	6.0184	0
Schwefel	0	0	6.174	0
SixHump	-1.0316	-1.0316	6.1273	0
			39.7517	**0**

Stochastic Universal Selection 7

Adaptfeasible Mutation

Stall Generation Limit:30

Population Size:1000

Singlepoint Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.3979	7.6899	0
Foxhole	0.998	0.998	8.355	0
GoldsteinPrice	3	3	6.7631	0
Rosenbrock	0	0	6.6196	0
Schwefel	0	0.0002	7.7884	0.0002
SixHump	-1.0316	-1.0316	6.7483	0
			43.9643	**0.0002**

Stochastic Universal Selection 8

Adaptfeasible Mutation

Stall Generation Limit:30

Population Size:1000

Two point Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.3979	7.9577	0
Foxhole	0.998	0.998	8.6542	0
GoldsteinPrice	3	3.0001	8.1191	0.0001
Rosenbrock	0	0	7.4651	0
Schwefel	0	0	7.4425	0
SixHump	-1.0316	-1.0316	7.3704	0
			47.009	**0.0001**

Stochastic Universal Selection 9

Adaptfeasible Mutation

Stall Generation Limit:30

Population Size:1000

Intermediate Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.3979	7.8308	0
Foxhole	0.998	0.998	8.6518	0
GoldsteinPrice	3	3	6.9837	0
Rosenbrock	0	0	7.3039	0
Schwefel	0	0	7.0849	0
SixHump	-1.0316	-1.0316	6.9936	0
			44.8487	**0**

Roulette wheel selection 10

Gaussian Mutation

Stall Generation Limit:30

Population Size:1000

Singlepoint Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.4074	5.7874	0.0095
Foxhole	0.998	0.998	7.7744	0
GoldsteinPrice	3	3.1854	8.209	0.1854
Rosenbrock	0	0.8147	8.5425	0.8147
Schwefel	0	0.0717	7.2959	0.0717
SixHump	-1.0316	-1.0315	7.3345	1E-04

			44.9437	**1.0814**

Roulette wheel selection 11

Gaussian Mutation

Stall Generation Limit:30

Population Size:1000

Two point Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.4179	6.9516	0.02
Foxhole	0.998	0.998	8.4479	0
GoldsteinPrice	3	32.0737	7.644	29.0737
Rosenbrock	0	0.1582	6.7657	0.1582
Schwefel	0	0.0798	6.3911	0.0798
SixHump	-1.0316	-1.0155	6.5289	0.0161
			42.7292	**29.3478**

Roulette wheel selection 12

Gaussian Mutation

Stall Generation Limit:30

Population Size:1000

Intermediate Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.3979	7.8637	0
Foxhole	0.998	0.998	7.7567	0
GoldsteinPrice	3	3.0001	6.8385	0.0001
Rosenbrock	0	0	6.3034	0
Schwefel	0	0	6.3741	0
SixHump	-1.0316	-1.0316	6.4623	0
			41.5987	**0.0001**

Roulette wheel selection 13

Uniform Mutation

Stall Generation Limit:30

Population Size:1000

Singlepoint Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.4003	15.8717	0.0024
Foxhole	0.998	0.998	8.8605	0
GoldsteinPrice	3	35.916	6.55	32.916
Rosenbrock	0	1.4017	6.6019	1.4017
Schwefel	0	0.3211	6.2275	0.3211
SixHump	-1.0316	-0.6054	6.326	0.4262
			50.4376	**35.0674**

Roulette wheel selection 14

Uniform Mutation

Stall Generation Limit:30

Population Size:1000

Two point Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.5749	7.763	0.177
Foxhole	0.998	0.999	7.8761	0.001
GoldsteinPrice	3	27.3964	6.1809	24.3964
Rosenbrock	0	0.3451	6.1584	0.3451
Schwefel	0	0.3202	6.061	0.3202
SixHump	-1.0316	-1.0307	6.2889	0.0009
			40.3283	**25.2406**

Roulette wheel selection 15

Uniform Mutation

Stall Generation Limit:30

Population Size:1000

Intermediate Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.3979	6.6773	0
Foxhole	0.998	0.998	9.3039	0
GoldsteinPrice	3	3	6.9817	0
Rosenbrock	0	0.0001	6.7345	0.0001

Schwefel	0	0	6.8726	0
SixHump	-1.0316	-1.0316	6.7507	0
			43.3207	**0.0001**

Roulette wheel selection 16

Adaptfeasible Mutation

Stall Generation Limit:30

Population Size:1000

Singlepoint Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.3979	8.5209	0
Foxhole	0.998	0.998	8.9883	0
GoldsteinPrice	3	3	7.5996	0
Rosenbrock	0	0.0001	8.5714	0.0001
Schwefel	0	0.0004	7.277	0.0004
SixHump	-1.0316	-1.0316	7.0769	0
			48.0341	**0.0005**

Roulette wheel selection 17

Adaptfeasible Mutation

Stall Generation Limit:30

Population Size:1000

Two point Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.3979	7.8842	0
Foxhole	0.998	0.998	8.6778	0
GoldsteinPrice	3	3	7.2064	0
Rosenbrock	0	0	7.3806	0
Schwefel	0	0	8.6659	0
SixHump	-1.0316	-1.0315	7.7646	1E-04
			47.5795	**1E-04**

Roulette wheel selection 18

Adaptfeasible Mutation

Stall Generation Limit:30

Population Size:1000

Intermediate Crossover

Function Name	Real Value	Experiment Value	Average Running Time	Error
Branin	0.3979	0.3979	7.8381	0
Foxhole	0.998	0.998	8.747	0
GoldsteinPrice	3	3	7.4614	0
Rosenbrock	0	0	7.2425	0
Schwefel	0	0	7.2607	0
SixHump	-1.0316	-1.0316	7.3388	0
			45.8885	**0**

Branin

Group Number	Running Time	Error
3	6.869	0
6	5.9438	0
7	7.6899	0
8	7.9577	0
9	7.8308	0
12	7.8637	0
15	6.6773	0
16	8.5209	0
17	7.8842	0
18	7.8381	0
13	15.8717	0.0024
10	5.7874	0.0095
1	13.5018	0.0115
11	6.9516	0.02
2	4.5658	0.0227
4	6.1832	0.0272
14	7.763	0.177
5	5.7984	0.7839

Foxhole

Group Number	Running Time	Error
2	4.5658	0.0227
3	7.7103	0
12	7.7567	0

10	7.7744	0
4	7.7954	0
14	7.8761	0.001
5	7.879	1.00E-04
7	8.355	0
11	8.4479	0
9	8.6518	0
8	8.6542	0
17	8.6778	0
18	8.747	0
13	8.8605	0
16	8.9883	0
6	9.2044	0
15	9.3039	0
1	10.2794	0

Goldstein Price

Group Number	Running Time	Error
3	6.1596	0
6	6.2838	0
7	6.7631	0
9	6.9837	0
15	6.9817	0
16	7.5996	0
17	7.2064	0
18	7.4614	0
8	8.1191	0.0001
12	6.8385	0.0001
10	8.209	0.1854
1	7.0003	0.991
2	3.3358	21.3153
14	6.1809	24.3964
4	6.1035	28.2417
11	7.644	29.0737
13	6.55	32.916
5	6.2036	156.8011

Rosenbrock

Group Number	Running Time	Error

3	6.052	0
6	6.0184	0
7	6.6196	0
8	7.4651	0
9	7.3039	0
12	6.3034	0
17	7.3806	0
18	7.2425	0
15	6.7345	0.0001
16	8.5714	0.0001
2	3.7151	0.0362
1	6.0908	0.1247
11	6.7657	0.1582
4	5.8863	0.2564
14	6.1584	0.3451
5	6.0293	0.5681
10	8.5425	0.8147
13	6.6019	1.4017

Schwefel

Group Number	Running Time	Error
3	5.9968	0
6	6.174	0
8	7.4425	0
9	7.0849	0
12	6.3741	0
15	6.8726	0
17	8.6659	0
18	7.2607	0
7	7.7884	0.0002
16	7.277	0.0004
5	6.3845	0.0345
1	7.1027	0.0637
10	7.2959	0.0717
11	6.3911	0.0798
4	5.9127	0.0843
2	4.2414	0.274
14	6.061	0.3202
13	6.2275	0.3211

SixHump

Group Number	Running Time	Error
3	6.175	0
6	6.1273	0
7	6.7483	0
8	7.3704	0
9	6.9936	0
12	6.4623	0
15	6.7507	0
16	7.0769	0
18	7.3388	0
10	7.3345	1.00E-04
17	7.7646	1.00E-04
14	6.2889	0.0009
1	7.2782	0.0032
11	6.5289	0.0161
2	3.2197	0.0231
5	7.1258	0.0314
4	6.0055	0.087
13	6.326	0.4262

References

1. Ahn TK, Ostrom E, Walker J. (2003) “Incorporating motivational heterogeneity into game theoretic models of collective action”, Public Choice 117:295–314.

2. Aike M., Bsat M., (1994) “Forecasting market trends with neural networks,” Inf. Syst. Manage., vol. 16, pp: 42–48.

3. Allen F., Karjalainen R., (1999) “Using genetic algorithms to find technical trading Rules, Journal of financial economics, pp: 245-271.
4. Allen F., Karjalainen, R., (1995) “Using genetic algorithms to find technical trading rules,” Working paper at Rodney L. White Centre for Financial Research.

5. Andreou A, Georgopoulos E., Likothanassis S., (2002) "Exchange-rates forecasting: A hybrid algorithm based on genetically optimized adaptive neural networks." Computational Economics, vol.20, no.3, pp: 191-210.

6. Angeletos G.M., Christian H., Alessandro P., (2007a) “Dynamic Global Games of Regime Change: Learning, Multiplicity, and the Timing of Attacks”, Econometrica ,75, pp. 711-756.

7. Angeletos, G.M., Christian H .,Alessandro P., (2007b) Defense Policies against Currency Attacks: On the Possibility of Predictions in a Global Game with Multiple Equilibria, Discussion Papers 1459, Northwestern University, Center for Mathematical Studies in Economics and Management Science.

8. Araujo R.D.A., Vasconcelos G.C., Ferreira T.A.E., (2007)“Hybrid differential evolutionary system for financial time series forecasting”, IEEE Congress on Evolutionary Computation, CEC 2007, art. no. 4425036, pp. 4329-4336.

9. Arcuri A., Yao X., (2008) “A Novel Co-evolutionary Approach to Automatic Software Bug Fixing”, 2008 IEEE Congress on Evolutionary Computation.

10. Avci E., Avci, D., (2009) “Speaker identification by using genetic wavelet adaptive network based fuzzy inference system”, Expert Systems with Applications, pp: 9928–9940.

11. Bäck, T. & Hoffmeister, F.(1991): Extended Selection Mechanisms in Genetic Algorithms. In [ICGA4], pp. 92-99, 1991.

12. Ba-Karait S., Shamsuddin S.M., Sudirman R., (2009)" Swarm Negative Selection Algorithm for Electroencephalogram Signals Classification", Journal of Computer Science 5 (12), pp: 998-1005.

13. Banzhaf W., Nordin P., Keller R. E., Francone F.D., (1998)"Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications, Morgan Kaufmann.

14. Baue R. J. Jr., (1994) "Genetic Algorithms and Investment Strategies", New York, John Wiley & Sons, Inc.

15. Beni G., Wang J., (1989) "Swarm Intelligence in Cellular Robotic Systems, Proceed", NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy.

16. Beyer H.G., (2001) "The Theory of Evolution Strategies", Springer April 27, 2001.

17. Black K., (2006) "Business statistics for contemporary decision making", Wiley, ISBN: 0-471-70563-2.

18. Bollerslev T., (1986). "Generalized autoregressive conditional heteroscdesticity," .Journal of Econometrics 31, pp.307- 327.

19. Bobtcheff C., Mariotti T.,(2012) "Potential competition in pre-emption games", Games and Economic Behaviour 75 (2012) 53–66.

20. Bollerslev., T (1986). "Generalized autoregressive conditional heteroscdesticity," ， Journal of Econometrics 31, pp.307- 327.

21. Booker L.,(1987) "Improving search in genetic algorithms," In Genetic Algorithms and Simulated Annealing, L. Davis (Ed.), pp. 61-73, Morgan Kaufmann Publishers, 1987.

22. Boyer B.H., (1997) "Pitfalls in tests for changes in correlations", International Finance Discussion Paper, Board of Governors of the Federal Reserve System.

23. Bramlette M. F., (1991) "Initialization, Mutation, and Selection Methods in Genetic Algorithms for Function Optimization", Proceedings of the Fourth International Conference on Genetic Algorithms, Morgan Kaufman: Los Altos, CA.

24. Camerer C., (2003) "Behavioral game theory", Princeton: Princeton University Press.

25. Campelo F., Guimarães F.G., Igarashi H., Ramírez J.A., Noguchi S., (2006) "A Modified Immune Network Algorithm for Multimodal Electromagnetic Problems", IEEE Trans. Magnetics 42 (4): pp: 1111-1114, ISSN 0018-9464.

26. Caporale G.M, Cipollini A.N., Spagnolo N., (2005) "Testing for contagion: a conditional correlation analysis", Journal of Empirical Finance, vol. 12, no. 3, pp. 476-489.

27. Caporale G.M., Serguieva A, Wu H., (2009) "Financial contagion: evolutionary optimization of a multinational agent-based model", Intelligent System in Accounting, Finance and Management, vol. 16, no 1-2, 2009, pp. 111-126.

28. Caro G.D., (1998) "Two Ant Colony Algorithms for Best-Effort Routing in Datagram Networks", Universit´e Libre de Bruxelles.

29. Chang P.C., Wang,Y.W., Yang,W.N., (2004)"An investigation of the hybrid forecasting models for stock price variation in Taiwan," J. Chin. Inst. Ind. Eng., vol. 21, no. 4, pp: 358–368.

30. Chen S.H., Yeh, C.H., (1996) "Toward a computable approach to the efficient market hypothesis: An application of genetic programming," Journal of Economic Dynamics and Control, pp: 1043-1063.

31. Chen S.H.,Yeh C.H., (2000) "Issues in the sensitivity analysis of agent-based modelling of artificial stock markets. AI-ECON Research Group Working Paper 2000-1, Department of Economics, National Chengchi University, Taipei, Taiwan.
32. Chen S.H., Yeh C.H., (2001)"Evolving traders and the business school with genetic programming: a new architecture of agent-based artificial stock market", Journal of economic dynamics and control， vol 25, pp: 363-393.
33. Chi S.C.H., Chen P., Cheng C.H., (1999) "A forecasting approach for stock index future using grey theory and neural networks," IEEE Washington vol. 6, pp. 3850–3855.

34. Claessens S., (2000) "Contagion: How it spreads and how it can be stopped?" , Tech.rep, World Bank Research Observer.

35. Claessens S., Forbes K., (2004) 'World Economy Forum/Reinventing Bretton Woods Committee Conference on "The IMF's Role In Emerging Market Economies" '.

36. Claman H. N,, Chaperon E. A., Triplett R. F., (1966) "Immunocompetence of transferred thymus-marrow cell combinations", J. Immunol. 97, pp: 928-832

37. Colorni A., Dorigo M., Maniezzo V., (1991) “Distributed Optimization by Ant Colonies”, actes de la première conférence européenne sur la vie artificielle, Paris, France, Elsevier Publishing, pp: 134-142.

38. Cont R., Bouchaud J., (2000) “Herd behaviour and aggregate fluctuations in financial markets,” Macroeconomic Dynamics, vol. 4, no. 2, pp. 170-196.

39. Coutinho A., (1995). "The network theory: 21 years later". Scand. J. Immunol. 42 (1): 3–8,PMID 7631141.

40. Crama Y., Schyns M., (2003)” Simulated annealing for complex portfolio selection problems”, European Journal of Operational Research 150 (2003) pp: 546–571.

41. Cramer H., Nichael L.,(1985)"A representation for the Adaptive Generation of Simple Sequential Programs", in Proceedings of an International Conference on Genetic Algorithms and the Applications, Grefenstette, John J. (ed.), Carnegie Mellon University.

42. Crutchley D.A., Zwolinski M., (2003)” Globally Convergent Algorithms for DC Operating Point Analysis of Nonlinear Circuits”, IEEE transaction on evolutionary computation, vol. 7, no. 1, pp: 1144-1150.

43. Dasgupta D., KrishnaKumar K., Berry D.W.M (2004) “Negative Selection Algorithm for Aircraft Fault Detection”, Artificial Immune Systems, pp: 1-13

44. Dasgupta A., (2004) Financial Contagion through Capital Connections: A Model of the Origin and Spread of Bank Panics, Journal of the European Economic Association 2, pp. 1049-1084.

45. Davis L., (1991) “Handbook of Genetic Algorithms”, Van Nostrand Reinhold.

46. Davis L., (1989) “Adapting Operator Probabilities in Genetic Algorithms”, Proc. ICGA 3, pp. 61-69, 1989.

47. De Long., Bradford J., Shleifer A, Summers L., Waldmann R.J., (1990) “Noise trader risk in financial markets,” Journal of Political Economy, vol. 98, no. 4, pp: 703-738.

48. deCastro L.N., VonZuben F.j., (2000) “An Evolutionary Immune Network for Data Clustering”, In Proc. of the IEEE SBRN (Brazilian Symposium on Artificial Neural Networks), pp:84-89.

49. deCastro L.N., VonZuben F.J.,(2000)” The Clonal Selection Algorithm with Engineering Applications”, Workshop Proceedings of GECCO’00, pp. 36-37,

Workshop on Artificial Immune Systems and Their Applications, Las Vegas, USA, July 2000.

50. deCastro L.N., VonZuben F.J.,(2002) “Learning and Optimization Using the Clonal Selection Principle”,, IEEE transactions on evolutionary computation, VOL. 6, NO. 3, JUNE 2002.

51. deCastro L.N., Timmis J.,(2002)“Artificial Immune Systems: A New Computational Intelligence Approach”. Springer. pp. 57–58. ISBN 1852335947, 9781852335946.

52. Dooley M.P (1997). "Are Recent Capital Inflows to Developing Countries a Vote for or against Economic Policy Reforms?" In Exchange Rates, Capital Flows, and Capital Mobility in A Changing World Economy, William Gruben, et al., eds. Boston: Kluwer Academic Publishers.

53. Dorigo M., (1992) “Optimization, Learning and Natural Algorithms”, PhD thesis, Politecnico di Milano, Italie.

54. Doyle S., Corcoran D.,Connel J.,(1999) “Automated mirror design using an evolution strategy ”, Opt. Eng., Vol. 38, 323 , doi:10.1117/1.602091.

55. Drezner Z., Marcoulides G.A., StohsM.H., (2001) “Financial Applications of a Tabu Search Variable Selection Model ”,Journal of Applied Mathematics and Decision Sciences, Vol 5, , pp:215 – 234.

56. Duncan G.R., (2002) "Radio emerges from the electronic soup." New Scientist, vol.175, no.2358, pp: 19 (August 31, 2002).

57. Eberhart R. C., Kennedy J., (1995) “A new optimizer using particle swarm theory”, Proceedings of the Sixth International Symposium on Micro-machine and Human Science, Nagoya, Japan, pp: 39-43

58. Evans R., Grant C. K., Cox H., Steel K. Alexander P., (1972) “Thymus-derived lymphocytes produce an immunologically specific macrophage-arming factor”, J. Exp. Med., 136, pp:1318-1322

59. Falkenauer E, Bouffouix S. (1991) “A genetic algorithm for job shop”. In: Proceedings of the IEEE international conference on robotics and automation, 1991. p. 824–9.

60. Fan K., Brabazon A., Sullivan C., (2008) “Quantum-Inspired Evolutionary Algorithms for Financial Data Analysis” Natural Computing Research and Applications Group University College Dublin, Ireland

61. Fang Y.S., Li J., (2010) A Review of Tournament Selection in Genetic Programming, Advances in Computation and Intelligence, Volume 6382/2010, 181-192, DOI: 10.1007/978-3-642-16493-4_19.

62. Farmer J.D, (2006) “Market force, ecology, and evolution,” Industrial and Corporate Change, vol. 11, pp. 895-953.

63. Farmer J.D., Lo A.W., (1999) “Frontiers of finance: Evolution and efficient markets”, Proc. Natl. Acad. Sci. USA Vol. 96, pp: 9991–9992.

64. Fogarty T. C., (1989) “Varying the Probability of Mutation in the Genetic Algorithm”, Proc. ICGA 3, pp. 104-109, 1989.

65. Fogel L.J., (1999) “Intelligence through Simulated Evolution: Forty Years of Evolutionary Programming”, John Wiley.

66. Fogel L.J., Owens, A.J., Walsh, M.J. (1966) “Artificial Intelligence through Simulated Evolution”, John Wiley.

67. Fogel LJ. Fogel D.B., Wirt Atmal W., (1993) “Evolutionary Programming for ASAT Battle Managment”, Statistics and Computing, pp: 113-16.

68. Forbes K., Rigobon R.,(2002) “No contagion, only interdependence: measuring stock market co-movements,” Journal of Finance, vol. 57, no. 5, pp. 2223-2261.

69. Forrest G., (1993)"Genetic algorithms: principles of natural selection applied to computation." Science, vol.261, pp: 872-878 (1993).

70. Forrest S., (1993) "Genetic Algorithms: Principles of Natural Selection Applied to Computation," Science, 261, pp: 872-878.

71. Forrest S., Perelson A.S., Allen L., Cherukuri R., (1994) "Self-nonself discrimination in acomputer" Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA, pp: 202–212.

72. Frankela D.M, Morrisb.S ,Pauznerc.A., (2008) Equilibrium selection in global games with strategic complementarities, Journal of Economic Theory 108 (2003) 1–44.

73. Garcia-Almanza A.L.，Tsang E., (2007)”repository method to suit different Investment strategies”，IEEE Transaction on Evolutionary Computation vol: 43 no.1, pp: 1205-1212.

74. García-Arnau M., Manrique D., , Ríos J., Rodríguez-Patón A. (2007) Initialization method for grammar-guided genetic programming, The 26th

SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Volume 20, Issue 2, March 2007, Pages 127–133.

75. Geng H.T., Song Q.X., Jiao F., Sun Y.J., (2009) “An evolution strategy with stochastic ranking for solving reactive power optimization”, PEITS 2009 - 2009 2nd Conference on Power Electronics and Intelligent Transportation System, 1, art. No, 5407050, pp. 14-17.

76. Giardina I, J. Bouchaud J.P (2003) “Bubbles, crashes and intermittency in agent based market models,” The European Physical Journal B -Condensed Matter, vol. 31, pp. 421–437.

77. Glover F., (1986) “Future Paths for Integer Programming and Links to Artificial Intelligence,” Computer and Operations Research, vol. 13, no. 5, pp. 533-549.

78. Glover F., (1989a) “Tabu Search – Part I,” INFORMS Journal on Computing, vol. 1, no. 3, pp. 190-206.

79. Glover F., (1989b) “Tabu Search – Part II,” INFORMS Journal on Computing, vol. 2, no. 1, pp. 4-32.

80. Goldberg D.E., (1989) Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley: Reading, MA.

81. Gou C., (2006) “The simulation of financial markets by an agent-based mixgame model”, Journal of Artificial Societies and Social Simulation, vol.9, no. 3, pp. 321-328. Aboudi R., Jörnsten K., (1994) “Tabu Search for General Zero-One Integer Programs Using the Pivot and Complement Heuristic”, Orsa Journal On Computing, Vol. 6, No. 1, winter 1994, pp: 82-93.

82. Greensmith J., Aickelin U., (2009) "Artificial Dendritic Cells: Multi-faceted Perspectives" Human-Centric Information Processing Through Granular Modelling: pp: 375–395.

83. Hangan T., (2000) “Neural Networks for Control”, School of Electrical & Computer Engineering Oklahoma State University, Demuth Electrical Engineering Department University of Idaho.

84. Harrald P.G., Kamstra M., (1997) “Evolving artificial neural networks to combine financial forecasts,” IEEE Trans. Evol. Comput, vol. 1, no. 1, pp: 40–52.

85. Harsanyi J. C., (1974), "An equilibrium point interpretation of stable sets", Management Science 20 (11): 1472–1495, DOI:10.1287/mnsc.20.11.1472

86. Hens T., Schenk–Hopp´e K.R., Stalder M., (2002) “An Application of Evolutionary Finance to Firms Listed in the Swiss Market Index” Institute for Empirical Research in Economics University of Zurich.

87. Hinchliffe M., Willis M., Tham M.,(1999) “Dynamic chemical process modelling using a multiple basis function genetic programming algorithm”, In: Banzhaf W, Daida J, Eiben AE, Garzon HE, Honavar V, Jakiela M, Smith RE, editors, Proceedings of the genetic and evolutionary computation conference (v2). Orlando: Morgan Kaufmann; 1999. pp: 13-17.

88. Hoffmann G. W., (1975) “A network theory of the immune system”, Eur. J. Immunol., 5, pp: 638-647.

89. Holland J. H., (1975) “Adaptation in Natural and Artificial Systems”, Univ. of Michigan Press: Ann Arbor, Reprinted in 1992 by MIT Press, Cambridge MA.

90. Hong W.C., (2007) “Evolutionary algorithms in a support vector regression financial forecasting model: case of continuous ant colony optimization algorithms”, Proceeding on the 2nd International Symposium on Intelligence Computation and Applications.

91. Igawa K., Ohashi H., (2009) “A negative selection algorithm for classification and reduction of the noise effect”, Applied Soft Computing, Volume 9, pp: 431-438.

92. Ihme M., Marsden A.L., Pitch h., (2006) “On the optimization of artificial neural networks for application to the approximation of systems”, Centre for Turbulence Research Annual Research Briefs, pp: 105-109.

93. Isasi P., Quintana D., Saez Y., (2007)”Applied finance and economics”, Computational Intelligence Volume 23 pp; 1123-1129.

94. Itay G , Pauzner A., (2004) Contagion of Self-Fulfilling Financial Crises Due to Diversification of Investment Portfolios, Journal of Economic Theory 119, pp. 51-183.

95. Jassara S., Liao Z., Zhao L., (2009) “Adaptive neuro-fuzzy based inferential sensor model for estimating the average air temperature in space heating systems”, Building and Environment , pp:1609–1616.

96. Jefferies P., Hart M., Hui P., Johnson N., (2001) “From market games to real-world markets,” The European Physical Journal B - Condensed Matter and Complex Systems, vol. 20, pp. 493-501.

97. Jeon J.Y., Kim J.H., Koh K.,(1996) “Experimental Evolutionary Programming-Based High-Precision Control”, Proceedings of the 1996 IEEE International Conference on Robotics and Automation, pp:2258-2263.

98. Jerne N. K., (1974) “Towards a network theory of the immune system”, Ann. Immunol. (Inst. Pasteur), 125C, pp: 373-389.

99. Jha G.K., Kumar S., Prasain H., Thulasiraman P., Thulasiraman R.K.,(2009)”Option pricing using particle swarm optimization”, ACM International Conference Proceeding Series, pp: 267-272.

100. Kaminsky G., Lizondo S., Reinhart C., (1998), “Leading indicators of currency crisis,” IMF Staff Papers, International Monetary Fund, vol. 45, no. 1.

101. Kaminsky G., (1999) “Currency and banking crises - the early warning of distress,” IMF Staff Papers, International Monetary Fund, no. 99/178.

102. Kannana S., Slochanal S.M.R., Subbaraj P., Narayana Prasad Padhy N.P.,(2003) “Application of particle swarm optimization technique and its variants to generation expansion planning problem”, Electric Power Systems Research 70 (2004) ,pp: 203–210.

103. Kapadi M.D., Gudi R.D., (2003) “Optimal control of fed-batch fermentation involving multiple feeds using Differential Evolution”, Process Biochemistry 39 (2004) pp: 1709–1721.

104. Karr. C. L., (1991) “Design of an Adaptive Fuzzy Logic Controller Using a Genetic Algorithm”, Proc. ICGA 4, pp:450-457, 1991.

105. Sourirajan K., Ozsen L., Uzsoy R., (2009) “A genetic algorithm for a single product network design model with lead time and safety stock considerations” European Journal of Operational Research, pp: 599–608.

106. Ken B., (2007), playing for real: a text on game theory. Oxford University Press US. ISBN 978-0-19-530057-4., chapters 1 & 7

107. Kendall G., Su Y., (2005) “A particle swarm optimisation approach in the construction of optimal risky portfolios”, IASTED International Conference on Artificial Intelligence and Applications, part of the 23rd Multi-Conference on Applied Informatics, Innsbruck, Austria, pp:234-239.

108. Kennedy J., Eberhart R.C., (1995) “Particle swarm optimization”, Proceedings of IEEE International Conference on Neural Networks, Piscataway, pp: 1942-1948.

109. Kephart J.O., (1994) "A biologically inspired immune system for computers", Proceedings of Artificial Life IV: The Fourth International Workshop on the Synthesis and Simulation of Living Systems. MIT Press, pp: 130–139.

110. Kimoto T., Asakawa, K., (1990) “Stock market prediction system with modular neural network,” IEEE Int, vol. 1, pp: 1–6.

111. Kishore, J.K and Patnaik, L.M and Mani, V and Agrawal, V.K (2000) “Application of Genetic Programming for Multi-category Pattern Classification”, In: IEEE Transactions on Evolutionary Computation, 4 (3), pp: 242-258.

112. Ko J.H., Kim K.T., Kim K.A., Kim H.S., (2010)“A study on an optimal design of a triple-band PIFA using the evolution strategy”, Transactions of the Korean Institute of Electrical Engineers, 59 (4), pp: 768-773.

113. Kolahan F., Abolbashari M.H., Mohitzadeh S., (2007)” Simulated Annealing Application for Structural Optimization”, World Academy of Science, Engineering and Technology 35.

114. Korczak J.J., Lipinski P., Roger P., (2002) “Evolution Strategy in Portfolio Optimization”, Institute of Computer Science, University of Wrocław, Wrocław, Poland.

115. Koza J. R., (1992) “Genetic Programming”, MIT Press: Cambridge, MA.

116. Koza J.R, Keane M.A, Streeter M.J. (2004) “Routine automated synthesis of five patented analog circuits using genetic programming”, Soft computing -a fusion of foundations. Methodol Appl 8, pp: 318-324.

117. Koza J.R., (1990) “Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems”, Stanford University Computer Science.

118. Koza J.T.,(1997) Genetic Programming: An Introduction ,Morgan Kaufmann, ISBN: 9781558605107

119. Krink T., Paterlini S., (2008) "Differential Evolution for Multi-objective Portfolio Optimization", Centro Studi di Banca e Finanza (CEFIN) (Center for Studies in Banking and Finance) 08012, Universita di Modena e Reggio Emilia, Facoltà di Economia "Marco Biagi".

120. Krugman.P. ,(1998). "What Happened to Asia?" Manuscript, MIT, January.

121. Kuzemin A, Lyashenko V.,(2007) "Fuzzy set theory approaches the basis of analysis of financial flows in the economical security system" International Journal "Information Theories & Applications" Vol.13, pp:600-609.

122. Kwok N.M., Fang G., Ha Q.P,(2009)"Moving average-based stock trading rules from particle swarm optimization", International Conference on Artificial Intelligence and Computational Intelligence, AICI 2009, 1, art. No. 5376058, pp. 149-153.

123. Lai L., (1996) "Application of evolutionary programming to transient and sub-transient parameter estimation", IEEE Transactions on Energy Conversion, Vol. 11, No. 3, pp: 1134-1140.

124. Langdon W B. 2000, Size Fair and Homologous Tree Crossovers for Tree Genetic Programming. Genetic Programming and Evolvable Machines, pp. 95-119.

125. Larsen H.L., Yager,R.R.,(2000) "A framework for fuzzy recognition technology," IEEE Trans. Syst. Man, Cybern. C, Appl., Rev., vol. 30, no. 1, pp: 65–76.

126. Lebaron B., (2006) "Agent-based computational finance: suggested readings and early research," Journal of Economic Dynamics and Control, vol. 23, no. 9-10, pp. 1487-1516.

127. Lebaron B., Arthur W.B., Palmer R., (1999) "Time series properties of an artificial stock market?" Journal of economic dynamic and control, Vol 23, pp: 1487-1526.

128. Lee C.H., Chiu M.H., (2009) "Recurrent neuro fuzzy control design for tracking of mobile robots via hybrid algorithms" Expert Systems with Applications, pp: 8993–8999.

129. Lee J.w., (2001) "Stock price prediction using reinforcement learning," in Proc. IEEE Int. Joint Conf. Neural Netw., Pusan, Korea, vol. 1, pp: 690–695.

130. Li D. (2000) "On Default Correlation: A Copula Function Approach" The Risk Metrics Group, Working Paper Number 99-07.

131. Li J., Tsang E.P.K., (1999) "Improving technical analysis predictions: an application of genetic programming" Annual Florida Artificial Intelligence International Research Conference, Orlando, Florida, USA.

132. Lim M.H., Wunsch D., Ho K.W., (2000) "Evolutionary programming methodology for portfolio selection ", IEEE/IAFE Conference on Computational Intelligence for Financial Engineering, Proceedings (CIFEr), pp: 42-46.

133. Lipson H., Bongard J., Zykov V., (2008) “Co-Evolutionary Methods in System Design and Analysis”, Cornell University, Ithaca NY 14853, USA.

134. Lohn J, Hornby G, Linden D. (2005) “Evolutionary antenna design for a NASA spacecraft”. In: O’Reilly UM, Riolo RL, Worzel B, editors. Genetic programming theory and practice II. Santa Clara: Springer, pp: 310–315.

135. Mahfoud S., Mani G., (1996) “Financial forecasting using genetic algorithms,” Journal of Applied Artificial Intelligence, Vol.10, pp: 543-565.

136. Mahfoud S., Ganesh M., (1996) "Financial forecasting using genetic algorithms." Applied Artificial Intelligence, vol.10, no.6, p.543-565.

137. Malasia K., (2006) “Artificial Neural Networks (ANN) Approach to PPG Signal Classification” International Journal of Computing and information Sciences Vol.2, pp: 100-107.

138. Marcos M.G., Tenreiro J.A., Machado, Azevedo-Perdicoúlis T.P (2009) “Trajectory planning of redundant manipulators using genetic algorithms”, Commun Nonlinear Sci Numer Simulat , pp: 2858–2869.

139. Marczyk A., (2004) “Genetic Algorithms and Evolutionary Computation”, the Talk-Origin Archive, pp: 235-239.

140. Markose S., Tsang E., (2001)”Evolutionary arbitrage for FTSE-100 index option and futures, IEEE Transaction on Evolutionary Computation vol: 43, no.1, pp: 1135-1140.

141. Martinez-Jaramillo S., Tsang E., (2009) “A heterogeneous and co-evolutionary GP-based financial market,” IEEE Transaction on Evolutionary Computation, vol. 63, no. 1, pp. 1105-1142.

142. Martinez-Jaramillo S., Tsang E., (2009) “A heterogeneous endogenous and Co evolutionary GP- Based Financial Market”, IEEE Transaction on Evolutionary Computation, vol: 63, No 1， pp: 1105-1142.

143. Mazzeo S., Loiseau I., (2004) “An Ant Colony Algorithm for the Capacitated Vehicle Routing”, Electronic Notes in Discrete Mathematics 18 (2004), pp: 181–186.

144. McKinnon.R., Huw.P.,(1997). "Over-borrowing: A Decomposition of Credit and Currency Risk." Unpublished manuscript, Harvard Business School, November.

145. Melanie M., (1993)” An Introduction to Genetic Algorithms”. MIT Press, 1996.

146. Michalewicz Z., (1996). “Genetic Algorithm +Data Structures = Evolution Programs”, Third ed. Springer-Verlag: New York.

147. Mitchison N. A., (1964) “Induction of immunological paralysis in two zones of dosage”. Royal Soc. London B161, pp: 275-292.

148. Mora A.M., Castillo P.A., Merelo J.J., (2008) “Discovering Causes of Financial Distress by Combining Evolutionary Algorithms and Artificial Neural Networks” Universidad of Granada, Spain

149. Moreno R., Gloria P., Eli R., (1998) "Asia's Financial Crisis: Lessons and Policy Responses." In Asia: Responding to Crisis. Tokyo, Asian Development Bank Institute. Also Working Paper PB98-02, Center for Pacific Basin Monetary and Economic Studies, Federal Reserve Bank of San Francisco.

150. Mühlenbein. H., Schlierkamp-Voosen D., (1993) “Predictive Models for the Breeder Genetic Algorithm”, Evolutionary Computation, Vol. 1, No. 1, pp. 25-49, 1993.

151. Nachbar R.B., (2000) “Molecular evolution: automated manipulation of hierarchical chemical topology and its application to average molecular structures”, Genet Program Evol; 1: pp: 57-94.

152. Nasraoui O., Gonzalez F., Cardona C., Rojas C., Dasgupta D., (2003)"A Scalable Artificial Immune System Model for Dynamic Unsupervised Learning”, Genetic and Evolutionary Computation, pp:196-199.

153. Neal M., (2003)"Meta-stable Memory in an Artificial Immune Network”, Artificial Immune Systems, pp: 168-180.

154. Neely C., Weller, P., Ditmar R., (1997) “Is technical analysis in the foreign exchange market profitable? A genetic programming approach” Journal of Financial and Quantitative, pp: 405-26.

155. Nenortaite J., Simutis R., (2004) “Stocks’ Trading System Based on the Particle Swarm Optimization Algorithm”, ICCS 2004, LNCS 3039, pp: 843–850.

156. Ooka R., Komamura K., (2009) “Optimal design method for building energy systems using genetic algorithms” Building and Environment, pp: 1538–1544.

157. Oosterbeek H., Sloof R., van de Kuilen G. (2004) “ Differences in ultimatum game experiments: evidence from a meta-analysis”, Exp Econ 7:171–188.

158. Osborne M. J., Ariel R. A., (1994) “Course in Game Theory”, Cambridge, MA: MIT, 1994.

159. Parpinelli R.S., Lopes H.S., Freitas A.A.,(2002)"Data Mining with an Ant Colony Optimization Algorithm",IEEE Trans on Evolutionary Computation, special issue on Ant Colony Algorithms,Vol6, No,4, pp:1223:1234.

160. Poundstone W., (1992) Prisoner's Dilemma Doubleday, NY NY.

161. Price T.C., (1997) "Using co-evolutionary programming to simulate strategic behavior in markets, Journal of evolutionary economics, vol 7, pp: 219-254.

162. Qin Z., Li X., (2008) "Option Pricing Formula for Fuzzy Financial Market" Journal of Uncertain Systems Vol.2, No.1, pp: 17-21.

163. Radelet S., Sachs J., (1998), "The Onset of the East Asian Financial Crisis",Manuscript,Harvard Institute for International Development,March.

164. Rechenberg I., (1971)"Evolutionsstrategie", Optimierung technischer Systeme nach Prinzipien der biologischen Evolution (PhD thesis). Reprinted by Fromman-Holzboog (1973)

165. Repoussis P.P., Tarantilis C.D., Bräysy O., Ioannou G., (2010)"A hybrid evolution strategy for the open vehicle routing problem",Computers and Operations Research, 37 (3), pp: 443-455.

166. Sato S., K. Otori A., Takizawa H., Sakai Y., Ando(2002) "Applying genetic algorithms to the optimum design of a concert hall" Journal of Sound and Vibration, vol.258, no.3, pp: 517-526 (2002).

167. Schwefel H.P., (1995)" Evolution and Optimum Seeking", New York: Wiley & Sons 1995.

168. Schwenker F., Marinai S., (2006) "Artificial Neural Networks in Pattern Recognition", Second IAPR Workshop, ANNPR 2006, Ulm, Germany, pp: 4087.

169. Shellam G. R., Nossal G. J. V., (1968) "The mechanism of induction of immunological paralysis", IV. The effects of ultra-low doses of flagellin. Immunology 14, pp: 273-284.

170. Snowden R., (2006) "Teach Yourself Freud", McGraw-Hill, pp: 105–107.

171. King M., Wadhwani S., (1990) "Transmission of volatility between stock markets," Review of Financial Studies, vol. 3, no. 1, pp. 5-33.

172. Song W., Li C.H., Park S.C., (2009) "Genetic algorithm for text clustering using ontology and evaluating the validity of various semantic similarity measures ", Expert Systems with Applications, pp: 9095–9104.

173. Song Y.H., Chou C.S.V., Min Y., (1999)" Large-Scale Economic Dispatch by Artificial Ant Colony Search Algorithms", Electric Machines and Power Systems, 27, pp: 679–690.

174. Spears W. M., De Jong. K. A.,(1991) "An Analysis of Multi-Point Crossover", In Foundations of Genetic Algorithms, J. E. Rawlins (Ed.), pp. 301-315, 1991.

175. Storn R., (1996) "On the usage of differential evolution for function optimization", Biennial Conference of the North American Fuzzy Information Processing Society, (NAFIPS), pp: 519-523.

176. Storn R., Price K., (1997) "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces"., Journal of Global Optimization, pp: 341-359.

177. Streichert F., (2002)"Introduction to Evolutionary Algorithms "University of Tubingen".

178. Tada T., Takemori T., (1974) "Selective roles of thymus-derived lymphocytes in the antibody response. I. Differential suppressive effect of carrier-primed T cells on hapten-specific IgM and IgG antibody responses", J. Exp. Med. 140, pp: 239-252.

179. Tanaka-Yamawaki M., Tokuoka S., (2006) "Minority game as a model for the artificial financial markets," Proceedings of 2006 IEEE Congress on Evolutionary Computation, pp. 2157-2162.

180. Taylor M. P., Allen H., (1992) "The use of technical analysis in the foreign exchange market," Journal of International Money and Finance, pp. 304-314.

181. Teich J., Zitzler E., Bhattacharyyav S., (2009) "Buffer Memory Optimization in DSP Applications: An Evolutionary Approach", Swiss Federal Institute of Technology.

182. Theimer M., Parastatidis S., Hey T., (2007) "An Evolutionary Approach to Realizing the Grid Vision", Microsoft Corporation.

183. Timmis J., (2000) "Artificial Immune Systems: A novel data analysis technique inspired by the immune network theory", PhD thesis, Department of Computer Science, University of Wales, Aberystwyth. Ceredigion. Wales.,

184. Timmis J., Neal M., Hunt J., (2000) "An artificial immune system for data analysis", BioSystems 55 (1), pp: 143–150.

185. Toth P., Vigo D., (2003) "The Granular Tabu Search and Its Application to the Vehicle-Routing Problem", informs Journal on Computing, Vol. 15, No. 4, fall 2003, pp: 333-346.

186. Tsai K.Y., Wang F.S., (2005) "Evolutionary optimization with data collocation for reverse engineering of biological networks", Bioinformatics, pp: 1180-1198.

187. Tversky A., (2004) Preference, belief, and similarity: selected writings. Cambridge: MIT Press.

188. Vajediana M., Saradjianb M.A., Sharifi c.,(2000) "Estimation of source parameters of 2003 earthquake from images using genetic algorithm technique, Center of Excellence in Geomatics Eng. and Disaster Management.

189. Van Valen Leigh, (1973) "A new evolutionary law," Evolutionary Theory, vol. 1, pp. 1-30.

190. Varela F.J, Coutinho A., (1991) "Second generation immune networks". Immunol. Today 12 (5): pp: 159–66,PMID 1878127.

191. Vovk V., Shafer G., (2008) "The Game-Theoretic Capital Asset Pricing Model", International Journal of Approximate Reasoning archive Volume 49 Issue 1, September, 2008 Pp 175-197

192. Wang S., Archer N.P., (1998) "A neural network based fuzzy set model for organizational decision making," IEEE Trans. Syst. Man, Cybern. C, Appl. Rev., vol. 28, pp: 194–203.

193. White J.A., Garrett S.M., (2003) "Improved Pattern Recognition with Artificial Clonal Selection", Artificial Immune Systems，pp:181-193.

194. Wilson G., Banzhaf W.,(2009) "Prediction of interday stock prices using developmental and linear genetic programming", Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5484 LNCS, pp. 172-181.

195. Wilson S. R., Cui W., (2004) "Applications of simulated annealing to peptides", Peptide Science, Vol, 29 Issue 1, pp: 225 – 235.

196. Yao J., Poh H.L., (1995)"Forecasting the KLSE index using neural networks," IEEE Int. Joint Conf. Neural Netw., Perth, WA, Australia, vol. 2, pp: 1012–1017.

197. Yıldırıma G., Özger M., (2009) "Neuro-fuzzy approach in estimating Hazen–Williams friction coefficient for small-diameter polyethylene pipes", Advances in Engineering Software, pp: 593–599.

198. Yoon Y., Swale J., (1991) “Prediction of stock price performance: A neural network approach,” IEEE Conf, vol. 2, pp: 156–162.

199. Yuan X.E., Zou Y., (2009) “Technology program financial forecast model based on CACO-SVM”, International Workshop on Intelligent Systems and Applications, ISA 2009, art. No, 5073158.

200. Yuan X.H., Wang C., Zhang Y.C., Yuan Y.B., (2004) “A survey on application of particle swarm optimization to electric power systems“, cnki: ISSN: 1000-3673.0.2004-19-003.

201. Zhang J., Avasarala V., Subbu R., (2010) “Evolutionary optimization of transition probability matrices for credit decision-making”, European Journal of Operational Research, 200 (2), pp. 557-567.

202. Zhang M., (2007)”Genetic programming techniques for multi-class object recognition”, Genetic and evolutionary computation for image processing and analysis. Cairo: Hindawi; 2007. pp: 349-370.