A Neuro-fuzzy-evolutionary Classifier of Low-risk Investments

Antoaneta Serguieva, Tatiana Kalganova
Department of Electronic and Computer Engineering
Brunel University, Uxbridge UB8 3PH, UK
Antoaneta.Serguieva@brunel.ac.uk,Tatiana.Kalganova@brunel.ac.uk

Abstract – The proposed paper demonstrates that a hybrid fuzzy neural network can serve as a classifier of low risk investment projects. The training algorithm for the regular part of the network is based on bidirectional incremental evolution proving more efficient than direct evolution. The approach is applied to empirical data on UK companies traded on the LSE.

1. INTRODUCTION

Standard investment appraisal techniques have been continuously revised. The criteria have been altered because of the effect of capital and labour market rationing [19] or reoptimised due to investment irreversibility [9] and the impact of a project on the investor's total risk [23]. It has been realised that the removal of any of the perfect market assumptions destroys the foundation and reduces the effectiveness of the methods. Alternatively, a fuzzy criterion does not attempt to cope with a specific drawback of standard techniques but permits into the calculations as much uncertainty as the market could possibly suffer. The outcome is an effective method under restricted information, uncertain data and market imperfections. A fuzzy criterion and investment rating technique are first introduced in [2], then considered in a broader framework of accumulation and discount models in [7], and recently modified with an alternative fuzzification of the project duration in [15]. While those studies are theoretical in nature, the empirical results are a major emphasis in [13,22], where stock projects are evaluated and UK companies traded on the London Stock Exchange are considered. Simultaneously, the analysis of the empirical solutions to the fuzzy criterion facilitates there the induction of three general conclusions. An investment risk measure, an estimate of the project robustness towards market uncertainty modelled with the fuzzified data, and an alternative ranking technique based on the two measures.

When compared with previous studies, the proposed paper demonstrates the following advantages. First, nine representative projects are chosen from the database employed in [22], thus consistently emphasising the empirical results. Second, the projects are rated according to a modification of the risk measure suggested there, extending further the developed method. Third, a fuzzy valued criterion is formulated and a regular fuzzy neural network (RFNN), trained with a genetic algorithm (GA), approximates its solution. Hence, the benefits of various soft techniques are blended to achieve a synergy in handling the investment appraisal problem. In comparison, [2,7,15] only study fuzzy criteria. Forth, the network is hybridised to discriminate between low-risk and high-risk projects. The threshold is agent dependent, communicating the acceptable levels of risk. The variety of market agents work within diverse risk ranges. In the extreme, the behaviour of an investment fund differs from the behaviour of an individual investor. Consequently, an agent dependent threshold will benefit the decision-maker. Fifth, an efficient training algorithm is suggested for the RFNN part of the network. GAs are a promising tool in training RFNNs and recent studies successfully apply direct evolution (DE) to optimise the fuzzy weights in small networks resembling particular types of univariable fuzzy functions.[5] The RFNN here approximates multivariable criterion and the size of the network depends on the investment horizon. The complexity of the problem requires a corresponding evolutionary strategy (ES) and the implementation of bidirectional incremental evolution (BIE) is suggested. Incremental strategy has been already applied to evolve neural networks controlling a robot's motion.[11] BIE incorporates divide-and-conquer evolution and incremental evolution. It gradually divides the complex task into simpler subtasks, evolves them separately and consecutively merges them incrementally to optimise the obtained solution. The technique allows overcoming the stalling effect in direct evolution and has already proved more efficient in evolving logic functions.[14]

The proposed method is developed for the following reasons. First, there have been suggested fuzzy techniques for investment appraisal and based on them ranking procedures [2,22], but no investment classifier is yet considered. Therefore, a decision would only be taken after applying the fuzzy criterion to all available projects, then rating them accordingly, and finally choosing the acceptable opportunities. The introduction of a classifying system will significantly simplify the process, especially for a large number of continuously updated projects and regularly taken investment decisions. Once trained, the network will be an effortless instrument in the hands of the decision-maker, whenever the information available is subject to change. Second, standard neural networks have been already successfully applied to classify takeover targets on the basis of company accounting data and financial ratios [10]. Mergers and takeovers are a specific type of investment activity.

Figure 1 describes the interrelations between standard and soft computing techniques as well as between theoretical and empirical studies. All these approaches are involved into the process of formulating the investment classifier.

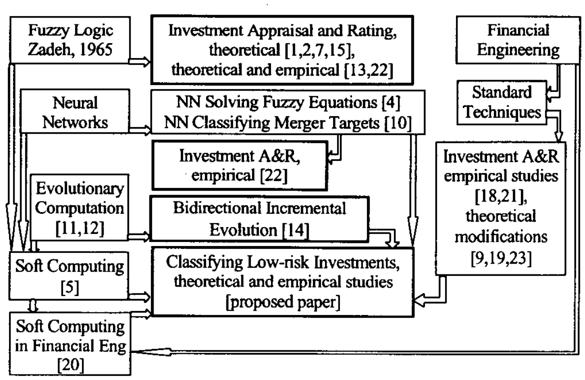

Figure 1: Technique interrelations in formulating the investment classifier
- standard and soft computing
- theoretical and empirical approaches
- bold line – developing the proposed method

II. FUZZY VALUED CRITERION

The first stage of the proposed method involves formulating a fuzzy valued criterion (FVC) for stock projects under time-varying discount rate. Stock prices are too volatile to be rational forecasts of future dividends discounted at a constant rate and empirical tests have convinced many financial economists that stock returns are time-varying rather than constant [8]. The assumption of time-varying returns transforms the price-dividend relation into nonlinear. Let us consider its loglinear approximation

$$r_{t_0+1} \approx \delta + p_{t_0+1} + (1-\lambda)dy_{t_0+1} - p_{t_0} \ , \qquad (1)$$

where p_{t_0+1}, r_{t_0+1} and dy_{t_0+1} stand for the log share price, log return and log dividend yield, correspondingly. Then, if equation (1) is solved forward for the log stock price, the following estimation is produced in period $t=t_0$,

$$\begin{aligned} &\hat{p}_{t_0} = \sum_{t=t_0+1}^{t_0+T} \lambda^{t-t_0-1}\left[(1-\lambda)(dy_t + p_t) + \delta - r_t\right] + \lambda^{t_0+T} p_{t_0+T} \ , \\ &1 < e^{p_t} - quotes\ in\ pence, 1 < e^{r_t} < 2, 0 < e^{dy_t} < 1, \\ &p_t > 0, 0 < r_t < ln(2), dy_t < 0, \end{aligned} \qquad (2)$$

where the parameters of linearisation, δ and λ, are evaluated with continuous functions of $(dy_1,...,dy_T)$,

$$\begin{aligned} &\lambda \equiv (1 + e^{\frac{1}{T}\sum_{t=1}^{T} dy_t})^{-1} = f_1(dy_1,...,dy_T), 0 < \lambda < 1, \\ &\delta \equiv (1-\lambda)ln(1-\lambda) - \lambda ln(\lambda) = f_2(dy_1,...,dy_T), 0 < \delta < 1. \end{aligned} \qquad (3)$$

If the investment horizon is T, then a project is profitable at $t_0=0$ when the estimated share price exceeds the market share price

$$\hat{p}_0 > p_0 \ . \qquad (4)$$

Based on (4), the FVC is formulated following a procedure in four steps. First, for each project, parameters δ and λ are obtained from (3) and considered crisp. Second, market uncertainty is introduced applying initially the calibration technique from [22] and producing triangular shaped membership functions for the fuzzified log data, $\widetilde{P}_t$, $\widetilde{R}_t$ and $D\widetilde{Y}_t$. Then, positive triangular shaped fuzzy coefficients $\widetilde{A}_t, \widetilde{B}_t, \widetilde{C}_t, 1 \le t \le T$, are obtained from

$$\begin{aligned} &\widetilde{A}_t = \frac{1}{N}\sum_{i=1}^{N}\frac{\widetilde{P}_{ti}}{p_{ti}}, \ \widetilde{B}_t = \frac{1}{N}\sum_{i=1}^{N}\frac{\widetilde{R}_{ti}}{r_{ti}}, \ \widetilde{C}_t = \frac{1}{N}\sum_{i=1}^{N}\frac{D\widetilde{Y}_{ti}}{dy_{ti}}, \\ &1 \le t \le T \quad , \quad N - number\ of\ projects \quad , \end{aligned} \qquad (5a)$$

and the initial calibration is slightly modified, assuming

$$\widetilde{P}_{ti} = \widetilde{A}_t p_{ti}, \widetilde{R}_{ti} = \widetilde{B}_t r_{ti}, D\widetilde{Y}_{ti} = \widetilde{C}_t dy_{ti}, 1 \le t \le T, 1 \le i \le N \ , \qquad (5b)$$

Third, the fuzzy log share price $\widetilde{P}$ at $t_0=0$ is presented as

$$\begin{aligned} &\widetilde{P}_0 = \sum_{t=1}^{T}\lambda^{t-1}\left[(1-\lambda)(\widetilde{C}_t dy_t + \widetilde{A}_t p_t) + \delta - \widetilde{B}_t r_t\right] + \lambda^T \widetilde{A}_T p_T = \\ &= \left[\widetilde{A}_1 p_1 (1-\lambda)\right] + ... + \left[\widetilde{A}_T p_T \lambda^{T-1}\right] - \left[\widetilde{B}_1 r_1\right] - ... - \left[\widetilde{B}_T r_T \lambda^{T-1}\right] + \\ &\left[\widetilde{C}_1 dy_1 (1-\lambda)\right] + ... + \left[\widetilde{C}_T dy_T \lambda^{T-1}(1-\lambda)\right] + \left[\sum_{t=1}^{T}\lambda^{t-1}\delta\right] = \widetilde{A}_1 g_{A1} + \\ &... + \widetilde{A}_T g_{AT} - \widetilde{B}_1 g_{B1} - ... - \widetilde{B}_T g_{BT} + ... + \widetilde{C}_T g_{CT} + g, \quad 0 < \lambda, \delta < 1, \end{aligned} \qquad (6a)$$

where
$$\begin{aligned} &g_{At} = g_{At}(x), g_{Bt} = g_{Bt}(x), g_{Ct} = g_{Ct}(x), g = g(x) \\ &x = (p_1,...,p_T, r_1,...,r_T, dy_1,...,dy_T), 1 \le t \le T, \\ &p_t > 0, 0 < r_t < ln(2), dy_t < 0, \end{aligned} \qquad (6b)$$

are continuous functions defined on the market data employed to evaluate a project. Thus, the fuzzy log share price from [22] is transformed into a continuous multivariable fuzzy valued (CMFV) function. The modification provides that the RFNN introduced in the next section is capable of approximating $\widetilde{P}_0$ to any degree of accuracy.

Forth, applying the extension principle, the triangular shaped membership function of the solution is described with

$$\begin{aligned} &\mu(y_P / \widetilde{P}_0) = sup\{\alpha / y_P \in \Omega_{P_0}(\alpha)\}, \\ &\Omega_{P_0}(\alpha) = \{a_1 g_{A1}(x) + ... + c_T g_{CT}(x) + g(x) / \\ &/ a_1 \in \widetilde{A}_1(\alpha), ..., c_T \in \widetilde{C}_T(\alpha), x = (p_1,...,dy_T)\}, \end{aligned} \qquad (6c)$$

and for the specific formulation of $\widetilde{P}_0$, the α*-cut* $\Omega_{P_0}(\alpha)$ is equivalent to the interval arithmetic solution

$$\begin{aligned} &\Omega_{P_0}(a) = \left[\underline{P_0(a)}, \overline{P_0(a)}\right] \ , \\ &\underline{P_0(a)} = \underline{A_1(a)} g_{A1}(x) + ... + \underline{A_T(a)} g_{AT}(x) - \overline{B_1(a)} g_{B1}(x) + ... + \\ &\qquad - \overline{B_T(a)} g_{BT}(x) + \underline{C_1(a)} g_{C1}(x) + ... + \overline{C_T(a)} g_{CT}(x) + g(x), \\ &\overline{P_0(a)} = \overline{A_1(a)} g_{A1}(x) + ... + \overline{A_T(a)} g_{AT}(x) - \underline{B_1(a)} g_{B1}(x) + ... + \\ &\qquad - \underline{B_T(a)} g_{BT}(x) + \underline{C_1(a)} g_{C1}(x) + ... + \underline{C_T(a)} g_{CT}(x) + g(x), \\ &A_t(a) > 0, B_t(a) > 0, C_t(a) > 0, g_{At}(x) > 0, g_{Bt}(x) > 0, \\ &g_{Ct}(x) < 0, g(x) > 0, x = (p_1,...,dy_T), 1 \le t \le T. \end{aligned} \qquad (6d)$$

Solution (6d) identifies the set of estimated log share price values corresponding to all future log share prices, dividend yields and discount rates possible at some level of uncertainty, u. This set is situated at the same level u. Therefore, there is a critical level of uncertainty, $u_{critical}$, embodied into the market data we use to evaluate a project and this level delimits the project's investment risk. The risk measure, $\alpha_{critical}$, below

$$\begin{aligned} &\alpha_{critical} = \mu(p_0 / \widetilde{P}_0) = sup\{\alpha / y_P = p_0\}, \\ &1 - u_{critical} = \alpha_{critical} \in [0,1] \end{aligned} \qquad (7)$$

is suggested for the following reasons. The lower the critical level of uncertainty at which there is a chance for the project being unprofitable, the higher the investment risk. Furthermore, $\alpha_{critical}$ is the membership level of the fuzzy log share price, below and at which the solution includes values smaller or equal to the initial log market price, and above which the project is profitable. Thus, the criterion is finally described as

$$\alpha_{critical} \le h_{agent} \ , \qquad (8)$$

where the threshold h is agent dependent and indicates the acceptable risk values.

III. FUZZY NEURAL NETWORK

The second stage of the method consists of building a regular fuzzy neural network, to approximate the CMFV function $\widetilde{P}_0$, and subsequently including two more layers to discriminate between risky projects. The approximating capabilities of RFNNs have been intensively studied in the last few years. Let $F_0(R)$ is the set of all fuzzy numbers on the real number set R. It is demonstrated in [5,6] that RFNNs are not able to represent to any degree of accuracy continuous fuzzy functions $F{:}F_0(R) \rightarrow F_0(R)$. On the other hand, [16] proves that they are universal approximator for continuous fuzzy valued functions described with $F{:}R \rightarrow F_0(R)$. It is also suggested there that similar results apply to multivariable functions $F{:}R^k \rightarrow F_0(R)$.

Based on the theorems proved in [16], we make the following conclusions.

Remark 1: Let $f: U_1 \times \ldots \times U_k \rightarrow R$ is a multivariable continuous crisp function on the compact sets $U_i \subset R$, $1 \leq i \leq k$. If $\tilde{A} \in F_0(R)$ is a fuzzy number and $F: U_1 \times \ldots \times U_k \rightarrow F_0(R)$ is a multivariable fuzzy valued function, where

$$F(x_1,\ldots,x_k) = (\tilde{A}f)(x_1,\ldots,x_k) = \tilde{A}f(x_1,\ldots,x_k), x_i \in U_i, 1 \leq i \leq k, \quad (9)$$

then $F(x_1,\ldots,x_k)$ is continuous on $U_1 \times \ldots \times U_k$.

Remark 2: Let

$$\Im\left\{ RFNN / RFNN(x_1,\ldots,x_k) = \sum_{i=1}^{q} \tilde{V}_i * \left(\sum_{j=1}^{m} \tilde{E}_{ij} * \sigma\left(\sum_{\ell=1}^{k} \tilde{W}_{j\ell} * x_\ell + \tilde{\Theta}_j \right)\right) \right.$$
$$\left. q,m \in N,\ \tilde{V}_i, \tilde{E}_{ij}, \tilde{W}_{j\ell}, \tilde{\Theta}_j \in F_0(R) \right\} \quad (10a)$$

describes the class of four layer feedforward RFNNs with sigmoid transfer functions and shift terms in the first hidden layer, and identity transfer functions with no shift terms in the second hidden layer. Restricting $\tilde{E}_{ij}, \tilde{W}_{j\ell}, \tilde{\Theta}_j \in F_0(R)$ to be $e_{ij}, w_{j\ell}, \theta_j \in R$, the subset $\Im_0$ of $\Im$ is obtained.

$$\Im_0\left\{ RFNN / RFNN(x_1,\ldots,x_k) = \sum_{i=1}^{q} \tilde{V}_i * \left(\sum_{j=1}^{m} e_{ij} * \sigma\left(\sum_{\ell=1}^{k} w_{j\ell} * x_\ell + \theta_j \right)\right) \right.$$
$$\left. q,m \in N,\ \tilde{V}_i \in F_0(R),\ e_{ij}, w_{j\ell}, \theta_j \in R \right\} \quad (10b)$$

Then $\Im$ and $\Im_0$ are universal approximators for the CMFV function $F: U_1 \times \ldots \times U_k \rightarrow F_0(R)$ from (9).

Remark 3: Let $F_1,\ldots,F_n$ are CMFV functions as the one in (9), then $\Im$ and $\Im_0$ from (10) are universal approximators for

$$F = \sum_{i=1}^{n} (\pm F_i). \quad (11)$$

Remark 4: $\tilde{P}_0$ from (6a) is a CMFV function of type (11). Consequently, there exists a four layer feedforward $RFNN \in \Im_0$ approximating $\tilde{P}_0$ to any degree of accuracy.

Figure 2 introduces the network structure classifying investment projects according to criterion (8). The dashed box outlines the RFNN part. Its input layer is fed with log stock prices, log returns and log dividend yields, while the output layer produces

$$\tilde{P}_{RFNN}(p_1,\ldots,p_T,r_1,\ldots,r_T,dy_1,\ldots,dy_T,) =$$
$$= \sum_{i=1}^{q} \tilde{V}_i\left(\sum_{j=1}^{m} e_{ij}\sigma\left(\sum_{\ell=1}^{T} \left(w_{j\ell}p_\ell + u_{j\ell}r_\ell + z_{j\ell}dy_\ell\right) + \theta_j \right)\right) \quad (12a)$$
$$q,m \in N,\ \tilde{V}_i \in F_0(R),\ e_{ij}, \theta_j, w_{j\ell}, u_{j\ell}, z_{j\ell} \in R\ ..$$

The extension principle and the interval arithmetic evaluation of such RFNN are equivalent and the *α-cuts* are computed from

$$\Omega_{PRFNN}(\alpha) = \left[\underline{P_{RFNN}}(\alpha), \overline{P_{RFNN}(\alpha)}\right] \ ,$$
$$\underline{P_{RFNN}}(\alpha) = \sum_{i=1}^{q} min(\underline{V_i}(\alpha) * (\sum_{j=1}^{m} e_{ij}\sigma_j), \overline{V_i(\alpha)} * (\sum_{j=1}^{m} e_{ij}\sigma_j)),$$
$$\overline{P_{RFNN}(\alpha)} = \sum_{i=1}^{q} max(\underline{V_i}(\alpha) * (\sum_{j=1}^{m} e_{ij}\sigma_j), \overline{V_i(\alpha)} * (\sum_{j=1}^{m} e_{ij}\sigma_j)), \quad (12b)$$
$$\sigma_j = \sigma(\sum_{\ell=1}^{T} \left(w_{j\ell}p_\ell + u_{j\ell}r_\ell + z_{j\ell}dy_\ell\right) + \theta_j) > 0,$$

In the additional part of the network, the transfer function φ is described in (7), while ψ is a hard limit transfer function with threshold h_{agent} from (8).

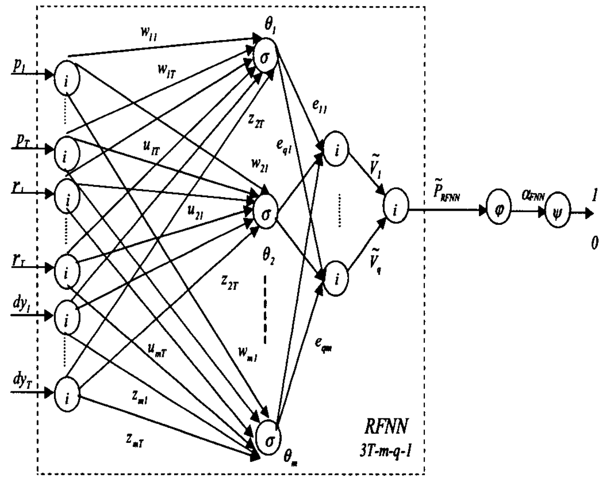

Figure 2: Neural network structure classifying low risk projects

Dashed box - outlines the RFNN part:
Crisp inputs - market data, p_ℓ, r_ℓ, dy_ℓ
Weights – real numbers $e_{ij}, w_{j\ell}, u_{j\ell}, z_{j\ell}$, triangular fuzzy numbers $\tilde{V}_i$
1st hidden layer-sigmoid transfer functions σ with shifts θ_j
2nd hidden layer - identity transfer functions *i* with no shift terms
Triangular shaped fuzzy output – $\tilde{P}_{RFNN}$ approximating $\tilde{P}_0$
Attached hybrid part:
φ - transfer function described in (7)
ψ - hard limit transfer function with threshold h_{agent} from (8)
output=1→low risk investments, *output=0*→high risk investments

IV. BIDIRECTIONAL INCREMENTAL EVOLUTION

At the next stage of the developed technique, the RFNN part of the fuzzy network in Figure 2 is trained to approximate the $\tilde{P}_0$ function, using a genetic algorithm and following a bidirectional incremental evolutionary strategy. The GA is specified with its initialisation, selection and recombination operators. The initialisation step includes chromosome encoding and generating the first population. Let a triangular fuzzy weight be presented by three real numbers corresponding to its support and vertex. Then, the RFNN is coded into the chromosome χ,

$$\chi = (\chi_1, \chi_2, \ldots, \chi_{3*m*T+m+q*m+3*q}) =$$
$$= (w_{11}, w_{12}, \ldots, w_{mT}, \ldots, u_{mT}, \ldots, z_{mT}, \ldots,$$
$$\theta_m, \ldots, e_{qm}, v_1^a, v_1^b, v_1^c, \ldots, v_q^a, v_q^b, v_q^c), \quad (13a)$$

$$\tilde{V}_1 = (v_1^a / v_1^b / v_1^c), \ldots, \tilde{V}_q = (v_q^a / v_q^b / v_q^c),$$
$$v_1^a < v_1^b < v_1^c, \ldots, v_q^a < v_q^b < v_q^c. \quad (13b)$$

The initial population $X = [\chi^{(1)} \ \ \chi^{(2)} \ \ldots \ \chi^{(s)}]$ of s individuals from (13a) is generated simultaneously as a $(3mT+m+qm+3q) \times s$ matrix whose elements are realisations of a random variable with standard normal distribution $N(0,1)$. A block representation of X, $X = [X^{(1)}_{(3mT+m+qm) \times s}\ X^{(2)}_{3 \times s} \ldots X^{(1+q)}_{3 \times s}]'$, helps to concurrently sort its elements according to restrictions (13b). Next, a breeding subpopulation X_{SUB} is selected, which consists of s_1 best fitted chromosomes, where $s_1<s$. The selection is based on the objective function Q,

$$Q=\min_{\chi^{(i)}}\left(\max_{\alpha}\left(\max\left(\left|\underline{P_0(\alpha)}-\underline{P_{RFNN}(\alpha)}\right|,\left|\overline{P_0(\alpha)}-\overline{P_{RFNN}(\alpha)}\right|\right)\right)\right),$$
$$\chi^{(i)}=\chi^{(1)},\dots,\chi^{(s)},\alpha=0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1, \quad (14)$$

minimising the error of the RFNN. Finally, the recombination process builds the new generation X_{NEW}. A multipoint crossover operator is applied on X_{SUB} to produce a temporary full-size population X_{TEMP}. The number η of crossover points is randomly chosen every generation. A triplet representing a fuzzy number is considered as a single gene and the position of crossover points is restricted to the following set.

$$index_i \in I=(1,2,3,4,5,6,\dots,3mT+m+qm,3mT+m+qm+1,\\ 3mT+m+qm+4,3mT+m+qm+7,\dots,\\ 3mT+m+qm+3q-5,3mT+m+qm+3q-2),\\ 1\le i\le\eta. \quad (15a)$$

Two randomly chosen chromosomes from X_{SUB} are combined to obtain two offspring. Only one of them is included in X_{TEMP}.

$$\chi^{(j)}_{TEMP}=\left[\chi^{(\ell)}_{SUB,1}\cdots\chi^{(\ell)}_{SUB,index1-1}\,\chi^{(k)}_{SUB,index1}\cdots\chi^{(k)}_{SUB,index2-1}\,\chi^{(\ell)}_{SUB,index2}\cdots\right.\\ \left.\cdots\chi^{(\ell)}_{SUB,index3-1}\,\chi^{(k)}_{SUB,index3}\cdots\chi^{(\ell)}_{SUB,index\eta-1}\,\chi^{(k)}_{SUB,index\eta}\cdots\chi^{(k)}_{SUB,3mT+m+qm+3q}\right]\\ 1\le j\le s,1\le\ell\le s_1,1\le k\le s_1,index_i\in I,1\le i\le\eta. \quad (15b)$$

Crossover points $index_i$ are randomly chosen every generation for each $\chi^{(j)}_{TEMP}$. The mutation operator concludes the recombination step, transforming the temporary population into the new generation of RFNN representations. It involves a constant rate τ, where $0<\tau<0.2$. Thus the number of mutated genes is constant, but their position is randomly chosen every generation, $index_j \in I$, $1 \leq j \leq \tau*(3mT+m+qm+q)$. Mutated genes are generated as realisations of a random variable with standard normal distribution N(0,1). All the mutated triplet genes are concurrently sorted according to restrictions (13b).

The GA described above is applied within a bidirectional incremental evolutionary strategy. BIE is suggested on the following grounds. A complex task is difficult to evolve, as it is not possible to directly discover a general solution.[7] The stalling effect in direct evolution may be defeated by implementing incremental strategies, where neural networks learn complex behaviour while starting with simple functioning and gradually increasing the complexity of the task.[11,12] The problem is that the relevant subset of simple tasks as well as their sequence is not uniformly defined, and so is an incremental strategy. Consequently, it is appropriate to identify the efficient subset of tasks and their efficient sequence.[14] BIE applied here deals with the question of efficiency by first identifying the subtasks and their sequence, then evolving them separately, and finally merging the tasks gradually while following the efficient sequence. When the investment paradigm is considered, the domain of subtasks is described as follows. If T is the investment horizon, N_1 is the number of periods in which projects are available, $\Delta t_1=[t_{01},t_{01}+T],\dots,\Delta t_{N_1}=[t_{0N_1},t_{0N_1}+T]$, and N_2 is the number of companies, then there are $N=N_1*N_2$ single-company single-period projects constituting the first level of subtasks with lowest complexity. The next levels consist of subsets of projects involving increasing number of single projects. They may concern the same company over several periods or the same type of companies over one or a number of periods, or may include different types of companies. According to its investment risk, a single project can be profitable and not risky, when $\alpha_{critical}=0$, risky, for $0<\alpha_{critical}<1$, or too risky and unprofitable, $\alpha_{critical}=1$. Consequently, there exist five types of companies. Those with investment risk 0 over all the periods are quite rare, having in mind that the market is modelled as highly uncertain. The other types include companies with continuously improving levels of risk, and companies with unstable risk levels $0\leq\alpha_{critical}\leq1$ without a particular direction. Finally, those with constantly worsening investment risk, and the ones with $\alpha_{critical}=1$ over all periods. Providing the training set includes companies of all types and single projects covering the three important modes of $\alpha_{critical}$, then the evolved RFNN will sufficiently predict the $\widetilde{P}_0$ of new investment opportunities.

The evolutionary strategy is described as follows.

1: Define the training set of projects $n\subset N$ and obtain their $\widetilde{P}_0$ from (6d).

2: Choose the probing step of generations N_{gen} and the parameters $Q^{(k)}_{DEC1}, Q^{(k)}_{DEC2}, Q^{(k)}_{DEC3}, Q^{(k)}_{DEC4}, Q^{(k)}_{DEC5}$, $Q_{DECEND}, Q_{INC}, Q_{INCEND}$. Here k=1,2,… is the corresponding level of decomposition and partitions are to be identified.

3: Initialise $k=1$, $n^{(1)}=n$, and generate a random initial population $\underset{s}{IP}$ of size s.

4: Evolve the RFNN for N_{gen} generations, using the complete training set $n^{(k)}$. During evolution, evaluate $\widetilde{P}_{RFNN}$ from (12b) and apply objective function Q in (14).

5: Keep the result of the evolution - the breeding subpopulation $\underset{s_1}{X}^{(k)}$ - where s_1 is the number of breeding chromosomes, $s_1<s$.

6: If the average value of Q over the breeding subpopulation $\underset{s_1}{X}^{(k)}$ in the complete training set is more then $Q^{(k)}_{DEC1}$, $\frac{1}{s_1}\sum_{i=1}^{s_1}\max_{n^{(k)}}(Q(\chi_i))>Q^{(k)}_{DEC1}$, then go to step 3.

7: If there do not exist single projects satisfying the condition

$$\frac{1}{s_1}\sum_{i=1}^{s_1}Q(\chi_i)<Q^{(k)}_{DEC2}, \quad (16a)$$

then generate an initial population $\underset{s}{IP}(\underset{s_1}{X}^{(k)})$ by recombination of $\underset{s_1}{X}^{(k)}$. Go to 4.

8: Group the single projects satisfying (16a) into $n_1^{(k)}$ subsets of projects, $n_1^{(k)}=\{n_{11}^{(k)},\dots,n_{1j}^{(k)},\dots,n_{1J_k}^{(k)}\}$, where each $n_{1j}^{(k)}$ is of maximum size, subject to

$$\frac{1}{s_1}\sum_{i=1}^{s_1}\max_{n_{1j}^{(k)}}(Q(\chi_i))<Q^{(k)}_{DEC2}. \quad (16b)$$

9: Partition the training set $n^{(k)}$ into $n^{(k)}=\{n_{11}^{(k)},\dots,n_{1j}^{(k)},\dots,n_{1J_k}^{(k)},n_2^{(k)}\}$, where J_k is the number of subsets satisfying (16b) and is specific for the level of decomposition k. The subset of single projects not satisfying condition (16b) is $n_2^{(k)}=n^{(k)}/n_1^{(k)}$. In the extreme, it can be $n_2^{(k)}=\{\varnothing\}$ or $n_2^{(k)}=n^{(k)}$.

10:Generate different initial populations $\underset{s}{IP}_{11}(\underset{s_1}{X}^{(k)})$, …, $\underset{s}{IP}_{1J_k}(\underset{s_1}{X}^{(k)})$ and $\underset{s}{IP}_2(\underset{s_1}{X}^{(k)})$ by recombination of the same breeding subpopulation $\underset{s_1}{X}^{(k)}$.

11:Evolve a separate RFNN in N_{gen} generations for each training subset of $n^{(k)}$. Keep the evolution results – the breeding subpopulations $\underset{s_1}{X}^{(k)}_{11},\dots,\underset{s_1}{X}^{(k)}_{1J_k}$ and $\underset{s_1}{X}^{(k)}_2$.

12:If $\frac{1}{s_1}\sum_{i=1}^{s_1}\max_{n_{1j}^{(k)}}(Q(\chi_i))<Q^{(k)}_{DEC3}$, then generate an initial population $\underset{s}{IP}_{1j}(\underset{s_1}{X}^{(k)}_{1j})$. Else, go to step 13.

13:Evolve an RFNN, using the training set $n_{1j}^{(k)}$, until the average value of Q over the first half of the breeding subpopulation $\underset{s_1/2}{X}^{(k)}$ is les than Q_{DECEND}, $\frac{2}{s_1}\sum_{i=1}^{s_1/2}\max_{n_{1j}^{(k)}}(Q(\chi_i))<Q_{DECEND}$. Keep the result of the evolution – half of the breeding subpopulation $\underset{s_1/2}{X}^{(k)}_{1jEND}$.

14:If $n_{1j}^{(k)}$ consists of a single project and $Q_{DEC3}^{(k)} < \frac{1}{s_1}\sum_{i=1}^{s_1} Q(\chi_i) < Q_{DEC2}^{(k)}$, then generate a new initial population $\underset{s}{IP}_{1j}(\underset{s_1}{X}^{(k)})$. Else, go to step 15.

15:Evolve an RFNN for N_{gen} generations, using the training set $n_{1j}^{(k)}$. Keep the result of the evolution– the breeding subpopulation $\underset{s_1}{X}_{1j}^{(k)}$. Go to step 11.

16:If $Q_{DEC3}^{(k)} < \frac{1}{s_1}\sum_{i=1}^{s_1} \max_{n_{1j}^{(k)}}(Q(\chi_i)) < Q_{DEC4}^{(k)}$ and $n_{1j}^{(k)}$ consists of a subset of projects, then generate an initial population $\underset{s}{IP}_{1j}(\underset{s_1}{X}_{1j}^{(k)})$. Consider the subset $n_{1j}^{(k)}$ as a complete training set $n^{(k+1)} = n_{1j}^{(k)}$ and increase $k=k+1$. Go to step 4.

17:If $Q_{DEC4}^{(k)} < \frac{1}{s_1}\sum_{i=1}^{s_1} \max_{n_{1j}^{(k)}}(Q(\chi_i)) < Q_{DEC2}^{(k)}$ and n_{1j} consists of a subset of projects, then generate a new initial population $\underset{s}{IP}_{1J}(\underset{s_1}{X}^{(k)})$. Consider the subset $n_{1j}^{(k)}$ as a complete training set and increase decomposition level $k=k+1$. Go to step 4.

18:If $\frac{1}{s_1}\sum_{i=1}^{s_1} \max_{n_2^{(k)}}(Q(\chi_i)) < Q_{DEC5}^{(k)}$, then generate an initial population $\underset{s}{IP}_2(\underset{s_1}{X}_2^{(k)})$, else generate a new initial population $\underset{s}{IP}_2(\underset{s_1}{X}^{(k)})$. Consider $n_2^{(k)}$ as a complete training set and increment $k=k+1$. Go to step 4.

19:If $n_2^{(k)}$ consists of a single project, then evolve an RFNN until $\frac{2}{s_1}\sum_{i=1}^{s_1/2} \max_{n_2^{(k)}}(Q(\chi_i)) < Q_{DECEND}$ and keep the result $\underset{s_1/2}{X}_{2END}^{(k)}$. Else consider $n_2^{(k)}$ as a complete training set, increment $k=k+1$ and go to step 4.

20:Set k at the highest level of partition $k=K=max(k)$. Consider the training set $n^{(K)} = \{n_{11}^{(K)},...,n_{1J_K}^{(K)},n_2^{(K)}\}$ and generate an initial population $\underset{s}{IP}_{INC}(\underset{s_1/2}{X}_{11END}^{(K)},...,\underset{s_1/2}{X}_{1J_K END}^{(K)},\underset{s_1/2}{X}_{2END}^{(K)})$.

21:Evolve an RFNN, using the training set $n^{(k)}$, until $\frac{2}{s_1}\sum_{i=1}^{s_1/2} \max_{n^{(k)}}(Q(\chi_i)) < Q_{INC}$. Keep the result $\underset{s_1/2}{X}_{INC}^{(k)}$.

22:Decrease $k=k-1$, which is equivalent to increasing the incremental level. Consider the training set $n^{(k)} = \{n_{11}^{(k)},...,n_{1J_K}^{(k)},n_2^{(k)} = n^{(k+1)}\}$ and generate an initial population $\underset{s}{IP}_{INC}(\underset{s_1/2}{X}_{11END}^{(k)},...,\underset{s_1/2}{X}_{1J_k END}^{(k)},\underset{s_1/2}{X}_{INC}^{(k+1)})$. If k>1, go to step 20.

23:Evolve an RFNN, using the training set $n^{(1)}$, until Q passes Q_{INCEND},

$$Q < Q_{INCEND}. \tag{17a}$$

Keep the best chromosome χ_{INCEND}^{best}. It represents the completely evolved RFNN.

The BIE strategy above implements a dynamic objective function. The fitness function f_Q, on the other hand, will be the same over all steps, for an easy comparison with the direct evolution results.

$$f_Q = \begin{cases} 0, Q > 0.1515 \\ (0.1515 - Q) * 1000/1.5, Q \leq 0.1515 \end{cases} \tag{17b}$$

Direct evolution uses objective (17a) and fitness function (17b).

V. EMPIRICAL RESULTS

The empirical data cover nine six-month projects over the period June 1998 to December 1999 and involve three UK companies: Goodwin, Dixons Group and Marks&Spencer. When the data fuzzification procedure from section II is applied, the resultant log cashflow streams incorporate modelled market uncertainty. Figure 3 illustrates different types of uncertainty that characterise the corresponding companies and affect the estimated fuzzy share price. The three firms are chosen from a database of 35 companies to represent three major types of

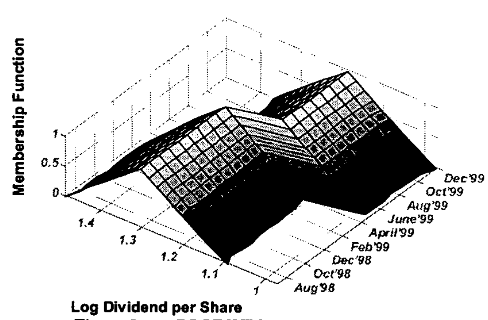

Figure 3a: GOODWIN – fuzzy log-cashflow stream July 1998 – December 1999

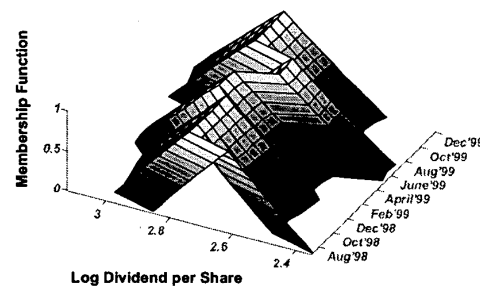

Figure 3b: DIXONS GROUP – fuzzy log-cashflow stream July 1998 – December 1999

company behaviour under increased market uncertainty. A firm - Goodwin - with continuously improving levels of investment risk, reaching $\alpha_{critical}=0$ in the final period. A company - Dixons Group - with oscillating risk levels $0 \leq \alpha_{critical} \leq 1$ over the periods. Finally, a company - Marks&Spencer - with $\alpha_{critical}=1$ over all periods. Thus, it is provided that companies of various types enter the training set. It is also guaranteed that the set includes single projects covering the three important risk ranges: $\alpha_{critical}=0$, the open interval $0<\alpha_{critical}<1$, and $\alpha_{critical}=1$. The risk values of all projects are presented in Table 1 and an exemplary log present value is depicted in Figure 4. The overall data are divided into three parts: a training set, a test set and a prediction project. The training set covers projects 1 to 6 from the first two rows in Table 1. Then, two of the projects in the third row are used as a test set for $\tilde{P}_{RFNN}$. Finally, the evolved RFNN is

TABLE 1: INVESTMENT RISK RESULTS

company / period	GOODWIN $\alpha_{critical}$		DIXONS GROUP $\alpha_{critical}$		MARKS&SPENCER $\alpha_{critical}$	
July-Dec'98	train1:	*1*	train3:	*1*	train5:	*1*
Jan-June'99	train2:	***0.683***	train4:	***0***	train6:	*1*
July-Dec'99	test1:	***0***	predict:	***0.858***	test2:	*1*

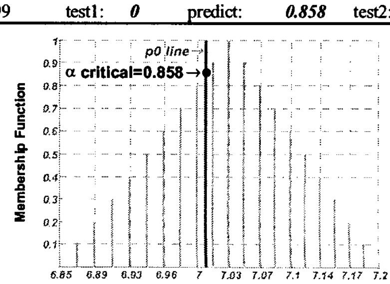

Figure 4: INVESTMENT RISK $0<\alpha_{CRITICAL}=0.858<1$ project DIXONS GROUP July – Dec 1999

applied to predict the investment risk of the project from Fig. 4.

The investment horizon is relatively short, $T=6$. Experimenting with several network structures and considering the trade-off between RFNN simplicity and evolution convergence, the values $m=5$ and $q=3$ are selected. Next, the set of parameters required by the empirical realisation of the algorithm are chosen: $s=100$, $s_l=30$, $\tau=0.1$, $Q^{(1)}_{DEC1}=0.1525$, $Q^{(1)}_{DEC2}=0.0525$, $Q^{(1)}_{DEC3}=0.0125$, $Q^{(1)}_{DEC5}=0.0775$, $Q^{(2)}_{DEC2}=0.0225$, $Q^{(2)}_{DEC5}=0.0675$, $Q^{(3)}_{DEC1}=0.064$, $Q^{(3)}_{DEC2}=0.0425$, $Q_{DECEND}=0.0025$, $Q_{INC}=0.002$, $Q_{INCEND}=0.0015$. The partition results include:

$$\begin{aligned}
&\text{Level}1: n^{(1)}=\{n^{(1)}_{11},n^{(1)}_{12},n^{(1)}_{2}\},\ n^{(1)}_{11}=\{project\,1\},\ n^{(1)}_{12}=\{project\,5\}\\
&\qquad n^{(1)}_{2}=\{project\,2, project\,3, project\,4, project\,6\}\\
&\text{Level}2: n^{(2)}=\{n^{(2)}_{11},n^{(2)}_{2}\},\ n^{(2)}_{11}=\{project\,6\},\\
&\qquad n^{(2)}_{2}=\{project\,2, project\,3, project\,4\}\\
&\text{Level}3: n^{(3)}=\{n^{(3)}_{11},n^{(3)}_{2}\},\ n^{(3)}_{11}=\{project\,3, project\,4\},\\
&\qquad n^{(3)}_{2}=\{project\,2\}
\end{aligned} \quad (18)$$

The performance of bi-directional incremental evolution and direct evolution is compared in Figure 5. Maximum fitness per generation is presented for each strategy. DE first progresses slightly and then stalls. Its plot, the lighter-colour line, is an average of *5* simulations. On the other hand, BIE proceeds through several task transitions and eventually evolves a fully functional RFNN after *148243* generations. Its graph, the black lines, is a result of one simulation.

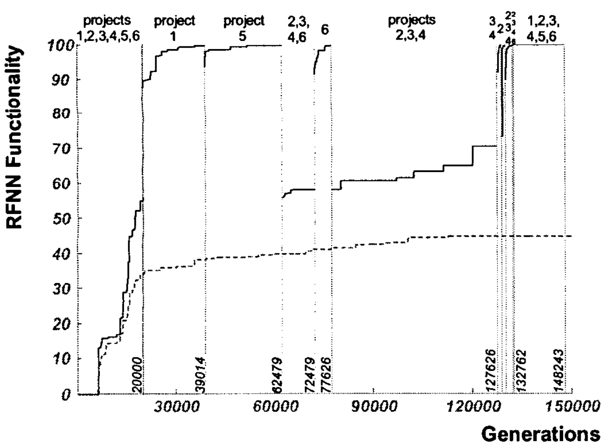

Figure 5: Performance of BIE and DE –maximum fitness per generation

- *black linse* – bidirectional incremental evolution: advances through several decomposition and incremental tasks and solves the general problem in *148243* generations
- *lighter line* – direct evolution: makes an initial progress and then stalls

Direct evolution reaches *46.33%* maximum fitness after *500 000* generations. Thus, the empirical results prove bidirectional incremental evolution quite more efficient.

Table 2 presents the error of the evolved RFNN over the test set. The error is relatively small and can be improved using a larger training set. The prediction for the investment risk measure of project Dixons Group, July'99 - December'99, is $\alpha_{critical}=0.88$ and can be compared with the true value in Fig.4.

TABLE 2: TEST RESULTS

company / period	GOODWIN *RFNN error, Q*	MARKS&SPENCER *RFNN error, Q*
July-Dec'99	***0.0091***	***0.0024***

VI. FUTURE RESEARCH

Further two stages in the proposed method should solve the following problems. First, at the moment the attached hybrid part of the fuzzy network only describes involved calculations. A corresponding training algorithm is required. Second, if the robustness measure from [22] is introduced, then a slightly modified network structure can be trained as a classifier of both low-risk and highly-robust projects.

VII. ACKNOWLEDGEMENTS

The authors are grateful to Prof. J. Buckley for proof-reading an earlier version of the fuzzy criterion and the investment risk measure. We would also like to acknowledge the cooperation of the Department of Economics and Finance, Brunel University, where the empirical data have been collected from Datastream.

VIII. REFERENCES

[1] J. Aluja, "Towards a New Paradigm of Investment Selection in Uncertainty," FSS, Vol. 84, pp 187-197, 1996.

[2] J. Buckley, "The Fuzzy Mathematics of Finance," FSS, Vol. 21, pp. 257-273, 1987.

[3] J. Buckley, "Solving Fuzzy Equations," FSS, Vol. 50, pp. 1-14, 1992.

[4] J. Buckley, E. Eslami and Y. Hayashi, "Solving Fuzzy Equations Using Neural Nets," FSS, Vol. 86, pp. 271-278, 1997.

[5] J. Buckley and T. Feuring, Fuzzy and Neural: Interactions and Applications, Physica-Verlag, 1999.

[6] J. Buckley and Y. Hayashi, "Can Fuzzy Neural Nets Approximate Continuous Fuzzy Functions," FSS, Vol. 61, pp. 323-330, 1994.

[7] M. LiCalzi, "Towards a General Setting for the Fuzzy Mathematics of Finance," FSS, Vol. 35, pp. 281-293, 1990.

[8] J. Campbell, A. Lo and A. MacKinlay, The Econometrics of Financial Markets, Princeton U. Press, 1997.

[9] A. Dixit, R. Pindyck and S. Sodal, "A Markup Interpretation of Optimal Rules for Irreversible Investment," National Bureau of Economic Research WP, Vol. 5971, 1997.

[10] D. Fairclough and J. Hunter, "The Ex-ante Classification of Takeover Targets Using Neural Networks", In A.-P. Refenes and J. Moody, Eds., Decision Technologies for Computational Finance, Kluwer Academic, 1998.

[11] D. Filliat, J. Kodjabachian and J. Meyer, "Incremental Evolution of Neural Controllers for Navigation in a 6-legged Robot," In Sugisaka and Tanaka, Eds., Proc. of the 4th Int. Symp. on Artificial Life and Robots, Oita U. Press, 1999.

[12] F. Gomez and R. Miikkulainen, "Incremental Evolution of Complex General Behaviour," Adaptive Behaviour, Vol. 5, pp. 317-342, 1997.

[13] J. Hunter and A. Serguieva, "Project Risk Evaluation Using an Alternative to the Standard Present Value Criteria," Neural Network World, Vol. 10, pp. 157-172, 2000.

[14] T. Kalganova, " Bidirectional Incremental Evolution in Extrinsic Evolvable Hardware," In J. John, A. Stoica, et al, Eds., Proc. of the 2nd NASA/DoD Workshop on Evolvable Hardware, pp. 65-74, 2000.

[15] D. Kuchta, "Fuzzy Capital Budgeting," FSS, Vol. 111, pp. 367-385, 2000.

[16] P. Liu, "Universal Approximations of Continuous Fuzzy-valued Functions by Multi-layer Regular Fuzzy Neural Networks," FSS, Vol. 119, pp. 313-320, 2001.

[17] J. Miller, T Kalganova, N. Lipnitskaya and D. Job, "The Genetic Algorithm as a Discovery Engine: Strange Circuits and New Principles", In Proc. of the Symposium on Creative Evolutionary Systems, pp. 65-74, 1999.

[18] R. Pike, "A Longitudinal Survey on Capital Budgeting Practices," J. Business Finance & Accounting, Vol. 23, pp. 79-92, 1996.

[19] M. Precious, Rational Expectations, Non-market Clearing and Investment Theory, Oxford U. Press, 1987.

[20] R. Ribeiro, H.-J. Zimmermann, R. Yager and J. Kacprzyk, Eds., Soft Computing in Financial Engineering, Physica-Verlag, 1999.

[21] A. Sangster, "Capital Investment Appraisal Techniques: A Survey of Current Usage," J. Business Finance & Accounting, Vol. 20, pp. 307-332, 1993.

[22] A. Serguieva, T. Kalganova and J. Hunter, "Soft Computing in Investment Risk Appraisal," in Proc. of the International Conference of the European Society in Fuzzy Logic and Technology, pp.214-219, 2001.

[23] R. Stulz, "What is Wrong with Modern Capital Budgeting?," Address delivered at the Eastern Finance Association meeting, Miami Beach, 1999.

[24] L. Zadeh, "Fuzzy Sets," Information & Control, Vol. 8, pp. 338-353 1965.