Chapter 11
Multi-agent Approaches to Economic Modeling: Game Theory, Ensembles, Evolution and the Stock Market

Abstract A multi-agent system that learns by using neural networks is implemented to simulate the stock market. Each committee of agents, which is regarded as a player in a game, is optimized by continually adapting the architecture of the agents through the use of genetic algorithms. The proposed procedure is implemented to simulate trading of three stocks, namely, the Dow Jones, the NASDAQ and the S&P 500.

11.1 Introduction

In this chapter a committee of agents is used to simulate the stock market (Marwala et al. 2001). Each committee of agents is viewed as a player in a game and, therefore, a game theoretic framework is applied in this chapter (Marwala et al. 2001). These players in a game compete and cooperate (Perrone and Cooper 1993). The committee of agents is optimized using a genetic algorithm (Holland 1975; Goldberg 1989). Perrone and Cooper (1993) introduced a committee of networks, which optimizes the decision-making of a population of non-linear predictive models (Bishop 1995). They attained this by assuming that the trained predictive models were accessible and then allocating, to each network, a weighting factor, which specifies the role that the network has on the total decision of a population of networks. The drawback of their proposal is that, in a condition where the problem is altering such as the stock market, the technique is not sufficiently elastic to permit for the dynamic evolution of the population of networks.

This chapter aims to relax this constraint on the committee technique by ensuring that the individual networks that create the committee are permitted to dynamically evolve as the problem evolves using genetic programming (Michalewicz 1996), and this was first conducted by Marwala et al. (2001).

The parameters describing the design of the networks, such as the number of hidden units, which form a committee, are defined as design variables and are

T. Marwala, *Economic Modeling Using Artificial Intelligence Methods*, Advanced Information and Knowledge Processing, DOI 10.1007/978-1-4471-5010-7_11,

permitted to evolve as the trading environment evolves. The network characteristics that are appropriate for survival substitute for those that are not appropriate. On applying a genetic algorithm to choose the appropriate individuals, three steps are followed: (1) crossover of network attributes within the population; (2) Mutation of each individual attributes; and (3) reproduction of the successful attributes. The simple crossover, the binary mutation, and roulette wheel reproduction techniques are used.

In conclusion, the proposed technique is applied to simulate the trading of three stocks. The scalability of the number of agents and players in the simulations with respect to computational time were investigated. The evolution of the complexity of the simulation as the players participate in more trading was also investigated. The next section describes game theory, which is a framework that is used to set up the simulation.

11.2 Game Theory

In this chapter, we apply game theory to model the stock market. Game theory essentially consists of players, set of actions (strategy), and pay-off function (Villena and Villena 2004; Ross 2006; van den Brink et al. 2008). Game theory has been applied to many areas of activity including economics (van den Brink et al. 2008), procurement of land (Hui and Bao 2013), auction (Laffont 1997), the hotel industry (Wei et al. 2012), facial recognition (Roy and Kamel 2012), medicine (McFadden et al. 2012) and computer science (Papadimitriou 2001). There are many types of games and, in this chapter we will illustrate the well-known prisoner's dilemma problem. Suppose players A and B are arrested for a crime and they are put into separate cells. They are given choices to either cooperate or defect, and this is represented in Table 11.1.

Game theory can be used to solve the problem in Table 11.1. In this table, if a player remains slilent, he gets either 2 months in prison or serves 1 year in prison. If he pleas bargains, he gets either 6 months in prison or goes free. According to John von Neumann, the best strategy is the one that guarantees you maximum possible outcome even if your opponent knew what choice you were going to make. In this case, the best strategy is to enter a plea bargain. The concept of Nash equilibrium states that the best strategy for each player is such that every player's move is a best

Table 11.1 Illustration of the prisoner's dilemma

	Prisoner B remains silent	Prisoner B plea bargains
Prisoner A remains silent	Each serves 2 months	Prisoner A serves 1 year Prisoner B goes free
Prisoner A plea bargains	Prisoner B serves 1 year Prisoner A goes free	Each serves 6 months

response to the other players' move. Therefore, entering a plea bargain is a Nash equilibrium. Of course, all these assume that each player is rational and aims to maximize pay-off (Beed and Beed 1999).

Hodgson and Huang (2012) compared evolutionary game theory and evolutionary economics and concluded that these methods improve understanding of structures and causal processes, whereas Christin (2011) applied game theory in network security games and concluded that it was vital to understand reasons of different players in a network to design systems to support appropriate outcome.

Hanauske et al. (2010) extended the hawk-dove game by a quantum method and demonstrated that evolutionary stable strategies, which are not forecast by traditional evolutionary game theory and where the total economic population applies a non-aggressive quantum strategy, can also emerge.

McCain (2009) studied theoretical and experimental results in game theory and, the neo-classical notion of inter-temporal discrepancy in choice to debate that the motivational theory which is common between neo-classical economics and non-cooperative game theory, mistakenly assumes that commitment never occurs in human decisions. They concluded that the conditions that favor commitment, other than alterations of an assumed utility, function to account for non-self-regarding motivations are advantageous in behavioral economics and game theory.

Roth (2002) applied game theory to the design of the entry level labor market for American doctors and the auctions of radio spectrum. He proposed that experimental and computational economics complemented game theory for design and debated that some of the tasks confronting both markets include handling with associated types of complementarities.

The example illustrated at the beginning of this chapter was a two player game. It becomes extremely difficult to deal with multiple player games and a computational technique has been developed and is able to handle, to some extent, multiple player games and this procedure is called a multi-agent system and is the subject of the next section.

11.3 Multi-agent Systems

A multi-agent system is, by definition, a system of multiple agents. An agent is an object that is autonomous, perceives its environment, and acts on its environment is intelligent, and operates autonomously in that environment (Russell and Norvig 2003; Franklin and Graesser 1996; Kasabov 1998). Agents have the following characteristics (Kasabov 1998):

- They are autonomous.
- They are flexible, reactive, proactive and social.
- They have control capability.

To illustrate the working of a multi-agent system, a well-known swarm intelligence theory can be used. In this example, agents or birds (in the case of the swarming of birds) operate using two simple rules. These are that, in seeking the next move, a bird considers the best position it has encountered and the best position the entire flock has encountered (where other birds are going). Using these simple rules, the swarm is able to solve very complex problems. More details on these can be found in the literature (Marwala 2009, 2010, 2012; Marwala and Lagazio 2011).

Teweldemedhin et al. (2004) presented an agent-based, bottom-up modeling technique to develop a simulation tool for estimating and predicting the spread of the human immunodeficiency virus (HIV) in a given population. They developed a simulation instrument to understand the spread of HIV.

Hurwitz and Marwala (2007) studied the deed of bluffing, which has perplexed game designers. They asserted that, the very act of bluffing was even open for debate, introducing additional difficulty to the procedure of producing intelligent virtual players that can bluff, and therefore play, truthfully. Through the application of intelligent, learning agents, and prudently designing agents, an agent was found to learn to predict its opponents' reactions based on its own cards and actions of other agents. They observed that, an agent can learn to bluff its opponents, with the action not indicating an "irrational" action as bluffing is usually regarded, but as an act of maximizing returns by an actual statistical optimization. They applied a TD lambda learning algorithm to adapt a neural network based agent's intelligence and demonstrated that agents were able to learn to bluff without outside encouragement.

Abdoos et al. (2011) applied a multi agent technique for traffic light control in non-stationary environments. The results they obtained indicated that the proposed method performed better than a fixed time technique under different traffic demands. Elammari and Issa (2013) applied model driven architecture to develop multi-agent systems, while Chitsaz and Seng (2013) successfully applied a multi agent system for medical image segmentation.

Stroeve et al. (2013) successfully applied event sequence analysis and multi-agent systems for safety assessments of a runway incursion scenario, while El-Menshawy et al. (2013) verified, successfully, conformance of multi-agent commitment-based protocols.

Montoya and Ovalle (2013) applied multi-agent systems for energy consumption by positioning a reactive inside wireless sensor networks, while Khalilian (2013) applied multi agent systems and data mining approaches towards a smart advisor's framework. Liu et al. (2012b) applied, successfully, multi-agent systems to bidding mechanism in an electricity auction.

In this chapter, the agent architecture implemented is shown in Fig. 11.1 and the multi-agent system is shown in Fig. 11.2. It has intelligence capability, which is a committee of a combination of multi-layer perceptrons and radial basis function network.

The agent is able to adapt using genetic programming, by adapting the committee structure. The next section describes neural networks which are applied to enable the agent to be intelligent.

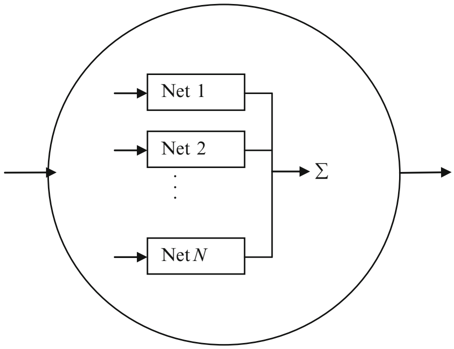

Fig. 11.1 Illustration of an agent

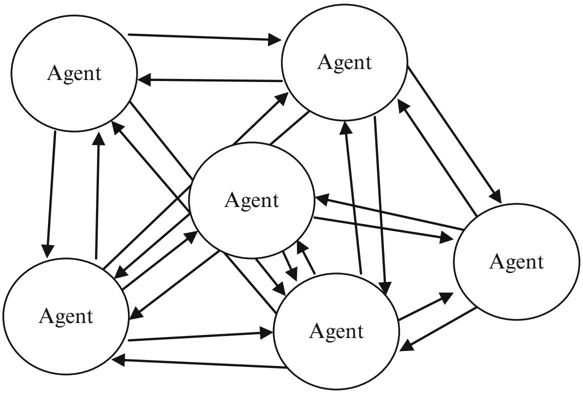

Fig. 11.2 Illustration of a multi-agent system

11.4 Neural Networks

This section describes neural networks which are used to model data. Neural networks are, by definition, mathematical models that are inspired by the way the human brain processes information. This section describes the type of neural networks that relate some information to another, and these are called supervised neural networks. Supervised neural networks take input data x and relate this to the output data y. In this chapter, we apply two types of supervised neural networks and these are radial basis functions and multi-layer pereceptron.

Radial basis function (RBF) is a neural network technique which is based on the distance of the data set from its origin (Bishop 1995). The RBF is usually structured with a single hidden layer of units with an activation function that is chosen from a type of functions called basis functions. The activation of the hidden units is characterized by a non-linear function of the distance between the input vector and a vector indicating the centers of gravity of the data (Bishop 1995). Despite the fact that the RBF is similar to a multi-layer perceptron (MLP), radial basis function networks have the following advantages:

- They are faster to train than the MLP networks
- They are less prone to problems with non-stationary inputs

The RBF network can be defined mathematically as follows (Buhmann and Ablowitz 2003; Marwala and Lagazio 2011):

$$y_k\left(\{x\}\right)=\sum_{j=1}^{M}w_{jk}\phi\left(\left\|\{x\}-\{c\}_j\right\|\right) \tag{11.1}$$

where, w_{jk} are the output weights, relating a hidden unit and an output unit, M shows the number of hidden units, $\{c\}_j$ is the center for the jth neuron, $\phi\left(\{x\}\right)$ is the jth non-linear activation function, $\{x\}$ is the input vector, and $k=1,2,3,\ldots,M$ (Bishop 1995; Marwala and Lagazio 2011). Radial basis functions are trained in this chapter using the k-nearest neighbor method to estimate the centers and the weights are then estimated using the pseudo-inverse technique, and the details of these can be found in Bishop (1995).

Radial basis functions have been successfully applied to many complex problems such as voice transformation (Nirmal et al. 2013), image analysis of deformation (Biancolini and Salvini 2012), analysis of hemodynamics pattern flow (Ponzini et al. 2012), analysis of gene expression data (Liu et al. 2012a), and the prediction of logistics demand (Chen et al. 2012).

The MLP is a feed-forward neural network technique that approximates a relationship between sets of input data and a set of output data. It applies three or more layers of neurons, also called nodes, with non-linear activation functions. It can distinguish data that is not linearly separable or separable by a hyper-plane.

The MLP neural network consists of multiple layers of computational components normally inter-connected in a feed-forward manner (Haykin 1999; Hassoun 1995; Marwala 2012). Every neuron in one layer is connected to the neurons of the subsequent layer and this can be mathematically represented as follows (Haykin 1999):

$$y_k=f_{outer}\left(\sum_{j=1}^{M}w_{kj}^{(2)}f_{inner}\left(\sum_{i=1}^{d}w_{ji}^{(1)}x_i+w_{j0}^{(1)}\right)+w_{k0}^{(2)}\right) \tag{11.2}$$

Here, $w_{ji}^{(1)}$ and $w_{ji}^{(2)}$ are weights in the first and second layers, correspondingly, from input i to hidden unit j, M is the number of hidden units, d is the number of output units, while $w_{j0}^{(1)}$ and $w_{k0}^{(2)}$ are the weight parameters that indicate the biases for the hidden unit j and the output unit k. These weight parameters can be viewed as a mechanism that enables the model to understand the data. The weight vector in Eq. 11.2 is identified using the scaled conjugate gradient technique that is based on the maximum-likelihood method (Møller 1993).

The MLP has been successfully applied in many areas and these include power transformer diagnosis (Souahlia et al. 2012), automatic musical intrument recognition (Azarloo and Farokhi 2012), diagnosing of cervical cancer (Sokouti et al. 2012), automatic vehilce type classification (Daya et al. 2012), fingerprint spoof detection (Pereira et al. 2012), and intrusion detection (Ahmad et al. 2011).

The agent proposed in Fig. 11.1 contains a group of neural networks that collectively make a decision, and this is either called a committee approach or an ensemble of networks and is the subject of the next section.

11.5 Ensembles of Networks

When a group of neural networks are used to collectively make a decision, then this is known as an ensemble approach. There are many types of ensembles and, in this chapter, we discuss few of these and these are: bagging, boosting, stacking, and evolutionary committees.

11.5.1 Bagging

Bagging is a method that is based on an amalgamation of models fitted to bootstrap samples of a training data set to decrease the variance of the prediction model (Breiman 1996). Bagging fundamentally involves randomly selecting a section of the training data, training a model with this selection, and then iterating this procedure and then all trained models are pooled together with equal weights to form an ensemble. Bagging has been successfully applied in many areas such as the detection of obsessive compulsive disorder (Parrado-Hernández et al. 2012), diagnosing of arrhythmia beats (Mert et al. 2012), fraud detection tools (Louzada and Ara 2012), identification of MicroRNA Precursors (Jha et al. 2012), land-cover classification (Ghimire et al. 2012), and intrusion detection (Syarif et al. 2012).

11.5.2 Boosting

Boosting is a method that incrementally constructs an ensemble by training each new model with data that the heretofore trained model misclassified. Then the ensemble, which is a combination of all trained models, is used for prediction. Jasra and Holmes (2011) successfully applied stochastic boosting algorithms which used sequential Monte Carlo methods, while Leitenstorfer and Tutz (2011) successfully applied boosting methods to estimate single-index models. Other successful applications of boosting include object classification (Piro et al. 2013), categorization of natural scenes (Nock et al. 2012), automatic anatomy detection (Tajbakhsh et al. 2012), multi-view face pose classification (Yun and Gu 2012), and automatic audio tagging (Foucard et al. 2012).

11.5.3 Stacking

The general approach in mathematical modeling is that one chooses from a set of models by comparing them on data that was not used to train the models. This insight can also be applied to choose a model using a method called cross-validation (Bishop 1995). This is achieved by apportioning the data set into a *held-in* data set, which is used to train the models, and a *held-out* data set which is used to test the trained models (Sill et al. 2009; Marwala 2012).

Stacking uses performance of the model on the held-out data to combine the models instead of selecting from them the best performing model when tested on the held-out data and this gives an ensemble that performs better than any single one of the trained models (Wolpert 1992). Stacking has been successfully applied to many areas such as instance-based ensemble learning algorithms (Homayouni et al. 2010), real estate appraisal (Graczyk et al. 2010), and metabonomic applications (Lienemann et al. 2009).

11.5.4 Evolutionary Committees

Evolutionary committees are methods that are adaptive techniques that adapt to the environmental changes. This is usually achieved by evolving the weighting function that defines the contribution of each individual technique, with respect to the overall outcome of the committee.

Marwala (2009) introduced committees of networks for missing data estimation. The first committee of networks was made of multi-layer perceptrons (MLPs), support vector machines (SVMs), and radial basis functions (RBFs); and entailed the weighted combination of these three networks. The second, third, and fourth committees of networks were evolved using a genetic programming method and used the MLPs, RBFs and SVMs, respectively. The committees of networks were applied, collectively, with a hybrid particle swarm optimization and genetic algorithm technique for missing data estimation. When they were tested on an artificial taster, as well as HIV datasets, and then compared to the individual MLPs, RBFs, and SVMs for missing data estimation, the committee of networks approach was observed to give better results than the three approaches acting in isolation. Nonetheless, this improvement came at a higher computational load than the individual methods. In addition, it was observed that evolving a committee technique was a good way of constructing a committee.

In this chapter, we apply the three member ensemble which is shown in Fig. 11.3. The ideas presented in this section are an adaptation of the work done by Perrone and Cooper (1993) where they introduced the concept of a committee of networks and confirmed that this committee provides results that are more reliable than when using networks in isolation.

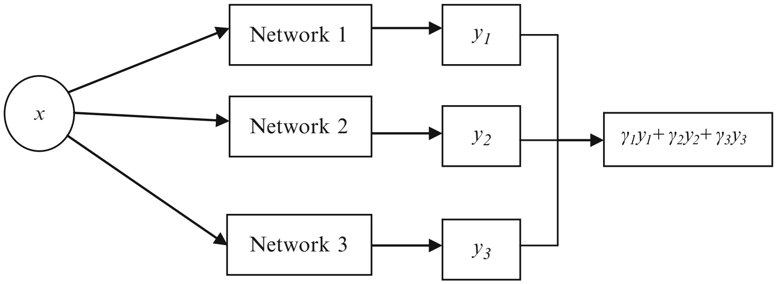

Fig. 11.3 Illustration of committee of networks

The mapping of the input x and output y can be expressed as the desired function plus an error. For notational accessibility, the mapping functions are assumed to have single outputs y_1, y_2, and y_3. This can be easily adapted to multiple outputs as follows (Perrone and Cooper 1993):

$$y_1(x) = h(x) + e_1(x) \tag{11.3}$$

$$y_2(x) = h(x) + e_2(x) \tag{11.4}$$

$$y_3(x) = h(x) + e_3(x) \tag{11.5}$$

Here, $h(\cdot)$ is the estimated mapping function; and $e(\cdot)$ is the error.

The mean square errors (MSE) for model $y_1(x)$, $y_2(x)$, and $y_3(x)$ may be expressed as follows (Perrone and Cooper 1993):

$$E_1 = \varepsilon\left[\{y_1(x) - h(x)\}^2\right] = \varepsilon\left[e_1^2\right] \tag{11.6}$$

$$E_2 = \varepsilon\left[\{y_2(x) - h(x)\}^2\right] = \varepsilon\left[e_2^2\right] \tag{11.7}$$

$$E_3 = \varepsilon\left[\{y_2(x) - h(x)\}^2\right] = \varepsilon\left[e_3^2\right] \tag{11.8}$$

Here, $\varepsilon\,[\bullet]$ denotes the expected value and corresponds to the integration over the input data, and is defined as follows (Perrone and Cooper 1993):

$$\varepsilon\left[e_1^2\right] \equiv \int e_1^2(x)p(x)dx \tag{11.9}$$

$$\varepsilon\left[e_2^2\right] \equiv \int e_2^2(x)p(x)dx \tag{11.10}$$

$$\varepsilon\left[e_3^2\right] \equiv \int e_3^2(x)p(x)dx \tag{11.11}$$

Here, $p\,[\bullet]$ is the probability density function; and $d\,[\bullet]$ is a differential operator. The average MSE of the three networks acting separately may be expressed as follows (Perrone and Cooper 1993):

$$\begin{aligned} E_{AV} &= \frac{E_1(x) + E_2(x) + E_3(x)}{3} \\ &= \frac{1}{3}\left(\varepsilon\left(e_1^2\right) + \varepsilon\left(e_2^2\right) + \varepsilon\left(e_3^2\right)\right) \end{aligned} \tag{11.12}$$

11.5.4.1 Equal Weights

The output of the committee is the average of the outputs from the three networks. The committee prediction may be expressed in the following form, by giving equal weighting functions (Perrone and Cooper 1993):

$$y_{COM} = \frac{1}{3}\left(y_1(x) + y_2(x) + y_3(x)\right) \tag{11.13}$$

The MSE of the committee can be written as follows:

$$\begin{aligned} E_{COM} &= \varepsilon\left[\left(\frac{1}{3}\left\{y_1(x) + y_2(x) + y_3(x)\right\} - \frac{1}{3}\left[h(x) + h(x) + h(x)\right]\right)^2\right] \\ &= \varepsilon\left[\left(\frac{1}{3}\left\{\left[y_1(x) - h(x)\right] + \left[y_2(x) - h(x)\right] + \left[y_3(x) - h(x)\right]\right\}\right)^2\right] \\ &= \varepsilon\left[\left(\frac{1}{3}\left\{e_1 + e_2 + e_3\right\}\right)^2\right] \\ &= \frac{1}{9}\left(\varepsilon\left[e_1^2\right] + 2\left(\varepsilon\left[e_1e_2\right] + \varepsilon\left[e_1e_2\right] + \varepsilon\left[e_2e_3\right] + \varepsilon\left[e_1e_3\right]\right) + \varepsilon\left[e_2^2\right] + \varepsilon\left[e_3^2\right]\right) \end{aligned} \tag{11.14}$$

If it is assumed that the errors (e_1, e_2, and e_3) are uncorrelated then

$$\varepsilon[e_1e_2] = \varepsilon[e_1e_2] = \varepsilon[e_2e_3] = \varepsilon[e_1e_3] = 0 \tag{11.15}$$

Substituting Eq. 11.15 in Eq. 11.14, the error of the committee can be related to the average error of the networks acting individually as follows (Perrone and Cooper 1993):

$$\begin{aligned} E_{COM} &= \frac{1}{9}\left(\varepsilon\left[e_1^2\right] + \varepsilon\left[e_2^2\right] + \varepsilon\left[e_3^2\right]\right) \\ &= \frac{1}{3}E_{AV} \end{aligned} \tag{11.16}$$

Equation 11.16 indicates that the MSE of the committee is one-third of the average MSE of the individual technique. This implies that the MSE of the committee is always equal to or less than the average MSE of the three methods acting individually.

11.5.4.2 Variable Weights

The three networks might not essentially have the same predictive capability. To accommodate the strength of each technique, the network should be given suitable weighting functions. It will be explained later how these weighting functions will be evaluated when there is no prior knowledge of the strength of each approach.

The output of the ensemble may be defined as the combination of the three independent methods with estimated weighting functions as:

$$y_{COM} = \gamma_1 y_1(x) + \gamma_2 y_2(x) + \gamma_3 y_3(x) \tag{11.17}$$

where γ_1, γ_2, and γ_3 are the weighting functions and $\gamma_1 + \gamma_2 + \gamma_3 = 1$. The MSE due to the weighted committee can be written as follows (Marwala 2000):

$$\begin{aligned} E_{COM} &= \varepsilon\left[(\gamma_1 y_1(x) + \gamma_2 y_2(x) + \gamma_3 y_3(x) - [\gamma_1 h(x) + \gamma_2 h(x) + \gamma_3 h(x)])^2\right] \\ &= \varepsilon\left[(\gamma_1 [y_1(x) - h(x)] + \gamma_2 [y_2(x) - h(x)] + \gamma_3 [y_3(x) - h(x)])^2\right] \\ &= \varepsilon\left[(\gamma_1 e_1 + \gamma_2 e_2 + \gamma_3 e_3)^2\right] \end{aligned} \tag{11.18}$$

Equation 11.18 may be rewritten in Lagrangian form as follows (Perrone and Cooper 1993):

$$E_{COM} = \varepsilon\left[(\gamma_1 e_1 + \gamma_2 e_2 + \gamma_3 e_3)^2\right] + \lambda\,(1 - \gamma_1 - \gamma_2 - \gamma_3) \tag{11.19}$$

where λ is the Lagrangian multiplier. The derivative of the error in Eq. 11.19 with respect to γ_1, γ_2, γ_3 and λ may be calculated and equated to zero as follows (Perrone and Cooper 1993):

$$\frac{dE_{COM}}{d\gamma_1} = 2e_1\varepsilon\left[(\gamma_1 e_1 + \gamma_2 e_2 + \gamma_3 e_3)\right] - \lambda = 0 \tag{11.20}$$

$$\frac{dE_{COM}}{d\gamma_2} = 2e_2\varepsilon\left[(\gamma_1 e_1 + \gamma_2 e_2 + \gamma_3 e_3)\right] - \lambda = 0 \tag{11.21}$$

$$\frac{dE_{COM}}{d\gamma_3} = 2e_3\varepsilon\left[(\gamma_1 e_1 + \gamma_2 e_2 + \gamma_3 e_3)\right] - \lambda = 0 \tag{11.22}$$

$$\frac{dE_{COM}}{d\lambda} = 1 - \gamma_1 - \gamma_2 - \gamma_3 = 0 \tag{11.23}$$

In solving Eqs. 11.20, 11.21, 11.22, and 11.23, the minimum error is obtained when the weights are (Perrone and Cooper 1993):

$$\gamma_1 = \frac{1}{1 + \frac{\varepsilon[e_1^2]}{\varepsilon[e_2^2]} + \frac{\varepsilon[e_1^2]}{\varepsilon[e_3^2]}} \tag{11.24}$$

$$\gamma_2 = \frac{1}{1 + \frac{\varepsilon[e_2^2]}{\varepsilon[e_1^2]} + \frac{\varepsilon[e_2^2]}{\varepsilon[e_3^2]}} \tag{11.25}$$

$$\gamma_3 = \frac{1}{1 + \frac{\varepsilon[e_3^2]}{\varepsilon[e_1^2]} + \frac{\varepsilon[e_3^2]}{\varepsilon[e_2^2]}} \tag{11.26}$$

Equations 11.24, 11.25, and 11.26 may be generalized for a committee with n-trained networks and may be written as follows (Perrone and Cooper 1993):

$$\gamma_i = \frac{1}{\sum_{j=1}^{n} \frac{\varepsilon[e_i^2]}{\varepsilon[e_j^2]}} \tag{11.27}$$

From Eq. 11.27, the following conditions may be derived as follows (Marwala 2000):

$$\varepsilon[e_1^2] = \varepsilon[e_2^2] = \varepsilon[e_3^2] \Rightarrow \gamma_1 = \gamma_2 = \gamma_3 = \frac{1}{3} \tag{11.28}$$

$$\varepsilon[e_3^2] < \varepsilon[e_2^2] < \varepsilon[e_1^2] \Rightarrow \gamma_1 < \gamma_2 < \gamma_3; \gamma_3 > \frac{1}{3} \tag{11.29}$$

$$\varepsilon[e_1^2] < \varepsilon[e_2^2] < \varepsilon[e_3^2] \Rightarrow \gamma_3 < \gamma_2 < \gamma_3; \gamma_1 > \frac{1}{3} \tag{11.30}$$

11.6 Genetic Algorithms

The multi-agent system proposed in this chapter is adaptive and this is enabled by a genetic algorithm. Genetic algorithms were enthused by Darwin's theory of natural evolution. In natural evolution, members of a population compete with each other to survive and reproduce. Evolutionary successful individuals reproduce, while weaker members disappear. Consequently, the genes that are successful are probably going to spread within the population. This natural optimization technique is applied in this chapter to optimize the decision of a committee of agents shown in Fig. 11.3. This essentially allows an agent to become better, based on how well it performed and, in this chapter, on trading in the stock market.

The basic genetic algorithm proposed by Holland (1975) is applied. The algorithm acts on a population of binary-string chromosomes. These chromosomes are acquired by utilizing the Gray algorithm. Each of these strings is a discretized representation of a point in the search space. Here we are searching for the most optimum combination of architectures that form a committee and that give the least errors. Consequently, the fitness function is the error offered by committee of agents. On producing a new population, three operators are executed: (1) crossover; (2) mutation; (3) and reproduction.

Similar to natural evolution, the probability of mutation happening is lower than that of crossover or reproduction. The crossover operator combines genetic information in the population by cutting pairs of chromosomes at random points along their length and exchanging over the cut sections. This operator has a potential of connecting successful operators together. Simple crossover is applied in this chapter. The mutation operator picks a binary digit of the chromosomes at random and inverts it. This has the potential of introducing to the population new information. Reproduction takes successful chromosomes and reproduces them in accordance to their fitness function. The fit parameters are allowed to reproduce and the weaker parameters are removed. This is conducted using the roulette wheel procedure.

Genetic algorithms have been applied successfully in many areas such as content based image retrieval (Syam and Rao 2013), variable selection in solar radiation estimation (Will et al. 2013), non-destructive characterization of tie-rods (Gentilini et al. 2013), assembly line worker assignment (Mutlu et al. 2013), sheep farming (Del Pilar Angulo-Fernandez et al. 2013), and power consumption (Shen and Zhang 2013).

11.7 Simulating the Stock Marketing

In this chapter, we apply a multi-agent system to model the stock market. Multi-agent systems have been applied to stock markets in the past (Tirea et al. 2012; Liu and Cao 2011; Yoshikazu and Shozo 2007; Ikeda and Tokinaga 2004). The structure that is proposed consists of committees of agents forming a player in the stock market. The simulation framework consists of a population of these players that compete for a fixed number of stocks. The agents learn through the use of neural networks. The structure of each agent evolves using a genetic algorithm such that its contribution to the overall function of a committee adapts to the evolutionary time-varying nature of the problem. The characteristics of the agents that evolve are the number of hidden units and the weight contribution of each network towards a player. The number of hidden units is constrained to fall within a given space, in this study 1 and 10. Each committee of agents, known as a player, trades stocks with other agents and when prices of stocks are announced, the players trade by following these rules:

- Once a price is announced, the committees look at the current price and the future price of stocks. The future price is determined from the agents that learn using neural networks. For a player, the predicted price is the average of the prediction of each agent within that particular player.
- If the predicted price of a stock is lower than the current price, then the player tries to sell the stock. If the predicted price for the stock is higher than the current price, then the committee tries to buy the stock.
- At any given stage, the committee is only prepared to sell the maximum of 40 % of the volume of stocks it has.
- The amount of stocks that a committee buys or sells depends on, amongst other factors, the predicted price. If the predicted price of a particular stock is x % higher than the current price, the committee tries to acquire x % of the volume available on the market of that particular stock. This simulation is started by choosing the number of players that participate in the trading of stocks, together with the number of agents that form a player. Then the agents are trained by randomly assigning the number of hidden units to fall in the interval [1 10] and assigning weighting functions of the committee. The agents are trained using the data from the previous 50 trading days. The trained agents are grouped into their respective players and are then used to predict the next price, given the current price. The simulation followed in this chapter is shown in Fig. 11.4.

After 50 days of trading have elapsed, the performance of each agent and the weighting functions are evaluated and these are transformed into 8 bits and each player exchanges bits with other players, a process called crossover. Thereafter, the agents mutate at low probability. The successful agents are duplicated, while the less successful ones are eliminated. Then the networks are retrained again and the whole process is repeated. When a price is announced, trading of stocks is conducted until the consensus is reached. At this state, the overall wealth of the committees does not increase as a result of trading.

The example that is considered in this study is the trading of three stocks, namely: (1) the Dow Jones; (2) NASDAQ; and (3) S&P 500. The time-histories of the stocks are downloaded from the Internet and used to train agents. For a given set of price of these stocks, the committee of agents predicts the future prices of stocks. It should be noted that, on implementing this procedure, the total number of stocks available is kept constant. This figure indicates that sometimes the players with successful strategies do not necessarily dominate indefinitely. This is due to the fact that, strategies that are successful in one time frame are not necessarily successful at a later time.

When the scalability of the simulations was studied, it was found that the method proposed was scalable. However, it was observed that the computational time increased with the increase in the number of agents and players. A linear relationship existed between the average computational time taken to run the complete simulation and the number of players as well as the number of agents that form a player.

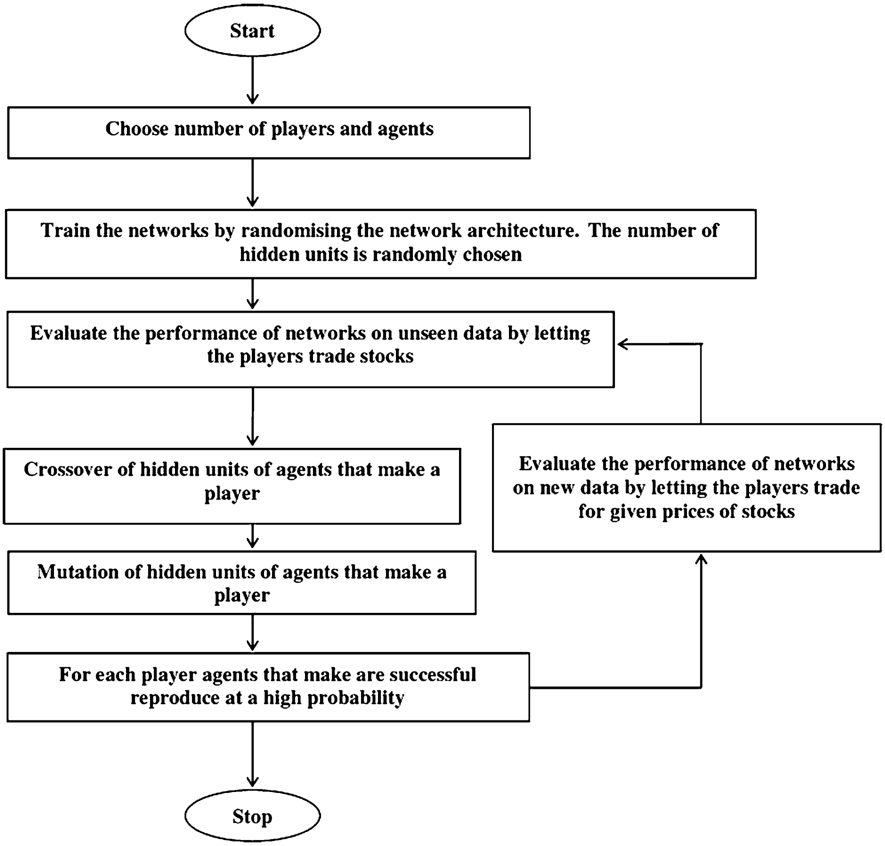

Fig. 11.4 Illustration of the simulation of the stock market

The complexity of the populations of agents that make players of the game was studied and defined as the measure of a degree of variation in a population of agents. Each species of agents form a dimension in space. Each dimension has a variation indicating the level of complexity of a population of that species. The results indicated that, as the system evolved, the number of hidden units for a given player steadily decreased and stabilized around 3. It was additionally observed that no player had the monopolistic advantage on the prediction of the stock market.

11.8 Conclusions

A simulation of the stock market was successfully implemented. It is established that the number of players and agents that form a player that partake in the trading game are directly proportional to the computational time taken to run the simulation. It is additionally found that, no player has the monopolistic advantage on the prediction of the stock market. The simulation also demonstrated that, as the time of the trading passes, the complexity of the players decrease. This is because of the fact that, as the time of trading elapsed, the players become more adapted to the time-varying nature of the problem, thus developing common features. Optimizing a committee of agents is observed to be a feasible method to modelling a player in the stock market.

References

Abdoos M, Mozayani N, Bazzan ALC (2011) Traffic light control in non-stationary environments based on multi agent Q-learning. In: Proceedings of the IEEE conference on intelligent transportation systems, Washington, DC, 2011, pp 1580–1585

Ahmad I, Abdullah A, Alghamdi A, Alnfajan K, Hussain M (2011) Intrusion detection using feature subset selection based on MLP. Sci Res Essays 6:6804–6810

Azarloo A, Farokhi F (2012) Automatic musical instrument recognition using K-NN and MLP neural networks. In: Proceedings of the 4th international conference on computational intelligence, communication systems and networks, Phuket, 2012, pp 289–294

Beed C, Beed C (1999) Intellectual progress and academic economics: rational choice and game theory. J Post Keynes Econ 22:163–185

Biancolini ME, Salvini P (2012) Radial basis functions for the image analysis of deformations. In: Proceedings of the international symposium CompIMAGE, Rome, Italy, pp 361–365

Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, Oxford

Breiman L (1996) Bagging predictors. Mach Learn 24:123–140

Buhmann MD, Ablowitz MJ (2003) Radial basis functions: theory and implementations. Cambridge University, Cambridge

Chen Y, Hu S, Liu H (2012) Prediction of the logistics demand for Heilongjiang province based on radial basis function algorithm. In: Proceedings of the 24th Chinese control and decision conference, Taiyuan, 2012, pp 2358–2361

Chitsaz M, Seng W (2013) Medical image segmentation using a multi-agent system approach. Int Arab J Inf Technol 10 (in press)

Christin N (2011) Network security games: combining game theory, behavioral economics and network measurements. Lect Notes Comput Sci 7037:4–6

Daya B, Akoum AH, Bahlak S (2012) Geometrical features for multiclass vehicle type recognition using MLP network. J Theor Appl Inf Technol 43:285–294

Del Pilar A-FI, Aguilar-Lasserre AA, Gonzalez-Huerta MA, Moras-Sanchez CG (2013) Investing in the sheep farming industry: a study case based on genetic algorithms. Lect Notes Electr Eng 130:337–346

Elammari M, Issa Z (2013) Using model driven architecture to develop multi-agent systems. Int Arab J Inf Technol 10 (in press)

El-Menshawy M, Bentahar J, El Kholy W, Dssouli R (2013) Verifying conformance of multi-agent commitment-based protocols. Expert Syst Appl 40:122–138

Foucard R, Essid S, Lagrange M, Richard G (2012) A regressive boosting approach to automatic audio tagging based on soft annotator fusion. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing, Kyoto, 2012, pp 73–76

Franklin S, Graesser A (1996) Is it an agent, or just a program?: A taxonomy for autonomous agents. In: Proceedings of the third international workshop on agent theories, architectures, and languages, pp 21–35

Gentilini C, Marzani A, Mazzotti M (2013) Nondestructive characterization of tie-rods by means of dynamic testing, added masses and genetic algorithms. J Sound Vib 332:76–101

Ghimire B, Rogan J, Galiano V, Panday P, Neeti N (2012) An evaluation of bagging, boosting, and random forests for land-cover classification in Cape Cod, Massachusetts, USA. GIScience Remote Sens 49:623–643

Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading

Graczyk M, Lasota T, Trawiński B, Trawiński K (2010) Comparison of bagging, boosting and stacking ensembles applied to real estate appraisal. Lect Notes Comput Sci 5991:340–350

Hanauske M, Kunz J, Bernius S, König W (2010) Doves and hawks in economics revisited: an evolutionary quantum game theory based analysis of financial crises. Physica A Stat Mech Appl 389:5084–5102

Hassoun MH (1995) Fundamentals of artificial neural networks. MIT Press, Cambridge, MA

Haykin S (1999) Neural networks. Prentice-Hall, Upper Saddle River

Hodgson GM, Huang K (2012) Evolutionary game theory and evolutionary economics: are they different species? J Evol Econ 22:345–366

Holland J (1975) Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor

Homayouni H, Hashemi S, Hamzeh A (2010) Instance-based ensemble learning algorithm with stacking framework. In: Proceedings of the 2nd international conference on software technology and engineering, San Juan, 2010, pp 164–169

Hui ECM, Bao H (2013) The logic behind conflicts in land acquisitions in contemporary China: a framework based upon game theory. Land Use Policy 30:373–380

Hurwitz E, Marwala T (2007) Learning to bluff. http://arxiv.org/ftp/arxiv/papers/0705/0705.0693.pdf. Last Accessed 7 Mar 2013

Ikeda Y, Tokinaga S (2004) Chaoticity and fractality analysis of an artificial stock market generated by the multi-agent systems based on the co-evolutionary genetic programming. IEICE Trans Fundam Electron Commun Comput Sci 87(9):2387–2394

Jasra A, Holmes CC (2011) Stochastic boosting algorithms. Stat Comput 21:335–347

Jha A, Chauhan R, Mehra M, Singh HR, Shankar R (2012) miR-BAG: bagging based identification of microRNA precursors. PLoS One 7:e45782

Kasabov N (1998) Introduction: hybrid intelligent adaptive systems. Int J Intell Syst 6:453–454

Khalilian M (2013) Towards smart advisor's framework based on multi agent systems and data mining methods. Lect Notes Electr Eng 156:73–78

Laffont J-J (1997) Game theory and empirical economics: the case of auction data. Eur Econ Rev 41:1–35

Leitenstorfer F, Tutz G (2011) Estimation of single-index models based on boosting techniques. Stat Model 11:203–217

Lienemann K, Plötz T, Fink GA (2009) Stacking for ensembles of local experts in metabonomic applications. Lect Notes Comput Sci 5519:498–508

Liu X, Cao H (2011) Price limit and the stability of stock market: an application based on multi-agent system. In: Proceedings of the 2nd international conference on artificial intelligence management science and electronic commerce, Deng Leng, 2011, pp 484–487

Liu W, Ji Z, He S, Zhu Z (2012a) Survival analysis of gene expression data using PSO based radial basis function networks. In: Proceedings of the IEEE congress on evolutionary computation, Brisbane, 2012, art. no. 6256144, pp 1–5

Liu Z, Yan J, Shi Y, Zhu K, Pu G (2012b) Multi-agent based experimental analysis on bidding mechanism in electricity auction markets. Int J Electr Power Energy Syst 43:696–702

Louzada F, Ara A (2012) Bagging k-dependence probabilistic networks: an alternative powerful fraud detection tool. Expert Syst Appl 39:11583–11592

Marwala T (2000) On damage identification using a committee of neural networks. J Eng Mech 126:43–50

Marwala T (2009) Computational intelligence for missing data imputation, estimation and management: knowledge optimization techniques. IGI Global Publications, New York

Marwala T (2010) Finite element model updating using computational intelligence techniques. Springer, London

Marwala T (2012) Condition monitoring using computational intelligence methods. Springer, London

Marwala T, Lagazio M (2011) Militarized conflict modeling using computational intelligence techniques. Springer, London

Marwala T, De Wilde P, Correia L, Mariano P, Ribeiro R, Abramov V, Szirbik N, Goossenaerts J (2001) Scalability and optimisation of a committee of agents using genetic algorithm. In: Proceedings of the 2001 international symposia on soft computing and intelligent systems for industry, Parsley, Scotland, arxiv 0705.1757

McCain RA (2009) Commitment and weakness of will in game theory and neoclassical economics. J Socio-Econ 38:549–556

McFadden DW, Tsai M, Kadry B, Souba WW (2012) Game theory: applications for surgeons and the operating room environment. Surgery (United States) 152:915–922

Mert A, Kiliç N, Akan A (2012) Evaluation of bagging ensemble method with time-domain feature extraction for diagnosing of arrhythmia beats. Neural Comput Appl:1–10. doi:10.1007/s00521-012-1232-7

Michalewicz Z (1996) Genetic algorithms + data structures = evolution programs. Springer, London

Møller AF (1993) A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw 6:525–533

Montoya A, Ovalle D (2013) Energy consumption by deploying a reactive multi-agent system inside wireless sensor networks. Lect Notes Electr Eng 152:925–934

Mutlu Ö, Polat O, Supciller AA (2013) An iterative genetic algorithm for the assembly line worker assignment and balancing problem of type-II. Comput Oper Res 40:418–426

Nirmal JH, Patnaik S, Zaveri MA (2013) Voice transformation using radial basis function. Lect Notes Electr Eng 150:345–351

Nock R, Piro P, Nielsen F, Bel Haj Ali W, Barlaud M (2012) Boosting k-NN for categorization of natural scenes. Int J Comput Vis 100:294–314

Papadimitriou CH (2001) Game theory and mathematical economics: a theoretical computer scientist's introduction. In: Proceedings of the annual symposium on foundations of computer science, Las Vegas, Nevada, pp 4–8

Parrado-Hernández E, Gómez-Verdejo V, Martínez-Ramón M, Shawe-Taylor J, Alonso P, Pujol J, Menchón JM, Cardoner N, Soriano-Mas C (2012) Voxel selection in MRI through bagging and conformal analysis: application to detection of obsessive compulsive disorder. In: Proceedings of the 2nd international workshop on pattern recognition in neuroimaging, London, 2012, pp 49–52

Pereira LFA, Pinheiro HNB, Silva JIS, Silva AG, Pina TML, Cavalcanti GDC, Ren TI, De Oliveira JPN (2012) A fingerprint spoof detection based on MLP and SVM. In: Proceedings of the international joint conference on neural networks, Brisbane, 2012, art. no. 6252582, pp 1–7

Perrone MP, Cooper LN (1993) When networks disagree: ensemble methods for hybrid neural networks. In: Mammone RJ (ed) Artificial neural networks for speech and vision. Chapman and Hall, London

Piro P, Barlaud M, Nock R, Nielsen F (2013) K-NN boosting prototype learning for object classification. Lect Notes Electr Eng 158:37–53

Ponzini R, Biancolini ME, Rizzo G, Morbiducci U (2012) Radial basis functions for the interpolation of hemodynamics flow pattern: a quantitative analysis. In: Proceedings of the international symposium CompIMAGE, Rome, Italy, pp 341–345

Ross D (2006) Evolutionary game theory and the normative theory of institutional design: binmore and behavioral economics. Polit Philos Econ 5:51–79

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

Roy K, Kamel MS (2012) Facial expression recognition using game theory. Lect Notes Comput Sci 7477:139–150

Russell SJ, Norvig P (2003) Artificial intelligence: a modern approach. Prentice Hall, Upper Saddle River

Shen G, Zhang Y (2013) Power consumption constrained task scheduling using enhanced genetic algorithms. Stud Comput Intell 432:139–159

Sill J, Takacs G, Mackey L, Lin D (2009) Feature-weighted linear stacking. http://arxiv.org/abs/0911.0460. Last Accessed 7 Mar 2013

Sokouti B, Haghipour S, Tabrizi AD (2012) A framework for diagnosing cervical cancer disease based on feedforward MLP neural network and ThinPrep histopathological cell image features. Neural Comput Appl:1–12. doi:10.1007/s00521-012-1220-y

Souahlia S, Bacha K, Chaari A (2012) MLP neural network-based decision for power transformers fault diagnosis using an improved combination of Rogers and Doernenburg ratios DGA. Int J Electr Power Energy Syst 43:1346–1353

Stroeve SH, Blom HAP, Bakker GJ (2013) Contrasting safety assessments of a runway incursion scenario: event sequence analysis versus multi-agent dynamic risk modelling. Reliab Eng Syst Saf 109:133–149

Syam B, Rao YS (2013) An effective similarity measure via genetic algorithm for content based image retrieval with extensive features. Int Arab J Inf Technol 10 (in press)

Syarif I, Zaluska E, Prugel-Bennett A, Wills G (2012) Application of bagging, boosting and stacking to intrusion detection. Lect Notes Comput Sci 7376:593–602

Tajbakhsh N, Wu H, Xue W, Gotway MB, Liang J (2012) A novel online boosting algorithm for automatic anatomy detection. Mach Vis Appl: (26 October 2012), pp 1–12. doi:10.1007/s00138-012-0455-z

Teweldemedhin E, Marwala T, Mueller C (2004) Agent-based modelling: a case study in HIV epidemic. In: Proceedings of the fourth international conference on hybrid intelligent systems, Tokyo, Japan, pp 154–159

Tirea M, Tandau I, Negru V (2012) Stock market multi-agent recommendation system based on the Elliott Wave Principle. Lect Notes Comput Sci 7465:332–346

van den Brink R, van der Laan G, Vasil'ev V (2008) Extreme points of two digraph polytopes: description and applications in economics and game theory. J Math Econ 44:1114–1125

Villena MG, Villena MJ (2004) Evolutionary game theory and Thorstein Veblen's evolutionary economics: is EGT Veblenian? J Econ Issues 38:585–610

Wei X, Qu H, Ma E (2012) Decisive mechanism of organizational citizenship behavior in the hotel industry – an application of economic game theory. Int J Hosp Manag 31:1244–1253

Will A, Bustos J, Bocco M, Gotay J, Lamelas C (2013) On the use of niching genetic algorithms for variable selection in solar radiation estimation. Renew Energy 50:168–176

Wolpert DH (1992) Stacked generalization. Neural Netw 5:241–259

Yoshikazu I, Shozo T (2007) Multi-fractality analysis of time series in artificial stock market generated by multi-agent systems based on the genetic programming and its applications. IEICE Trans Fundam Electron Commun Comput Sci 90(10):2212–2222

Yun Y, Gu IYH (2012) Multi-view face pose classification by boosting with weak hypothesis fusion using visual and infrared images. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing, Kyoto, 2012, pp 1949–1952