Computational Intelligence Techniques in Asset Risk Analysis

by

Antoaneta Serguieva

submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy

at

Department of Electronic and Computer Engineering

in collaboration with Department of Economics and Finance

Brunel University

BRUNEL UNIVERSITY

May 2004

Computational Intelligence Techniques in Asset Risk Analysis

Antoaneta Serguieva

Department of Electronic and Computer Engineering
in collaboration with Department of Economics and Finance
Brunel University, Uxbridge, Middlesex UB8 3PH, UK

PhD Thesis, May 2004

Abstract

The problem of asset risk analysis is positioned within the computational intelligence paradigm. We suggest an algorithm for reformulating asset pricing, which involves incorporating imprecise information into the pricing factors through fuzzy variables as well as a calibration procedure for their possibility distributions. Then fuzzy mathematics is used to process the imprecise factors and obtain an asset evaluation. This evaluation is further automated using neural networks with sign restrictions on their weights. While such type of networks has been only used for up to two network inputs and hypothetical data, here we apply thirty-six inputs and empirical data. To achieve successful training, we modify the Levenberg-Marquart backpropagation algorithm.

The intermediate result achieved is that the fuzzy asset evaluation inherits features of the factor imprecision and provides the basis for risk analysis. Next, we formulate a risk measure and a risk robustness measure based on the fuzzy asset evaluation under different characteristics of the pricing factors as well as different calibrations. Our database, extracted from DataStream, includes thirty-five companies traded on the London Stock Exchange. For each company, the risk and robustness measures are evaluated and an asset risk analysis is carried out through these values, indicating the implications they have on company performance. A comparative company risk analysis is also provided. Then, we employ both risk measures to formulate a two-step asset ranking method. The assets are initially rated according to the investors' risk preference. In addition, an algorithm is suggested to incorporate the asset robustness information and refine further the ranking benefiting market analysts.

The rationale provided by the ranking technique serves as a point of departure in designing an asset risk classifier. We identify the fuzzy neural network structure of the classifier and develop an evolutionary training algorithm. The algorithm starts with suggesting preliminary heuristics in constructing a sufficient training set of assets with various characteristics revealed by the values of the pricing factors and the asset risk values. Then, the training algorithm works at two levels, the inner level targets weight optimization, while the outer level efficiently guides the exploration of the search space. The latter is achieved by automatically decomposing the training set into subsets of decreasing complexity and then incrementing backward the corresponding subpopulations of partially trained networks. The empirical results prove that the developed algorithm is capable of training the identified fuzzy network structure. This is a problem of such complexity that prevents single-level evolution from attaining meaningful results.

The final outcome is an automatic asset classifier, based on the investors' perceptions of acceptable risk. All the steps described above constitute our approach to reformulating asset risk analysis within the approximate reasoning framework through the fusion of various computational intelligence techniques.

Acknowledgements

I warmly thank my supervisor Dr Tatiana Kalganova, Department of Electronic and Computer Engineering, Brunel University, for her encouragement and support throughout my student life. I am also thankful to Dr Tariq Khan, Brunel School of Business and Management, for the opportunity to incorporate the ideas presented here in a broader framework. Acknowledgements are further due to Dr John Hunter, Department of Economics and Finance, Brunel University, for the introduction to the PhD study at Brunel.

I am grateful to Prof. Didier Dubois and Prof. Henri Prade, Editors-in-Chief of Fuzzy Sets and Systems, for their guidance toward an optimised presentation of the article published there and drawn on the work in this thesis. I express my gratitude to Prof. Piero Bonissone, President of IEEE Neural Network Society for 2002-2003, for reviewing articles based on the thesis, and refereing me for the IEEE NNS grant. I also appreciate the review by Prof. James Buckley, Department of Mathematics, University of Alabama at Birmingham, on a partial draft of this work. His publications have initially inspired me for the PhD study.

I am indebted to my family for giving me the strength and the reason to go on.

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 in May 2004 under a supervised PhD programme 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 [P2, P3],
- Chapter 4 in [P3,P 4], Chapter 5 in [P4,P 5],
- Chapter 6 in [P6], Chapter 7 in [P7],
- Chapter 8 in [P8,P 9, P10].

The following are considered original contributions:

- definition of an algorithm for fuzzy asset evaluation,
- definition of an asset risk measure,
- definition of an asset robustness measure,
- developing an asset ranking technique,
- designing a soft asset classifier,
- developing an evolutionary training strategy.

Finally, the suggested approach to knowledge representation in risk analysis is subject to further research.

The original contributions results from the adopted view that the problem of asset risk analysis should be in the centre of the research effort. Therefore, the effort is focused on revealing the real-world problem in its various aspects, and then suggesting means to resolve them. This is in contrast with a view that prioritises technical and/or original developments from the outset. Thus, we define measures, develop algorithms, techniques and strategies, and design a classifier system, when and how the problem requires. Intriguingly, this results at the end in original contributions due to the following major reasons: the intrinsic complexity of the problem; our conviction that it should be investigated in its entirety rather than concentrating on one feature and ignoring the other characteristics; the deficiency of methods for tackling problems of such complexity.

Here, the contributions are revisited again providing details and justification for each of them.

- definition of an algorithm for fuzzy asset evaluation:

The rationale behind this contribution is as follows. The emerging field of behavioural finance is receiving increasing recognition as financial theory. This emphasises the role the perceptions of various market agents have on the development of the market itself. Shiller [1] tests conventional theories against the impressive evidence- particularly toward the end of the last century - suggesting that the available economic information alone does not explain asset price levels. He identifies characteristics of human behaviour that have major effects on asset price trends. Apparently, we can conclude that the asset pricing problem is influenced by both measurement-based (economic) information and perception-based (behavioural) information, and the solution should allow for the fusion of these types of information. Conventional pricing techniques lack effective means in approaching the task of information fusion. Their answer is to identify information restrictions under which the corresponding technique is valid. The best that can be achieved with a standard pricing method is to attempt to relax to some extent some of the restrictions, prove that the initial method is not valid, and introduce partial modifications so that the revised method is valid under the revised set of restrictions. This describes the trend in asset pricing over some decades, i.e. as shown in the literature on the capital asset pricing model [2-15] and the arbitrage pricing theory [16-28], summarised in [P3]. On the other hand, the computational theory of perceptions introduced recently by Zadeh [29-33] provides a powerful approach to information fusion and processing based on fuzzy logic. Aluha [34] anticipated earlier that investment knowledge should not start with the estimation of economic and financial variables in terms of certainty or probability but with the perceptions of concepts inherent or surrounding the investment process whose character is not principally measurable and therefore can be handled by the nonnumeric mathematics of fuzzy logic. This gives us the reason to reformulate asset evaluation as a fuzzy problem, avoid limited attempts to modify crisp techniques, and approach the problem allowing any relevant information. The above description provides the rationale to develop an algorithm for fuzzy asset evaluation. We suggest when and how to introduce fuzzy variables, how to calibrate the membership functions and when to modify the calibration, and how to relate asset imprecision with factor imprecision. In comparison with relevant studies, our technique differs in the following way. Fuzzy modelling of simpler concepts in finance is attempted in [35-38] and not the reformulation of asset evaluation approached here. Also [35-38] only use made-up data, therefore no meaningful calibration procedure is suggested and the results are not valuable empirically. On the other hand, we have extracted a database of thirty-five assets traded on the LSE with information on their pricing factors over twenty-five years. Finally, previous studies have not suggested any algorithm, while here we describe step by step when and how to introduce imprecise information and when to extract what measures relevant to the subsequent analysis. Initially, we consider reformulation of the price-dividend relation, based on the reasoning in Shiller [1] that dividends and risk-free rates do not explain asset prices in the last decades of the twentieth century – our database is over the same period. Then, we further generalise the algorithm for any crisp pricing technique.

The next three contributions are interrelated and we will describe them together.

- definition of an asset risk measure,
- definition of an asset robustness measure,
- developing an asset ranking technique:

Some of the crisp asset pricing techniques, including models derived from the arbitrage pricing theory, consider a vector of pricing factors and a vector of noises. The latter are considered sources of risks. The attempt is to reduce these risks by identifying relevant factors. We adopt an alternative approach incorporating factor incompleteness as well as factor imprecision into the factor representation. Then, we consider the noise components as sources of factor uncertainty rather than asset risks. Our objective is to allow any sources of uncertainty and to process the relevant information producing the resultant asset evaluation. Thus, we focus on a single all-inclusive asset risk. This is relevant to the general interpretation of processing uncertainty presented by Dubois and

Prade in [39]. Next, consider the capital asset pricing model - it is effectively a single-factor model where the pricing factor is the return on the market portfolio. Though, some CAPM modifications involve a three-fund separation theorem [9], so there are two pricing factors. Generally, the CAPM interprets the asset variance as the total risk, which consists of systematic and unsystematic risk. The systematic risk is that part of the total risk that can be explained with the variance of the market portfolio through the asset beta, which is the covariance of the asset and the market portfolio. The unsystematic risk is the unexplained part of the asset variance, which is the variance of the noise component. For example. the partial character of the asset beta in the capital asset pricing model is revealed by the existence of further measures of asset performance, introduced by Jensen, Treynor and Sharpe, correspondingly. They are used to capture the part of the asset return unexplained by the model [5]. In comparison, we do not consider the factor variance and the noise variance as risks but rather as sources of uncertainty on which basis to derive the risk. The fusion of perception-based uncertainty is further welcome in addition to probabilistically calculated variances. The intuition behind the risk measure we introduce is as follows. The risk is measured through the level of membership of the observed asset price to the evaluated fuzzy asset price, which itself is based on processing any involved uncertainty and reflect those in its membership function. Thus the risk measure is a single number, however all-inclusive of any type of uncertainty, and therefore more informative. Another reason for the measure being more informative is its focus on the final objective. For example, if one considers investing in the asset today, our risk measure will indicate the chance, as measured today, of realising loss on the investment. Focusing on the final objective is also relevant to the general interpretation of processing uncertainty and decision-making under uncertainty, presented by Dubois and Prade in [39]. They suggest that any kind of uncertainty should be incorporated into a problem and processed though its solution, and finally an indication should be provided for the total effect on the major problem objective. We further continue with introducing a robustness measures. It is based on evaluating the asset and its risk measure while changing the modelled factor imprecision. As a result, the asset membership function will change, for some assets more and for others less. So, we have a reason to call the measure an asset robustness measure. However, the induced change will further modify the membership value of the observed asset price to the evaluated price. Therefore, it is easier to measure asset robustness through the change of the risk value. Thus, this will again be one number but all-inclusive and focused on the final objective. It is focused on the final objective for the following reason. The objective is to evaluate reliably the chance of realising loss on the asset. The revised asset evaluation may change a lot and still affect little the initial risk value. On the other hand, the asset evaluation may change only a little but producing a significant effect on the risk value. Therefore, the measure of asset robustness we suggest is dictated by the final objective. Finally, though interrelated the two measures analyse an asset from different perspectives. Thus, the asset may be low risky but low robust, or highly risky but highly robust as well, etc. Considering Definition 3.1 and Definition 3.3 in Chapter 3, the intuition is that a market agent will prefer less risky and highly robust assets. It is why we formulate an asset ranking technique based on the two measures, and suggest a particular procedure and rules of incorporating the measures and refining the final ranking. As a result of the rationale presented in this paragraph, the ranking technique is considerably original. For the reasons of comparison, it may be related to techniques of ranking fuzzy numbers [40-43]. However they represent approaches to ordering membership functions, rather than formulating risk and robustness measures and incorporating their effect. So, the former ranking will not address the final objective and thus will not be helpful. If considering a ranking based on risk measures from the crisp asset evaluation methods, they will experience the limitations of those measures, as described here earlier, and the lack of robustness measures. Therefore, there is no much ground for comparison. Finally, we investigate here the price-dividend relation, because it is the relation Shiller [1] bases its argument on the market behaviour toward the end of last century. The argument is that perception based information is important, and we provide a technique to incorporate this. At the end, we compare for that model the crisp results and the results from our approach involving computational intelligence techniques. On the other hand, the same approach can be applied starting from the crisp capital asset pricing model or a crisp arbitrage pricing theory model, and then to compare for them the crisp results and the soft evaluations. This will require a corresponding database.

Finally, the last two contributions involves the design and training of an asset risk classifier:

- designing a soft asset classifier,
- developing an evolutionary training strategy.

The classifier is based on the rationale behind the developed measures and ranking technique above. The identification of the architecture involves investigating the approximation capabilities of crisp and fuzzy neural networks. We first three-layer crisp networks with fuzzy restrictions on the weights to approximate fuzzy asset evaluation. This type of networks are suggested in Buckley [44], however only networks of maximum two inputs have been trained and the training set involves made-up data. In comparison, here we train networks with 36 inputs and use an empirical database of 36 assets over 25 years. Buckley [45], suggests that backpropagation is not applicable in the training, however we modify the Levenberg-Marquart algorithm to accommodate for the sign restrictions and successfully train the networks. Still, the results prove that this type of network is not able to approximate the complex problem of fuzzy asset classification. In Liu [46], it is theoretically proved that four-layer fuzzy networks are universal approximator of fuzzy-valued functions, while no experimental result are considered. We extend the proof provided there to the multivariable case, and train the fuzzy classifier within the empirical database. Furthermore, we develop an evolutionary training algorithm. The algorithm works on two levels, out of the three levels described in Yao [47]. The first level involves optimisation of the crisp and fuzzy network weights, therefore it is concerned with searching the parameters space. The second level is concerned with exploring the training set itself, or guidance of the learning process for the parameters from the first level, i.e. it is optimisation of the optimisation. Furthermore, the two-level exploratory algorithm is based on the general concepts of divide-and-conquer evolution and incremental evolution, the same concepts that are the basis of the bidirectional incremental evolution in Kalganova [48]. Based on those concepts, the algorithm developed in this thesis is entirely new in its design and in its implementation. As an initial stage in the algorithm, we identify database heuristics and use them in constructing the training set, then design a dynamic objective training function. The empirical results are compared with those of a conventional evolutionary algorithm and prove the efficiency of the new algorithm. Finally, the complexity of the problem and approaches by other authors to relevant problems are provided in a comparative table. The important contribution is the development of an evolutionary algorithm able to resolve a problem of high complexity. A significant aspect of the contribution is the application to a real-world problem. The complexity of the problem arises from the following reasons. The asset risk takes values within different qualitative ranges. Assets are also characterised with qualitatively different risk behaviour in time. The fuzzification of the asset pricing factors and the processing of imprecise information is performed by the classifier itself. The neural network structure involves fuzzy weights.

All the contributions, in the sequence explained here, work towards resolving the real-world problem.

Statement of Copyright

Contents

Symbols

D_t	is the dividend per share in period t
DY_t	is the dividend yield in period t
$\tilde{DY}_t$	is the fuzzy interval substituted for the dividend yield in period t
$\breve{DY}_t$	is the possibilistic variable substituted for the dividend yield in period t
dy_t	is the logarithmic dividend yield in period t
$\tilde{dy}_t$	is the fuzzy interval substituted for the logarithmic dividend yield in period t
$\breve{dy}_t$	is the possibilistic variable substituted for the logarithmic dividend yield in period t
$f(x)$	is a crisp function, which is a mapping from $\mathbb{R}$ to $\mathbb{R}$
$\tilde{f}(\tilde{x})$	is a fuzzy function, which is a mapping from $\Im(\mathbb{R})$ to $\Im(\mathbb{R})$
$\tilde{f}(x)$	is a fuzzy-valued function, which is a mapping from $\mathbb{R}$ to $\Im(\mathbb{R})$
I	is an identity transfer function of a neuron in the neural network
$\tilde{I}$	is a fuzzy identity transfer function of a neuron in the fuzzy network

$\underset{\gamma}{IP}\left(\underset{\gamma_I}{X^{(\kappa)}}\right)$

is a population of full-size γ to be used over a complete training set $task^{(\kappa)}$ and generated by recombination of a breeding subpopulation $\underset{\gamma_I}{X^{(\kappa)}}$ evolved over a complete training set at level of decomposition κ

$IP_{\gamma}{}_{1j}\left(X^{(\kappa)}_{\gamma_1}\right)$

is a population of full-size γ to be used over
a training subset $task^{(\kappa)}_{1j}$, $1 \le j \le J_{\kappa}$, and generated
by recombination of a breeding subpopulation $X^{(\kappa)}_{\gamma_1}$
evolved on a complete training set $task^{(\kappa)}$

$IP_{\gamma}{}_{1j}\left(X^{(\kappa)}_{\gamma_1}{}_{1j}\right)$

is a population of full-size γ to be used over
a training subset $task^{(\kappa)}_{1j}$, $1 \le j \le J_{\kappa}$, and generated
by recombination of a breeding subpopulation $X^{(\kappa)}_{\gamma_1}{}_{1j}$
evolved on the subset $task^{(\kappa)}_{1j}$

$IP_{\gamma}{}_{INC}\left(X^{(\kappa_m)}_{\gamma_1/2}{}_{11END}, \ldots, X^{(\kappa_m)}_{\gamma_1/2}{}_{1J_{\kappa_m}END}, X^{(\kappa_m)}_{\gamma_1/2}{}_{2END}\right)$

is the full-size population to be used at the first incremental level
over the set $task^{(\kappa_m)}$, and generated by recombination of
breeding subpopulations of size $\gamma_1/2$ evolved separately over
the subsets $task^{(\kappa_m)}_{1j}, task^{(\kappa_m)}_{2}, 1 \le j \le J_{\kappa_m}$ at the final decomposition level

$IP_{\gamma}{}_{INC}\left(X^{(\kappa)}_{\gamma_1/2}{}_{11END}, \ldots, X^{(\kappa)}_{\gamma_1/2}{}_{1J_{\kappa}END}, X^{(\kappa+1)}_{\gamma_1/2}{}_{INC}\right)$

is the full-size population to be used at some incremental level $\kappa_m - \kappa + 1$
over the set $task^{(\kappa)}$, and generated by recombination of
breeding subpopulations of size $\gamma_1/2$ evolved separately over
the subsets $task^{(\kappa)}_{1j}, 1 \le j \le J_{\kappa_m}$ at some level of decomposition κ,
and the breeding subpopulation evolved over $task^{(\kappa+1)}$
at the previous incremental level

mse	is the minimum mean square error of the crisp neural network
N_{gen}	is the number of generations in the probing step used in the evolutionary algorithm
P_t	is the asset price in period t
$\tilde{P}_t$	is the fuzzy interval substituted for the asset price in period t
$\tilde{P}_0$	is the evaluated fuzzy interval for the current asset price
$\breve{P}_t$	is the possibilistic variable substituted for the asset price in period t
p_t	is the logarithmic asset price in period t
$\tilde{p}_t$	is the fuzzy interval substituted for the logarithmic asset price in period t
$\tilde{p}_0$	is the evaluated fuzzy interval for the logarithmic current asset price
$\breve{p}_t$	is the possibilistic variable substituted for the logarithmic asset price in period t
$p_{0_{ANN}}$	is the output of the crisp neural network approximating the $\alpha-cuts$ of $\tilde{p}_0$
$\tilde{p}_{0_{FNN}}$	is the fuzzy logarithmic asset price approximated by the fuzzy neural network
$Poss$	is the possibility operator

$Poss\left[\breve{DY}_t = x_{DY_t}\right]$
is the calibrated possibility distribution of the dividend yield in period t

$Poss\left[\breve{dy}_t = x_{dy_t}\right]$
is the calibrated possibility distribution of the logarithmic dividend yield in period t

$Poss\left[\breve{P}_t = x_{P_t}\right]$
is the calibrated possibility distribution of the asset price in period t

$Poss\left[\breve{P}_0 = x_{P_0}\right]$
is the evaluated possibility distribution for the current asset price

$Poss\left[\breve{p}_t = x_{p_t}\right]$
is the transformed possibility distribution of
the logarithmic asset price in period t

$Poss\left[\breve{p}_0 = x_{p_0}\right]$
is the evaluated possibility distribution for the logarithmic current asset price

$Poss\left[\breve{R} = x_R\right]$
is the calibrated possibility distribution of the constant return

$Poss\left[\breve{r}_t = x_{r_t}\right]$
is the transformed possibility distribution of
the time-varying return in period t

Q is the current trading price

q is the logarithmic current trading price

R is the constant asset rate of return

$\tilde{R}$ is the fuzzy interval substituted for the constant rate of return

$\breve{R}$ is the possibilistic variable substituted for the constant rate of return

R_t is the time-varying rate of return

$\tilde{R}_t$ is the fuzzy interval substituted for the time-varying rate of return in period t

$\breve{R}_t$ is the possibilistic variable substituted for
the time-varying rate of return in period t

r_t is the logarithmic time-varying rate of return

$\tilde{r}_t$ is the fuzzy interval substituted for the logarithmic time-varying rate of return in period t

$\breve{r}_t$ is the possibilistic variable substituted for the logarithmic time-varying rate of return in period t

$task$ is the training set for the fuzzy network and consists of elements $(asset \times period, \tilde{p}_0)$

$task^{(\kappa)}$ is the training set for the fuzzy network at level of decomposition κ

$task_1^{(\kappa)}, task_2^{(\kappa)}$
are the subsets in the initial decomposition of the overall task $task^{(\kappa)}$ at level κ

$task_{1j}^{(\kappa)}$ is a subset in the consequent decomposition of $task_1^{(\kappa)}$ at level κ, where $1 \le j \le J_\kappa$; the subsets, their number J_κ and their size are automatically discovered by the BIE

α is a level of membership

$\alpha - cut$ is a level interval in a fuzzy interval

χ is an individual chromosome encoding a fuzzy network

χ_{best} is the best fitted chromosome in a population

χ_{INCEND}^{best} is the best fitted chromosome at the end of the two-level exploratory algorithm

X is a population of chromosomes

X_{SUB} is a breeding subpopulation of chromosomes

$X_{\gamma_1}^{(\kappa)}$ is a breeding subpopulation of size γ_1 evolved over

a complete training set $task^{(\kappa)}$ at level of decomposition κ

$X_{\gamma_1}{}_{1j}^{(\kappa)}$ is a breeding subpopulation of size γ_1 evolved over

a subset $task_{1j}^{(\kappa)}$ at level of decomposition κ, where $1 \leq j \leq J_\kappa$,

$task_1^{(\kappa)} = \left\{ task_{11}^{(\kappa)}, \ldots, task_{1J_k}^{(\kappa)} \right\}$ and $task^{(\kappa)} = \left\{ task_1^{(\kappa)}, task_2^{(\kappa)} \right\}$

$X_{\gamma_1/2}{}_{INC}^{(\kappa)}$ is the better half of a breeding subpopulation evolved over

a complete set $task^{(\kappa)}$ at some incremental level $\kappa_m - \kappa + 1$

δ_1, δ_2 are parameters of linearisation in the logarithmic pricing function

$\varepsilon_{interval}$ is the error evaluating the interval approximating capability of the trained crisp neural network

$\Phi\left(\tilde{p}_{0_{FNN}}, q\right)$

is the neuron evaluating the asset risk in the classifier structure

γ is the number of chromosomes in a full-size population at each step of the evolutionary algorithm

γ_1 is the size of a breeding subpopulation

λ is the number of crossover points applied to a parent pair of chromosomes to produce an offspring

λ_1 is the number of crossover points applied over the single-number genes of the parent pair of chromosomes

λ_2 is the number of crossover points applied over the triplet genes of the parent pair of chromosomes

μ is a membership function

$\mu\left(x_{DY_t} \middle| \tilde{DY}_t\right)$
is the calibrated membership function of the fuzzy dividend yield in period t

$\mu\left(x_{dy_t} \middle| \tilde{dy}_t\right)$
is the transformed membership function of
the fuzzy logarithmic dividend yield in period t

$\mu\left(x_{P_t} \middle| \tilde{P}_t\right)$ is the calibrated membership function of the fuzzy asset price in period t

$\mu\left(x_{P_0} \middle| \tilde{P}_0\right)$
is the evaluated membership function for the current asset price

$\mu\left(x_{p_t} \middle| \tilde{p}_t\right)$
is the transformed membership function of
the fuzzy logarithmic asset price in period t

$\mu\left(x_{p_0} \middle| \tilde{p}_0\right)$
is the evaluated membership function of the current logarithmic asset price

$\mu\left(x_R \middle| \tilde{R}\right)$ is the calibrated membership function of the fuzzy constant return

$\mu\left(x_{R_t} \middle| \tilde{R}_t\right)$
is the calibrated membership function of
the fuzzy time-varying return in period t

$\mu\left(x_{r_t} \middle| \tilde{r}_t\right)$ is the transformed membership function of
the fuzzy logarithmic return in period t

π is the level of possibility

θ is the bias term of the sigmoid neuron-transfer function

Ω is a sigmoid transfer function of a neuron in the neural network

τ is the rate of mutation used in the evolutionary algorithm

υ is a level of uncertainty in the pricing factors

υ_R is the level of uncertainty in the pricing factors which brings about the risk value in the evaluated current asset price

$\xi(\chi_i)$ is the cost function evaluating the error of the regular fuzzy network encoded in chromosome χ_i

$\xi_{DEC1}^{(\kappa)}, \xi_{DEC2}^{(\kappa)}, \xi_{DEC3}^{(\kappa)}, \xi_{DEC4}^{(\kappa)}, \xi_{DEC5}^{(\kappa)}$

are parameters used at different steps of the decomposition part of the two-level exploratory algorithm in its dynamic objective function

ξ_{DECEND} is the parameter used at final decomposition steps in the two-level exploratory algorithm in its dynamic objective function

ξ_{INC} is the parameter used throughout the incremental part of the two-level exploratory algorithm in its dynamic objective function

ξ_{INCEND} is the parameter used in the final objective of the two-level exploratory algorithm

ξ_{min}, ξ_{max}

are parameters defining a scope of network error in the fitness function $\zeta(\xi)$ used in the evolutionary algorithm

Ξ is the dynamic objective used throughout the steps of the two-level exploratory algorithm

$\Psi(\Re_{FNN}, \Re_{agent})$

is the neuron introducing the agent preferences into the risk classifier

$\zeta(\xi)$ is the fitness function used in both algorithms, single-level evolution and two-level evolution

Δ is the robustness measure

Δ_{21} is the robustness measure under a broader range of imprecision

Δ_{31} is the robustness measure under time-varying return

$\Delta_{acceptable}$

is a level of robustness that falls within the preference range of an investment agent

Δ_{agent} is the level of robustness which delimits the preference range of an investment agent $\Delta_{acceptable} \geq \Delta_{agent}$

$\Im(\mathbb{R})$ is the set of all fuzzy intervals defined on the real-number set $\mathbb{R}$

$\mathbb{N}$ is the set of natural numbers

$\wp$ is a class of fuzzy neural networks

$\wp\left\{FNN \middle| \tilde{f}_{FNN}(x)\right\}$

is a class of fuzzy neural networks with fuzzy-valued network-transfer functions

$\mathbb{R}$ is the set of all real numbers

$\Re$ is the investment risk measure

$\Re_1$ is the risk measure under initial calibration

$\Re_2$ is the risk measure under a broader range of imprecision

$\Re_3$ is the risk measure under time-varying return

$\Re_{acceptable}$

is a level of risk that falls within the preference range of an investment agent

$\Re_{agent}$ is the level of risk which delimits the preference range of an investment agent $\Re_{acceptable} \leq \Re_{agent}$

$\Re_{FNN}$ is the asset risk evaluated by the fuzzy neural network

$(asset \times period, \tilde{p}_0)$

is an element in the training set for the fuzzy network;
the element consists of real-valued pricing factors denoted with $asset \times period$ and the corresponding fuzzy asset evaluation $\tilde{p}_0$

$\left[\underline{p_{0_{FNN,\chi}}(\alpha)}, \overline{p_{0_{FNN,\chi}}(\alpha)} \right]$

is the $\alpha - cut$ of the fuzzy logarithmic asset price approximated by the fuzzy neural network encoded in chromosome χ

$\left\{ x_{DY_t} \middle| Poss\left[\breve{D}Y_t = x_{DY_t} \right] \geq \alpha \right\}$

represents all the values for the dividend yield in period t whose level of possibility is at least $\pi = \alpha$

$\left\{ x_{P_0} \middle| Poss\left[\breve{P}_0 = x_{P_0} \right] \geq \alpha \right\}$

represents all the values in the evaluated current asset price whose level of possibility is at least $\pi = \alpha$

$\left\{ x_{P_t} \middle| Poss\left[\breve{P}_t = x_{P_t} \right] \geq \alpha \right\}$

represents all the values for the asset price in period t whose level of possibility is at least $\pi = \alpha$

$\left\{ x_R \middle| Poss\left[\breve{R} = x_R \right] \geq \alpha \right\}$

represents all the values for the constant return whose level of possibility is at least $\pi = \alpha$

Abbreviations

ANN	artificial neural network
APT	arbitrage pricing theory
AR	approximate reasoning
BAA	British Accounting Association
BBN	Bayesian belief network
BIE	bidirectional incremental evolution
CAPM	capital asset pricing model
CC	chaotic computing
CIS CFTC	CIS Technical Committee on Computational Finance
CMFV	continuous multivariable fuzzy-valued function
CTP	computational theory of perceptions
CW	computing with words
DAG	directed acyclic graph
DNA	deoxyribonucleic acid
DSTE	Dempster-Shafer theory of evidence
EC	evolutionary computing
EFNN	evolutionary fuzzy neural network
ENN	evolutionary neural network
EFRBS	evolutionary fuzzy rule-based system
EFS	evolutionary fuzzy system
EP	evolutionary programming

FEA	fuzzy evolutionary algorithm
ES	evolutionary strategy
f-granularity	fuzzy granularity
fg-generalisation	fuzzification and fuzzy granulation (of concepts or techniques)
FL	fuzzy logic
FL/E	the epistemic facet of fuzzy logic
FL/L	the logical facet of fuzzy logic
FL/R	the relational facet of fuzzy logic
FL/S	the set-theoretic facet of fuzzy logic
FNN	fuzzy neural network
GA	genetic algorithm
GP	genetic programming
HFNN	hybrid fuzzy neural network
ICAEW	Institute of Chartered Accountants in England and Wales
IEEE	Institute of Electrical and Electronics Engineers
IEEE CIS	IEEE Computational Intelligence Society
LSE	London Stock Exchange
ML	machine learning
NC	neurocomputing
NFS	neuro fuzzy system
NYSE	New York Stock Exchange
PR	probabilistic reasoning
RFNN	regular fuzzy neural network

Figures

Tables

Definitions and Algorithms

Chapter 1: Introduction

1.1 Motivation

The area of computational intelligence has emerged recently on the basis of many computing disciplines introducing their symbiotic use. The principle components include fuzzy logic, neurocomputing, evolutionary computing and probabilistic reasoning. However further disciplines and forthcoming problem-solving technologies are continuously incorporated into the area. These involve chaotic computing, memetic algorithms, artificial life, swarm intelligence, DNA computing, to name a few. Indicative for the fast development of the area and its recognition is the establishment of a corresponding IEEE Society in 2002, with formulated interests in

> *'the theory, design, application and development of biologically and linguistically motivated computational paradigms emphasizing neural networks, genetic algorithms, evolutionary programming, fuzzy systems, and hybrid intelligent systems in which these paradigms are contained'*,

A further step in this direction is the decision to change the name of the Society from Neural Networks to Computational Intelligence, as from June 2004. The new name is more descriptive and inclusive than the old one, which only covered a fraction of the Society's scope, and the mission is extended with

> *'promoting activities in emerging fields such as data mining, bio-informatics, computational finance, computational neuroscience, autonomous mental development, and intelligent systems applications'*.

The characteristic quality of the methodologies whose coalition constitutes computational intelligence - in comparison with traditional hard computing - is their tolerance towards imprecision, uncertainty and partial truth, when such tolerance achieves better performance, higher autonomy, greater tractability, lower solution cost and better rapport with reality. Furthermore, each methodology provides complementary reasoning and searching methods to solve real-world problems. Fuzzy logic can represent qualitative knowledge and works with a robust interpolative reasoning mechanism. Neural networks are computational structures that can be trained to learn patterns from examples. Evolutionary algorithms perform randomised global search in a solution space. Probabilistic reasoning provides the capability to update outcome estimates by conditioning them with newly available evidence. Thus, desirable features lacking in one technology are present in another, and their fusion attains synergetic results. Incorporating further emerging technologies with complementary characteristics, it becomes apparent how the area accumulates computational intelligence to perform approximate reasoning – a departure from classical reasoning and modelling approaches, crisp classification and deterministic search – in solving real-world problems.

Pursuing progress from perceptions to measurement, science has achieved remarkable successes. Still, in some areas the progress has been slower and more difficult to realise. The underlying modes of reasoning in these areas are approximate rather than exact, and they require methodologies where the objects of computation are perceptions of attributes of physical and mental objects rather than their measurements. Such methodologies will enhance the ability to solve real-world problems where decision-relevant information is a mixture of measurements and perceptions. Relevant problems include automating driving in heavy traffic, translating in different languages at the level of a human interpreter, building robots that move with the agility of animals, modelling the behaviour of economic and financial systems, etc. Particularly, there is much to be desired in improving the ability to model financial decisions, and this is the focus of interest of the Technical Committee on Computational Finance, one of the eight technical committees in the IEEE Computational Intelligence Society.

Decisions in financial markets are effectively taken on the fusion of measurement-based and perception-based information. The role of perceptions is emphasised by the emerging field of behavioural finance, which is looking less as a small subfield and more like a pillar of serious finance theory. It takes into account details of human behaviour including psychology and sociology, and tests conventional theories against the impressive evidence suggesting that price levels are more than merely the sum of the available economic information. Contributing factors include the sports-style media coverage of market indices, the broadly available internet trading, and the psychological attributes of the market anchors that limit the feedback from price changes to further price changes amplifying market movements. These are some of the factors partly responsible for the developments in the end of the last century that influenced Alan Greenspan, Chairman of the Federal Reserve Board, to describe the behaviour of stock market investors as *'irrational exuberance'*.

The principle difference between measurements and perceptions is that measurements are crisp and quantitative while perceptions are fuzzy and qualitative. The area of computational intelligence provides the means to reformulate real-world problems and incorporate qualitative information. In particular, it will be possible to reformulate asset pricing and account for the perceptions of stock market investors.

1.2 Objectives

The objective of the thesis is the reformulation of the asset pricing problem incorporating imprecise information and the development of a decision-support method accounting for the perceptions of stock market investors. This is decomposed to the following aims:

- modelling imprecise factors time-series
  and solving imprecise pricing equations;
- formulating investment-risk and asset-robustness measures
  based on the modelled and processed imprecise information;
- developing a qualitative asset ranking technique
  based on the formulated risk and robustness measures;
- identifying the structure of an asset classifier
  based on the developed ranking technique
  and allowing for investors' perceptions of acceptable risk;
- identifying characteristics of the search space,
  formulating heuristics and developing a training strategy
  to evolve the parameters in that classifier structure.

Working consecutively through the aims, they are achieved by first employing fuzzy logic, then involving both fuzzy logic and neural networks, and finally the development of the soft classifier exploits the fusion of fuzzy logic, neural networks and evolutionary computation. Thus, a synergetic effect is achieved when integrating different computational intelligence techniques to provide a decision-support method to market agents.

Finally, the work on the above objective has revealed how the developed method can be incorporated into a broader problem. The imprecise models and the evolved classifier are included into a multiple-model knowledge representation of asset pricing and trading. The representation framework is constructed along several dimensions, where the perspective of imprecision plays a principal role and modelling imprecise relations involve various computational intelligence techniques. Finalising the framework is a focus for further research.

1.3 Computational Intelligence

The area of computational intelligence merges four principal components - fuzzy logic (FL), neurocomputing (NC), evolutionary computing (EC) and probabilistic reasoning (PR). We discuss them first separately and then in their fusion. Each technology contributes desirable features to approximate reasoning in solving real-world problems.

Fuzzy logic [39,49-52] is usually interpreted in a wide sense as involving four major facets –logical (FL/L), set-theoretic (FL/S), relational (FL/R) and epistemic (FL/E). The first facet is a logical system which is not truth-functional in nature and underlies inference from imprecisely defined premises [53]. The second aspect is focused on the theory of sets with unsharp bounds, and mostly related to fuzzy mathematics [54,55]. FL/R is concerned with representation and analysis of imprecise dependencies, and exploits the concepts of linguistic variables and fuzzy if-then rules applied to fuzzy system analysis and control [56]. The epistemic aspect is relevant to knowledge, meaning and imprecise information, and includes the possibility theory [57]. These four facets are overlapping and have unclear boundaries, however they all share the ability to model and process information at fuzzy granular level. Therefore, fuzzy logic is a methodology for dealing with fuzzy granularity (f-granularity). Though, FL was initially introduced as reconciling mathematical modelling and human knowledge in engineering sciences, the area of application of the methodology has extended to natural, cognitive and social sciences, involving fg-generalisation (fuzzification and fuzzy granulation) of concepts and techniques. The important direction in FL is towards computing with words and perceptions [29-33], which allows reformulation of problems in various domains where imprecision plays a key role, as is the case of decision-making in finance [58-59].

Neurocomputing [60] is concerned with processing information, which involves a learning process within an artificial neural network (ANN) architecture. The architecture responds to inputs according to a defined learning rule, and therefore has a mechanism for extracting knowledge from data. NNs are divided into feedforward networks, used in supervised mode, and recurrent networks, typically employed in unsupervised learning. Since it was proven that feedforward multilayer NNs are universal functional approximators [61], they have attracted the focus of attention. Those networks implement backpropagation training algorithms [62], and most research has focused on improving the convergence speed of the algorithms [63,64]. Depending on the scope of network characteristics involved in the training process, the learning is parametric or structural – the counterparts, respectively, of parametric estimation and system identification in classical system theory. Once trained, networks can be used to perform certain tasks depending on the application – like pattern recognition, event classification, and nonlinear-system control and identification – and we are most concerned with neurocomputing applications in finance [65-69].

Evolutionary computing [69-72] is a paradigm for randomised global search including variations as genetic algorithms (GA), genetic programming (GP), evolutionary programming (EP) and evolutionary strategies (ES). All approaches share the same generic concepts: a population of competing candidate solutions, random combination and alteration of potentially useful structures to generate new solutions, and a selection mechanism to increase the proportion of better solutions. The variations are distinguished by the genetic structures – chromosomes - that undergo adaptation and the genetic operators – crossover and mutation - that generate new candidate solutions. Implementing GAs requires to address the genetic representation of candidate solutions, the way to create an initial population, the evaluation function that describes the quality of each individual, the genetic operators that generate new variants during reproduction, and the values of parameters like population size, number of generations and probabilities of applying genetic operators [73]. GP is concerned with the automatic generation of computer programs and mostly employs a tree structure to encode them, and then breed over many generations a population of improving programs that solve particular tasks [74]. ES are distinguished by self-adaptation of additional strategy parameters, which enables them to adapt the evolutionary optimisation process to the structure of the fitness landscape [75]. EP shares a number of features with ES, however differs in generating new variants solely by means of mutation and not employing any crossover operator [76]. The boundaries between the variations of evolutionary computation are not clear, and they continuously borrow or combine features while new techniques are being developed. Promising directions in EC involve dynamic, multi-objective, and knowledge-incorporating evolutionary optimisation [77-79]. EC has been successfully applied to a remarkable variety of different domains, and we are predominantly interested in the application to problems in finance [80-82].

Probabilistic reasoning [83-85] suggests mechanisms for evaluating the outcome of systems affected by probabilistic uncertainty. The mechanisms share the common feature of performing inference while updating probability estimates through conditioning them on new available evidence. Two main currents within PR involve Bayesian belief networks (BBN) and Dempster-Shafer theory of evidence (DSTE). BBNs are concerned with propagating probability values over a network structure like trees, poly-trees or directed acyclic graphs (DAG), and considerable efforts have been directed recently towards improving the computational efficiency of propagation over general graphs. DSTE defines a mapping from basic probability assignments - masses assigned to subsets of the frame of discernment - to the computation of the lower bound (belief) and the upper bound (plausibility) of a proposition – regions defined in the same frame of discernment. Some interesting applications of probabilistic reasoning to finance are presented in [86-89].

It has been realised that the above four major technologies provide complementary characteristics in soft modelling, computing and reasoning, while attempting real-world problems. Fuzzy logic enables the translation and embedment of empirical and qualitative knowledge about a problem to be solved into reasoning systems capable of performing approximate pattern matching and interpolation. FL, however, does not have learning features as it lacks mechanisms to extract knowledge from data. On the other hand, this is the typical characteristic of neurocomputing. Still, NC may become mired in local optima, and powerful search and adaptation techniques intrinsic to evolutionary computing become desirable. Finally, probability reasoning contributes to the ability to handle various types of uncertainty and imprecision. Uncertainty in probabilism is derived from the nondeterministic membership of a point from the sample space in a well-defined region of that space. The well-defined region represents the probable event. The characteristic function of the region dichotomises the sample space. A probability value describes the tendency with which the probabilistic variable takes values inside the region. Probabilistic inference is performed through conditioning. On the other hand, uncertainty in fuzziness is derived from the partial membership of a point from the universe of discourse in an imprecisely defined region of that space. The region represents a fuzzy set. The characteristic function of the fuzzy set does not create a dichotomy in the universe of discourse. A membership value describes the degree to which the particular element of the universe of discourse satisfies the property that characterises the fuzzy set. Fuzzy reasoning is based on the extension principle. The complementarity of captured imprecision is supported by the introduction of probability measures of fuzzy events and the definition of belief functions in fuzzy events. Considering further the frequentistic and the subjective interpretations of probability, as well as the interpretation of fuzzy membership as possibility, similarity, desirability or preference, enriched is the scope of handled imprecision. Therefore, the fusion of different computational intelligence approaches will result in an effective approximate-reasoning methodology that explores the don't-know, don't-need, can't-solve, and can't-define rationales. The don't-know rationale applies when the values of variables or parameters are not known sufficiently precise to justify using conventional modelling techniques. The don't-need motivation presents situations where exploiting the inherent tolerance for imprecision achieves tractability, robustness or low solution cost. The can't-solve reason reveals problems that cannot be solved through quantitative modelling and computing. The can't-define principle relates to concepts that are too complex to allow definition through a set of numerical criteria.

A body of literature is growing on the hybrid implementation of computational intelligence techniques. The interaction between fuzzy logic and neurocomputing results in neuro fuzzy systems (NFS) or fuzzy neural networks (FNN), depending on the dominant component. The work in the field relates to approximations between fuzzy systems and neural nets [90,91], building hybrid NNs to equal fuzzy systems [92,93], FL controlling parameters in NC [94] or NC tuning FL [95-97], using neurocompiting to solve fuzzy problems [44,46,98], investigating the approximating capabilities of fuzzy NNs [99-103], and constructing and implementing hybrid fuzzy neural networks (HFNN) [93,104]. Further, two major approaches in integrating of fuzzy logic and evolutionary computation include fuzzy evolutionary algorithms (FEA) and evolutionary fuzzy systems (EFS). An FEA uses FL to improve its performance through controlling parameters as mutation and crossover rates or population size [105-107], or taking advantage of the tolerance for imprecision and saving computational resources through fuzzifying those operators [108,109]. An EFS is a FS augmented with an evolutionary tuning process, and the most extended class of EFS corresponds to evolutionary fuzzy rule-based systems (EFRBS) [110,111]. Next, the integration of EC and NC employs evolution as another form of adaptation in addition to learning. Evolutionary algorithms perform various tasks, such as training weights, designing architectures [112,113], adapting learning rules, selecting input features, initialising weights, etc. A general framework for evolutionary neural networks (ENN) is introduced in [47] suggesting three levels of evolution that concern weights, learning rules and architectures, correspondingly. Each inner level of evolution is included in the next outer level, if such exist in the problem, then the lower the level, the faster the time scale of the evolution. Further, merging evolutionary computing and probabilistic reasoning [114] has been used in evolving the optimal structure of Bayesian networks [115], as well as in modelling EC with BBNs and producing Bayesian optimisation algorithms (BOA) [116,117]. The combination of EC, NC and FL typically involves augmenting regular or hybrid feedforward multilayer FNN with evolutionary learning capabilities into evolutionary fuzzy neural networks (EFNN). EFNNs include fuzzy connection weights, or fuzzy operations in the nodes of the network, or fuzzy nodes that represent membership functions, and the learning process implements evolutionary techniques to achieve coarse-granularity followed by backpropagation for fine-granularity search, or to obtain the weights of the network, or to adapt the transfer functions of the nodes, or to optimise the topology of the net [118-122]. The fusion of all computational intelligence techniques has created the area of computational intelligence which is gathering recognition as a driving engine for artificial intelligence [33,123-129].

As a result, computational intelligence has emerged as a methodological paradigm for representing, incorporating and processing uncertain, incomplete, imprecise or perception-based information at granular level. The consequence is the ability to reformulate the representation of real-world problems and the techniques for their solution, as well as the competence to build in these developments into autonomous or decision-support intelligent systems. Financial markets constitute one of the areas that will directly benefit from the methodological advances, and exploit the rationales of don't-know, don't-need, can't-solve, and can't-define. Relevant problems involve forecasting market movements, volatility modelling, asset and derivatives pricing, developing investment strategies for asset and derivatives traders and hedgers, mastering market timing and switching in and out of various securities classes, portfolio management, risk analysis and management, trading systems design, and agent-based modelling and simulation of artificial stock markets [130-134].

1.4 Thesis Outline

Relevant computational intelligence techniques, as described in section 1.3, are employed throughout the thesis in the following sequence. Chapter 2 applies fuzzy logic in its broad sense, and specifically fuzzy mathematics, to evaluate the fuzzy asset price. The epistemic aspect is also concerned while modelling the imprecise pricing factors. Chapter 3 explores two measures of the information encoded into the fuzzy asset evaluation. Chapter 4 still works with the results of implementing fuzzy logic to the problem, and builds an asset ranking techniques on the basis of the measures of risk and robustness from Chapter 3. Chapter 5 elaborates on putting the rationale behind the ranking technique into more practical use - building an asset classifier. This involves a further computational intelligence technique, neural networks, or rather fuzzy neural networks. Chapter 6 is focused on the risk module of the classifier, therefore the risk classifier, and a two-level exploratory algorithm is developed as its training strategy. Thus a further element of the computational intelligence paradigm is employed, and the technique builds up as evolutionary fuzzy neural computing. Chapter 7 presents the empirical results of training, validating, and predicting with the risk classifier. Finally, Chapter 8 involves the intermediate products of the approach – the fuzzy asset evaluation procedure, the ranking technique, and the soft risk classifier – into the knowledge representation module of an intelligent system in asset risk analysis. The system further applies the computational theory of perceptions – another component in the computational intelligence paradigm – in user analysis with the objective to improve the efficiency of decision support or the quality of tutoring in the domain of asset evaluation and risk analysis.

Chapter 2: Fuzzy and Possibilistic Asset Pricing

2.1 Introduction

We focus on the price dividend relation, introducing fuzzy intervals or possibilistic variables to model uncertain asset prices, dividend yields and interest rates. However, the approach can be applied to any of the pricing techniques, while involving an alternative modelling of the imprecision in the corresponding pricing factors.

Using fuzzy intervals allows one to take into account a broader range of imprecision, beyond the probability type of uncertainty. Furthermore, substituting possibilistic distributions for the fuzzy intervals, and applying multilevel interval calculus, the asset price is evaluated at various possibility levels. The possibility levels or the degrees of membership corresponding to the $\alpha - cuts$ of the evaluated asset price match those related to the $\alpha - cuts$ of the uncertain factors. This feature of the solution is beneficial as giving an idea of the levels of uncertainty a market agent could accordingly choose or prefer to work at. Each level involves some of the modelled imprecision. Thus if we attempt to represent the broader range of imprecision the market could possibly suffer, then an agent may choose the level of uncertainty within that range corresponding to his preferences. That level however delimits the degree with which the evaluated interval of asset prices will belong to the true price. On the other hand, one may use the possibilistic solution to compare the intervals corresponding to different levels of uncertainty.

2.2 Modelling Uncertainty with Fuzzy Intervals

The analysis is focused on common shares and the asset rate of return is described with

$$R = \frac{P_{t+1}\left(DY_{t+1}+1\right)}{P_t} - 1 \ , \tag{2.1}$$

where P_t denotes the ex-dividend share price at the end of period t and DY_{t+1} is the next-period dividend yield. The equation may be solved backward or forward for the asset price at t. Then, an evaluation of the current price at $t=0$ will involve weighted past and future prices,

$$P_0 = \sum_{t=1}^{T_1} A_t P_{-t} + \sum_{t=1}^{T_2} B_t P_t \ , \tag{2.2}$$

where A_t and B_t are the parameters of the model. Past prices are known, however may be considered as chance realisation of a highly volatile process. Future prices may be estimated, however the estimates are only reliable to some degree. Fuzzy intervals may be used in modelling both past and future prices, and the evaluation of the current price will be based on the weighted fuzzy values. A fuzzy interval is a fuzzy set in the real line $\mathbb{R}$, whose level-cuts are intervals.

As an illustration, we will solve the equation (2.1) forward. The intuition is that a fair value for the current price is the present value of all expected proceeds on the asset in the future [135].

$$P_0 = \sum_{t=1}^{\infty} \frac{P_t DY_t}{(1+R)^t} \tag{2.3}$$

However, an evaluation is only produced over a limited time horizon and the estimated proceeds are approximate. Therefore,

$$P_0 = \sum_{t=1}^{T} \frac{P_t DY_t}{(1+R)^t} + \frac{P_T}{(1+R)^T} \quad , \tag{2.4}$$

where the fuzzy intervals $\tilde{P}_t$, $\tilde{D}Y_t$ and $\tilde{R}$ will be substituted for the uncertain P_t, DY_t and R, correspondingly.

$$\tilde{P}_0 = \sum_{t=1}^{T} \frac{\tilde{P}_t \tilde{D}Y_t}{\left(1+\tilde{R}\right)^t} + \frac{\tilde{P}_T}{\left(1+\tilde{R}\right)^T} \quad , \tag{2.5}$$

An alternative view on the formula (2.5), in line with the model (2.2), will describe the evaluated current asset price as based on the fuzzified variables P_t, $1 \leq t \leq T$, and the fuzzified parameters B_t resulting from DY_t and R with no further weighting, while the parameters $A_t = 0$. Thus the fuzzy model involves both fuzzified variables and parameters, and one may argue that the current asset price is evaluated while approaching the broader imprecision the market could possibly suffer, well beyond the probability type of uncertainty.

Let the notations

$$\mu\left(x_{P_t}\left|\tilde{P}_t\right.\right)=\left(P_{tb}/P_{tv}/P_{te}\right) \ , \tag{2.6.1}$$

$$\mu\left(x_{DY_t}\left|\tilde{DY}_t\right.\right)=\left(DY_{tb}/DY_{tv}/DY_{te}\right) \ , \tag{2.6.2}$$

$$\mu\left(x_R\left|\tilde{R}\right.\right)=\left(R_b/R_v/R_e\right) \ , \tag{2.6.3}$$

stand for the membership functions of $\tilde{P}_t$, $\tilde{DY}_t$ and $\tilde{R}$, respectively. The graph of the membership function $\mu\left(x_{P_t}\left|\tilde{P}_t\right.\right)$ is a triangle with a base on the interval $\left[P_{tb}, P_{te}\right]$ and a vertex at the point $x_{P_t} = P_{tv}$, where $0 < P_{tb} < P_{tv} < P_{te}$. Analogous descriptions apply for the functions $\mu\left(x_{DY_t}\left|\tilde{DY}_t\right.\right)$ and $\mu\left(x_R\left|\tilde{R}\right.\right)$. The weak $\alpha - cut$ of a fuzzy interval [52] is defined for $\tilde{P}_t$ as

$$\tilde{P}_t(\alpha)=\left[\underline{P_t(\alpha)}\ ,\overline{P_t(\alpha)}\right]=\begin{cases}\left[P_{tb}, P_{te}\right] \ , & \alpha = 0 \\ \left\{x_{P_t}\left|\mu\left(x_{P_t}\left|\tilde{P}_t\right.\right)\geq\alpha\right.\right\} \ , & 0<\alpha\leq 1\end{cases} \ , \tag{2.7}$$

and the $\alpha - cuts$ $\tilde{DY}_t(\alpha)$ and $\tilde{R}(\alpha)$ are specified in a similar way.

Considering the character of the crisp function (2.4), the domain area of the arguments and the choice of their membership functions, we can derive the following conclusions and apply the calculi of fuzzy intervals [39,52]. The arguments $\tilde{P}_t$, $\tilde{DY}_t$ and $\tilde{R}$ are fuzzy intervals in the positive real line $\mathbb{R}^+$. In this domain area, the crisp function P_0 is continuous on all its arguments, while monotonically increasing on P_t and DY_t, and monotonically decreasing on R. Therefore, the conditions in the following proposition are satisfied.

Proposition 2.1: *If a function has a finite number of arguments and is continuous and monotonic with respect to each of them, then it commutes with level-cutting when fuzzy intervals are substituted for the arguments.* [39]

Thus, the evaluated price will be also a fuzzy interval $\tilde{P}_0$, whose $\alpha-cuts$ $\tilde{P}_0(\alpha)$ are calculated from

$$\tilde{P}_0(\alpha)=\left[\underline{P_0(\alpha)},\overline{P_0(\alpha)}\right],\ 0\leq\alpha\leq 1\ , \tag{2.8}$$

$$\underline{P_0(\alpha)} = \sum_{t=1}^{T}\frac{\underline{P_t(\alpha)}\ \underline{DY_t(\alpha)}}{\left(1+\overline{R(\alpha)}\right)^t} + \frac{\underline{P_T(\alpha)}}{\left(1+\overline{R(\alpha)}\right)^T}\ ,$$

$$\overline{P_0(\alpha)} = \sum_{t=1}^{T}\frac{\overline{P_t(\alpha)}\ \overline{DY_t(\alpha)}}{\left(1+\underline{R(\alpha)}\right)^t} + \frac{\overline{P_T(\alpha)}}{\left(1+\underline{R(\alpha)}\right)^T}\ ,$$

and whose membership function $\mu\left(x_{P_0}\middle|\tilde{P}_0\right)$ is defined as

$$\mu\left(x_{P_0}\middle|\tilde{P}_0\right)=sup\left\{\alpha\middle|x_{P_0}\in\tilde{P}_0(\alpha),0\leq\alpha\leq 1\right\}\ . \tag{2.9}$$

In conclusion, using fuzzy intervals one is able to evaluate the asset price while involving various types of uncertainty. The approach is not restricted to the price-dividend function, and one may consider other pricing formulas. In each case, reasoning on the character of the function, the choice of arguments and their domain area, as well as the appropriate fuzzification, will decide on the features of the corresponding solution.

2.3 Modelling Uncertainty with Possibility Distributions

From the theory of fuzzy sets and the possibility theory, it is known that the membership function of a fuzzy set may be interpreted as the possibility distribution of a fuzzy variable. This conclusion is used in [35,136-138] to introduce alternative uncertainty modelling. We will apply here the general result from [35,136-138] to provide an interpretation of the approach we introduced in section 2.3 and to analyse further the obtained solution. This is described in our publications [P1,P2].

Let substitute possibilistic variables $\breve{P}_t$, $\breve{DY}_t$ and $\breve{R}$ for the arguments in the function (2.4). Again, the possibility of an event differs from its probability, and allows modelling a wider range of imprecision. A possibility distribution is less restrictive than a probability distribution. The possibility that the variable $\breve{P}_t$ will take a specific value may be further interpreted as a measure of the belief that the value will happen. Therefore, values whose possibility is at least π where $0 < \pi \leq 1$, are the values whose level of uncertainty is at most $\upsilon = 1 - \pi$. Thus, if a possibility distribution models some considered range of imprecision, then υ is a level of uncertainty within that range. For example, the level $\upsilon = 1$ covers the whole range, while the level $\upsilon = 0.9$ only accounts for 90% of the range.

Evaluating the possibility distribution of a function requires producing the joint distribution of the arguments which may involve interactivity. In the general case, the analytical solution is computationally infinite. However, a good approximation is found in the fuzzy interval solution. With the appropriate choice of possibility distributions for the arguments, the fuzzy interval evaluation of the function serves as an upper boundary for its possibility distribution evaluation, in the sense that the former at least includes the latter. In the case of the pricing function (2.4), and the choice of calibration technique described in the next section, the two solutions will be equal.

To provide that the evaluation of

$$\breve{P}_0 = \sum_{t=1}^{T} \frac{\breve{P}_t \breve{D}Y_t}{\left(1+\breve{R}\right)^t} + \frac{\breve{P}_T}{\left(1+\breve{R}\right)^T} \tag{2.10}$$

is equal to that in the formula (2.5), we choose the possibility distributions for the arguments as follows. The possibility that the variable $\breve{P}_t$ takes some value will be equal to the degree of membership of this value to the fuzzy interval $\tilde{P}_t$,

$$Poss\left[\breve{P}_t = x_{P_t}\right] = \mu\left(x_{P_t} \middle| \tilde{P}_t\right),\ 1 \le t \le T \ . \tag{2.11.1}$$

By analogy, the distributions of $\breve{D}Y_t$ and $\breve{R}$ are described with

$$Poss\left[\breve{D}Y_t = x_{DY_t}\right] = \mu\left(x_{DY_t} \middle| \tilde{D}Y_t\right),\ 1 \le t \le T \ , \tag{2.11.2}$$

$$Poss\left[\breve{R} = x_R\right] = \mu\left(x_R \middle| \tilde{R}\right) \ . \tag{2.11.3}$$

As a result of this choice, the distribution of the current asset price $\breve{P}_0$ will be

$$Poss\left[\breve{P}_0 = x_{P_0}\right] = \mu\left(x_{P_0} \middle| \tilde{P}_0\right) \ , \tag{2.12}$$

where the membership function $\mu\left(x_{P_0} \middle| \tilde{P}_0\right)$ is evaluated from (2.9).

Now we can benefit from the computability of the fuzzy interval solution and the logic behind the possibilistic reasoning. Let us consider again all the values of the variable $\breve{P}_t$ whose possibility to happen is at least π, or as described earlier, whose level of uncertainty is at most $\upsilon = 1 - \pi$. From the definitions (2.7) and (2.11), it follows that those values are presented with the $\alpha - cut$ of the fuzzy interval $\tilde{P}_t$ for $\alpha = \pi$,

$$\left\{x_{P_t} \middle| Poss\left[\breve{P}_t = x_{P_t}\right] \geq \alpha\right\} = \tilde{P}_t(\alpha),\ 0 \leq \alpha \leq 1,\ 1 \leq t \leq T \ . \tag{2.13.1}$$

Correspondingly, for the rest of the arguments, the values at the level of uncertainty $\upsilon = 1 - \alpha$ are described with

$$\left\{x_{DY_t} \middle| Poss\left[\breve{DY}_t = x_{DY_t}\right] \geq \alpha\right\} = \tilde{DY}_t(\alpha),\ 0 \leq \alpha \leq 1,\ 1 \leq t \leq T \ , \tag{2.13.2}$$

$$\left\{x_R \middle| Poss\left[\breve{R} = x_R\right] \geq \alpha\right\} = \tilde{R}(\alpha),\ 0 \leq \alpha \leq 1 \ . \tag{2.13.3}$$

This will also apply for the evaluated asset price,

$$\left\{x_{P_0} \middle| Poss\left[\breve{P}_0 = x_{P_0}\right] \geq \alpha\right\} = \tilde{P}_0(\alpha) = \left[\underline{P_0(\alpha)}\ ,\overline{P_0(\alpha)}\right],\ 0 \leq \alpha \leq 1 \ . \tag{2.14}$$

Furthermore, the computation of the fuzzy interval solution with the formula (2.8) produces the level-cut interval of the function $\left[\underline{P_0(\alpha)}\ ,\overline{P_0(\alpha)}\right]$ for some α, as based on the level-cut intervals of the arguments $\tilde{P}_t(\alpha)$, $\tilde{DY}_t(\alpha)$, $\tilde{R}(\alpha)$ at the same level α. Therefore, we can formulate the following definition, as described in our publication [P2] and subsequently apply the relevant line of reasoning.

Definition 2.1: Factor and asset imprecision

All the values of all arguments in the pricing formula that are assigned the same level of imprecision produce or are responsible for all the values of the evaluated asset price at the same level of imprecision. [P2]

2.4 Empirical Results

The emphasis in this thesis is the analysis of imprecision, while providing for various types of uncertainty without concentrating on a specific type or source. To illustrate the method, we could use unreliable predictions of the arguments in the pricing formula (2.5) and based on them evaluate the current price. Alternatively, we could move the evaluation point back in time, and use historic volatile data for the arguments. We settled for the latter.

Datastream data are employed from 1975 to 2000, on share prices and dividend yields for thirty-five UK companies traded on the London Stock Exchange (LSE), as listed in Table 2.1. The companies come from various sectors and have a diverse market capitalisation. The selection was initially based on a favourable price-to-book value. Furthermore, formula (2.5) is applied to evaluating any company's share price, and the argument $\tilde{R}$ acts as a discounting factor for the proceeds from investing in those shares. To model $\tilde{R}$, we use the three-month UK treasury bill rate approximating the risk free rate of return.

Table 2.1: List of companies

company	*sector*	*market capitalisation*, £m*
BASS	Restaurants, Pubs & Breweries	7,616.94
BBA GROUP	Engineering – General	2,186.50
BENTALLS	Retailers - Multi Department	26.79
BLUE CIRCLE INDUSTRIES	Building & Construction Materials	2,912.45
BOC GROUP	Chemicals – Commodity	6,530.61
BOOTS CO.	Retailers - Multi Department	5,445.32
BP AMOCO	Oil – Integrated	121,241.85
BRITISH AMERICAN TOBACCO	Tobacco	7,654.29
BUNZL	Business Support Services	1,553.66
COATS VIYELLA	Other Textiles & Leather Goods	292.00
DIXONS GROUP	Retailers – Hardlines	7,103.14
GOODWIN	Engineering Fabricators	5.65
GREAT UNIVERSAL STORES	Retailers - Multi Department	3,641.28
HANSON	Building & Construction Materials	3,383.77
INCHCAPE	Vehicle Distribution	242.97
LEX SERVICE	Vehicle Distribution	436.58
MARKS & SPENCER	Retailers - Multi Department	8,464.95
NORTHERN FOODS	Food Processors	609.23
PILKINGTON	Building & Construction Materials	926.97
RANK GROUP	Leisure Facilities	1,516.28
RMC GROUP	Building & Construction Materials	2,218.21
SAINSBURY (J)	Food & Drug Retailers	6,707.54
SCOTTISH & NEWCASTLE	Restaurants, Pubs & Breweries	2,684.77
SMITH (WH) GROUP	Retailers - Soft Goods	1,200.80
SMITHS INDUSTRIES	Aerospace	2,912.40
TARMAC	Building & Construction Materials	1,126.40
TATE & LYLE	Food Processors	1,820.48
TAYLOR WOODROW	Other Construction	510.63
TI GROUP	Engineering – General	2,401.66
TRANSPORT DEVELOPMENT GROUP	Rail, Road & Freight	161.18
UNILEVER	Food Processors	13,261.69
UNITED BISCUITS HOLDINGS	Food Processors	1,255.64
WHITBREAD	Restaurants, Pubs & Breweries	3,102.43
WIMPEY (GEORGE)	House Building	415.45
WOLSELEY	Builders Merchants	2,725.25

*Market Capitalisation in 2000.

Then the crisp data are transformed into fuzzy intervals, using the following calibration procedure for the membership functions. The support of the membership function of $\tilde{P}_t$ at a particular t, that is the $\alpha-cut$ $\tilde{P}_t(\alpha)$ for $\alpha=0$, is chosen as *2.5%* wider than the *99%* normal-distribution confidence interval of the crisp P_t over the whole horizon of the pricing factors $1 \le t \le T$.

$$\tilde{P}_t(0)=\left[\underline{P_t(0)}\ ,\overline{P_t(0)}\right]\ , \tag{2.15.1}$$

$$\underline{P_t(0)}=P_t-1.025\bullet 2.576\bullet\sqrt{\sum_{\tau=1}^{T}\left(P_\tau-\sum_{\tau=1}^{T}P_\tau/T\right)^2\Big/(T+1)},\ 1\leq t\leq T\ ,$$

$$\overline{P_t(0)}=P_t+1.025\bullet 2.576\bullet\sqrt{\sum_{\tau=1}^{T}\left(P_\tau-\sum_{\tau=1}^{T}P_\tau/T\right)^2\Big/(T+1)},\ 1\leq t\leq T\ .$$

This permits for a broader range of possible values to be considered when modelling uncertainty than the range of probable values. Further, the vertex of the membership function, that is the value with a membership level of 1, is selected equal to the crisp value,

$$\tilde{P}_t(1)=P_t,\ 1\leq t\leq T\ . \tag{2.15.2}$$

Finally, we avoid non-positive values of rates, yields and prices, e.g. in case in the calibrated membership function $\underline{P_t(0)}\leq 0$, then $\underline{P_t(0)}=\varepsilon>0$ is chosen. The calibration procedure is repeated for $\tilde{D}Y_t,\ 1\leq t\leq T$, and for $\tilde{R}$ where $\tilde{R}(1)=\sum_{t=1}^{T}R_t\Big/T$.

To illustrate the imprecision, modelled within each company's data, we will present the fuzzy trajectory of its dividend per share over the investment horizon, $\tilde{D}$. It is calculated as the following Cartesian product,

$$\tilde{D}(\alpha)=\prod_{t=1}^{T}\left(\tilde{P}_t(\alpha)\tilde{D}Y_t(\alpha)\right),\ 0\le\alpha\le 1\ . \tag{2.16}$$

Alongside the fuzzy trajectory of the arguments, the graphics below also present the evaluated with the pricing formula fuzzy interval for the asset price at $t=0$, January 1975. We include here the graphics for the first ten companies, and provide the complete set in Appendix A1.

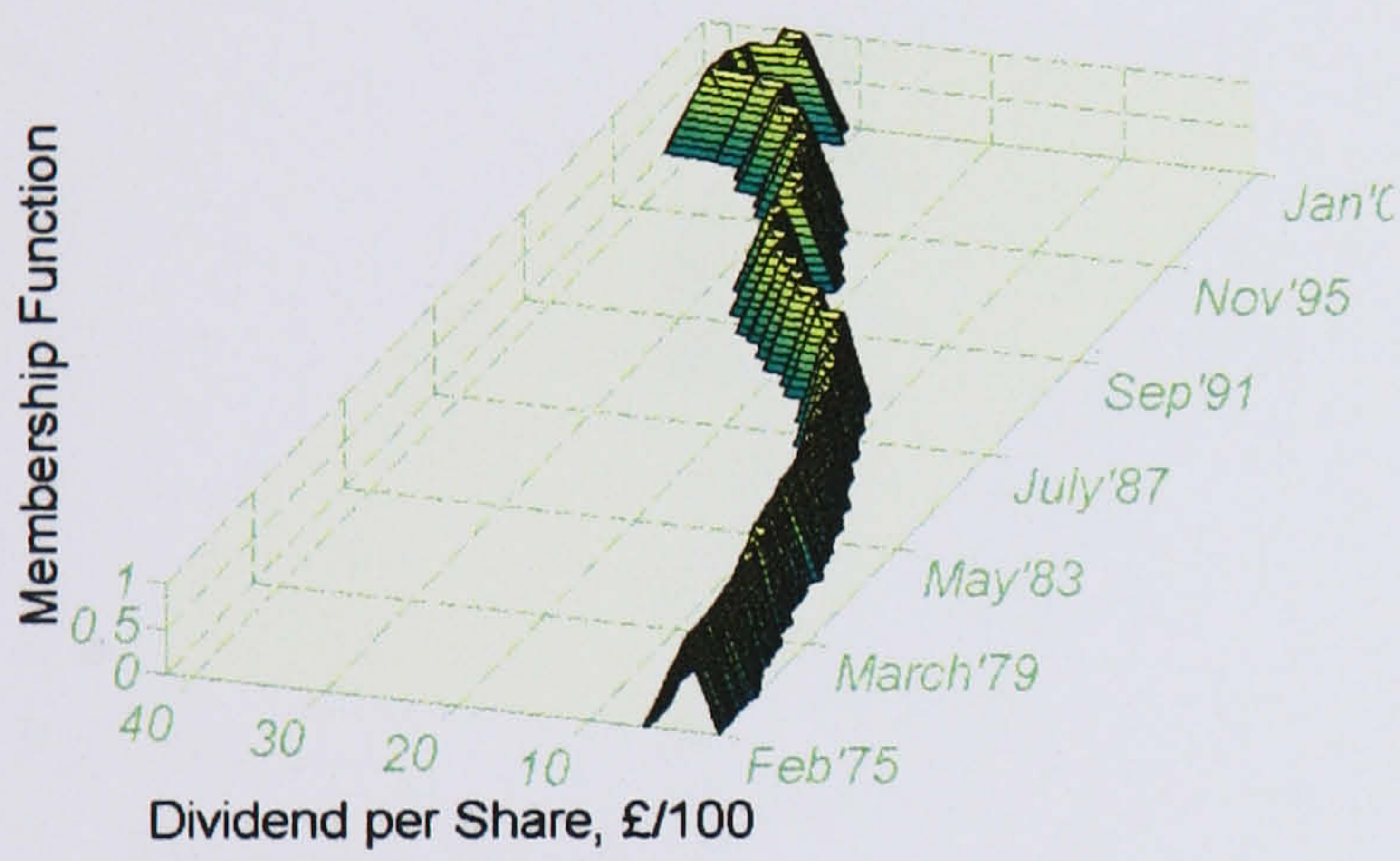


Figure 2.1a: BASS - fuzzified data

Figure 2.1b: BASS - evaluated fuzzy share price

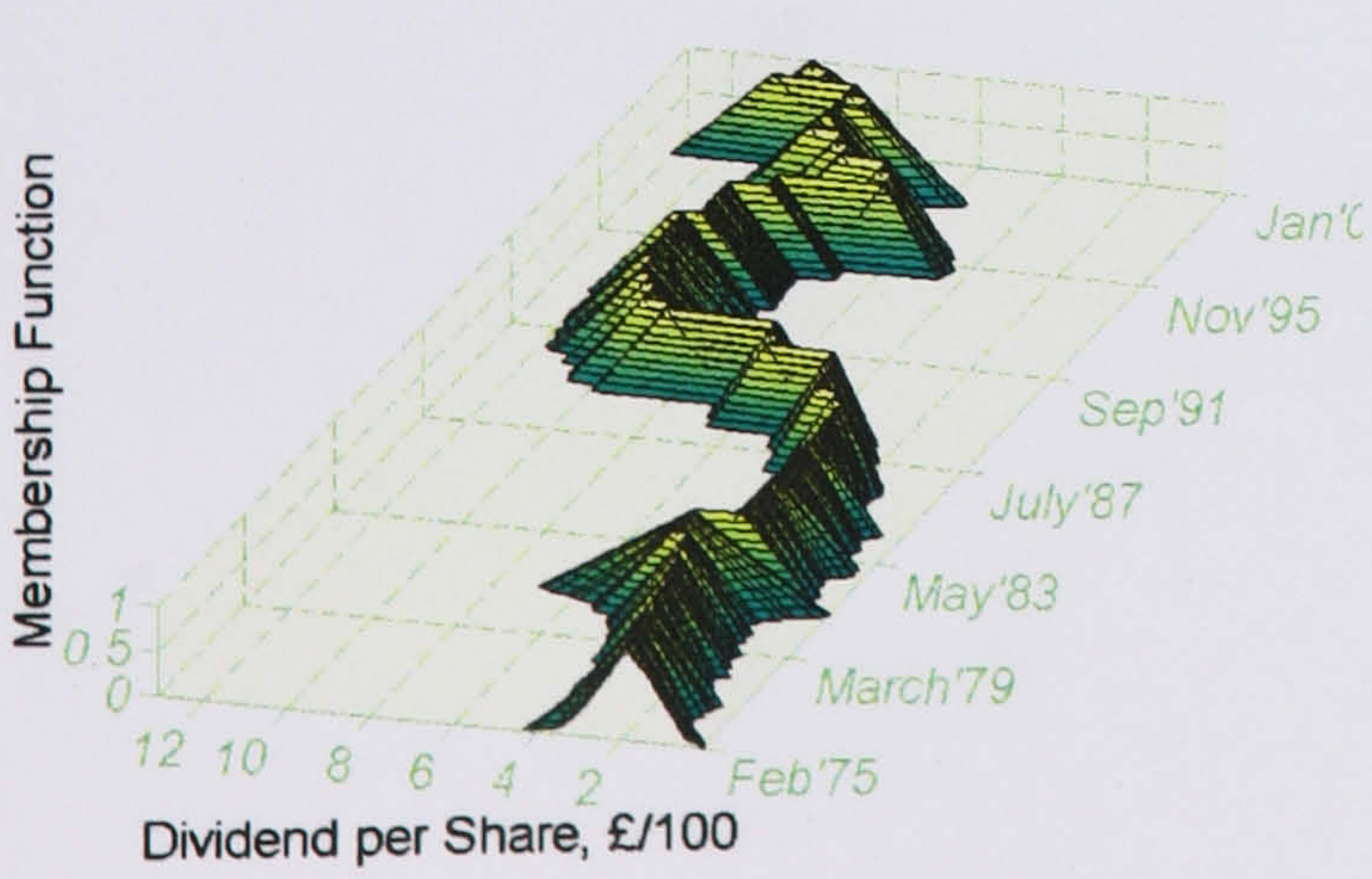


Figure 2.2a: BBA GROUP - fuzzified data

Figure 2.2b: BBA GROUP - evaluated fuzzy share price

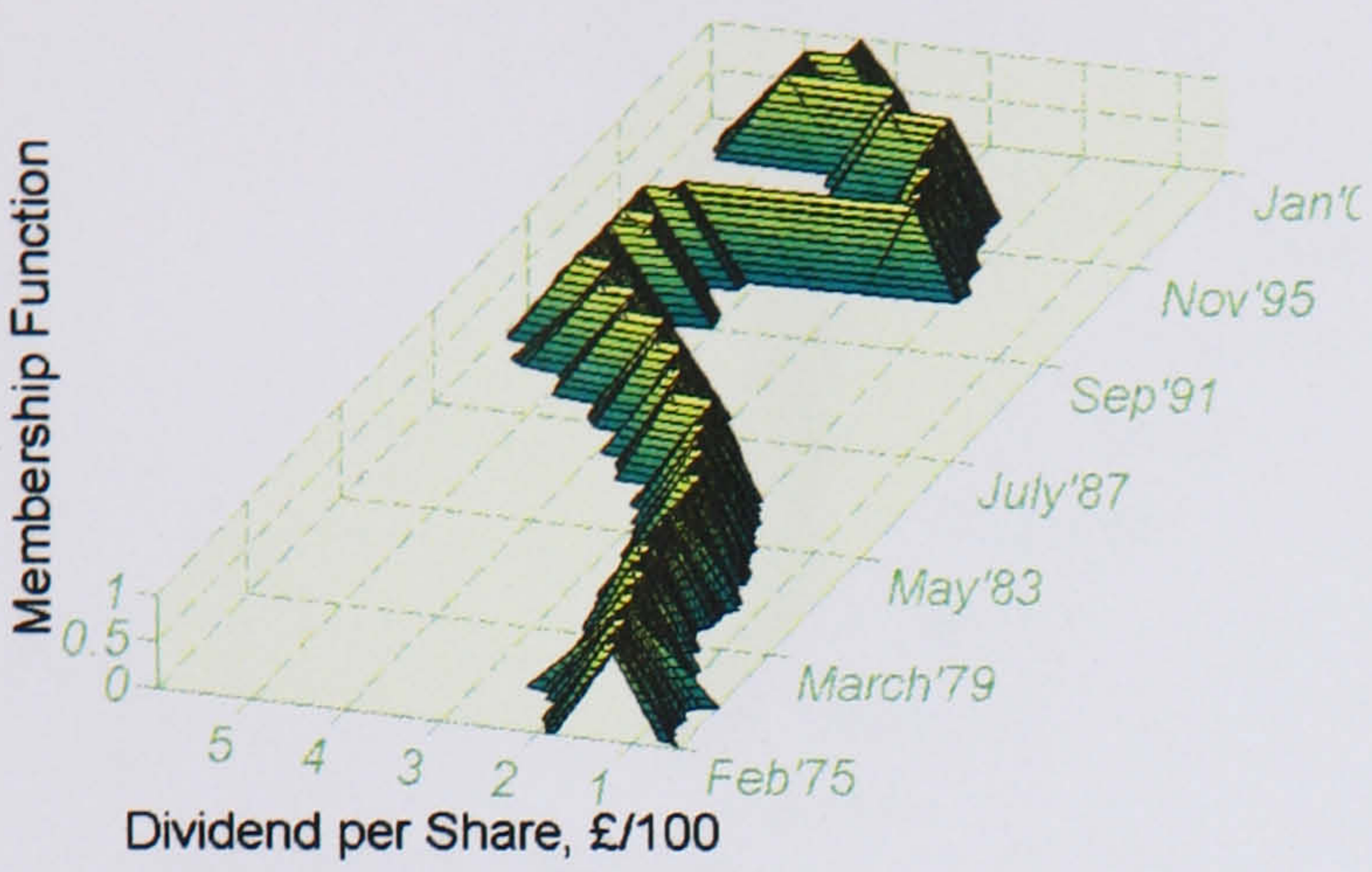


Figure 2.3a: BENTALS - fuzzified data

Figure 2.3b: BENTALS – evaluated fuzzy share price

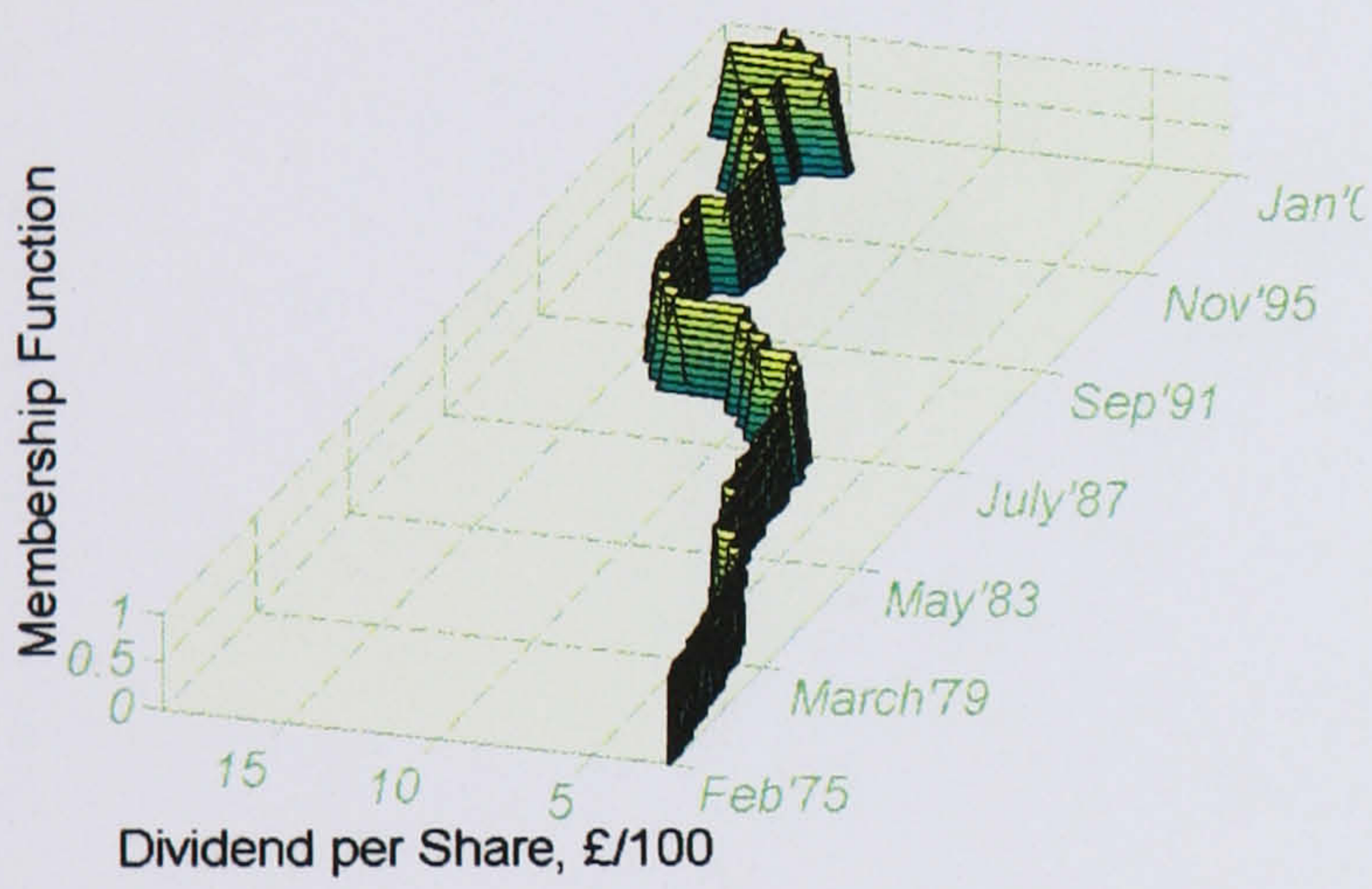


Figure 2.4a: BLUE CIRCLE INDUSTRIES - fuzzified data

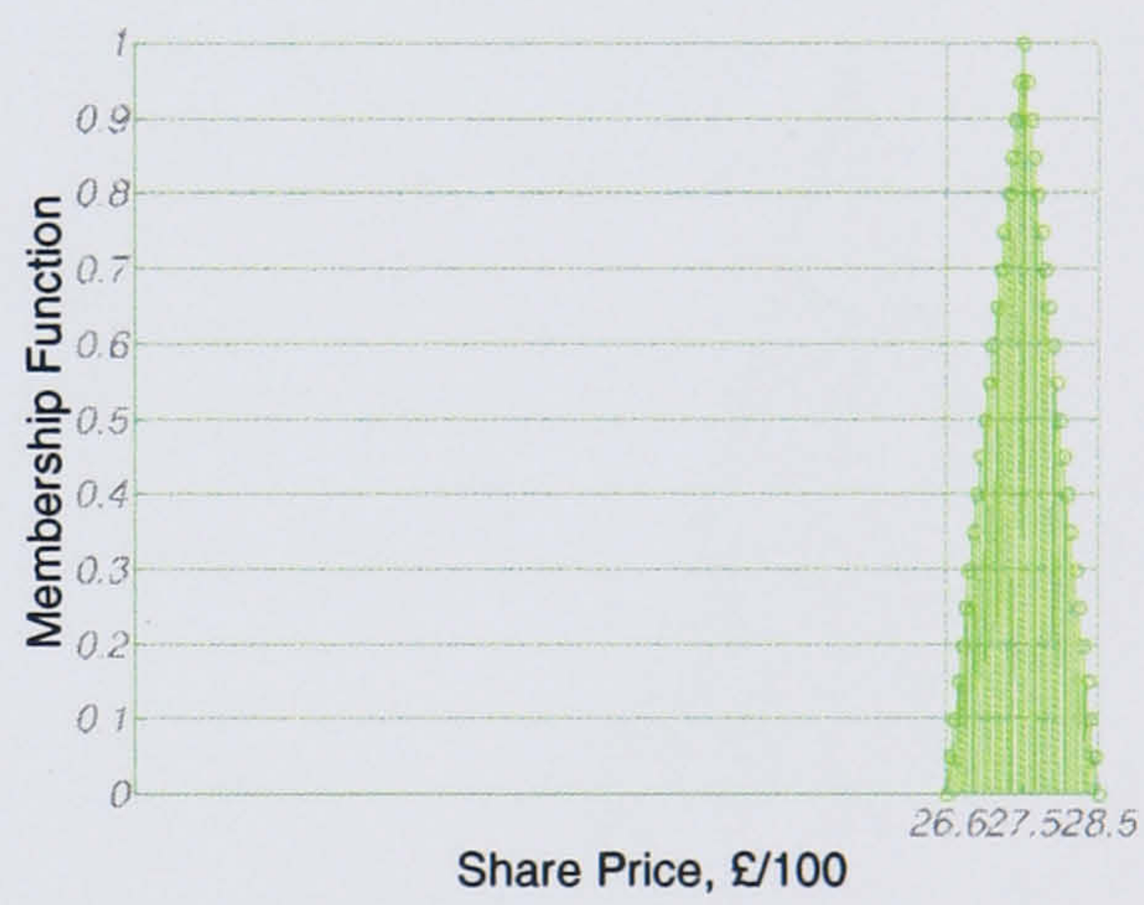


Figure 2.4b: BLUE CIRCLE INDUSTRIES - evaluated fuzzy share price

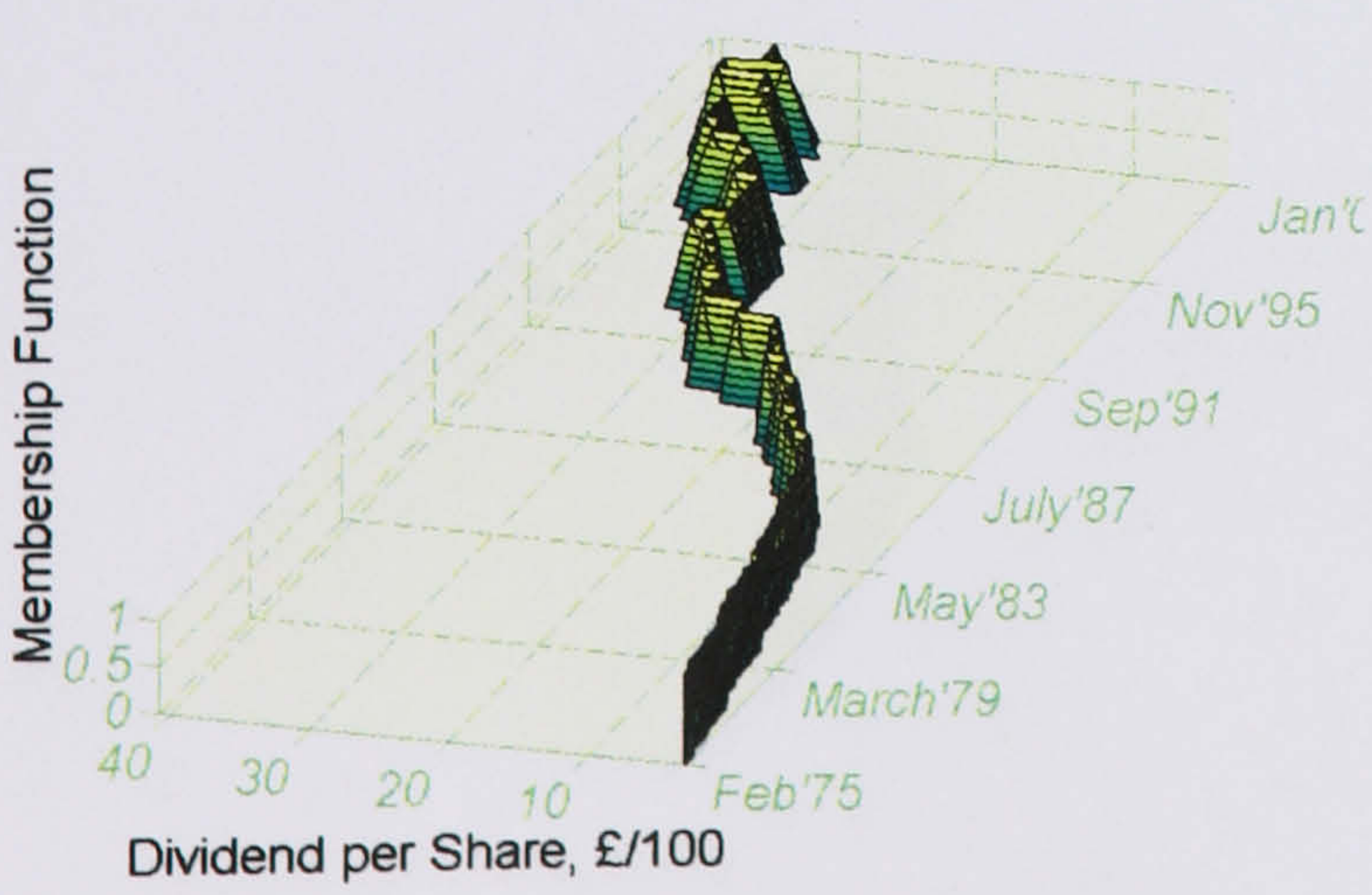


Figure 2.5a: BOC GROUP - fuzzified data

Figure 2.5b: BOC GROUP - evaluated fuzzy share price

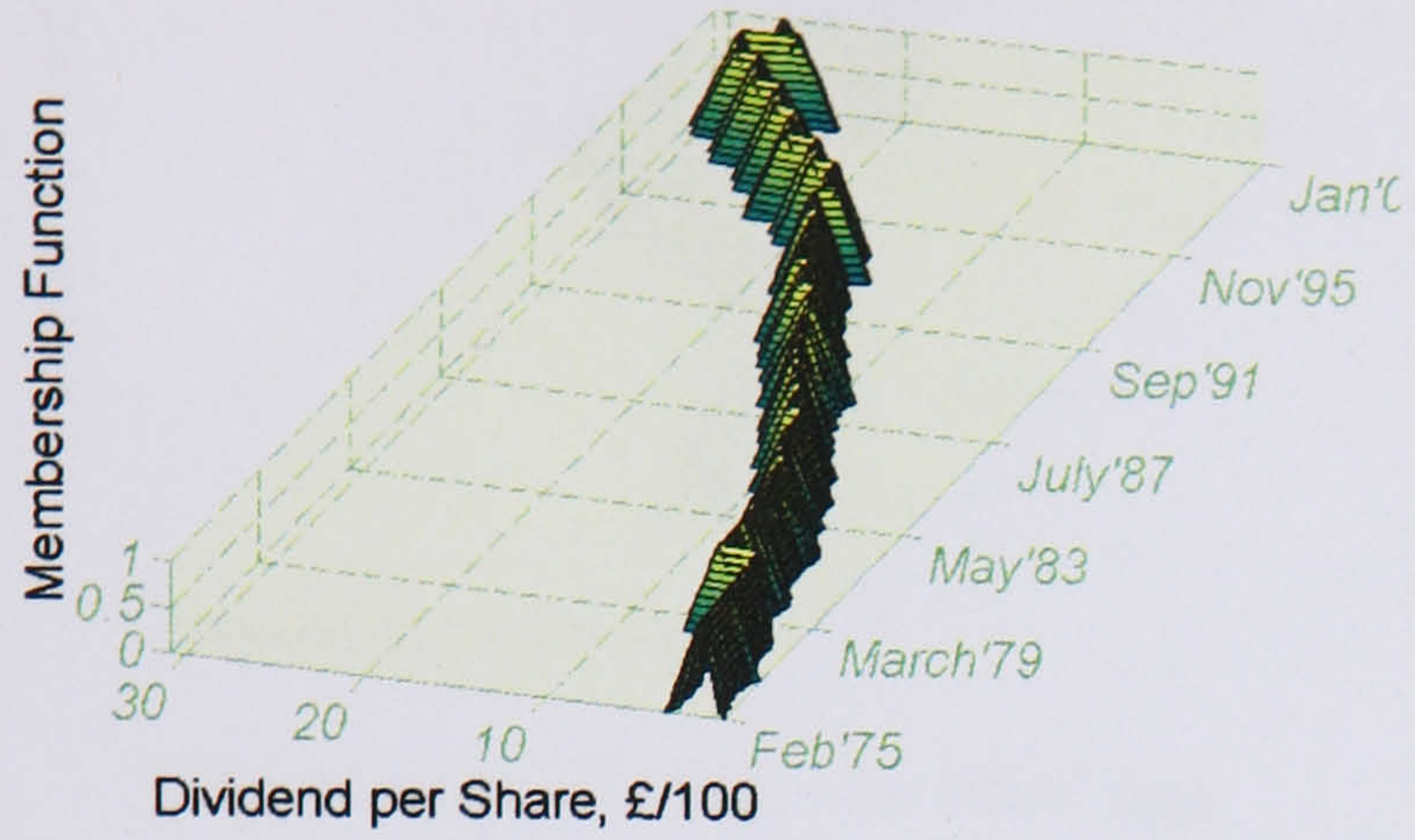


Figure 2.6a: BOOTS CO. - fuzzified data

Figure 2.6b: BOOTS CO. - evaluated fuzzy share price

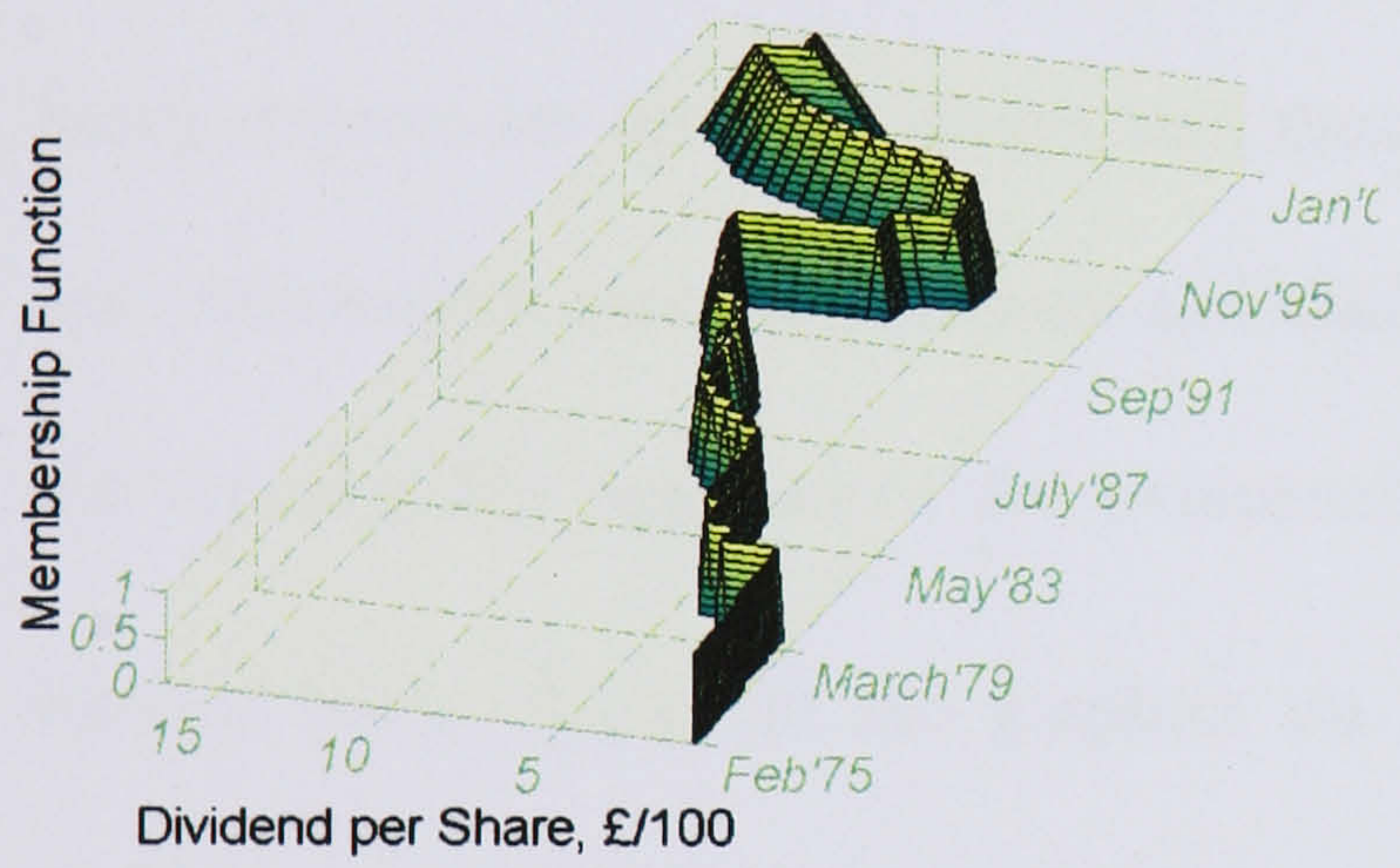


Figure 2.7a: BP AMOCO - fuzzified data

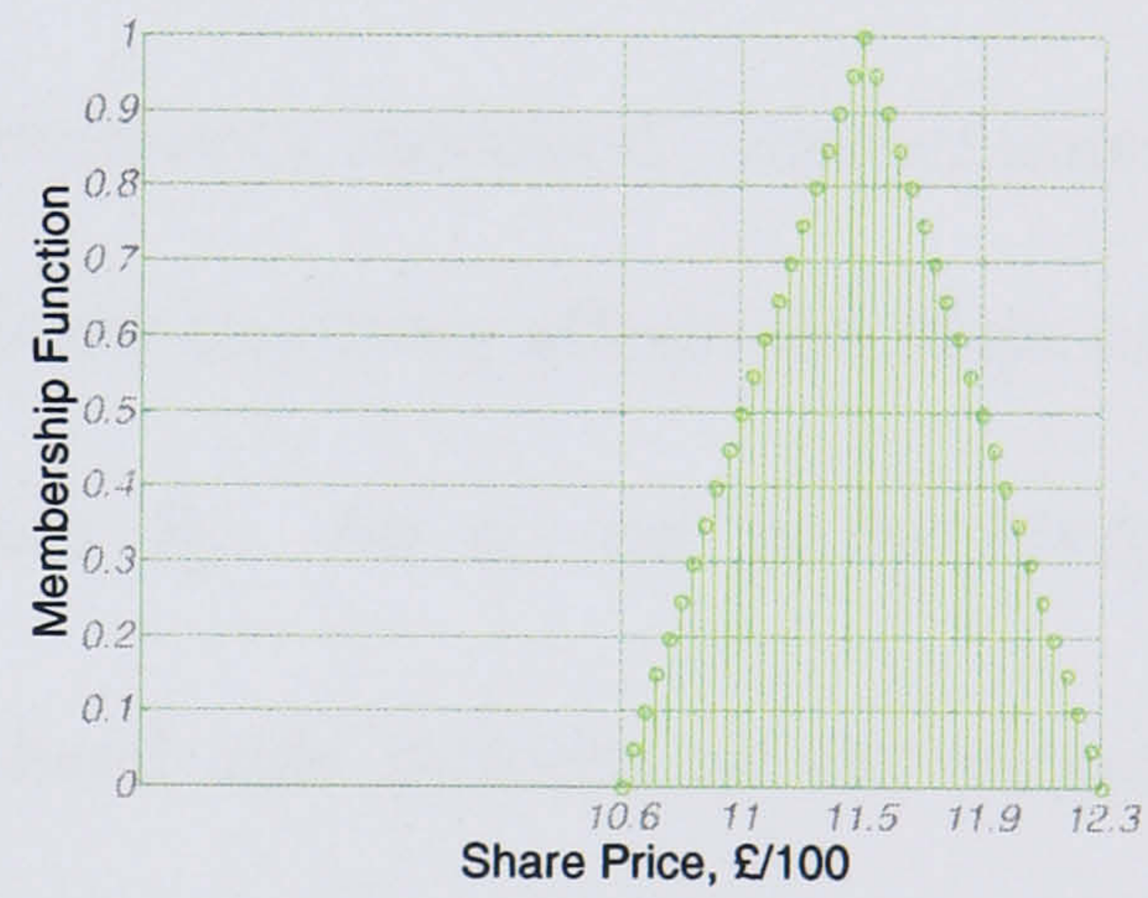


Figure 2.7b: BP AMOCO - evaluated fuzzy share price

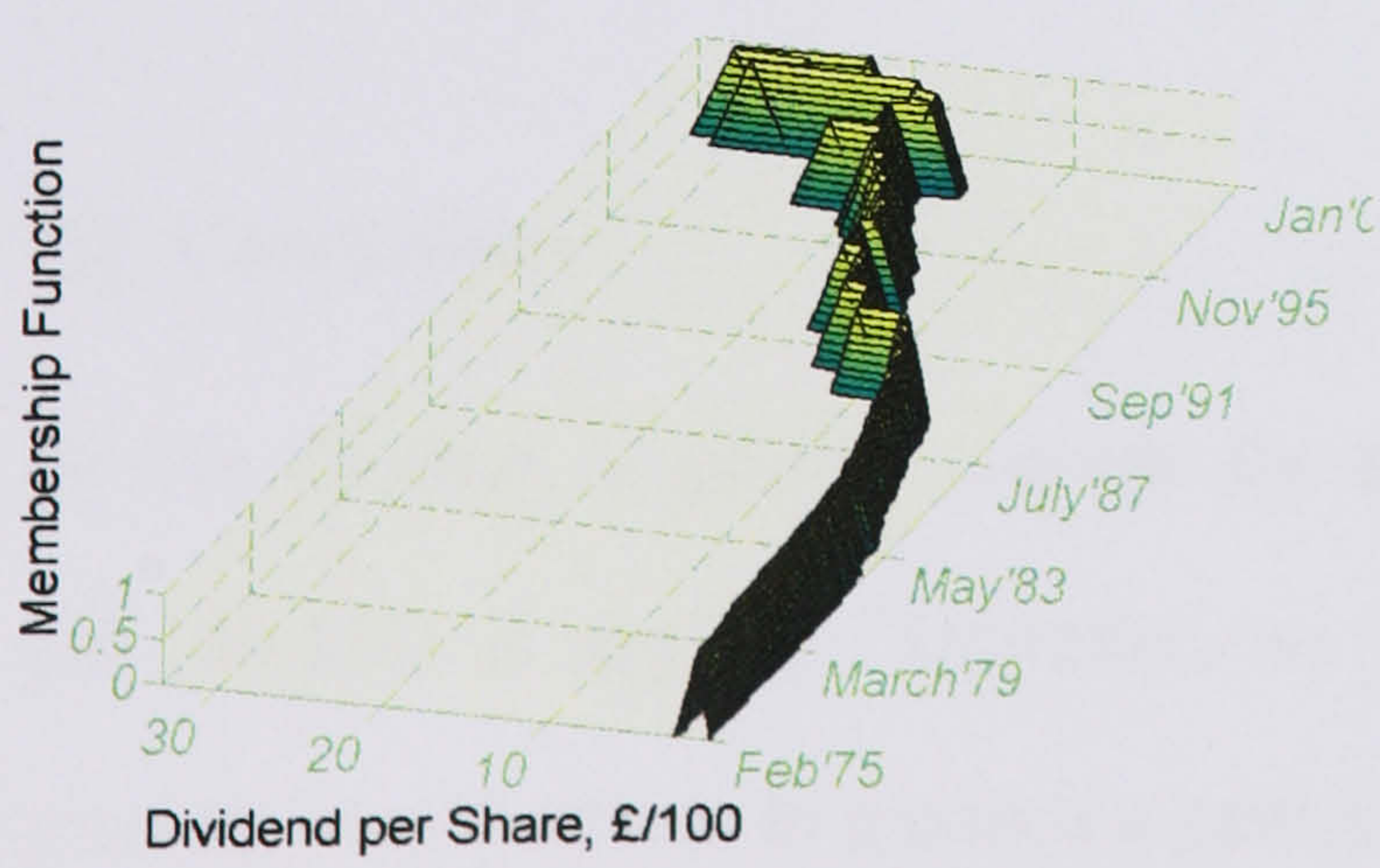


Figure 2.8a: BRITISH AMERICAN TOBACCO - fuzzified data

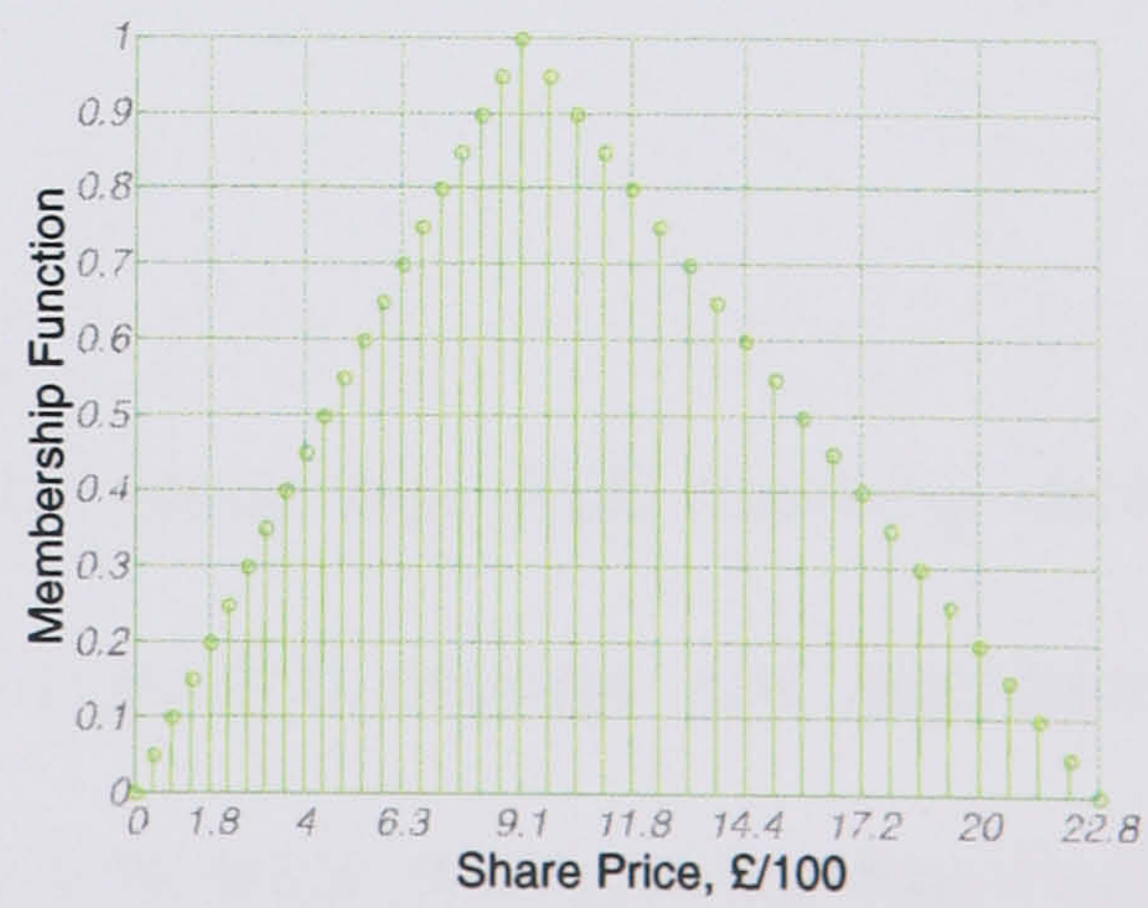


Figure 2.8b: BRITISH AMERICAN TOBACCO evaluated fuzzy share price

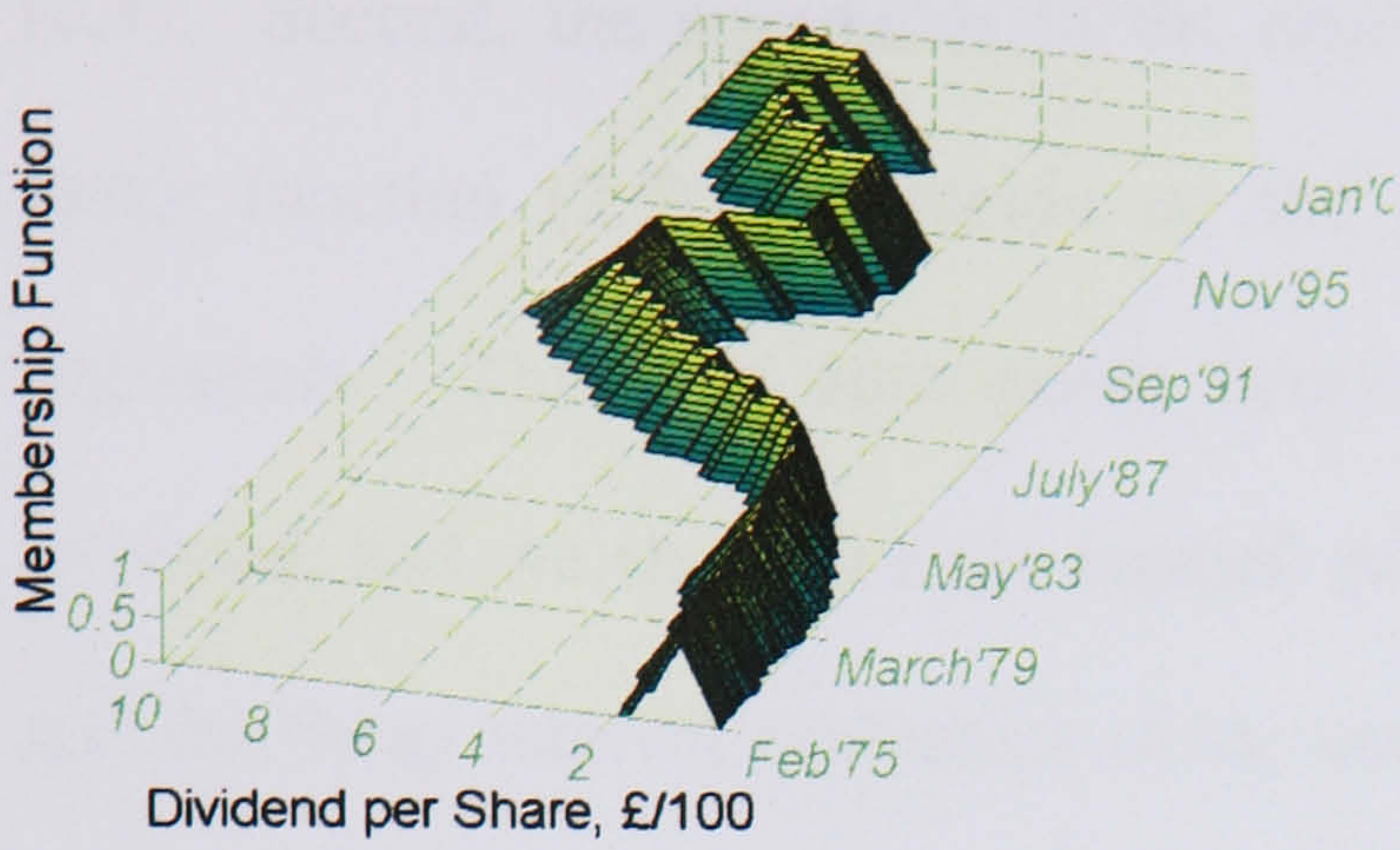


Figure 2.9a: BUNZL - fuzzified data

Figure 2.9b: BUNZL - evaluated fuzzy share price

From the empirical results above, we can make the following conclusions. Although the same data calibration procedure is applied to all companies, the resultant fuzzy trajectories differ in shape, and thus in the uncertainty modelled. Furthermore, one can observe how the character and shape of the fuzzy trajectory affects the shape of the membership function of the evaluated share price $\tilde{P}_0$. An $\alpha - cut$ in the fuzzy interval price $\tilde{P}_0(\alpha)$ in the graphics on the right-hand side includes all the values possible as a result of all the values associated with the same level of uncertainty in the graphics on the left-hand side. Each $\alpha - cut$ $\tilde{P}_0(\alpha)$ is an interval estimate of the price at the corresponding level of uncertainty.

2.5 Conclusion

In this Chapter, a general concept for solving fuzzy equations with made-up data [35,136-138] is applied. However, we specify this general concept and introduce relevant modifications to produce a particular approach to asset pricing using empirical data. The recommended steps are as follows. First, the crisp equation is solved with regard to the variable of interest, and we solve the equation (2.1) to produce the function

(2.4). Second, the arguments in the resultant function are fuzzified, and we get the fuzzy function (2.5) and decide on the shape of the membership functions of its arguments. Third, an appropriate fuzzy interval approximation of the solution is provided, and we show that the applied pricing function commutes with level-cutting, and the fuzzy interval evaluation of the asset price is produced with the formulas (2.8) and (2.9).

The same approach can be applied to solving single equations or systems of equations: linear, non-linear, difference or differential. Therefore, following the line of reasoning presented in this Chapter, a fuzzy solution to any crisp pricing model may be derived, and the results are not restricted to the model considered here. Thus increasing the computational flexibility, any imprecision relevant to the problem could be included in the asset evaluation.

In comparison, in hard computing imprecision and uncertainty are undesirable properties. Standard financial techniques focus on the most probable solutions, regard relatively large market shifts as too unlikely to matter, and completely neglect extreme situations. Such techniques may account for what occurs 'most' of the time in the market, but the picture they present does not reflect the reality, as major events happen in the residual time and investors are surprised by 'unexpected' market movements. In [141], it is argued that this events are not as extreme as it is usually assumed and that standard techniques ignore events that may happen every month on the market, or even every week.

The approach presented in this Chapter exploits the tolerance of soft computing for imprecision and uncertainty in order to achieve a tractable asset evaluation, which then serves as a basis for an asset risk and robustness analysis, introduced in the next Chapter.

Chapter 3: Measures of Risk and Robustness

3.1 Introduction

A soft computing solution to a problem allows an enhanced analysis of the results. We illustrated this quality in the previous Chapter, with the interpretation of the $\alpha-cuts$ of the evaluated asset price and the level of uncertainty in the pricing factors. In this Chapter, the analysis is continued and we introduce measures of risk and robustness.

The former measure communicates the risk that the market overvalues an asset, i.e. the risk that the asset costs effectively less than the price we pay. This is relevant to the situation toward the end of the twentieth century [1,142,143]. The fuzzy asset evaluation may be based on various pricing factors. When the factors include estimated future performance, then the measure presents the risk that we will not be able to recover the initial outlay.

A fuzzy evaluation of an asset is produced as a result of some considered range of imprecision modelled into the pricing factors. Then the risk measure is derived from this resultant evaluation. If the range of data imprecision is further broadened, this will modify the membership function of the evaluated price and affect the risk value. A robustness measure is introduced to estimate how sensitive is the evaluated fuzzy asset price and particularly its risk level toward increased imprecision.

The analytical conclusions for both measures are followed by empirical results and their interpretation.

3.2 Asset Risk Measure: Definition and Empirical Results

The decision to invest in an asset is generally based on an estimate of how favourable this choice is. The following risk measure serves the role of such an indicator. We have the fuzzy evaluation for the asset price, and observe how that asset is currently priced on the market. The fuzzy interval evaluates the true value of the asset. Every crisp price element of the fuzzy interval is assigned a degree of membership to the true price, $\mu\left(x_{P_0}\middle|\tilde{P}_0\right)$. Then, if the current market price Q is hypothetically considered as element of the fuzzy interval, it can also be assigned a degree of membership. This degree indicates up to what level α_Q, $0 \leq \alpha_Q \leq 1$, we believe the market price is the true asset price, i.e. the asset is neither undervalued nor overvalued.

$$\alpha_Q = \mu\left(x_{P_0} = Q\middle|\tilde{P}_0\right) \tag{3.1}$$

When the market price is situated on the left-hand side of the point with a membership level of 1 in the graph of the evaluated fuzzy interval price $\tilde{P}_0(1)$, then the asset could be undervalued currently and thus attractive for investment at this time. The opposite is valid when the market price is situated on the right-hand side.

Let us focus on the former case, $Q < \tilde{P}(1)$. If a price value with a membership level lower than α_Q happens to be the true value, $\mu\left(x_{P_0}\middle|P_0\right) < \alpha_Q$, then the investor effectively overpays and bears the risk of realising a loss. When the market price accords with the true value, then no loss or profit is expected, and for values with

$\mu\left(x_{P_0}\middle|P_0\right)>\alpha_Q$, the asset would be underpaid for. Therefore α_Q differentiates between the two outcomes, and may serve as a risk measure $\Re$, $0\le\Re\le 1$, as introduced in our publications [P2,P3].

Definition 3.1: Asset risk measure

The asset risk measure can be formulated as:

$$\Re=\alpha_Q=\mu\left(Q\middle|\tilde{P}_0\right)=Poss\left[\tilde{P}_0=Q\right] \ , \tag{3.2}$$

where α_Q *is the membership level of the current market price* Q *to the evaluated fuzzy asset price* $\tilde{P}_0$. [P2,P3]

If we consider the case where $Q>\tilde{P}(1)$, the risk there is too high for any value of α_Q. Thus the risk measure $\Re=1$ is accepted for all α_Q.

The above analysis is illustrated with the following graphics presenting the risk evaluation for the first ten companies. The complete set of graphics is provided in Appendix A2.

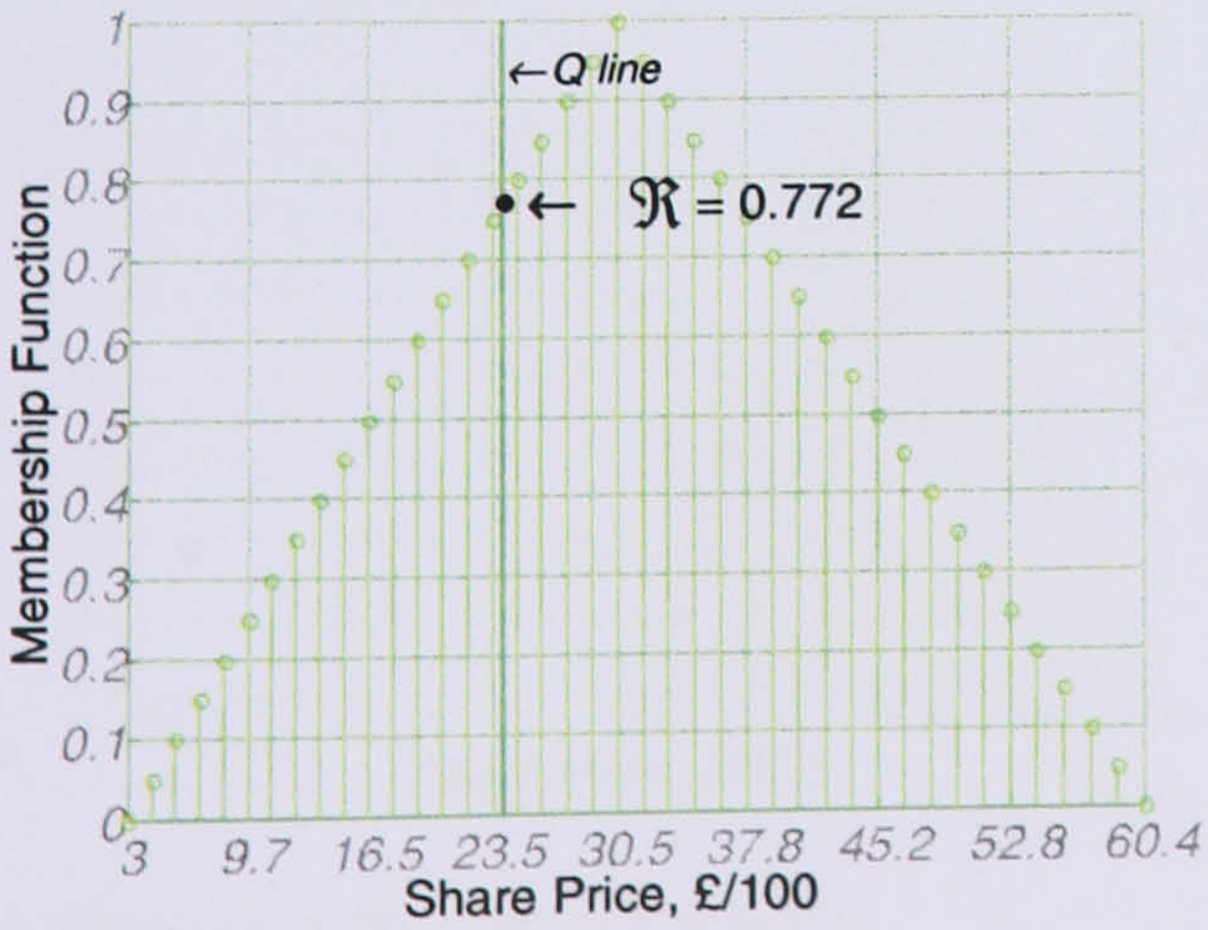


Figure 3.1: BASS
evaluated risk $\Re=0.772$

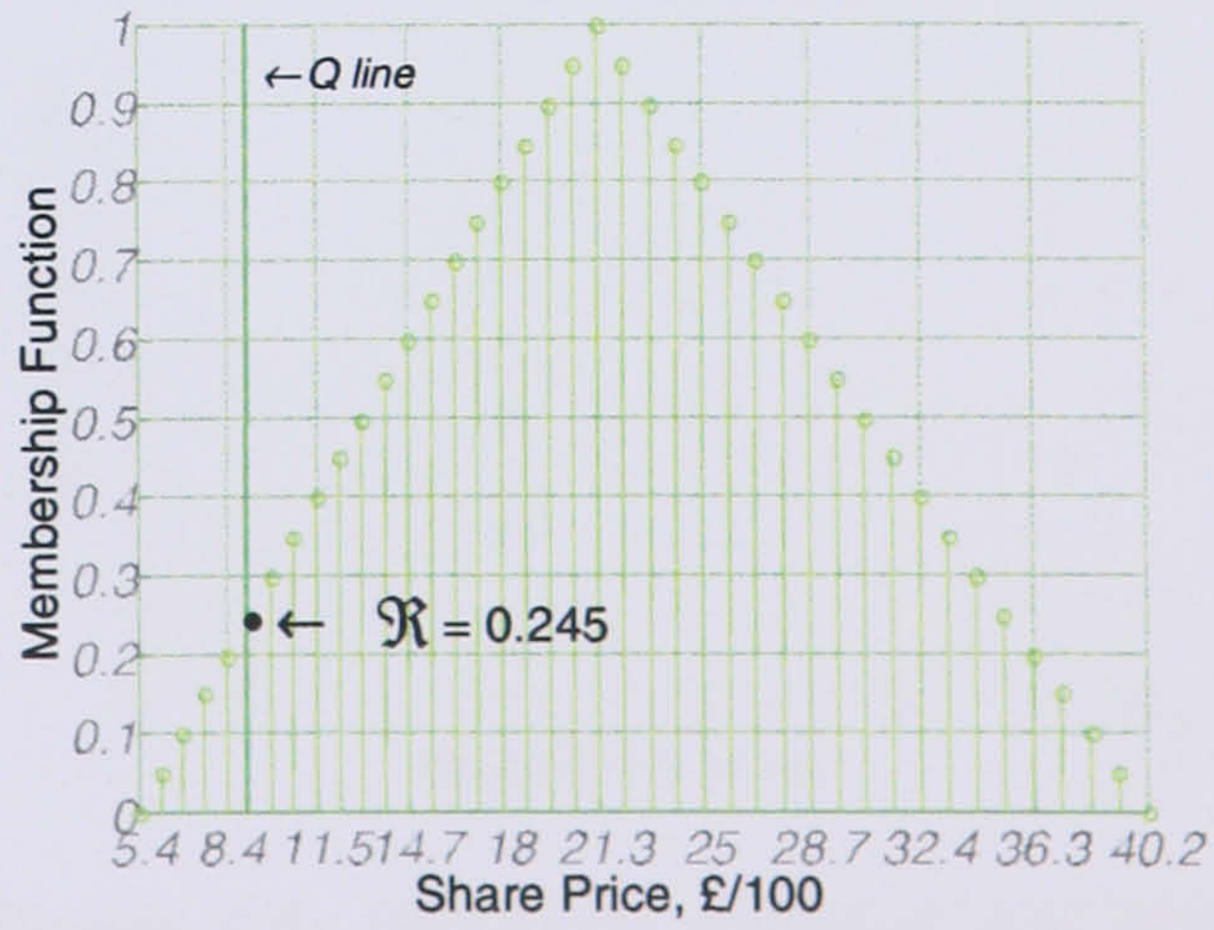


Figure 3.2: BBA GROUP
evaluated risk $\Re=0.245$

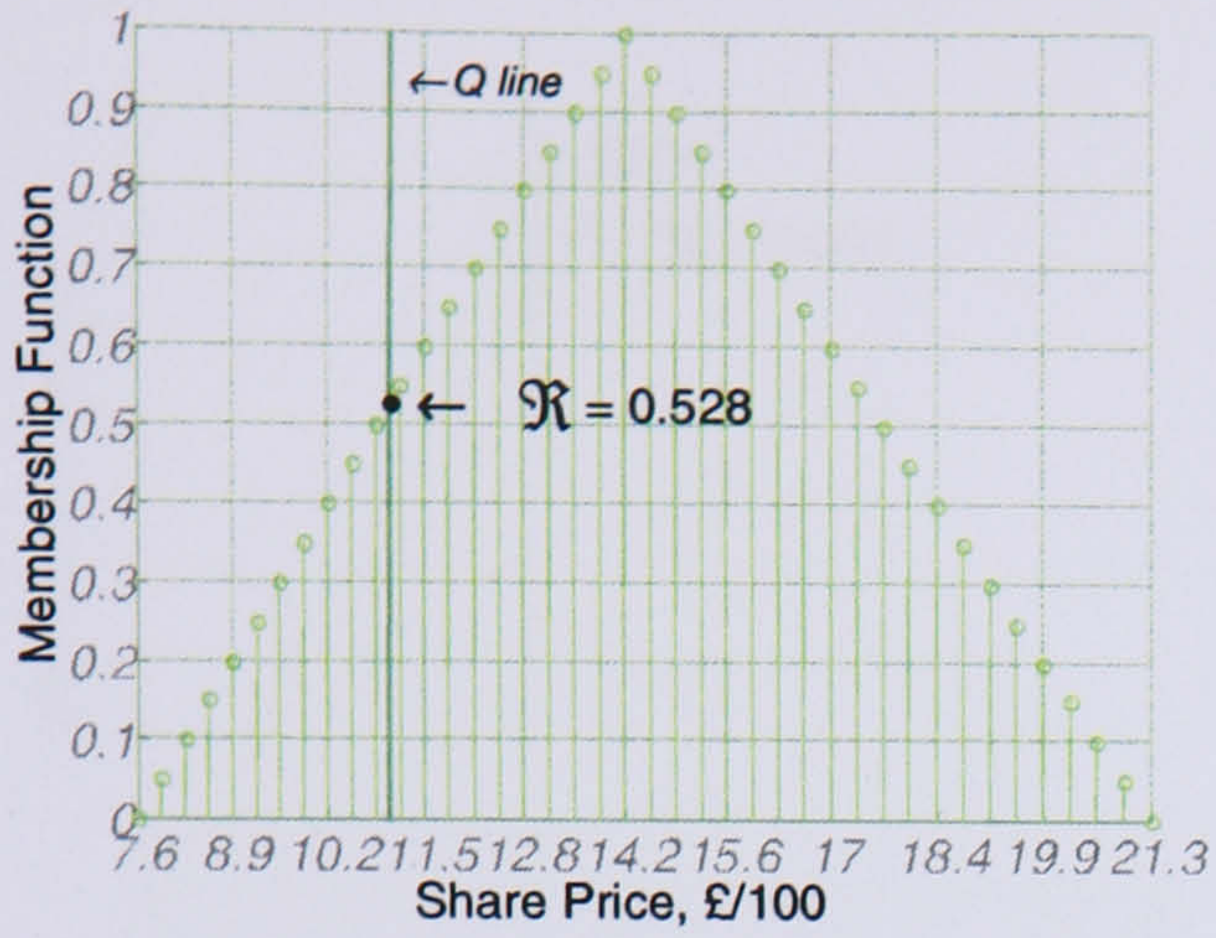


Figure 3.3: BENTALLS
evaluated risk $\Re = 0.528$

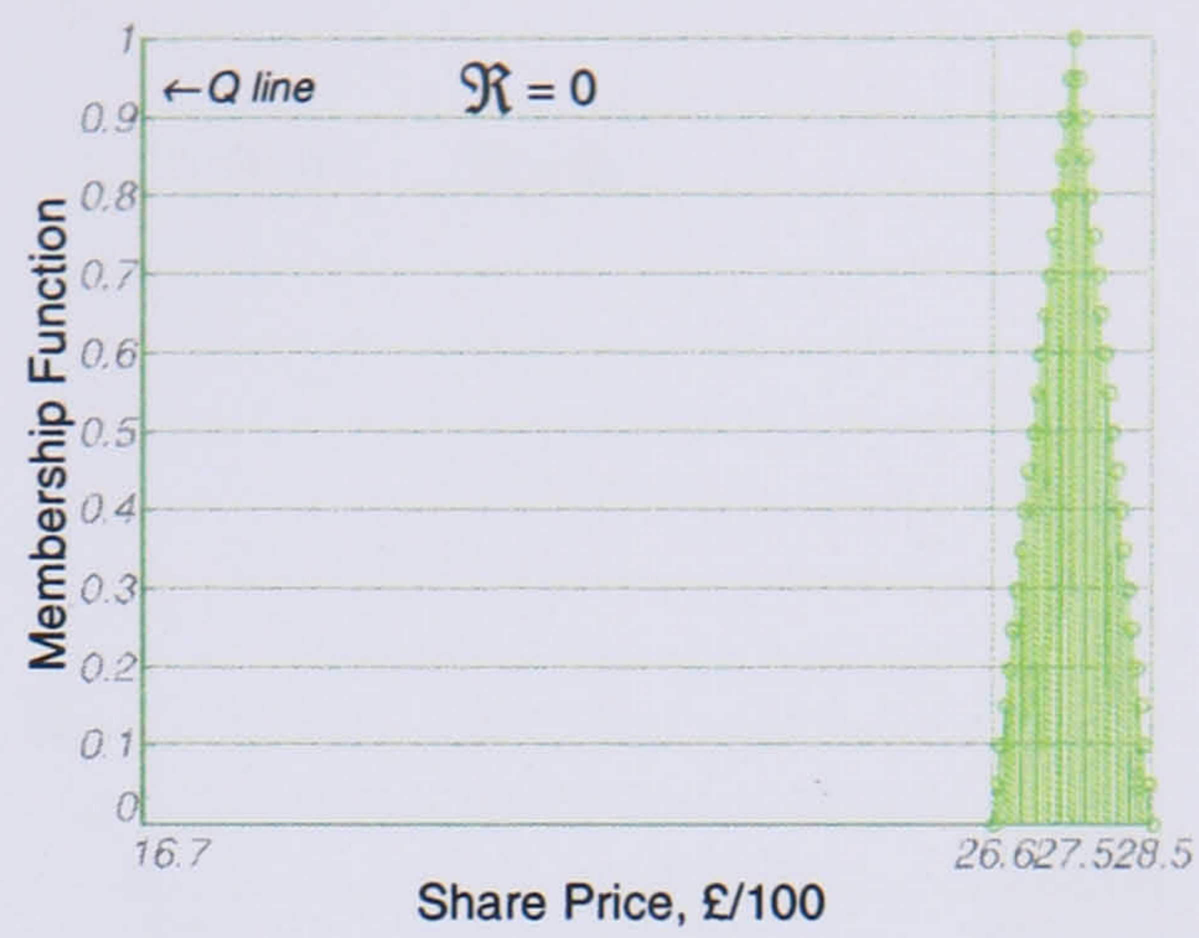


Figure 3.4: BLUE CIRCLE INDUSTRIES
evaluated risk $\Re = 0$

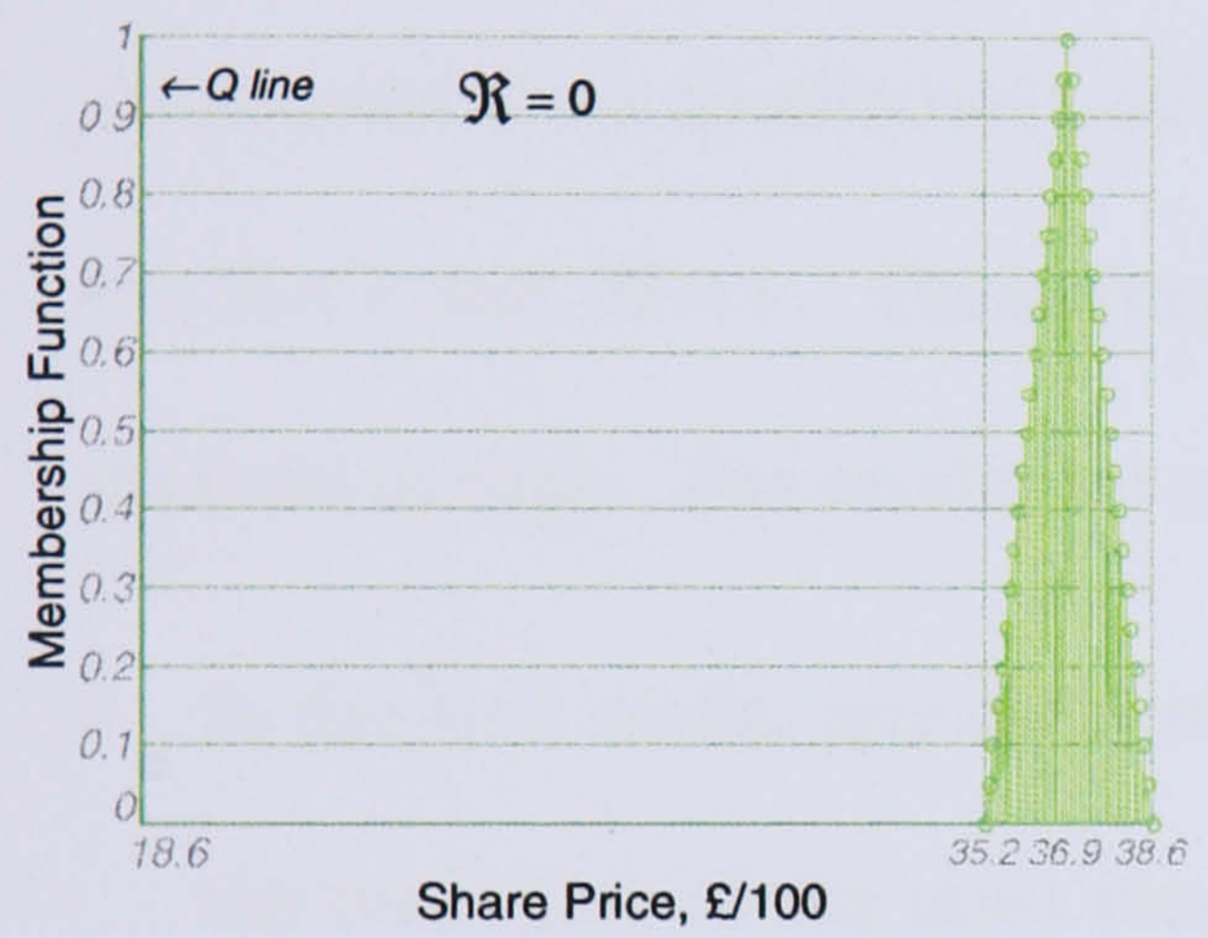


Figure 3.5: BOC GROUP
evaluated risk $\Re = 0$

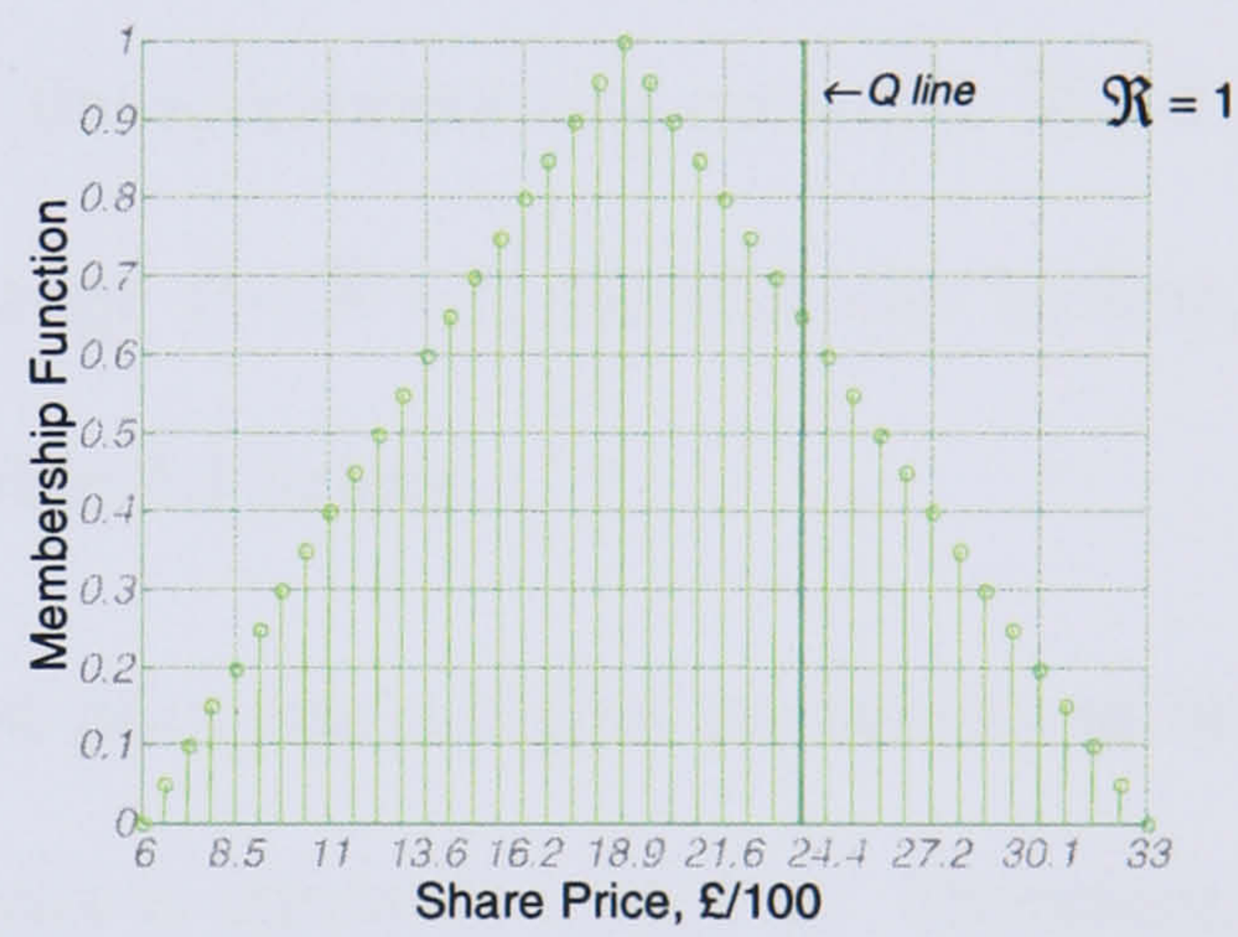


Figure 3.6: BOOTS CO.
evaluated risk $\Re = 1$

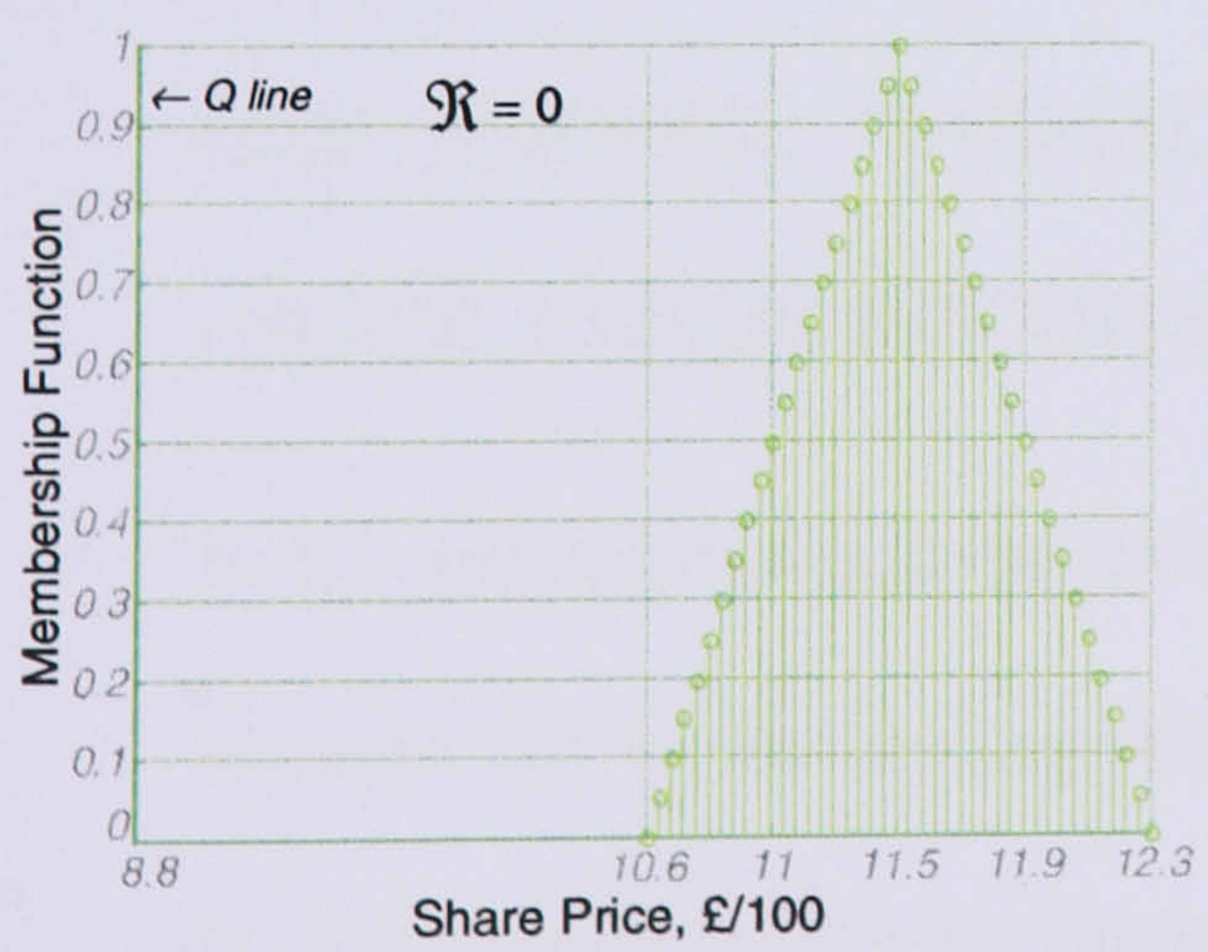


Figure 3.7: BP AMOCO
evaluated risk $\Re = 0$

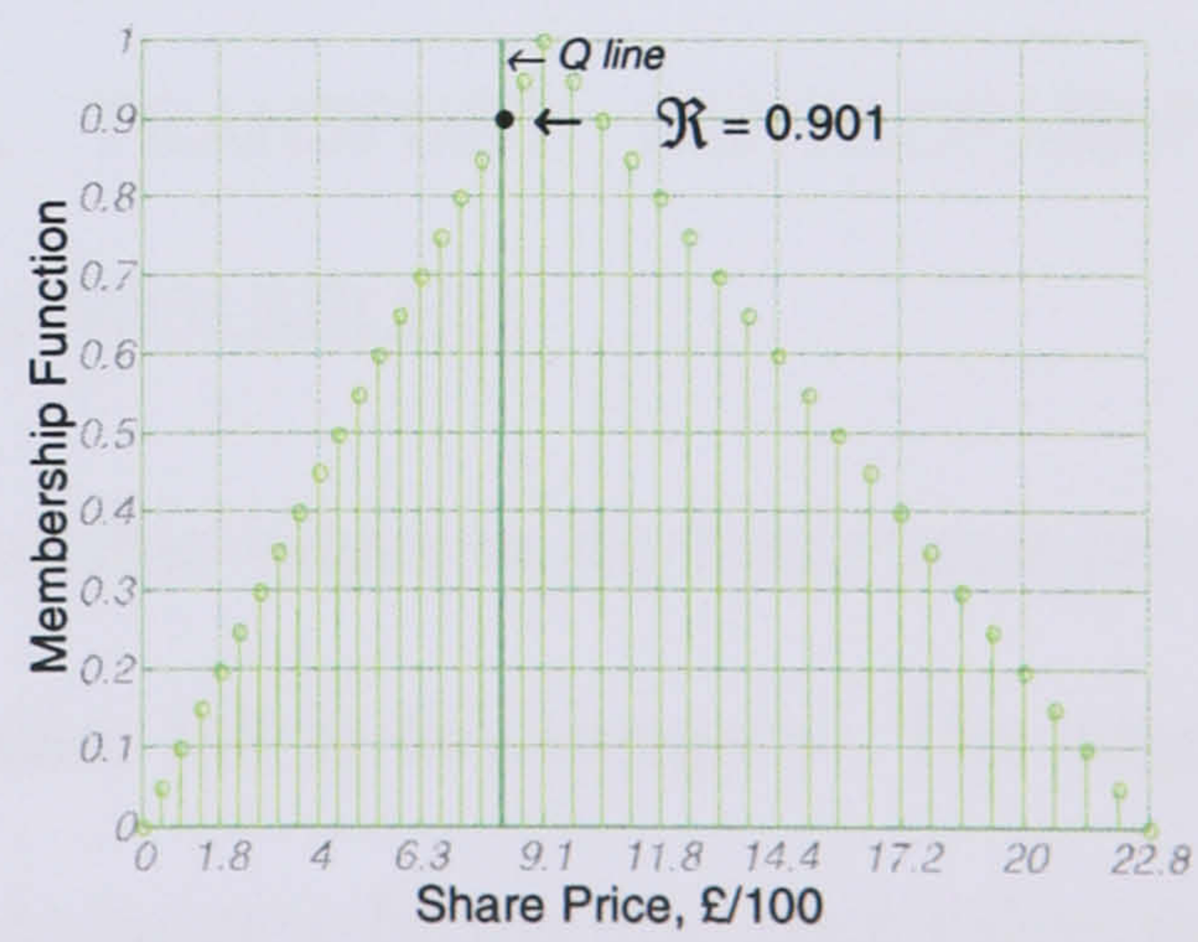


Figure 3.8: BRITISH AMERICAN TOBACCO
evaluated risk $\Re = 0.901$

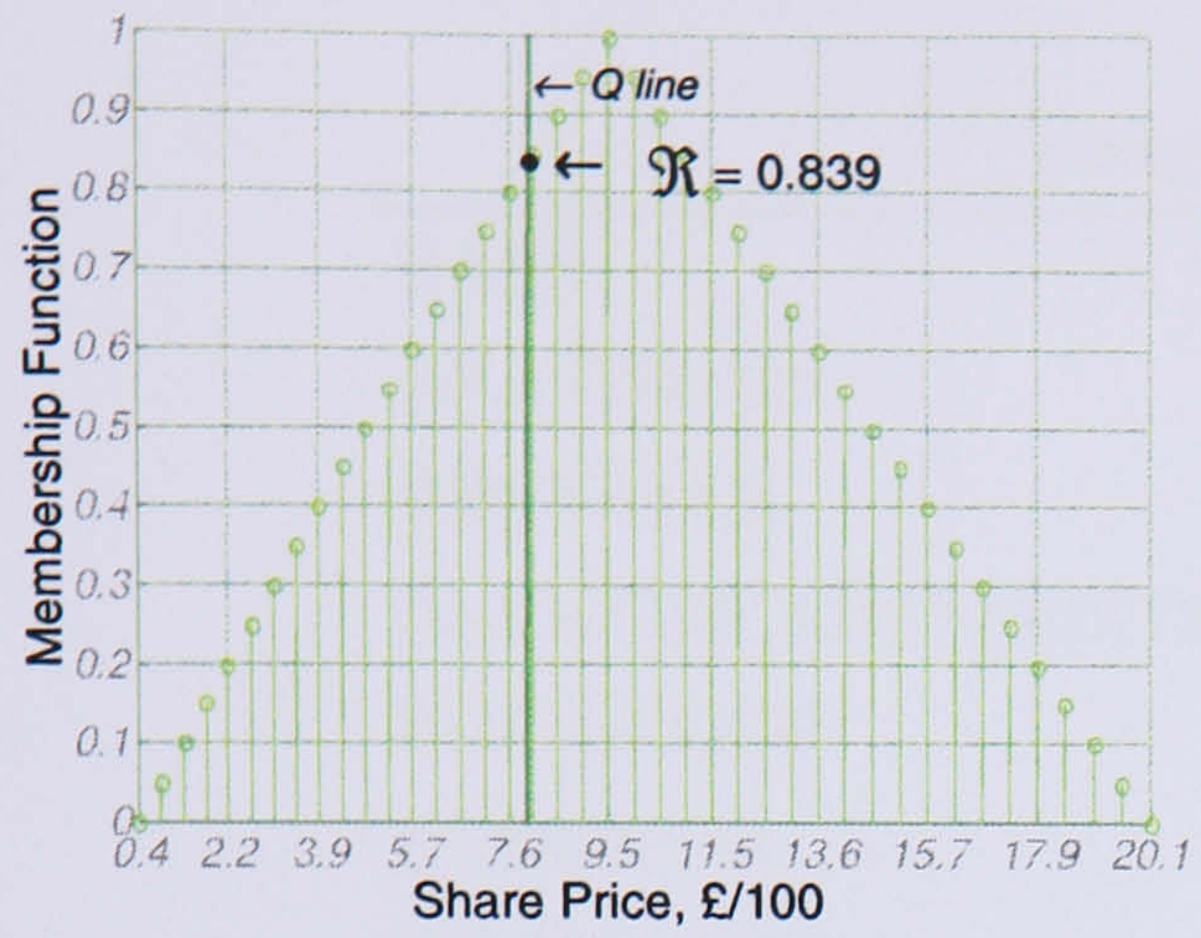


Figure 3.9: BUNZL
evaluated risk $\Re = 0.839$

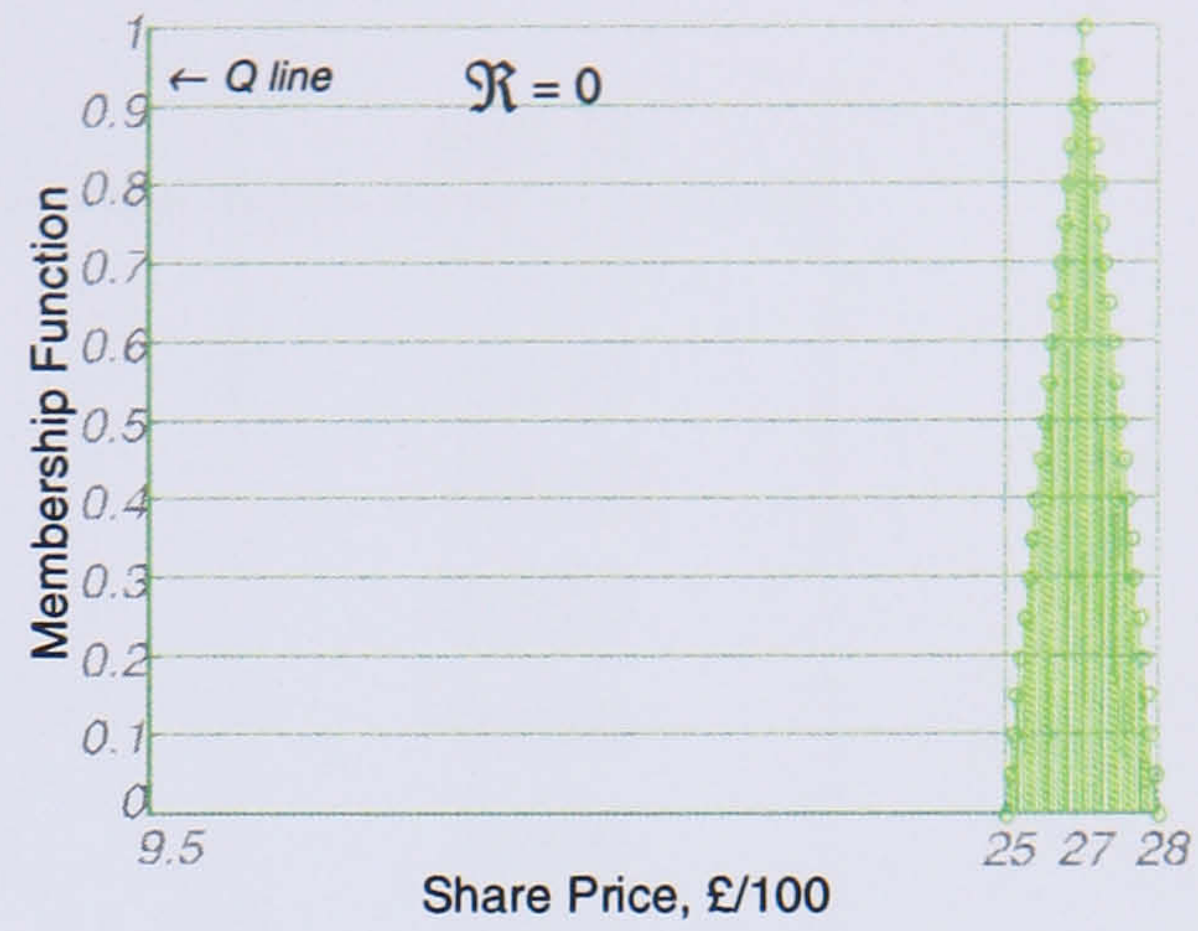


Figure 3.10: COATS VIYELLA
evaluated risk $\Re = 0$

There exist three qualitatively different modes for the investment risk measure: $\Re = 0$, $0 < \Re < 1$ and $\Re = 1$. Furthermore, within the range $0 < \Re < 1$, the risk can be low, medium or high. The results are summarised in Table 3.1 below.

- In the first mode, any element of the evaluated price has a higher possibility to be the true value of the asset then the price the asset is currently traded at. Therefore, no loss is expected and the asset is a favourable investment. The empirical results show that the shares of the following companies qualify as favourable assets: BLUE CIRCLE INDUSTRIES, BOC GROUP, BP AMOCO, COATS VIYELLA, LEX SERVICE, TARMAC, TI GROUP, TRANSPORT DEVELOPMENT GROUP, UNITED BISCUITS HOLDINGS and WOLSELEY.

- In the second mode, there exists a subset of possible values in the fuzzy asset price situated left to the trading price. Most companies fall in this category. The larger the subset and the higher the possibility of its elements to be the true asset value, the higher the investment risk. For example, PILKINGTON with $\Re = 0.026$, TATE & LYLE with $\Re = 0.190$ and BBA GROUP with $\Re = 0.245$ all have a low

Table 3.1: Evaluated risk measure by company

company	*risk* $\Re$
BASS	0.772
BBA GROUP	0.245
BENTALLS	0.528
BLUE CIRCLE INDUSTRIES	0.000
BOC GROUP	0.000
BOOTS CO.	1.000
BP AMOCO	0.000
BRITISH AMERICAN TOBACCO	0.901
BUNZL	0.839
COATS VIYELLA	0.000
DIXONS GROUP	0.656
GOODWIN	0.421
GREAT UNIVERSAL STORES	0.942
HANSON	0.507
INCHCAPE	1.000
LEX SERVICE	0.000
MARKS & SPENCER	0.944
NORTHERN FOODS	0.339
PILKINGTON	0.026
RANK GROUP	0.658
RMC GROUP	0.479
SAINSBURY (J)	1.000
SCOTTISH & NEWCASTLE	0.487
SMITH (WH) GROUP	0.986
SMITHS INDUSTRIES	0.584
TARMAC	0.000
TATE & LYLE	0.190
TAYLOR WOODROW	0.937
TI GROUP	0.000
TRANSPORT DEVELOPMENT GROUP	0.000
UNILEVER	1.000
UNITED BISCUITS HOLDINGS	0.000
WHITBREAD	0.765
WIMPEY (GEORGE)	1.000
WOLSELEY	0.000

risk measure. The risk values for BENTALLS, DIXONS GROUP, GOODWIN, HANSON, NORTHERN FOODS, RANK GROUP, RMC GROUP, SCOTISH & NEWCASTLE and SMITHS INDUSTRIES fall within the medium range $0.250 < \Re < 0.750$. Next, WHITBREAD with $\Re = 0.765$, BASS with $\Re = 0.772$ and BUNZL with $\Re = 0.772$ are assigned a relatively high risk, while BRITISH AMERICAN TOBACCO, GREAT UNIVERSAL STORES, MARKS & SPENCER, SMITH (WH) GROUP and TAYLOR WOODROW all have a measure that exceeds 0.9 and thus approach the maximum risk value.

- In the third mode, the trading price exceeds the element with the highest possibility to be the true asset value, $\tilde{P}_0(1)$. There may still exist elements of the fuzzy interval price situated right to the trading price, however in these cases we will consider the investment as too risky and assigned the measure $\Re = 1$. The results indicate that the shares of the following companies qualify as unfavourable assets at that particular time, January 1975: BOOTS CO., INCHCAPE, SAINSBURY (J), UNILEVER and WIMPEY (GEORGE).

The approach is illustrated here with monthly data, as higher frequency data are expensive to obtain. However, modelling uncertainty with fuzzy intervals and evaluating a fuzzy asset price will make even better sense with daily or hourly data when the market is less settled and the fluctuations respond to various sources of imprecision.

Other techniques analyse different types of risk that may cause a project or investment to be unprofitable. Here those are rather considered as various sources of uncertainty, and allowances are maid to incorporate most types of imprecision while producing the overall risk measure. This feature of the approach is consistent with a line of reasoning in [144] suggesting the introduction of allowances for any type of imprecision while deriving the solution to a problem, and then analysing the character of the solution toward an overall criterion.

3.3 Asset Robustness Measure: Definition and Empirical Results

In Chapter 2, we described how the $\alpha - cuts$ of the fuzzified data can be considered as the values of the pricing factors that are situated at the same level of uncertainty $u = 1 - \alpha$, and that those values are involved in producing the corresponding $\alpha - cut$ of the asset price $\tilde{P}_0$. In section 3.2, we proposed the membership degree of the trading price Q to the evaluated asset price as a measure of the investment risk $\Re$. That membership degree determines the highest $\alpha - cut$ in the evaluated asset price which includes the trading price, and we will call it the $\Re - cut$. The elements of all $\alpha - cuts$ situated above the $\Re - cut$ have a higher possibility to be the true asset value, and the investment will be profitable if any of them happens to be the asset value. On the other hand, any $\alpha - cut$ situated below the $\Re - cut$ in the solution involves elements that will make the investment unprofitable, if they happen to be the asset value. The higher such $\alpha - cuts$ are situated, the more values with a higher possibility to be the asset value will have the chance to turn the investment unprofitable. This is another justification for the correct choice of the risk measure. Furthermore, it allows us to give one more interpretation of that measure. There is a critical level of the uncertainty embodied into the fuzzified pricing factors that delimits the risk of a project investing into the asset to become unprofitable. As $u = 1 - \alpha$, then starting with $u = 0$ and gradually increasing the level of factor uncertainty, the level

$$u_{\Re} = 1 - \Re \tag{3.3}$$

is reached. The investment is profitable at all levels of uncertainty lower than $u_{\Re}$, $u < u_{\Re}$, and there is a chance of becoming unprofitable at levels $u > u_{\Re}$. Generally, a low $u_{\Re}$ means that the chance of realising a loss appears at a low level of factor imprecision and thus the investment is highly risky. If for some asset $u_{\Re} = 0$, then the investment is unprofitable even at the lowest and practically at any level of uncertainty. Therefore, the highest risk $\Re = 1$ is correctly assigned to such assets. If $u_{\Re} = 1$, then the asset constitutes a profitable investment at all involved levels of imprecision. Accordingly, the lowest risk $\Re = 0$ is associated with such assets.

In Chapter 2, we also explain that the fuzzified factors model a range of imprecision, and their level-cuts only represent levels of uncertainty within that range. Figure 3.11 illustrates how the fuzzy factor trajectory for BP AMOCO transforms under a broader range of imprecision. Further illustration of the transformation of the fuzzy factor trajectories by company is available by comparing figures A1.1a to A1.35a in Appendix 1 with figures A3.1a to A3.35a in Appendix 3.

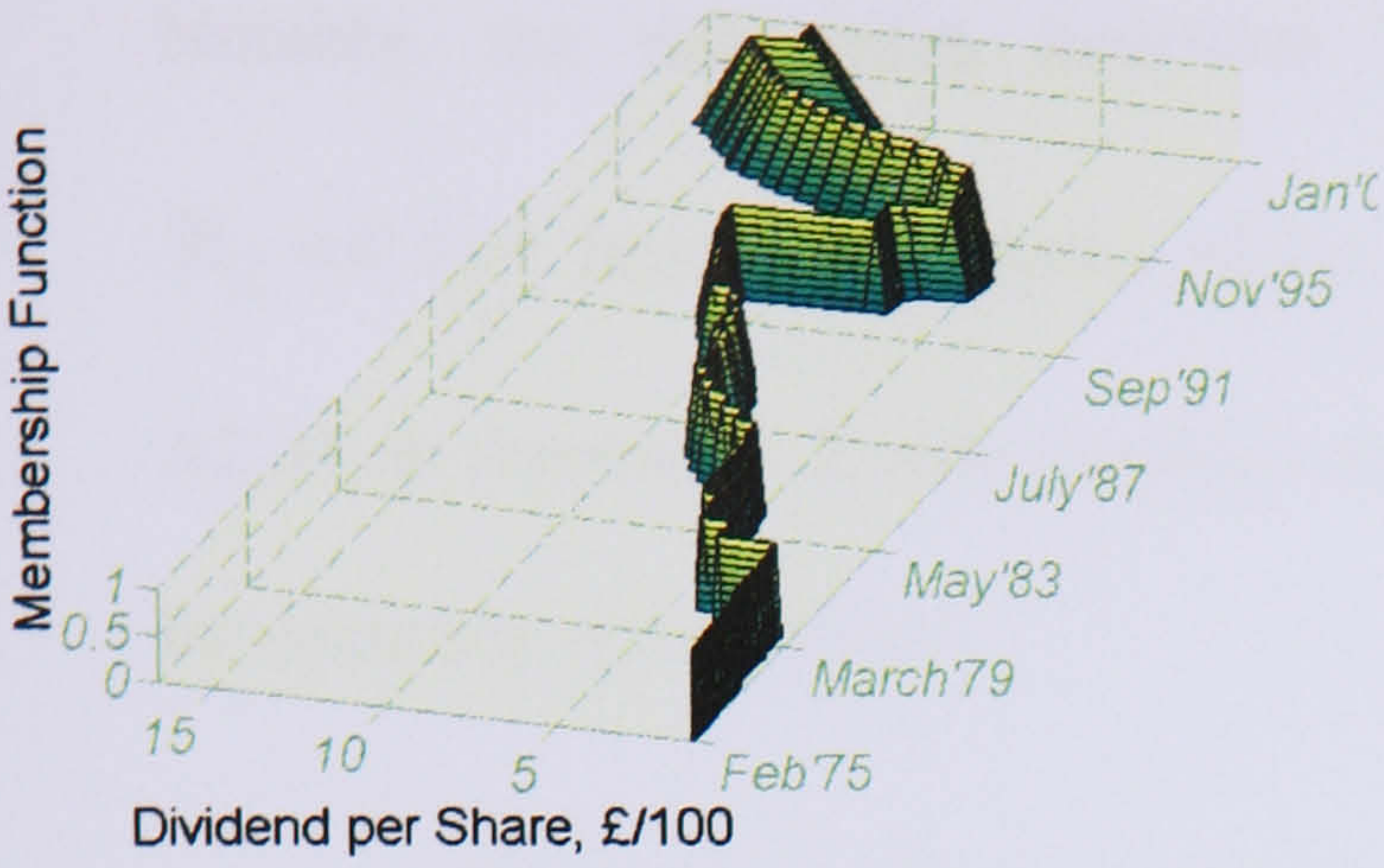


Figure 3.11a: BP AMOCO - fuzzified data under initial calibration

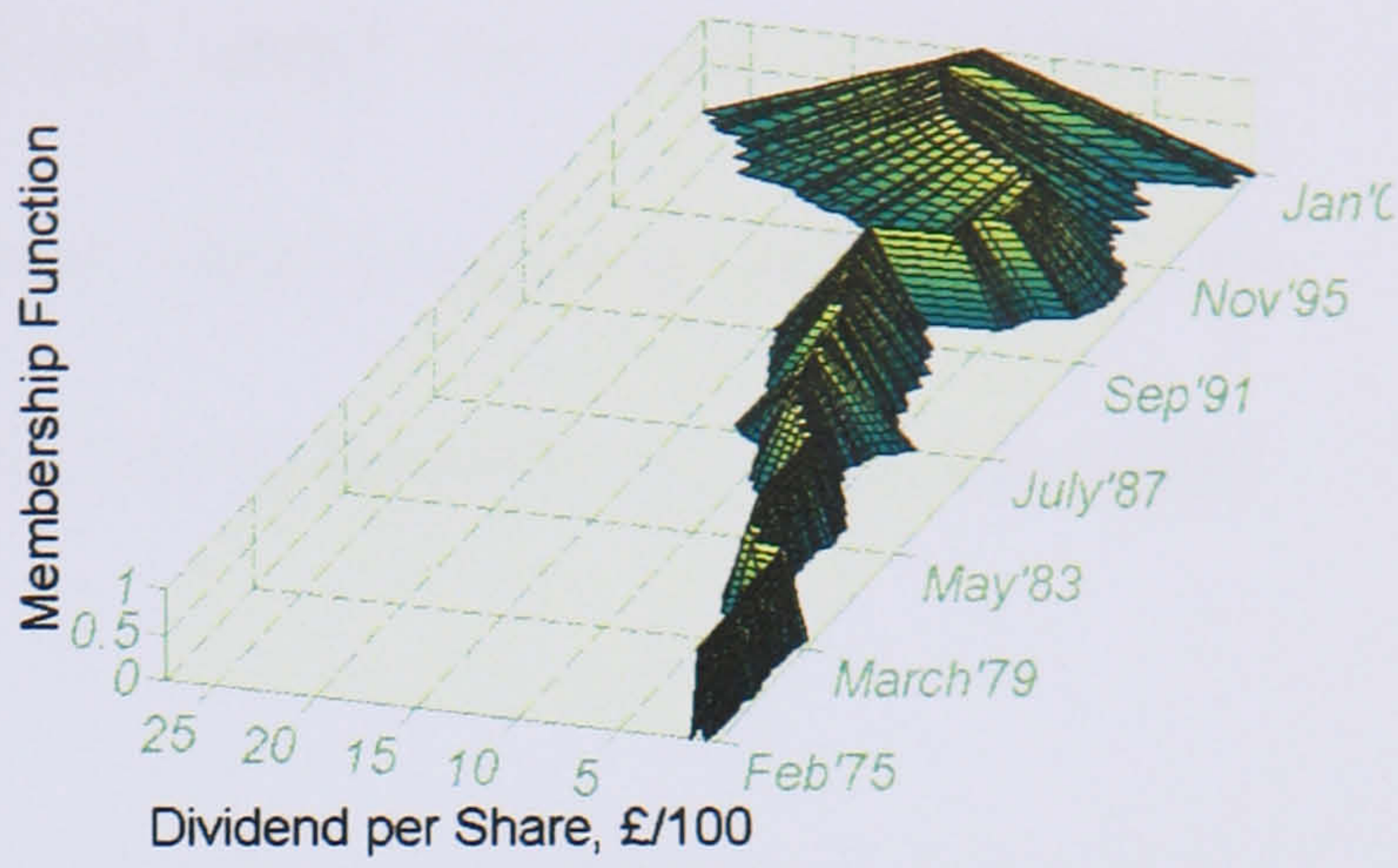


Figure 3.11b: BP AMOCO - fuzzified data under a broader range of modelled imprecision

Factors modelling a broader range of imprecision will produce a modified fuzzy asset evaluation. Therefore, the transformed factor trajectories will affect the corresponding risk measures. The chance of realising a loss will appear at lower levels of uncertainty $u_{\Re_2}$ within the broader range of data imprecision,

$$u_{\Re_2} \leq u_{\Re_1} \quad . \tag{3.4.1}$$

Consequently, the evaluated risk measure should increase,

$$\Re_2 \geq \Re_1 \quad . \tag{3.4.2}$$

Figure 3.12 illustrates this analytical conclusion with the empirical result for BP AMOCO.

Figure 3.12a: BP AMOCO – evaluated asset price and risk measure $\Re_1 = 0$ under initial calibration

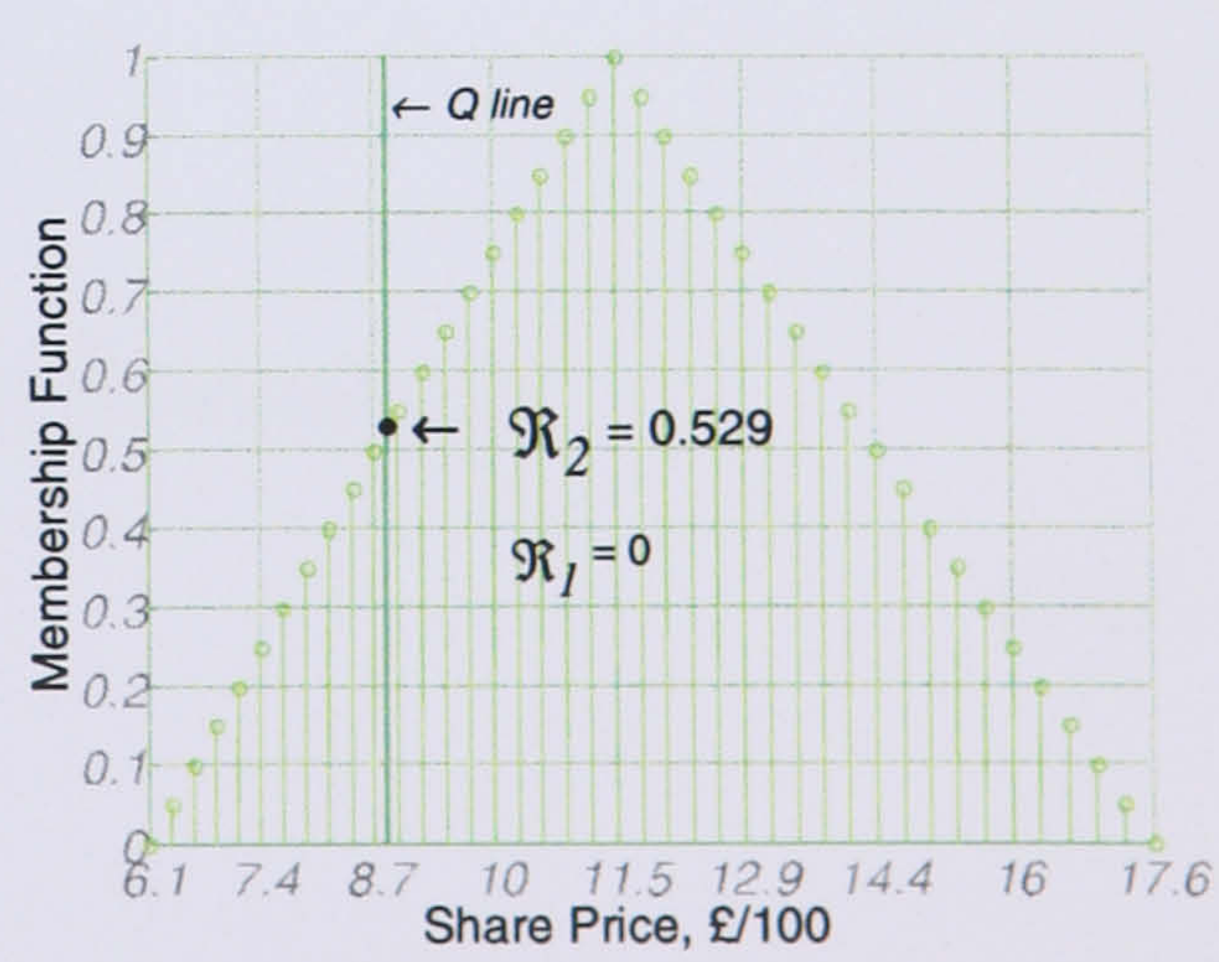


Figure 3.12b: BP AMOCO - evaluated asset price and risk measure $\Re_2 = 0.529$ under a broader range of modelled imprecision

Notably, the risk value increases from $\Re_1 = 0$ under the initial calibration to $\Re_2 = 0.529$ here. Analogous effect is observed when comparing graphics A2.1 to A2.35 in Appendix 2 with graphics A3.1b to A3.35b in Appendix 3, where the results by company are provided.

The analysis from the beginning of the section up to this point deducts the following definition, as introduced in our publications [P2,P3].

Definition 3.2: Asset risk and modelled factor imprecision

There exists a critical level of the uncertainty embodied into the pricing factors, and that level delimits the investment risk for the asset.

- *The critical level shifts downwards under a broader range of imprecision modelled with the fuzzified data.*
- *This results in an increased risk measure.* [P2,P3]

Therefore, we can introduce a further measure Δ, $0 \le \Delta \le 1$, evaluating another feature of an asset, how robust is its risk measure. The more robust assets should be assigned a higher measure, suggesting the following definition, as introduced in our publications [P3,P4].

Definition 3.3: Asset robustness measure

The asset robustness measure can be formulated as:

$$\Delta = 1 - (\Re_2 - \Re_1) \quad , \tag{3.5}$$

$$0 \le \Re_1 \le 1,\ 0 \le \Re_2 \le 1,\ \Re_2 \ge \Re_1 \quad ,$$

where $\Re_1$ and $\Re_2$ are risk measures under different calibration of the uncertain pricing factors. [P3,P4]

Let us present the robustness evaluation of the shares for the list of companies in our database. First, a broader range of data imprecision is modelled through slightly modifying the calibration procedure from Chapter 2. The support of the membership function of a factor, e.g. $\tilde{DY}_t$ at a particular $t \in [1,T]$, is chosen as wide as the 95% confidence interval of the Student's distribution, with only 6 degrees of freedom, of the crisp factor DY_t over the whole horizon $1 \le t \le T$.

$$DY_t(0) = \left[\underline{DY_t(0)} \; , \overline{DY_t(0)} \right] \; , \tag{3.6}$$

$$\underline{DY_t(0)} = DY_t - 2.447 \sqrt{\sum_{\tau=1}^{T} \left(DY_\tau - \sum_{\tau=1}^{T} DY_\tau \Big/ T \right)^2} \Bigg/ \sqrt{6} \, , \; 1 \le t \le T \; ,$$

$$\overline{DY_t(0)} = DY_t + 2.447 \sqrt{\sum_{\tau=1}^{T} \left(DY_\tau - \sum_{\tau=1}^{T} DY_\tau \Big/ T \right)^2} \Bigg/ \sqrt{6} \, , \; 1 \le t \le T \; .$$

The choice of the vertex of the membership function has not changed. The procedure is repeated for the rest of the factors $\tilde{P}_t$, $1 \le t \le T$, and $\tilde{R}$. Next, the evaluation of the robustness measure for the first six companies in the database is presented in figures 3.13b to 3.18b. The complete set of graphics is provided in Appendix A3.

Again, there are three qualitatively different cases for the robustness measure: $\Delta = 1$, $0 \le \Delta < 1$ and *'no value assigned'*. Additionally, the second case has subcategories, as some assets demonstrate lower robustness, while others are associated with higher values.

- The highest value $\Delta = 1$ is assigned to companies with $\Re_2 = 0$, and therefore $\Re_1 = 0$. It is unlikely for the risk measure to stay the same $\Re_2 = \Re_1$ if the initial risk is $0 < \Re_1 < 1$. Graphics 3.16 and 3.17 identify BLUE CIRCLE INDUSTRIES and BOC GROUP, correspondingly, as maximum robustness companies. The summary of the results in table 3.2 show the same quality for COATS VIYELLA, LEX SERVICE, TARMAC, UNITED BISCUITS HOLDINGS and WOLSELEY.

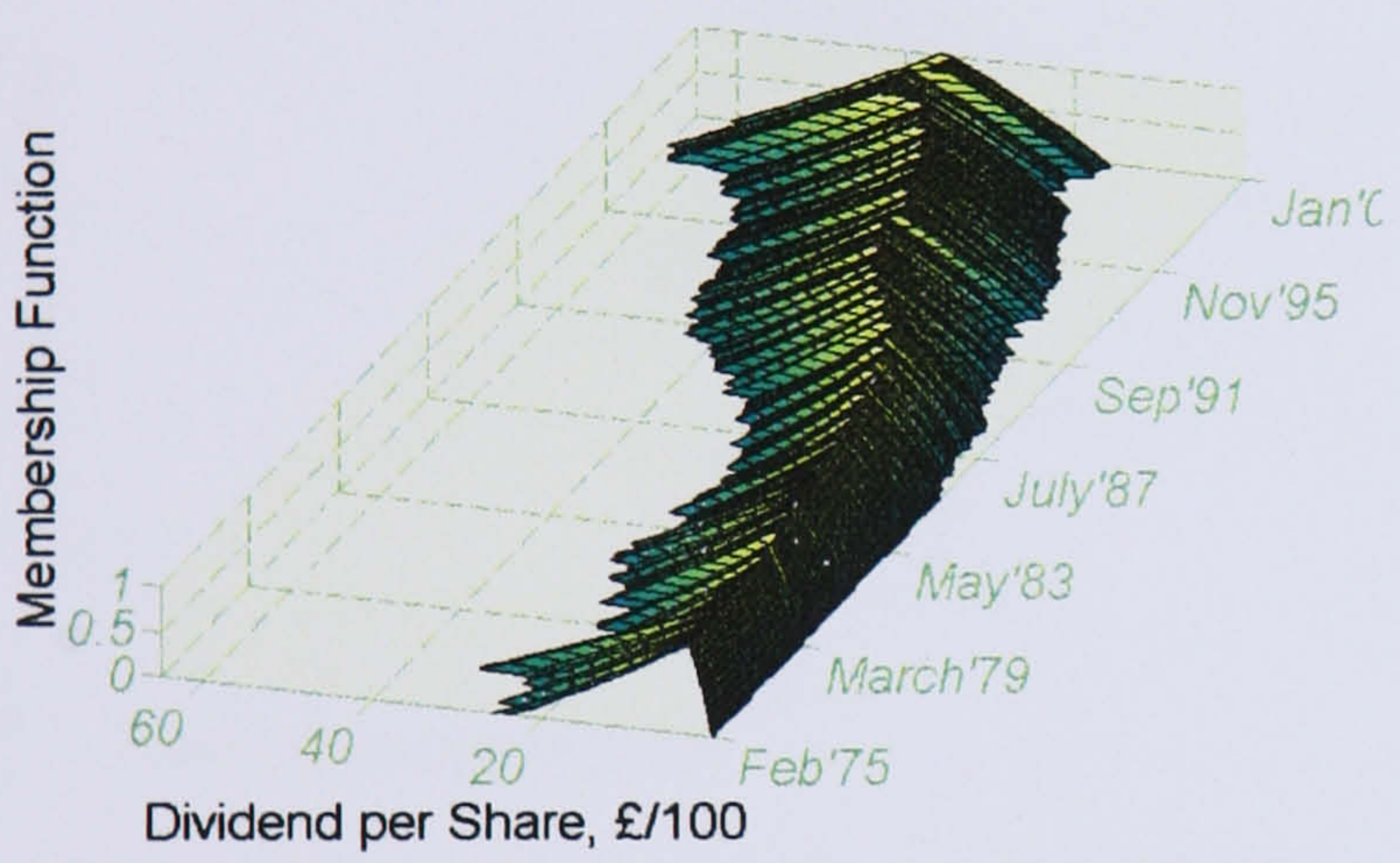


Figure 3.13a: BASS - fuzzified data under a broader range of imprecision

Figure 3.13b: BASS - evaluated robustness $\Delta = 0.952$

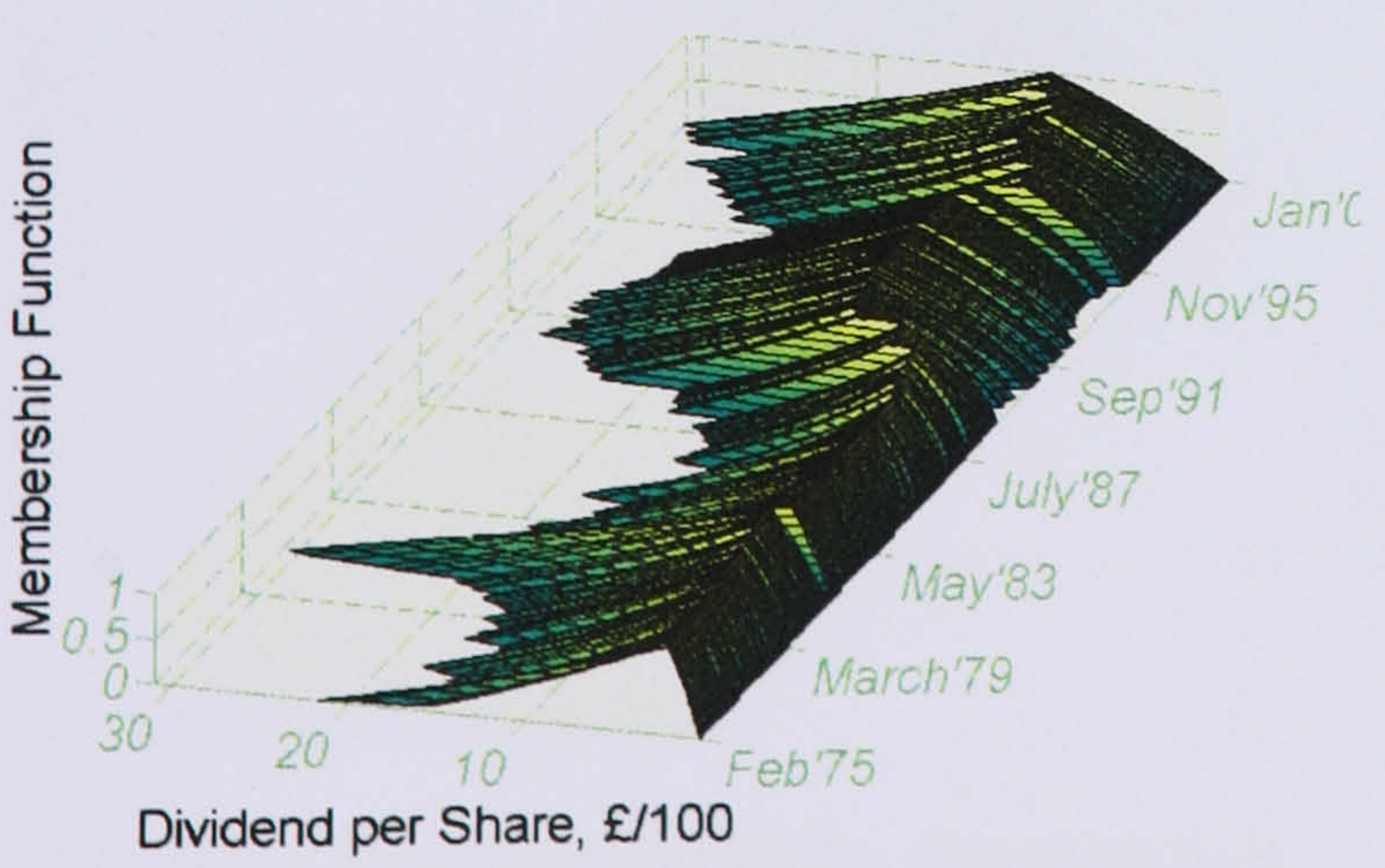


Figure 3.14a: BBA GROUP - fuzzified data under a broader range of imprecision

Figure 3.14b: BBA GROUP - evaluated robustness $\Delta = 0.718$

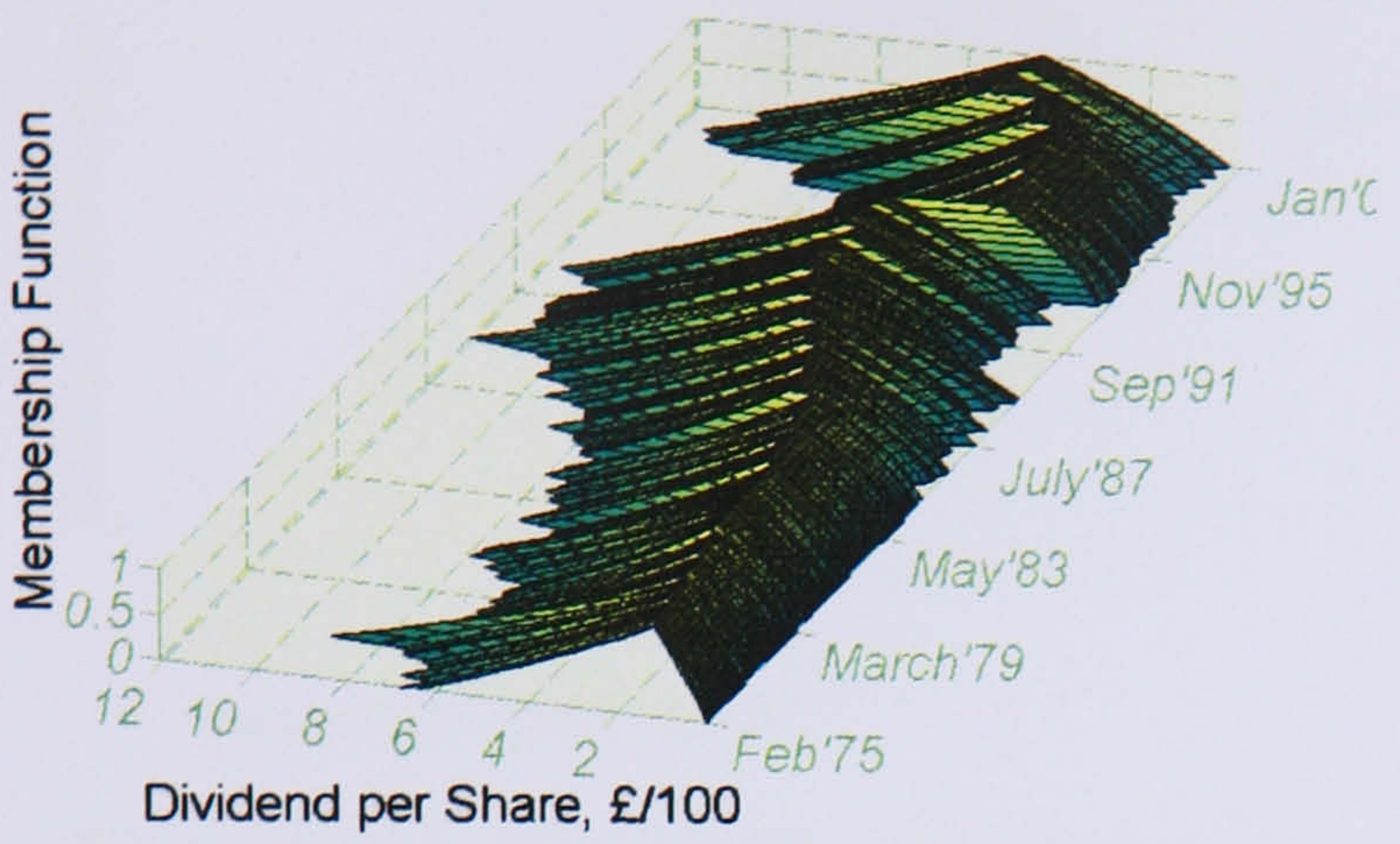


Figure 3.15a: BENTALLS - fuzzified data under a broader range of imprecision

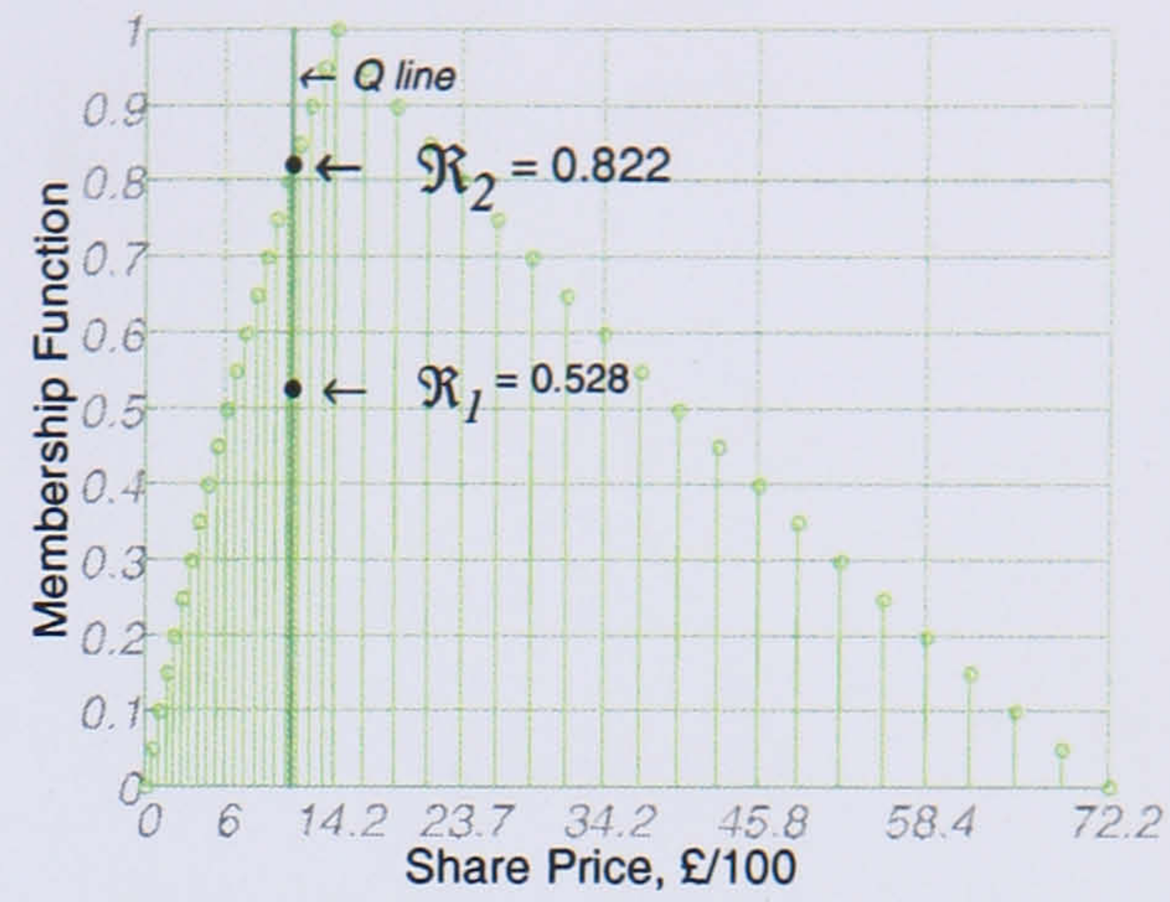


Figure 3.15b: BENTALLS - evaluated robustness $\Delta = 0.706$

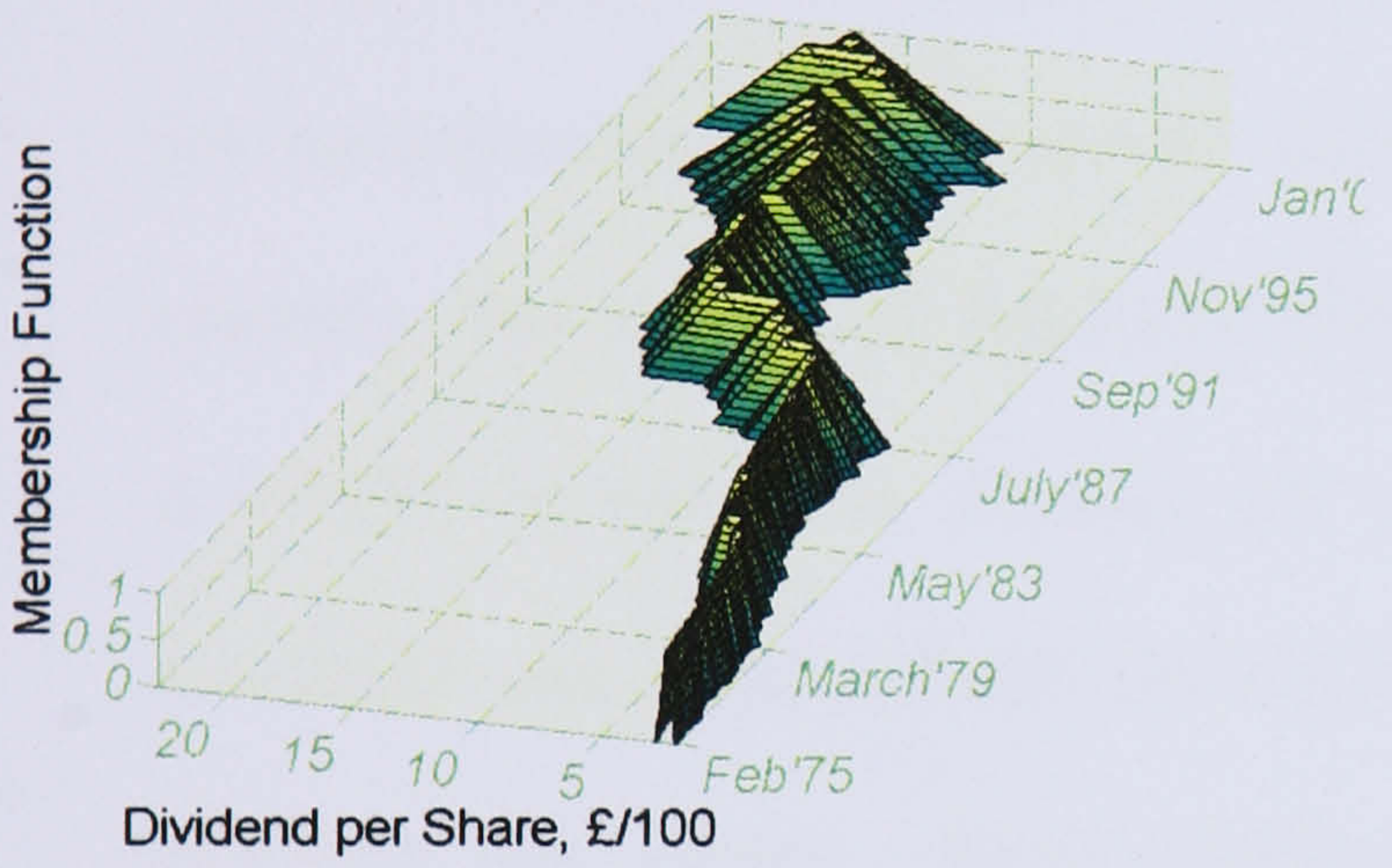


Figure 3.16a: BLUE CIRCLE INDUSTRIES - fuzzified data under a broader range of imprecision

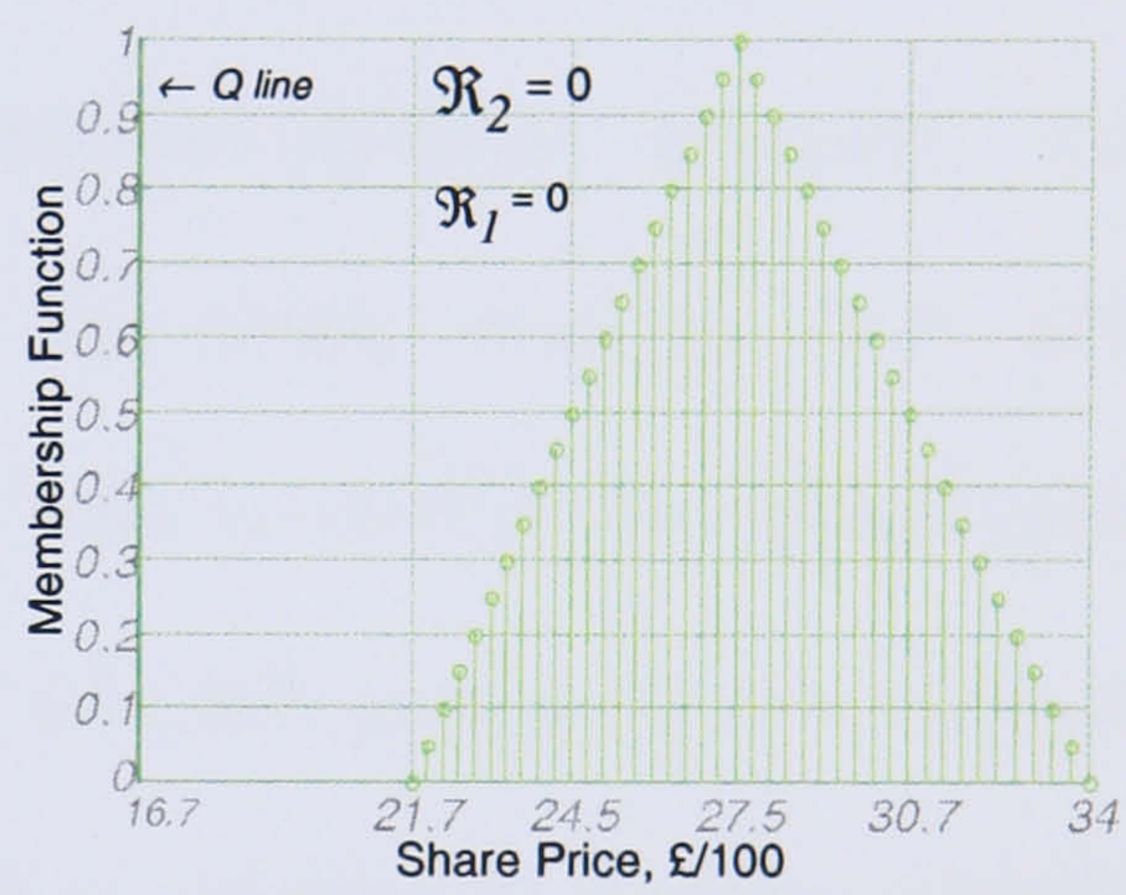


Figure 3.16b: BLUE CIRCLE INDUSTRIES - evaluated robustness $\Delta = 1$

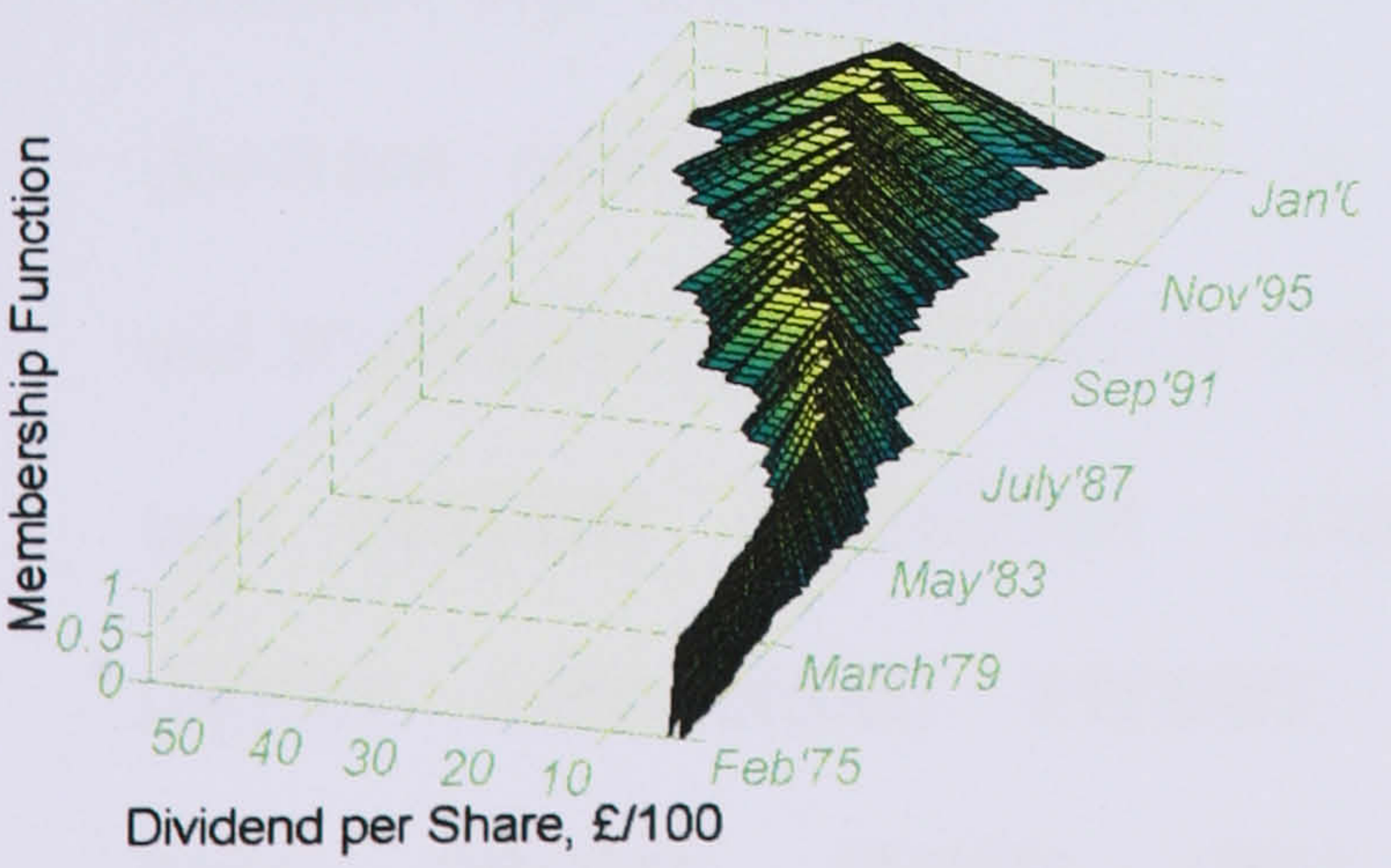


Figure 3.17a: BOC GROUP - fuzzified data under a broader range of imprecision

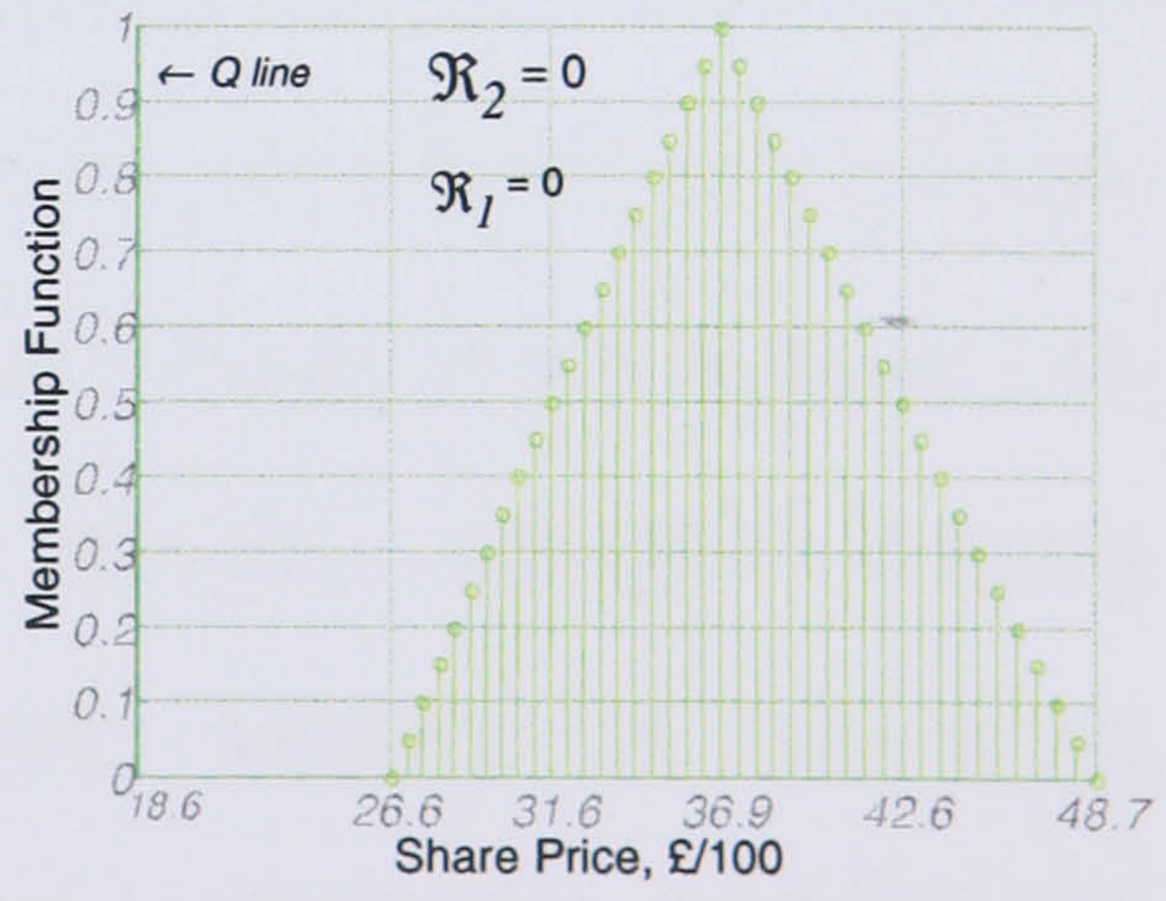


Figure 3.17b: BOC GROUP - evaluated robustness $\Delta = 1$

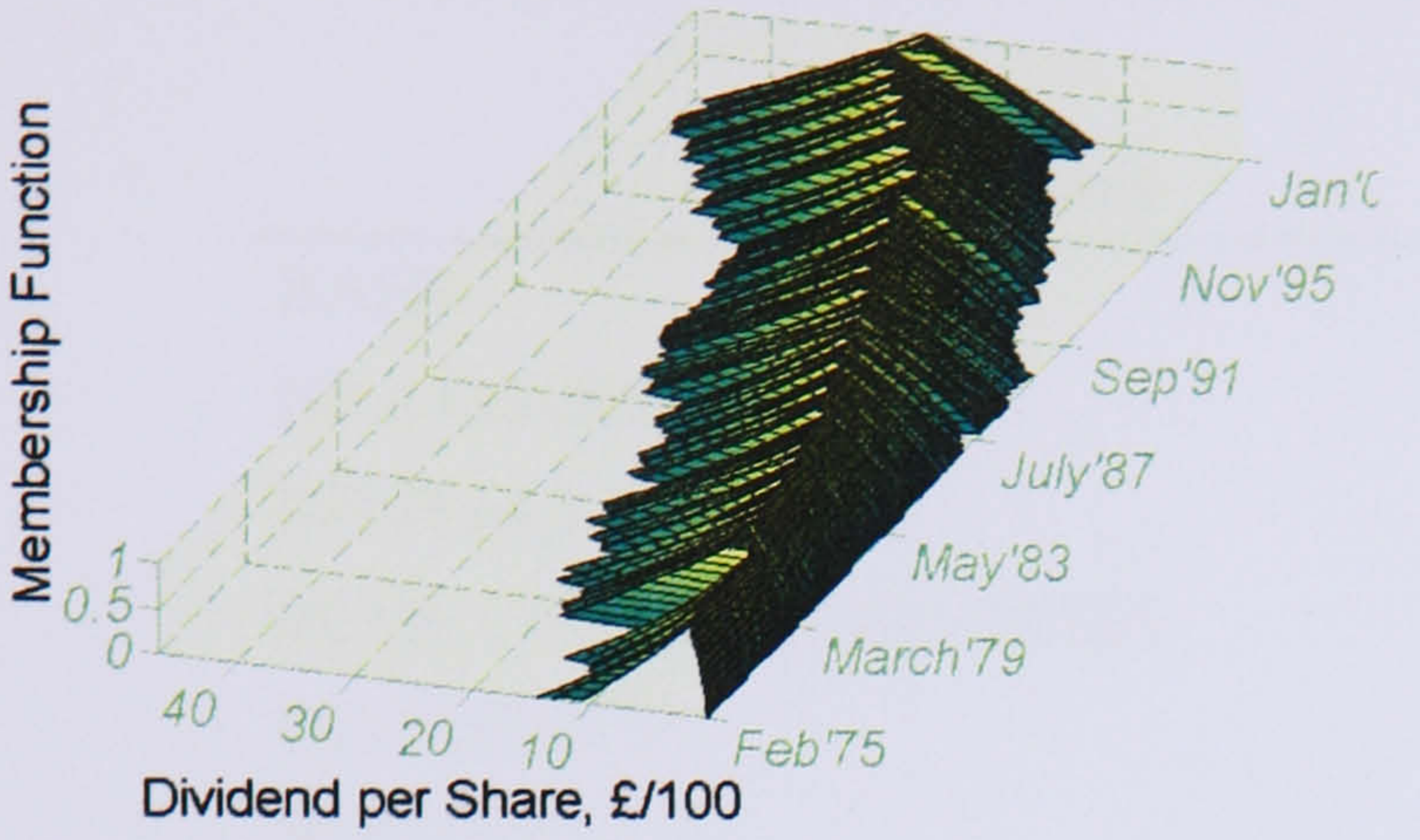


Figure 3.18a: BOOTS CO. - fuzzified data under a broader range of imprecision

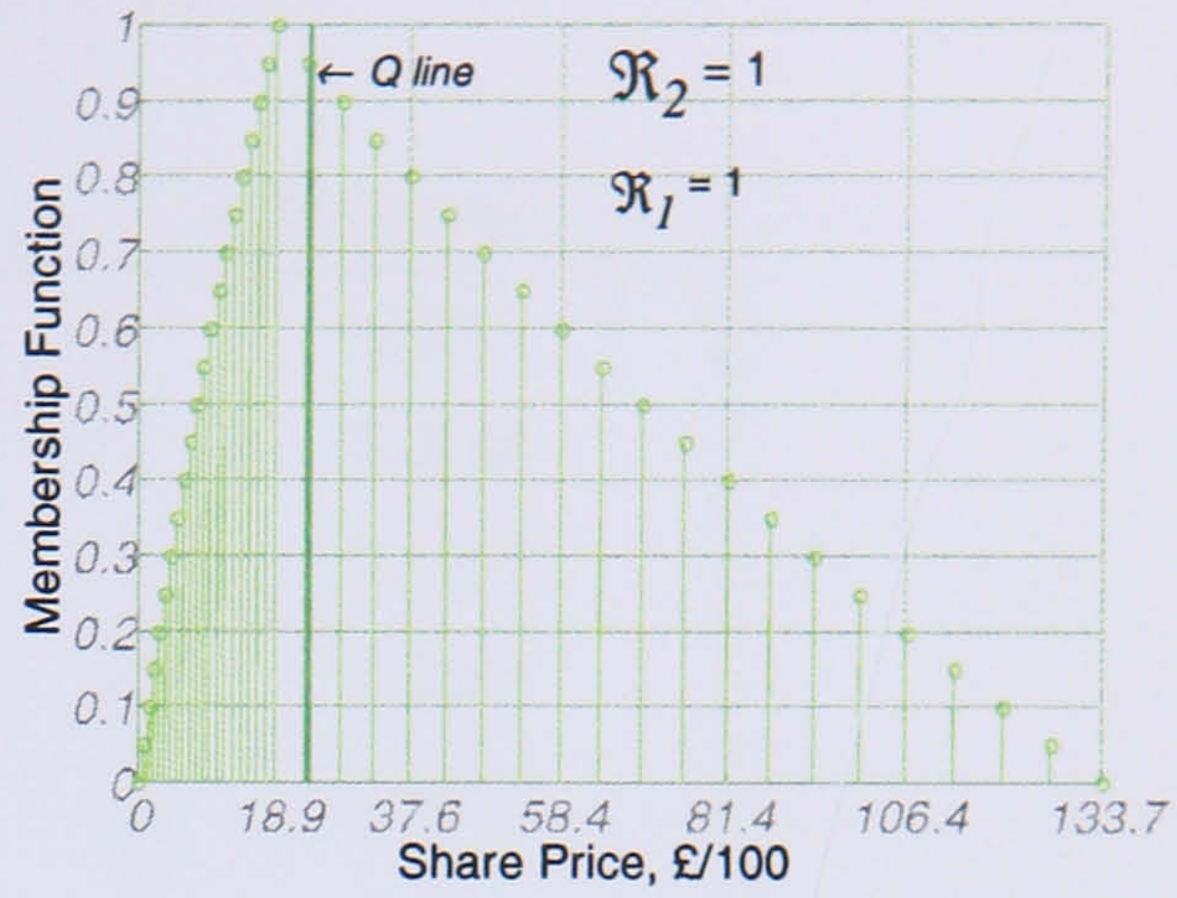


Figure 3.18b: BOOTS CO. - no robustness measure is assigned

- Although most of the companies are associated with the second case, this mode is not homogeneous and we should distinguish between different submodes. For example, the shares of PILKINGTON are a low robust asset at $t=0$, with $\Delta=0.257$ in table 3.2. Further, BP AMOCO with $\Delta=0.471$, TI GROUP with $\Delta=0.437$ and TRANSPORT DEVELOPMENT GROUP with $\Delta=0.339$ all show relatively low values. Next, TATE & LYLE is assigned a medium measure $\Delta=0.522$. On the other hand, figures 3.14 and 3.15 present BBA GROUP with $\Delta=0.710$ and BENTALLS with $\Delta=0.706$, correspondingly, as companies with relatively high robustness. In table 3.2, the same is valid for GOODWIN with $\Delta=0.846$, NORTHERN FOODS with $\Delta=0.796$, RANK GROUP with $\Delta=0.761$ and SCOTTISH & NEWCASTLE with $\Delta=0.813$. Finally, BASS in figure 3.13, and BRITISH AMERICAN TOBACCO, BUNZL, DIXONS GROUP, GREAT UNIVERSAL STORES, HANSON, MARKS & SPENCER, RMC GROUP, SMITH (WH) GROUP, SMITHS INDUSTRIES, TAYLOR WOODROW and WHITBREAD in table 3.2, all demonstrate high measures of $\Delta>0.9$.

Table 3.2: Evaluated robustness measure by company

company	*robustness* Δ
BASS	0.952
BBA GROUP	0.718
BENTALLS	0.706
BLUE CIRCLE INDUSTRIES	1.000
BOC GROUP	1.000
BOOTS CO.	–
BP AMOCO	0.471
BRITISH AMERICAN TOBACCO	0.981
BUNZL	0.959
COATS VIYELLA	1.000
DIXONS GROUP	0.943
GOODWIN	0.846
GREAT UNIVERSAL STORES	0.990
HANSON	0.934
INCHCAPE	–
LEX SERVICE	1.000
MARKS & SPENCER	0.989
NORTHERN FOODS	0.796
PILKINGTON	0.257
RANK GROUP	0.761
RMC GROUP	0.936
SAINSBURY (J)	–
SCOTTISH & NEWCASTLE	0.813
SMITH (WH) GROUP	0.995
SMITHS INDUSTRIES	0.945
TARMAC	1.000
TATE & LYLE	0.522
TAYLOR WOODROW	0.956
TI GROUP	0.437
TRANSPORT DEVELOPMENT GROUP	0.339
UNILEVER	–
UNITED BISCUITS HOLDINGS	1.000
WHITBREAD	0.959
WIMPEY (GEORGE)	–
WOLSELEY	1.000

- When the initial risk is $\Re_1 = 1$, and inevitably $\Re_2 = 1$, then no robustness value is assigned, as both the calculated $\Delta = 1$ or the choice of $\Delta = 0$ would not adequately correspond to the case. Figure 3.18 identifies BOOTS CO. in that state. This is also the case with the following companies - INCHCAPE, SAINSBURY (J), UNILEVER and WIMPEY (GEORGE).

3.4 Conclusion

The analysis in this Chapter further demonstrates how a fuzzy solution to a problem is more informative than the crisp one. We introduce two measures, the risk $\Re$ and the robustness Δ, characterising each asset as an investment choice. Though related, the two measures focus on different features of the investment choice. Thus an asset may be assigned a low risk value and a high robustness value, or a high risk measure and a low robustness measure, etc. The empirical results confirm this analysis, as the shares of the companies in our database represent risk and robustness measures in various combinations.

Each measure is associated with three qualitatively different modes: $\Re = 0$, $0 < \Re < 1$ and $\Re = 1$ for the risk value, and *'no value assigned'*, $0 \leq \Delta < 1$ and $\Delta = 1$ for the robustness measure. The middle modes are most interesting and versatile. The measures for the majority of assets take values there. To distinguish further quality within those modes, we introduce *'low'*, *'medium'* and *'high'* risk and robustness. On the other hand, the boundary modes have special meaning. Thus, $\Re = 0$ represents investments that are not associated with any loss at all levels of uncertainty, while $\Re = 1$ stands for those that will realise a loss at any level of uncertainty. Assets that keep a zero risk measure under the broader range of imprecision $\Re_2 = 0$ are assigned maximum robustness $\Delta = 1$, while those whose risk value under the narrower range is $\Re_1 = 1$ are assigned no robustness value.

The logical step forward is to suggest a ranking procedure that applies the two measures and orders the assets on a preference scale. This is the subject of the next Chapter.

Chapter 4: Asset Ranking Technique

4.1 Introduction

Here we design an asset ranking technique based on the conclusions in the preceding Chapters. In the sense of the formulated risk and robustness measures, lower values for the former and higher values for the latter are preferable. Assets are initially ordered according to their risk value. Then, among those with relatively close risk measures, assets with a higher robustness measure are given a higher ranking. Therefore, the technique involves two steps, risk and robustness related accordingly.

The first step starts with evaluating the risk measure for each asset under conditions and a range of imprecision that are closer to reality. Modelling conditions closer to reality depends on the choice of pricing factors. It involves decisions like whether to consider a factor as constant or time-varying, whether the effect of a factor can be approximated with a linear or a nonlinear relation, etc. Next, considering the fuzzy interval data, the choice relates to the calibrating procedure that will introduce a range of imprecision sufficiently broad to approach that involved in real problems.

The second step in the ranking technique starts with the choice of qualitative values for the robustness measure. This resolves the problem of ranking assets with close both risk and robustness measures. Two assets with close risk values only exchange their position in the ranking table if the slightly riskier asset has a qualitatively higher robustness.

The two-step technique provides a market agent with a soft asset ranking adjusted to data imprecision and market imperfections.

4.2 Time-Varying Return

The ranking technique starts with evaluating the risk measure under conditions approaching reality. In Chapter 2, the pricing formula is derived from the price-dividend relation under the assumption of a constant return, though compensating for that by modelling a fuzzy interval return. Here we will follow the argument that stock returns are time-varying rather than constant, as regression tests have convinced financial economists [135]. Thus a modified pricing formula is produced.

The assumption of time-varying returns increases the nonlinearity of the solution. Therefore, a loglinear approximation is required. The logarithmic prices, dividend yields and returns are denoted with the small letters $p_t \equiv ln(P_t)$, $dy_t \equiv ln(DY_t)$ and $r_t \equiv ln(1+R_t)$, correspondingly. Starting from the logarithmic price-dividend relation

$$r_{t+1} = p_{t+1} - p_t + ln\left(1 + e^{dy_{t+1}}\right) \ , \tag{4.1}$$

the following pricing formula is derived

$$p_0 = \sum_{t=1}^{T} \delta_1^{t-1}\left[(1-\delta_1)(dy_t + p_t) + \delta_2 - r_t\right] + \delta_1^T p_T \ . \tag{4.2}$$

Here δ_1 and δ_2 are parameters of linearisation,

$$\delta_1 = \frac{1}{1 - e^{\left(\sum_{t=1}^{T} dy_t / T\right)}} \quad , \quad 0 < \delta_1 < 1 \ , \tag{4.3}$$

$$\delta_2 = -ln(\delta_1) - (1 - \delta_1) ln\left(\frac{1}{\delta_1} - 1\right) \quad , \quad 0 < \delta_2 < 1 \ .$$

To obtain the fuzzy evaluation of the log asset price $\tilde{p}_0$, let us substitute in (4.2) the fuzzy intervals $\tilde{p}_t$, $\tilde{dy}_t$ and $\tilde{r}_t$, $1 \leq t \leq T$, for the crisp p_t, dy_t and r_t, correspondingly.

$$\tilde{p}_0 = \sum_{t=1}^{T} \delta_1^{t-1} \left[(1 - \delta_1)\left(\tilde{dy}_t + \tilde{p}_t \right) + \delta_2 - \tilde{r}_t \right] + \delta_1^T \tilde{p}_T \tag{4.4}$$

The parameters of linearisation are considered crisp. The function (4.2) satisfies proposition 2.1 in Chapter 2. Therefore, the extension principle solution for (4.4) will commute will level-cutting. Thus, the $\alpha - cuts$ and the membership function of the log asset price are obtained as follows.

$$\tilde{p}_0(\alpha) = \left[\underline{p_0(\alpha)} \ , \overline{p_0(\alpha)} \right], \ 0 \leq \alpha \leq 1 \ , \tag{4.5}$$

$$\underline{p_0(\alpha)} \ = \ \sum_{t=1}^{T} \delta_1^{t-1} \left[(1 - \delta_1)\left(\underline{dy_t(\alpha)} + \underline{p_t(\alpha)} \right) + \delta_2 - \overline{r_t(\alpha)} \right] + \delta_1^T \underline{p_T(\alpha)} \ ,$$

$$\overline{p_0(\alpha)} \ = \ \sum_{t=1}^{T} \delta_1^{t-1} \left[(1 - \delta_1)\left(\overline{dy_t(\alpha)} + \overline{p_t(\alpha)} \right) + \delta_2 - \underline{r_t(\alpha)} \right] + \delta_1^T \overline{p_T(\alpha)} \ .$$

$$\mu\left(x_{p_0} \middle| \tilde{p}_0 \right) = sup\left\{ \alpha \middle| x_{p_0} \in \tilde{p}_0(\alpha), 0 \leq \alpha \leq 1 \right\} \tag{4.6}$$

If we instead substitute in (4.3) the possibilistic variables $\breve{p}_t$, $\breve{dy}_t$ and $\breve{r}_t$ for p_t, dy_t and r_t, $1 \le t \le T$, and choose the possibility distributions as

$$Poss\left[\breve{p}_t = x_{p_t}\right] = \mu\left(x_{p_t} \middle| \tilde{p}_t\right),\ 1 \le t \le T \ . \tag{4.7}$$

$$Poss\left[\breve{dy}_t = x_{dy_t}\right] = \mu\left(x_{dy_t} \middle| \tilde{dy}_t\right),\ 1 \le t \le T \ ,$$

$$Poss\left[\breve{r}_t = x_{r_t}\right] = \mu\left(x_{r_t} \middle| \tilde{r}_t\right),\ 1 \le t \le T \ ,$$

then the possibility distribution of the evaluated asset price will correspond to the membership function in (4.6),

$$Poss\left[\breve{p}_0 = x_{p_0}\right] = \mu\left(x_{p_0} \middle| \tilde{p}_0\right) \ . \tag{4.8}$$

Once the modified pricing formula is derived and its fuzzy solution provided, the next decision concerns the calibration of the fuzzy interval data. We will use the procedure from Chapter 3 applied to each $\tilde{p}_t$, $\tilde{dy}_t$ and now each $\tilde{r}_t$ as well, as this will introduce quite a broad range of imprecision approaching real conditions. Moreover, we want to provide comparability with the results obtained so far for the risk and robustness measures. Accordingly, the calibration procedure is not directly used to produce new triangular membership functions for the logarithmic data, but is applied to the nonlogarithmic $\tilde{P}_t$, $\tilde{DY}_t$ and $\tilde{R}_t$, instead, as it is in Chapters 2 and 3. Indeed, the only difference with the calibration in Chapter 3 is that now a fuzzy interval for each R_t, $1 \le t \le T$, is considered rather than just one for $R = \sum_{t=1}^{T} R_t \Big/ T$. Then, the fuzzy intervals for $\tilde{p}_t$, $\tilde{dy}_t$ and $\tilde{r}_t$ are produced from $\tilde{P}_t$, $\tilde{DY}_t$ and $\tilde{R}_t$ through the following transformation,

$$\mu\left(x_{p_t}\,|\tilde{p}_t\right)=\mu\left(ln\left(x_{P_t}\right)\middle|\tilde{P}_t\right),\ 1\le t\le T\ , \tag{4.9}$$

$$\mu\left(x_{dy_t}\,|\tilde{dy}_t\right)=\mu\left(ln\left(x_{DY_t}\right)\middle|\tilde{DY}_t\right),\ 1\le t\le T\ ,$$

$$\mu\left(x_{r_t}\,|\tilde{r}_t\right)=\mu\left(ln\left(x_{R_t}\right)\middle|\tilde{R}_t\right),\ 1\le t\le T\ ,$$

resulting in nonlinear memberships functions for the logarithmic pricing factors, as illustrated with figure 4.1.

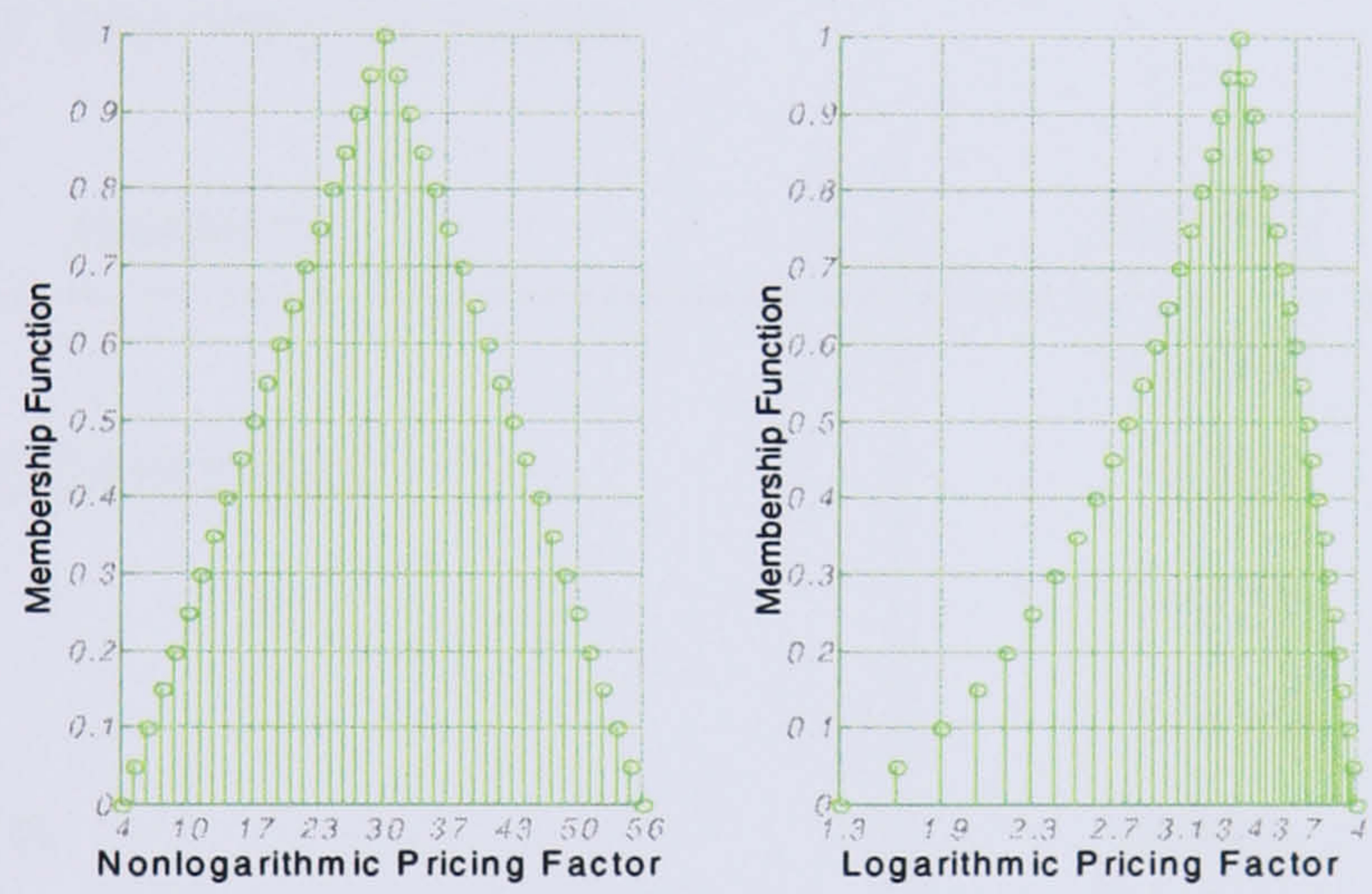


Figure 4.1: Transformation of data membership functions.

Now the risk measure $\Re_3$ for each asset is re-evaluated under realistic conditions. Comparing $\Re_3$ with the risk $\Re_1$ under a narrower range of imprecision, as presented in Section 3.2, the final robustness measure is produced,

$$\Delta_{31}=1-\left(\Re_3-\Re_1\right)\ . \tag{4.10}$$

We will only include in the ranking table assets with risk $\Re_3<1$, as those with $\Re_3=1$ are not attractive investments at $t=0$. Within the database of thirty-five companies, nineteen firms meet this requirement. Their graphics are included in Appendix A4, where $q = ln(Q)$ indicates the logarithmic current trading price. The results are summarised in table 4.1. The comparison of table 4.1 with table 3.2 reveals that the time-varying return induces the following relation for the robustness measure,

$$\Delta_{31} \leq \Delta_{21} \quad , 0 \leq \Delta_{31} \leq 1 \ , 0 \leq \Delta_{21} \leq 1 \ . \tag{4.11}$$

Accordingly, the risk values augment as $\Re_3 \geq \Re_2 \geq \Re_1$, $0 \leq \Re_1, \Re_2, \Re_3 \leq 1$.

Table 4.1: Evaluated risk and robustness measures by company under time-varying return

company	*risk* $\Re_3$	*robustness* Δ_{31}
BBA GROUP	0.696	0.549
BLUE CIRCLE INDUSTRIES	0.000	1.000
BOC GROUP	0.000	1.000
BP AMOCO	0.904	0.096
COATS VIYELLA	0.000	1.000
GOODWIN	0.925	0.496
HANSON	0.673	0.834
LEX SERVICE	0.000	1.000
NORTHERN FOODS	0.656	0.683
PILKINGTON	0.919	0.107
RMC GROUP	0.739	0.740
SCOTTISH & NEWCASTLE	0.775	0.712
SMITHS INDUSTRIES	0.959	0.625
TARMAC	0.000	1.000
TATE & LYLE	0.778	0.412
TI GROUP	0.664	0.336
TRANSPORT DEVELOPMENT GROUP	0.736	0.264
UNITED BISCUITS HOLDINGS	0.000	1.000
WOLSELEY	0.092	0.908

4.3 Two-Step Ranking: Definition and Empirical Results

The ranking procedure involves two steps, risk and robustness related, correspondingly. The risk measure has a higher priority to investors. Therefore, at the first step, the assets are ordered in accordance with their risk values. We consider here the final rather than the intermediate risk measures, as it is beneficial to hold a risk ranking produced under realistic conditions, though further assisted by the robustness measures. As an illustration, Table 4.2 presents the ranking of the nineteen companies from Section 4.2 according to the measure $\mathfrak{R}_3$.

A preparatory point for the second step is the choice of appropriate qualitative values for the robustness measure. For example, based on the dispersion of the measure Δ_{3l} in table 4.1, we introduce the following qualitative ranges,

$$\textit{robustness } \Delta_{3l}: \quad \overbrace{0 \cdots 0.300}^{low} \quad \overbrace{0.301 \cdots 0.525}^{relatively\ low} \quad \overbrace{0.526 \cdots 0.725}^{medium} \quad \overbrace{0.726 \cdots 0.950}^{relatively\ high} \quad \overbrace{0.951 \cdots 1}^{high} \tag{4.12}$$

The qualitative robustness value for each asset is included alongside its quantitative risk in table 4.2.

Table 4.2: Risk ranking

rank	*company*	*risk* $\Re_3$	*robustness* Δ_{31}
1	BLUE CIRCLE INDUSTRIES	0.000	high
1	BOC GROUP	0.000	high
1	COATS VIYELLA	0.000	high
1	LEX SERVICE	0.000	high
1	TARMAC	0.000	high
1	UNITED BISCUITS HOLDINGS	0.000	high
2	WOLSELEY	0.092	relatively high
3	NORTHERN FOODS	0.656	medium
4	TI GROUP	0.664	relatively low
5	HANSON	0.673	relatively high
6	BBA GROUP	0.696	medium
7	TRANSPORT DEVELOPMENT GROUP	0.736	low
8	RMC GROUP	0.739	relatively high
9	SCOTTISH & NEWCASTLE	0.775	medium
10	TATE & LYLE	0.778	relatively low
11	BP AMOCO	0.904	low
12	PILKINGTON	0.919	low
13	GOODWIN	0.925	relatively low
14	SMITHS INDUSTRIES	0.959	medium

Let us illustrate each qualitative value with a representative asset. In figure 4.2, the results for BP AMOCO describe a case of low robustness $\Delta_{31} = low$, as the risk value has increased from the minimum $\Re_1 = 0$ up to $\Re_3 = 0.904$. TATE & LYLE in figure 4.3 demonstrates relatively low robustness due to the significant change from $\Re_1 = 0.190$ to $\Re_3 = 0.778$. The graphics for SCOTTISH & NEWCASTLE in figure 4.4 indicates robustness $\Delta_{31} = medium$. Further, the qualitative value

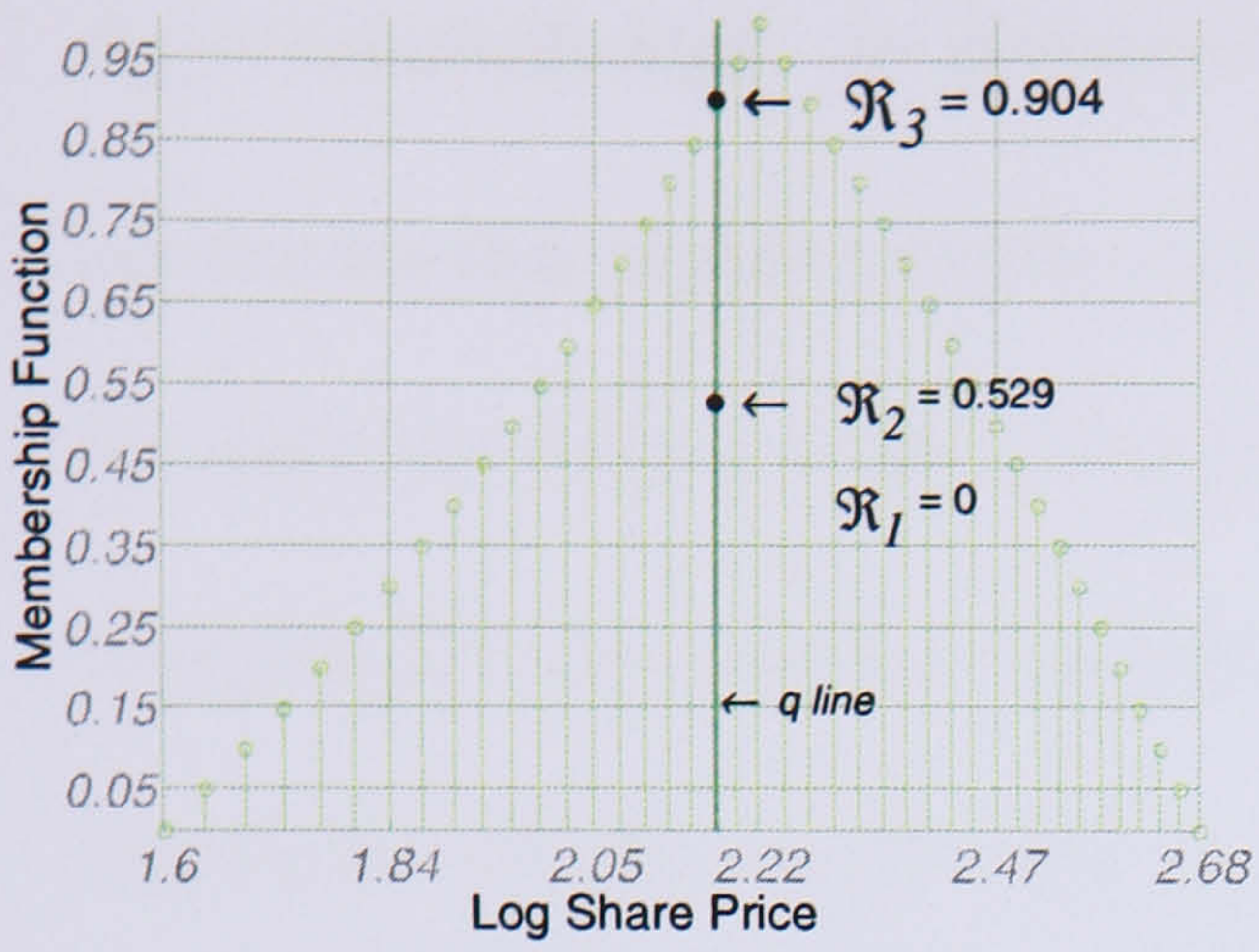


Figure 4.2: BP AMOCO - qualitative robustness value: $\Delta_{31} = low$

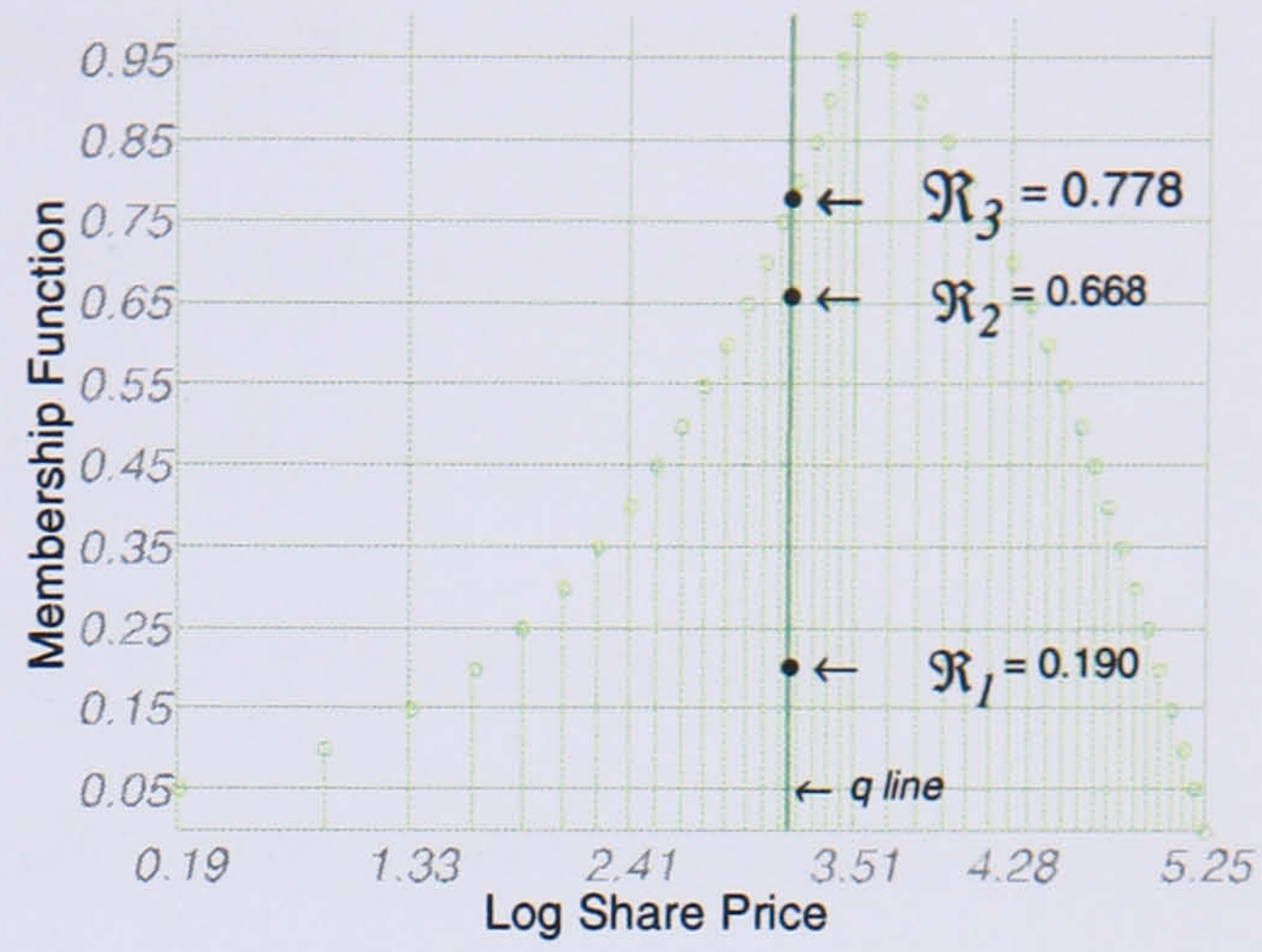


Figure 4.3: TATE & LYLE - qualitative robustness value: $\Delta_{31} = relatively\ low$

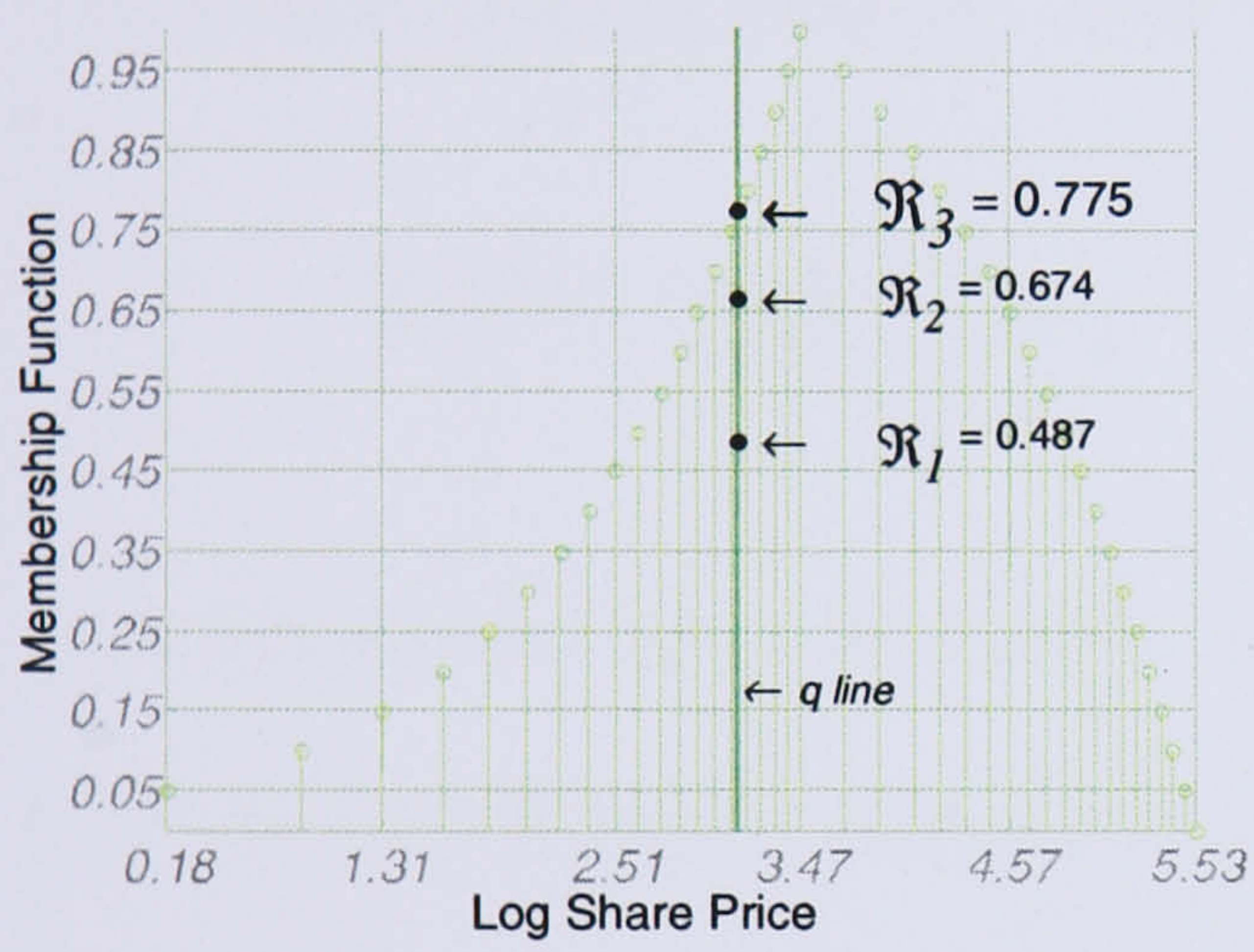


Figure 4.4: SCOTTISH & NEWCASTLE - qualitative robustness value: $\Delta_{31} = medium$

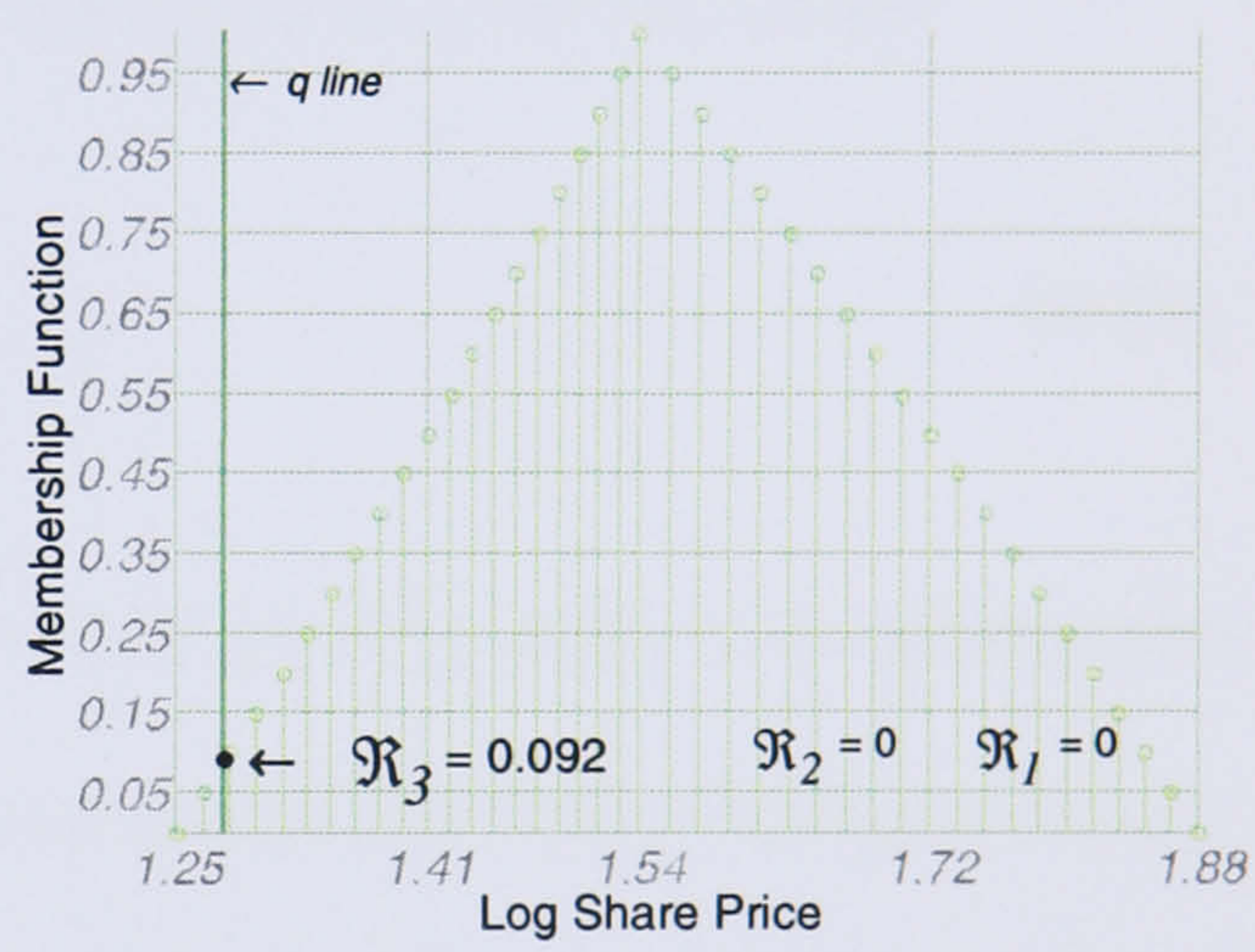


Figure 4.5: WOLSELEY - qualitative robustness value: $\Delta_{31} = relatively\ high$

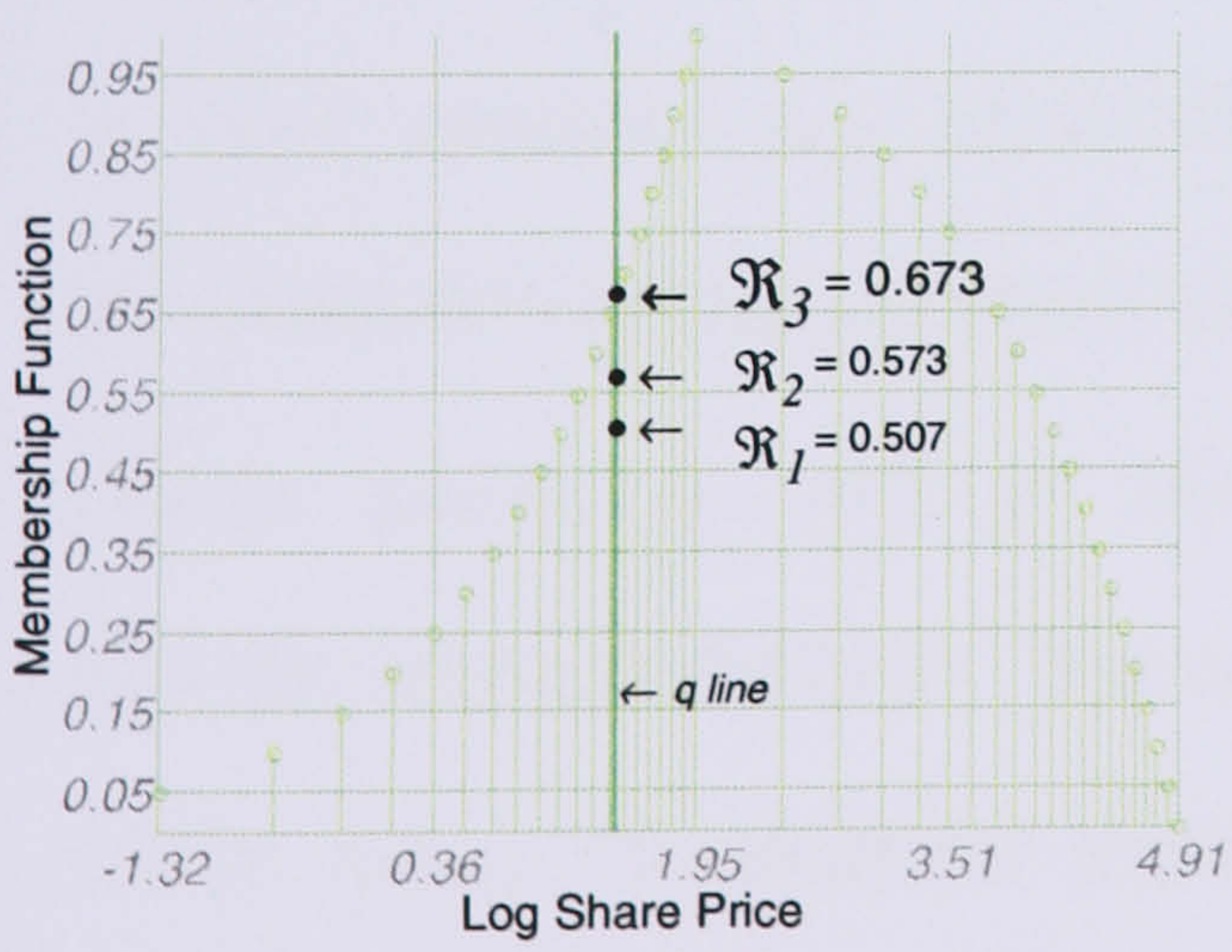


Figure 4.6: HANSON - qualitative robustness value: $\Delta_{31} = relatively\ high$

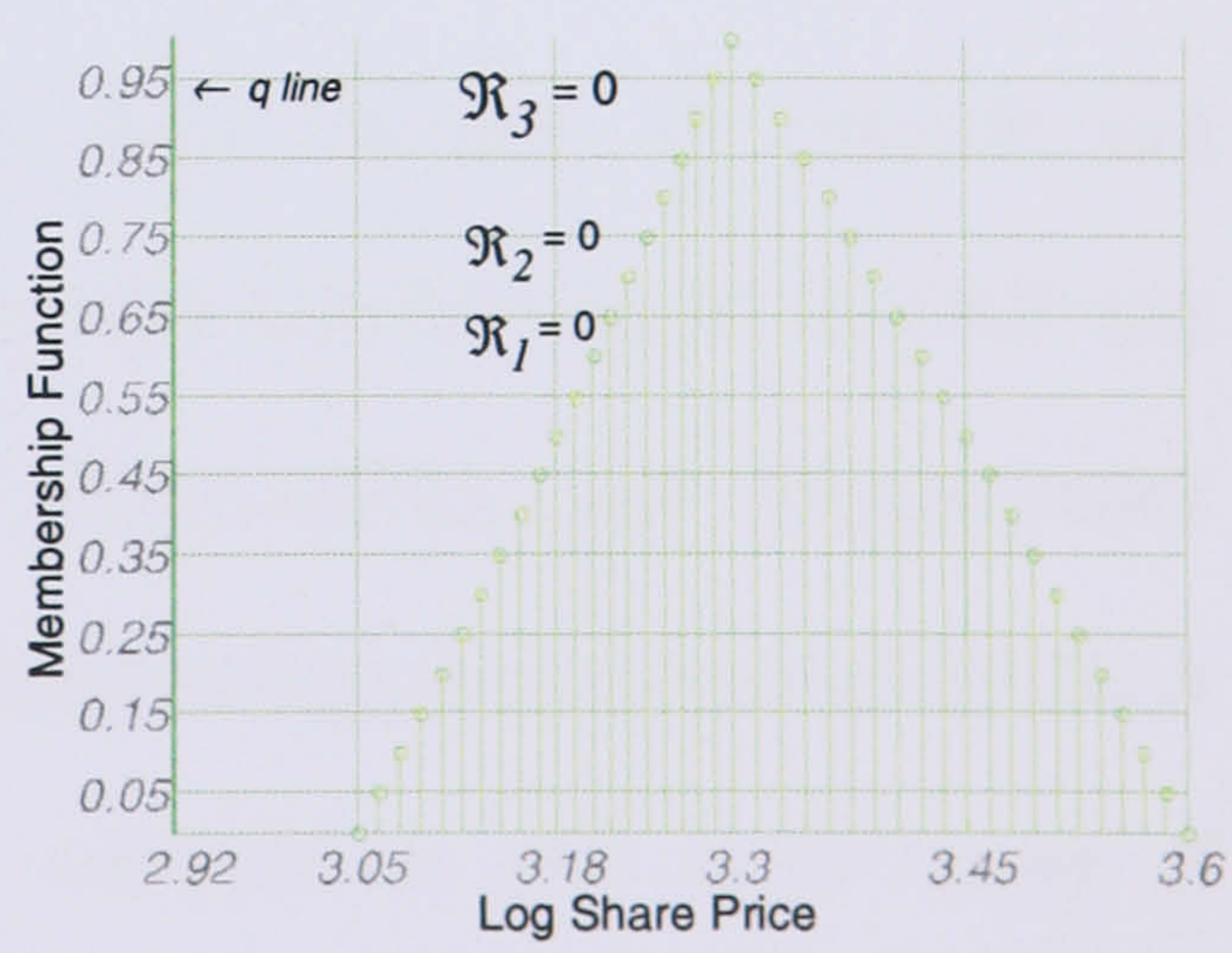


Figure 4.7: BOC GROUP- qualitative robustness value: $\Delta_{31} = high$

$\Delta_{31} = \textit{relatively high}$ is presented with two companies to illustrate the earlier conclusion that higher robustness does not necessarily imply lower risk. While figure 4.5 indicates only a slight increase in the minimum risk $\Re_1 = 0$ to $\Re_3 = 0.092$ for WOLSELEY, the graphics for HANSON in figure 4.6 demonstrates a stable and yet significant risk, $\Re_1 = 0.507$ and $\Re_3 = 0.673$. Finally, BOC GROUP in figure 4.7 is an example of a highly robust asset.

Next, the ranking from the first step of the procedure is adjusted in accordance with the robustness measure. Between two assets with close risk values, where

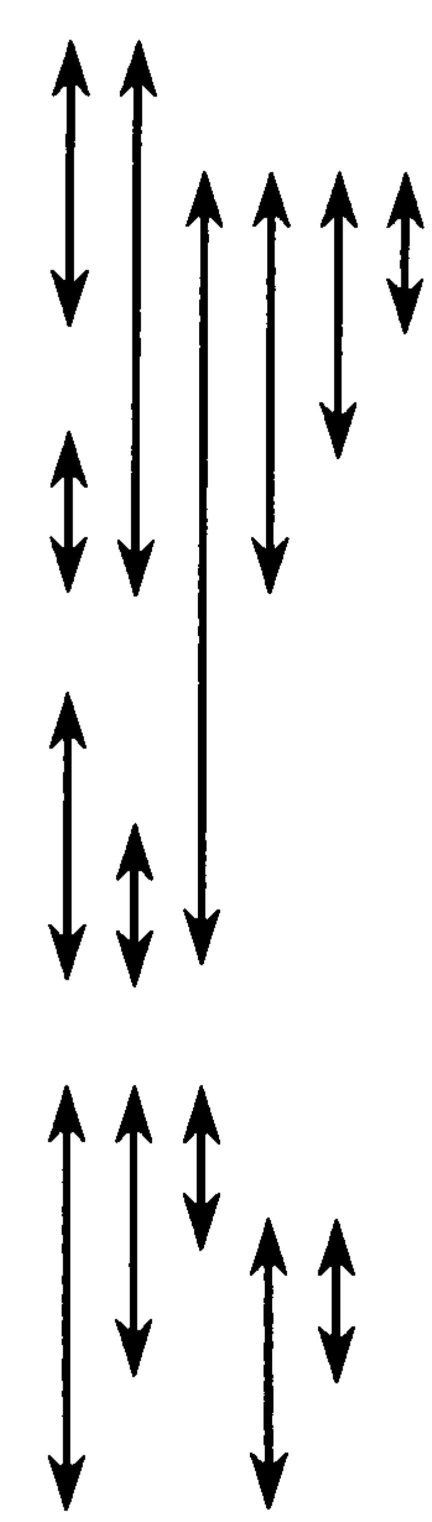
$$\left|\Re_3^{asset1} - \Re_3^{asset2}\right| \leq 0.1 \ , \tag{4.13}$$

the more robust one is preferable. In order to ensure the priority of the risk ranking, the two assets will only change places if they have qualitatively different robustness values. For example, the shares of HANSON with $\Delta_{31} = \textit{relatively high}$ and $\Re_3 = 0.673$ are preferable to the shares of NORTHERN FOODS with $\Delta_{31} = \textit{medium}$ and $\Re_3 = 0.656$ and TI GROUP with $\Delta_{31} = \textit{relatively low}$ and $\Re_3 = 0.664$. By analogy, RMC GROUP is more attractive than NORTHERN FOODS, BBA GROUP, TI GROUP and TRANSPORT DEVELOPMENT GROUP. Next, BBA GROUP and TI GROUP will change places, as well as SCOTTISH & NEWCASTLE and TRANSPORT DEVELOPMENT GROUP. The robustness adjusted ranking is in favour of TATE & LYLE before TRANSPORT DEVELOPMENT GROUP, as well as in favour of GOODWIN before BP AMOCO and PILKINGTON. Finally, SMITHS INDUSTRIES with $(\Delta_{31} = \textit{medium}, \Re_3 = 0.959)$ is preferable to GOODWIN with

$(\Delta_{31} = \mathit{relatively\ low}, \ \mathfrak{R}_3 = 0.925)$, BP AMOCO with $(\Delta_{31} = \mathit{low}, \ \mathfrak{R}_3 = 0.904)$ and PILKINGTON with $(\Delta_{31} = \mathit{low}, \ \mathfrak{R}_3 = 0.919)$. Table 4.2 presents the robustness adjusted ranking, where the modifications described in the lust column are applied to the risk ranking in table 4.1.

Table 4.3: Robustness adjusted ranking

rank	*company*	*risk* $\mathfrak{R}_3$	*robustness* Δ_{31}	*modification*
1	BLUE CIRCLE INDUSTRIES	0.000	high	
1	BOC GROUP	0.000	high	
1	COATS VIYELLA	0.000	high	
1	LEX SERVICE	0.000	high	
1	TARMAC	0.000	high	
1	UNITED BISCUITS HOLDINGS	0.000	high	
2	WOLSELEY	0.092	relatively high	
3	HANSON	0.673	relatively high	
4	RMC GROUP	0.739	relatively high	
5	NORTHERN FOODS	0.656	medium	
6	BBA GROUP	0.696	medium	
7	TI GROUP	0.664	relatively low	
8	SCOTTISH & NEWCASTLE	0.775	medium	
9	TATE & LYLE	0.778	relatively low	
10	TRANSPORT DEVELOPMENT GROUP	0.736	low	
12	SMITHS INDUSTRIES	0.959	medium	
11	GOODWIN	0.925	relatively low	
13	BP AMOCO	0.904	low	
14	PILKINGTON	0.919	low	



In the Section Original Contributions, we have described how the measures of asset risk and robustness we introduce with Definitions 3.1 and 3.3, correspondingly, relate to the crisp asset risk analysis. Particularly to the crisp capital asset pricing model and the arbitrage pricing theory. Considering the discussion there and the available information in our database, we can only suggest the following meaningful comparison. A crisp, yet all-inclusive risk measure is introduced as the potential profit, i.e. the difference between the crisp price-dividend evaluation and the observed asset price. The larger the difference the smaller the risk and the more preferable the asset.

Table 4.4: Crisp asset ranking

rank	*company*	*crisp measure*
1	LEX SERVICE	1.31
2	COATS VIYELLA	0.97
3	TI GROUP	0.53
4	TARMAC	0.51
5	HANSON	0.48
6	BBA GROUP	0.47
7	NORTHERN FOODS	0.46
8	BOC GROUP	0.38
9	UNITED BISCUITS HOLDINGS	0.37
10	RMC GROUP	0.36
11	TRANSPORT DEVELOPMENT GROUP	0.34
12	TATE & LYLE	0.32
13	SCOTTISH & NEWCASTLE	0.31
14	BLUE CIRCLE INDUSTRIES	0.27
15	WOLSELEY	0.26
16	PILKINGTON	0.12
17	GOODWIN	0.10
18	SMITHS INDUSTRIES	0.05
19	BP AMOCO	0.04

Table 4.4 does not make much sense on its own, as the crisp measure is not an effective risk measure. There is also no indication how reliable the crisp measure is. For example, the crisp measure can be high and there may still exist highly possible price values that are smaller than the observed price. This is the case with TI GROUP. For TI GROUP, the most possible profit is high at 0.53, which positions it high in the crisp ranking. However, the observed price is still well within the range of the evaluated fuzzy price and has a membership value of 0.664, which positions it low in the fuzzy ranking. Moreover, the fuzzy ranking indicates that the robustness value is relatively low, which suggests that the risk value is prone to increase. The latter is an argument that the crisp profit, though high, is unreliable. Another line of reasoning is the case when the crisp measure is low, however a tight membership function for the evaluated asset provides that the fuzzy risk measure is low and highly robust. This is the case with BLUE CIRCLE INDUSTRIES. The low crisp measure of 0.27 suggests a low most possible profit, however the fuzzy ranking indicates that the profit is highly reliable as the fuzzy risk measure is 0 and the robustness is high. As a result, though starting from the same crisp formula in both cases, the ranking in Table 4.3 and Table 4.4 is quite different. Therefore, the crisp ranking in the latter may be deceptive due to the ignored perception-based information. Finally, though it is ineffective, we have still chosen the crisp measure as closely related to the way the fuzzy measures are derived. Thus, it is slightly complementary rather than completely irrelevant. For example, once we are aware of the asset risk and the asset robustness, we may want to know the profit we risk. However, this could be better represented by a deffuzzificaion other than the most possible value, which is the crisp value. An alternative deffuzzification will effectively introduce a third measure derived from the fuzzy problem reformulation that will contribute to the analysis. Still it was hinted by the crisp measure.

The same approach can be applied starting from an arbitrage pricing theory model or a capital asset pricing formula, and the corresponding crisp and fuzzy results can be compared. This will require extracting a database for relevant economic factors and market indices, correspondingly.

4.4 Conclusion

The fuzzy asset evaluation provided in Chapter 2 is used as a basis for deriving the risk and robustness measures in Chapter 3 that are the core of the ranking procedure developed in this Chapter. The assets are initially ordered in relation to the risk values and then their positions are adjusted according to the robustness values. Thus the final ranking informs investors about the attractive less risky and highly robust assets.

This method can be applied to any crisp asset pricing technique. Therefore, in the general case, the proposed ranking procedure will be formulated as follows.

Algorithm 4.1: Reformulating asset evaluation and analysis

In order to rank assets in a preference scale of low risk and high robustness,

a. choose a crisp asset pricing model,

b. substitute fuzzy intervals for the pricing factors,

c. calibrate the membership functions of the fuzzy intervals to model broader factor imprecision than the probabilistic type of uncertainty,

d. evaluate the fuzzy asset price and the initial risk measure,

e. identify realistic conditions relevant to the chosen pricing model and transform the factor membership functions,

f. produce the final risk measure and the robustness measure,

g. rank the assets in relation to their final risk values,

h. adjust the ranking according to the robustness values.

The rational step ahead is to build an asset classifier based on the developed ranking technique. This is the topic of the next Chapter.

Chapter 5: Asset Classifier: Architecture

5.1 Introduction

The ranking technique from Chapter 4 will order a set of assets, and this will require the evaluation of all the assets and their corresponding measures. Let us consider the variety of market agents with their diverse ranges of preference. Once the ranking is completed, each agent will be able to make the decision which assets have acceptable characteristics. The decision of an individual investor will differ from the decision for a mutual fund, as their risk and robustness preference ranges are different.

On the other hand, we can implement the major conclusions of the ranking technique, without having to consider the whole set of assets. The technique favours less risky and highly robust assets, and each investment agent works within the preference ranges $\Re_{accepatbel} \leq \Re_{agent}$ and $\Delta_{acceptable} \geq \Delta_{agent}$. Therefore, it will be beneficial for him to be able to classify directly assets which fall within those ranges.

In this Chapter we identify the network structure that allows classifying assets with attractive characteristics. The classification is agent-dependent, as the same opportunity may be attractive to some investors while unacceptable for others. The major task is the analysis of the approximating capabilities of neural networks toward fuzzy functions.

5.2 Crisp Neural Network with Sign Restrictions on the Weights: Configuration and Empirical Results

The first step in building the soft classifier is to identify the artificial neural network (ANN) structure capable of approximating the fuzzy asset pricing function. We start with the configuration introduced in [44-46,145], as it relates to the fuzzy evaluation approach presented in Chapter 2. The structure has been shown to approximate some types of fuzzy functions that are α-cuts and interval arithmetic extensions of continuous real-valued functions. Furthermore, in order to achieve effective training, this configuration accommodates for a function being increasing or decreasing on some of its arguments, by imposing sign constraints on the network weights.

The three-layer feed-forward neural network to be employed is presented in figure 5.1. The input neurons distribute the input signals - the logarithmic pricing factors p_t, r_t and dy_t, $1 \le t \le T$ - to the neurons in the hidden layer. Therefore, the transfer functions of the input neurons are identity relations, I_{p_t}, I_{r_t}, and I_{dy_t}, $1 \le t \le T$, correspondingly, with no weights and no bias terms. Next, the hidden layer consists of m neurons with sigmoid transfer functions $\Omega_j, 1 \le j \le m$, where θ_j is the bias term in the function Ω_j. Furthermore, w_{tj} is the weight of the input p_t, u_{tj} is the weight of r_t and z_{tj} is the weight of dy_t in Ω_j.

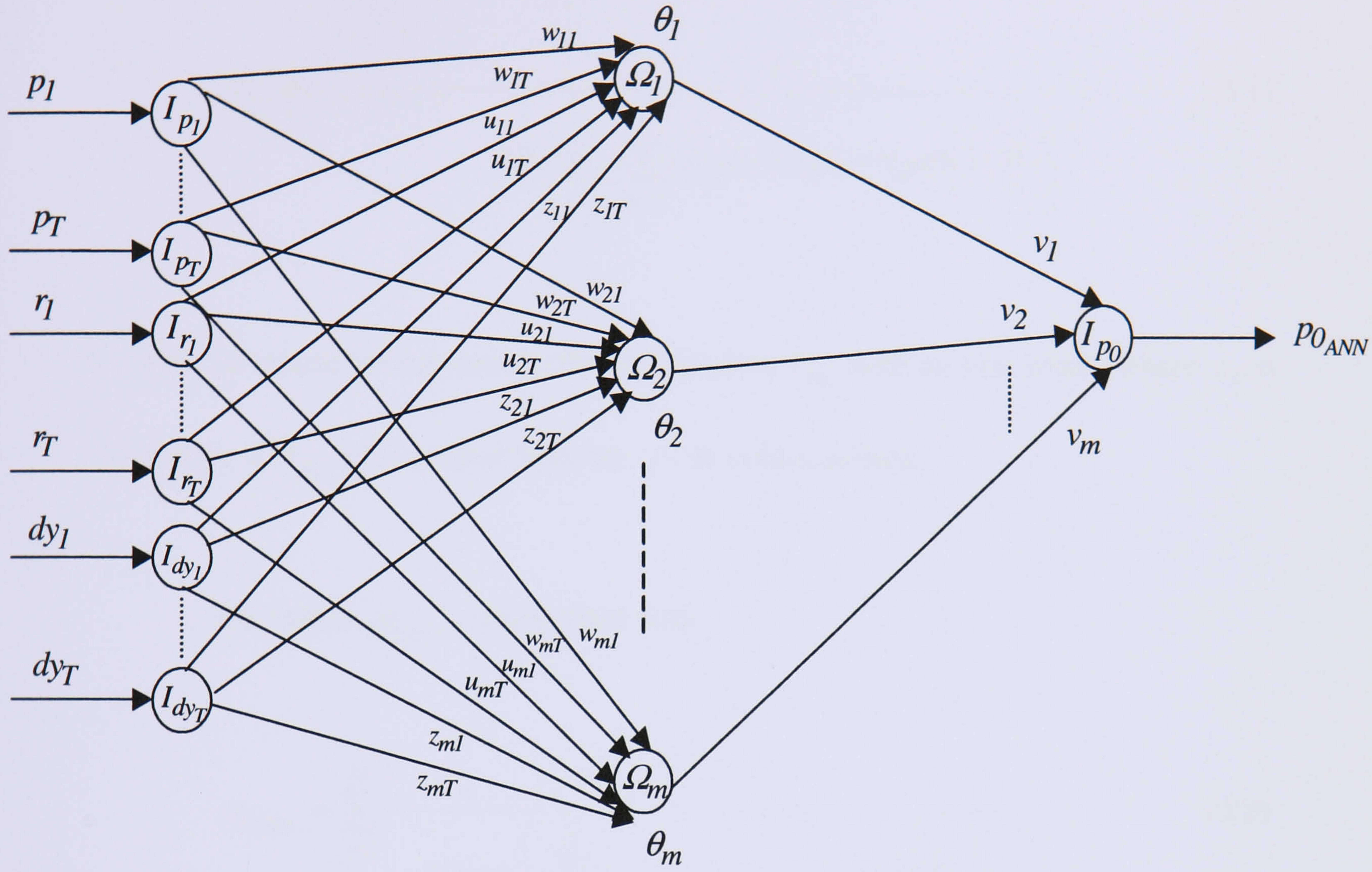


Figure 5.1: Crisp neural network approximating the fuzzy pricing function

- the inputs are the logarithmic pricing factors $p_1, \ldots, dy_T$;
- the transfer functions in the input layer are the identity relations $I_{p_1}, \ldots, I_{dy_T}$ with no weights and bias terms;
- the transfer functions in the hidden layer are the sigmoid relations $\Omega_1, \ldots, \Omega_m$ with weights $\{w_{1t}, u_{1t}, z_{1t} | 1 \leq t \leq T\}, \ldots, \{w_{mt}, u_{mt}, z_{mt} | 1 \leq t \leq T\}$ and bias terms $\theta_1, \ldots, \theta_m$;
- the transfer function of the output neuron is the identity relation I_{po} with weights $v_1, \ldots, v_m$ and no bias terms.

$$\Omega_j(p_1,\ldots,dy_T) = \frac{1}{1+exp\left(-\sum_{t=1}^{T}\left(w_{jt}p_t + u_{jt}r_t + z_{jt}dy_t\right) - \theta_j\right)} \tag{5.1}$$

Finally, the output neuron has an identity function I_{p_0} with no bias terms, where v_j is the weight in I_{p_0} of the signal from the $j-$th hidden neuron.

The output $p_{0_{ANN}}$ is described with

$$p_{0_{ANN}} = \sum_{j=1}^{m} \frac{v_j}{1+exp\left(-\sum_{t=1}^{T}\left(w_{jt}p_t + u_{jt}r_t + z_{jt}dy_t\right) - \theta_j\right)}\ , \tag{5.2}$$

and trained to approximate function (4.2). All the signals, weights and bias terms in the network are crisp real numbers, and the training involves crisp computation. However, the objective is to approximate the fuzzy asset price. Therefore, when the network is simulated with the $\alpha-cuts$ of the input pricing factors, it should produce the corresponding $\alpha-cut$ of the asset price:

$$\left[\underline{po_{ANN}(\alpha)}\ ,\overline{po_{ANN}(\alpha)}\right] \approx \left[\underline{po(\alpha)}\ ,\overline{po(\alpha)}\right] \ , \tag{5.3}$$

$$\sum_{j=1}^{m} \frac{v_j}{1+exp\left(-\sum_{t=1}^{T}\left(w_{jt}\underline{p_t(\alpha)}+u_{jt}\overline{r_t(\alpha)}+z_{jt}\underline{dy_t(\alpha)}\right)-\theta_j\right)} \approx$$

$$\approx \sum_{t=1}^{T} \delta_1^{t-1}\left[(1-\delta_1)\left(\underline{dy_t(\alpha)}+\underline{p_t(\alpha)}\right)+\delta_2-\overline{r_t(\alpha)}\right]+\delta_1^{T}\underline{p_T(\alpha)} \ ,$$

$$\sum_{j=1}^{m} \frac{v_j}{1+exp\left(-\sum_{t=1}^{T}\left(w_{jt}\overline{p_t(\alpha)}+u_{jt}\underline{r_t(\alpha)}+z_{jt}\overline{dy_t(\alpha)}\right)-\theta_j\right)} \approx$$

$$\approx \sum_{t=1}^{T} \delta_1^{t-1}\left[(1-\delta_1)\left(\overline{dy_t(\alpha)}+\overline{p_t(\alpha)}\right)+\delta_2-\underline{r_t(\alpha)}\right]+\delta_1^{T}\overline{p_T(\alpha)} \ .$$

It has been shown in [44-45,145] that

Proposition 5.1: *A layered feedforward crisp neural network with sigmoid transfer functions in the hidden layers can approximate the $\alpha-cuts$ of the fuzzy interval extension of a continuous function f with nonnegative or nonpositive arguments, if certain sign constraints are imposed on the network weights. The constraints are such that the partial derivatives of the network transfer function f_{ANN} towards each of the network inputs have the same signs as the partial derivatives of f towards each of its arguments. [46]*

In our case, $p_t = ln(P_t) \geq 0$ for $P_t \geq 1penny$, $r_t = ln(1+R_t) \geq 0$ for $R_t \geq 0\%$, and $dy_t = ln(DY_t) \geq 0$ for $DY_t \geq 0\%$, $1 \leq t \leq T$. The interpretation of the arguments suggests that it is the usual case. Furthermore, we consider functions (4.2) and (5.2), and derive the following conditions,

$$\partial po_{ANN} / \partial p_t = \partial p_0 / \partial p_t > 0 \ , \tag{5.4}$$

$$\partial po_{ANN} / \partial r_t = \partial p_0 / \partial r_t < 0 \ ,$$

$$\partial po_{ANN} / \partial dy_t = \partial p_0 / \partial dy_t > 0 \ .$$

There are two alternative sets of sign constraints that satisfy conditions (5.4) for the structure in figure 5.1,

$$w_{jt} > 0 , u_{jt} \leq 0 , z_{jt} > 0 , v_j > 0 , \quad 1 \leq t \leq T, 1 \leq j \leq m \ , \tag{5.5.1}$$

$$w_{jt} \leq 0 , u_{jt} > 0 , z_{jt} \leq 0 , v_j \leq 0 , \quad 1 \leq t \leq T, 1 \leq j \leq m \ . \tag{5.5.2}$$

Regarding the empirical results, we choose the Levenberg-Marquart algorithm to train the network from figure 5.1, as this is one of the fastest backpropagation techniques available. We further modify the algorithm to accommodate for the sign restrictions. Also, all the input vectors in the training set are presented to the network concurrently, as batching of concurrent inputs is computationally more efficient. The same thirty-five companies are considered, and to optimize computations further, the horizon of the pricing factors is shortened to twelve months $T = 12$. The fuzzy interval price $\tilde{p}_0$ is now evaluated with formulas (4.5) and (4.6) in January 1999. The supports $\left\{ \left[\underline{p_0(\alpha)} \ , \overline{p_0(\alpha)} \right]_c \middle| \alpha = 0, 1 \leq c \leq 35 \right\}$ of $\tilde{p}_0$ for all companies c are provided in Appendix A5 and show that

$$\min_{c}\left(\underline{p_0(0)}\right)_c > 0 \ . \tag{5.6}$$

Thus, constraints (5.5.1) are applied, as $v_j > 0$ will produce a positive network output. The training involves a set of crisp vectors comprising elements of the supports of the fuzzified data. It yields a minimum mean square error of the output $p_{0_{ANN}}$ compared with the corresponding solution p_0 of (4.2). On the other hand, the testing involves a set of interval vectors that are $\alpha-cuts$ of the fuzzified data. Therefore, the interval approximating capability of the trained network is evaluated by comparing the interval output with the $\alpha-cuts$ of $\tilde{p}_0$ obtained with (4.5) and (4.6),

$$\varepsilon_{interval} = \max_{1\le i\le i_{test}} \left|network_i - target_i\right| = \tag{5.7}$$

$$= \max\left(\max_{\alpha_i}\left|\underline{p_{0_{ANN}}(\alpha_i)} - \underline{p_0(\alpha_i)}\right|, \max_{\alpha_i}\left|\overline{p_{0_{ANN}}(\alpha_i)} - \overline{p_0(\alpha_i)}\right|\right), \alpha_i \in \{0,\dots,1\} \ ,$$

where i_{test} is the number of test vectors.

We have successfully trained a network for each asset. The training set consists of 362 crisp vectors in each case and the goal is $mse \le 1{\bullet}10^{-4}$. To avoid overfitting, the network with the minimum number of hidden neurons $m \in N$ is chosen every time and then tested with a set of 39 interval vectors aiming at $\varepsilon_{interval} \le 2.5{\bullet}10^{-2}$. No element of the training set is a boundary of an interval element of the test set. The results are summarised in table 5.1.

Table 5.1: Crisp neural network performance

company	$mse\left(training\right), 10^{-4}$	$\varepsilon_{interval}\left(test\right), 10^{-2}$
BASS	0.988	1.29
BBA GROUP	0.877	0.99
BENTALLS	0.862	0.96
BLUE CIRCLE INDUSTRIES	0.140	0.86
BOC GROUP	0.745	0.95
BOOTS CO.	0.010	0.18
BP AMOCO	0.035	0.19
BRITISH AMERICAN TOBACCO	0.349	0.62
BUNZL	0.330	0.98
COATS VIYELLA	0.966	2.10
DIXONS GROUP	0.236	0.55
GOODWIN	0.958	2.10
GREAT UNIVERSAL STORES	0.608	1.37
HANSON	0.054	0.36
INCHCAPE	0.939	1.39
LEX SERVICE	0.012	0.14
MARKS & SPENCER	0.260	1.15
NORTHERN FOODS	0.019	0.21
PILKINGTON	0.161	0.67
RANK GROUP	0.651	1.23
RMC GROUP	0.148	0.68
SAINSBURY (J)	0.316	0.63
SCOTTISH & NEWCASTLE	0.956	1.41
SMITH (WH) GROUP	0.191	0.89
SMITHS INDUSTRIES	0.423	1.86
TARMAC	0.980	1.41
TATE & LYLE	0.070	0.47
TAYLOR WOODROW	0.454	1.40
TI GROUP	0.385	1.25
TRANSPORT DEVELOPMENT GROUP	0.020	0.39
UNILEVER	0.121	0.54
UNITED BISCUITS HOLDINGS	0.967	1.10
WHITBREAD	0.925	2.08
WIMPEY (GEORGE)	0.234	1.52
WOLSELEY	0.058	0.48

The fact that it is possible to train a crisp network to present an asset's fuzzy price proves that neural networks with sign restrictions on their weights have some interval approximation capabilities. Again the result is not restricted to the pricing function used here. Once trained, a network can be simulated any time the information related to the asset is subject to change. For example, an agent believes he has obtained more reliable information, then the network is simulated with modified data membership functions.

A hybrid part attached to the regular network from figure 5.1 will further produce the relevant risk and robustness measures. However, we will not advance in that direction yet, as our objective was to train a network over all assets. Though using various number m of hidden neurons and introducing a second hidden layer, it was not possible to achieve that goal. We consider two reasons. First, although the proposition 5.1 has been theoretically proven [46], it has only been practically applied to simpler fuzzy functions and networks with no more than two input neurons [44-46,145]. Second, the parameters of linearization δ_1 and δ_2 are asset dependent, though determined by the network inputs. This further complicates the character of the function and may entangle training across assets.

We have considered reasonable to approach the complex task of network approximation of fuzzy functions with the structure presented in this section, as it relates to the level-cutting evaluation technique introduced in the previous Chapters, and has been successfully applied to simpler functions. Though some partial result is achieved, the conclusion is that the objective of discriminating between different assets requires exploration of further network structures. In the next section, the approximating capability of a fuzzy neural network is investigated.

5.3 Hybrid Fuzzy Neural Network: Configuration

The research interest in ANN is motivated to some extent by the fact that they are universal approximators for crisp functions [146]. Then, the anticipated question is whether the fuzzy extensions of the networks are also universal approximators for the fuzzy extensions of the functions. Therefore, the problem regarding the approximating capabilities of fuzzy neural networks (FNN), both regular and hybrid, has been studied recently [45,99-103,147]. Though regular crisp networks represent continuous crisp functions to an arbitrary degree of accuracy, regular fuzzy networks do not demonstrate corresponding approximating qualities towards continuous fuzzy functions [100,147]. However, regular FNN (RFNN) are universal approximators of continuous fuzzy-valued functions [103]. In this section, we will identify the regular structure capable of representing the asset pricing function to an arbitrary degree of accuracy, and will subsequently configure the hybrid FNN (HFNN) that will discriminate among assets and classify those that are less risky and highly robust.

Let us introduce relevant notations. RFNN are fully fuzzified multilayer feed-forward networks, where fuzzy arithmetic is used to compute the output of neurons with standard transfer functions, e.g. identity or sigmoid. On the other hand, the choice of neuron-transfer functions in a hybrid FNN (HFNN) is more versatile and does not necessarily involve fuzzy arithmetic. The hybrid networks may include regular parts. Next, $\mathbb{N}$ denotes the set of natural numbers, $\mathbb{R}$ is the set of all real numbers, and $\Im(\mathbb{R})$ stands for the set of all fuzzy intervals defined on $\mathbb{R}$. A crisp function is a mapping from and to $\mathbb{R}$, $f_{crisp}(y)$: $\mathbb{R} \rightarrow \mathbb{R}$. If the argument is defined on a compact set $S \subset \mathbb{R}$, then the mapping is modified to $f_{crisp}(y): S \to \mathbb{R}$, and in the multivariable case takes the form $f_{crisp}(y_1,\dots,y_k): S_1,\dots,S_k \to \mathbb{R}$. Furthermore, a fuzzy function $\tilde{f}_{fuzzy}(\tilde{y})$ is a projection from and to the fuzzy-interval set $\Im(\mathbb{R})$, $\tilde{f}_{fuzzy}(\tilde{y}): \Im(\mathbb{R}) \to \Im(\mathbb{R})$. Alternatively, a fuzzy-valued function $\tilde{f}_{fuzzy-valued}(y)$, is a projection from the real-number set to the fuzzy-interval set, $\tilde{f}_{fuzzy-valued}(y): \mathbb{R} \to \Im(\mathbb{R})$. When considering the multivariable case and the domain area of the variables, the mapping transforms into

$$\tilde{f}_{fuzzy-valued}(y_1,\dots,y_k): S_1 \times \dots \times S_k \to \Im(\mathbb{R}) \ , S_i \subset \mathbb{R}, 1 \le i \le k \quad . \tag{5.8}$$

Finally, let $\wp\left\{RFNN \middle| \tilde{f}_{RFNN\,fuzzy-valued}(y_1,\dots,y_k)\right\}$ stands for a class of RFNN with fuzzy-valued network-transfer functions, where

$$\wp_{min}\left\{FNN \middle| \tilde{f}_{FNN\,fuzzy-valued}(y_1,\dots,y_k)\right\} \tag{5.9}$$

is the subclass with minimum fuzzification.

Based on the conclusions in [103] for the single-variable case, we will formulate with (5.10) and (5.11) types of continuous multivariable fuzzy-valued (CMFV) functions, and propose in (5.12) a minimum-fuzzification network structure capable of representing those functions to an arbitrary degree of accuracy. The proof is analogous to that provided in [103]. Let $f_{crisp}(y_1,\dots,y_k): S_1 \times \dots \times S_k \to \mathbb{R}$ is a multivariable continuous crisp function on the compact sets $S_i \subset \mathbb{R}, 1 \le i \le k$. If f$\tilde{a}\in\Im(\mathbb{R})$ is a uzzy interval and $\tilde{f}_{fuzzy-valued}(y_1,\dots,y_k): S_1\times\dots\times S_k\to\Im(\mathbb{R})$ is a multivariable fuzzy-valued function, where

$$\tilde{f}_{fuzzy-valued}(y_1,\dots,y_k)=(\tilde{a}\bullet f_{crisp})(y_1,\dots,y_k)= \tag{5.10}$$

$$=\tilde{a}\bullet f_{crisp}(y_1,\dots,y_k),\; y_i\in S_i, 1\le i\le k\ ,$$

then $\tilde{f}_{fuzzy-valued}(y_1,\dots,y_k)$ is continuous on $S_1\times\dots\times S_k$. Furthermore, if $\tilde{f}_j(y_1,\dots,y_k)$, $1\le j\le K$, are CMFV functions of type (5.10), then the linear combination

$$\tilde{f}_{fuzzy-valued}=\sum_{j=1}^{K}\left(\pm\tilde{f}_j(y_1,\dots,y_k)\right)=\sum_{j=1}^{K}\left(\pm\tilde{a}_j\bullet f_j(y_1,\dots,y_k)\right)\ , \tag{5.11}$$

$$\tilde{a}_j\in\Im(\mathbb{R}), 1\le j\le K\quad ,\quad y_i\in S_i, S_i\subset\mathbb{R}, 1\le i\le k\ ,$$

is also a CMFV function on $S_1\times\dots\times S_k$. It can be proven that the following class $\wp$ of four-layer feed-forward RFNNs is a universal approximator for the function from definition (5.11),

$$\wp\left\{ RFNN \;\middle|\; \tilde{f}_{RFNN}(y_1,\dots,y_k)=\sum_{i=1}^{n}\tilde{v}_i\left|\;\sum_{j=1}^{m}\frac{\tilde{\ell}_{ij}}{1+exp\left(-\sum_{l=1}^{k}\tilde{w}_{jl}y_l-\tilde{\theta}_j\right)}\right)\right. ,$$

$$\left. n,m\in\mathbb{N}\quad ,\quad y_l\in\mathbb{R}\quad ,\quad \tilde{v}_i,\tilde{\ell}_{ij},\tilde{w}_{jl},\tilde{\theta}_j\in\Im(\mathbb{R})\quad ,\quad \tilde{f}_{RFNN}:\mathbb{R}\to\Im(\mathbb{R})\right\} .$$

The network structure includes sigmoid transfer functions with fuzzy shift terms $\tilde{\theta}_j$ in the first hidden layer, and identity transfer functions with no shift terms in the input, the second hidden and the output layer. Furthermore, $\tilde{w}_{jl}$ are the fuzzy weights on the signals from the input to the first hidden layer, $\tilde{\ell}_{ij}$ are the fuzzy weights from the fist to the second hidden layer, and $\tilde{v}_i$ are the fuzzy weights towards the output layer. Finally, $\tilde{f}_{RFNN}(y_1,\dots,y_k)$ is the fuzzy-valued network-transfer function. In order to reduce the complexity of the training task, the minimum-fuzzified structure is required. The following subclass $\wp_{min}$ of $\wp$ can be derived in (5.12), that retains approximating capabilities towards CMFV functions of type (5.11). The weights and the shift terms $\tilde{\ell}_{ij}, \tilde{w}_{jl}, \tilde{\theta}_j \in \Im(\mathbb{R})$ are restricted to be real numbers $\ell_{ij}, w_{jl}, \theta_j \in \mathbb{R}$, with the exception of the weights to the output layer $\tilde{v}_i \in \Im(\mathbb{R})$ which are triangular fuzzy intervals, thus simplifying the fuzzy-value network-transfer function $\tilde{f}_{RFNN}(y_1,\dots,y_k)$.

$$\wp_{min}\left\{RFNN \mid \tilde{f}_{RFNN}(y_1,\dots,y_k)=\sum_{i=1}^{n}\tilde{v}_i\left(\sum_{j=1}^{m}\frac{\ell_{ij}}{1+exp\left(-\sum_{l=1}^{k}w_{jl}y_l-\theta_j\right)}\right),\right.$$

$$\left. n,m\in \mathbb{N},\ \ell_{ij},w_{jl},\theta_j,y_l\in\mathbb{R},\ \tilde{v}_i,\in\Im(\mathbb{R}),\ \tilde{f}_{RFNN}:\mathbb{R}\to\Im(\mathbb{R})\right\} \quad (5.12)$$

Therefore, if the pricing functions are presented in the form (5.11), then we will be able to build the hybrid fuzzy network in figure 5.2, which comprises regular parts and is able to discriminate between assets according to the general rules of the ranking technique introduced in Chapter 4.

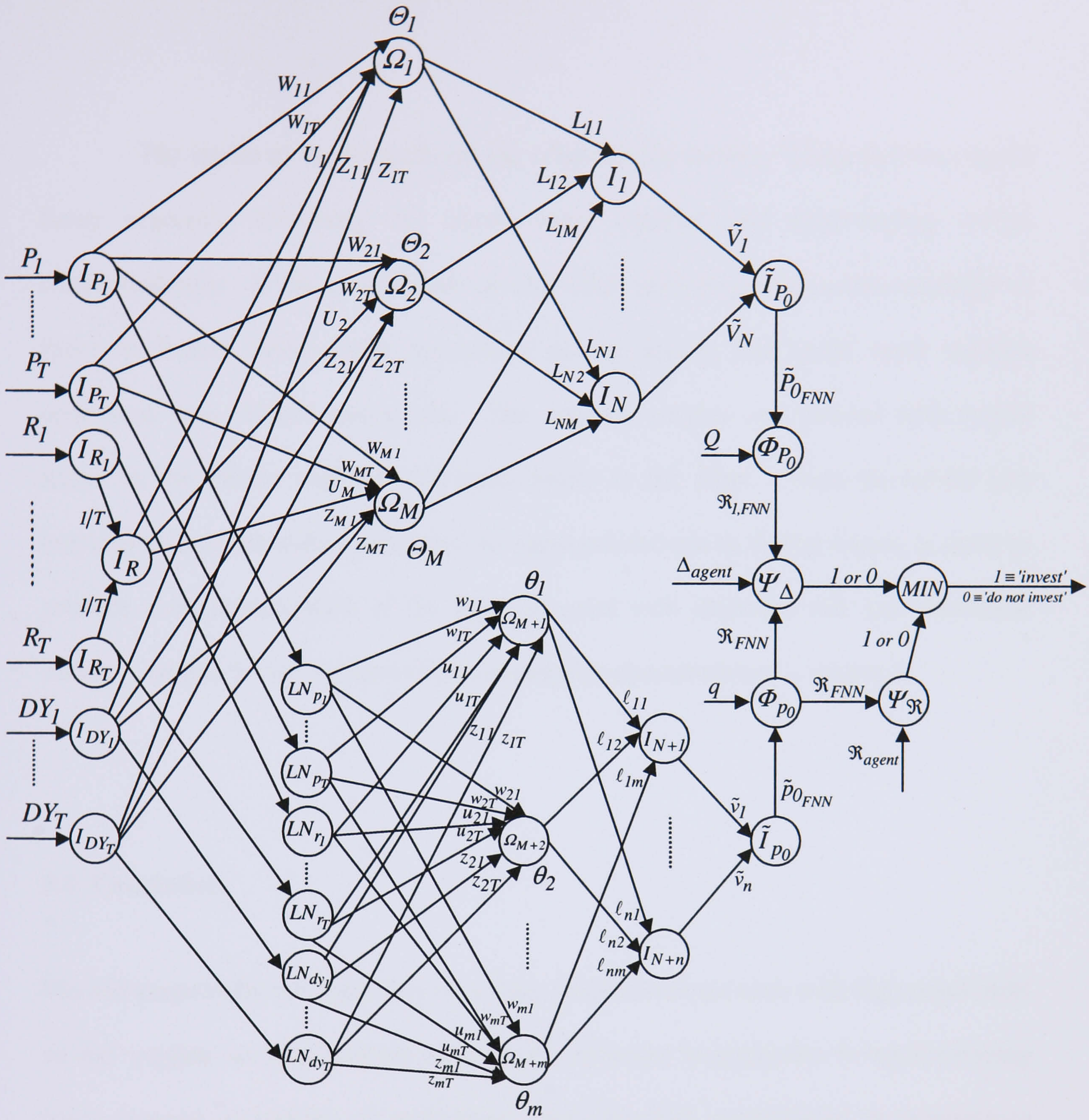


Figure 5.2: Fuzzy neural network structure classifying assets with attractive risk and robustness values

- the network inputs are the level (nonlogarithmic) pricing factors $P_1,\ldots,DY_T$, and the transfer functions in the input layer $I_{P_1},\ldots,I_{DY_T}$ are identity with no weights or bias terms, while the node I_R produces the average of the factors $R_1,\ldots,R_T$;
- the transfer functions in the hidden layer $LN_{p_1},\ldots,LN_{dy_T}$ are logarithmic with no weights or bias terms;
- the transfer functions in the hidden layer $\Omega_1,\ldots,\Omega_M$, $\Omega_{M+1},\ldots,\Omega_{M+m}$ are sigmoid with weights $\{W_{1t},U_{1t},Z_{1t}\},\ldots,$ $\{W_{Mt},U_{Mt},Z_{Mt}\}$, $\{w_{1t},u_{1t},z_{1t}\},\ldots,\{w_{mt},u_{mt},z_{mt}\}$, $1 \le t \le T$, and bias terms $\Theta_1,\ldots,\Theta_M$, $\theta_1,\ldots,\theta_m$, correspondingly;
- the transfer functions in the hidden layer $I_1,\ldots,I_N$, $I_{N+1},\ldots,I_{N+n}$ are identity with no bias terms and with weights $L_{11},\ldots,L_{NM}$, $l_{11},\ldots,l_{nm}$ on the connections from the neurons $\Omega_1,\ldots,\Omega_M$, $\Omega_{M+1},\ldots,\Omega_{M+m}$;
- the hidden layer $\tilde{I}_{P_0}$, $\tilde{I}_{p_0}$ has fuzzy identity transfer functions with fuzzy weights $\tilde{V}_1,\ldots,\tilde{V}_N$, $\tilde{v}_1,\ldots,\tilde{v}_n$ and no bias terms;
- the transfer functions in nodes Φ_{P_0}, Φ_{p_0} produce approximations of the initial and the final risk measure, correspondingly;
- the nodes $\Psi_\Re$, Ψ_Δ have hard limit transfer functions with agent-dependent thresholds $\Re_{agent}$, Δ_{agent};
- the output neuron produces the network advice on the asset: 1 (attractive risk and robustness values), 0 (disqualified asset).

The inputs to the network are the crisp pricing factors. There are two regular fuzzy segments evaluating the asset under constant and time-varying return, correspondingly. In the general case or with other pricing functions, this translates as fuzzy evaluation under some restrictions on the factors and under more realistic conditions with relaxed restrictions. The network weights are denoted with capital letters in the former segment and small letters in the latter. Then the hybrid part employs the current trading price and the agent preferences as further inputs, in order to produce a recommendation of the type 'an asset with attractive risk and robustness values' – output 1 – or 'an asset with unacceptable characteristics' – output 0.

5.4 Conclusion

We test empirically the capability of a crisp feedforward network with sign constraints on the weights, as the literature suggests the structure is applicable to approximating fuzzy interval extensions of continuous functions with nonnegative or nonpositive arguments. Some partial result is achieved, however the objective of discriminating between different assets requires exploration of further types of networks. We direct our attention toward fuzzy networks and identify the regular fuzzy network structure able to represent continuous multivariable fuzzy-valued functions. Thus no sign restrictions are imposed on the arguments of the pricing function, which works toward generalising the approach. Considering fuzzy-valued rather than fuzzy functions suggests that the fuzzification of the factors is incorporated into the information processing abilities of the network, which is expected to have a positive effect on the user of the decision-support technique. Now he will work directly with the crisp market data as inputs to the network – no data preprocessing and fuzzification is necessary once the network is trained – and will get as output an advice on the asset based on fuzzy logic.

Having a decision about the classifier structure, the next step is to develop a weight training strategy. This is the subject of the next Chapter.

Chapter 6: Risk Classifier: Evolutionary Training Algorithm

6.1 Introduction

In Chapter 5, the regular fuzzy networks approximating the pricing functions under different conditions, or two functions involving different number of pricing factors, are embedded into a hybrid network to produce asset classification according to a general version of the criterion introduced in Chapter 4. Here we will focus on the risk classification of assets and develop a network-weight training strategy based on evolutionary computing.

The preliminary step includes outlining the structure of the risk classifier, and modifying the pricing relation to a fuzzy-valued function. The next important step involves representing the network weights and bias terms into the chromosome, and deciding the search operators of crossover and mutation. Finally, an evolutionary algorithm is designed based on the general concepts of divide-and-conquer evolution and incremental evolution. These concepts are also employed by bidirectional incremental evolution (BIE) [48], however the two-level exploratory algorithm we suggest here is completely new as a design. The algorithm not only trains the weights and the bias terms but also adapts the learning process to the training sample and the random initial condition toward more efficient training. If the general framework for evolving artificial neural networks [47] is considered, then we may argue that the algorithm concerns the inner two layers, the evolution of weights and the evolution of learning, while the decision about the architecture has been taken on the basis of the reasoning in the previous Chapter. The quality of the algorithm can be recognised when compared with single-level evolution.

6.2 Risk Classifier: Structure and Function

We will focus on asset classification according to the risk measure from Chapter 3 and the agent's risk preference. The classifier architecture is correspondingly reduced to the following network.

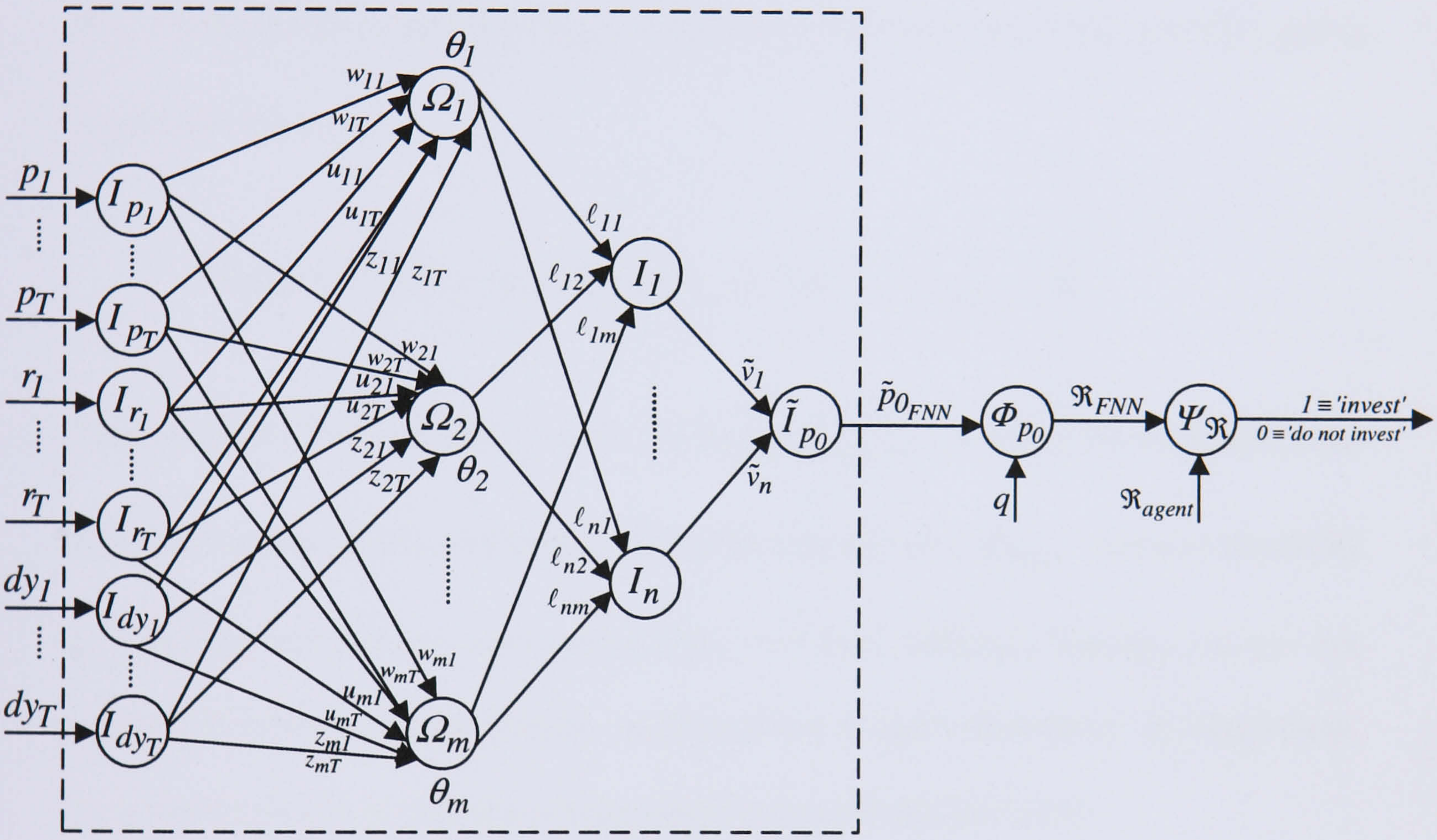


Figure 6.1: Risk classifier architecture

- the regular fuzzy segment has two hidden layers, with the crisp logarithmic pricing factors as inputs and the evaluated fuzzy logarithmic price as output; the weights and the bias terms in the segment are crisp except the fuzzy weights $\tilde{v}_1, \ldots, \tilde{v}_n$ on the connections to its output neuron which performs fuzzy arithmetic;
- in the hybrid part, the neuron Φ_{p_0} approximates the membership value of the logarithmic trading price to the evaluated fuzzy price, and the neuron Φ_{p_0} compares the evaluated asset risk with the risk preference of the agent to produce an advice in favour or to disqualify the asset.

Here $I_{p_1},\ldots,I_{dy_T}$ and $I_1,\ldots,I_n$ are crisp identity functions, $\Omega_1,\ldots,\Omega_m$ are sigmoid relations, $\tilde{I}_{p_0}$ is a fuzzy identity function, and m,n are the numbers of nodes in the two hidden layers of the regular segment, respectively. The output of the regular segment $\tilde{p}_{0_{FNN}}$ approximates the fuzzy logarithmic price, and the transfer function Φ_{p_0} produces the asset risk $\Re_{FNN}$ relevant to the evaluated price and the trading logarithmic price q,

$$\Re_{FNN} = \Phi\left(\tilde{p}_{0_{FNN}}, q\right) = \mu\left(q \mid \tilde{p}_{0_{FNN}}\right) = sup\left\{\alpha \mid x_{p_{0_{FNN}}} = q\right\} .$$

Then, the hard-limit transfer function $\Psi\left(\Re_{FNN}, \Re_{agent}\right)$ compares the evaluated asset risk with the agent's risk preference, where the tolerable risk $\Re_{agent}$ acts as a threshold. It is recognised that the same opportunity will have different bearings on the risk position of different market players, and the advice is agent-dependent. In comparison, the general practice is to produce asset-related recommendations only.

Our task is to train the regular network segment, which requires modifying the fuzzy pricing formula (4.4) into a fuzzy-valued function. This involves a four-step procedure:

- For each asset, the parameters of linearisation $\delta_1 = \delta_1\left(dy_1,\ldots,dy_T\right)$ and $\delta_2 = \delta_2\left(dy_1,\ldots,dy_T\right)$ are evaluated from relations (4.3) and considered crisp.

- Factor imprecision is introduced following initially the calibration technique from Chapter 4, which produces triangular fuzzy numbers for the level data – corresponding to fatter-tail possibility distributions – and nonlinear membership functions for the logarithmic data. Next, positive fuzzy coefficients $\tilde{a}_t$, $\tilde{b}_t$, $\tilde{c}_t$ are obtained from

$$\tilde{a}_t = \frac{1}{s}\sum_{j=1}^{s}\left(\tilde{p}_{tj} \Big/ p_{tj}\right), \; \tilde{b}_t = \frac{1}{s}\sum_{j=1}^{s}\left(\tilde{r}_{tj} \Big/ r_{tj}\right), \; \tilde{c}_t = \frac{1}{s}\sum_{j=1}^{s}\left(\tilde{dy}_{tj} \Big/ dy_{tj}\right), \; 1 \le t \le T,$$

where s is the number of assets in the sample, T is the time horizon of the pricing factors, and p_{tj}, r_{tj}, dy_{tj} are the logarithmic factors. Then, the initial data calibration is modified to

$$\tilde{p}_{tj} = \tilde{a}_t p_{tj} \;,\; \tilde{r}_{tj} = \tilde{b}_t r_{tj} \;,\; \tilde{dy}_{tj} = \tilde{c}_t dy_{tj} \;,\; 1 \le t \le T \;,\; 1 \le j \le s \;.$$

Thus the final calibration further introduces characteristics of the sample imprecision. Therefore the classifier can be trained with a representative sample for a market sector, introducing relevant sector imprecision.

- The fuzzy evaluation of the logarithmic asset price is presented as

$$\tilde{p}_0 = \sum_{t=1}^{T} \delta_1^{t-1}\left[(1-\delta_1)(\tilde{c}_t dy_t + \tilde{a}_t p_t) + \delta_2 - \tilde{b}_t r_t\right] + \delta_1^T \tilde{a}_T p_T = \left[\tilde{a}_1 p_1 (1-\delta_1)\right] + \cdots + \left[\tilde{a}_T p_T \delta_1^{T-1}\right] -$$

$$-\left[\tilde{b}_1 r_1\right] - \cdots - \left[\tilde{b}_T r_T \delta_1^{T-1}\right] + \left[\tilde{c}_1 dy_1 (1-\delta_1)\right] + \cdots + \left[\tilde{c}_T dy_T \delta_1^{T-1}(1-\delta_1)\right] + \left[\sum_{t=1}^{T} \delta_1^{t-1}\delta_2\right] =$$

$$= \tilde{a}_1 g_{a_1} + \cdots + \tilde{a}_T g_{a_T} - \tilde{b}_1 g_{b_1} - \cdots - \tilde{b}_T g_{b_T} + \tilde{c}_1 g_{c_1} + \cdots + \tilde{c}_T g_{c_T} + g, \; 0 < \delta_1, \delta_2 < 1, \qquad (6.1)$$

where $g_{a_t} = g_{a_t}(y)$, $g_{b_t} = g_{b_t}(y)$, $g_{c_t} = g_{c_t}(y)$, $g = g(y)$ are continuous functions defined on the crisp factors $y = (p_1, \dots, p_T, r_1, \dots, r_T, dy_1, \dots, dy_T)$, $p_t > 0$, $0 < r_t < ln(2)$, $dy_t < 0$, $1 \le t \le T$, employed to evaluate an asset. Thus the asset price is described as a continuous multivariable fuzzy-valued function of type (5.11).

- Finally, applying the extension principle, the nonlinear membership function of the asset price in formula (6.1) is defined by

$$\mu\left(x_{p_0} \mid \tilde{p}_0\right) = sup\left\{\alpha \middle| x_{p_0} \in \tilde{p}_0(\alpha)\right\} , \tag{6.2}$$

$$\tilde{p}_0(\alpha) = \left\{a_1 g_{a_1}(y) + \cdots + c_T g_{c_T}(y) + g(y) \middle| a_1 \in \tilde{a}_1(\alpha), \dots, c_T \in \tilde{c}_T(\alpha), y = (p_1, \dots, dy_T)\right\}$$

  where the $\alpha-cut$ $\tilde{p}_0(\alpha)$ commutes with interval arithmetic computation.

The above procedure yields the fuzzy-valued price $\tilde{p}_0(y)$ in (6.1) as dependent on the crisp factors y – this is a CMFV function of type (5.1). The four-layer feedforward RFNN segment in figure 6.1 corresponds to the subclass $\wp_{min}\{RFNN \mid \tilde{f}_{RFNN}(y)\}$ of networks in definition (5.12) that are capable of approximating such type of functions. The inputs to the segment are the crisp factors y and the segment-transfer function $\tilde{f}_{RFNN}(y) = \tilde{p}_{0_{FNN}}(y)$ is fuzzy-valued. Considering its fuzzy parameters $\tilde{v}_1, \dots, \tilde{v}_n$, the segment-transfer function meets the conditions of proposition 2.1 and commutes with level-cutting, where the $\alpha-cut$ $\tilde{p}_{0_{FNN}}(\alpha)$ is computed through interval arithmetic.

6.3 Chromosome Encoding and Search Operators Definition

We will develop an algorithm that evolves $\wp_{min}\left\{RFNN \mid \tilde{p}_{0_{FNN}}(y)\right\}$. The first task is to represent the network weights and bias terms into the chromosome and to generate the initial population. The crisp weights w_{jt}, u_{jt}, z_{jt}, ℓ_{ij} and bias terms θ_j, for $1 \le t \le T, 1 \le j \le m, 1 \le i \le n$, are presented with real numbers. A nonisosceles triangular fuzzy weight $\tilde{v}_i$, $1 \le i \le n$, is described with the real triple $\left(v_i^{begin}, v_i^{vertex}, v_i^{end}\right)$, corresponding to its support and vertex, where $v_i^{begin} < v_i^{vertex} < v_i^{end}$. Thus the fuzzy network is encoded into a chromosome χ of $3mT + m + mn + 3n$ real numbers,

$$\chi = \left\{w_{11}, w_{12}, \ldots, w_{mT}, \ldots, u_{mT}, \ldots, z_{mT}, \ldots, \theta_m, \ldots, \ell_{nm}, \ldots, \left(v_n^{begin}, v_n^{vertex}, v_n^{end}\right)\right\} \quad (6.3)$$

$$v_i^{begin} < v_i^{vertex} < v_i^{end}, \ 1 \le i \le n,$$

where each weight and bias term is recognised as a single gene. The initial population $X = \left[\chi_1 \ \ \chi_2 \ \ \cdots \ \ \chi_\gamma\right]$ of γ individuals χ from definition (6.3) is generated simultaneously as a matrix of size $(3mT + m + mn + 3n) \times \gamma$, whose elements are realisations of a random variable with standard normal distribution. A block representation of X as $\left[X^{(1)}_{(3mT+m+nm)\times\gamma} \ \ X^{(2)}_{3\times n} \ \ \cdots \ \ X^{(1+n)}_{3\times\gamma}\right]'$ helps to concurrently sort its elements according to the inequality restrictions in (6.3).

The next task is to choose the search operators of crossover and mutation. A multipoint crossover operator is considered, where the number λ of crossover points is randomly chosen for every generation. A triple representing a fuzzy weight is recognised as a single gene, which restricts the crossover points $index_i, 1 \le i \le \lambda$, to the following set of positions in the chromosome indicating the beginning of a gene.

$$index_i \in \{ 1,\ 2,\ 3,\ \dots,\ 3mT+m+nm,\ 3mT+m+nm+1 \quad 3mT+m+nm+4,\ \dots,\ 3mT+m+nm+3n-2 \},\ 1 \le i \le \lambda . \tag{6.4}$$

Two randomly chosen chromosomes from the parent population are combined to obtain two offspring, only one of which is included in the new population. For each parent pair, the position of the λ points is randomly chosen from the set described in (6.4).

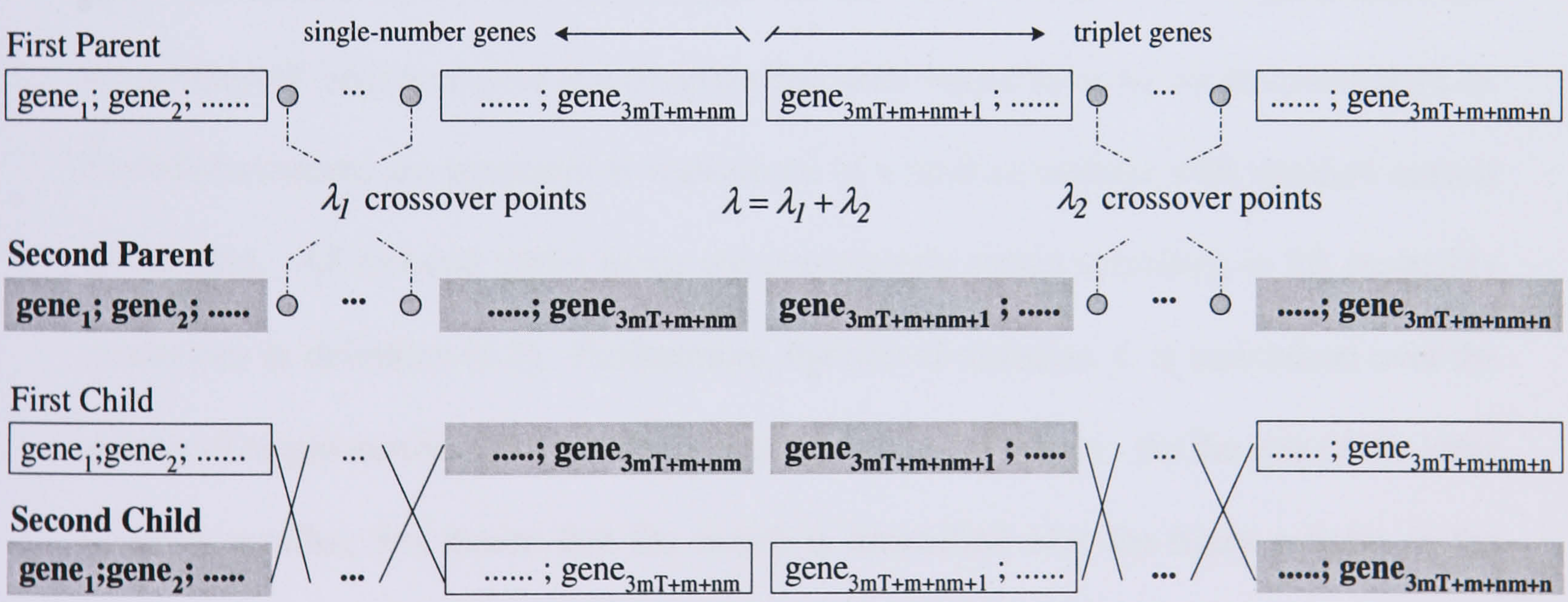


Figure 6.2: Multipoint crossover operator

- the crossover points are only set between genes, a single gene consists either of one real-number network parameter or of a triple representing a fuzzy-number weight;
- parent chromosomes belong to the best subpopulation, only the first-child chromosomes are added to that breeding subpopulation to form the next full-size population.

However, it is assured that the crossover points are proportionally distributed over the two types of subsets representing crisp network parameters and fuzzy weights, respectively. Thus λ_1 points are chosen within the former subset and λ_2 points are used in the latter, as shown in figure 6.2.

$$\lambda_1 = \frac{3mT + m + nm}{3mT + m + nm + n}\lambda \quad , \qquad \lambda_2 = \frac{n}{3mT + m + nm + n}\lambda$$

Next, the mutation operator transforms a temporary full-size population, obtained from a breeding subpopulation through crossover, into the new generation of network representations. The operator involves a constant rate τ and thus the number of mutated genes is constant. However, their positions are randomly chosen for every generation, and the position of a mutated gene in a chromosome takes values from the set described in (6.4). Modified elements are generated as realisations of a random variable with standard normal distribution. All mutated triplet genes are concurrently sorted according to the inequality restrictions in definition (6.3). Furthermore, the rate of mutation τ is maintained over the subsets of single-number genes and triplet genes, correspondingly. As the size of the latter subset is smaller, this means that the search is intensified over the fuzzy weights in the network. The reasoning is that they introduce to the network features of the factor imprecision along with features of the asset evaluation, and thus have an important contribution to the information processing ability of the network.

Finally, a breeding subpopulation X_{SUB} of size γ_1 is selected from the full-size population X at each generation, through minimising a cost function $\xi(\chi_i)$.

$$\min_i(\xi(\chi_i)) \quad , \; 1 \le i \le \gamma \; , \; \gamma_1 < \gamma$$

The cost function evaluates the error of the RFNN encoded in a chromosome $\chi_i, 1 \le i \le \gamma$,

$$\xi(\chi_i) = \max_{\alpha \in \{0,\ldots,1\}} \left(\max \left(\left| \underline{p_{O_{FNN,\chi_i}}(\alpha)} - \underline{p_O(\alpha)} \right| \; , \; \left| \overline{p_{O_{FNN,\chi_i}}(\alpha)} - \overline{p_O(\alpha)} \right| \right) \right) . \tag{6.5}$$

Therefore, single-level evolution goes through initialisation, selection and recombination steps. Initialisation involves chromosome encoding and generating the initial population, selection identifies the best subpopulation, and recombination employs the crossover and mutation operators. The selection and recombination steps are repeated until the best chromosome χ_{best} attains fitness 100%. The fitness function ζ is formulated as

$$\zeta(\xi) = \begin{cases} 0 & , \; \xi > \xi_{max} \\ \dfrac{\xi_{max} - \xi}{\xi_{max} - \xi_{min}} 100 & , \; \xi \le \xi_{max} \end{cases} \tag{6.6}$$

where the parameters ξ_{max} and ξ_{min} define a scope of network error ξ.

6.4 Two-Level Exploratory Algorithm: Design

Single-level evolution may not cope with the network training, and we design a two-level exploratory algorithm to overcome the anticipated stalling effect. The algorithm adapts the process of learning features of asset evaluation through exploiting characteristics of the training sample. Improved performance is achieved through first decomposing the overall task while identifying efficient subtasks and their efficient training sequence, then evolving the subtasks and merging the subsolutions gradually while following the identified sequence. The subtasks involve different number of various assets evaluated at varied periods. Thus divide-and-conquer evolution and incremental evolution are incorporated to optimise the network over the entire training sample. The algorithm is outlined as follows.

Algorithm 6.1: Two-level exploratory algorithm: Design

I: Define the training sample as a set of s different assets evaluated at T_1 different periods. For each $asset \times period$ element of the set obtain the price $\tilde{p}_0$ with formula (6.1). In fact, every element $asset \times period \subset \mathbb{R}$ is a set of real-number asset-pricing factors. Then define the overall task as the set $task$ of $s_1 = sT_1$ elements of type $(asset \times period, \tilde{p}_0)$, therefore

$$task = \left\{ (asset \times period, \tilde{p}_0)_i \;\middle|\; asset \times period \subset \mathbb{R}, \tilde{p}_0 \in \Im(\mathbb{R}), 1 \le i \le s_1 \right\} .$$

II: Choose the probing step of generations N_{gen} and the population sizes γ and γ_1.

Choose the parameters $\xi^{(\kappa)}_{DEC1}$, $\xi^{(\kappa)}_{DEC2}$, $\xi^{(\kappa)}_{DEC3}$, $\xi^{(\kappa)}_{DEC4}$, $\xi^{(\kappa)}_{DEC5}$, ξ_{DECEND}, ξ_{INC}, ξ_{INCEND} of the dynamic objective function Ξ in definitions (6.7). Here $\kappa = 1, 2, \ldots$ denotes a level of decomposition of *task*.

III: Encode the chromosome χ according to rule (6.3) and generate a random initial population $\underset{\gamma}{IP}$ of size γ. Initialise $\kappa = 1$, $task^{(\kappa)} = task$.

IV: Evolve a fuzzy network for N_{gen} generations over the complete training sample $task^{(\kappa)}$, using the cost function $\xi(\chi)$ from formula (6.5).

V: Keep the result of the evolution – the breeding subpopulation $\underset{\gamma_1}{X^{(\kappa)}}$ of size γ_1, $\gamma_1 < \gamma$.

VI: Apply the criterion

$$\Xi_1 = \frac{1}{\gamma_1} \sum_{i=1}^{\gamma_1} \left(\max_{task^{(\kappa)}} \left(\xi(\chi_i) \right) \right) - \xi^{(\kappa)}_{DEC1} < 0 \quad , \chi_i \in \underset{\gamma_1}{X^{(\kappa)}} . \tag{6.7.1}$$

If the average network error of the breeding subpopulation $\underset{\gamma_1}{X^{(\kappa)}}$ over the complete $task^{(\kappa)}$ is above the parameter $\xi^{(\kappa)}_{DEC1}$, then go to step **III**.

VII: Apply the criterion

$$\Xi_2 = \frac{1}{\gamma_1}\sum_{i=1}^{\gamma_1}\xi(\chi_i) - \xi_{DEC2}^{(\kappa)} < 0 \quad , \chi_i \in \underset{\gamma_1}{X}^{(\kappa)} \ . \tag{6.7.2}$$

to the separate elements of $task^{(\kappa)}$. If no element of $task^{(\kappa)}$ exists for which the average network error of the breeding subpopulation $\underset{\gamma_1}{X}^{(\kappa)}$ is smaller than the parameter $\xi_{DEC2}^{(\kappa)}$, then generate a full-size population $\underset{\gamma}{IP}\left(\underset{\gamma_1}{X}^{(\kappa)}\right)$ by recombination of $\underset{\gamma_1}{X}^{(\kappa)}$, and go to step **IV**.

VIII: Group the elements of $task^{(\kappa)}$ satisfying the criterion (6.7.2) into the set $task_1^{(\kappa)}$ of J_κ subsets of $task^{(\kappa)}$, where $task_1^{(\kappa)} = \left\{task_{11}^{(\kappa)},\ldots,task_{1j}^{(\kappa)},\ldots,task_{1J_\kappa}^{(\kappa)}\right\}$ and each $task_{1j}^{(\kappa)}, 1 \le j \le J_\kappa$, is of maximum size subject to the condition

$$\Xi_3 = \frac{1}{\gamma_1}\sum_{i=1}^{\gamma_1}\left(\max_{task_{1j}^{(\kappa)}}\left(\xi(\chi_i)\right)\right) - \xi_{DEC2}^{(\kappa)} < 0 \quad , \chi_i \in \underset{\gamma_1}{X}^{(\kappa)} \ . \tag{6.7.3}$$

IX: Partition the training set into $task^{(\kappa)} = \left\{task_{11}^{(\kappa)},\ldots,task_{1j}^{(\kappa)},\ldots,task_{1J_k}^{(\kappa)},task_2^{(\kappa)}\right\}$, where J_κ is the number of subsets satisfying condition (6.7.3). J_κ is specific for the level of decomposition κ. The subset $task_2^{(\kappa)}$ consists of elements not meeting the objective (6.7.3) and $task_2^{(\kappa)} = task^{(\kappa)} \big| task_1^{(\kappa)}$. In the extreme, $task_2^{(\kappa)} = \{\varnothing\}$ or $task_2^{(\kappa)} = task^{(\kappa)}$.

X: Generate different full-size populations $\underset{\gamma}{IP}_{11}\left(\underset{\gamma_1}{X}^{(\kappa)}\right), \dots, \underset{\gamma}{IP}_{1J_\kappa}\left(\underset{\gamma_1}{X}^{(\kappa)}\right)$ and $\underset{\gamma}{IP}_{2}\left(\underset{\gamma_1}{X}^{(\kappa)}\right)$ by recombination of the same breeding subpopulation $\underset{\gamma_1}{X}^{(\kappa)}$.

XI: Evolve a separate fuzzy network in N_{gen} generations for each training subset of $task^{(\kappa)}$. Keep the evolution results – the breeding subpopulations $\underset{\gamma_1}{X}^{(\kappa)}_{11}, \dots, \underset{\gamma_1}{X}^{(\kappa)}_{1J_\kappa}$ and $\underset{\gamma_1}{X}^{(\kappa)}_{2}$.

XII: For those of the subpopulations $\underset{\gamma_1}{X}^{(\kappa)}_{1j}, 1 \le j \le J_\kappa$, evolved in step **XI** that meet the objective

$$\Xi_4 = \frac{1}{\gamma_1}\sum_{i=1}^{\gamma_1}\left(\max_{task^{(\kappa)}_{1j}}\left(\xi(\chi_i)\right)\right) - \xi^{(\kappa)}_{DEC3} < 0 \quad , \ \chi_i \in \underset{\gamma_1}{X}^{(\kappa)}_{1j} \ , \tag{6.7.4}$$

generate full-size populations $\underset{\gamma}{IP}_{1j}\left(\underset{\gamma_1}{X}^{(\kappa)}_{1j}\right)$, and then continue with step **XIII.**

For those subsets $task^{(\kappa)}_{1j}$ that does not satisfy objective (6.7.4), go to step **XIV**.

XIII: Continue evolving a separate fuzzy network for each training subset $task^{(\kappa)}_{1j}$ that meets objective (6.7.4), until the average network error over the better half of its breeding subpopulation $\underset{\gamma_1/2}{X}^{(\kappa)}_{1j}$ is smaller than the parameter ξ_{DECEND}. Therefore, the new objective is

$$\Xi_5 = \frac{2}{\gamma_1} \sum_{i=1}^{\gamma_1/2} \left(\max_{task_{1j}^{(\kappa)}} (\xi(\chi_i)) \right) - \xi_{DECEND} < 0 \quad , \chi_i \in \underset{\gamma_1/2}{X}{}_{1j}^{(\kappa)} . \tag{6.7.5}$$

Keep the result of the evolution – the better halves of the breeding subpopulations $\underset{\gamma_1/2}{X}{}_{1j\,END}^{(\kappa)}$.

XIV: If a $task_{1j}^{(\kappa)}$ consists of one element and satisfies the following conditions,

$$\Xi_6 = \begin{cases} \frac{1}{\gamma_1} \sum_{i=1}^{\gamma_1} \xi(\chi_i) - \xi_{DEC2}^{(\kappa)} < 0 \\ \frac{1}{\gamma_1} \sum_{i=1}^{\lambda_1} \xi(\chi_i) - \xi_{DEC3}^{(\kappa)} > 0 \end{cases} , \chi_i \in \underset{\gamma_1}{X}{}_{1j}^{(\kappa)} \quad , \tag{6.7.6}$$

then discard the evolved subpopulation $\underset{\gamma_1}{X}{}_{1j}^{(\kappa)}$ and generate a new full-size population $\underset{\gamma}{IP}{}_{1j}\left(\underset{\gamma_1}{X}{}^{(\kappa)}\right)$ by recombination of subpopulation $\underset{\gamma_1}{X}{}^{(\kappa)}$ from step **V**. Else, go to step **XVI**.

XV: For each single-element subsets $task_{1j}^{(\kappa)}$ satisfying (6.7.6), evolve a fuzzy network in N_{gen} generations. Keep the result of the evolution – the breeding subpopulations $\underset{\gamma_1}{X}{}_{1j}^{(\kappa)}$. Go to step **XII**.

XVI: If a subset $task_{1j}^{(\kappa)}$ consists of more than one element and the following conditions are satisfied,

$$\Xi_7 = \begin{cases} \frac{1}{\gamma_1}\sum_{i=1}^{\gamma_1}\left(\max_{task_{1j}^{(\kappa)}}(\xi(\chi_i))\right) - \xi_{DEC4}^{(\kappa)} < 0 \\ \frac{1}{\gamma_1}\sum_{i=1}^{\gamma_1}\left(\max_{task_{1j}^{(\kappa)}}(\xi(\chi_i))\right) - \xi_{DEC3}^{(\kappa)} > 0 \end{cases}, \quad \chi_i \in \underset{\gamma_1}{X}{}_{1j}^{(\kappa)}, \tag{6.7.7}$$

then generate a full-size population $\underset{\gamma}{IP}_{1j}\left(\underset{\gamma_1}{X}{}_{1j}^{(\kappa)}\right)$. Consider the subset $task_{1j}^{(\kappa)}$ as a complete training set $task^{(\kappa+1)} = task_{1j}^{(\kappa)}$ and augment the decomposition level $\kappa = \kappa + 1$. Go to step **IV**.

XVII: If a subset $task_{1j}^{(\kappa)}$ consists of more than one element and the following conditions are satisfied,

$$\Xi_8 = \begin{cases} \frac{1}{\gamma_1}\sum_{i=1}^{\gamma_1}\left(\max_{task_{1j}^{(\kappa)}}(\xi(\chi_i))\right) - \xi_{DEC2}^{(\kappa)} < 0 \\ \frac{1}{\gamma_1}\sum_{i=1}^{\gamma_1}\left(\max_{task_{1j}^{(\kappa)}}(\xi(\chi_i))\right) - \xi_{DEC4}^{(\kappa)} > 0 \end{cases}, \quad \chi_i \in \underset{\gamma_1}{X}{}_{1j}^{(\kappa)}, \tag{6.7.8}$$

then discard the evolved subpopulation $\underset{\gamma_1}{X}{}_{1j}^{(\kappa)}$ and generate a new full-size population $\underset{\gamma}{IP}_{1j}\left(\underset{\gamma_1}{X}{}^{(\kappa)}\right)$ by recombination of subpopulation $\underset{\gamma_1}{X}{}^{(\kappa)}$ from step **V**. Consider the subset $task_{1j}^{(\kappa)}$ as a complete training set $task^{(\kappa+1)} = task_{1j}^{(\kappa)}$ and augment the decomposition level $\kappa = \kappa + 1$. Go to step **IV**.

XVIII: If the objective

$$\Xi_9 = \frac{1}{\gamma_1} \sum_{i=1}^{\gamma_1} \left(\max_{task_2^{(\kappa)}} \left(\xi(\chi_i) \right) \right) - \xi_{DEC5}^{(\kappa)} < 0 \quad , \ \chi_i \in X_{\gamma_1\, 2}^{(\kappa)} , \tag{6.7.9}$$

is satisfied for the subset $task_2^{(\kappa)}$, then generate a full-size population $IP_{\gamma\ 2}\left(X_{\gamma_1\, 2}^{(\kappa)} \right)$ by recombination of the evolved subpopulation $X_{\gamma_1\, 2}^{(\kappa)}$ from step **XI**, and continue with step **XIX**. Else, generate the new population $IP_{\gamma\ 2}\left(X_{\gamma_1}^{(\kappa)} \right)$ by recombination of the breeding subpopulation $X_{\gamma_1}^{(\kappa)}$ from step **V**. Consider $task_2^{(\kappa)}$ as a complete training set $task^{(\kappa+1)}$ at the level of decomposition $\kappa+1$, therefore $task^{(\kappa+1)} = task_2^{(\kappa)}$. Augment $\kappa = \kappa+1$ and go to step **IV**.

XIX: If $task_2^{(\kappa)}$ consists of one element, then evolve a fuzzy network until meeting the objective

$$\Xi_{10} = \frac{2}{\gamma_1} \sum_{i=1}^{\gamma_1/2} \left(\xi(\chi_i) \right) - \xi_{DECEND} < 0 \quad , \ \chi_i \in X_{\gamma_1\, 2}^{(\kappa)} , \tag{6.7.10}$$

keep the result – the better half of the breeding subpopulation $X_{\gamma_1/2\ 2\,END}^{(\kappa)}$ – and continue with step **XX**. Else consider $task_2^{(\kappa)}$ as a complete training set $task^{(\kappa+1)} = task_2^{(\kappa)}$ at the level of decomposition $\kappa+1$. Augment $\kappa = \kappa+1$ and go to step **IV**.

XX: Set $\kappa = \kappa_m$ at the highest level of partition $\kappa_m = max(\kappa)$. This corresponds to the lowest incremental level $\kappa_m - \kappa + 1 = 1$. Consider the training set $task^{(k)} = \left\{ task_{11}^{(\kappa_m)}, \ldots, task_{1J_{k_m}}^{(\kappa_m)}, task_2^{(\kappa_m)} \right\}$. Generate a full-size population $\underset{\gamma}{IP}{}_{INC}\left(\underset{\gamma_1/2}{X}{}_{11END}^{(\kappa_m)}, \ldots, \underset{\gamma_1/2}{X}{}_{1J_{\kappa_m}END}^{(\kappa_m)}, \underset{\gamma_1/2}{X}{}_{2END}^{(\kappa_m)} \right)$ through recombination of the better halves of the breeding subpopulations $\underset{\gamma_1/2}{X}{}_{11END}^{(\kappa_m)}$, ... , $\underset{\gamma_1/2}{X}{}_{1J_{\kappa_m}END}^{(\kappa_m)}$, $\underset{\gamma_1/2}{X}{}_{2END}^{(\kappa_m)}$ evolved when last visiting steps **XIII** and **XIX**.

XXI: Evolve a fuzzy network over the training set $task^{(\kappa)}$ until meeting the objective

$$\Xi_{11} = \frac{3}{\gamma_1} \sum_{i=1}^{\gamma_1/3} \left(\max_{task^{(\kappa)}} \left(\xi(\chi_i) \right) \right) - \xi_{INC} < 0 \quad , \quad \underset{\gamma_1/3}{X}{}_{INC}^{(\kappa)} . \tag{6.7.11}$$

Keep the result $\underset{\gamma_1/2}{X}{}_{INC}^{(\kappa)}$.

XXII: Decrease $\kappa = \kappa - 1$, which is equivalent to increasing the incremental level. Consider the set $task^{(\kappa)} = \left\{ task_{11}^{(\kappa)}, \ldots, task_{1J_\kappa}^{(\kappa)}, task_2^{(\kappa)} = task^{(\kappa+1)} \right\}$.

Generate population $\underset{\gamma}{IP}{}_{INC}\left(\underset{\gamma_1/2}{X}{}_{11END}^{(\kappa)}, \ldots, \underset{\gamma_1/2}{X}{}_{1J_\kappa END}^{(\kappa)}, \underset{\gamma_1/2}{X}{}_{INC}^{(\kappa+1)} \right)$ of full-size, through recombination of the better half of subpopulation $\underset{\gamma_1/2}{X}{}_{INC}^{(\kappa+1)}$ evolved when last visiting step **XXI**, as well as the better halves of subpopulations $X^{(\kappa)}_{\gamma_1/2\ 11END}, \ \dots \ , \ X^{(\kappa)}_{\gamma_1/2\ 1J_\kappa END}$ evolved when visiting step **XIII** for the κ^{th} time. If $\kappa > 1$, go to step **XXI**.

XXIII: Evolve a fuzzy network over the whole training set $task = task^{(1)}$ until the final objective is achieved,

$$\Xi_{12} = \max_{task^{(1)}} \left(\xi \left(\chi^{best}_{INCEND} \right) \right) - \xi_{INCEND} < 0 \quad , \tag{6.7.12}$$

where χ^{best}_{INCEND} is the best chromosome in the population. Keep chromosome χ^{best}_{INCEND} – it represents the completely evolved network.

First a random initial population $\underset{\gamma}{IP}$ of size γ is generated and a fuzzy network is evolved over the full training set $task$ for a probing number of generations N_{gen}. If the objective based on the average value of the network error ξ – on all elements of type $(asset \times period, \tilde{p}_0)$ in the training set and over the resulting subpopulation $X^{(1)}_{\gamma_1}$ of γ_1 best-fitted chromosomes – is above the limit $\xi^{(1)}_{DEC1}$, then evolution starts again with a new random initial population. Otherwise, each element of $task$ is probed against an updated objective Ξ_2 over the best subpopulation $X^{(1)}_{\gamma_1}$. If there does not exist a single element $(asset \times period, \tilde{p}_0)$ of $task$ with an objective value below the updated limit $\xi^{(1)}_{DEC2}$, then a new full-size population is generated by recombination of the best subpopulation and the evolution is continued for another

N_{gen} generations. Otherwise, all the elements satisfying the condition are grouped into the training subset $task_1^{(1)}$ and the rest are included into the subset $task_2^{(1)}$, where $task = task^{(1)} = \left\{ task_1^{(1)}, task_2^{(1)} \right\}$. The elements of $task_1^{(1)}$ are further probed in different combinations against the second limit $\xi_{DEC2}^{(1)}$ over the subpopulation $\underset{\gamma_1}{X}^{(1)}$ in order to identify training subsets $task_{1j}^{(1)}$ of maximum number of elements according to objective Ξ_3. Then $task_1^{(1)}$ is partitioned into the subsets $task_1^{(1)} = \left\{ task_{11}^{(1)}, \dots, task_{1j}^{(1)}, \dots, task_{1J_1}^{(1)} \right\}$, where J_1 is the number of subsets, each maximum-size subset $task_{1j}^{(1)}$ satisfies objective Ξ_3, and every element of $task_{1j}^{(1)}$ meets objective Ξ_2. Thus the decomposition at the first level $\kappa = 1$ is produced. Now $J_1 + 1$ different full-size populations $\underset{\gamma}{IP}_{11}\left(\underset{\gamma_1}{X}^{(1)} \right), \dots, \underset{\gamma}{IP}_{1J_\kappa}\left(\underset{\gamma_1}{X}^{(1)} \right)$ and $\underset{\gamma}{IP}_{2}\left(\underset{\gamma_1}{X}^{(1)} \right)$ are generated by recombination of the same breeding subpopulation and a separate fuzzy network is evolved in N_{gen} generations for each subset of $task^{(1)}$. The results are the breeding subpopulations $\underset{\gamma_1}{X}_{11}^{(1)}, \dots, \underset{\gamma_1}{X}_{1J_1}^{(1)}$ and $\underset{\gamma_1}{X}_{2}^{(1)}$. For those of subpopulations $\underset{\gamma_1}{X}_{1j}^{(1)}$ that meet objective Ξ_4 over their corresponding subset $task_{1j}^{(1)}$, the next full-size population is generated by the recombination $\underset{\gamma}{IP}_{1j}\left(\underset{\gamma_1}{X}_{1j}^{(1)} \right)$, and the network evolution over $task_{1j}^{(1)}$ continues until the average error over the better half of the breeding subpopulation $\underset{\gamma_1/2}{X}{}_{1j}^{(1)}$ is smaller than the parameter ξ_{DECEND}. For those of subpopulations $\underset{\gamma_1}{X}{}_{1j}^{(1)}$ that do not meet objective Ξ_4, we consider two cases – when their corresponding subsets $task_{1j}^{(1)}$ consist of a single element $(asset \times period, \tilde{p}_0)$ or a number of elements. If condition Ξ_6 is satisfied in the single-element case, then the evolved $\underset{\gamma_1}{X}{}_{1j}^{(1)}$ is discarded, the next full-size population $\underset{\gamma}{IP}{}_{1j}\left(\underset{\gamma_1}{X}{}^{(1)}\right)$ is generated on the basis of the subpopulation evolved earlier over the complete training set $task^{(1)}$, and a network is evolved over $task_{1j}^{(1)}$ in another N_{gen} generations in order to approach objective Ξ_4 again. Otherwise, if $task_{1j}^{(1)}$ is a multiple-element subset, it is considered as a complete training set at the next decomposition level, with a starting full-size population obtained as $\underset{\gamma}{IP}{}_{1j}\left(\underset{\gamma_1}{X}{}_{1j}^{(1)}\right)$ if objective Ξ_7 is met or as $\underset{\gamma}{IP}{}_{1j}\left(\underset{\gamma_1}{X}{}^{(1)}\right)$ if condition Ξ_8 is valid. Considering subset $task_2^{(1)}$, if objective Ξ_9 introducing $\xi_{DEC5}^{(1)}$ is not satisfied, then $\underset{\gamma_1}{X}{}_2^{(1)}$ evolved over $task_2^{(1)}$ is discarded, $\underset{\gamma}{IP}{}_2$ is produced from $\underset{\gamma_1}{X}{}^{(1)}$ evolved earlier over $task^{(1)}$, and $task_2^{(1)}$ is assumed a complete set at a further level of decomposition. Otherwise, we generate the next population as the recombination $\underset{\gamma}{IP}_2\left(\underset{\gamma_I}{X}{}_2^{(I)}\right)$, and again involve single-element and multiple-element cases. If $task_2^{(I)}$ is a single-element subset meeting objective Ξ_9, then a fuzzy network is evolved until the error over this element averaged on the better half of the breeding subpopulation falls below the limit $\xi_{DECEND}^{(I)}$ and this half $\underset{\gamma_I/2}{X}{}_{2\,END}^{(\kappa)}$ is kept as one of the temporary results. . If $task_2^{(I)}$ is a multiple -element subset meeting objective Ξ_9, then $task_2^{(I)}$ is assumed a complete set at a further level of decomposition. Then the steps up to here are repeated with revised parameters in the objectives $\Xi_1,\ldots,\Xi_{10}$ until several levels of decomposition are identified, where each level is characterised with a unique partitioning into subsets and a specific number J_κ. The decomposition stage completes when the neural networks evolved over each corresponding subset meet objective Ξ_5 or objective Ξ_{10}. Then, the incremental part of the evolution starts from the highest decomposition level $\kappa_m = max(\kappa)$. Each of the training subsets at this level has an evolved breeding subpopulation associated with it. A full-size population is generated by recombination based on the better halves of these subpopulations $\underset{\gamma}{IP}_{INC}\left(\underset{\gamma_I/2}{X}{}_{11END}^{(\kappa_m)},\ldots,\underset{\gamma_I/2}{X}{}_{1J_{\kappa_m}END}^{(\kappa_m)},\underset{\gamma_I/2}{X}{}_{2\,END}^{(\kappa_m)}\right)$. The evolution continues over the whole training set $task^{(\kappa_m)}$ existing at the highest decomposition step until the objective Ξ_{11} – involving the parameter ξ_{INC} and the subpopulation $\underset{\gamma_I/3}{X}{}_{INC}^{(\kappa_m)}$ – is met. This is the first incremental step. Then the second highest decomposition level is considered. The training set existing at this level includes by definition the whole training set from the highest decomposition step and some further subsets. Again one full-size population is generated by recombination of the better halves of the breeding subpopulations associated with each training subset here. The subset equivalent to the highest decomposition set of projects is presented with the subpopulation evolved at the first incremental level $X_{\gamma_1/2}{}^{(\kappa_m)}_{INC}$. The parameter in objective Ξ_{11} is still ξ_{INC} but Ξ_{11} is evaluated over a larger training set $task^{(\kappa_m - 1)}$, where $task^{(\kappa_m)} \subset task^{(\kappa_m - 1)}$. This is the second incremental level. The incremental procedure continues by analogy until the first decomposition step is reached, where the partitioning was applied over the initial full set. Therefore, the fuzzy network evolved at the final incremental level considers all elements $(asset \times period, \tilde{p}_0)$ in $task$. The evolution concludes when the error of the best chromosome reaches the limit ξ_{INCEND} in the final objective Ξ_{12}. This best chromosome χ^{best}_{INCEND} represents the completely evolved network.

The two-level exploratory algorithm is visualised in figure 6.3. For simplicity, the decomposition part of the evolution in figure 6.3 is slightly generalized, omitting objectives Ξ_6, Ξ_7, Ξ_8 that involves parameters $\xi^{(k)}_{DEC2}, \xi^{(k)}_{DEC3}, \xi^{(k)}_{DEC4}$.

Figure 6.3: Two-level exploratory algorithm

6.5 Conclusion

We suggest a single-level algorithm and a two-level exploratory algorithm. Though both algorithms employ the cost function $\xi(\chi_i)$, they do this in a different manner. By definition (6.5), $\xi(\chi_i)$ evaluates the network error associated with a single chromosome over a single element $(asset \times period, \tilde{p}_0)$ of the training set. Throughout single-level evolution, the error is calculated on a single chromosome over the complete training set $task$, $\max_{task} \xi(\chi_i)$. On the other hand, throughout the two-level exploratory algorithm, $\xi(\chi_i)$ is involved in varying ways in the dynamic objective function Ξ from definitions (6.7). At different steps, Ξ evaluates the error of a single chromosome – as in the final objective Ξ_{12} – or the averaged error on subpopulations of size γ_1 – as in objectives Ξ_1, Ξ_2, Ξ_3, Ξ_4, Ξ_6, Ξ_7, Ξ_8, Ξ_9 – or subpopulations of size $\gamma_1/2$ in Ξ_5, Ξ_{10}, and $\gamma_1/3$ in Ξ_{11}. Furthermore, Ξ calculates the error over training subsets involving various number of different elements of $task$ rather than over the full set $task$. Thus, Ξ_1 and Ξ_{11} involve set $task^{(\kappa)}$ where $task^{(\kappa)} \subseteq task$, Ξ_2 involves a single element of $task^{(\kappa)}$, then $\Xi_3, \Xi_4, \Xi_5, \Xi_7, \Xi_8$ involve subset $task_{1j}^{(\kappa)}$ where $task_{1j}^{(\kappa)} \subset task_1^{(\kappa)} = \left\{task_{11}^{(\kappa)}, \dots task_{1J_\kappa}^{(\kappa)}\right\}$ and $task_1^{(\kappa)} \subset task^{(\kappa)}$, Ξ_6 involves a single-element subset $task_{1j}^{(\kappa)}$, Ξ_9 involves subset $task_2^{(\kappa)}$ where $task_2^{(\kappa)} \subset task^{(\kappa)} = \left\{task_1^{(\kappa)}, task_2^{(\kappa)}\right\}$, Ξ_{10} involves a single-element subset $task_2^{(\kappa)}$, and finally Ξ_{12} involve the full set $task$. Furthermore, Ξ includes different parameters – some out of $\xi_{DEC1}^{(\kappa)}$, $\xi_{DEC2}^{(\kappa)}$, $\xi_{DEC3}^{(\kappa)}$, $\xi_{DEC4}^{(\kappa)}$, $\xi_{DEC5}^{(\kappa)}$, ξ_{DECEND}, ξ_{INC} ξ_{INCEND} – at different steps in the algorithm. Moreover, the objective Ξ depends on the partitioning or incrementing level it is evaluated at, as $\xi_{DEC1}^{(\kappa-1)} \neq \xi_{DEC1}^{(\kappa)}$, $\xi_{DEC2}^{(\kappa-1)} \neq \xi_{DEC2}^{(\kappa)}$, $\xi_{DEC3}^{(\kappa-1)} \neq \xi_{DEC3}^{(\kappa)}$, $\xi_{DEC4}^{(\kappa-1)} \neq \xi_{DEC4}^{(\kappa)}$, $\xi_{DEC5}^{(\kappa-1)} \neq \xi_{DEC5}^{(\kappa)}$, and $task^{(\kappa-1)} \subset task^{(\kappa)}$, $task_{1j}^{(\kappa-1)} \neq task_{1j}^{(\kappa)}$, $task_2^{(\kappa-1)} \neq task_2^{(\kappa)}$.

On the other hand, both single-level and two-level evolution use the same fitness function $\zeta\left(\xi\left(\chi_{best}\right)\right)$ from definition (6.6) concerning the ongoing best chromosome χ_{best}, whether on the full set or on a partial subset. This allows the comparison of the two algorithms, which is the subject of the next Chapter 7, where the empirical training, validating and predicting are also included.

Chapter 7: Risk Classifier: Empirical Results

7.1 Introduction

The two-level exploratory algorithm described in Chapter 6 is employed here to train the risk classifier over a sample of assets evaluated on asset pricing factors that involve time horizons of length T. The initial task is to construct a representative training set. This requires considering preliminary heuristics about the types of assets and the modes of risk.

Then the central task is training the fuzzy neural network. The steps in the two-level exploratory algorithm are presented that take effect throughout training. Though the general description of the algorithm in Chapter 6, every implementation will invoke different steps. By different implementation we mean training the classifier on a entirely different complete training set. On the other hand training over subsets of a complete training set are considered steps in the algorithm and not different implementation. We include the empirical results at each effectuated step, the value of the ongoing objective, the automatically discovered number of decomposition levels, the automatically discovered partitioning at each decomposition level, and the automatically discovered sequence to follow when enlarging the training set throughout the incremental levels.

Next, the results reached through the two-level exploratory algorithm are compared with those obtained from single-level evolution. The comparison involves the number of generations and the value of the approached final fitness, as well as the graphics of the fitness function throughout generations.

Finally, the consequent task is to validate the classifier on elements not included in the training set, and then use it in prediction.

7.2 Training Set: Preliminary Heuristics

A single element $(asset \times period, \tilde{p}_0)$ of the training set *task* consists of crisp pricing factors $asset \times period$, involved in evaluating an asset at some period, as well as the corresponding fuzzy price $\tilde{p}_0$. The number of assets is s, the number of periods in which they are evaluated is T_l, and the horizon of the involved factors is T. Then $s_l = sT_l$ is the number of elements $(asset \times period, \tilde{p}_0)$ in the complete set *task*. Evolving a fuzzy network over a single element $(asset \times period, \tilde{p}_0)$ constitutes a problem of lowest complexity. Problems with a higher level of complexity concern evolving a network over a multiple-element subset of *task*. The elements in a subset may concern the same asset evaluated in different periods, or different assets evaluated at the same period, or a mixture of those. The problem of highest complexity is evolving the network over the complete training set of size $s_l = sT_l$. The concept of the tow-level exploratory algorithm is to identify problems with decreasing complexity and their efficient sequence, then to solve the lowest complexity problems and to merge them incrementally while following the efficient sequence, in order to optimise the solution of the overall problem. Notably, each single element of *task* has an intrinsic complexity. Thus it is possible for a subset, consisting of elements with a low intrinsic complexity, to be evolved to the final decomposition objective involving parameter ξ_{DECEND}, without the need for further partitioning. Therefore, such subset will be included in $task^{(\kappa_m)}$ at the highest decomposition level, along with some single elements. Furthermore, the complexity of evolving a network over a multiple-element subset depends on the size of the subset, as well as whether different evaluation periods or different assets are involved.

The intrinsic complexity of an element depends on the mode of risk and the type of asset the element represents. There exist three modes of risk: $\Re = 0$, $0 < \Re < 1$ and $\Re = 1$. Further, three types of assets are considered. Having the broad range of modelled imprecision at each evaluation period as well as the varying imprecision involved at different periods, it is unrealistic to consider assets with constant $\Re = 0$. Thus the first type relates to assets with decreasing risk values in consecutive periods. The second kind concerns assets with unstable risk values without particular direction or assets with an increasing risk measure. The third kind includes assets with constant $\Re = 1$ throughout. In order to provide for good predicting capabilities of the classifier, it should be trained over a set that includes elements representing the three modes of risk as well as the three types of assets.

The empirical data involve three UK companies - GOODWIN, DIXONS GROUP and MARKS & SPENCER - from June 1998 to December 1999. Some of the related factor imprecision is visualised in figure 7.1. Furthermore, a six-month factor horizon is selected producing three projects per company and nine in total. Table 7.1 displays the evaluated risk level for each project.

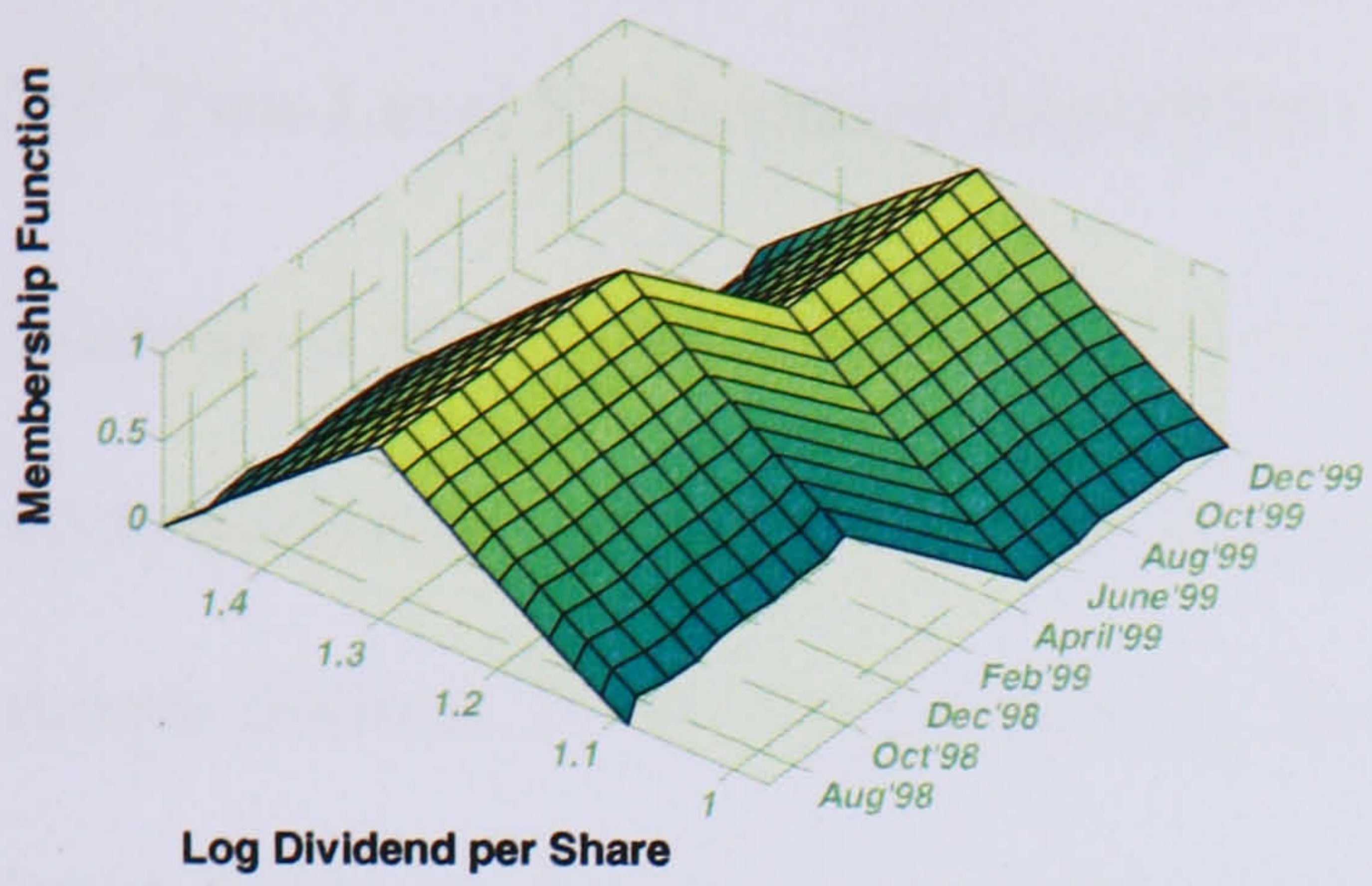


Figure 7.1a: GOODWIN - factor imprecision $\tilde{a}_t p_t + \tilde{c}_t dy_t$

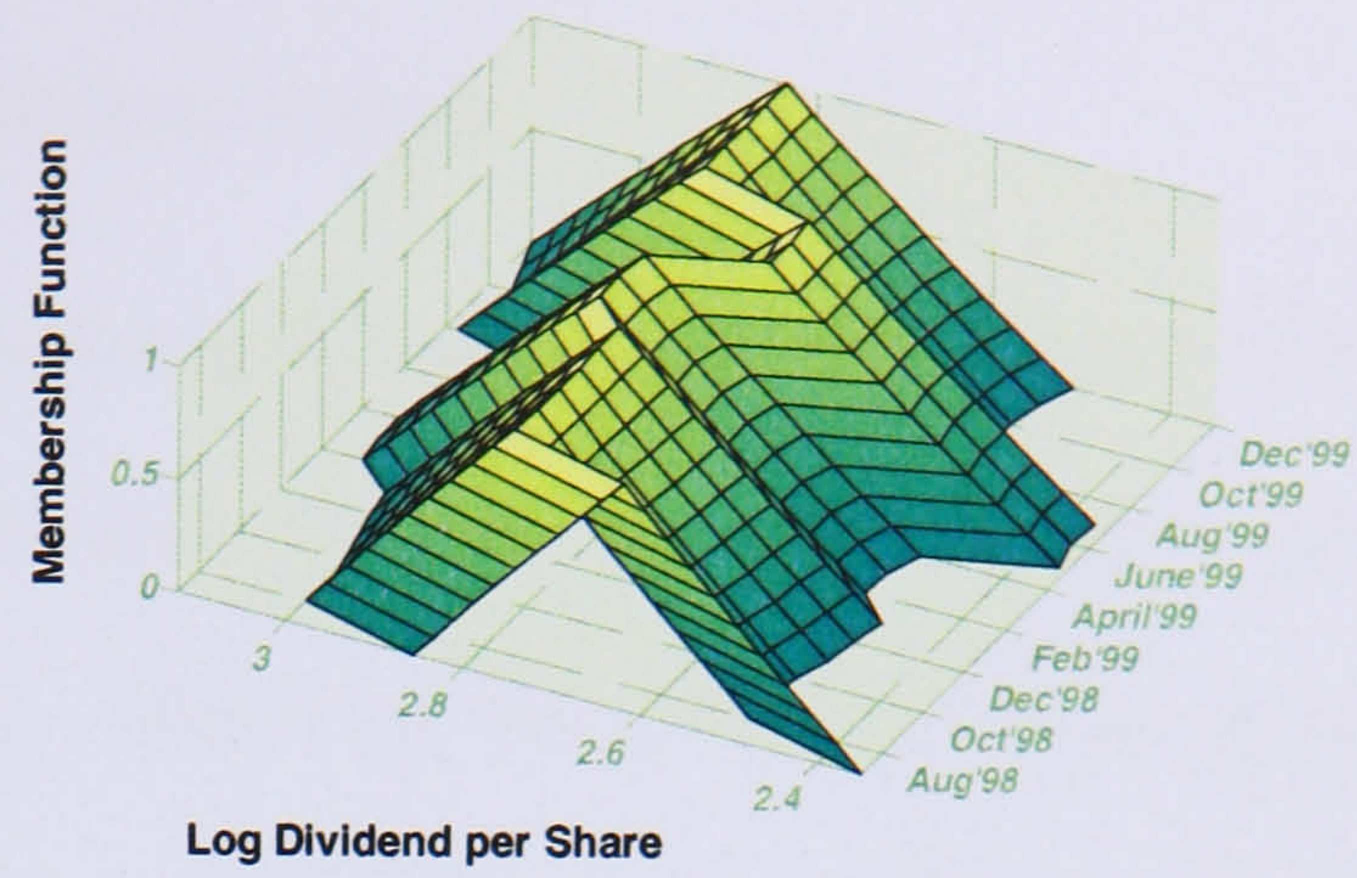


Figure 7.1b: DIXONS GROUP - factor imprecision $\tilde{a}_t p_t + \tilde{c}_t dy_t$

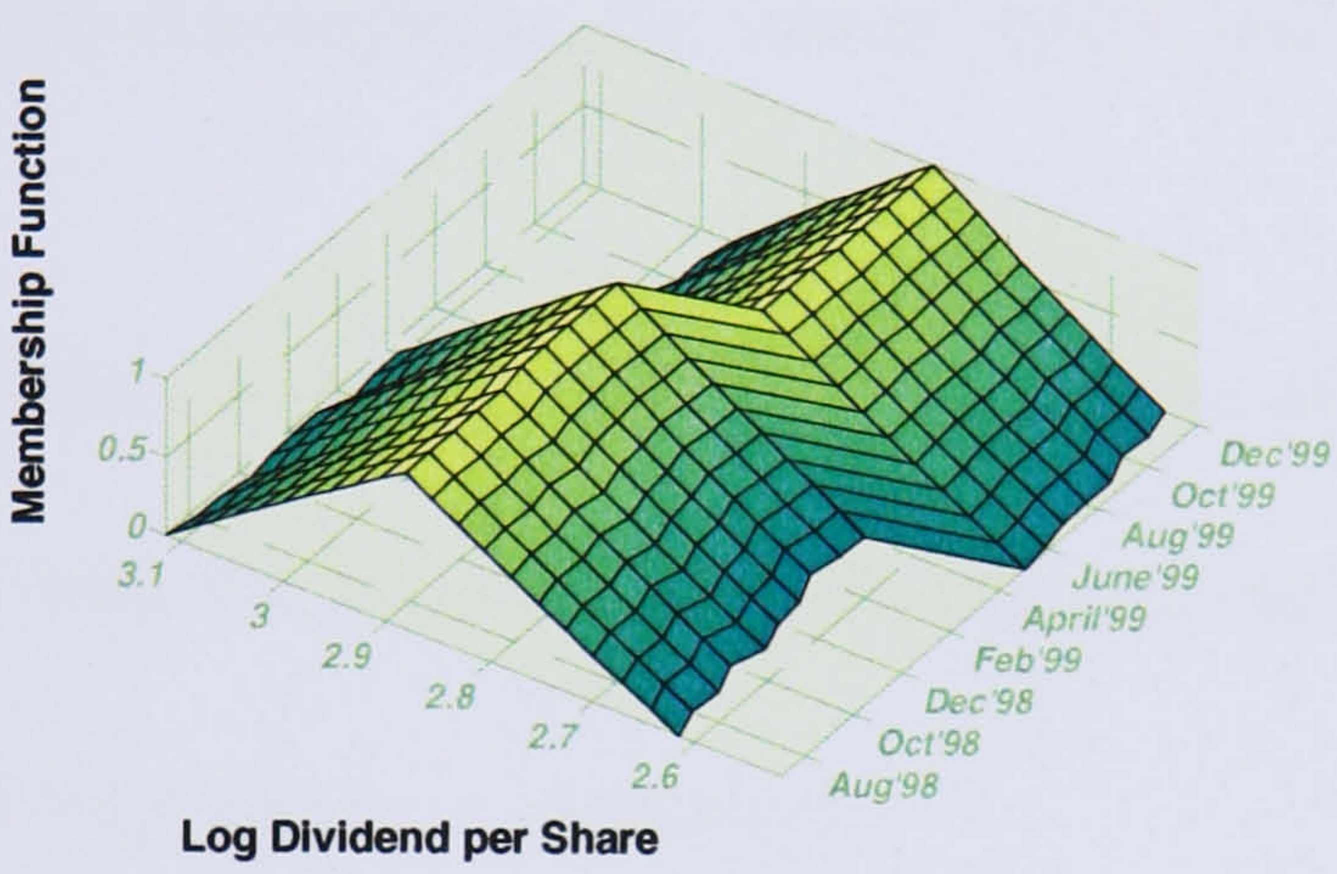


Figure 7.1c: MARKS & SPENCER - factor imprecision $\tilde{a}_t p_t + \tilde{c}_t dy_t$

All risk modes are represented. Furthermore, in the second column, GOODWIN demonstrates a continuously improving risk measure and thus an asset of the first type. In the third column DIXONS GROUP indicates oscillating risk levels – the second asset type – and in the fourth column MARKS & SPENCER maintains $\Re = 1$ throughout – the third type. Figure 7 illustrates how the risk measure for a project is derived based on the estimated fuzzy-valued log share price $\tilde{P}_0$ and the market price p_0.

Table 7.1: Types of assets

period / company	*GOODWIN* *project: risk*	*DIXONS GROUP* *project: risk*	*MARKS & SPENCER* *project: risk*
July-December, 1998	$project_1$: $\Re = 1$	$project_3$: $\Re = 1$	$project_5$: $\Re = 1$
January-June, 1999	$project_2 : \Re = 0.683$	$project_4$: $\Re = 0$	$project_6$: $\Re = 1$
July-December, 1999	$project_7$: $\Re = 0$	$project_9 : \Re = 0.858$	$project_8$: $\Re = 1$

7.3 Two-Level Exploratory Algorithm: Implementation

Training is attempted using single-level evolution and two-level exploratory algorithm, correspondingly, over the first six projects in table 7.1. Each project comprises a six-month horizon $T=6$ for the pricing factors. Therefore, the number of nodes in the input layer of the fuzzy network is $3T=18$. Experimenting preliminary with the number of nodes in the two hidden layers and considering a trade-off between simpler configuration and evolution convergence, the values $m=5$ and $n=3$ are selected, respectively.

Single-level evolution is applied over the six projects simultaneously and the performance is measured with the fitness ζ of the ongoing best chromosome χ_{best} over the complete set $task=\{project_1,\ldots,project_6\}$,

$$\zeta=\begin{cases} 0 & , \ \xi>\xi_{max} \\ \dfrac{\xi_{max}-\max\limits_{\{project_1,\ldots,project_6\}}\left(\xi_{project_j}\left(\chi_{best}\right)\right)}{\xi_{max}-\xi_{min}}100 & , \ \xi\leq\xi_{max} \end{cases} \tag{7.1}$$

$$\xi_{project_j}\left(\chi_{best}\right)=\max_{\alpha\in\{0,\ldots,1\}}\left(\max\left(\left|\underline{po_{FNN,\chi_{best},project_j}(\alpha)}-\underline{po_{0,project_j}(\alpha)}\right|,\right.\right.$$
$$\left.\left.\left|\overline{po_{FNN,\chi_{best},project_j}(\alpha)}-\overline{po_{0,project_j}(\alpha)}\right|\right)\right)$$

where $\xi_{max}=0.1515$ and $\xi_{min}=0.0015$. The training makes some initial progress and then stalls. To get more representative results, five simulations of the single-level algorithm are averaged. The averaged fitness function reaches *46.33%* in *500,000* generations.

The implementation of the training algorithm went through the following steps, where the relevant parameters are selected as shown in Table 7.2.

Table 7.2: Implementation: Algorithm parameters

description	*notation*	*value*
population size	γ	*100*
breeding subpopulation size at decomposition levels	γ_l	*30*
probing number of generations	N_{gen}	*10 000*
parameter in the fitness function ζ	ξ_{min}	*0.0015*
parameter in the fitness function ζ	ξ_{max}	*0.1515*
parameter in objective Ξ_1 at the first decomposition level	$\xi_{DEC1}^{(1)}$	*0.1525*
parameter in objectives Ξ_2 and Ξ_3 at the first decomposition level	$\xi_{DEC2}^{(1)}$	*0.0525*
parameter in objective Ξ_4 at the first decomposition level	$\xi_{DEC3}^{(1)}$	*0.0125*
parameter in objective Ξ_9 at the first decomposition level	$\xi_{DEC5}^{(1)}$	*0.0775*
parameter in objective Ξ_5 during the decomposition stage and related to least complexity problems	ξ_{DECEND}	*0.0025*
parameter in objectives Ξ_2 at the first decomposition level	$\xi_{DEC2}^{(2)}$	*0.0225*
parameter in objectives Ξ_9 at the first decomposition level	$\xi_{DEC5}^{(2)}$	*0.064*
parameter in objectives Ξ_2 and Ξ_3 at the third decomposition level	$\xi_{DEC2}^{(3)}$	*0.0425*
parameter in objective Ξ_{11} during the incremental steps	ξ_{INC}	*0.002*
parameter in objective Ξ_{12} at the final incremental steps	ξ_{INCEND}	*0,0015*

Algorithm 7.1: Two-level exploratory algorithm: Implementation

Decomposition

i: Set the decomposition level $\kappa = 1$. Start with the training set $task^{(1)} = \{project_1, project_2, project_3, project_4, project_5, project_6\}$. Generate a random initial population $\underset{\gamma}{IP}$ of size $\gamma = 100$ and evolve a fuzzy neural network for $N_{gen} = 10\,000$ generations. Probe whether objective Ξ_1 is satisfied,

$$\Xi_1 = \frac{1}{\gamma_1} \sum_{i=1}^{\gamma_1} \left(\max_{\{project_1, \dots, project_6\}} \left(\xi_{project_j}(\chi_i) \right) \right) - \xi^{(1)}_{DEC1} < 0 \ ,$$

$$\chi_i \in \underset{\gamma_1}{X^{(1)}} \ ,$$

where $\xi^{(1)}_{DEC1} = 0.1525$ and $\gamma_1 = 30$. The result in table 7.3 shows that the objective is met. Therefore, we keep the evolved breeding subpopulation $\underset{30}{X^{(1)}}$ of size 30.

Table 7.3: Implementation: Objective Ξ_1 – first attempt

	$\{project_1, project_2, project_3, project_4, project_5, project_6\}$
Ξ_1	-0.0249

ii: Probe whether a single project satisfies the second objective Ξ_2,

$$\Xi_2 = \frac{1}{30} \sum_{i=1}^{30} \xi_{project_j}(\chi_i) - \xi^{(1)}_{DEC2} < 0 \quad , \ \chi_i \in \underset{30}{X^{(1)}} \ ,$$

where $\xi^{(1)}_{DEC2} = 0.0525$.

Table 7.4: Implementation: Objective Ξ_2 – first attempt

	$project_1$	$project_2$	$project_3$	$project_4$	$project_5$	$project_6$
Ξ_2	0.0730	0.0751	0.0589	0.0681	0.0447	0.0178

Therefore, no project satisfies the objective and no partition is yet possible.

iii: Generate a full-size population $\underset{100}{IP}\left(\underset{30}{X}^{(1)}\right)$ by recombination of the breeding subpopulation $\underset{30}{X}^{(1)}$ evolved in step **i**. Continue training over $task^{(1)}$ for another $N_{gen} = 10\,000$ generations. Probe again whether a single project satisfies objective Ξ_2 as formulated in step **ii**.

Table 7.5: Implementation: Objective Ξ_2 – second attempt

	$project_1$	$project_2$	$project_3$	$project_4$	$project_5$	$project_6$
Ξ_2	−0.0077	0.0214	0.0166	0.0108	−0.0053	0.0032

Therefore, $project_1$ and $project_5$ meet the objective. Group them in $task_1^{(1)}$ and group the rest of the projects in $task_2^{(1)}$. Thus the first partitioning is identified.

$$task^{(1)} = \left\{task_1^{(1)}, task_2^{(1)}\right\}$$

$$task_1^{(1)} = \{project_1, project_5\}\ ,\ task_2^{(1)} = \{project_2, project_3, project_4, project_6\}$$

Keep the breeding subpopulation $\underset{30}{X}^{(1)}$ evolved in this step.

iv: Probe whether objective Ξ_3 is satisfied for some combination of elements of $task_1^{(1)}$.

$$\Xi_3 = \frac{1}{30}\sum_{i=1}^{30}\left(\max_{\{project_1, project_5\}}\left(\xi_{project_j}(\chi_i)\right)\right) - 0.0525 < 0 \quad , \chi_i \in \underset{30}{X}^{(1)}$$

Here $\chi_i \in \underset{30}{X}^{(1)}$ is the subpopulation evolved in step **iii.** The only possible combination is $\{project_1, project_5\}$ and the result is $\Xi_3 = 0.0395$.

Table 7.6: Implementation: Objectives Ξ_3 – first attempt

	$\{project_1, project_5\}$
Ξ_3	0.0395

Therefore, the objective is not satisfied and no further grouping of projects is possible. Thus the number $J_1 = 2$ is identified and the final partitioning at the first level of decomposition is described as

$$task^{(1)} = \left\{task_{11}^{(1)}, task_{12}^{(1)}, task_2^{(1)}\right\} \tag{7.2}$$

$$task_{11}^{(1)} = \{project_1\} \quad , \quad task_{12}^{(1)} = \{project_5\}$$

$$task_1^{(1)} = \{project_1, project_5\} \quad , \quad task_2^{(1)} = \{project_2, project_3, project_4, project_6\}$$

v: Generate different full-size populations $\underset{100}{IP}_{11}\left(\underset{30}{X}^{(1)}\right)$, $\underset{100}{IP}_{12}\left(\underset{30}{X}^{(1)}\right)$, $\underset{100}{IP}_{2}\left(\underset{30}{X}^{(1)}\right)$ by recombination of the same breeding subpopulation $\underset{30}{X}^{(1)}$

from step **iii**. Evolve a separate fuzzy network in $N_{gen} = 10\,000$ generations for each subset $task_{11}^{(1)}$, $task_{12}^{(1)}$, $task_{2}^{(1)}$ from description (7.2). Keep the evolved subpopulations $\underset{30}{X}{}_{11}^{(1)}$, $\underset{30}{X}{}_{12}^{(1)}$ and $\underset{30}{X}{}_{2}^{(1)}$.

vi: Probe whether $task_{11}^{(1)}$ or $task_{12}^{(1)}$ meet objective Ξ_4,

$$task_{11}^{(1)}: \quad \Xi_4 = \frac{1}{30}\sum_{i=1}^{30} \xi_{project_1}(\chi_i) - \xi_{DEC3}^{(1)} < 0 \ , \ \chi_i \in \underset{30}{X}{}_{11}^{(1)} \ ,$$

$$task_{12}^{(1)}: \quad \Xi_4 = \frac{1}{30}\sum_{i=1}^{30} \xi_{project_5}(\chi_i) - \xi_{DEC3}^{(1)} < 0 \ , \ \chi_i \in \underset{30}{X}{}_{12}^{(1)} \ ,$$

where $\xi_{DEC3}^{(1)} = 0.0125$ and $\underset{30}{X}{}_{11}^{(1)}, \underset{30}{X}{}_{12}^{(1)}$ are the subpopulations evolved in step **v**.

Also check if $task_{2}^{(1)}$ meets objective Ξ_9,

$$\Xi_9 = \frac{1}{30}\sum_{i=1}^{30}\left(\max_{\{project_2, project_3, project_4, project_6\}} \left(\xi_{project_j}(\chi_i)\right) \right) - \xi_{DEC5}^{(1)} < 0 \ ,$$

$$\chi_i \in \underset{30}{X}{}_{2}^{(1)} \ ,$$

where $\xi_{DEC5}^{(1)} = 0.0775$ and $\underset{30}{X}{}_{2}^{(1)}$ is taken from step **v**.

Table 7.7: Implementation: Objectives Ξ_4 and Ξ_9 – first attempt

	$\{project_1\}$	$\{project_5\}$	$\{project_2, project_3, project_4, project_6\}$
Ξ	$\Xi_4 = -0.0061$	$\Xi_4 = -0.0089$	$\Xi_9 = -0.0131$

Table 7.7 shows that the objectives are met by the corresponding subsets. Therefore, none of the breeding subpopulations $\underset{30}{X}{}^{(1)}_{11}$, $\underset{30}{X}{}^{(1)}_{12}$, $\underset{30}{X}{}^{(1)}_{2}$ is discarded.

vii: Generate full-size populations $\underset{100}{IP}{}_{11}\left(\underset{30}{X}{}^{(1)}_{11}\right)$ and $\underset{100}{IP}{}_{12}\left(\underset{30}{X}{}^{(1)}_{12}\right)$ by recombination of the subpopulations approved in step **vi**. Evolve a separate fuzzy network for $task^{(1)}_{11}$ and $task^{(1)}_{12}$, correspondingly, until the average network error over the better half of the breeding subpopulation falls below $\xi_{DECEND} = 0.0025$.

$$\Xi_5 = \frac{1}{15}\sum_{i=1}^{15} \xi_{project_1}(\chi_i) - 0.0025 < 0 \quad , \chi_i \in \underset{15}{X}{}^{(1)}_{11}$$

$$\Xi_5 = \frac{1}{15}\sum_{i=1}^{15} \xi_{project_5}(\chi_i) - 0.0025 < 0 \quad , \chi_i \in \underset{15}{X}{}^{(1)}_{12}$$

Table 7.8 includes the number of generations run to reach the objective.

Table 7.8: Implementation: Objective Ξ_5 – first attempt

	$\{project_1\}$	$\{project_5\}$
Ξ_5	-0.000018	-0.0000004
N_{gen,Ξ_5}	9014	13465

Keep the best $\gamma_1/2 = 15$ chromosomes, $\underset{15}{X}{}^{(1)}_{11END}$ and $\underset{15}{X}{}^{(1)}_{12END}$, in the evolved populations, correspondingly. This is the last decomposition step at which $project_1$ and $project_5$ participate.

viii: Set the level of decomposition $\kappa = 2$. Consider the subset $task_2^{(1)}$ from level one as the full set $task^{(2)}$ at level two,

$$task^{(2)} = task_2^{(1)} = \{project_2, project_3, project_4, project_6\} \ .$$

Probe whether a single element of this set satisfies the modified objective Ξ_2 where $\xi_{DEC2}^{(2)} = 0.0225$ and the subpopulation $\chi_i \in \underset{30}{X}{}_2^{(1)}$ is the one approved in step **vi**.

$$\Xi_2 = \frac{1}{30}\sum_{i=1}^{30} \xi_{project_j}(\chi_i) - \xi_{DEC2}^{(2)} < 0 \quad , \chi_i \in \underset{30}{X}^{(1)} \quad , project_j \in task^{(2)}$$

Table 7.9: Implementation: Objective Ξ_2 – third attempt

	$project_2$	$project_3$	$project_4$	$project_6$
Ξ_2	*0.0419*	*0.0406*	*0.0383*	*–0.0044*

Table 7.9 indicates that $project_6$ satisfies the objective. Therefore, the partitioning at the second decomposition level is identified as in description (7.3).

$$task^{(2)} = \left\{task_{11}^{(2)}, task_2^{(2)}\right\} \tag{7.3}$$

$$task_1^{(2)} = task_{11}^{(2)} = \{project_6\} \quad , \quad task_2^{(2)} = \{project_2, project_3, project_4\}$$

As only one project meets Ξ_2, this sets $J_2 = 1$ and no further grouping is possible.

ix: Generate a full-size population $\underset{100}{IP}\ {}_{11}\left(\underset{30}{X}{}_{2}^{(1)}\right)$ and evolve a separate network for $task_{11}^{(2)}$ until the average error over the best fifteen chromosomes falls below $\xi_{DECEND} = 0.0025$.

$$\Xi_5 = \frac{1}{15}\sum_{i=1}^{15} \xi_{project_6}(\chi_i) - 0.0025 < 0 \;,\; \chi_i \in \underset{15}{X}{}_{11}^{(2)}$$

Table 7.10: Implementation: Objective Ξ_5 – second attempt

	$\{project_6\}$
Ξ_5	-0.000032
N_{gen,Ξ_5}	5 147

Notice that $N_{gen,\Xi_5} = 5\,147 < N_{gen} = 10\,000$ which avoids overfitting. Keep the evolved subpopulation of size 15, $\underset{15}{X}{}_{11END}^{(2)}$. This is the last decomposition step at which $project_6$ participates.

x: Generate a full-size population $\underset{100}{IP}\ {}_{2}\left(\underset{30}{X}{}_{2}^{(1)}\right)$ by recombination of the subpopulation approved in step **vi**, and evolve a fuzzy network for $N_{gen} = 10\,000$ generations over the training set $task_2^{(2)}$. Check whether objective Ξ_9 is satisfied,

$$\Xi_9 = \frac{1}{30}\sum_{i=1}^{30}\left(\max_{\{project_2, project_3, project_4\}} \left(\xi_{project_j}\left(\chi_i\right)\right) \right) - \xi^{(2)}_{DEC5} < 0 \;, \chi_i \in X^{(2)}_{30\;2} ,$$

where $\xi^{(2)}_{DEC5} = 0.064$. The result in table 7.11 suggests we should keep the evolved subpopulation $X^{(2)}_{30\;2}$.

Table 7.11: Implementation: Objective Ξ_9 – second attempt

	$\{project_2, project_3, project_4\}$
Ξ_9	-0.0002

xi: Set the level of decomposition at $\kappa = 3$. Consider the subset $task^{(2)}_2$ from the second level as the full set $task^{(3)}$ at the third level,

$$task^{(3)} = task^{(2)}_2 = \{project_2, project_3, project_4\} .$$

Generate a full-size population $\underset{100}{IP}\left(X^{(2)}_{30\;2}\right)$ by recombination of the subpopulation evolved in step **x**. Evolve further the network over $task^{(3)}$ for a multiple of $N_{gen} = 10\,000$ generations until the modified objective Ξ_1 is satisfied,

$$\Xi_1 = \frac{1}{30}\sum_{i=1}^{30}\left(\max_{\{project_2, project_3, project_4\}} \left(\xi_{project_j}\left(\chi_i\right)\right) \right) - \xi^{(3)}_{DEC1} < 0 \;, \chi_i \in X^{(3)}_{30} .$$

Table 7.12 shows that this is achieved after $4N_{gen}$. Keep the evolved subpopulation $\underset{30}{X}^{(3)}$.

Table 7.12: Implementation: Objective Ξ_1 – second attempt

	$\{project_2, project_3, project_4\}$
Ξ_1	-0.0071
N_{gen,Ξ_1}	$40\,000$

xii: Probe whether some of the single elements of $task^{(3)}$ satisfies the modified objective Ξ_2 where $\xi^{(3)}_{DEC2} = 0.0425$ and $\underset{30}{X}^{(3)}$ is the subpopulation evolved in step **xi**.

$$\Xi_2 = \frac{1}{30}\sum_{i=1}^{30}\xi_{project_j}(\chi_i) - \xi^{(3)}_{DEC2} < 0 \quad , \chi_i \in \underset{30}{X}^{(1)} \quad , project_j \in task^{(3)}$$

The result in table 7.13 indicates that $project_3$ and $project_4$ meet the objective.

Table 7.13: Implementation: Objective Ξ_2 – fourth attempt

	$project_2$	$project_3$	$project_4$
Ξ_2	-0.0071	-0.0071	-0.0071

Therefore, the following initial partitioning of $task^{(3)}$ is identified.

$$task^{(3)} = \left\{ task_1^{(3)}, task_2^{(3)} \right\}$$

$$task_1^{(3)} = \left\{ project_3, project_4 \right\} \quad , \quad task_2^{(2)} = \left\{ project_6 \right\}$$

xiii: Check whether some combination of elements of $task_1^{(3)}$ satisfies a modified objective Ξ_3 where $\xi_{DEC2}^{(3)} = 0.0425$ and $\underset{30}{X}^{(3)}$ is the subpopulation evolved in step **xi**. The is only one possible combination, therefore

$$\Xi_3 = \frac{1}{30} \sum_{i=1}^{30} \left(\max_{\{project_3, project_4\}} \left(\xi_{project_j} (\chi_i) \right) \right) - \xi_{DEC2}^{(3)} < 0 \quad , \chi_i \in \underset{30}{X}^{(3)}$$

Table 7.14: Implementation: Objectives Ξ_3 – second attempt

	$\{project_3, project_4\}$
Ξ_3	-0.0004

The result suggests that the two elements of $task_1^{(3)}$ should be grouped into $task_{11}^{(3)}$. Thus the number $J_3 = 1$ is identified as well as the final partitioning at the third level of decomposition,

$$task^{(3)} = \left\{ task_{11}^{(3)}, task_2^{(3)} \right\} \tag{7.4}$$

$$task_{11}^{(3)} = \left\{ project_3, project_4 \right\} ,$$

$$task_1^{(3)} = task_{11}^{(3)} \quad , \quad task_2^{(3)} = \left\{ project_2 \right\}$$

xiv: Generate different full-size populations $\underset{100}{IP}{}_{11}\left(\underset{30}{X}^{(3)}\right)$ and $\underset{100}{IP}{}_{2}\left(\underset{30}{X}^{(3)}\right)$ by recombination of the same subpopulation from step **xi**. Evolve a separate fuzzy network over $task_{11}^{(3)}$ and $task_{2}^{(3)}$, correspondingly, until the average network error of the best 15 chromosomes falls below ξ_{DECEND}.

$$\Xi_5 = \frac{1}{15}\sum_{i=1}^{15}\left(\max_{\{project_3, project_4\}} \xi_{project_j}(\chi_i)\right) - 0.0025 < 0 \quad , \chi_i \in \underset{15}{X}{}_{11}^{(3)}$$

$$\Xi_{10} = \frac{1}{15}\sum_{i=1}^{15}\left(\xi_{project_2}(\chi_i)\right) - \xi_{DECEND} < 0 \quad , \chi_i \in \underset{15}{X}{}_{2}^{(3)}$$

Table 7.15 includes the number of generations run to reach the objectives. Note that these are smaller than $N_{gen} = 10\,000$.

Table 7.15: Implementation: Objectives Ξ_5–third attempt and Ξ_{10}–first attempt

	$\{project_3, project_4\}$	$\{project_2\}$
Ξ	-0.000074	-0.000024
$N_{gen,\Xi}$	$1\,403$	$1\,312$

Keep the evolved subpopulations $\underset{15}{X}{}_{11END}^{(3)}$ and $\underset{15}{X}{}_{2END}^{(3)}$ of size 15. This concludes the decomposition part of the algorithm.

Incrementing

xv: Set $\kappa = max(\kappa) = 3$. Consider the training set $task^{(3)}$ from description (7.4) in step **xiii**. Generate a full-size population $\underset{100}{IP}{}_{INC}\left(\underset{15}{X}{}^{(3)}_{11END}, \underset{15}{X}{}^{(3)}_{2END}\right)$ by recombination of $\underset{15}{X}{}^{(3)}_{11END}$ and $\underset{15}{X}{}^{(3)}_{2END}$ from step **xiv**. Evolve a network over $task^{(3)}$ until meeting the objective Ξ_{11} where $\xi_{INC} = 0.002$,

$$\Xi_{11} = \frac{1}{10}\sum_{i=1}^{10}\left(\max_{\{\{project_3, project_4\},\{project_2\}\}}\left(\xi_{project_j}(\chi_i)\right)\right) - \xi_{INC} < 0, \chi_i \in \underset{10}{X}{}^{(3)}_{INC}.$$

Table 7.16: Implementation: Objective Ξ_{11} –first attempt

	$\{\{project_3, project_4\},\{project_2\}\}$
Ξ_{11}	*-0.000038*
$N_{gen,\Xi_{11}}$	*1918*

Thus the subset $task_1^{(3)} = task_{11}^{(3)} = \{project_3, project_4\}$ is incremented to the subset $task_2^{(3)} = \{project_2\}$, and the first incremental level completes with the network optimised over $task^{(3)} = \{project_2, project_3, project_4\}$. Keep half of the resulting breeding subpopulation $\underset{15}{X}{}^{(3)}_{INC}$.

xvi: Set $\kappa = 2$. Consider the training set $task^{(2)}$ from description (7.3) in step **viii**. Increment subset $task_1^{(2)} = task_{11}^{(2)} = \{project_6\}$ to subset $task_2^{(2)} = task^{(3)} = \{project_2, project_3, project_4\}$ from the first incremental step **xv** into $task^{(2)} = \{project_2, project_3, project_4, project_6\}$. Generate a full-size population $\underset{100}{IP}_{INC}\left(\underset{15}{X}^{(2)}_{11END}, \underset{15}{X}^{(3)}_{INC}\right)$ by recombination of $\underset{15}{X}^{(2)}_{11END}$ from step **ix** and $\underset{15}{X}^{(3)}_{INC}$ from step **xv**. Evolve a network over $task^{(2)}$ until meeting the modified objective Ξ_{11},

$$\Xi_{11} = \frac{1}{5}\sum_{i=1}^{5}\left(\max_{\{\{project_6\},\{project_2, project_3, project_4\}\}}\left(\xi_{project_j}(\chi_i)\right)\right) - 0.002 < 0,$$

$$\chi_i \in \underset{5}{X}^{(2)}_{INC}.$$

Table 7.17: Implementation: Objective Ξ_{11} –second attempt

	$\{\{project_6\},\{project_2, project_3, project_4\}\}$
Ξ_{11}	-0.000016
$N_{gen,\Xi_{11}}$	503

This completes the second incremental level. Keep half of the resulting breeding subpopulation $\underset{15}{X}^{(3)}_{INC}$.

xvii: Set $\kappa = 1$. Consider the training set $task^{(1)}$ from description (7.2) in step **iv**. Increment subsets $task_{11}^{(1)} = \{project_1\}$ and $task_{12}^{(1)} = \{project_5\}$ to subset $task_2^{(1)} = task^{(2)} = \{project_2, project_3, project_4, project_6\}$ from the second incremental step **xvi** into the set $task^{(1)}$ and thus obtain the complete training set $task = task^{(1)} = \{project_1, project_2, project_3, project_4, project_5, project_6\}$.

Generate a full-size population $\underset{100}{IP}\ {}_{INC}\left(\underset{15}{X}{}^{(1)}_{11\,END}, \underset{15}{X}{}^{(1)}_{12\,END}, \underset{15}{X}{}^{(2)}_{INC}\right)$ by recombination of $\underset{15}{X}{}^{(1)}_{11\,END}$ and $\underset{15}{X}{}^{(1)}_{12\,END}$ from step **xvi** and $\underset{15}{X}{}^{(2)}_{INC}$ from step **xvi**. Evolve the network over $task$ until meeting the final objective Ξ_{12},

$$\Xi_{12} = \max_{\{\{project_1\},\{project_5\},\{project_2, project_3, project_4, project_6\}\}} \left(\xi\left(\chi_{INCEND}^{best}\right)\right) - \xi_{INCEND} < 0,$$

where $\xi_{INCEND} = 0.0015$ and $\chi_{INCEND}^{best} \in \underset{30}{X}{}^{(1)}_{INC}$.

Table 7.18: Implementation: Final objective Ξ_{12}

	$\{\{project_1\},\{project_5\},\{project_2, project_3, project_4, project_6\}\}$
Ξ_{12}	-0.0000096
$N_{gen,\Xi_{12}}$	$15\,481$

Keep the best chromosome χ_{INCEND}^{best}. It encodes the completely evolved network. This is the final incremental part in the implementation and completes the evolutionary algorithm.

Steps **i** to **xiv** in the implementation correspond to the decomposition part of the algorithm where the complex problem is divided into subtasks of decreasing complexity. The implementation automatically discovered three levels of decomposition and the corresponding partitioning in descriptions (7.2), (7.3) and (7.4). Then in steps **xv** to **xvii**, the subtasks are merged incrementally in reverse direction to obtain the overall solution. Figure 7.2 presents the result of the automatic partitioning and incrementing.

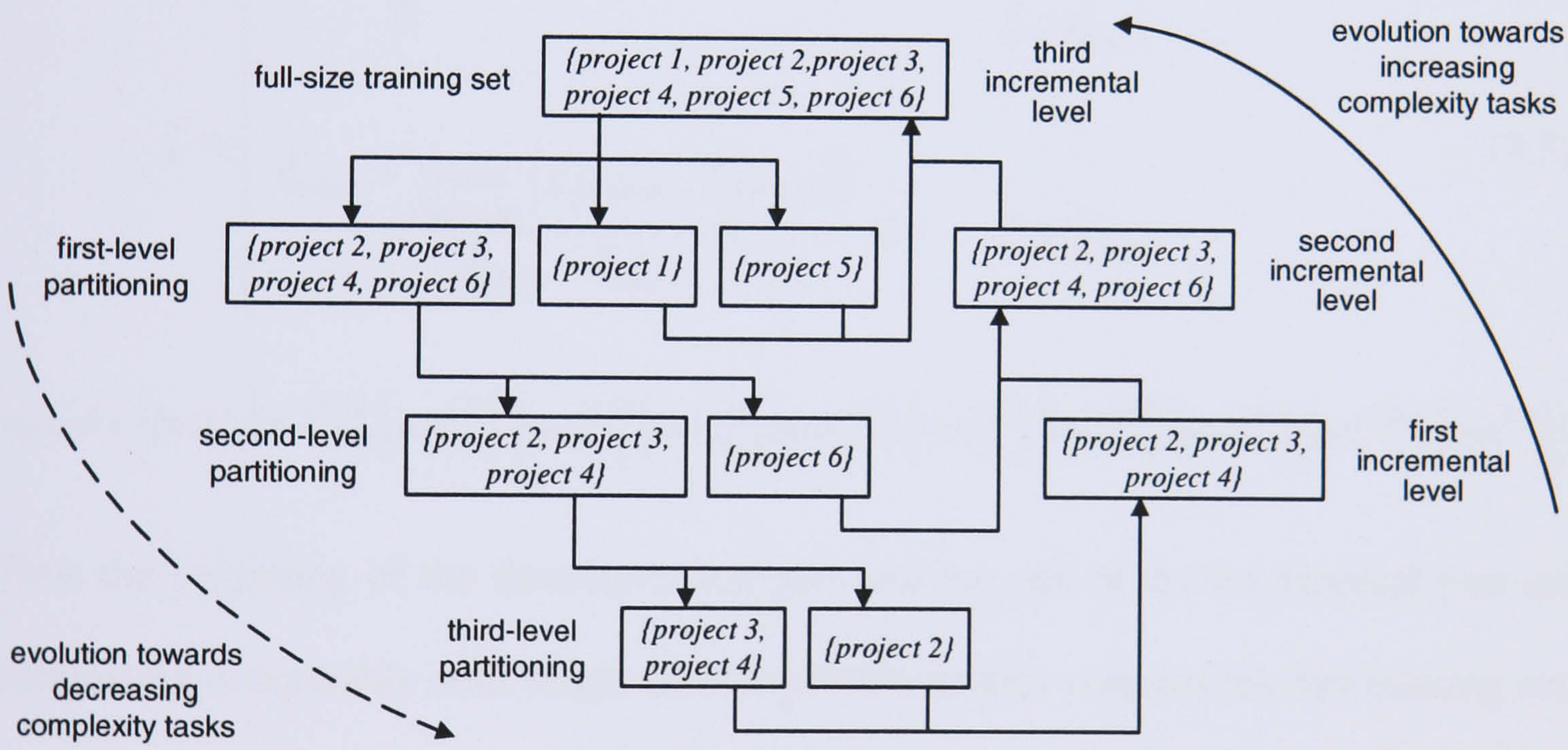


Figure 7.2: Implementation: Training set exploration

- *exploration strategy*: the training set is explored at several levels and the evolution proceeds through tasks with decreasing and then increasing complexity toward solving the integral problem;
- *dynamic objective:* is applied throughout the decomposition and incremental levels.

7.4 Risk Classifier: Training, Validating, Predicting

The performance of single-level evolution is measured with the fitness function ζ from definition (7.1). It represents the fitness of the best chromosome χ_{best} at each generation, over the complete set $task$. To provide for the comparability of the results, the same fitness function ζ is considered in the two-level exploratory algorithm. It again represents the fitness of the best chromosome χ_{best} at each generation and uses the same parameters $\xi_{max} = 0.1515$ and $\xi_{min} = 0.0015$. However, the fitness at each generation is measured over the corresponding training set $subtask$ at that generation.

$$\zeta = \begin{cases} 0 & , \quad \xi > \xi_{max} \\ \dfrac{\xi_{max} - \max\limits_{subtask}\left(\xi_{project_j}(\chi_{best})\right)}{\xi_{max} - \xi_{min}} 100 & , \quad \xi \le \xi_{max} \end{cases} \tag{7.5}$$

$$subtask \in \left\{\{task\}, \left\{task_{11}^{(1)}\right\}, \left\{task_{12}^{(1)}\right\}, \left\{task_{2}^{(1)}\right\}, \left\{task_{11}^{(2)}\right\}, \left\{task_{2}^{(2)}\right\}, \left\{task_{11}^{(3)}\right\}, \left\{task_{2}^{(3)}\right\}, \left\{task^{(3)}\right\}, \left\{task^{(2)}\right\}, \left\{task^{(1)}\right\}\right\}$$

Thus the beginning of the decomposition part and the end of the incremental part are completely comparable with single-level evolution as they concern the full training set. Also, the network error $\xi_{project_j}(\chi_{best})$, both in definition (7.1) and (7.5), is evaluated with

$$\xi_{project_j}(\chi_{best}) = \max_{\alpha \in \{0,\dots,1\}} \left(\max \left(\left| \underline{p0_{FNN,\chi_{best},project_j}(\alpha)} - \underline{p0_{,project_j}(\alpha)} \right| , \left| \overline{p0_{FNN,\chi_{best},project_j}(\alpha)} - \overline{p0_{,project_j}(\alpha)} \right| \right) \right) \tag{7.6.1}$$

where the network approximation $\tilde{p}_{0_{FNN},\chi_{best},project_j}(\alpha)$ is computed from

$$\underline{p_{0_{FNN},\chi_{best},project_j}(\alpha)} = \sum_{i=1}^{n} min\left(\underline{v_{i,\chi_{best}}(\alpha)} \left(\sum_{e=1}^{m} \ell_{ie,\chi_{best}} \Omega_e \right), \overline{v_{i,\chi_{best}}(\alpha)} \left(\sum_{e=1}^{m} \ell_{ie,\chi_{best}} \Omega_e \right) \right),$$

$$\overline{p_{0_{FNN},\chi_{best},project_j}(\alpha)} = \sum_{i=1}^{n} max\left(\underline{v_{i,\chi_{best}}(\alpha)} \left(\sum_{e=1}^{m} \ell_{ie,\chi_{best}} \Omega_e \right), \overline{v_{i,\chi_{best}}(\alpha)} \left(\sum_{e=1}^{m} \ell_{ie,\chi_{best}} \Omega_e \right) \right),$$

$$\Omega_e = \frac{1}{1 + exp\left(-\sum_{t=1}^{T} \left(w_{et,\chi_{best}} p_{t,project_j} + u_{et,\chi_{best}} r_{t,project_j} + z_{et,\chi_{best}} dy_{t,project_j} \right) - \theta_{e,\chi_{best}} \right)} .$$

(7.6.2)

Here $T = 6, m = 5, n = 3$, and $\tilde{v}_{i,\chi_{best}}, \ell_{ie,\chi_{best}}, w_{et,\chi_{best}}, u_{et,\chi_{best}}, z_{et,\chi_{best}}, \theta_{e,\chi_{best}}$ are the network weights and bias terms, as shown figure 6.1, coded into the best chromosome χ_{best}. Next, $p_{t,project_j}, r_{t,project_j}, dy_{t,project_j}$ are the factors related to $project_j$, and $\tilde{p}_{0,project_j}$ is the corresponding asset evaluation with

$$\underline{p_{0,project_j}(\alpha)} = \sum_{t=1}^{T} \delta_1^{t-1} \left[(1-\delta_1)\left(\underline{c_t(\alpha)} dy_{t,project_j} + \underline{a_t(\alpha)} p_{t,project_j} \right) + \delta_2 - \overline{b_t(\alpha)} r_{t,project_j} \right] + \\ + \delta_1^T \underline{a_T(\alpha)} p_{T,project_j} ,$$

$$\overline{p_{0,project_j}(\alpha)} = \sum_{t=1}^{T} \delta_1^{t-1} \left[(1-\delta_1)\left(\overline{c_t(\alpha)} dy_{t,project_j} + \overline{a_t(\alpha)} p_{t,project_j} \right) + \delta_2 - \underline{b_t(\alpha)} r_{t,project_j} \right] + \\ + \delta_1^T \overline{a_T(\alpha)} p_{T,project_j} .$$

(7.6.3)

Figure 7.3 compares the fitness function in the single-level and two-level evolution. Notice that the parameter $\xi_{INCEND} = 0.0015$ of the final objective Ξ_{12} in step **xvii** in the implementation of the two-level exploratory algorithm is chosen equal to the parameter $\xi_{min} = 0.0015$ in the fitness function, and Ξ_{12} involves a single chromosome rather then a subpopulation of chromosomes. Therefore, the final objective in the two-level exploratory algorithm is equivalent to the objective throughout the single-level algorithm. Furthermore, in the first part and in the final part in figure 7.3, both algorithms work with the complete training set. Notice that in the first part, single-level evolution progresses slowly, while two-level evolution exploits the flexibility of discarding unpromising subpopulations and well outperforms, though working on the same full set. The intermediate parts in the figure indicate the track of the two-level algorithm through subtasks of decreasing and then increasing complexity. Though visualised with the fitness function over the corresponding training subset, each intermediate part stops when the relevant objective Ξ is reached rather than when *100%* fitness is reached. Thus each intermediate part completes with a different fitness value, and the next one starts with another fitness value as it involves a different subset. On the other hand, the fitness of single-level evolution progresses smoothly as the objective and the training set are constant.

Exploring the divide-and-conquer approach, the algorithm suggested here automatically partitions the set and discovers the efficient decomposition sequence, according to the intrinsic complexity of the subtasks. The algorithm also performs efficient training over the subtasks. Then, exploring the incremental approach, the algorithm merges the identified subtasks following the efficient sequence in reverse. The subsets involve in each part of the graphics as well as the number of generations at which each part completes are indicated in figure 7.3. Thus the final part in the figure is reached in a superior position in comparison to single level evolution. Our algorithm completes successfully the training and in *148 243* generations reaches *100%* fitness, which is *100%* functionality, over the full set, of the network encoded in the best chromosome. On the other hand, the standard algorithm approaches *46.33%* fitness after *500 000* and the stalling effect prevents further progress.

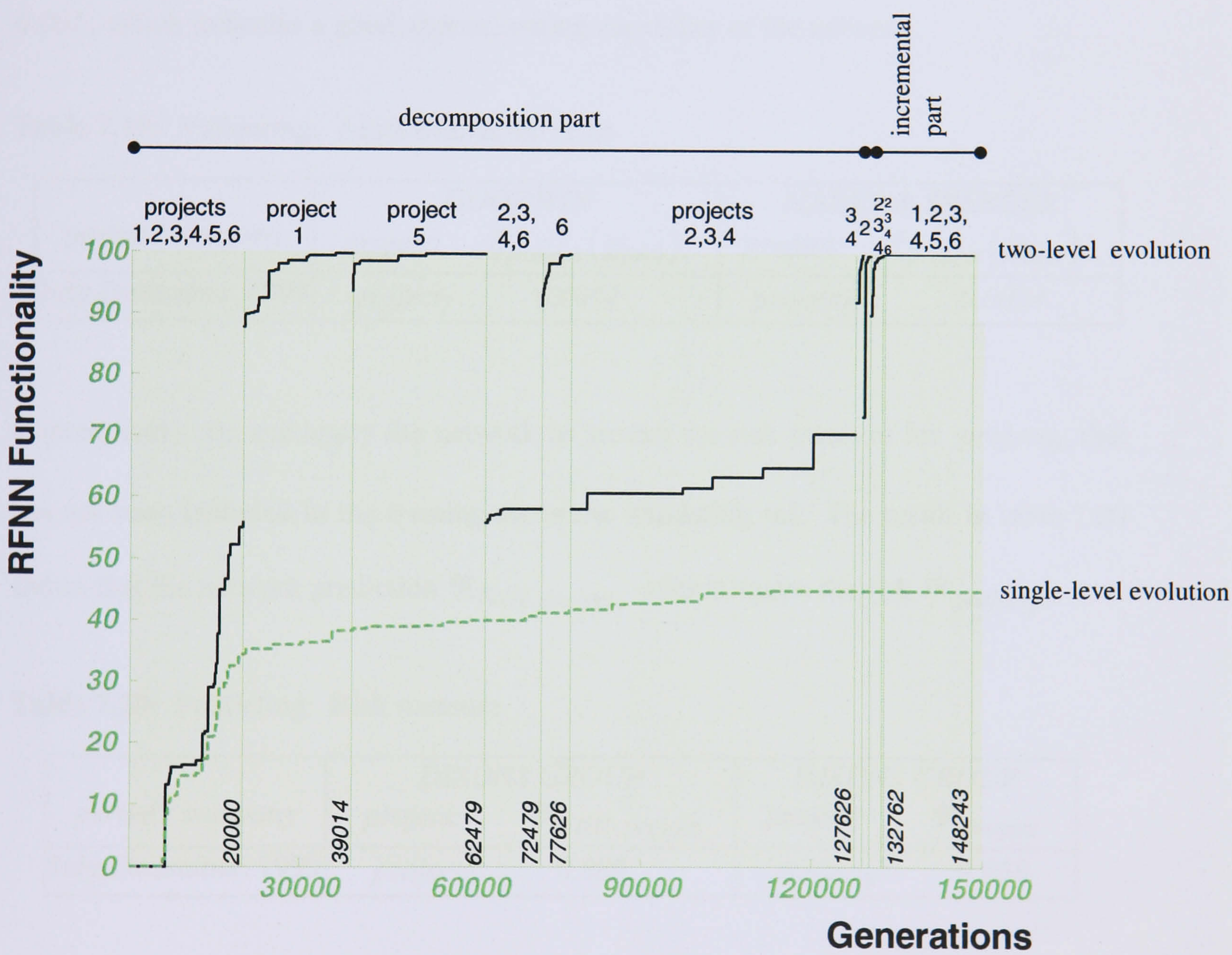


Figure 7.3: Performance of single-level and two-level evolution

- *black line*: two-level evolution solves the overall problem in *148 243* generations;
- *green line*: single-level evolution makes initial progress and then stalls.

Once the fuzzy network is trained, it is validated on $project_7$ and $project_8$ from table 7.1, which has not been included in the training set. The results for the corresponding network errors $\xi_{project_7}(\chi_{best})$ and $\xi_{project_8}(\chi_{best})$ are calculated from formulas 7.6.1, 7.6.2, 7.6.3 and presented in table 7.19. Both errors are below 0.001, which indicates a good approximating capability of the network.

Table 7.19: Validating: Asset evaluation error

period / company	*GOODWIN*		*MARKS & SPENCER*	
	project:	$\xi_{project}(\chi_{best})$	*project:*	$\xi_{project}(\chi_{best})$
July-December, 1999	$project_7$:	*0.0091*	$project_8$:	*0.0024*

Consequently, we can apply the network to predict the risk measure for $project_9$ that has not been included in the training set or the validating set. The result in table 7.20 shows that the network prediction $\Re_{FNN,project_9}$ approximates the risk $\Re_{project_9}$.

Table 7.20: Predicting: Risk measure

period / company	*DIXONS GROUP*		*DIXONS GROUP*	
	project:	$\Re_{FNN,project}$	*project:*	$\Re_{project}$
July-December, 1999	$project_9$:	*0.880*	$project_9$:	*0.858*

Finally, consider the agent-dependent threshold $\Re_{agent}$. If the threshold is set at $\Re_{agent} = 0.9$ then $project_9$ will be accepted. Alternatively, if $\Re_{agent} = 0.8$ then $project_9$ will be rejected as quite risky.

7.5 Conclusion

The empirical results demonstrate decisively that the two-level exploratory algorithm successfully copes with the complex task of training a fuzzy risk classifier. To position the problem, the complexity and the achieved result, we provide a comparison with relevant approaches in Table 7.21.

Table 7.21: Problem complexity: Comparison

author	*J. Buckley et al. [44]*	*A. Serguieva et al. [P3,P4]*	*J. Buckley et al. [46]*	*P. Liu [103]*	*A. Serguieva et al. [P6,P7]*
year	1997	2000	1999	2001	2002
network	crisp network with sign restriction on the weights	crisp network with sign restriction on the weights	crisp network with sign restriction on the weights	fuzzy network	fuzzy network
number of inputs	2	36	-	1	18
technical problem	approximating a single-variable or maximum a two-variable fuzzy function	approximating a muli(36)-variable fuzzy function	approximating a multi-variable fuzzy function (no indication of the number for which a network was trained)	approximating single-variable fuzzy-valued function	approximating multi(18)-variable fuzzy-valued function
training algorithm	backpropagation	modified fast Levenberg-Marrquart backpropagation to accommodated for the sign restrictions	single-level evolutionary training: optimisation of network weights	not addressed	two-level evolutionary training: optimisation of network weights and guidance of the learning process
empirical data	not used	not used	not used	not used	database of 35 assets over 25 years
data heuristics	not used	not used	not used	not used	identified and used as part of the training algorithm
real-world problem	not addressed	fuzzy asset evaluation	not addressed	not addressed	designing and training a fuzzy asset risk classifier

The implementation of the algorithm verifies some of its characteristics and helps identify further ones. In evolving neural networks, an argument against crossover has a reasonable place. However, the flexibility of discarding unpromising and keeping favourable subpopulations at different steps in the decomposition part of the algorithm works toward preserving learned features and still exploring the search space intensively. The performance of the decomposition part proves its ability of partitioning the overall problem and of efficient training over the subtasks to sufficiently low error limits. On the other hand, the incremental part starts with the final favourable subpopulations evolved in the decomposition part, and do not discard subpopulations from this point on. The implementation managed to keep the same error limit ξ_{INC} through all incremental steps while enlarging the training set and reducing the size of the averaged subpopulations, until ξ_{INCEND} is reached over the complete training set and on the best chromosome only. We consider using recombination based on the mutation operator only, as a way to improve further the efficiency of the incremental part.

The algorithm also employs a number of parameters in the dynamic objective function. They have been chosen after some preliminary experiments. The parameter ξ_{INCEND} relates to the desired final accuracy of the best chromosome over the complete training set, and therefore it is reasonable that this is set in advance. However, the rest of the parameters depend on the size and the characteristics of the training set as well as on the random population initialization. Then, it is logical that they are discovered automatically. This is considered as a point for future work. Another focus for further research is embedding our algorithm into a broader algorithm for ensemble training. The ensemble should be trained simultaneously and interactively using negative correlation learning [148]. Notice that we use a small training set, and the validation and prediction is performed on assets representing the same companies involved in the training set, however evaluated in different periods. This is not to depreciate the significance of the results and the advantages of the two-level exploratory algorithm. Training a fuzzy network to approximate a fuzzy-valued function is a considerably complex problem. Also, the same companies perform differently in different periods, and this is one of the characteristics that constitute the problem of asset risk. Still we have somewhat restrained ourselves from involving the whole complexity of the problem. It is expected that the proved generalization abilities of cooperative neural network ensembles [148] will allow the development of the risk classifier in that direction.

The soft classifier described in Chapter 6 and the current Chapter 7 represents an approach to the problem of asset pricing and risk analysis. A logical extension of the study is to consider the developed classifier within a framework of different approaches to the same problem, i.e. a knowledge representation framework for the domain of asset pricing and risk analysis. This is the subject in the next Chapter.

Chapter 8: Further Research and Conclusions

8.1 Intelligent System: Knowledge Representation Framework

The fuzzy asset evaluation technique introduced in Chapter 2, the measures of asset risk and robustness formulated in Chapter 3, the alternative asset ranking procedure developed in Chapter 4, and the soft risk classifier designed in Chapter 5 and Chapter 6 and trained in Chapter 6 and Chapter 7, all these will be involved in the knowledge representation module of an intelligent system in asset pricing and risk analysis. Two modes of operation are considered, as an intelligent tutoring system and as a decision support system. We adopt the view that knowledge is better represented and communicated through a variety of models, each providing explanation or solution from a different perspective. Thus further models will be introduced to expand the domain representation module.

The first task is to identify the types of relations involved in representing problem-solving knowledge. The variety of problem-solving information is appreciated through the types described in Figure 8.1. Structural knowledge captures taxonomic and compositional dependencies. Taxonomic knowledge communicates type-subtype subordination and inheritance of properties between objects. Any subtype object will inherit the general type characteristics and attached problems, however they can be further specified. Compositional relations provide information about the elements of an object. Next, behavioural knowledge is accounted for by a number of relational types, where the strength of dependencies increases from temporal and co-occurrence, through correlational and enablement, to teleological and causal relations.

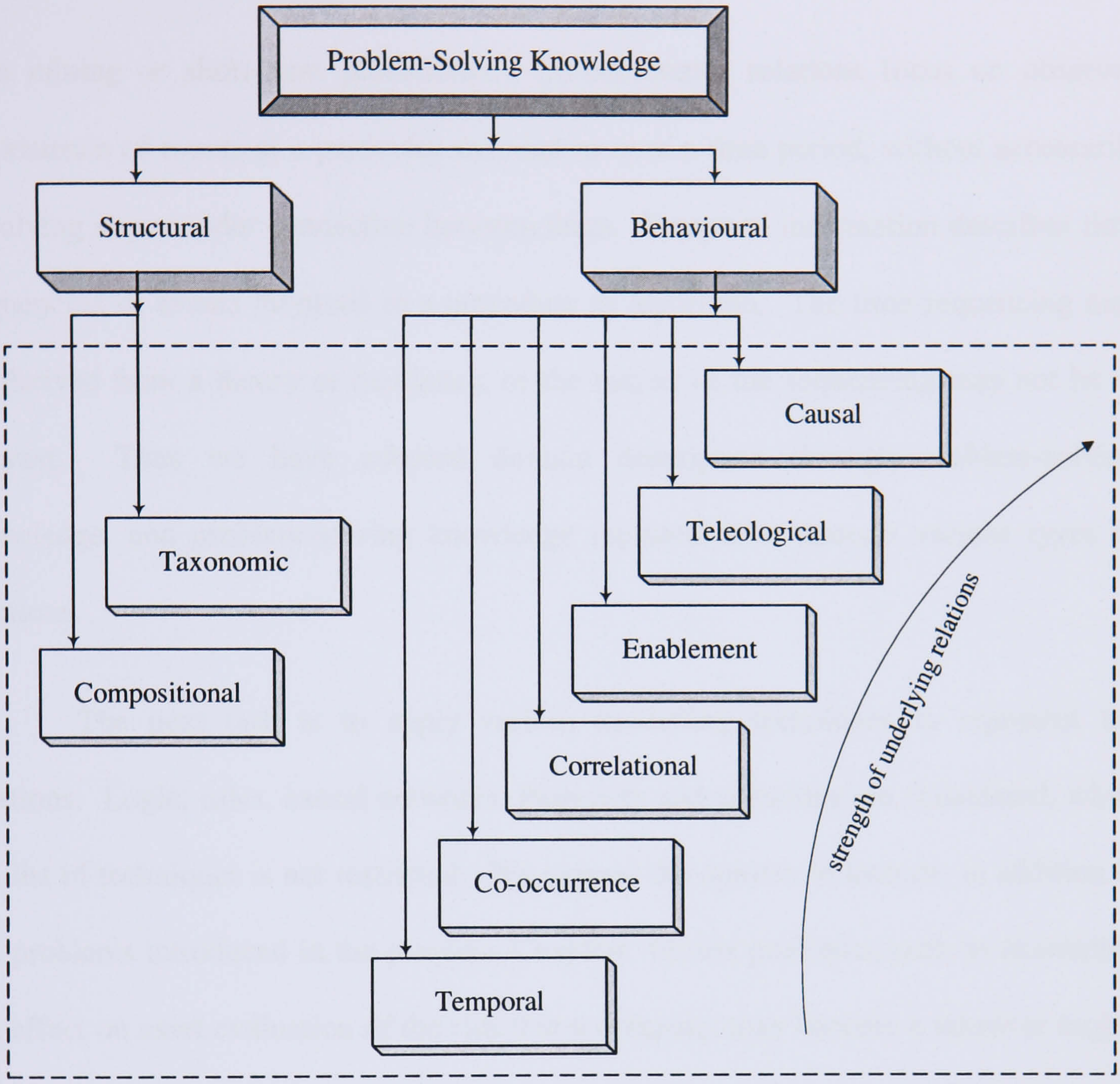


Figure 8.1: Classification of problem-solving knowledge

Causal knowledge aims at cause-effect dependencies among variables. Teleological dependencies focus on a goal and pull the conditions that will bring that goal. Enablement knowledge reveals that one event capacitates another event, and there exists a prerequisite relation between them. Still the relation is not as strong as the preceding two types, and does not imply that any of the events is the driving or the dragging force for the other. Correlation suggests observed dependencies between events but not responsibility of the one for causing the other. Correlational models are mostly used in data mining or short-term predictions. Co-occurrence relations focus on observed coexistence of events at a particular moment or over a time period, without necessarily involving any sounder connection between them. Temporal information describes time sequencing of events involved in a procedure or algorithm. The time sequencing may be derived from a theory or heuristics, or the reason of the sequencing may not be of interest. Thus we have adopted domain description through problem-solving knowledge, and problem-solving knowledge representation through various types of relations.

The next task is to apply various modelling techniques to represent the relations. Logic, rules, causal networks, Petri nets and equations are considered, while the list of techniques is not restricted. We expand the domain to include, in addition to the problems introduced in the previous Chapters, further problems, such as examining the effect on asset evaluation of the risk that a company may become a takeover target, estimating the risk that the company may become a takeover target, examining the effect on asset evaluation of the risk of failure, estimating the risk of failure, as well as investigating the effect of the risk of failure and the takeover risk on the overall asset risk measure. Then, alternative descriptions are considered for each domain problem at more or at less detailed levels of compositional knowledge, e.g. asset evaluation including a greater or a smaller number of pricing factors, or estimating the pricing factors risk-of-failure and takeover-risk using different number of financial ratios. The problems are also arranged in different classes of taxonomy knowledge, e.g. asset classification is in the highest taxonomy class, estimating the overall asset risk is in a lower class, asset evaluation is in yet a lower class, then comes estimating different pricing factors, etc. Finally, the type of behavioural knowledge is considered and the strength of the involved relations. For example, a financial ratio may be included in estimating the risk of takeover either due to a management theory suggesting the causal effect of the ratio on the risk, or because some correlation has been discovered through data mining. Once all alternative descriptions of a problem have been elaborated, then each description is modelled through appropriate modelling techniques. Moreover the same description may be modelled with different techniques. We further consider a soft modification of each model to allow for handling imprecise information. This involves soft extensions of the modelling techniques. Thus, the set of modelling techniques is expanded with multi-valued logic, fuzzy rules, probabilistic causal networks, interval and fuzzy interval timed Petri nets, fuzzy equations and neural networks approximating their solutions. As a result we built a domain of about thirty different models.

The next important task is to suggest a domain structure. It will be based on the knowledge types from figure 8.1. In addition, the modelling phase has emphasised the exploitation of the tolerance to imprecision while representing the same problem. This will effectively introduce a new dimension of relations, and we assume the following relational types in the imprecision dimension, as specified in [32,33]: equal, possibilistic, veristic, probabilistic, probability value, usuality, random set, fuzzy random set and fuzzy graph. Therefore, the models can be arranged within the multiple-model multi-dimensional structure in figure 8.2.

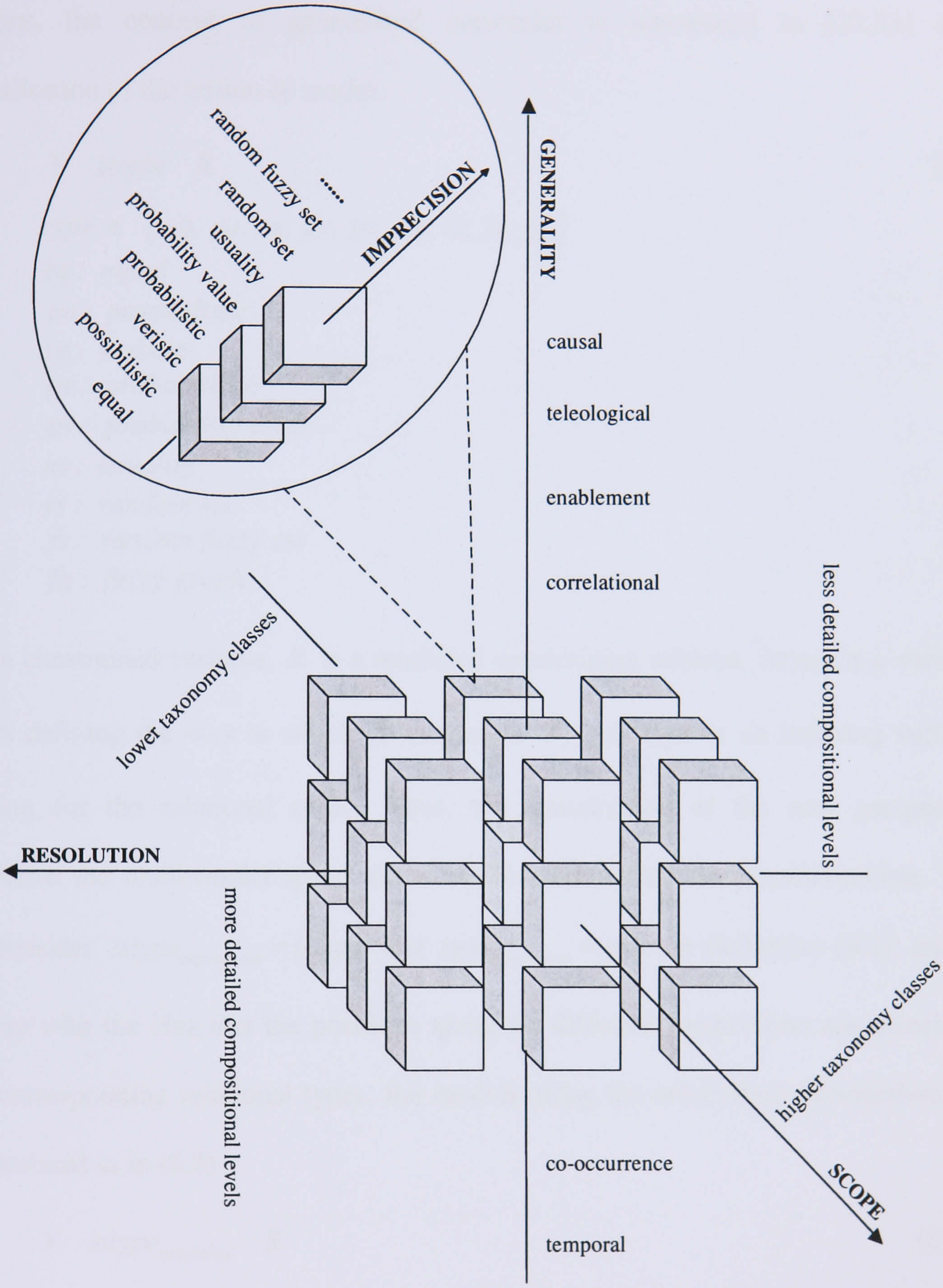


Figure 8.2: Multiple-constraint multi-perspective domain representation

Furthermore, in order to express the wealth of information along the imprecision line of relations, the concept of generalised constraint is introduced in [32,33] as a generalisation of the notion of model.

$$Y \quad istype \quad R \tag{8.1}$$

$$\begin{aligned}
&type \in \{ eq, po, ve, pr, pv, us, rs, fs, fg \} \\
&eq: equal \\
&po: possibilistic \\
&ve: veristic \\
&pr: probabilistic \\
&pv: probability\ value \\
&us: usuality \\
&rs: random\ set \\
&fs: random\ fuzzy\ set \\
&fg: fuzzy\ graph
\end{aligned}$$

Y is a constrained variable, R is a modelled constraining relation, $istype$ is a variable copula defining the way in which R constrains Y, and $type$ is an indexing variable standing for the relational type. Thus, the introduction of the new perspective transforms the multi-model space into a multi-constraint domain representation. We will consider $istype_{imprecision} = istype$ and $type_{imprecision} = type$ in definition (8.1), and to keep up with the idea that the positions along the different perspectives are associated with corresponding relational types, the models along the other three perspectives are reformulated as in (8.2).

$$Y \quad istype_{generality} \quad R \tag{8.2a}$$

$$\begin{aligned}
&type_{generality} \in \{te; co; cl; en; tl; ca\} \\
&te: temporal \\
&co: cooccurrence \\
&cl: correlational \\
&en: enablement \\
&tl: teleological \\
&ca: causal
\end{aligned}$$

$$Y \quad istype_{scope} \quad R \qquad (8.2b)$$
$$type_{scope} \in \{taxonomy\ classes\}$$

$$Y \quad istype_{resolution} \quad R \qquad (8.2c)$$
$$type_{resolution} \in \{compositional\ levels\}$$

The perspective of generality unfolds along the behavioural types of knowledge, where weaker relations - i.e. temporal and co-occurrence - involve specific knowledge as used in procedures, while stronger relations - e.g. teleological and causal - rely on more abstract domain principles. Scope corresponds to taxonomic knowledge, suggesting that a higher taxonomy class is consistent with a broader problem scope. Resolution aligns compositional relations, where a detailed compositional level communicates increased resolution. Then, we can introduce with (8.3) the description of a generalised constraint in the structured domain.

$$Y \quad istype_{resolution} type_{scope} type_{generality} type_{imprecision} \quad R \qquad (8.3)$$

$$type_{resolution} \in \{compositional\ levels\}$$
$$type_{scope} \in \{taxonomy\ classes\}$$
$$type_{generality} \in \{te;\ co;\ cl;\ en;\ tl;\ ca\}$$
$$type_{imprecision} \in \{eq;\ po;\ ve;\ pr;\ pv;\ us;\ rs;\ fs;\ fg\}$$

In figure 8.2, a cuboid indicates the position of constraints with identical identifiers along the three dimensions, and then each cuboid expands in the imprecision dimension. The generalised constraints along this fourth dimension are able to represent perception based information, which is particularly important in the domain of asset pricing and risk analysis, where knowledge no longer starts from the estimation of financial variables in terms of certainty or probability but in the perception of concepts inherent or surrounding the investment process, whose character is not principally measurable.

On the one hand, the multiple-constraint multi-dimensional space allows investigating a domain problem from different perspectives, and thus a greater flexibility in communicating the problem or solving the problem. On the other hand, the developed domain structure facilitates the diagnosis of student knowledge – in the tutoring mode – and the analysis of decision-maker preferences – in the decision support mode of the intelligent system. Cognitive diagnosis or preference analysis is performed through generalised constraint propagation. The formulation of the antecedent part and the consequent part of the cognitive-diagnosis rule or the preference-analysis rule involves domain generalised constraints – following the user choices and thus the trajectory through the domain space – as well as further generalised constraints explicitating perceptions of student-understanding requirements or perceptions of preference categories. Anticipating that cognitive diagnosis and preference analysis will involve such perceptions and knowing of the ability of generalised constraints to explicitate perception based information, those were the reasons to describe the domain through generalised constraints. Thus the diagnostic rule or the preference rule, involving both domain information and preference or diagnostic information, can be formulated uniformly through generalised constraints. The introduction of generalised constraints to domain description, however, is beneficial in its own right, as they are capable of representing all three modes of information granulation – singular, crisp granular (c-granular) or qualitative, and fuzzy granular (f-granular). All these modes are involved in the description of the domain of asset risk analysis.

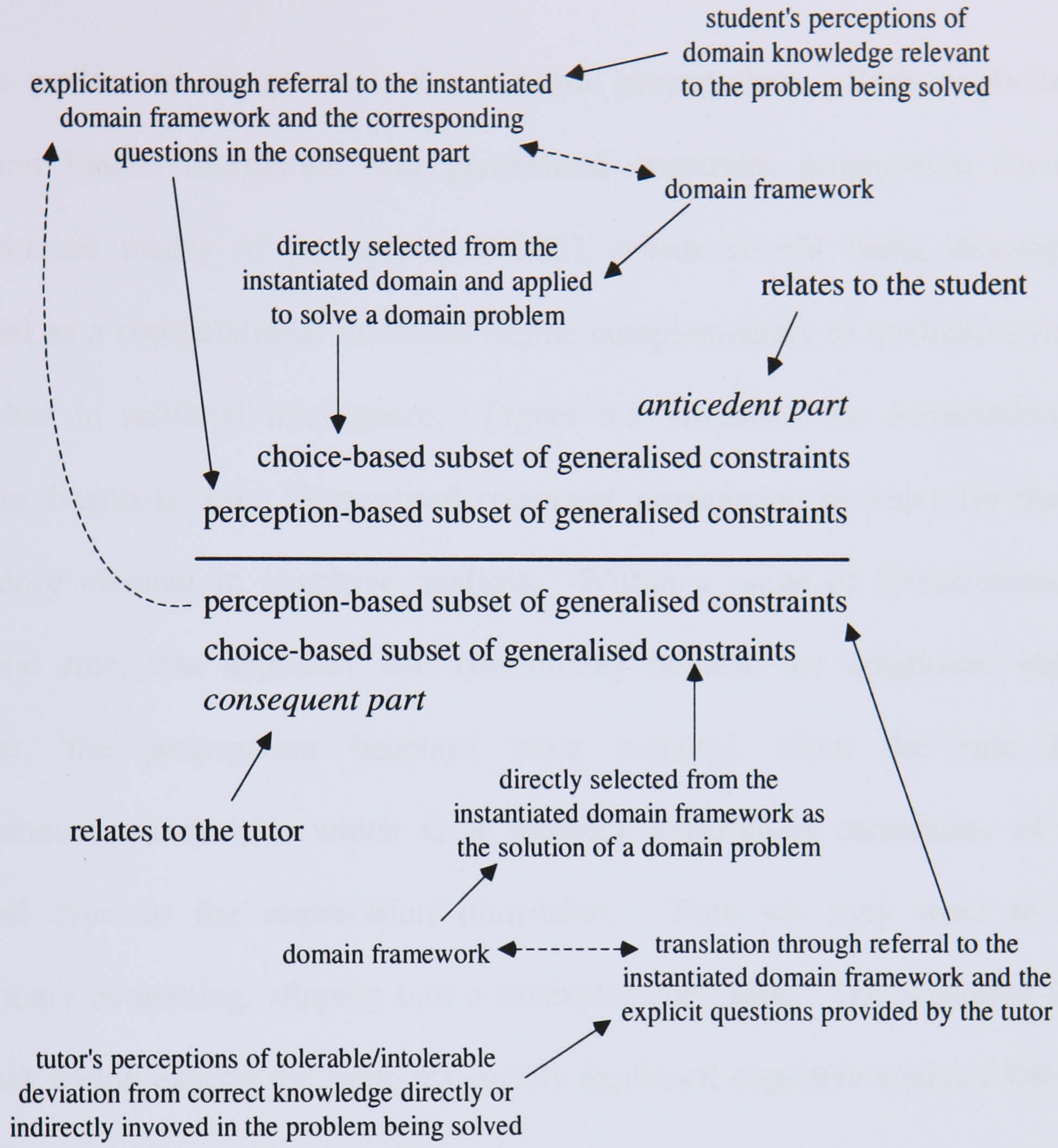


Figure 8.3: Formulation of the cognitive diagnosis rule

In summary, the intelligent systems will be able to offer the most appropriate choice of generalised constraint to solve a user-defined problem, matching the characteristics of the problem with the position of the constraint in the multi-perspective space. In addition, the system will be capable of analysing user characteristics. The formulation of the user-analysis rule involves perception based diagnostic or preference information. This perception based information is explicitated into generalised constraints while using the developed domain as an explanatory database. The analysis itself is performed as generalised constraint propagation. Both explicitation of perception based information and generalised constraint propagation involve the computational theory of perceptions [32,33], which is still being developed and suggested as a computational inference engine complementary to qualitative reasoning approaches in artificial intelligence. Figure 8.3 visualises the formulation of the cognitive diagnosis rule. Generalised constraint propagation provides for the valued explanatory element in cognitive analysis. Within a range of formulations of the diagnostic rule, this approach will comfortably achieve the diagnostic objectives. However, the propagation becomes more complex when the rule involves heterogeneous constraints, which is it includes generalised constraints of various relational type in the imprecision dimension. Then we may want to involve evolutionary computing, slipping into a subexplanatory zone. The argument is that it will work within exactly the same reasonably explained cognitive-analysis framework, though bringing the results more efficiently.

Analysing user characteristics, in any of the two operational modes of the intelligent system, works toward improving the quality of tutoring or improving the efficiency of decision support, both in the domain of asset evaluation and risk analysis. Also, the computational theory of perceptions employed here is part of the soft computing paradigm. Thus in this Chapter, we continue developing the basic idea of this thesis, which is the implementation and the synergetic fusion of soft computing approaches to the benefit of the domain of asset risk analysis.

8.2 Conclusions

The motivation in this thesis is to reformulate asset evaluation and risk analysis within the soft computing paradigm. We start with introducing a procedure applicable to fuzzy evaluation of various crisp asset pricing models. The evaluation is performed through processing a wide range of pricing factor imprecision and relaxing assumptions on market behaviour. Thus the solution is inclusive rather than exclusive, and provides a basis for further analysis. As part of this analysis, two measures are formulated focusing on different characteristics of the assets, the risk of overvaluation and the robustness toward further imprecision. Those two measures are used as a point of departure in developing an asset ranking technique. Assets are initially ordered in relation to the risk values and then their positions are adjusted according to the robustness values. For those with relatively close risk measures, assets with a qualitatively higher robustness measure are ranked higher. Thus the final ranking informs agents about attractive less risky and highly robust assets. This conclusion is the basis for building an asset classifier discriminating between assets in accordance with agent-dependent thresholds for the risk and the robustness values. We focus on training the module that discriminates between the risk values, i.e. training a risk classifier. This involves training a fuzzy neural network, which is a complex problem that can not be accomplished by single-level evolution. Alternatively, we design a two-level algorithm that successfully trains the network using a dynamic objective function. The number of decomposition levels, the partitioning into subtasks at each decomposition level, the number and size of the subtasks in each partitioning, i.e. the efficient decomposition sequence is automatically identified, while exploiting the flexibility to discard unpromising and keep favourable subpopulations throughout the decomposition steps. The incremental part follows in reverse the efficient sequence and merges the subtasks toward optimizing the overall solution. We also suggest directions for further improvements in the algorithm. Considering the programming stage, we use Matlab 6.5 and its toolboxes. Particularly, for configuration of the crisp network we use the neural network toolbox, and for the network training we choose the fastest backpropagation algorithm available in the toolbox and modify the source code to accommodate for the sign restrictions on the network weights. On the other hand, we program the fuzzy network and the evolutionary training algorithms, as there are no helpful functions available in the toolboxes.

Finally, the suggested asset evaluation procedure, the formulated risk and robustness measures, the developed asset ranking technique and the trained risk classifier, along with further models, are all included in a domain representation in asset pricing and risk analysis. A knowledge representation framework is designed based on multiple models positioned within a multiperspective domain, where different relational types unfold along the different dimensions. The imprecision perspective has a special role and introduces the transformation of the multimodel space into a multiple constraint domain. The developed knowledge representation framework will be embedded into an intelligent system in asset pricing and risk analysis. This may operate as an intelligent tutoring system or as a decision support system. In both modes, the system will be able to analise user characteristics, either student understanding or decision-maker preferences, and this works toward improving the quality of tutoring or the efficiency of decision support in the area asset risk analysis. The analysis is facilitated by the introduced domain representation framework, and employs generalized constraint propagation, which relates to the computational theory of perceptions. Thus we have exploit the fusion of various soft techniques, from fuzzy logic, through neural networks and evolutionary algorithms, to the computational theory of perception to achieve synergetic results in handling complex problems to the benefit of the area of asset risk analysis.

References

[1] Shiller R.: *Irrational Exuberance*, Princeton University Press, 2001.

[2] Sharpe W.: Capital asset prices: A theory of market equilibrium under conditions of risk, *Journal of Finance*, vol. 19, pp 425-442, 1964.

[3] Fama E.: The behaviour of stock market prices, *Journal of Business*, vol. 38, pp 34-105, 1965.

[4] Lintner J.: The aggregation of investors' diverse judgements and preferences in perfectly competitive security markets, *Journal of Financial and Quantitative Analysis*, vol. 4, pp 347-400, 1969.

[5] Friend I. and Blume M.: Measurement of portfolio performance under uncertainty, *American Economic Review*, vol. 60, pp 561-575, 1970.

[6] Black F., Jensen M. and Scholes M.: The capital asset pricing model: Some empirical tests. In Jensen M., Editor: *Studies in the Theory of Capital Markets*, pp 79-124, Praeger Publishers, 1972.

[7] Mayers D.: Non-marketable assets and the capital market equilibrium under uncertainty. In Jensen M., Editor: *Studies in the Theory of Capital Markets*, pp 223-248, Praeger Publishers, 1972.

[8] Black F.: Capital market equilibrium with restricted borrowing, *Journal of Business*, vol. 45, pp 444-455, 1972.

[9] Merton R.: An intertemporal capital asset pricing model, *Econometrica*, vol. 41, pp 867-887, 1973.

[10] Basu S.: Investment performance of common stocks in relation to their price-earnings ratios: A test of the efficient market hypothesis, *Journal of Finance*, vol. 32, pp 663-682, 1977.

[11] Roll R.: A critique of the asset pricing theory's tests, *Journal of Financial Economics*, vol. 4, pp 129-176, 1977.

[12] Ross S.: The capital asset pricing model, short sales restrictions and related issues, *Journal of Finance*, vol. 32, pp 177-184, 1977.

[13] Litzenberger R. and Ramaswamy K.: The effect of personal taxes and dividends on capital asset prices: Theory and empirical evidence, *Journal of Financial Economics*, vol. 7, pp 163-195, 1979.

[14] Banz R.: The relation between return and market value of common stocks, *Journal of Financial Economics*, vol. 9, pp 3-18, 1981.

[15] Reinganum M.: Misspecification of capital asset pricing: Empirical anomalies based on earnings yields and market values, *Journal of Financial Economics*, vol. 9, pp 19-46, 1981.

[16] Ross S.: The arbitrage theory of capital asset pricing, *Journal of Economic Theory*, vol. 13, pp 341-360, 1976.

[17] Roll R. and Ross S.: An empirical investigation of the arbitrage pricing theory, *Journal of Finance*, vol. 35, pp 1073-1103, 1980.

[18] Reinganum M.: The arbitrage pricing theory: Some empirical results, *Journal of Finance*, vol. 36, pp 313-321, 1981.

[19] Chen N.: Some empirical tests of the theory of arbitrage pricing, *Journal of Finance*, vol. 38, pp 1393-1414, 1983.

[20] Chen N., Roll R. and Ross S.: Economic forces and the stock market: Testing the APT and alternative asset pricing theories, *UCLA Working Papers*, vol. 20, 1983.

[21] Stambaugh R.: Arbitrage pricing with information, *Journal of Financial Economics*, vol. 12, pp 357-369, 1983.

[22] Jagannathan R. and Viswanathan S.: Linear factor pricing, term structure of interest rates and the small firm anomaly, *Northwestern University Finance Working Papers*, vol. 57, 1988.

[23] McElroy M. and Burmeister E.: Arbitrage pricing theory as a restricted nonlinear multivariate regression model, *Journal of Business Economics and Statistics*, vol. 6, pp 29-42, 1988.

[24] Connor G. and Korajczyk R.: An intertemporal equilibrium beta pricing model, *Review of Financial Studies*, vol. 2, pp 373-392, 1989.

[25] Latham M.: The arbitrage pricing theory and supershares, *Journal of Finance*, vol. 44, pp 263-281, 1989.

[26] Bansal R. and Viswanathan S.: No arbitrage and arbitrage pricing: A new approach, *Journal of Finance*, vol. 48, pp 1231-1262, 1993.

[27] Fama E. and French K.: Common risk factors in the returns on stocks and bonds, *Journal of Financial Economics*, vol. 33, pp 3-56, 1993.

[28] Connor G. and Korajczyk R.: The arbitrage pricing theory and multifactor models of asset returns. In Jarrow R., Maksimovic V. and Ziemba W., Editors: *Handbooks in Operations Research and Management Science, vol. 9: Finance*, pp 87-144, Elsevier, 1995.

[29] Zadeh L.: Fuzzy logic = computing with words, *IEEE Transactions on Fuzzy Systems*, vol. 4, pp 103-111, 1996.

[30] Zadeh, L.: Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, vol. 90, pp 111-127, 1997.

[31] Zadeh, L.: From computing with numbers to computing with words - from manipulation of measurements to manipulation of perceptions, *IEEE Transactions on Circuits and Systems*, vol. 45, pp 105-119, 1999.

[32] Zadeh, L.: Outline of computational theory of perceptions based on computing with words. In Gupta M. and Sinha N., Editors, *Soft Computing and Intelligent Systems: Theory and Applications*, pp 3-22, Academic Press, 2000.

[33] Zadeh, L.: A new direction in AI: Toward a computational theory of perceptions, *Artificial Intelligence Magazine*, vol. 22, pp 73-84, 2001.

[34] Aluja J.: Towards a new paradigm of investment selection in uncertainty, *Fuzzy Sets and Systems*, vol. 84, pp 187-197, 1996.

[35] Buckley J.: Solving fuzzy equations in economics and finance, *Fuzzy Sets and Systems*, vol. 48, pp 289-296, 1992.

[36] Buckley J.: The fuzzy mathematics of finance, *Fuzzy Sets and Systems*, vol. 21, pp 257-273, 1987.

[37] LiCalzi M.: Towards a general setting for the fuzzy mathematics of finance, *Fuzzy Sets and Systems*, vol. 35, pp 281-293, 1990.

[38] Kuchta D.: Fuzzy capital budgeting, *Fuzzy Sets and Systems*, vol. 111, pp 367-385, 2000.

[39] Dubois D. and Prade H., Editors: *Fundamentals of Fuzzy Sets*. In *Handbooks of Fuzzy Sets Series*, Kluwer Academic Publishers, 2000.

[40] Buckley J. and Chanas S.: A fast method of ranking alternatives using fuzzy numbers, *Fuzzy Sets and Systems*, vol. 30, pp 337-338, 1989.

[41] Yoon K.: A probabilistic approach to rank complex fuzzy numbers, *Fuzzy Sets and Systems*, vol. 80, pp 167-176, 1996.

[42] Cheng C.-H.: A new approach for ranking fuzzy numbers by distance method, *Fuzzy Sets and Systems*, vol. 95, pp 307-317, 1998.

[43] Wang X. and Kerre E.: Reasonable properties for the ordering of fuzzy quantities (I) and (II), *Fuzzy Sets and Systems*, vol. 118, pp 375-385 and pp 387-405, 2001.

[44] Buckley J., Eslami E. and Hayashi Y.: Solving fuzzy equations using neural nets, *Fuzzy Sets and Systems*, vol. 86, pp 271-278, 1997.

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

[46] Buckley J. and Hayashi Y.: Can neural nets be universal approximators for fuzzy functions?, *Fuzzy Sets and Systems*, vol. 101, pp 323-330, 1999.

[47] Yao X.: Evolving artificial neural networks, *Proceedings of the IEEE*, vol. 87, pp 1423-1447, 1999.

[48] Kalganova T.: Bidirectional incremental evolution in extrinsic evolvable hardware. In John J., Stoica A., Keymeulen D. and Colombano S., Editors, *Proceedings of the Second NASA/DoD Workshop on Evolvable Hardware*, pp 65-74, IEEE Press, 2000.

[49] Zadeh L.: The birth and evolution of fuzzy logic, *International Journal of General Systems*, vol. 17, pp 95-105, 1990.

[50] Klir G. and Yuan B., Editors: *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lofti Zadeh.* In *Advances in Fuzzy Systems - Applications and Theory Series*, World Scientific Publishing, 1996.

[51] Pedrycz W. and Gomide F.: *An Introduction to Fuzzy Sets: Analysis and Design.* In *Complex Adaptive Systems Series*, MIT Press, 1998.

[52] Zimmermann H-J.: *Fuzzy Set Theory: And Its Applications*, Kluwer Academic Publishers, 2001.

[53] Gerla G.: *Fuzzy Logic: Mathematical Tools for Approximate Reasoning.* In *Trends in Logic Series*, Kluwer Academic Publishers, 2001.

[54] Höhle U. and Rodabaugh S.: *Mathematics of Fuzzy Sets.* In *Handbooks of Fuzzy Sets Series*, Kluwer Academic Publishers, 1998.

[55] Mordeson J. and Nair P.: *Fuzzy Mathematics.* In *Studies in Fuzziness and Soft Computing Series*, Physica-Verlag, 2001.

[56] Mendel J.: *Introduction to Rule-based Fuzzy Logic Systems*, IEEE Press, 2001.

[57] Dubois D and Prade H.: *Possibility Theory: An Approach to Computerised Processing of Uncertainty*, Kluwer Academic Publishers, 1988.

[58] Bojadziev G. and Bojadziev M.: *Fuzzy Logic for Business, Finance, and Management.* In *Advances in Fuzzy Systems - Applications and Theory Series*, World Scientific Publishing, 1997.

[59] Peray K.: *Mutual Fund Investing by Fuzzy Logic*, St Lucie Press, 1999.

[60] Ham F. and Kostanic I.: *Principles of Neurocomputing for Science and Engineering*, McGraw-Hill Education, 2001.

[61] Hornick K., Stinchcombe, M. and White H.: Multilayer feedforward networks are universal approximators, *Neural Networks*, vol. 2, pp 359–366, 1989.

[62] Werbos P.: *Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting.* In *Adaptive and Learning Systems Series for Signal Processing, Communications and Control*, John Wiley and Sons, 1994.

[63] Fahlman S.: Faster-learning variations on backpropagation: An empirical study. In Touretzky D., Hinton G. and Sejnowski T., Editors, *Proceedings of the Connectionist Model Summer School*, Morgan Kaufmann, 1988.

[64] Tollensere T.: SuperSAB: fast adaptive backpropagation with good scaling properties, *Neural Networks*, vol. 3, pp 561–573, 1990.

[65] Refenes A.-P., Editor: *Neural Networks in the Capital Markets*, John Wiley and Sons, 1994.

[66] Refenes A.-P., Abu-Mostafa Y. and Moody J., Editors: *Neural Networks in Financial Engineering*, World Scientific Publishing, 1996.

[67] Trippi R. and Turban E., Editors: *Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real-world Performance*, Irwin Professional, 1996.

[68] Fairclough D. and Hunter J.: The ex-ante classification of takeover targets using neural networks. In Refenes A.-P., Burgess A. and Moody J., Editors, *Decision Technologies for Computational Finance*, pp 381-388, Kluwer Academic Publishers, 1998.

[69] Shadbolt J. and Taylor J., Editors: *Neural Networks and the Financial Markets: Predicting, Combining and Portfolio Optimisation.* In *Perspectives in Neural Computing Series*, Springer-Verlag, 2002.

[70] Fogel, D.: *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, John Wiley and Sons, 1999.

[71] Baeck T., Fogel D. and Michalewicz Z., Editors: *Evolutionary Computation 2: Advanced Algorithms and Operators*, The Institute of Physics, 2000.

[72] Bonissone P., Editor: *Evolutionary Computation and Applications*, IEEE Press, 2000.

[73] Mitchell M.: *An Introduction to Genetic Algorithms.* In *Complex Adaptive Systems Series*, MIT Press, 1998.

[74] Koza J., Keane M., Streeter M., Mydlowec W., Yu J. and Lanza G.: *Genetic Programming: Routine Human-Competitive Machine Intelligence.* In *Genetic Programming Series*, Kluwer Academic Publishers, 2003.

[75] Schwefel H.: *Evolution and Optimum Seeking.* In *Sixth Generation Computer Technologies Series*, John Wiley and Sons, 1995.

[76] Fogel L.: *Intelligence Through Simulated Evolution: Four Decades of Evolutionary Programming.* In *Series on Intelligent Systems*, John Wiley and Sons, 1999.

[77] Deb K.: *Multi-objective Optimization Using Evolutionary Algorithms.* In *Interscience Series in Systems & Optimization*, John Wiley and Sons, 2001

[78] Branke J.: *Evolutionary Optimization in Dynamic Environments.* In *Genetic Algorithms and Evolutionary Computation Series*, Kluwer Academic Publishers, 2001.

[79] Branke J. and Deb K.: Integrating user preferences into evolutionary multi-objective optimization. In Jin Y., Editor, *Knowledge Incorporation in Evolutionary Computation*, Springer, to appear.

[80] Bauer R.: *Genetic Algorithms and Investment Strategies: Alternative Approach to Neural Networks and Chaos Theory.* In *Finance Editions Series*, John Wiley and Sons, 1994.

[81] Chen S.-H.: *Genetic Algorithms and Genetic Programming in Computational Finance*, Kluwer Academic Publishers, 2002.

[82] Jain L., Editor: *Evolutionary Computation in Economics and Finance.* In *Studies in Fuzziness and Soft Computing Series*, Physica-Verlag, 2002.

[83] Shafer G.: *Mathematical Theory of Evidence*, Princeton University Press, 1992.

[84] Yager R., Fedrizzi M. and Kacprzyk J., Editors: *Advances in the Dempster-Theory of Evidence*, John Wiley and Sons, 1993.

[85] Pearl J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1997.

[86] Koop G.: An objective Bayesian analysis of common stochastic trends in international stock prices and exchange rates, *Journal of Empirical Finance,* vol. 1, pp 343-364. 1994.

[87] Jammernegg W.: *Sequential Binary Investment Decisions: A Bayesian Approach.* In *Lecture Notes in Economics and Mathematical Systems Series*, Springer-Verlag, 1998.

[88] Shafer G. and Vovk V.: *Probability and Finance: It's Only a Game!* In *Series in Probability and Statistics*, John Wiley and Sons, 2001.

[89] Annaert J., van den Broeck J. and Vennet R.: Determinants of mutual fund underperformance: A Bayesian stochastic frontier approach, *European Journal of Operational Research*, vol. 151, pp 617-632, 2003.

[90] Hayashi Y. and Buckley J.: Approximations between fuzzy expert systems and neural networks, *International Journal of Approximate Reasoning*, vol. 10, pp 63-73, 1994.

[91] Buckley J., Hayashi Y. and Czogała E.: On the equivalence of neural nets and fuzzy expert systems, *Fuzzy Sets and Systems*, vol. 100, pp 145-150, 1999.

[92] Buckley J. and Hayashi Y.: Hybrid neural nets can be fuzzy controllers and fuzzy expert systems, *Fuzzy Sets and Systems*, vol. 60, pp 135-142, 1993.

[93] Buckley J. and Hayashi Y.: Neural nets for fuzzy systems, *Fuzzy Sets and Systems*, vol. 71, pp 265-276, 1995.

[94] Eom K., Jung K. and Sirisena H.: Performance improvement of backpropagation algorithm by automatic activation function gain tuning using fuzzy logic, *Neurocomputing*, vol. 50, pp 439-460, 2003.

[95] Takagi H. and Hayashi I.: NN-driven fuzzy reasoning, *International Journal of Approximate Reasoning*, vol. 5, pp 191-212, 1991.

[96] Hayashi I., Nomura H., Yamasaki H. and Wakami N.: Construction of fuzzy inference rules by NDF and NDFL, *International Journal of Approximate Reasoning*, vol. 6, pp 241-266, 1992.

[97] Nauck D. and Kruse R.: A neuro-fuzzy method to learn fuzzy classification rules from data, *Fuzzy Sets and Systems*, vol. 89, pp 277-288, 1997.

[98] Buckley J., Feuring T. and Hayashi Y.: Neural net solutions to fuzzy linear programming, *Fuzzy Sets and Systems*, vol. 106, Pages 99-111, 1999.

[99] Buckley J. and Hayashi Y.: Fuzzy neural networks: A survey, *Fuzzy Sets and Systems*, vol. 66, pp 1-13, 1994.

[100] Buckley J. and Hayashi Y.: Can fuzzy neural nets approximate continuous fuzzy functions?, *Fuzzy Sets and Systems*, vol. 61, pp 43-51, 1994.

[101] Feuring T. and Lippe W.-M.: The fuzzy neural network approximation lemma, *Fuzzy Sets and Systems*, vol. 102, pp 227-236, 1999.

[102] Liu P.: Analyses of regular fuzzy neural networks for approximation capabilities, *Fuzzy Sets and Systems*, vol. 114, pp 329-338, 2000.

[103] Liu P.: Universal approximations of continuous fuzzy-valued functions by multi-layer regular fuzzy neural networks, *Fuzzy Sets and Systems*, vol. 119, pp 313-320, 2001.

[104] Keller J., Hayashi Y. and Chen Z.: Additive hybrid networks for fuzzy logic, *Fuzzy Sets and Systems*, vol. 66, pp 307-313, 1994.

[105] Voget S. and Kolonko M.: multidimensional optimisation with a fuzzy genetic algorithm, *Journal of Heuristics*, vol. 4, pp 221-244, 1998.

[106] Streifel, R., Marks R., Red R., Choi J. and Healy M.: Dynamic fuzzy control of genetic algorithms parameter coding, *IEEE Transactions on Systems, Man and Cybernetics. Part B*, vol. 29, pp 426-433, 1999.

[107] Herrera F. and Lozano M.: Adaptive genetic operators based on coevolution with fuzzy behaviours, *IEEE Transactions on Evolutionary Computation*, vol. 5, pp 149 – 165, 2001.

[108] Herrera F., Lozano M. and Verdegay J.: Fuzzy connectives based crossover operators to model genetic algorithms population diversity, *Fuzzy Sets and Systems*, vol. 92, pp 21-30, 1997.

[109] Herrera F. and Lozano, M.: Gradual distributed real-coded genetic algorithms, *IEEE Transactions on Evolutionary Computation*, vol. 4, pp 43-63, 2000.

[110] Casillas J., Cordón O., del Jesus M. and Herrera F.: Genetic feature selection in a fuzzy rule-based classification system learning process for high-dimensional problems, *Information Sciences*, vol. 136, pp 135-157, 2001.

[111] Cordón O., Gomide F., Herrera F., Hoffmann F. and Magdalena L.: Ten years of genetic fuzzy systems: current framework and new trends, *Fuzzy Sets and Systems*, vol. 141, pp 5-31, 2004.

[112] Whitley D., Starkweather T. and Bogart C.: Genetic algorithms and neural networks: optimizing connections and connectivity, *Parallel Computing*, vol. 14, pp 347-361, 1990.

[113] Angeline P.; Saunders G. and Pollack J.: An evolutionary algorithm that constructs recurrent neural networks, *IEEE Transactions on Neural Networks*, vol. 5 ,pp 54 – 65, 1994.

[114] Larrañaga P. and Lozano J.: Synergies between evolutionary computation and probabilistic graphical models, *International Journal of Approximate Reasoning*, vol. 31, pp 155-156, 2002.

[115] Larranaga P., Kuijpers C., Murga R. and Yurramendi Y.: Learning Bayesian network structures by searching for the best ordering with genetic algorithms, *IEEE Transactions on Systems, Man and Cybernetics. Part A*, vol. 26, pp 487-493, 1996.

[116] Pelikan M., Goldberg D. and Lobo F.: A survey of optimization by building and using probabilistic models, in *Proceedings of the American Control Conference*, pp 3289-3293, IEEE Press, 2000.

[117] Pelikan M., Sastry K. and Goldberg D.: Scalability of the Bayesian optimization algorithm, *International Journal of Approximate Reasoning*, vol. 31, pp 221-258, 2002.

[118] Krishnamraju P., Buckley J., Reilly K. and Hayashi Y.: Genetic learning algorithms for fuzzy neural nets. In *Proceedings of the Third IEEE International Conference on Fuzzy Systems*, part of *IEEE World Congress on Computational Intelligence*, pp 1969-1974, 1994.

[119] Ishigami H., Fukuda T., Shibata T. and Arai F.: Structure optimization of fuzzy neural network by genetic algorithm, *Fuzzy Sets and Systems*, vol. 71, pp 257-264, 1995.

[120] Buckley J., Reilly K. and Penmetcha K.: Backpropagation and genetic algorithms for training fuzzy neural nets. In *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, pp 2-6, IEEE Press, 1996.

[121] Seng T., Khalid M. and Yusof R.: Tuning of a neuro-fuzzy controller by genetic algorithm, *IEEE Transactions on Systems, Man and Cybernetics. Part B*, vol. 29, pp 226-236, 1999.

[122] Chung I-F., Lin C.-J. and Lin C.-T.: A GA-based fuzzy adaptive learning control network, *Fuzzy Sets and Systems*, vol. 112, pp 65-84, 2000.

[123] Gupta M. and Sinha N., Editors: *Soft Computing and Intelligent Systems: Theory and Applications*, Academic Press, 2000.

[124] Azvine B., Azarmi N. and Nauck D., Editors: *Intelligent Systems and Soft Computing: Prospects, Tools and Applications*. In *Lecture Notes in Artificial* Intelligence Series, Springer-Verlag, 2000.

[125] Pedrycz W., Editor: *Granular Computing: An Emerging Paradigm*. In Studies in *Fuzziness and Soft Computing Series*, Physica-Verlag, 2001.

[126] Ding L., Editor: *New Paradigm of Knowledge Engineering by Soft Computing.* In *Fuzzy Logic Systems Institute (FLSI) Soft Computing Series*, World Scientific Publishing, 2001.

[127] Bouchon-Meunier B., Editor: *Technologies for Constructing Intelligent Systems: Tools.* In *Studies in Fuzziness & Soft Computing Series*, Physica-Verlag, 2002.

[128] Abraham A., Editor: *Recent Advances in Intelligent Paradigms and Applications.* In *Studies in Fuzziness and Soft Computing Series*, Physica-Verlag, 2003.

[129] Fogel D. and Robinson C., Editor: *Computational Intelligence: The Experts Speak*, John Wiley and Sons, 2003.

[130] Deboeck G., Editor: *Trading on the Edge: Neural, Genetic and Fuzzy Systems for Chaotic Financial Markets*, John Wiley and Sons, 1994.

[131] Refenes A.-P., Burgess A. and Moody J., Editors: *Decision Technologies for Computational Finance.* In *Advances in Computational Management Science Series*, Kluwer Academic Publishers, 1998.

[132] Ribeiro R., Zimmermann H.-J., Yager R. and Kacprzyk J., Editors: *Soft Computing in Financial Engineering.* In *Studies in Fuzziness and Soft Computing Series*, Physica-Verlag, 1999.

[133] Abu-Mostafa Y., LeBaron B., Lo A. and Weigent A., Editors: *Computational Finance*, MIT Press, 2000.

[134] Chen S.-H. and Wang P., Editors: *Computational Intelligence in Economics and Finance.* In *Advanced Information Processing Series*, Springer-Verlag, 2003.

[135] Campbell J., Lo A. and MacKinlay A.: *The Econometrics of Financial Markets*, Princeton University Press, 1997.

[136] Buckley J.: Solving fuzzy equations, *Fuzzy Sets and Systems*, vol. 50, pp 1-14, 1992.

[137] Buckley J. and Qu Y.: Solving systems of linear fuzzy equations, *Fuzzy Sets and Systems*, vol. 43, pp 33-43, 1991.

[138] Buckley J. and Qu Y.: Solving fuzzy equations: A new solution concept, *Fuzzy Sets and Systems*, vol. 39, pp 291-301, 1991.

[139] Buckley J. and Feuring T.: Fuzzy differential equations, *Fuzzy Sets and Systems*, vol. 110, pp 43-54, 2000.

[140] Buckley J. and Feuring T.: Introduction to fuzzy partial differential equations, *Fuzzy Sets and Systems*, vol. 105, pp 241-248, 1999.

[141] Mandelbrot B.: A multifractal walk down Wall Street, *Scientific American*, pp 50-53, 1999.

[142] Shiller R.: *Market Volatility*, MIT Press, 1992.

[143] Shiller R.: The New Financial Order: Risk in the 21st Century, Princeton University Press, 2003.

[144] Dubois D. and Prade H.: Fuzzy sets in approximate reasoning, Part 1: Inference with possibility distributions, *Fuzzy Sets and Systems*, vol. 100, pp 73-132, 1999.

[145] Buckley J. and Eslami E.: Neural net solutions to fuzzy problems: The quadratic equation, *Fuzzy Sets and Systems*, vol. 86, pp 289-298, 1997.

[146] White H.: *Artificial Neural Networks: Approximation and Learning*, Blackwell, 1992.

[147] Buckley J. and Feuring T.: Universal approximators for fuzzy functions, *Fuzzy Sets and Systems*, vol. 113, pp 411-415, 2000.

[148] Islam M., Yao X. and Murase K.: A constractive algorithm for training cooperative neural network ensembles, *IEEE Transactions on Neural Networks*, vol. 14, pp 820-834, 2003.

List of Publications

[P1] Hunter J. and Serguieva A.: Project risk evaluation, presented at the *Workshop on Finance and Capital Markets Research at the BAA-ICAEW Doctoral Colloquium*, Manchester, UK, 1999; extended presented at the *Second Annual Conference of the International Economics and Finance Society*, Uxbridge, UK, 1999.

[P2] Hunter J. and Serguieva A.: Project risk evaluation using an alternative to the standard present value criteria, presented at the *Seventh International PASE Workshop on Managing the Unexpected: Unmodelled Dynamics in Engineering and Finance*, Leuven, Belgium, 2000. Published in *Neural Network World*, vol. 10, pp 157–172, VSP, 2000. (ISSN 1210-0552)

[P3] Serguieva A. and Hunter J.: Investment risk appraisal, *Department of Economics and Finance Discussion Papers*, vol. 15, Brunel University, 2000. (ISSN 1466-5182)

[P4] Serguieva A. and Hunter J.: Fuzzy interval methods in investment risk appraisal, *Fuzzy Sets and Systems*, vol. 142, no. 3, pp 443-466, Elsevier, 2004. (ISSN 0165-0114) *

* This article is rated sixth in the list 'Top 25 of most downloaded articles' for the first quarter of 2004 in the website of Fuzzy Sets and Systems, Elsevier.

[P5] Serguieva A., Hunter J. and Kalganova T.: Soft computing in investment appraisal. In *Proceedings of the Second Conference of the European Society for Fuzzy Logic and Technology*, pp 214–219, Leicester, UK, 2001.

[P6] Serguieva A. and Kalganova T.: A neuro-fuzzy-evolutionary classifier of low-risk investments, presented at the *2002 World Congress on Computational Intelligence*, Honolulu, Hawaii. Published in Little P., Editor, *Proceedings of the Eleventh IEEE International Conference on Fuzzy Systems*, pp 997–1002, IEEE Press, 2002. (ISBN 0-7803-7280-8)

[P7] Serguieva A., Kalganova T. and Khan T.: An intelligent system for risk classification of stock investment projects, *Journal of Applied Systems Studies*, vol. 4, no. 2, pp 236-261, Cambridge International Science, 2003. (ISSN 1466-7738)

[P8] Serguieva A. and Khan T.: Knowledge representation in risk analysis, *Brunel School of Business and Management Working Papers*, vol. 1, Brunel University, 2004. (ISSN 1743-6494)

[P9] Serguieva A. and Khan T.: Domain representation assisting cognitive analysis, to be presented at the *Sixteenth European Conference on Artificial Intelligence* Valencia, Spain. To be published in Saitta L, Editor, as a volume of *Frontiers in Artificial Intelligence and Applications Series*, IOS Press, 2004. (ISBN 1-58603-???-?)

[P10] Serguieva A. and Khan T.: Student representation assisting cognitive analysis, to be presented at the *Seventh International Conference on Intelligent Tutoring Systems*, Maceio, Brazil. To be published in Lester J., Editor, as a volume of *Lecture Notes in Computer Science Series*, Springer-Verlag, 2004. (ISBN 3-540-?????-?)

Appendix A1: Fuzzified Data and Evaluated Fuzzy Share Price by Company

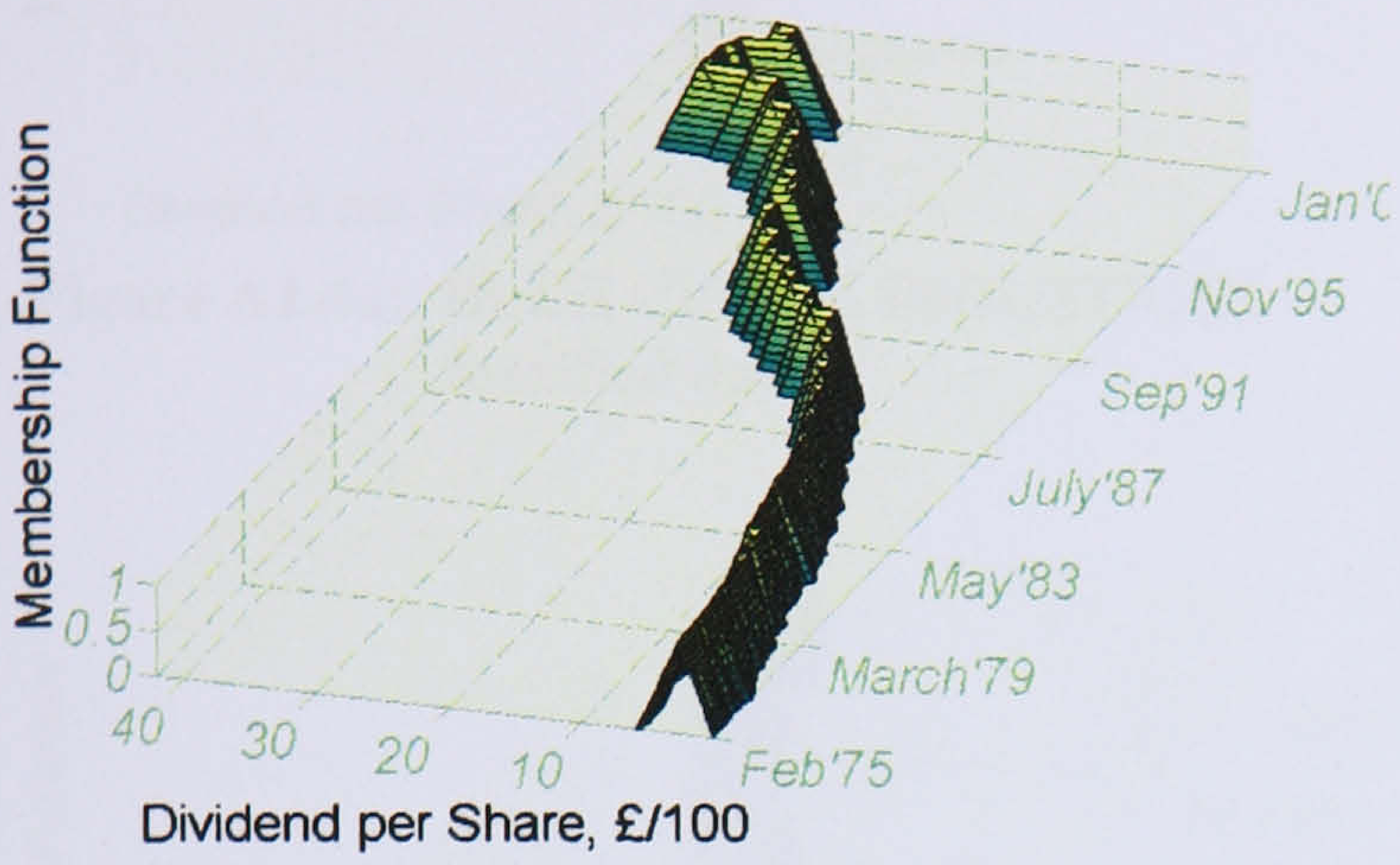


Figure A1.1a: BASS - fuzzified data

Figure A1.1b: BASS - evaluated fuzzy share price

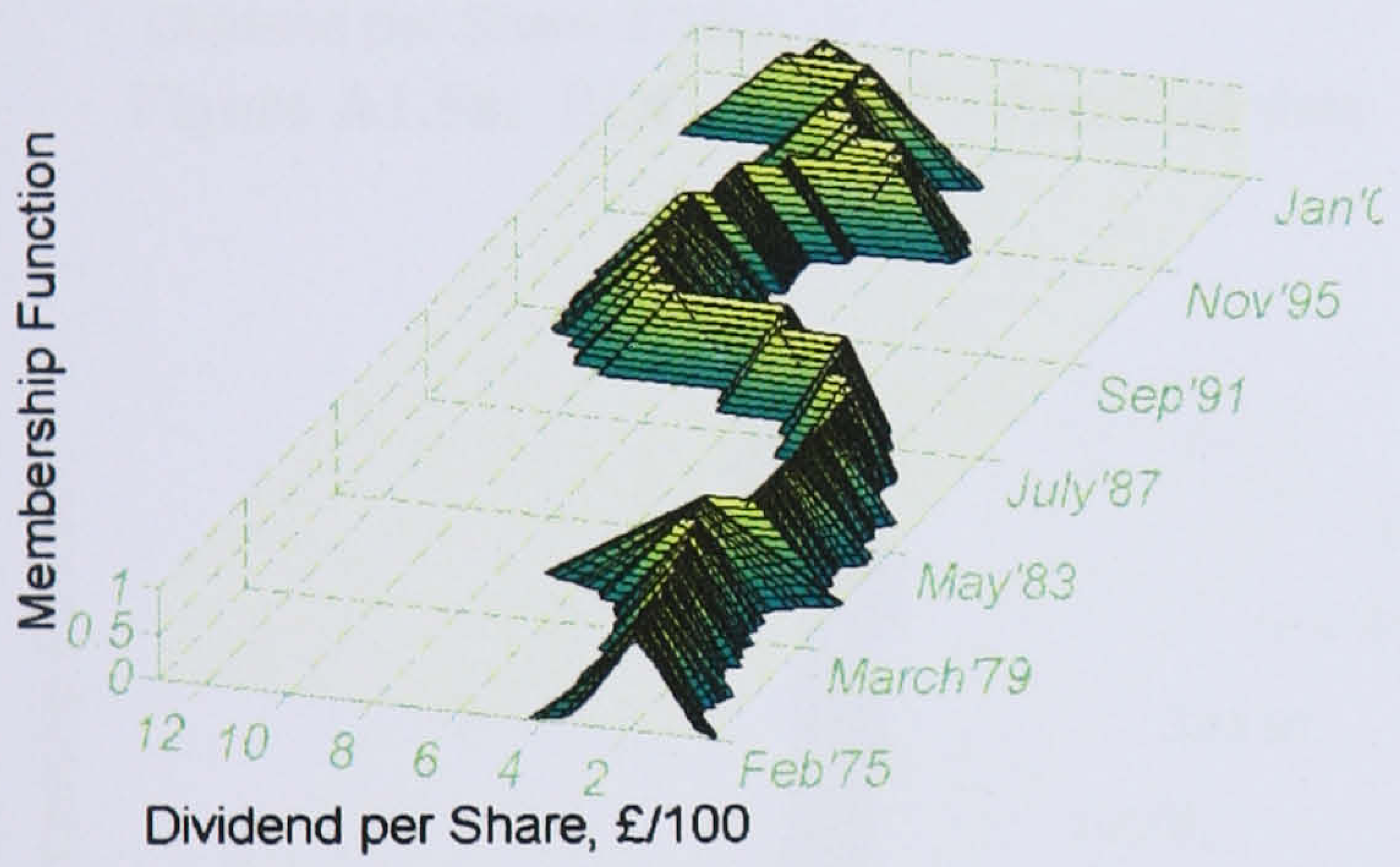


Figure A1.2a: BBA GROUP - fuzzified data

Figure A1.2b: BBA GROUP - evaluated fuzzy share price

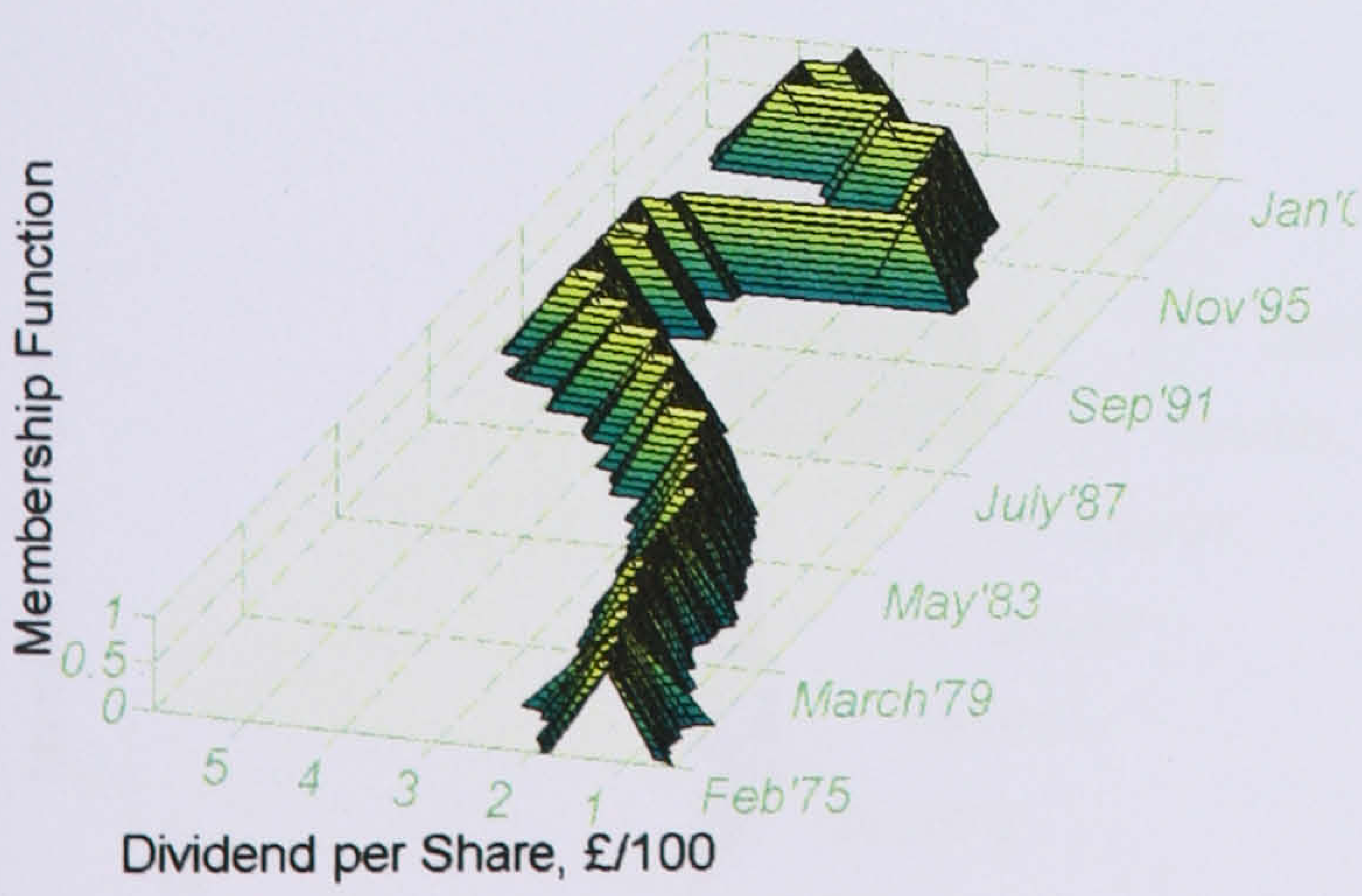


Figure A1.3a: BENTALS - fuzzified data

Figure A1.3b: BENTALS – evaluated fuzzy share price

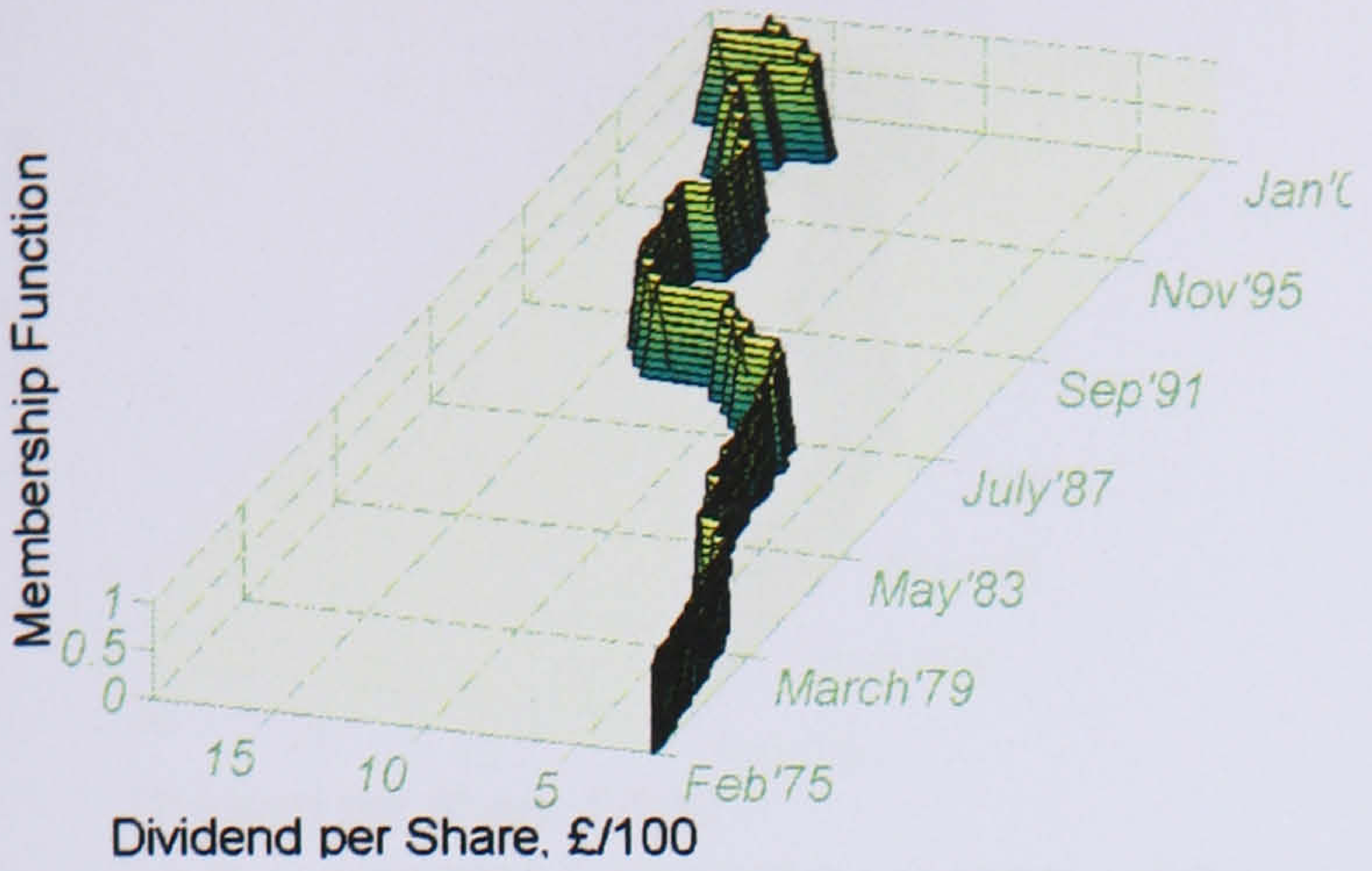


Figure A1.4a: BLUE CIRCLE INDUSTRIES - fuzzified data

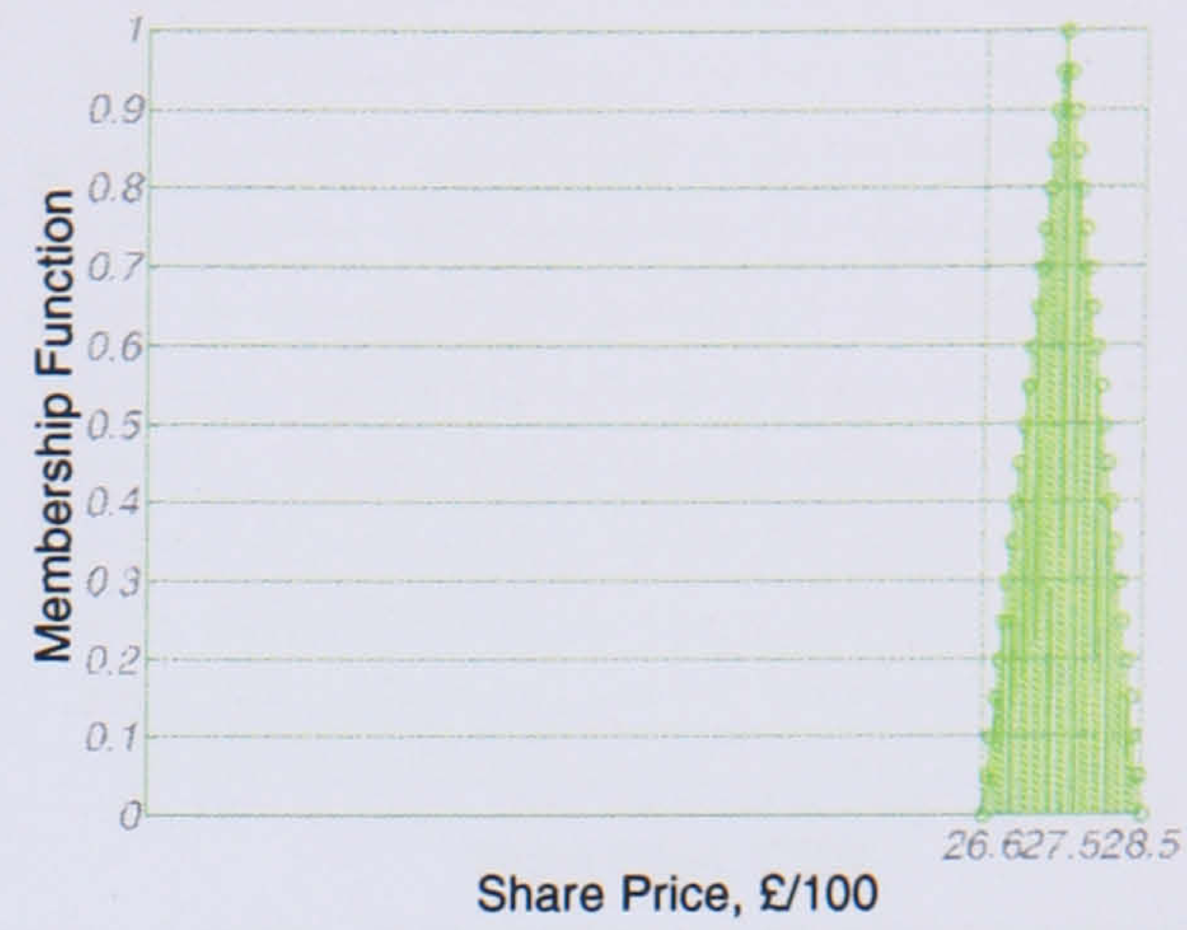


Figure A1.4b: BLUE CIRCLE INDUSTRIES - evaluated fuzzy share price

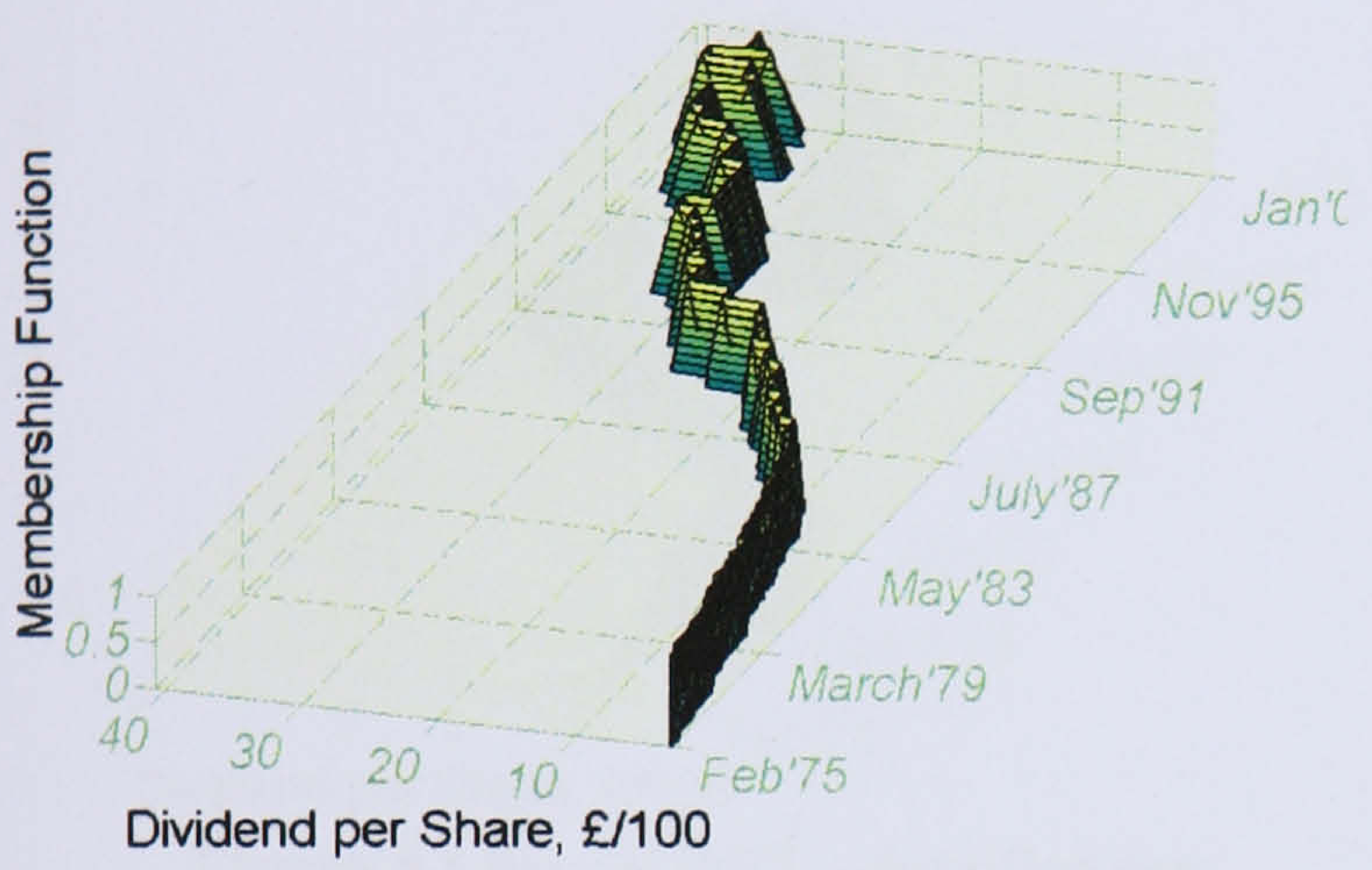


Figure A1.5a: BOC GROUP - fuzzified data

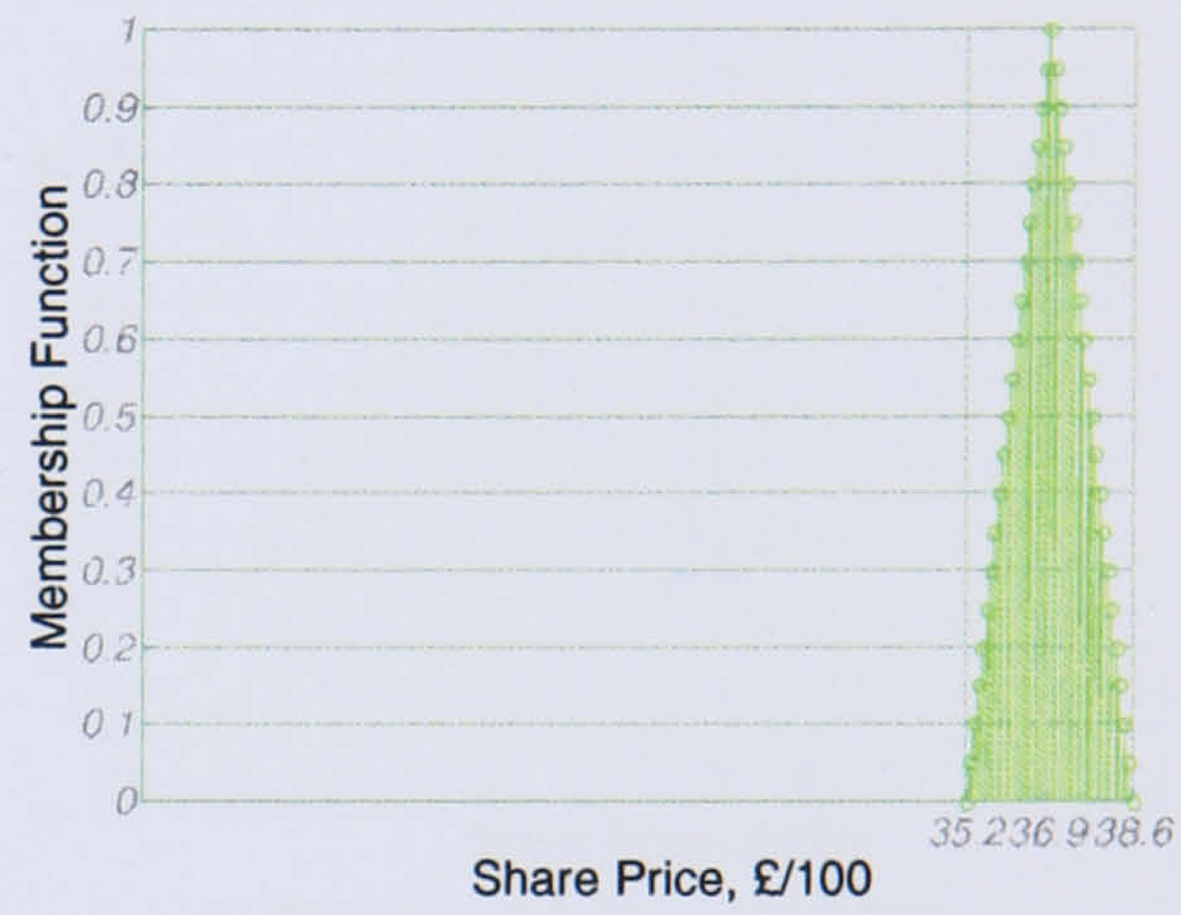


Figure A1.5b: BOC GROUP - evaluated fuzzy share price

Figure A1.6a: BOOTS CO. - fuzzified data

Figure A1.6b: BOOTS CO. - evaluated fuzzy share price

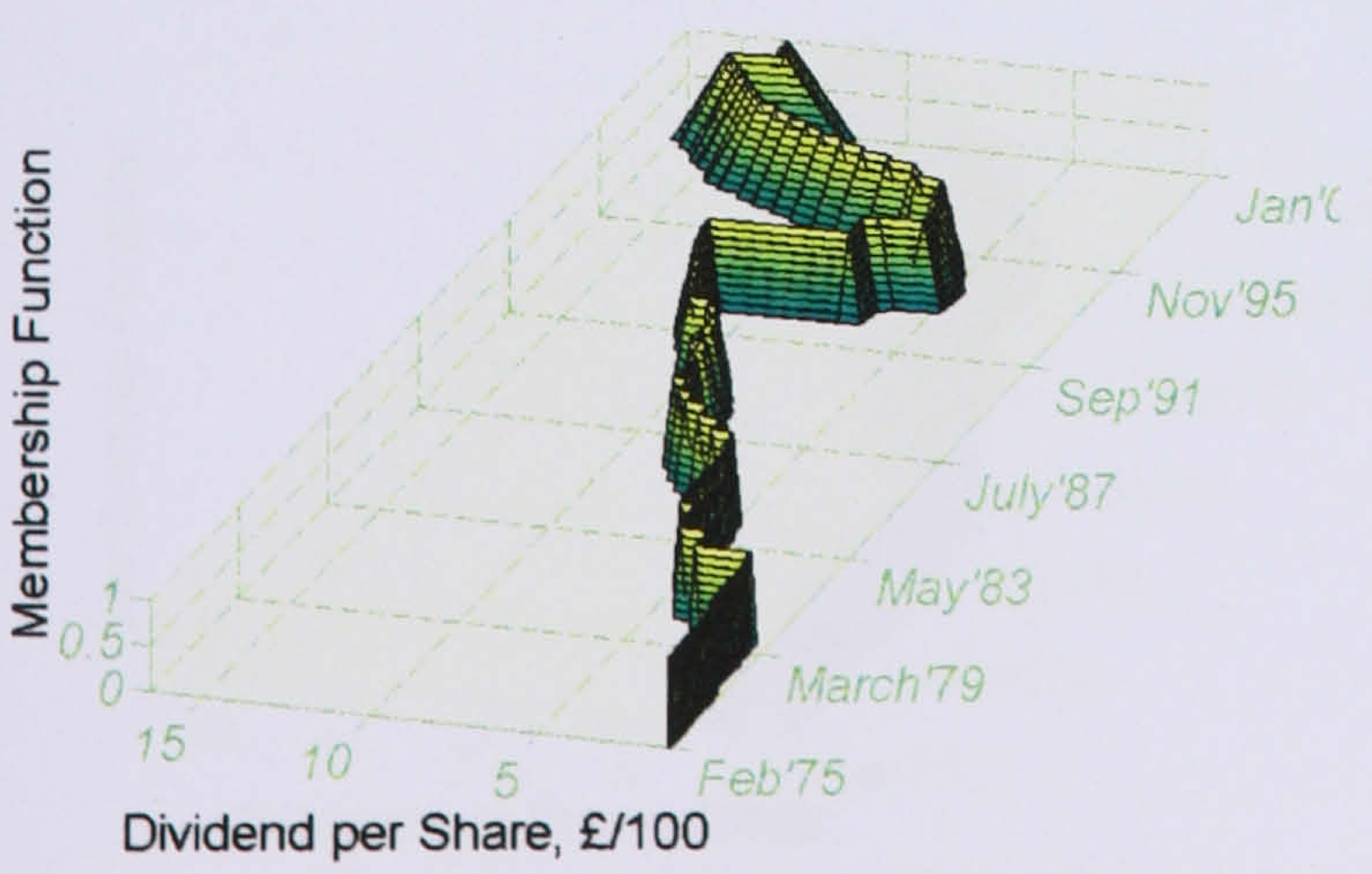


Figure A1.7a: BP AMOCO - fuzzified data

Figure A1.7b: BP AMOCO - evaluated fuzzy share price

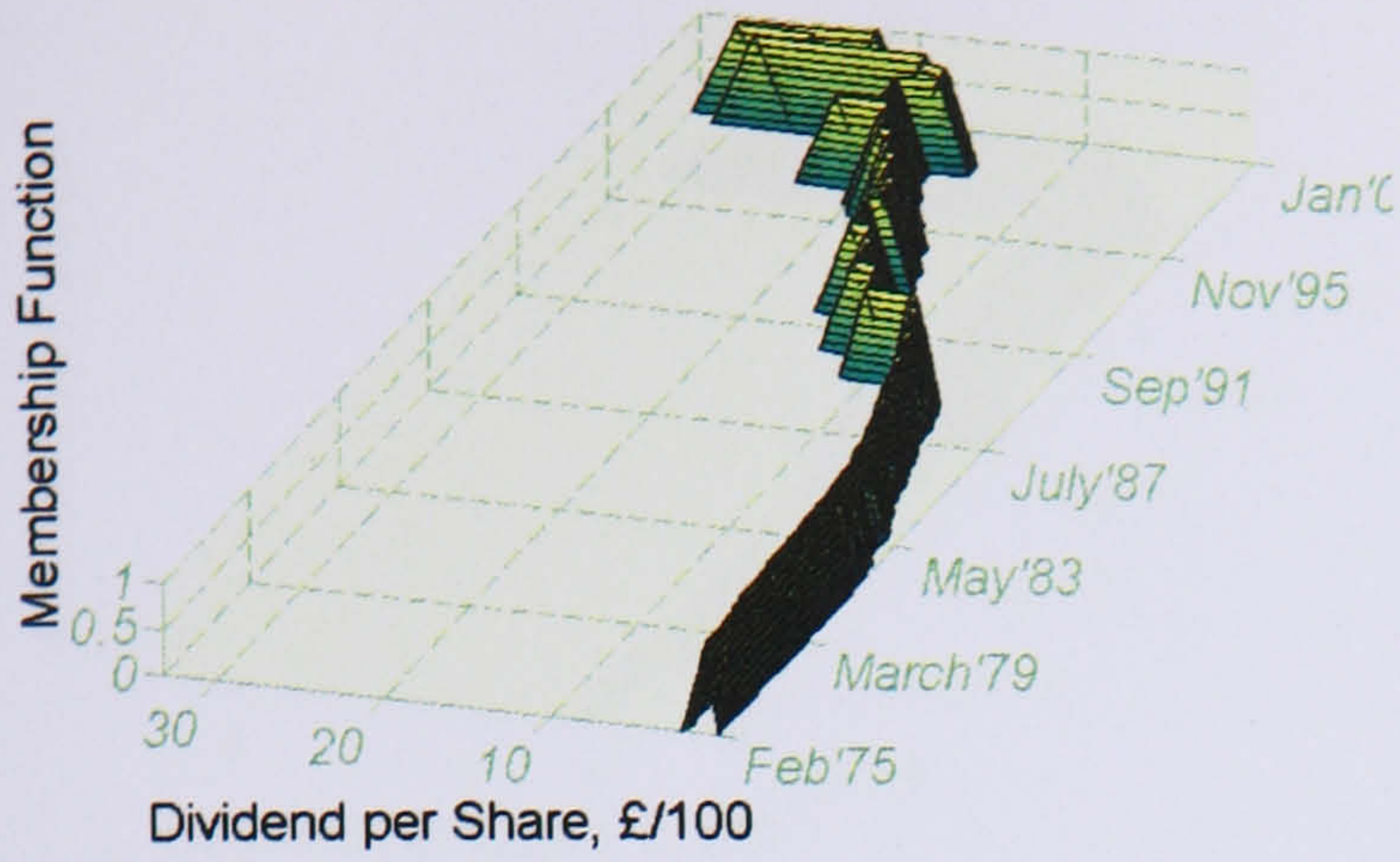


Figure A1.8a: BRITISH AMERICAN TOBACCO - fuzzified data

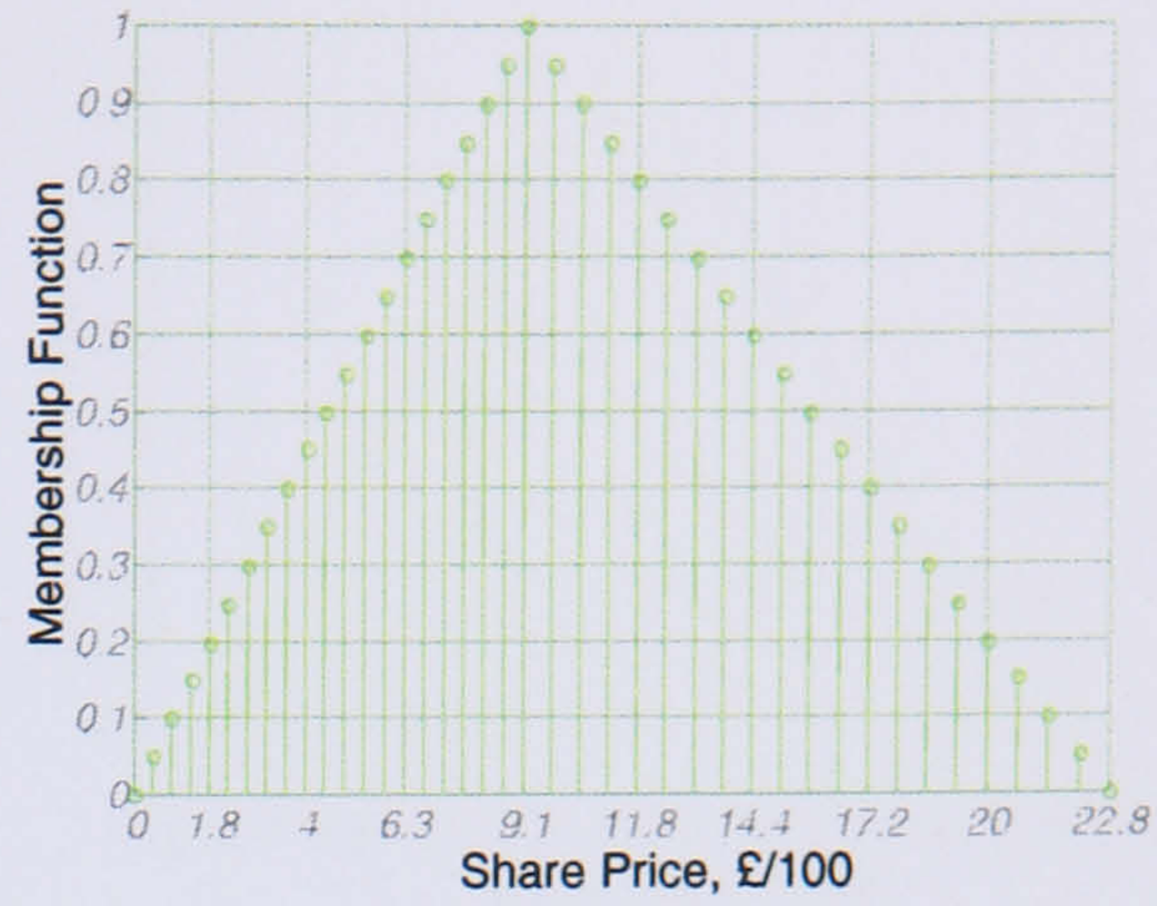


Figure A1.8b: BRITISH AMERICAN TOBACCO - evaluated fuzzy share price

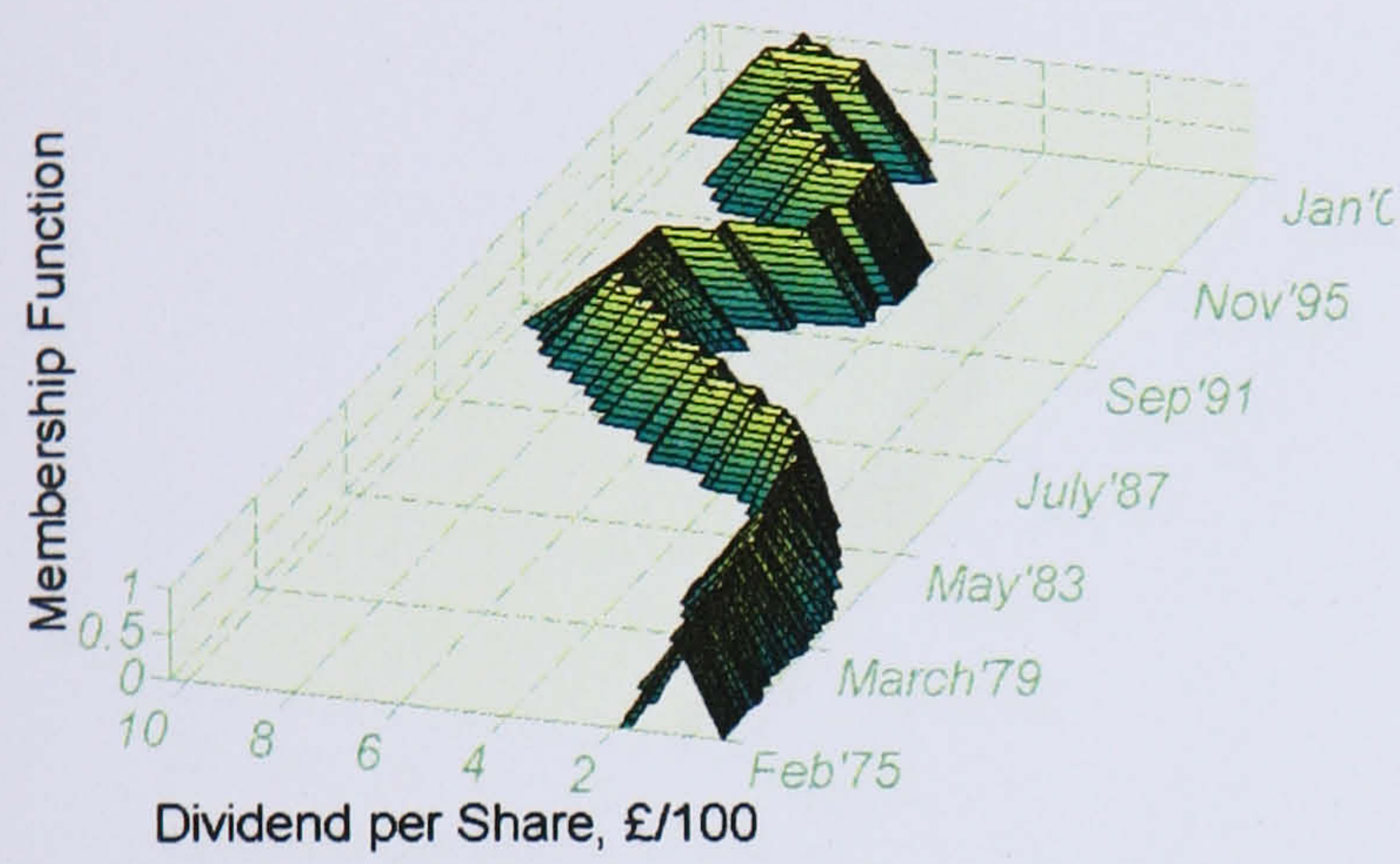


Figure A1.9a: BUNZL - fuzzified data

Figure A1.9b: BUNZL - evaluated fuzzy share price

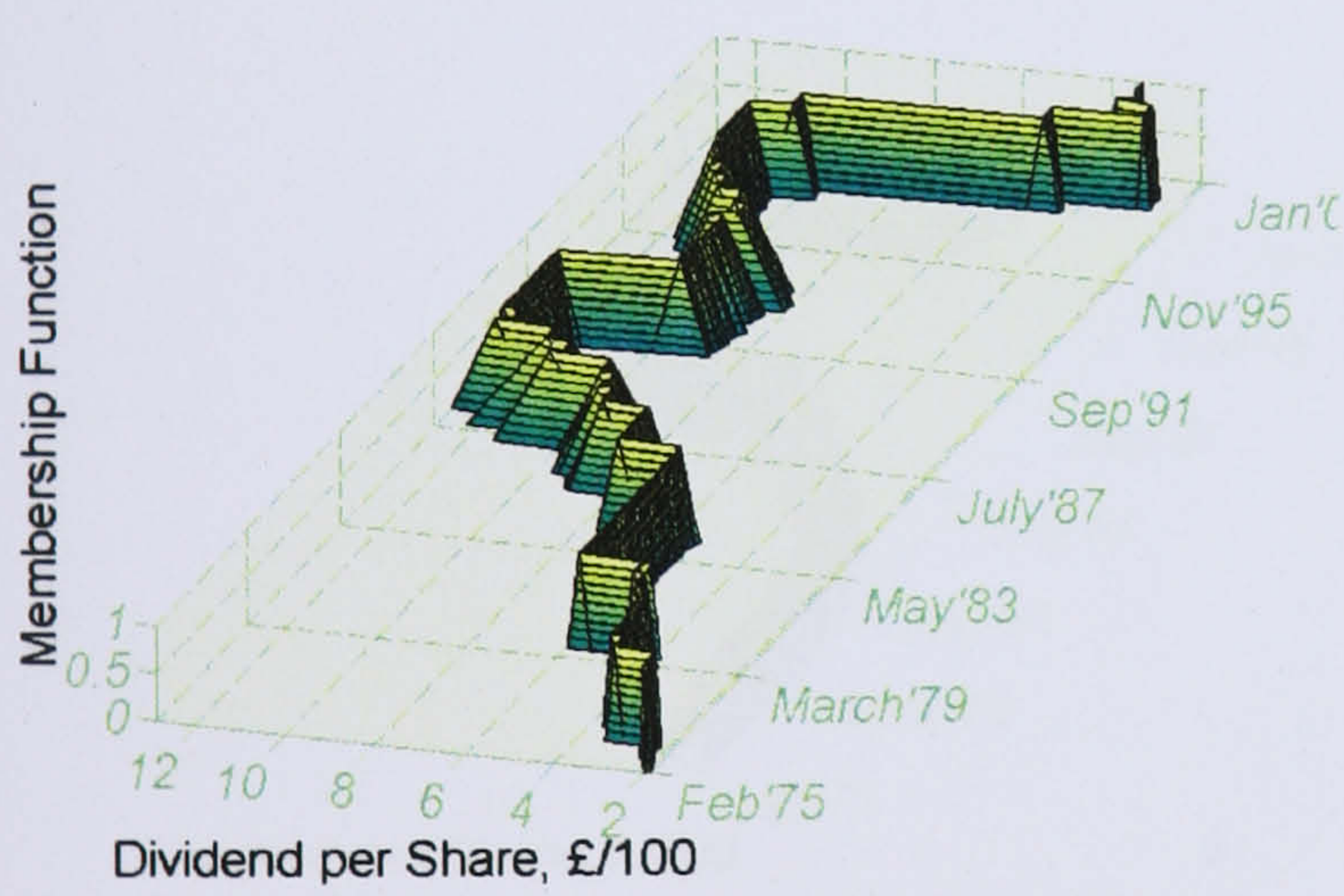


Figure A1.10a: COATS VIYELLA - fuzzified data

Figure A1.10b: COATS VIYELLA - evaluated fuzzy share price

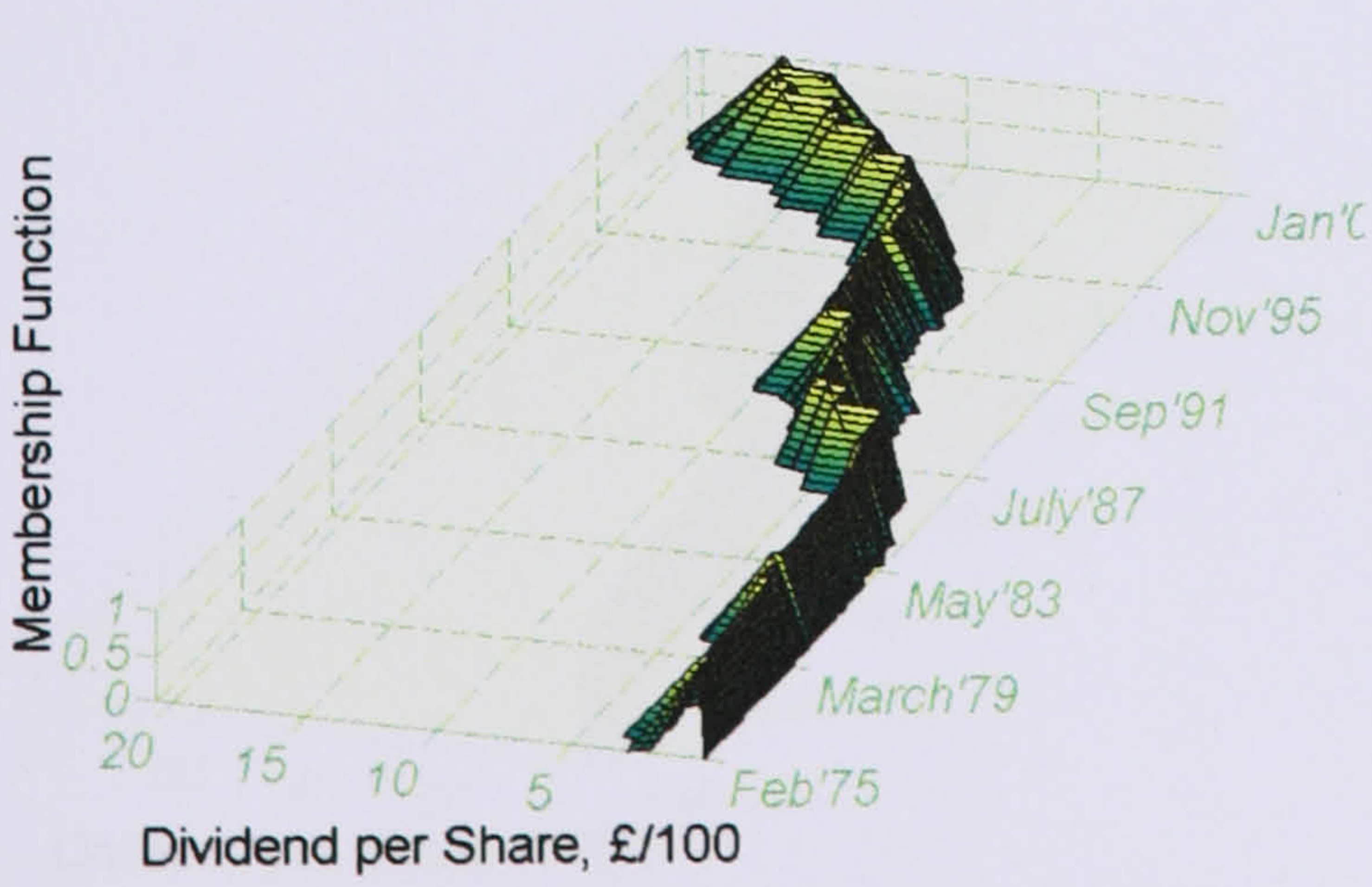


Figure A1.11a: DIXONS GROUP - fuzzified data

Figure A1.11b: DIXONS GROUP - evaluated fuzzy share price

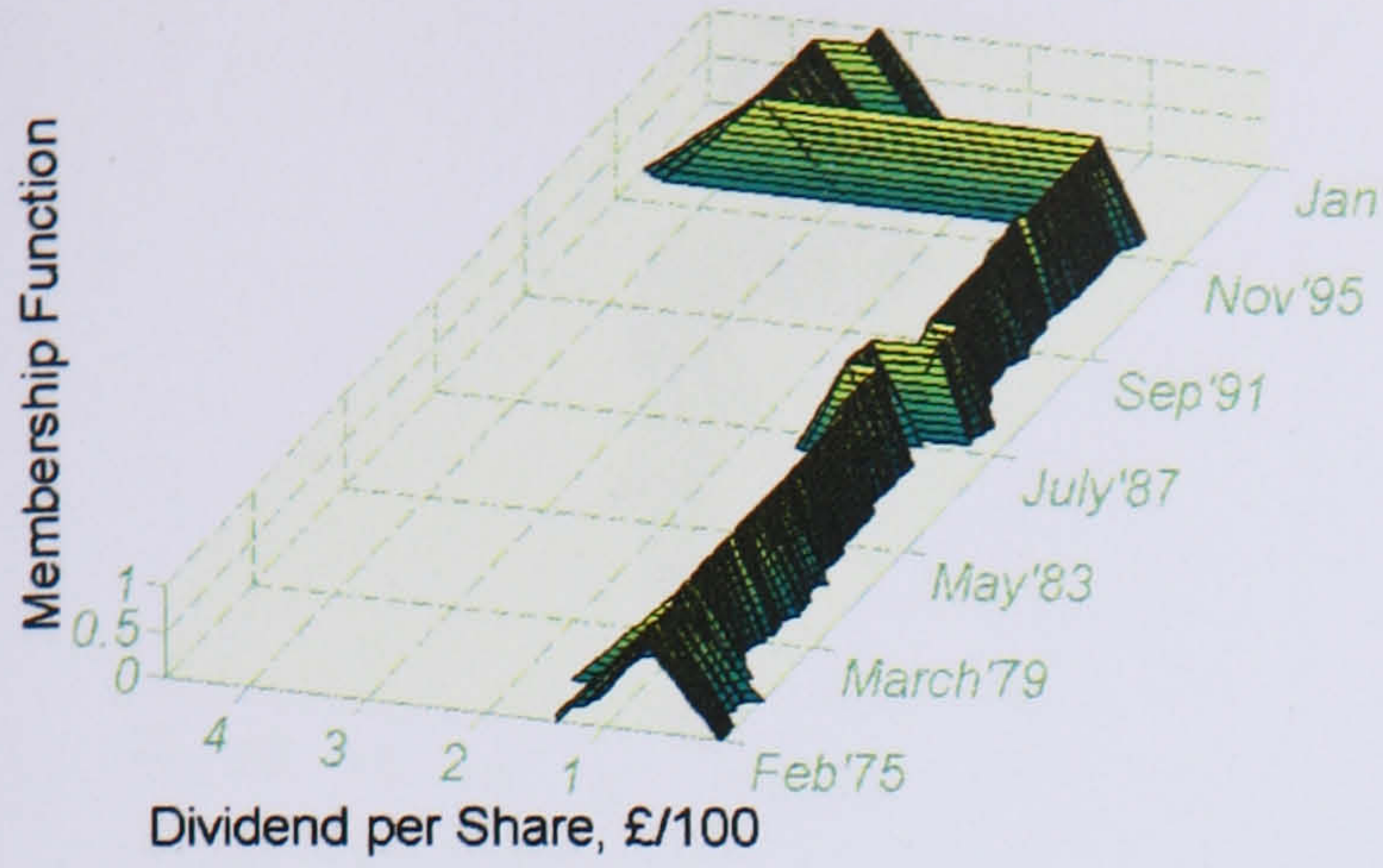


Figure A1.12a: GOODWIN - fuzzified data

Figure A1.12b: GOODWIN - evaluated fuzzy share price

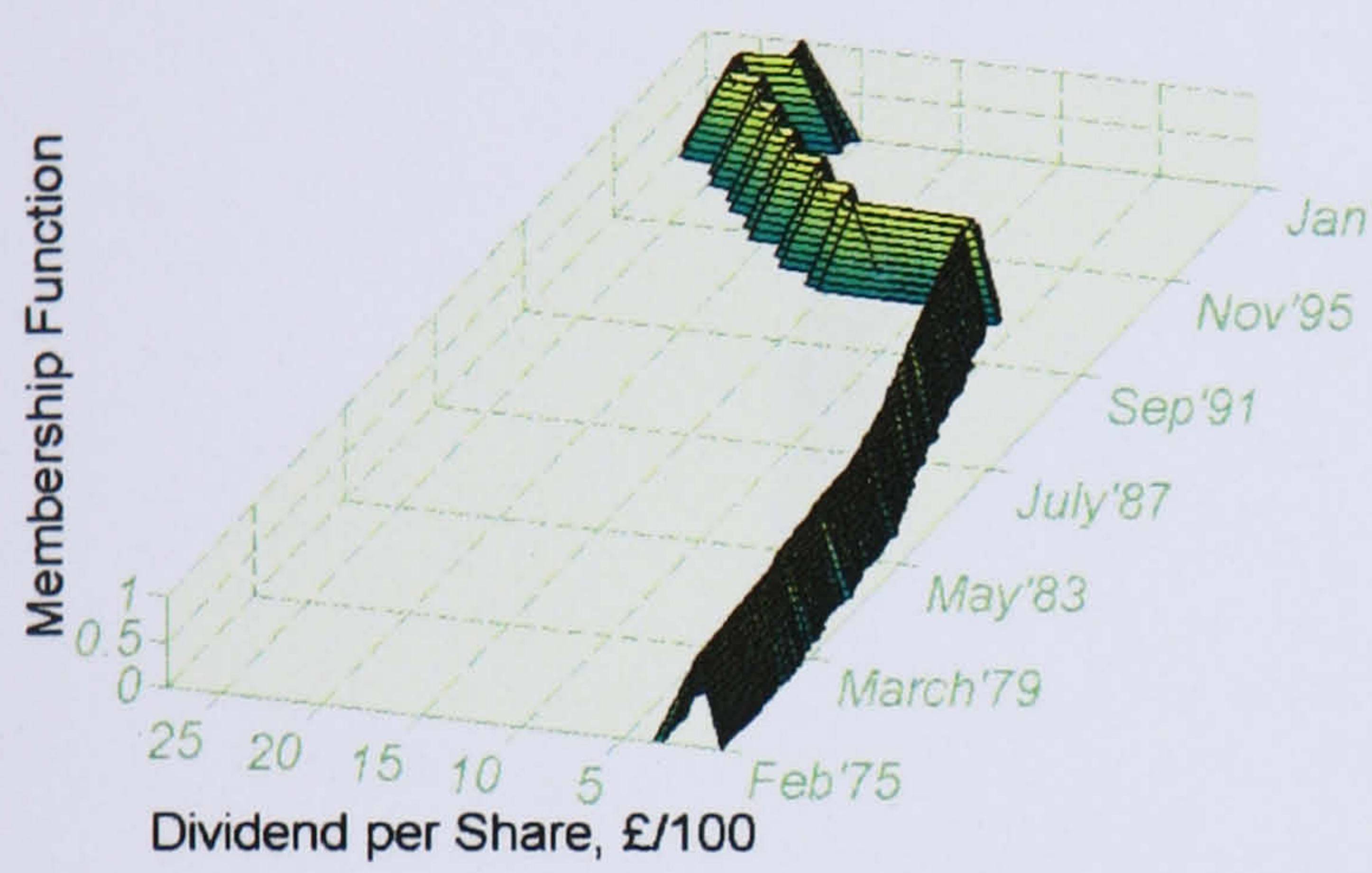


Figure A1.13a: GREAT UNIVERSAL STORES - fuzzified data

Figure A1.13b: GREAT UNIVERSAL STORES - evaluated fuzzy share price

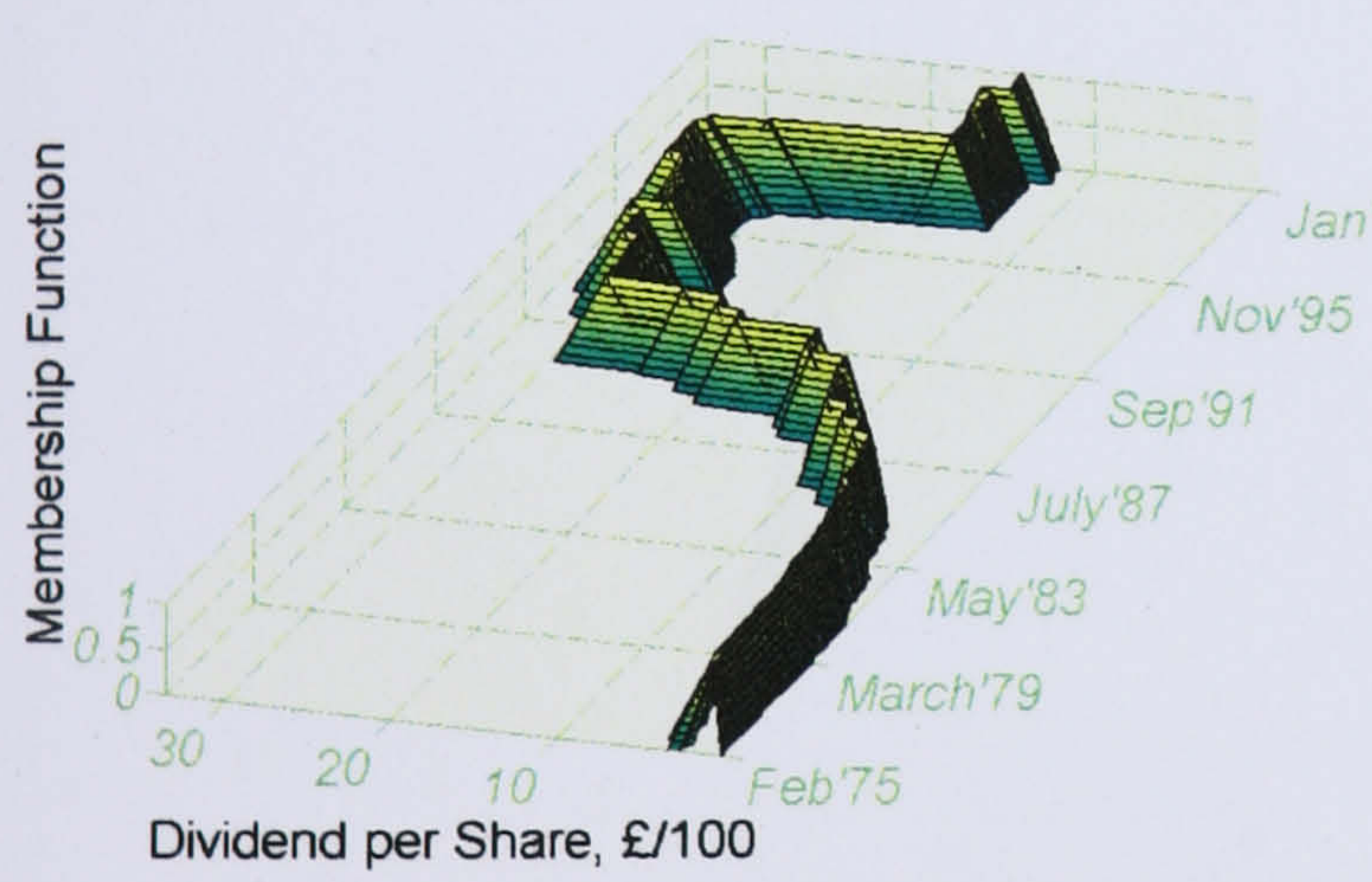


Figure A1.14a: HANSON - fuzzified data

Figure A1.14b: HANSON - evaluated fuzzy share price

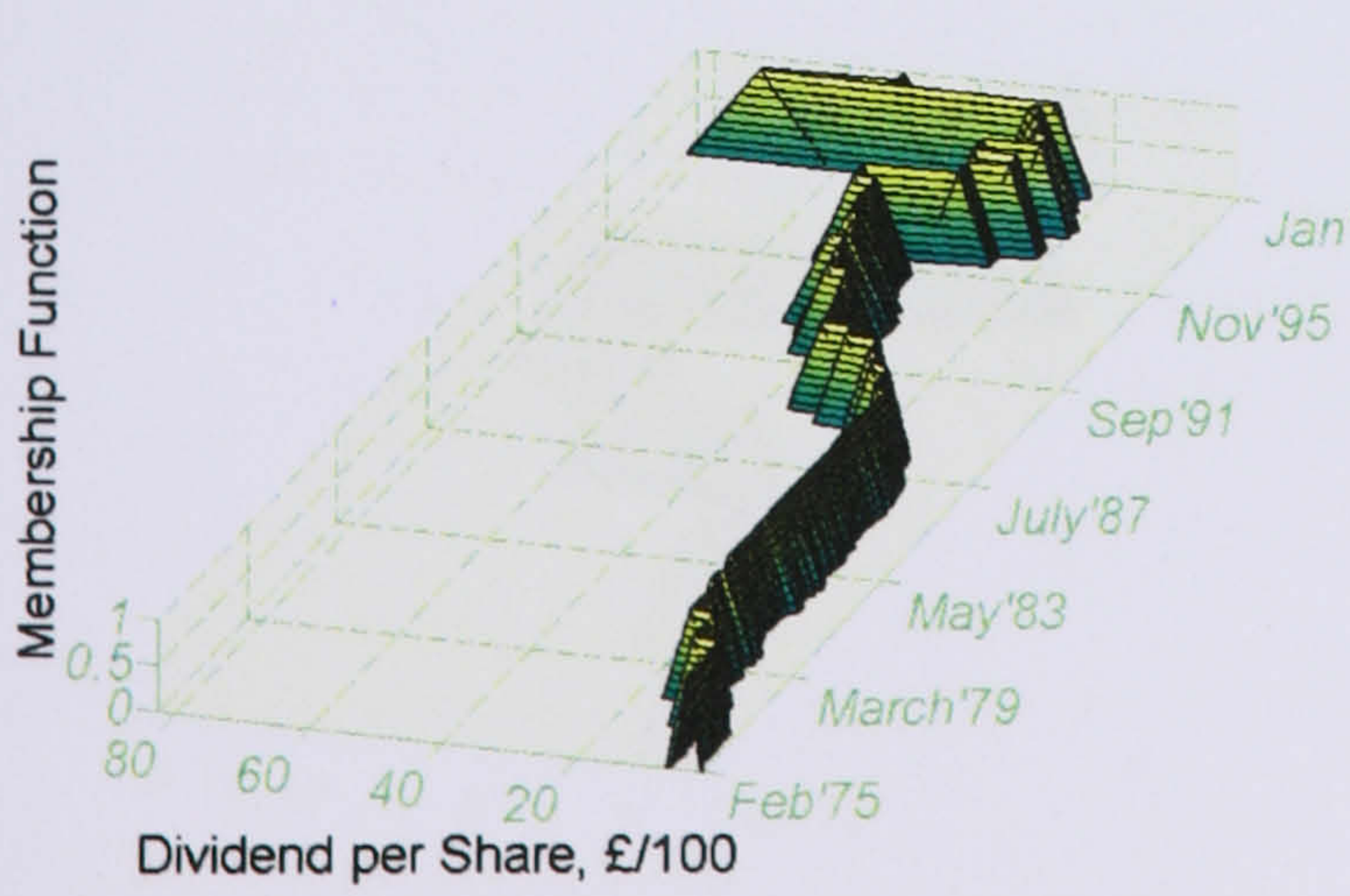


Figure A1.15a: INCHCAPE - fuzzified data

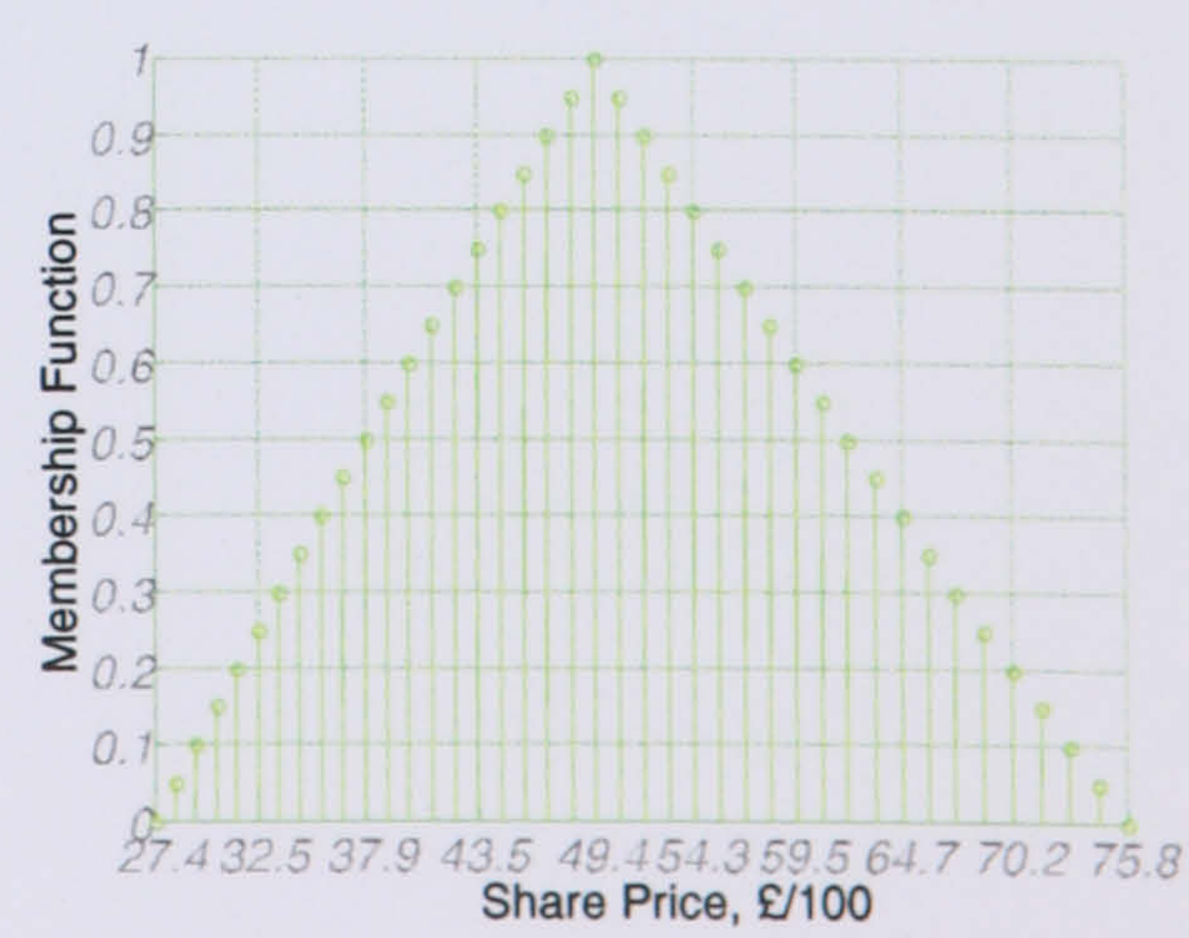


Figure A1.15b: INCHCAPE - evaluated fuzzy share price

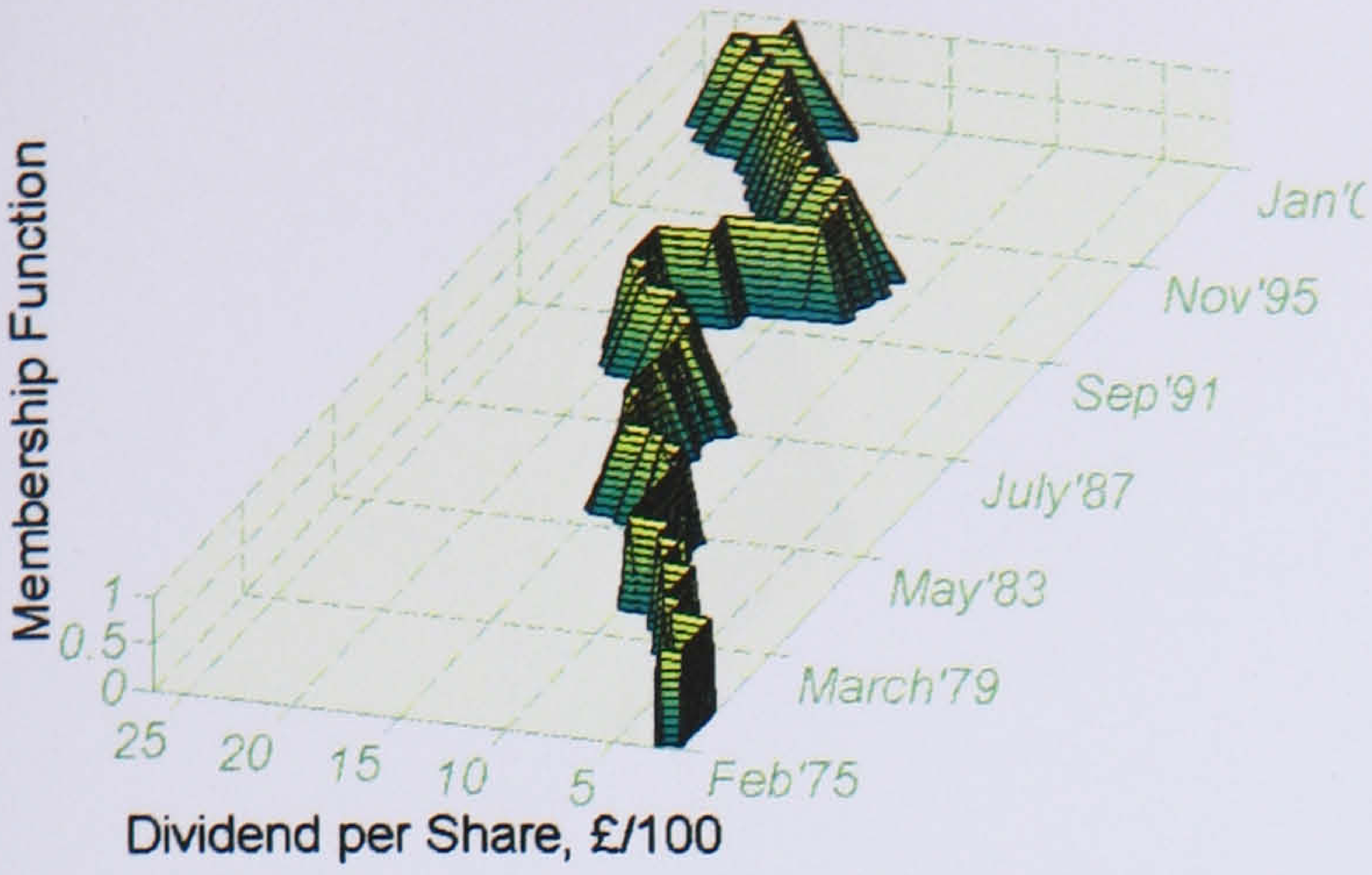


Figure A1.16a: LEX SERVICE - fuzzified data

Figure A1.16b: LEX SERVICE - evaluated fuzzy share price

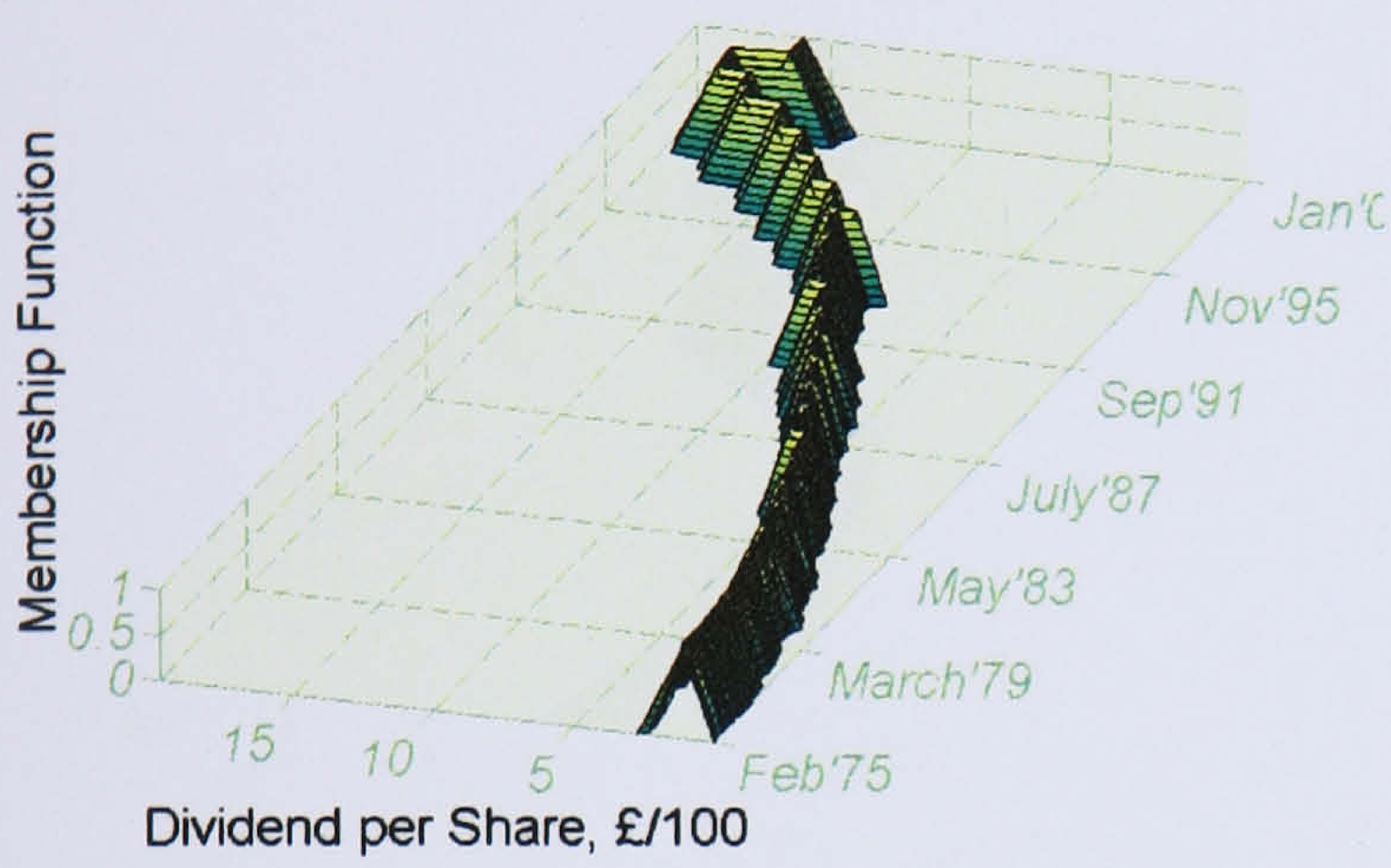


Figure A1.17a: MARKS & SPENCER - fuzzified data

Figure A1.17b: MARKS & SPENCER - evaluated fuzzy share price

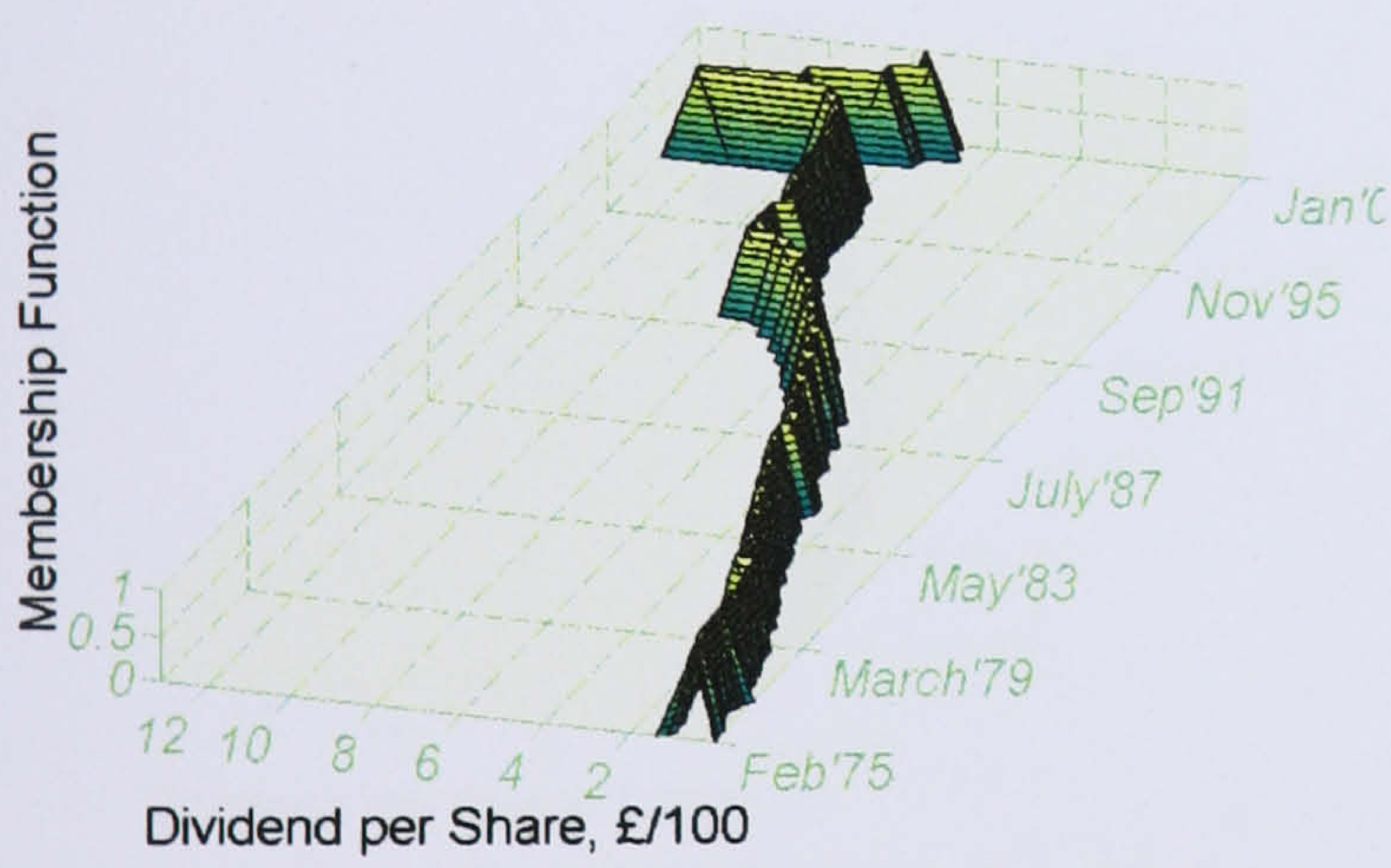


Figure A1.18a: NORTHERN FOODS - fuzzified data

Figure A1.18b: NORTHERN FOODS - evaluated fuzzy share price

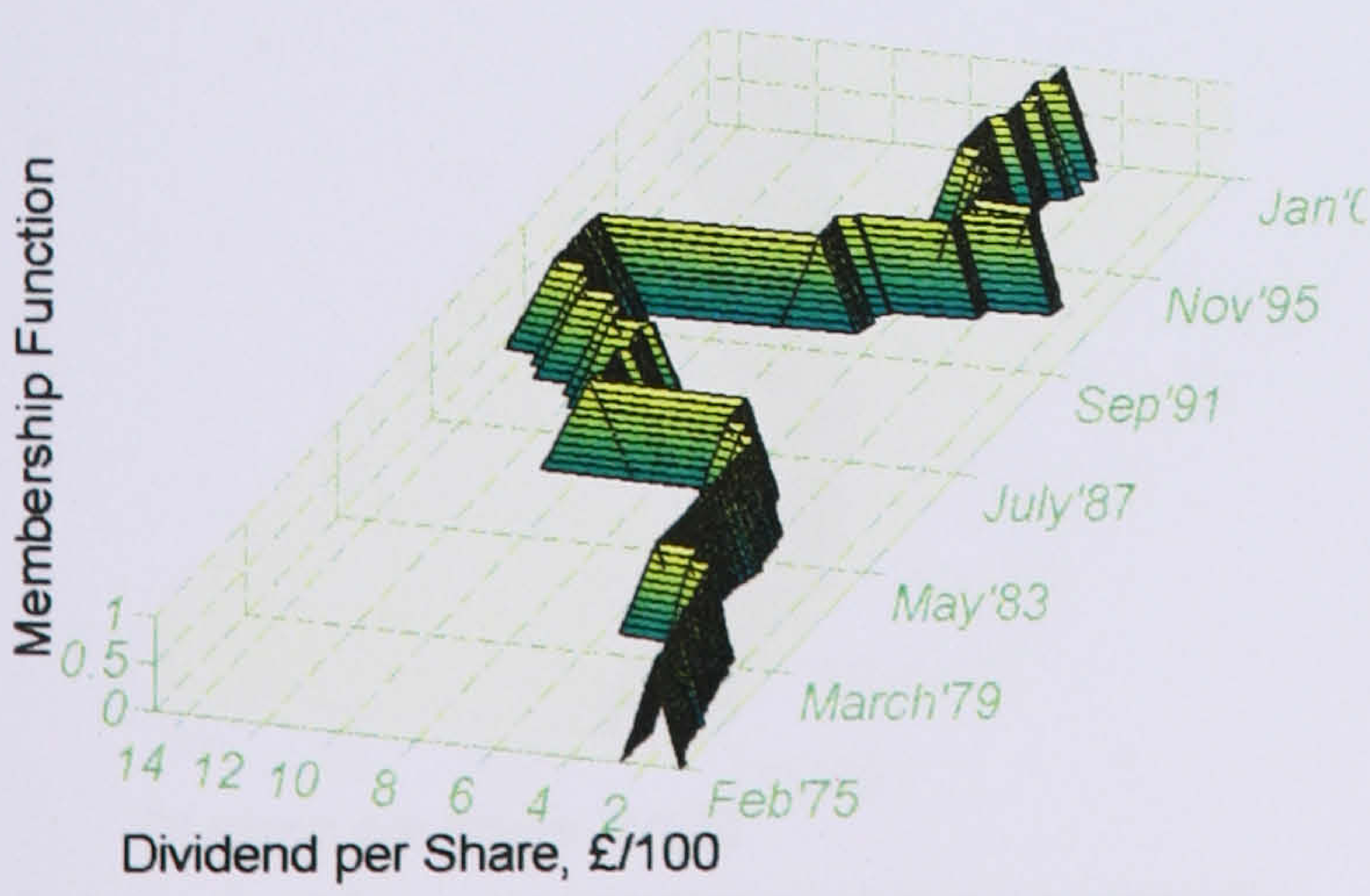


Figure A1.19a: PILKINGTON - fuzzified data

Figure A1.19b: PILKINGTON - evaluated fuzzy share price

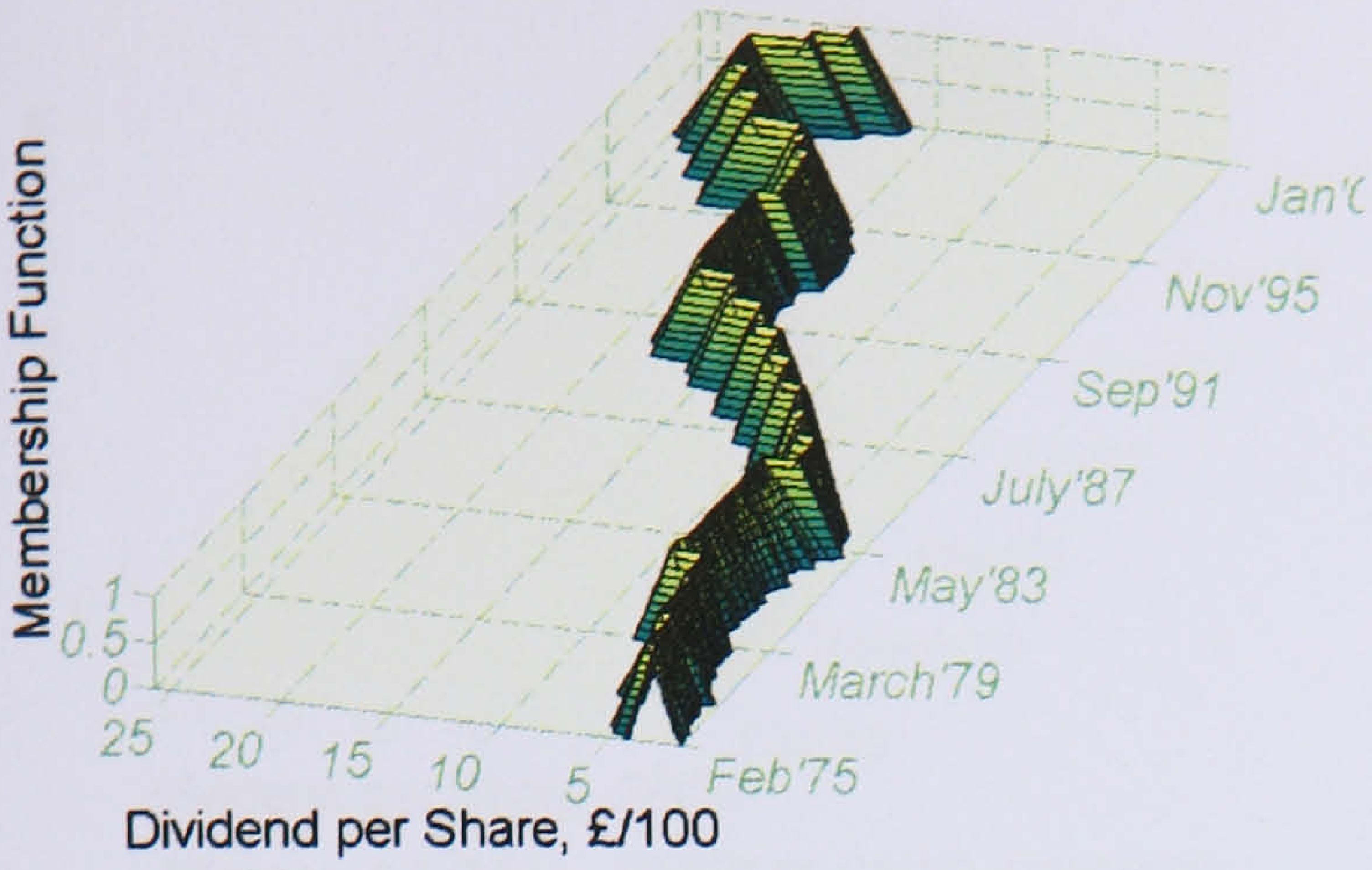


Figure A1.20a: RANK GROUP - fuzzified data

Figure A1.20b: RANK GROUP - evaluated fuzzy share price

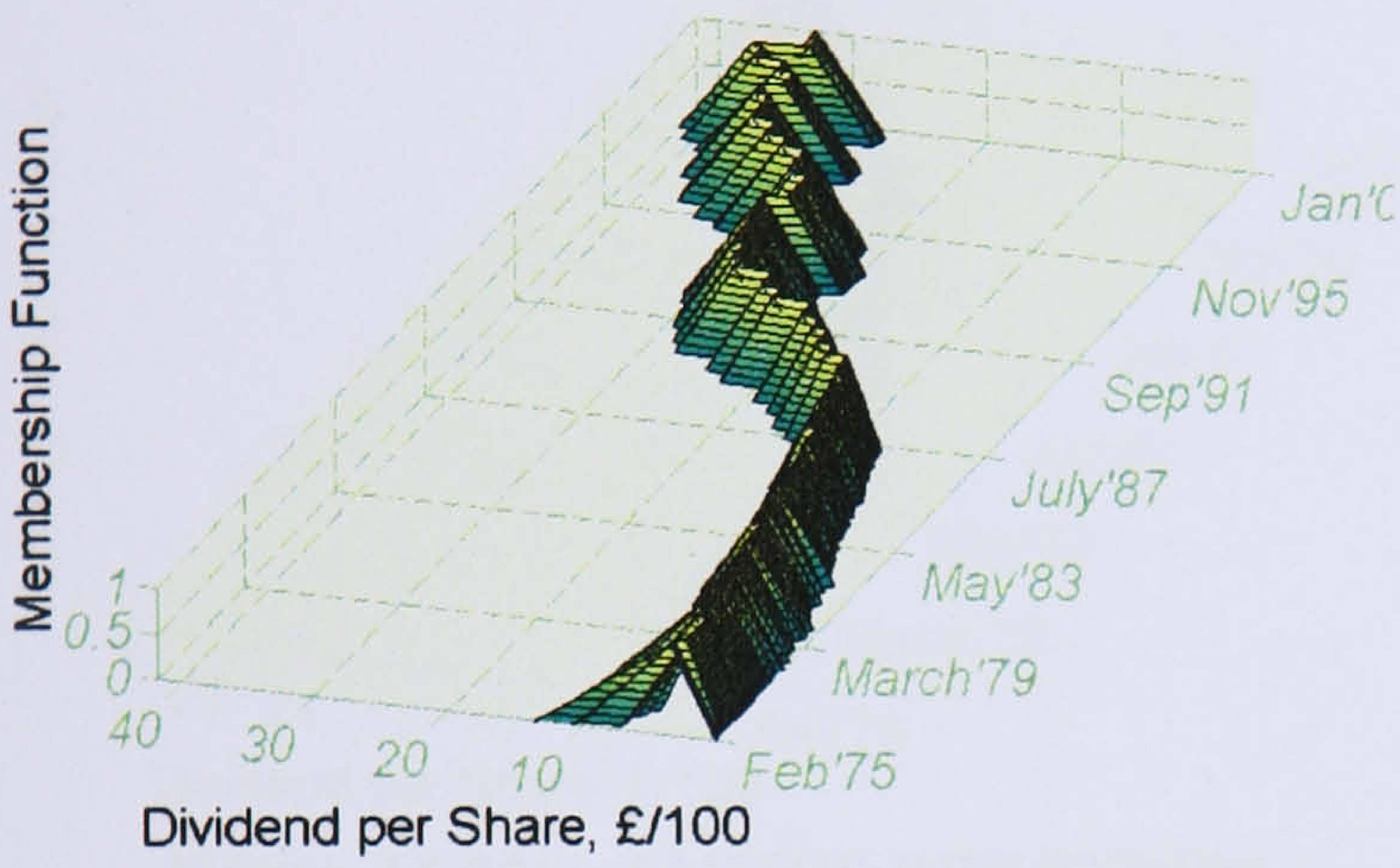


Figure A1.21a: RMC GROUP - fuzzified data

Figure A1.21b: RMC GROUP - evaluated fuzzy share price

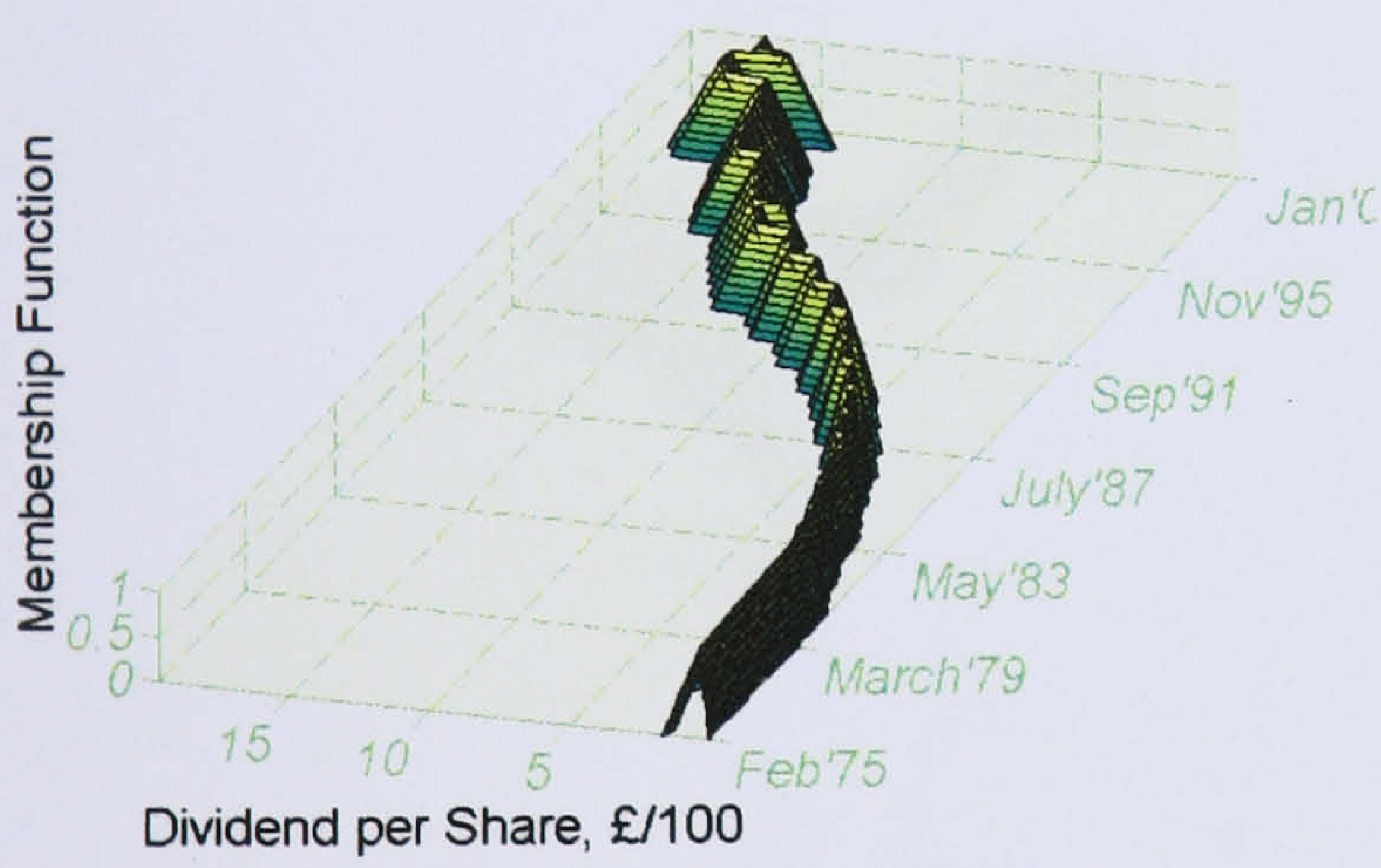


Figure A1.22a: SAINSBURY (J) - fuzzified data

Figure A1.22b: SAINSBURY (J) - evaluated fuzzy share price

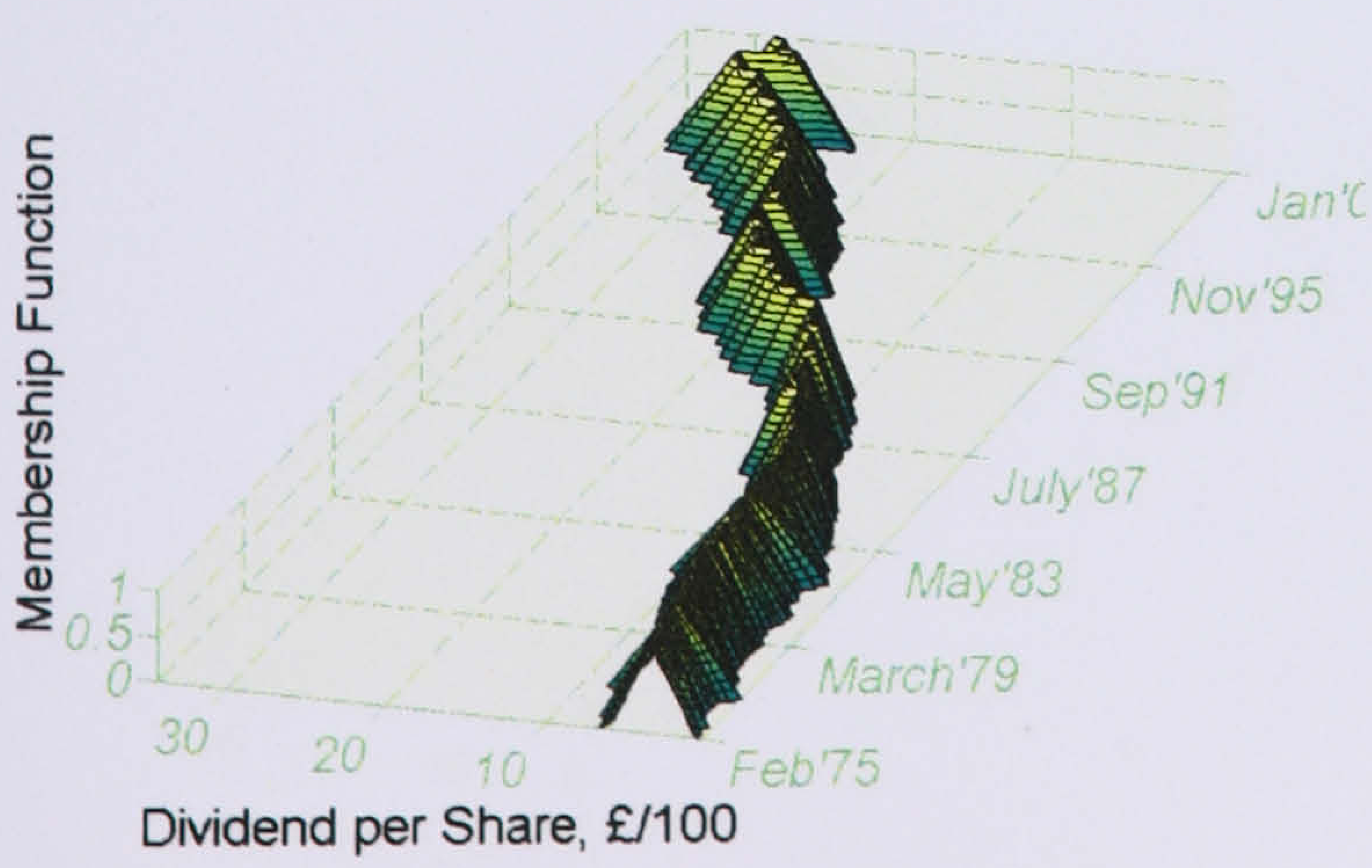


Figure A1.23a: SCOTTISH & NEWCASTLE - fuzzified data

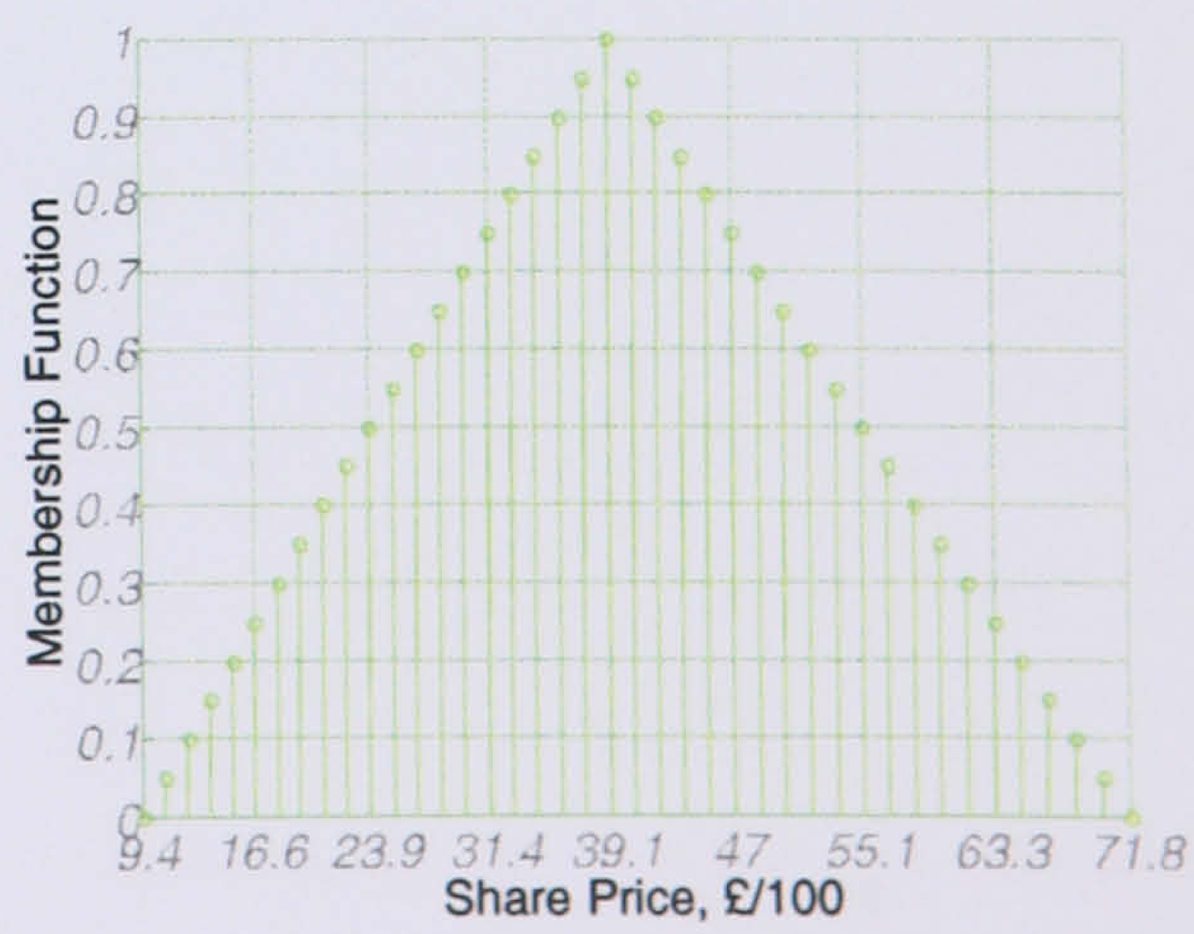


Figure A1.23b: SCOTTISH & NEWCASTLE - evaluated fuzzy share price

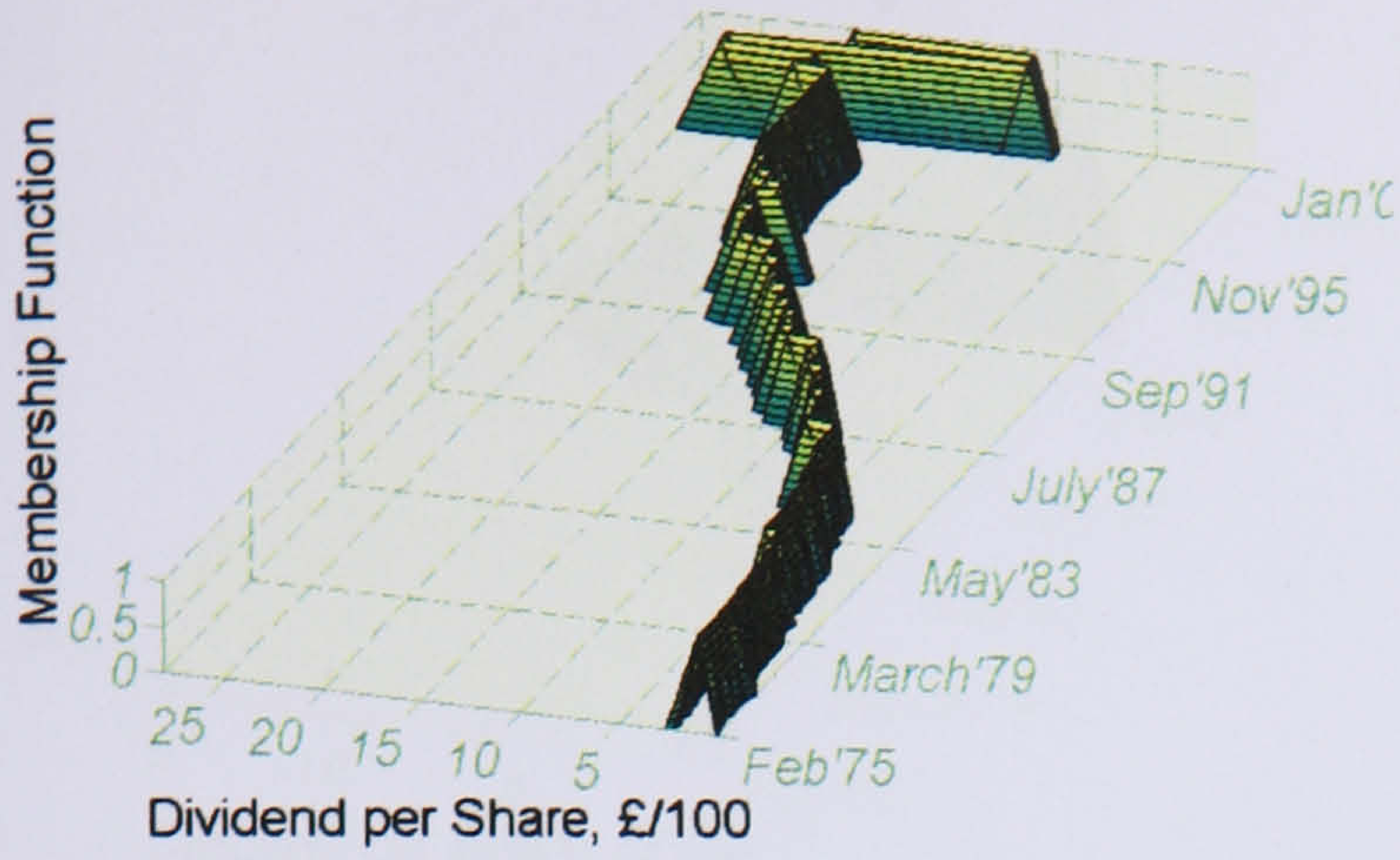


Figure A1.24a: SMITH (WH) GROUP - fuzzified data

Figure A1.24b: SMITH (WH) GROUP - evaluated fuzzy share price

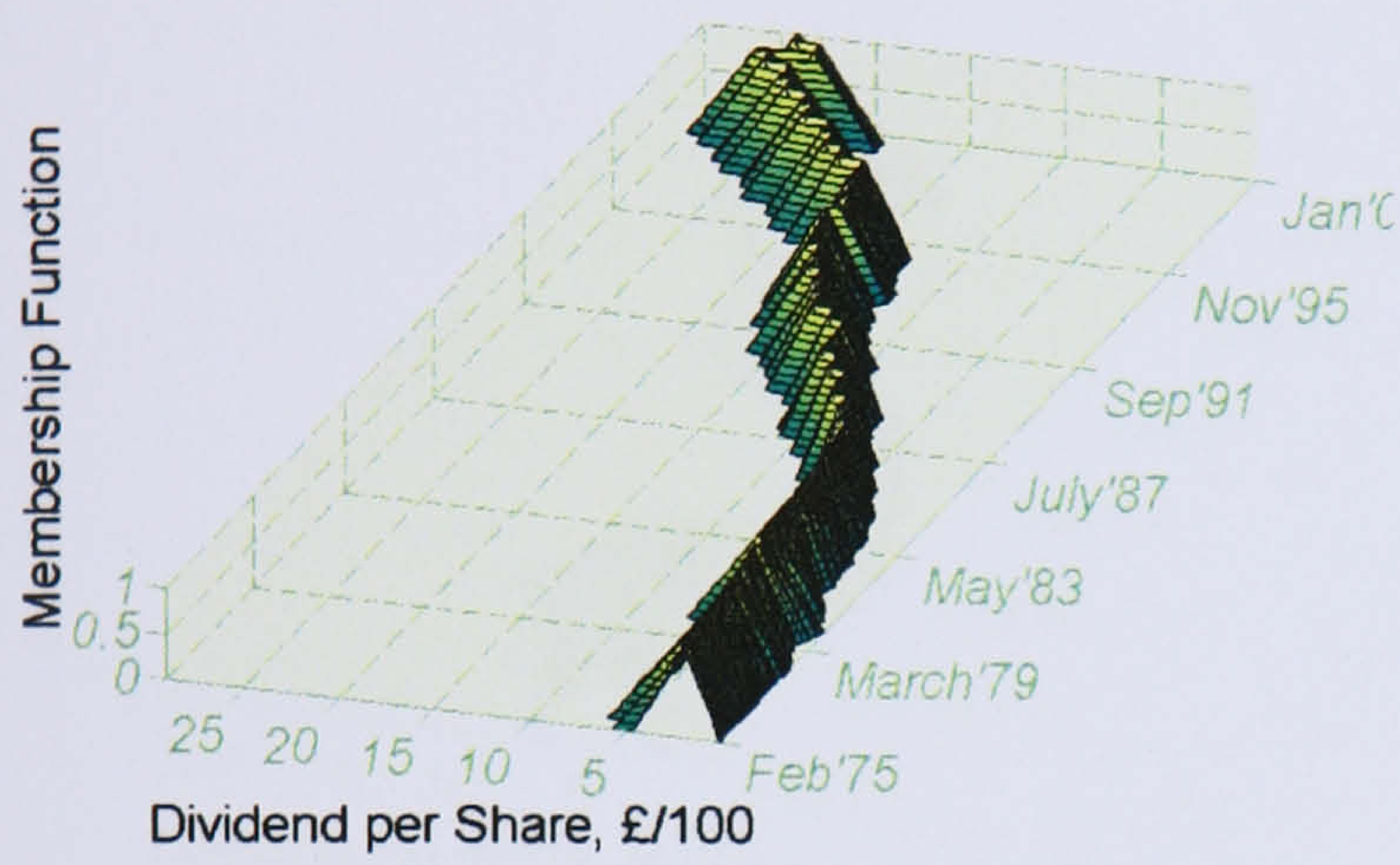


Figure A1.25a: SMITHS INDUSTRIES - fuzzified data

Figure A1.25b: SMITHS INDUSTRIES - evaluated fuzzy share price

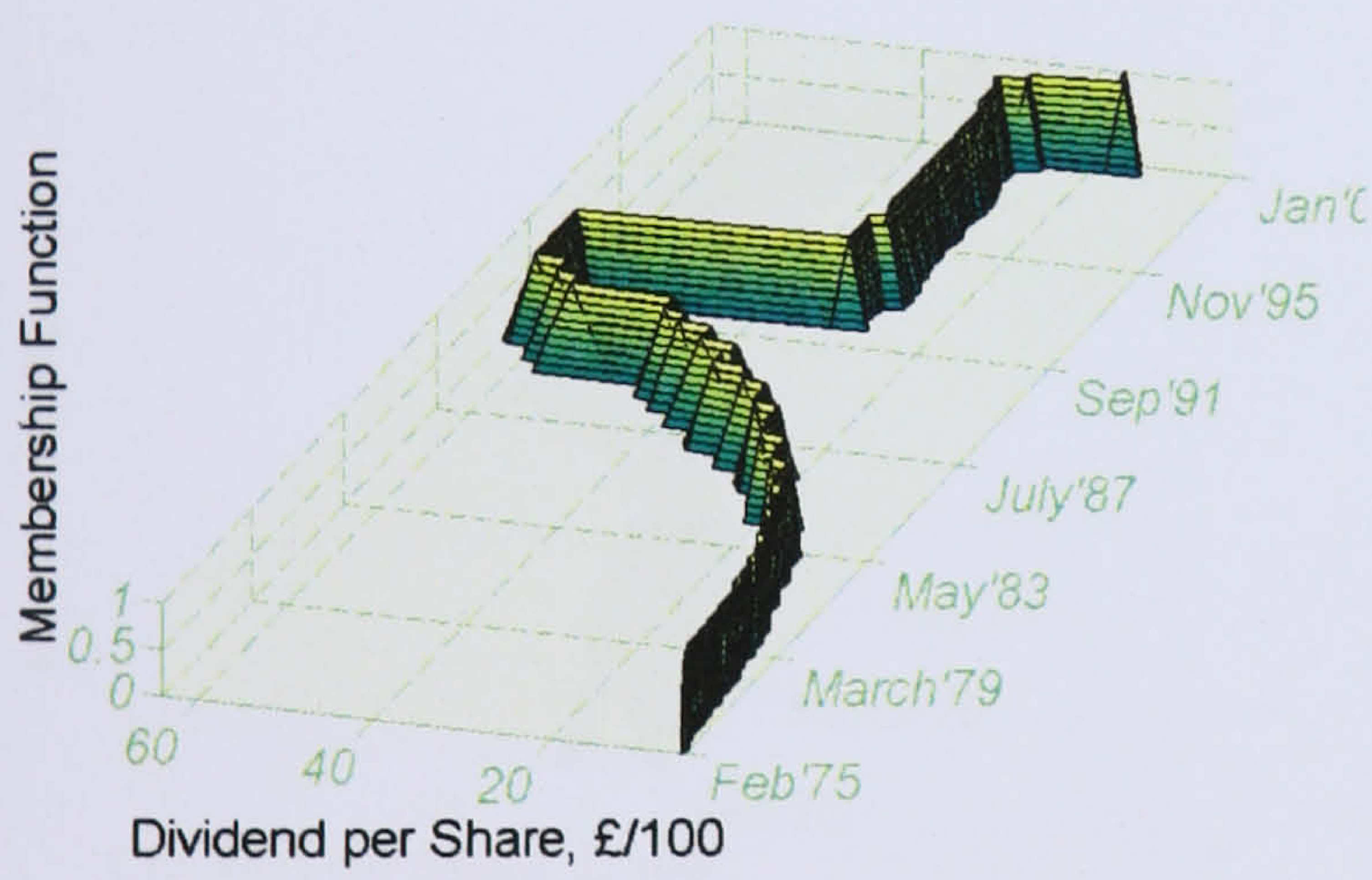


Figure A1.26a: TARMAC - fuzzified data

Figure A1.26b: TARMAC - evaluated fuzzy share price

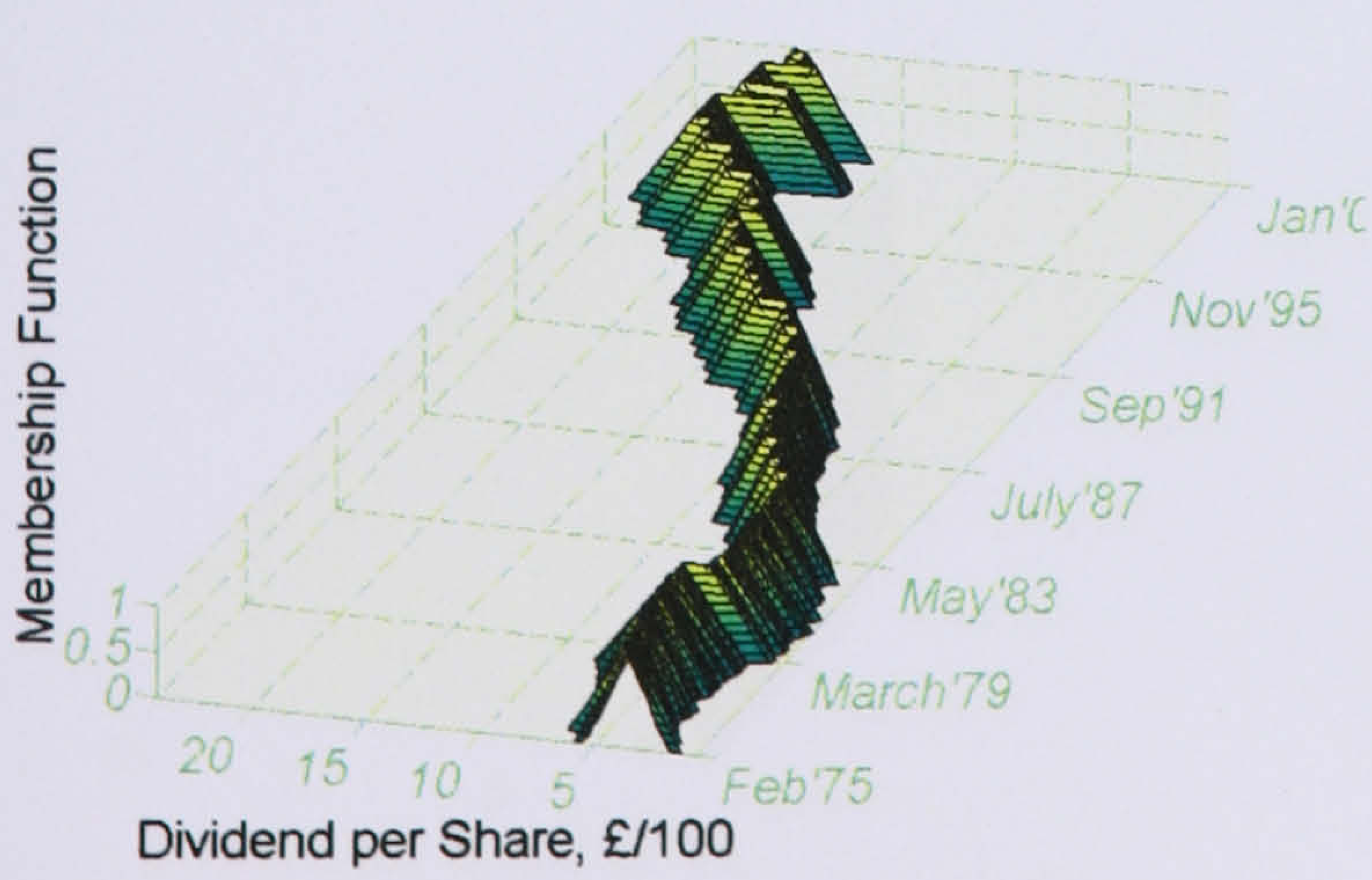


Figure A1.27a: TATE &LYLE - fuzzified data

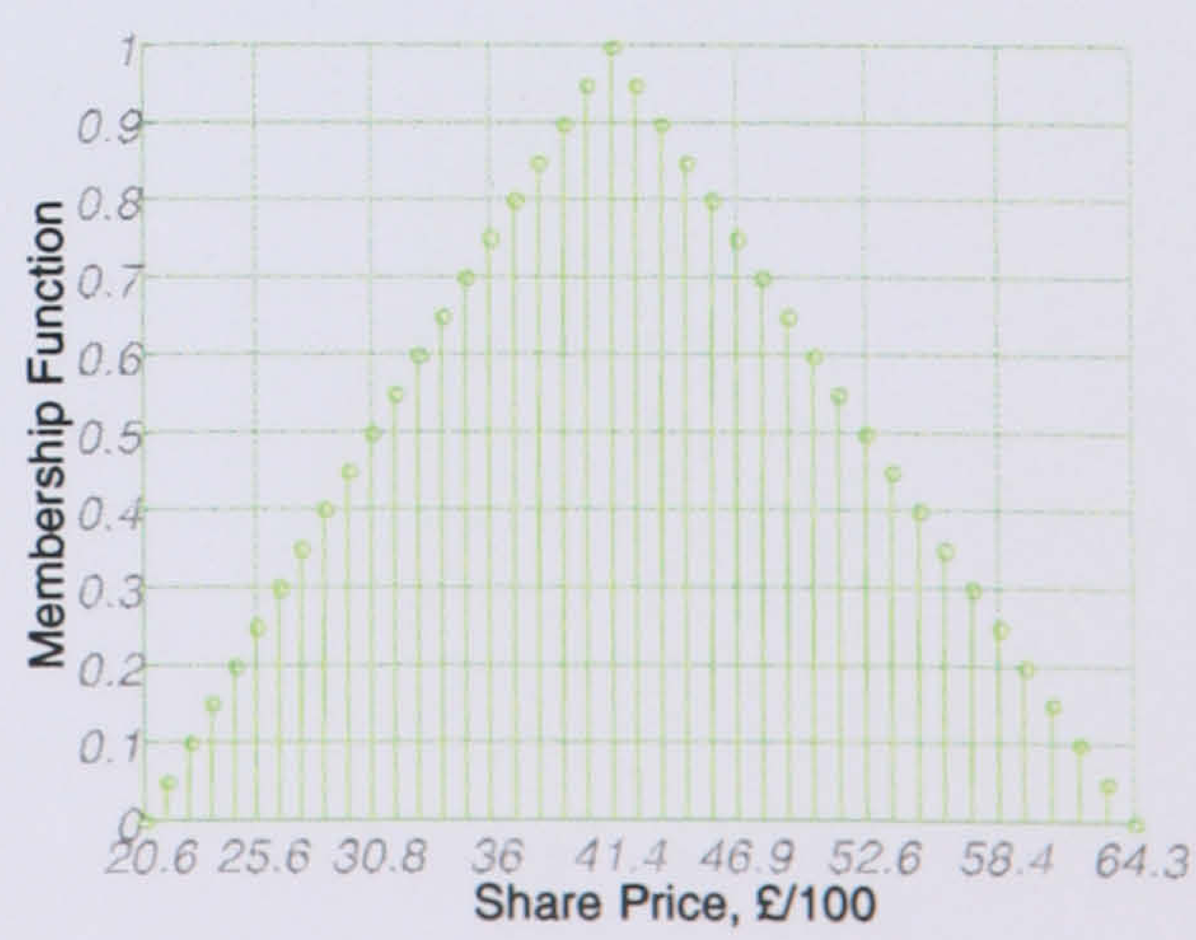


Figure A1.27b: TATE &LYLE - evaluated fuzzy share price

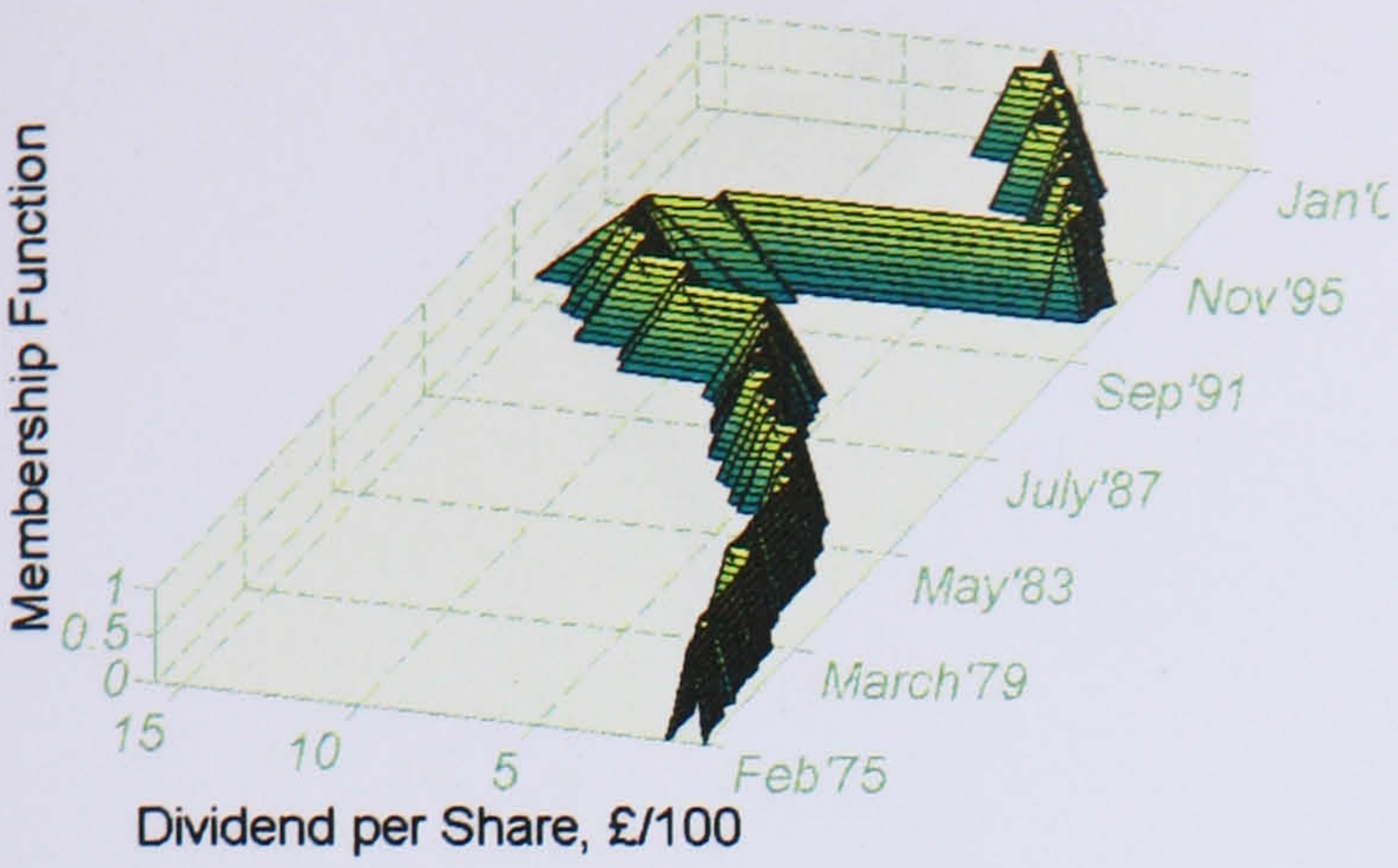


Figure A1.28a: TAYLOR WOODROW - fuzzified data

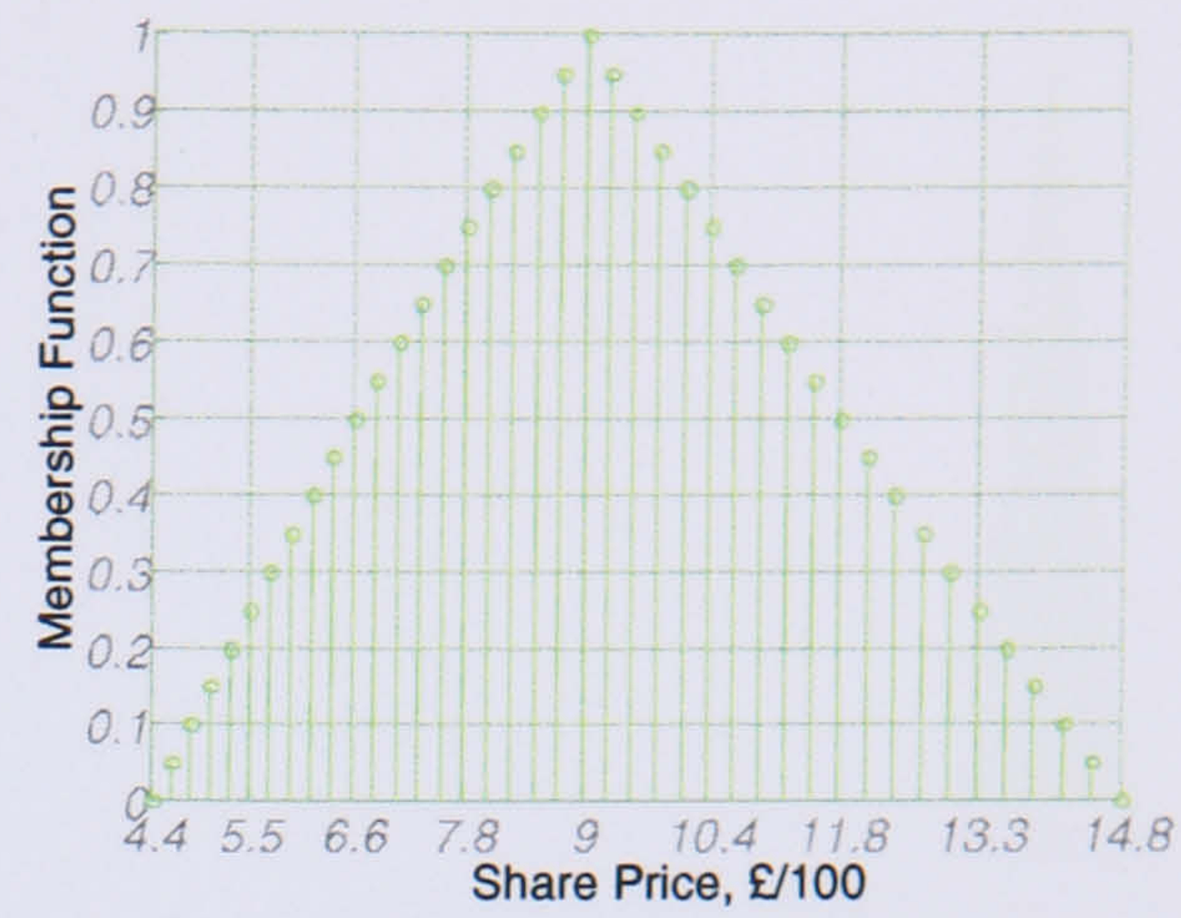


Figure A1.28b: TAYLOR WOODROW - evaluated fuzzy share price

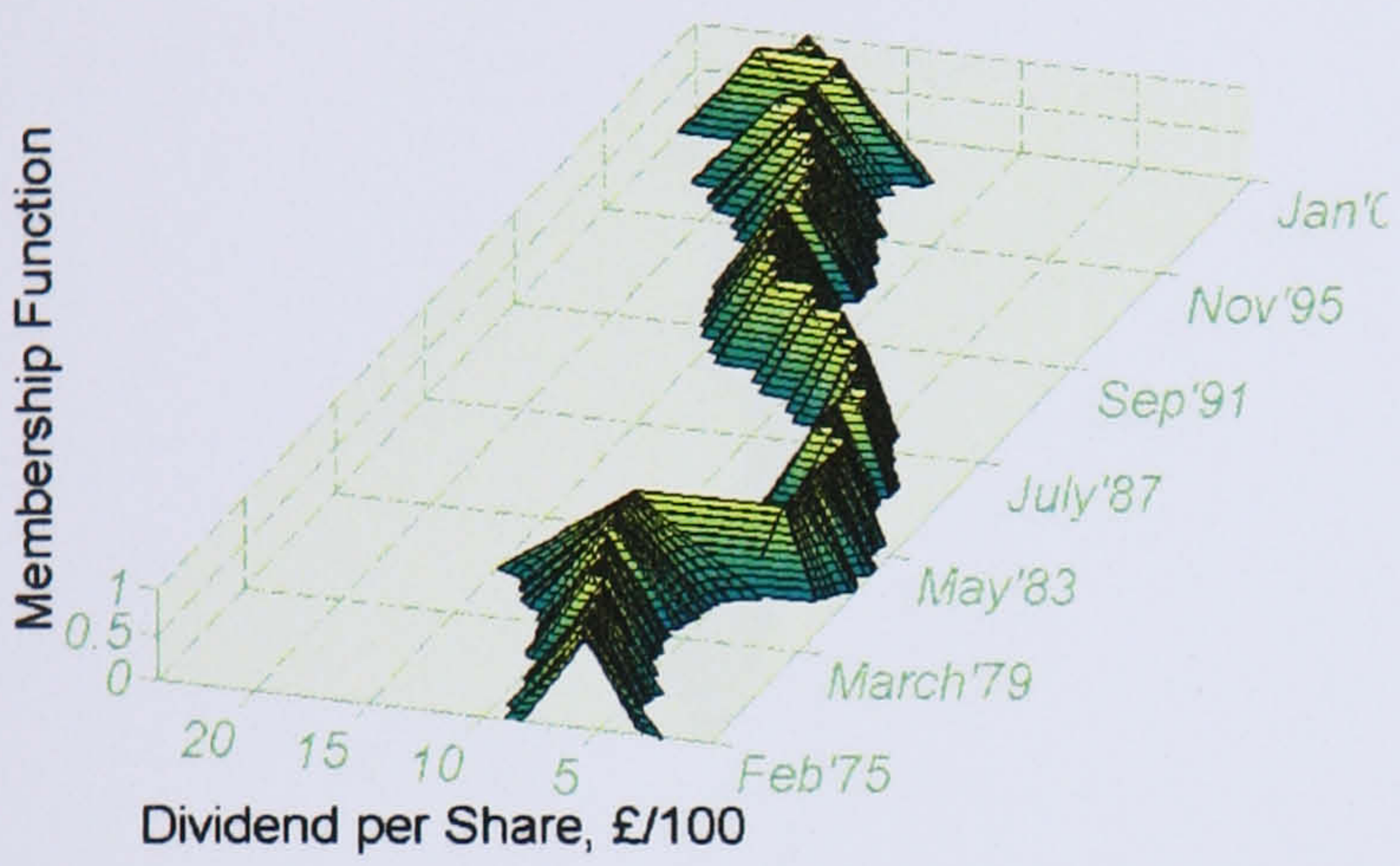


Figure A1.29a: TI GROUP - fuzzified data

Figure A1.29b: TI GROUP - evaluated fuzzy share price

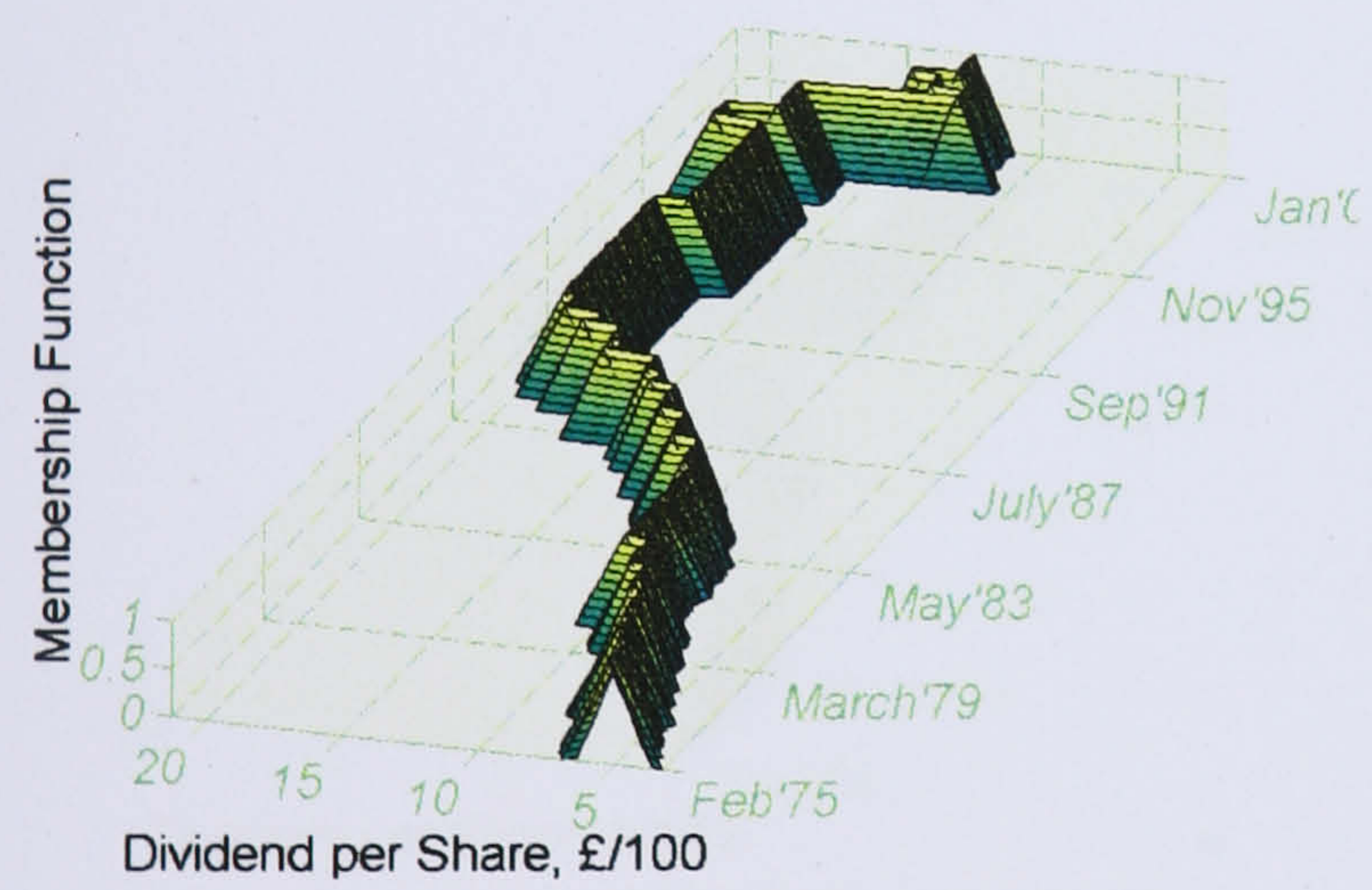


Figure A1.30a: TRANSPORT DEVELOPMENT GROUP - fuzzified data

Figure A1.30b: TRANSPORT DEVELOPMENT GRO evaluated fuzzy share price

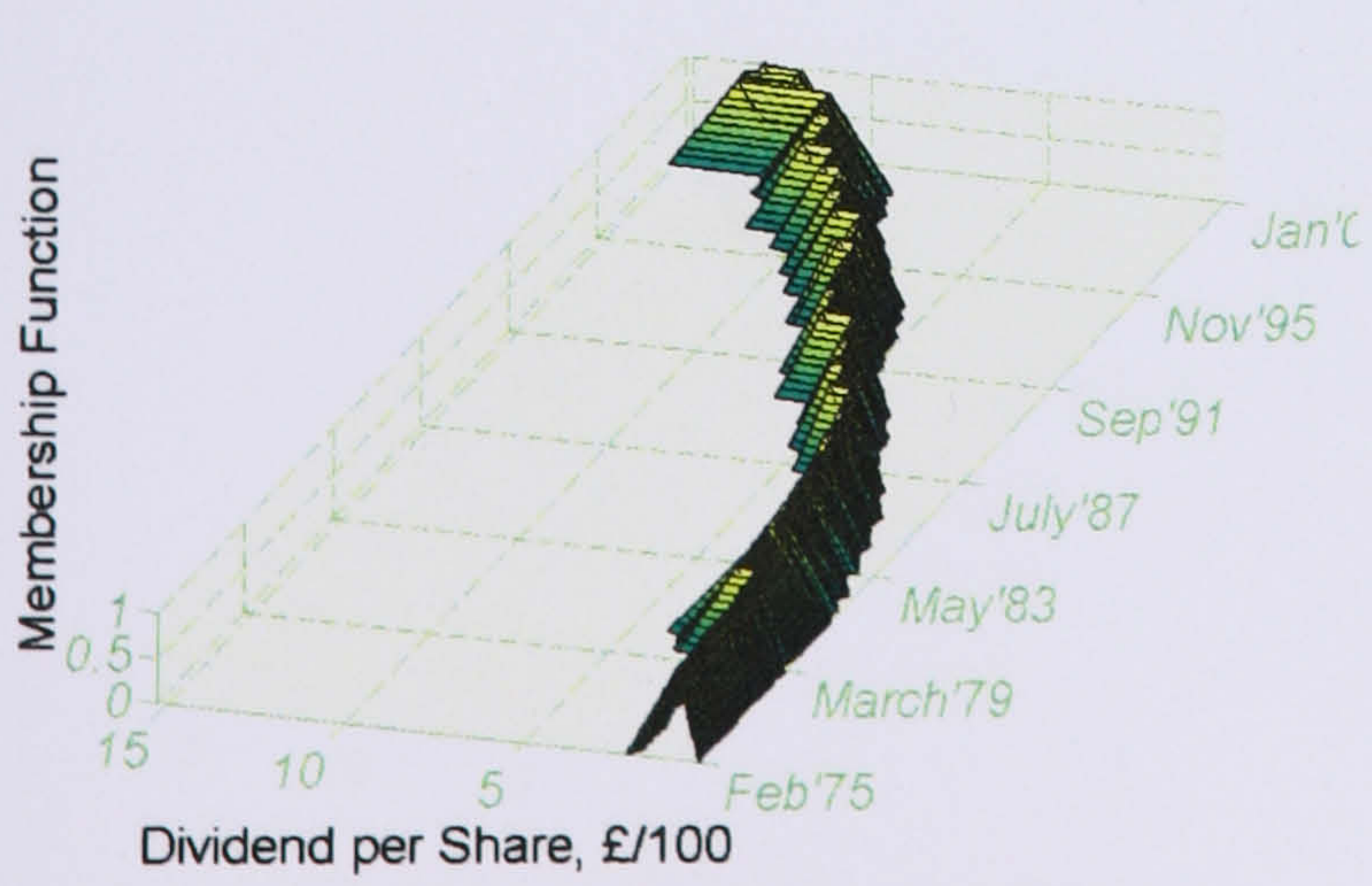


Figure A1.31a: UNILEVER - fuzzified data

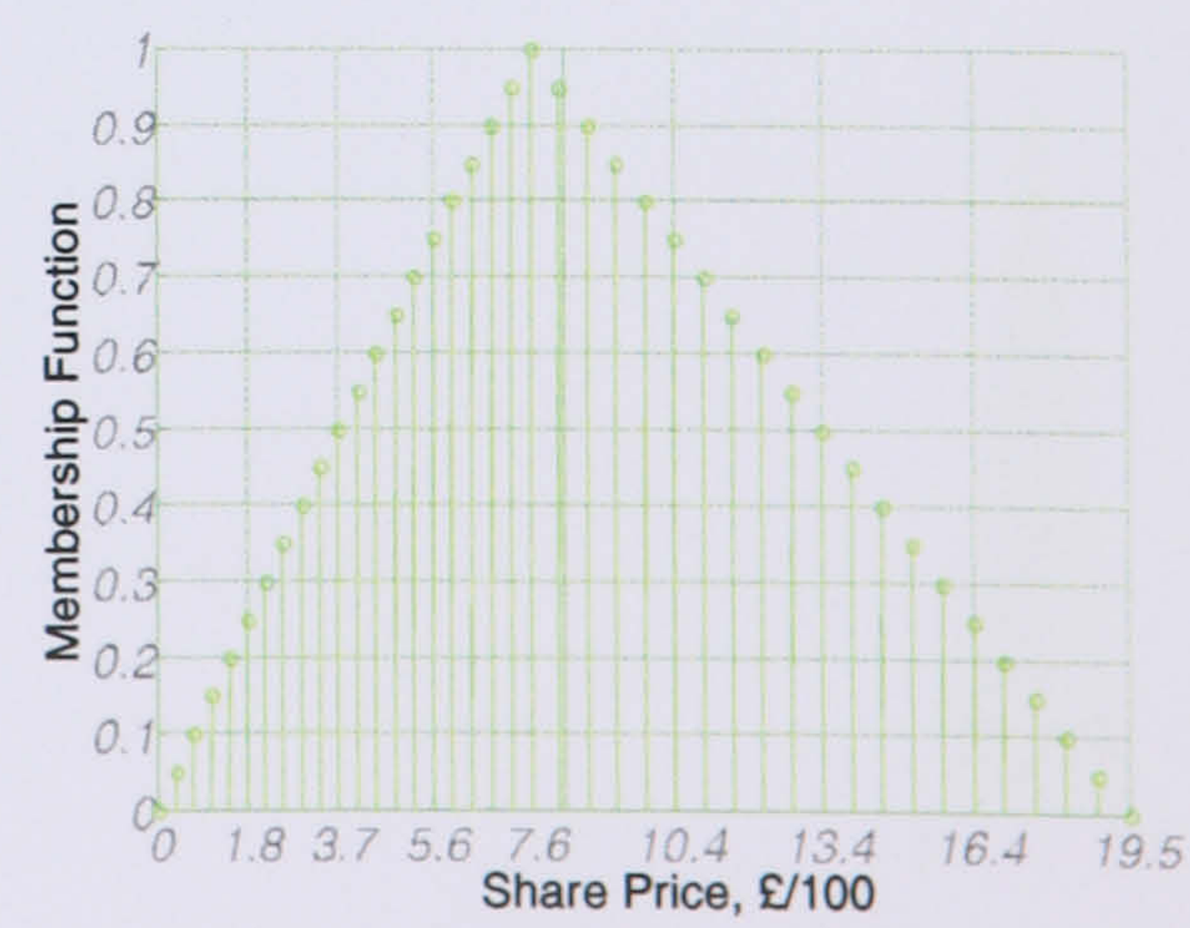


Figure A1.31b: UNILEVER - evaluated fuzzy share price

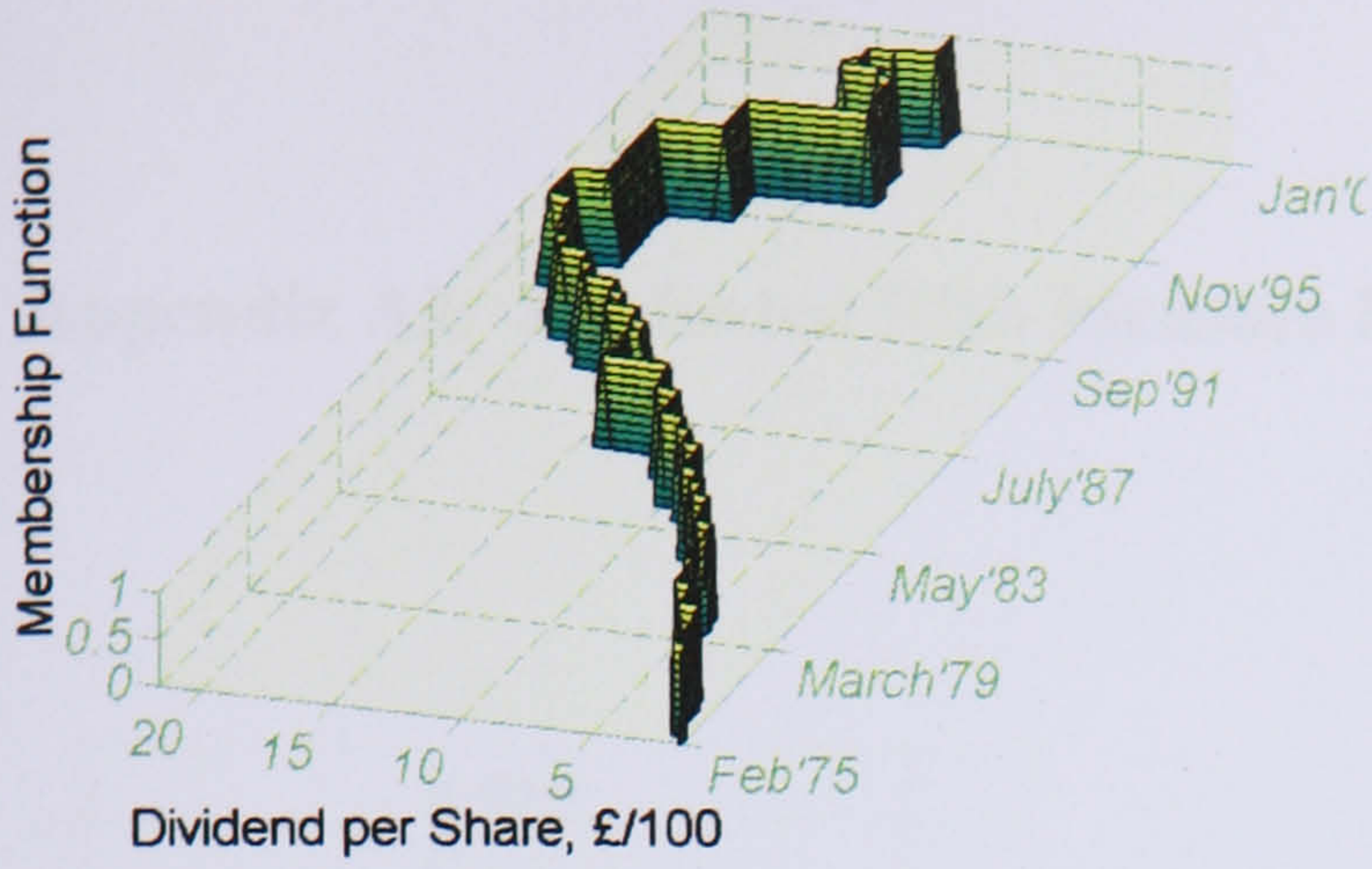


Figure A1.32a: UNITED BISCUITS HOLDINGS - fuzzified data

Figure A1.32b: UNITED BISCUITS HOLDINGS - evaluated fuzzy share price

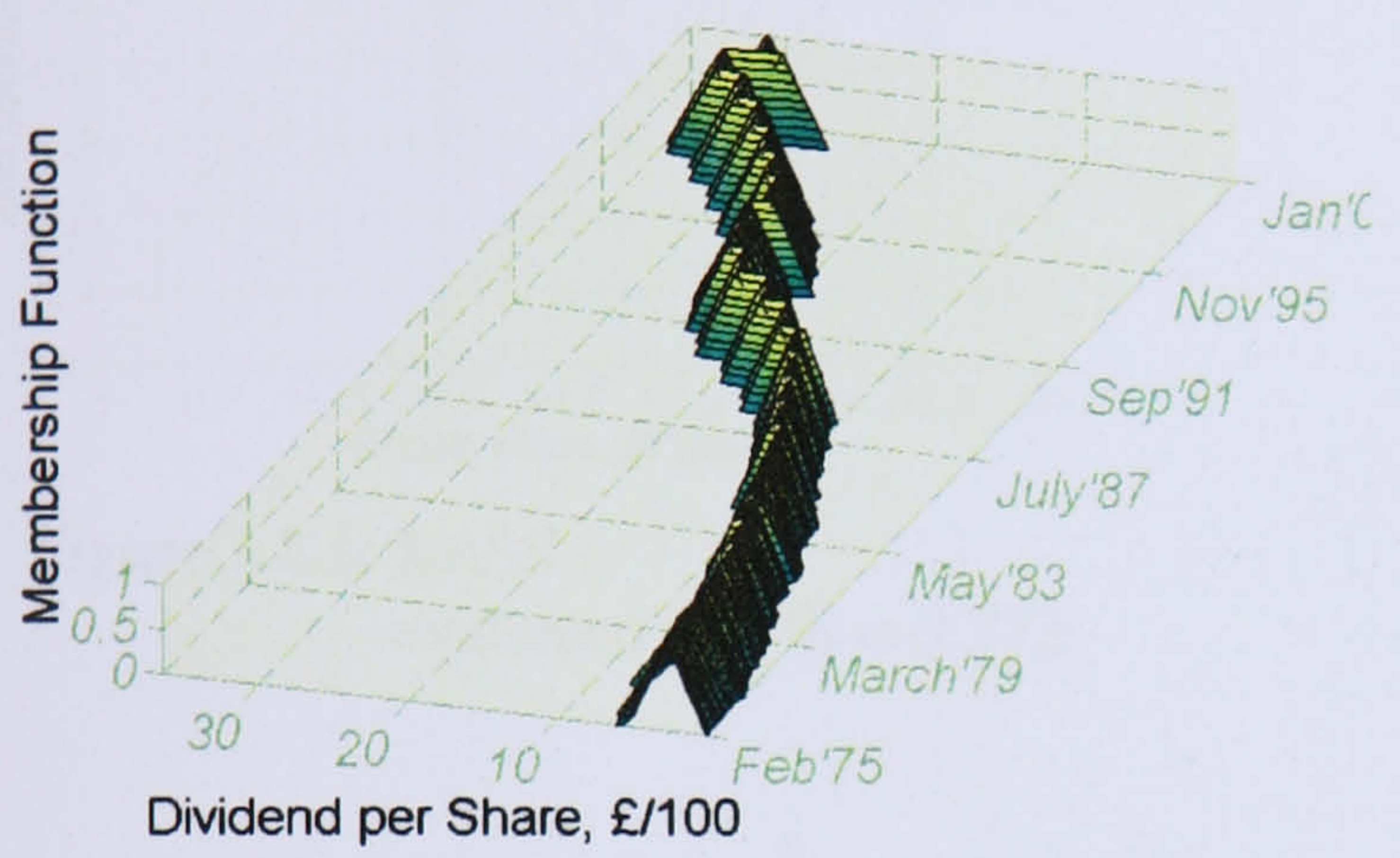


Figure A1.33a: WHITBREAD - fuzzified data

Figure A1.33b: WHITBREAD - evaluated fuzzy share price

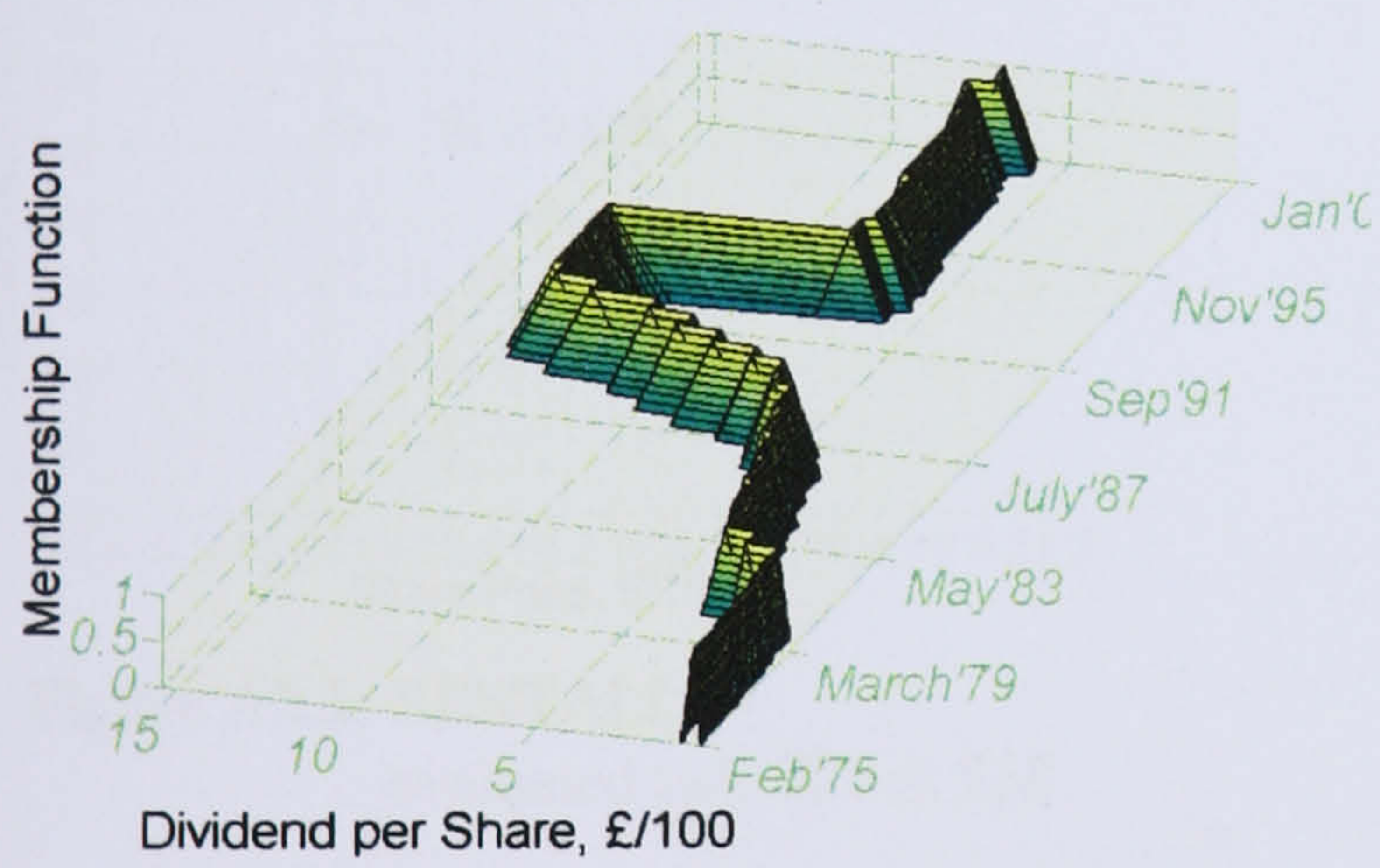


Figure A1.34a: WIMPEY (GEORGE) - fuzzified data

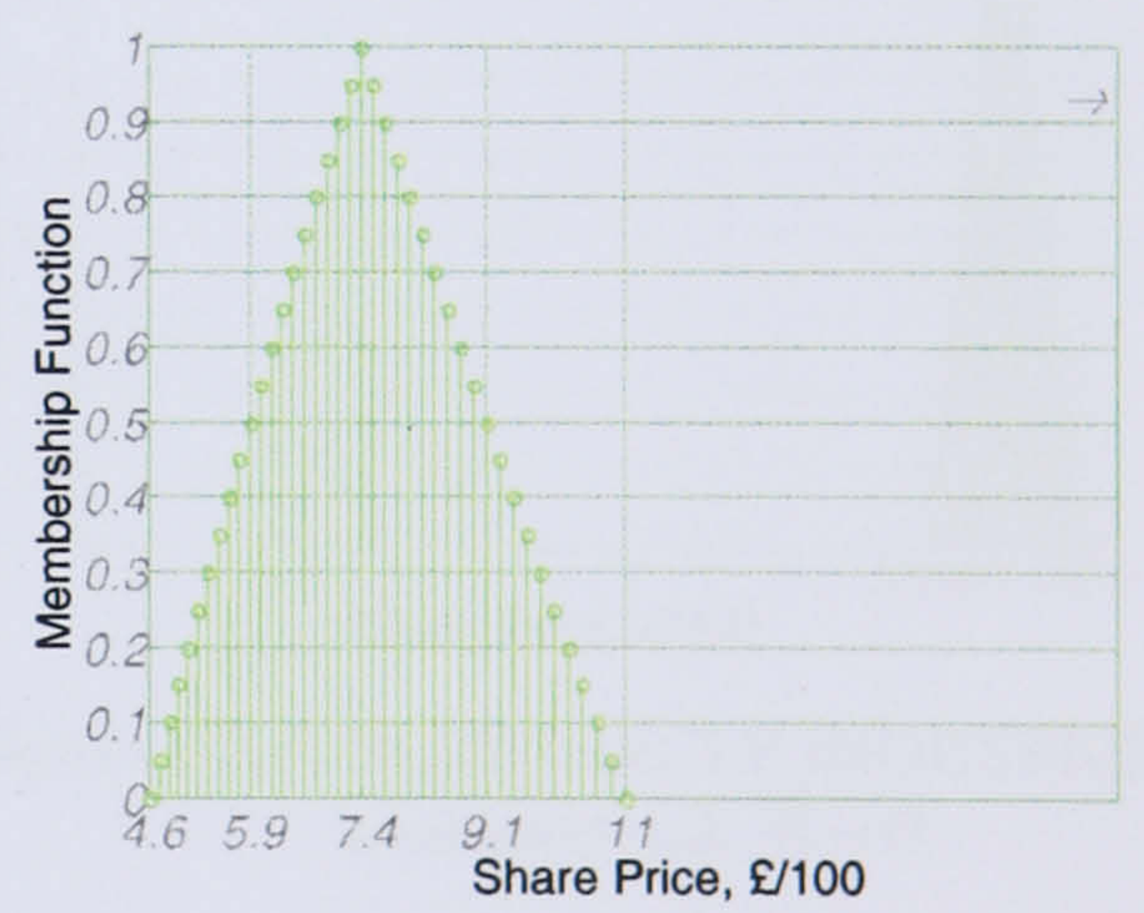


Figure A1.34b: WIMPEY (GEORGE) - evaluated fuzzy share price

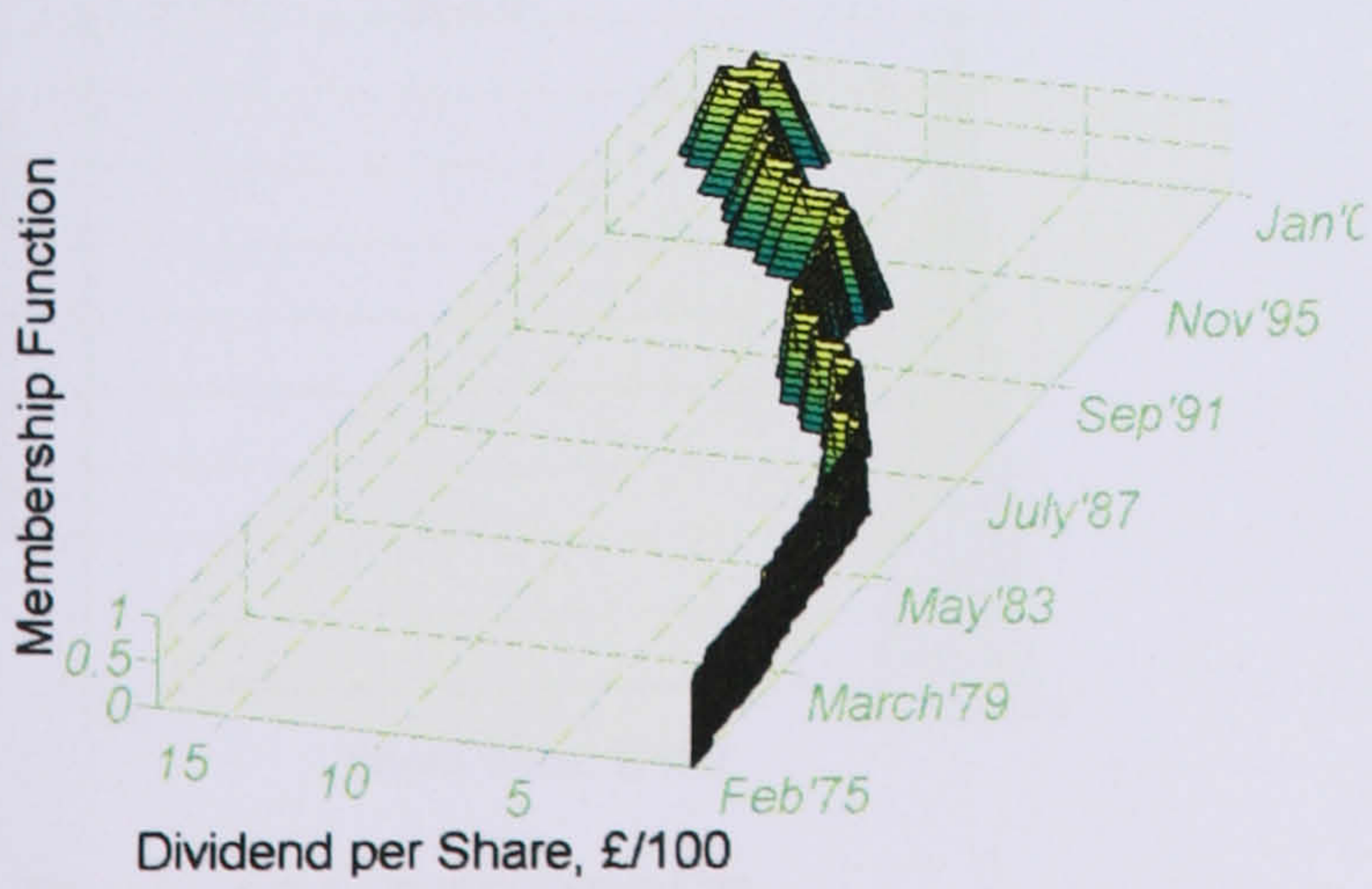


Figure A1.35a: WOLSELEY - fuzzified data

Figure A1.35b: WOLSELEY - evaluated fuzzy share price

Appendix A2: Evaluated Risk Measure by Company

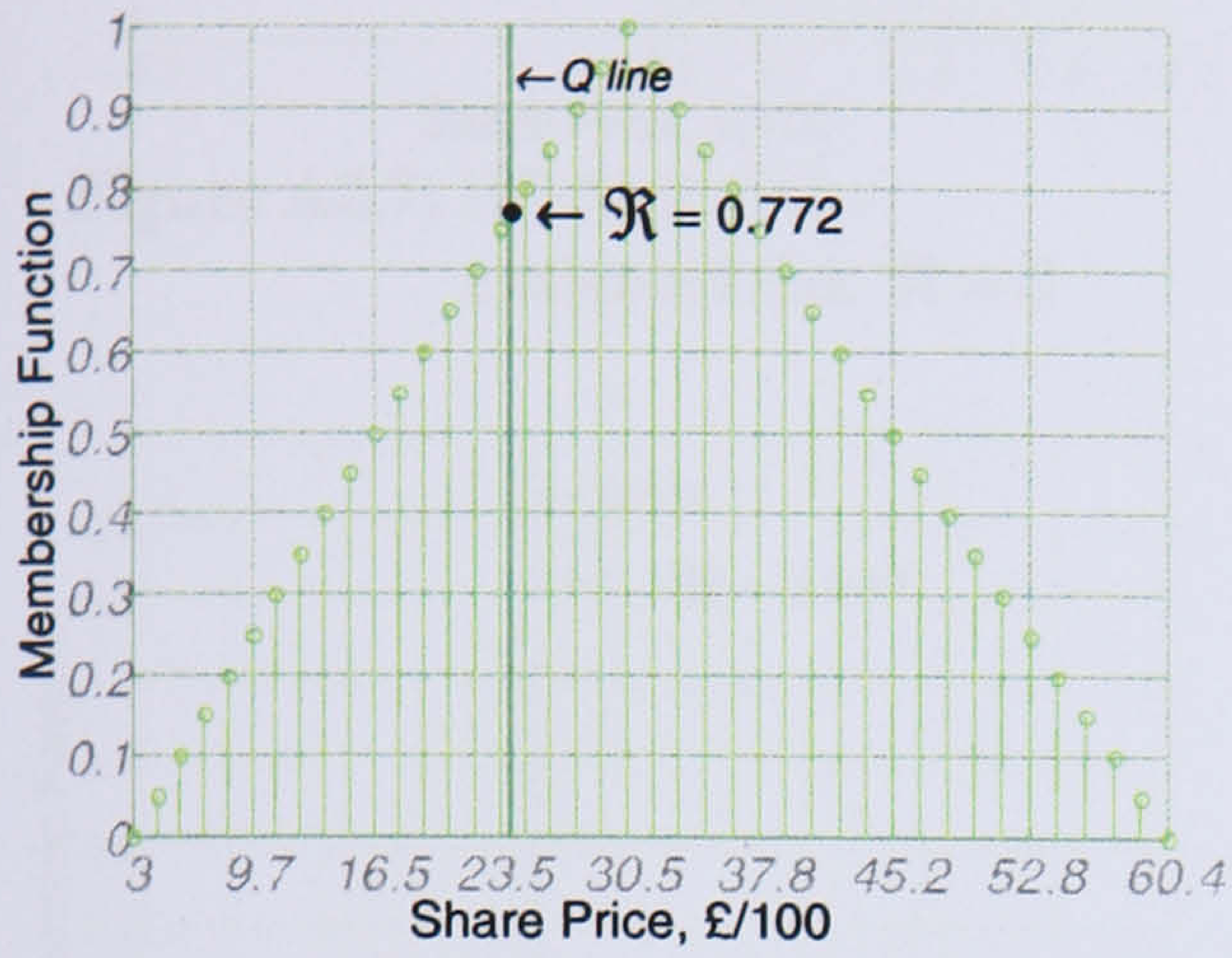


Figure A2.1: BASS - evaluated risk $\Re = 0.772$

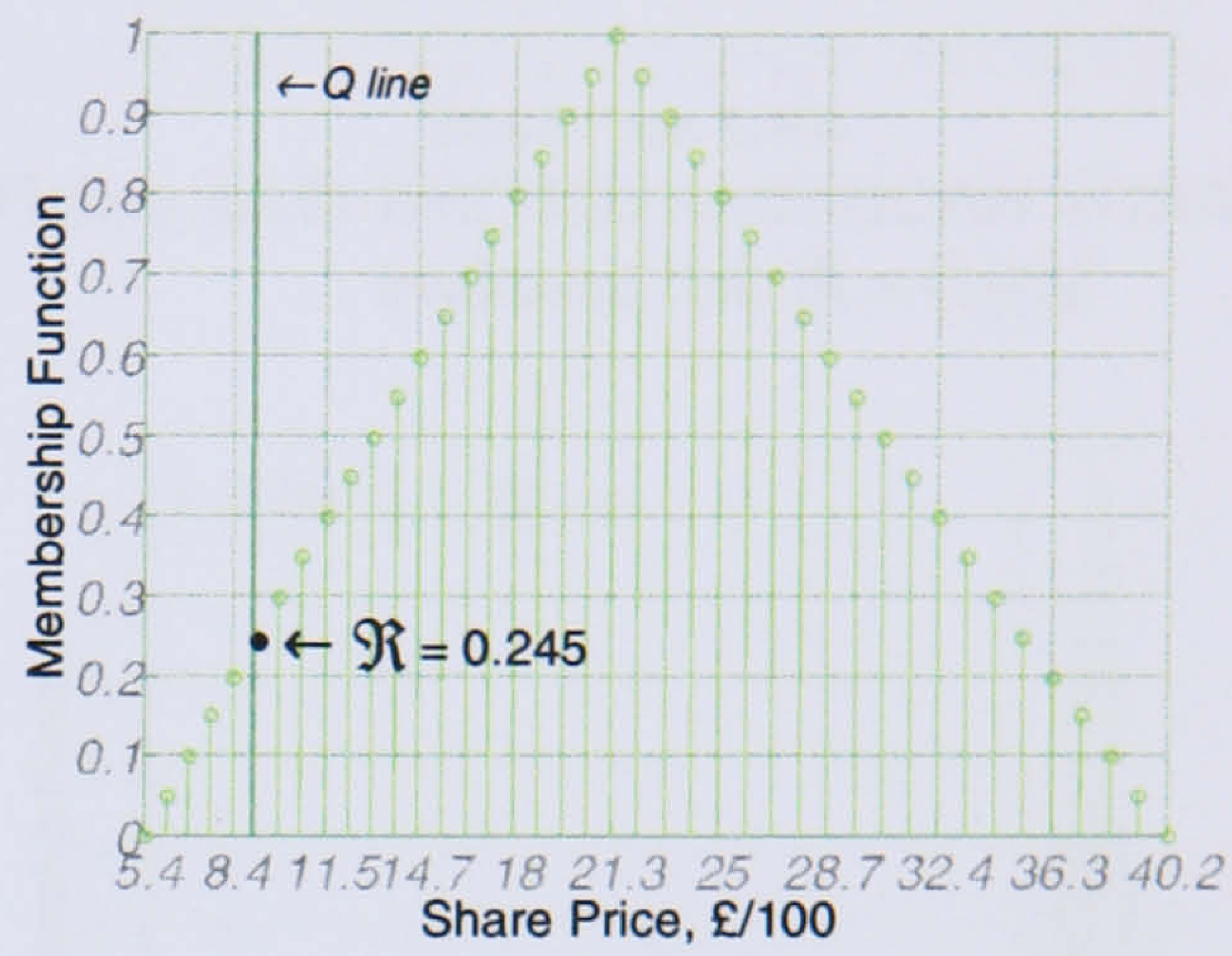


Figure A2.2: BBA GROUP - evaluated risk $\Re = 0.245$

Figure A2.3: BENTALLS - evaluated risk $\Re = 0.528$

Figure A2.4: BLUE CIRCLE INDUSTRIES - evaluated risk $\Re = 0$

Figure A2.5: BOC GROUP - evaluated risk $\Re = 0$

Figure A2.6: BOOTS CO. - evaluated risk $\Re = 1$

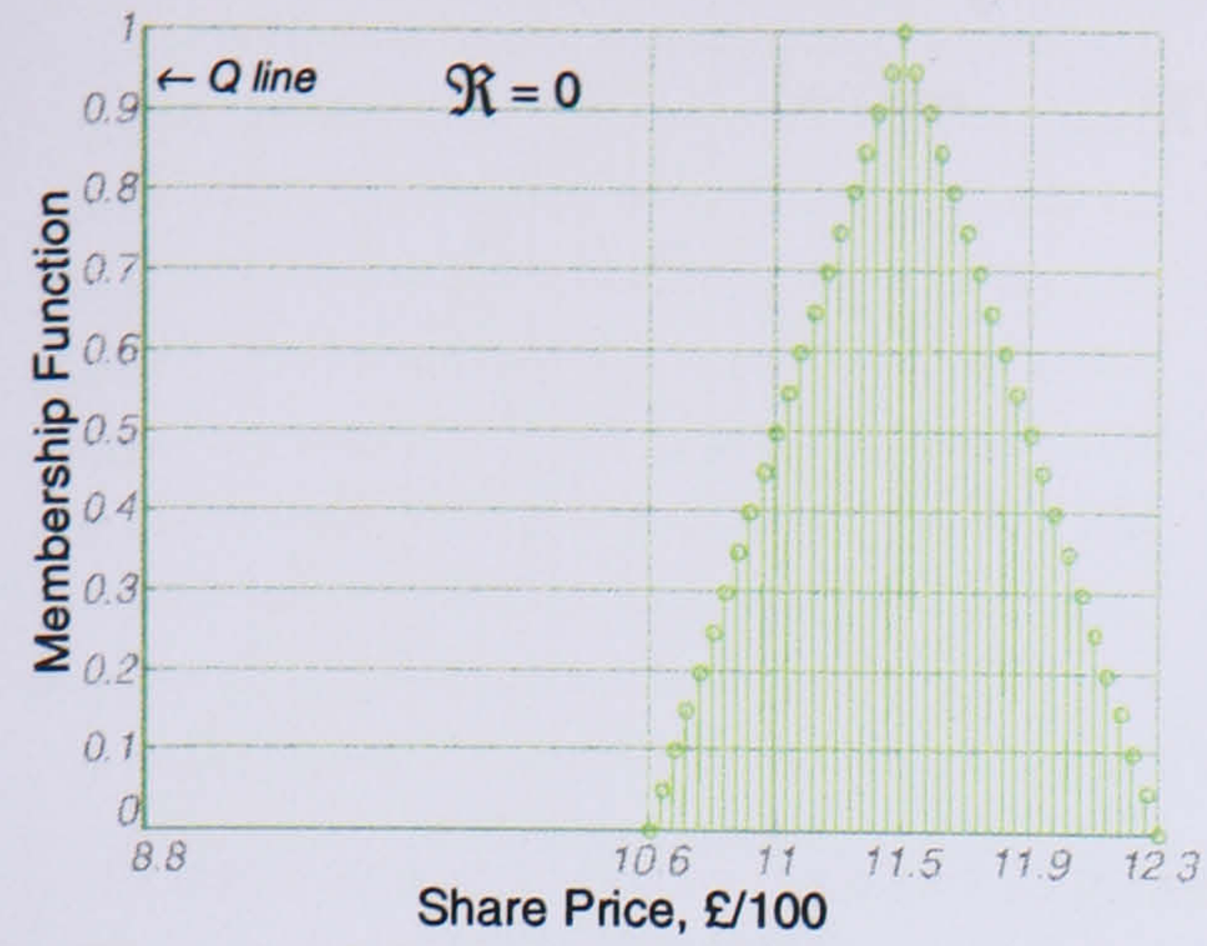


Figure A2.7: BP AMOCO - evaluated risk $\Re = 0$

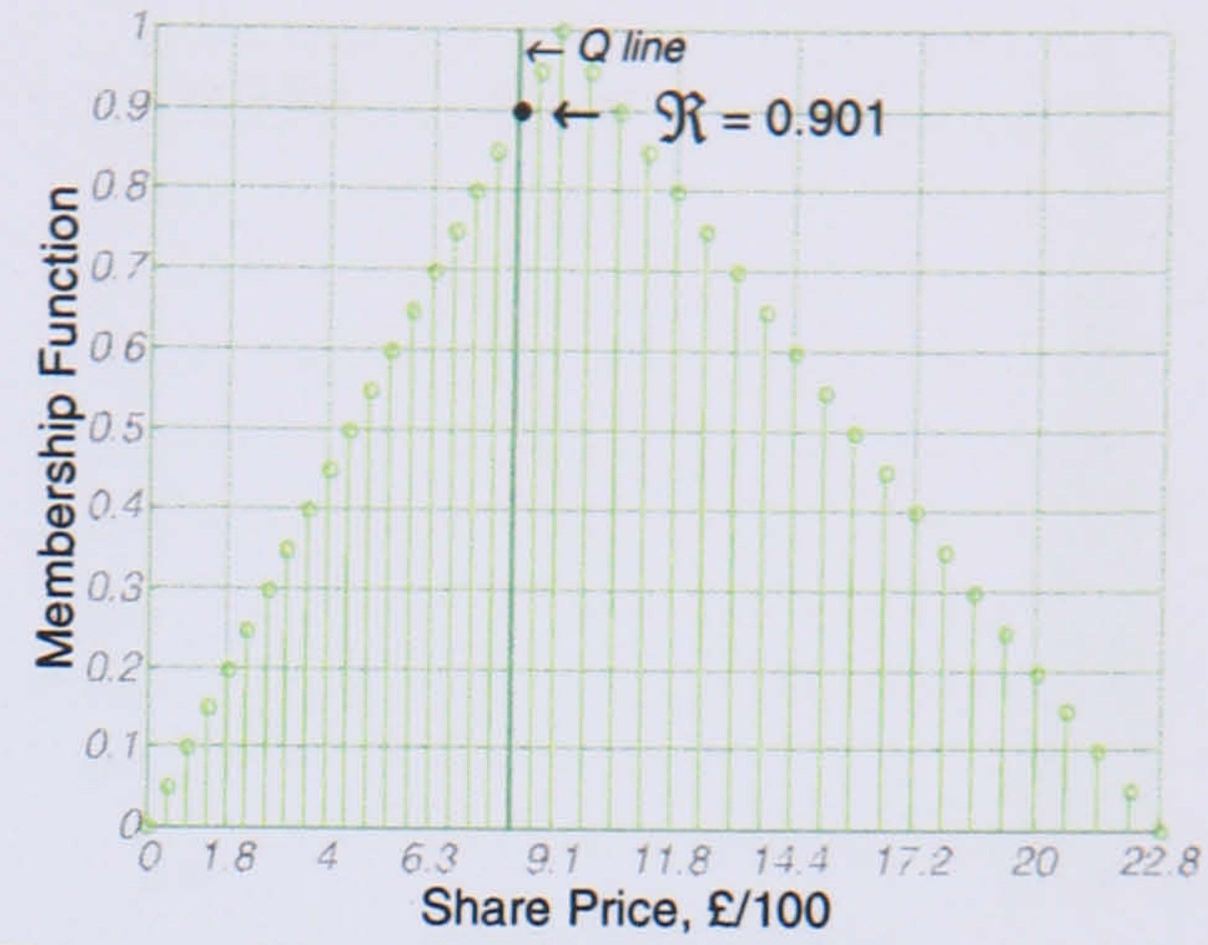


Figure A2.8: BRITISH AMERICAN TOBACCO - evaluated risk $\Re = 0.901$

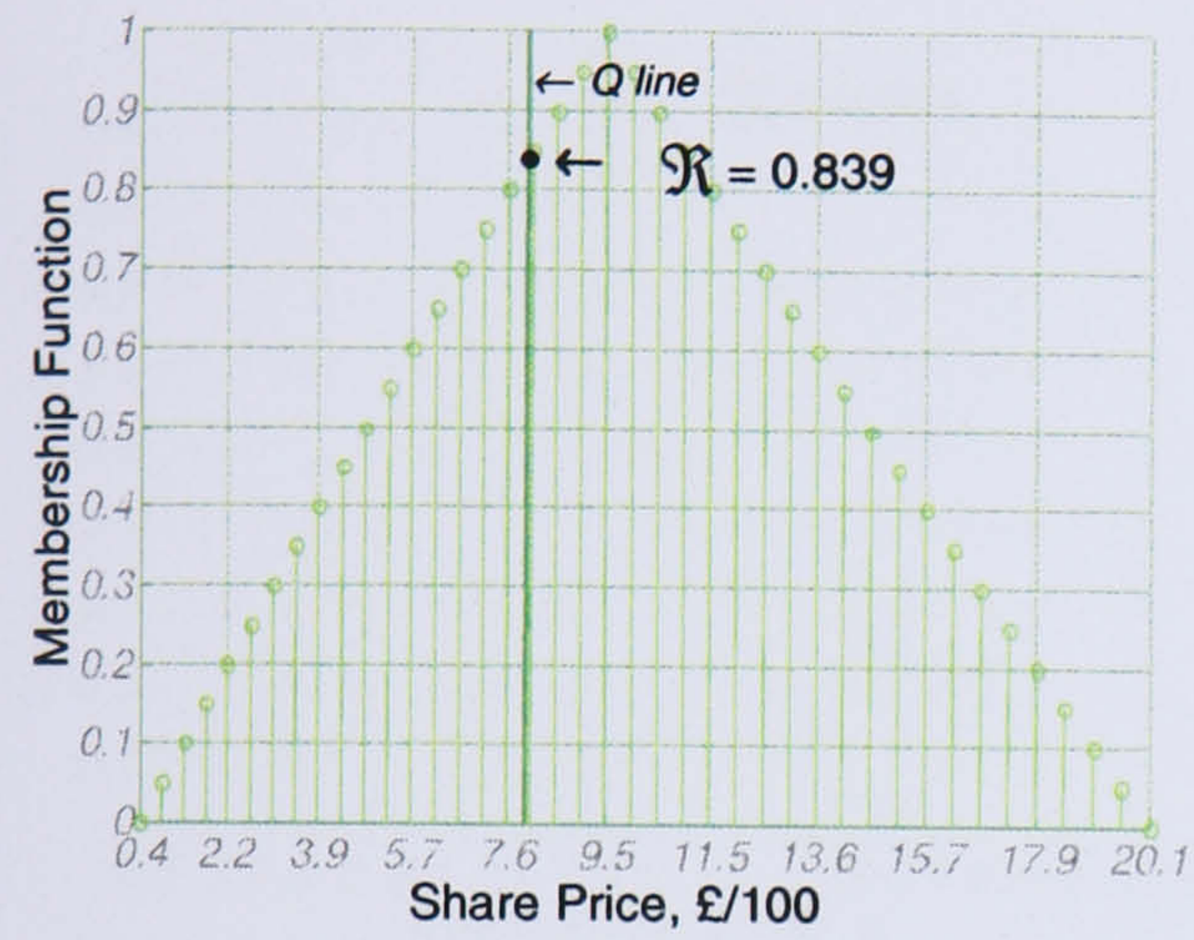


Figure A2.9: BUNZL - evaluated risk $\Re = 0.839$

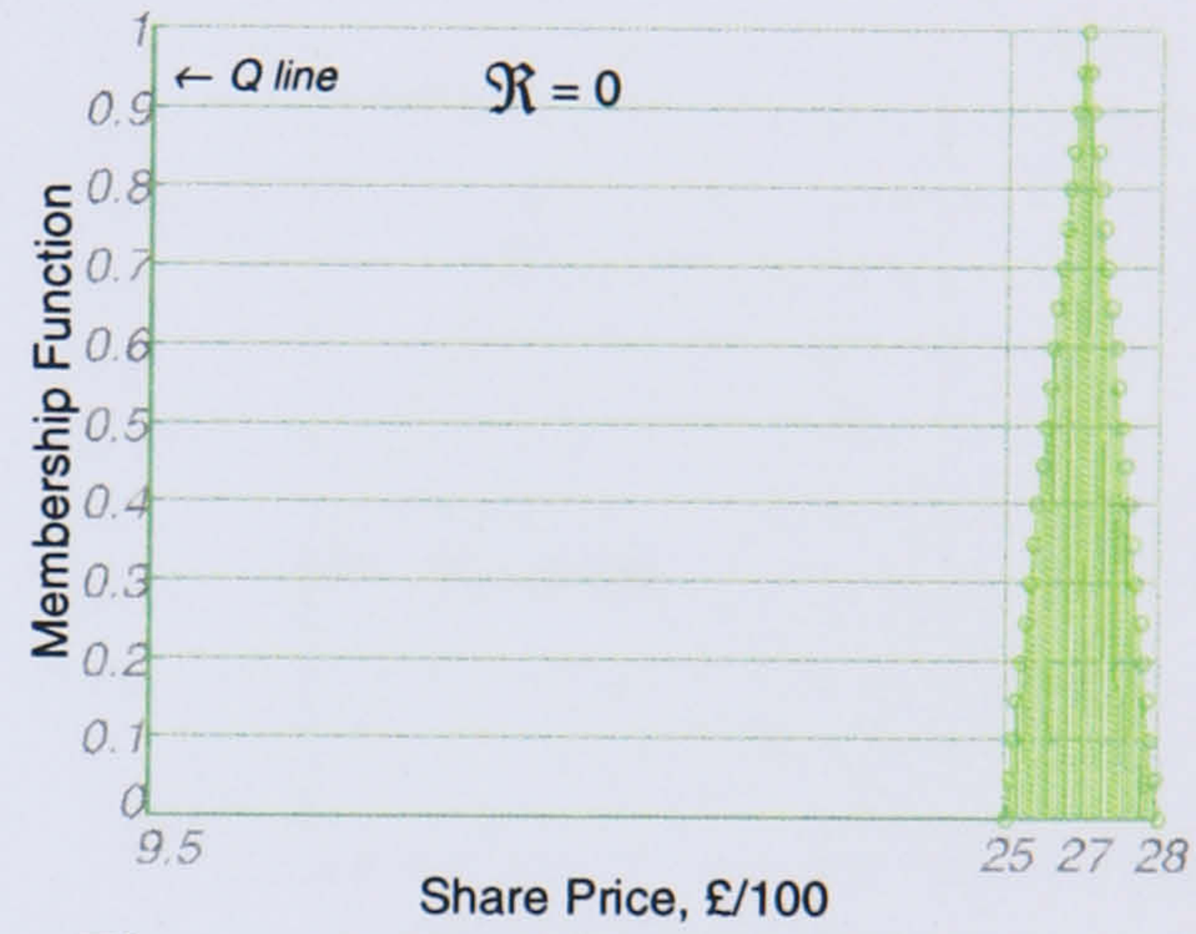


Figure A2.10: COATS VIYELLA - evaluated risk $\Re = 0$

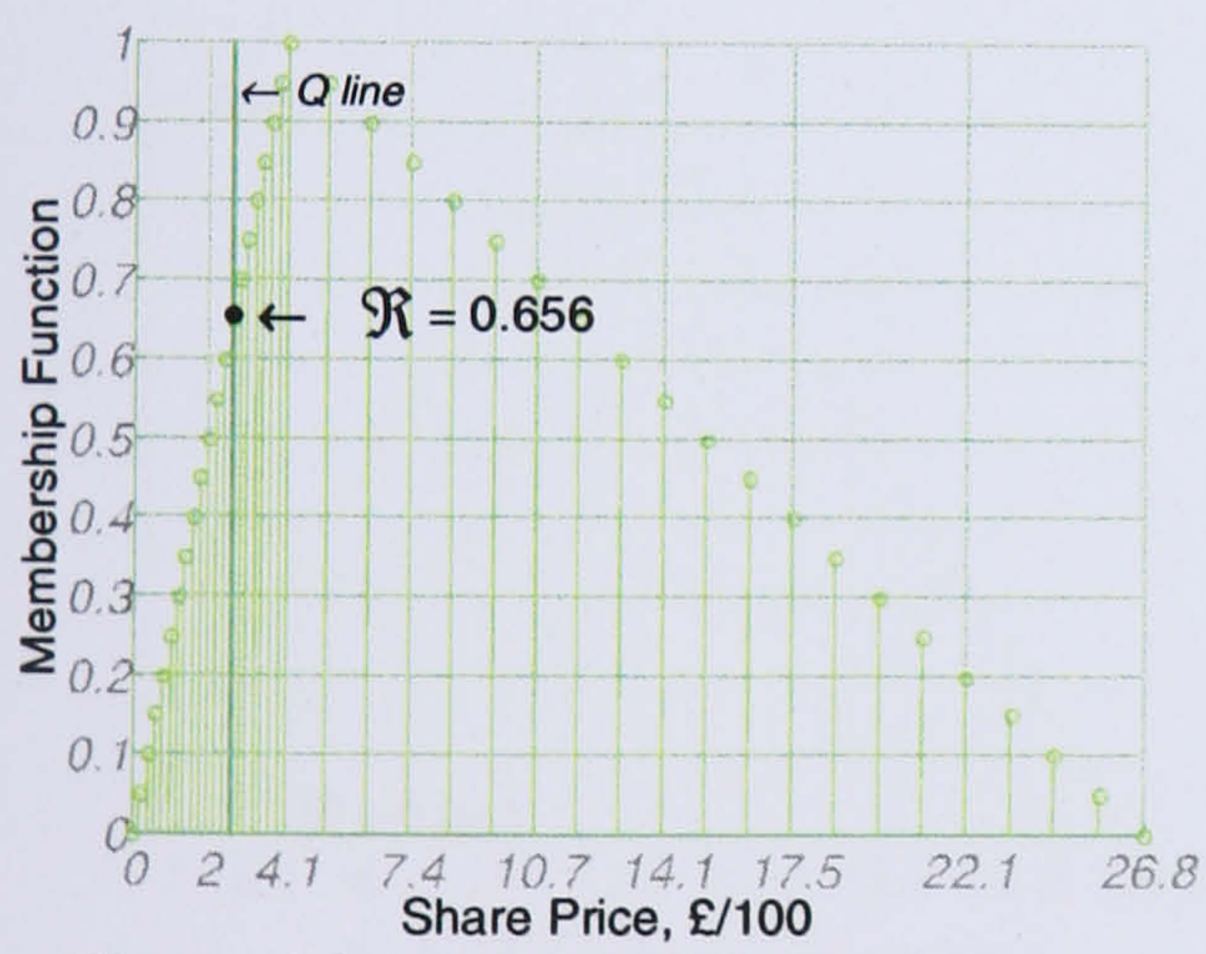


Figure A2.11: DIXONS GROUP - evaluated risk $\Re = 0.656$

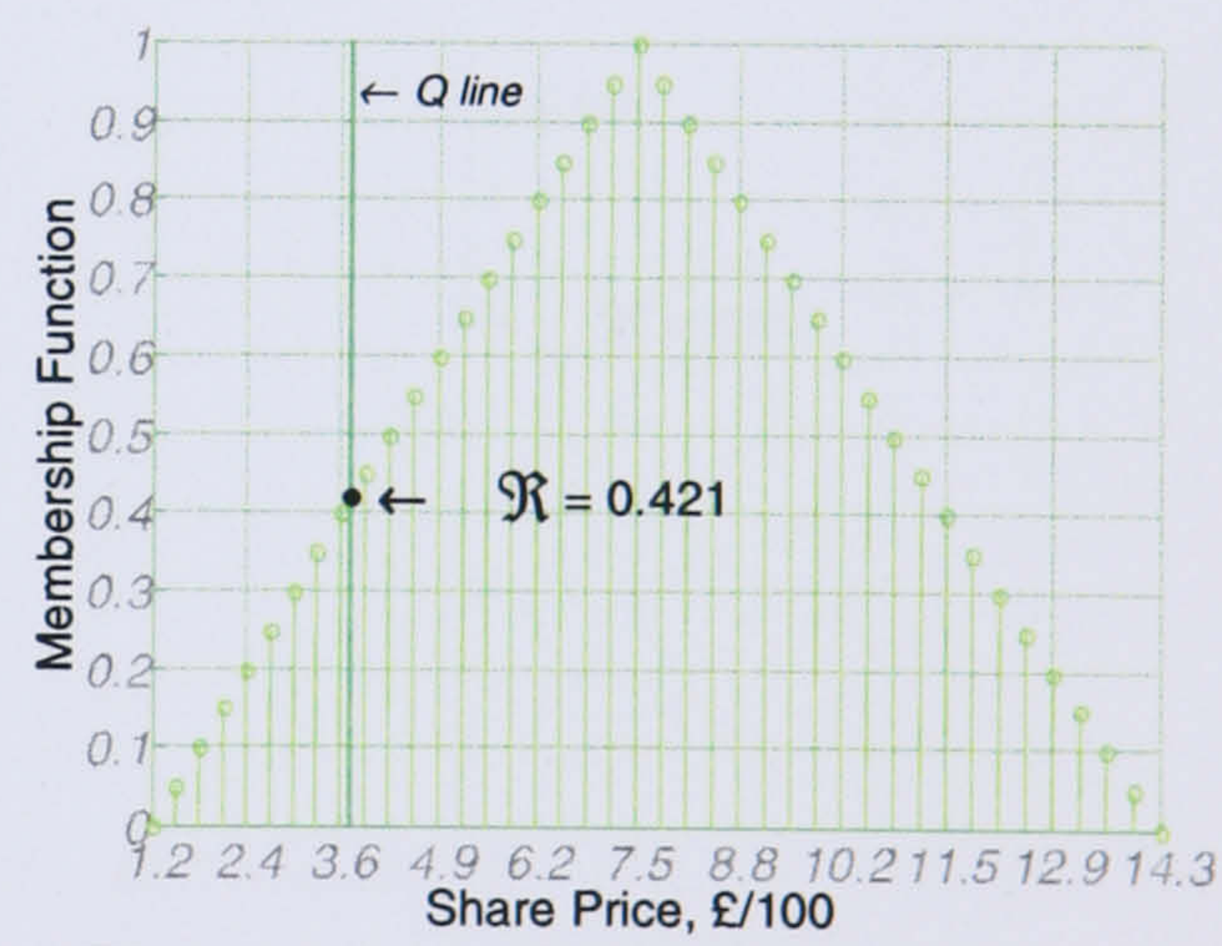


Figure A2.12: GOODWIN - evaluated risk $\Re = 0.421$

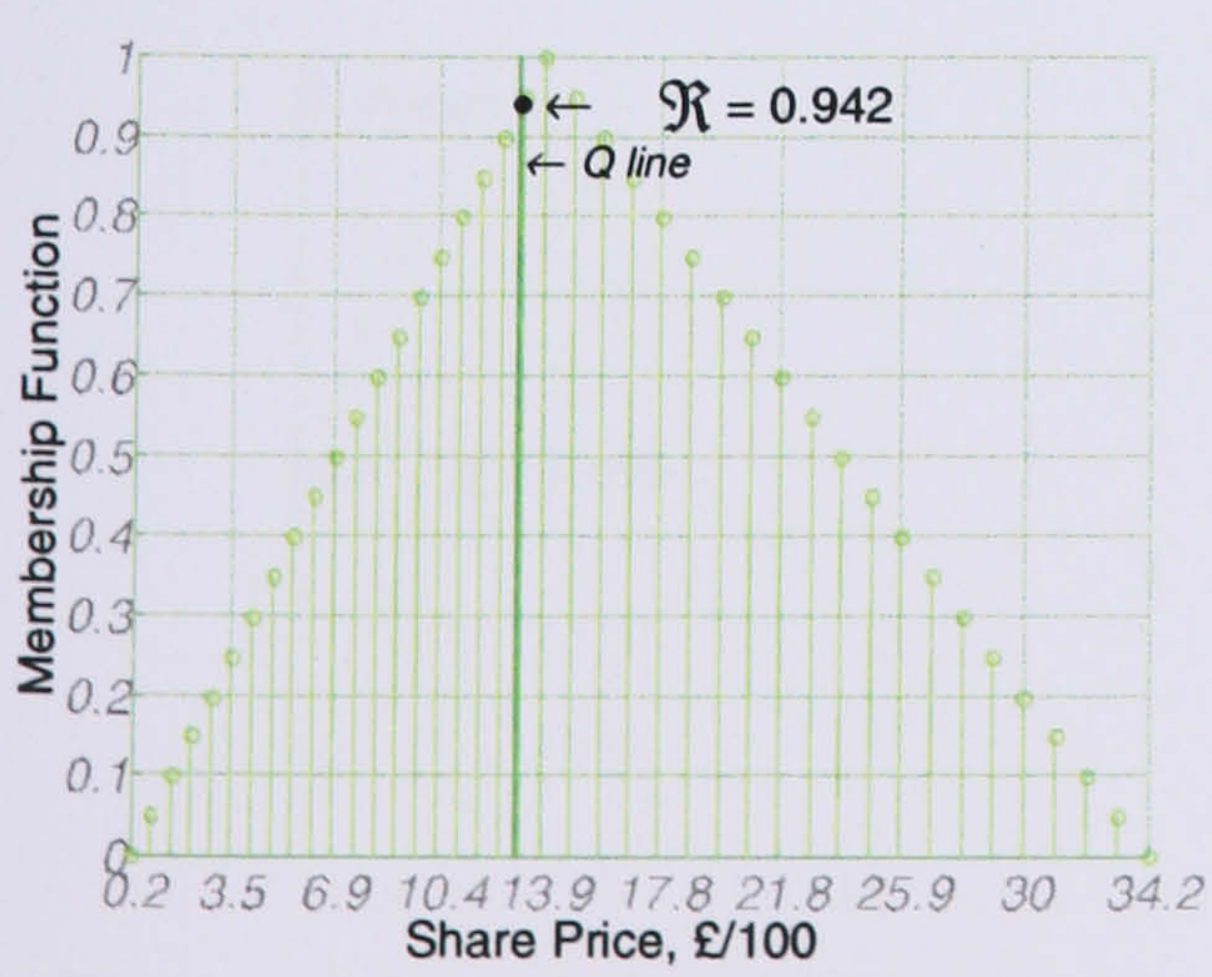


Figure A2.13: GREAT UNIVERSAL STORES - evaluated risk $\Re = 0.942$

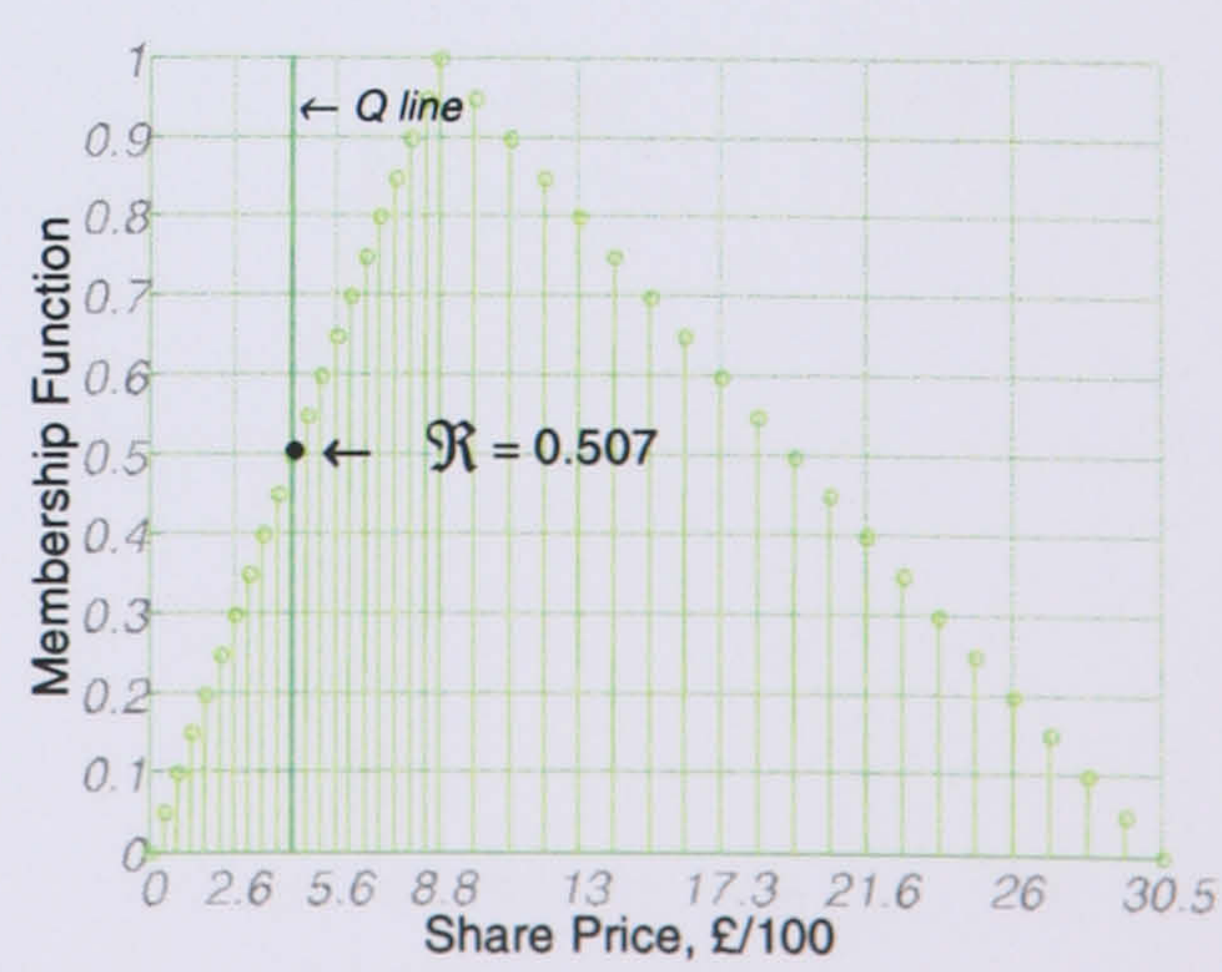


Figure A2.14: HANSON - evaluated risk $\Re = 0.507$

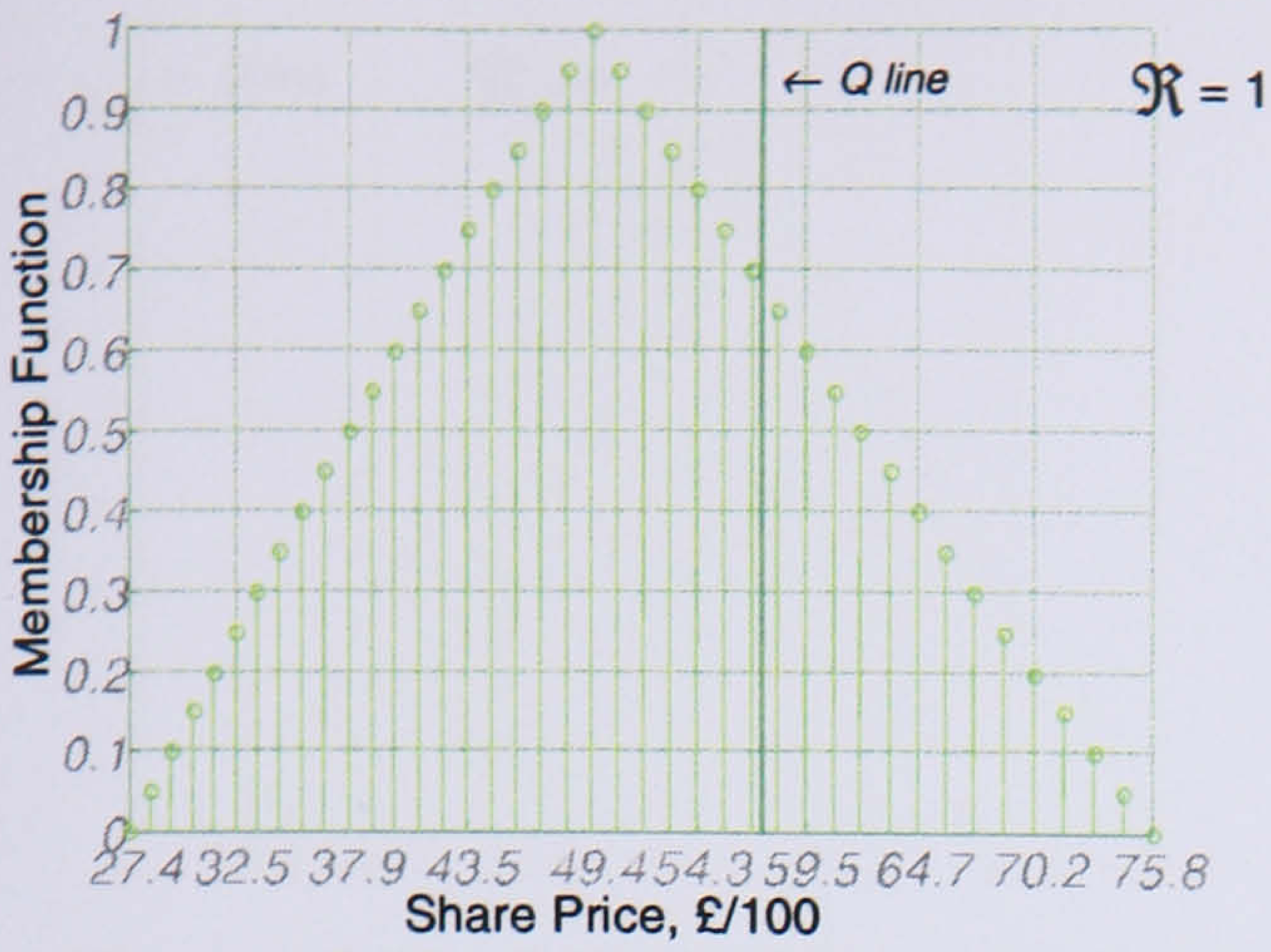


Figure A2.15: INCHCAPE - evaluated risk $\Re = 1$

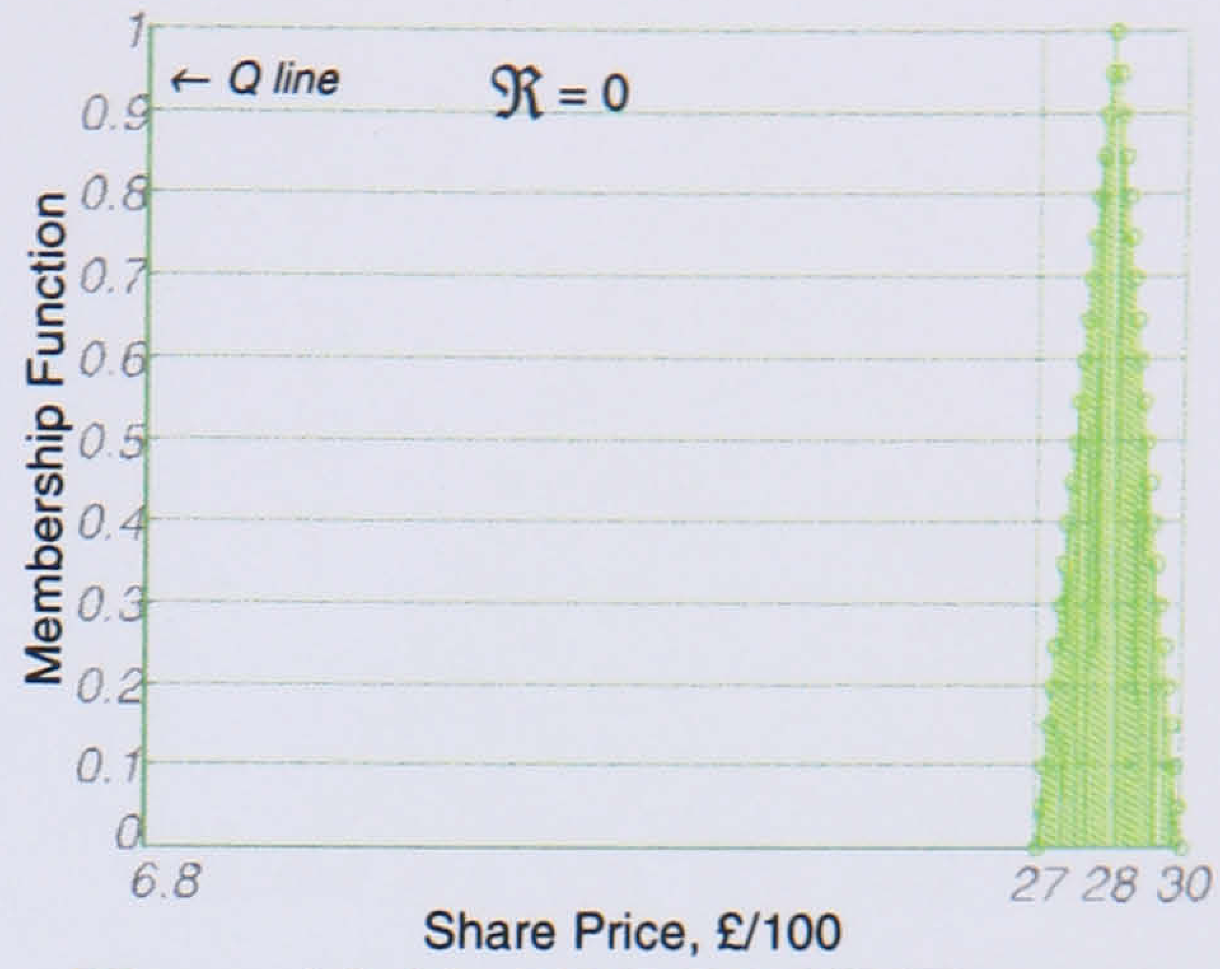


Figure A2.16: LEX SERVICE - evaluated risk $\Re = 0$

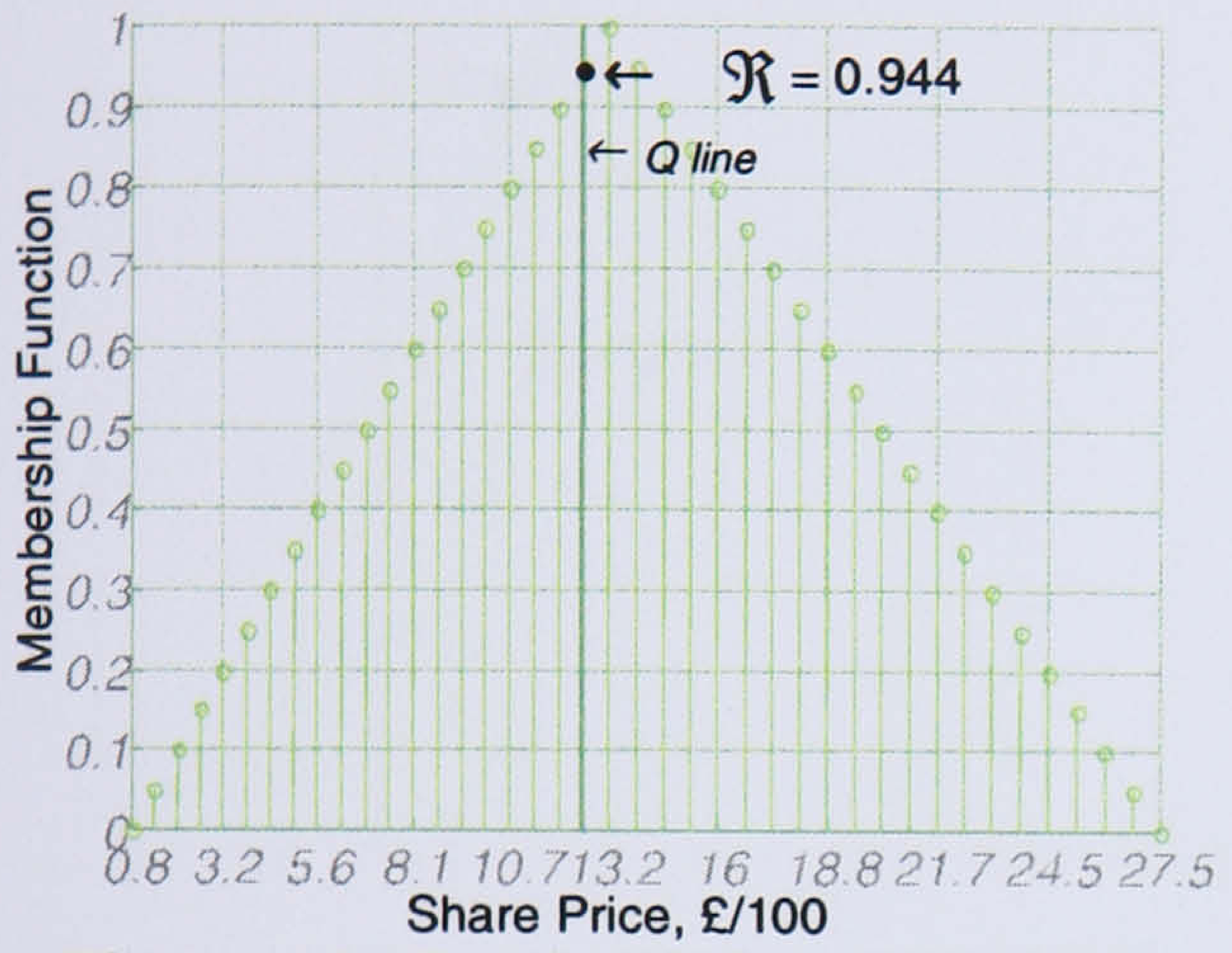


Figure A2.17: MARKS & SPENCER - evaluated risk $\Re = 0.944$

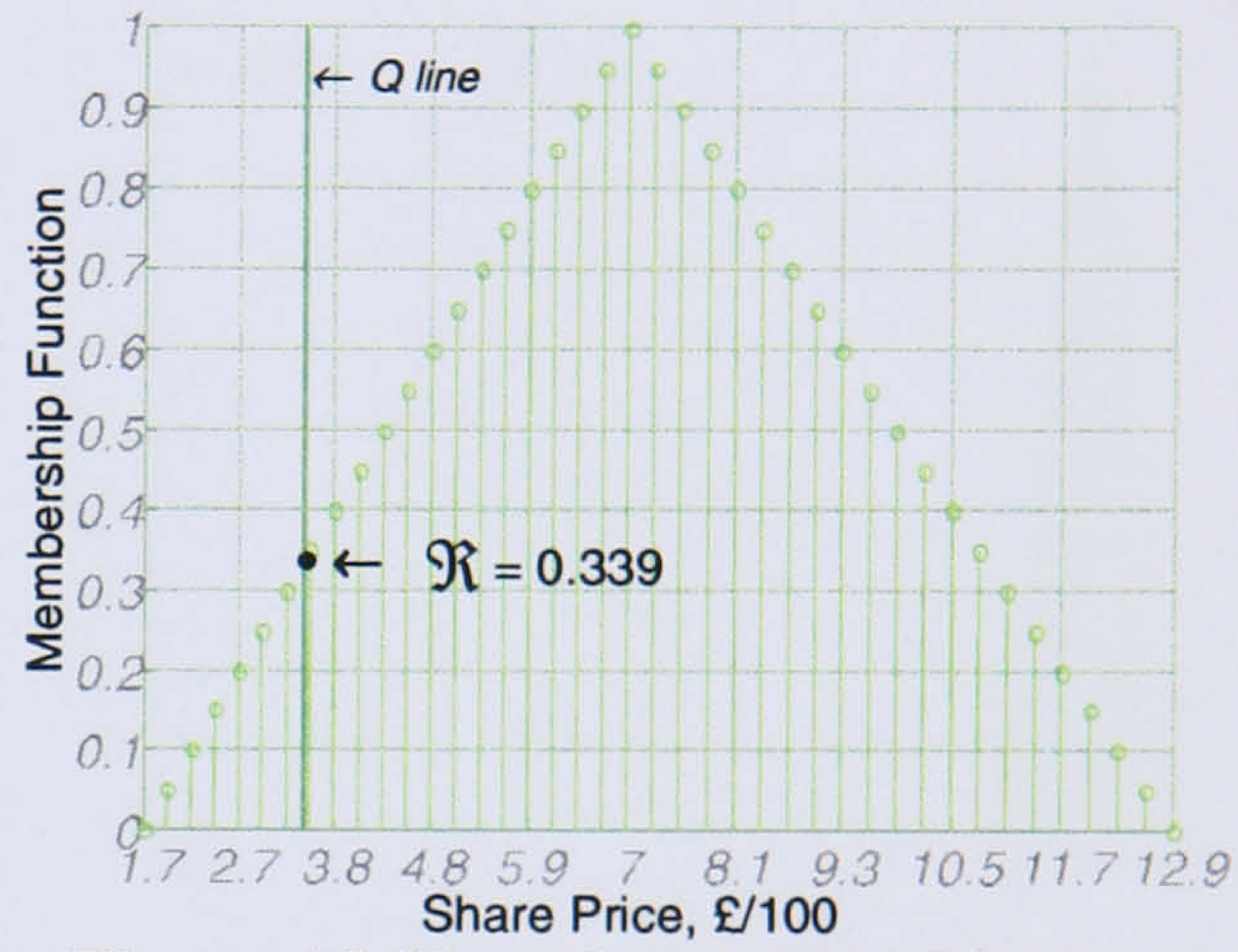


Figure A2.18: NORTHERN FOODS - evaluated risk $\Re = 0.339$

Figure A2.19: PILKINGTON - evaluated risk $\Re = 0.026$

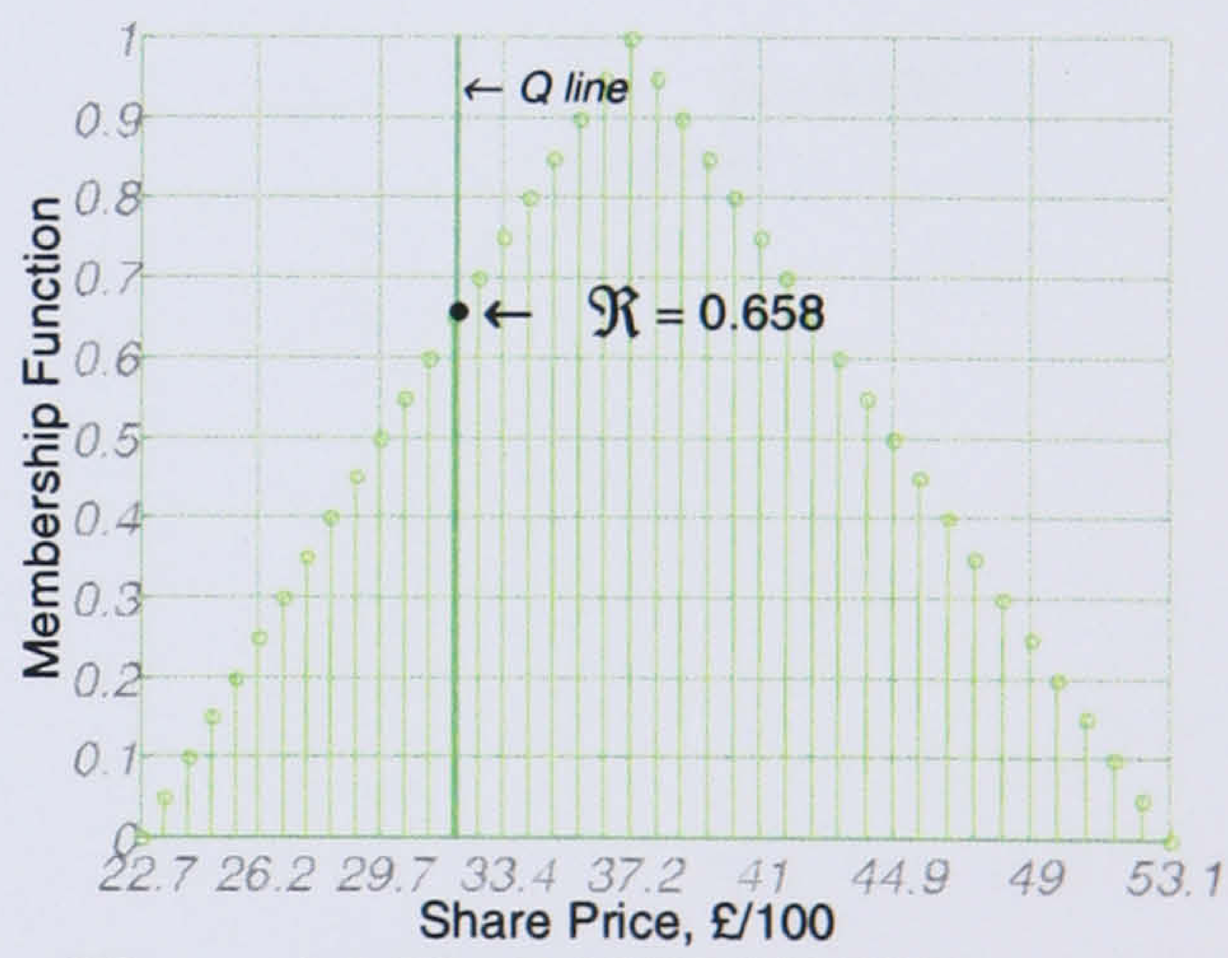


Figure A2.20: RANK GROUP - evaluated risk $\Re = 0.658$

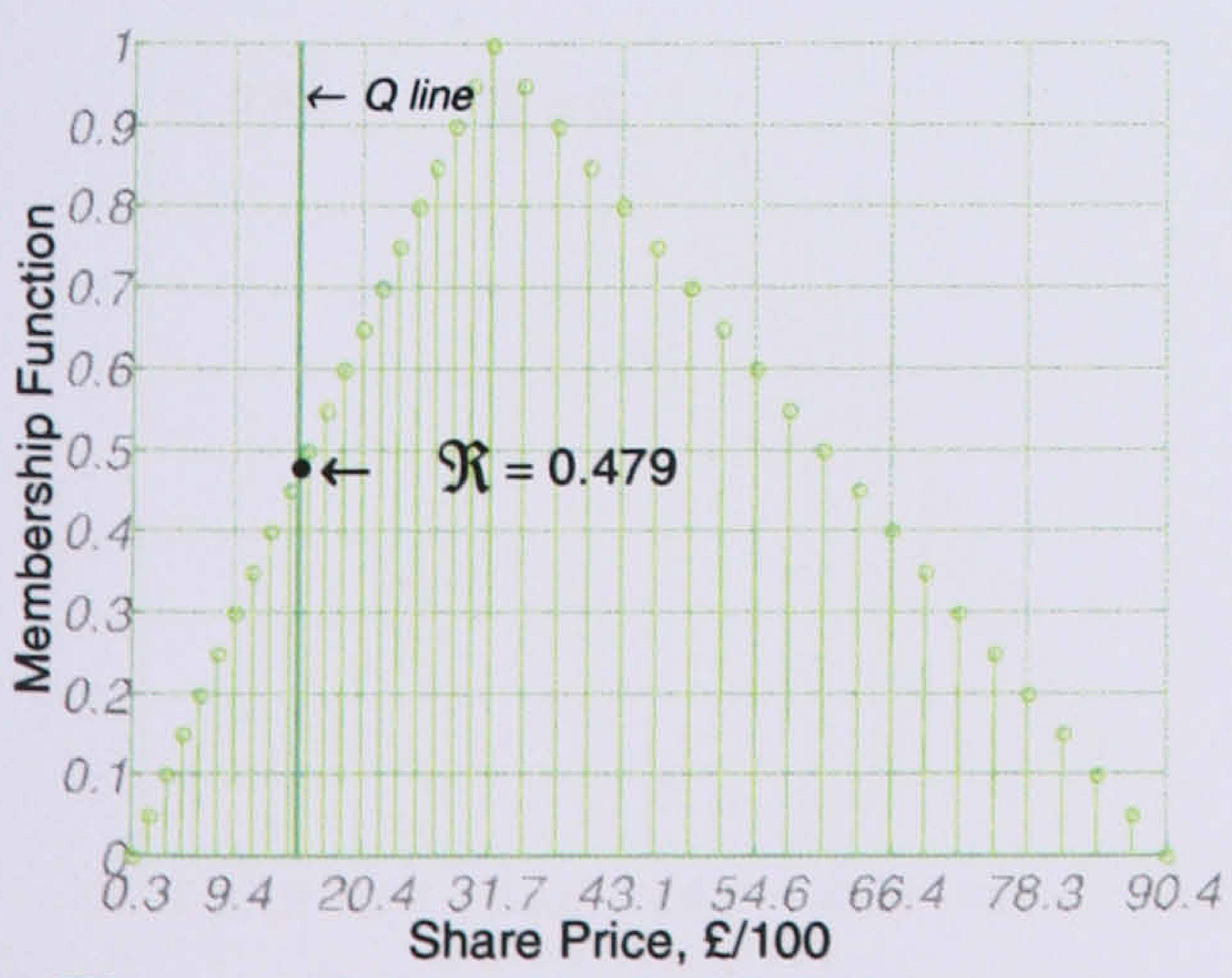


Figure A2.21: RMC GROUP - evaluated risk $\Re = 0.479$

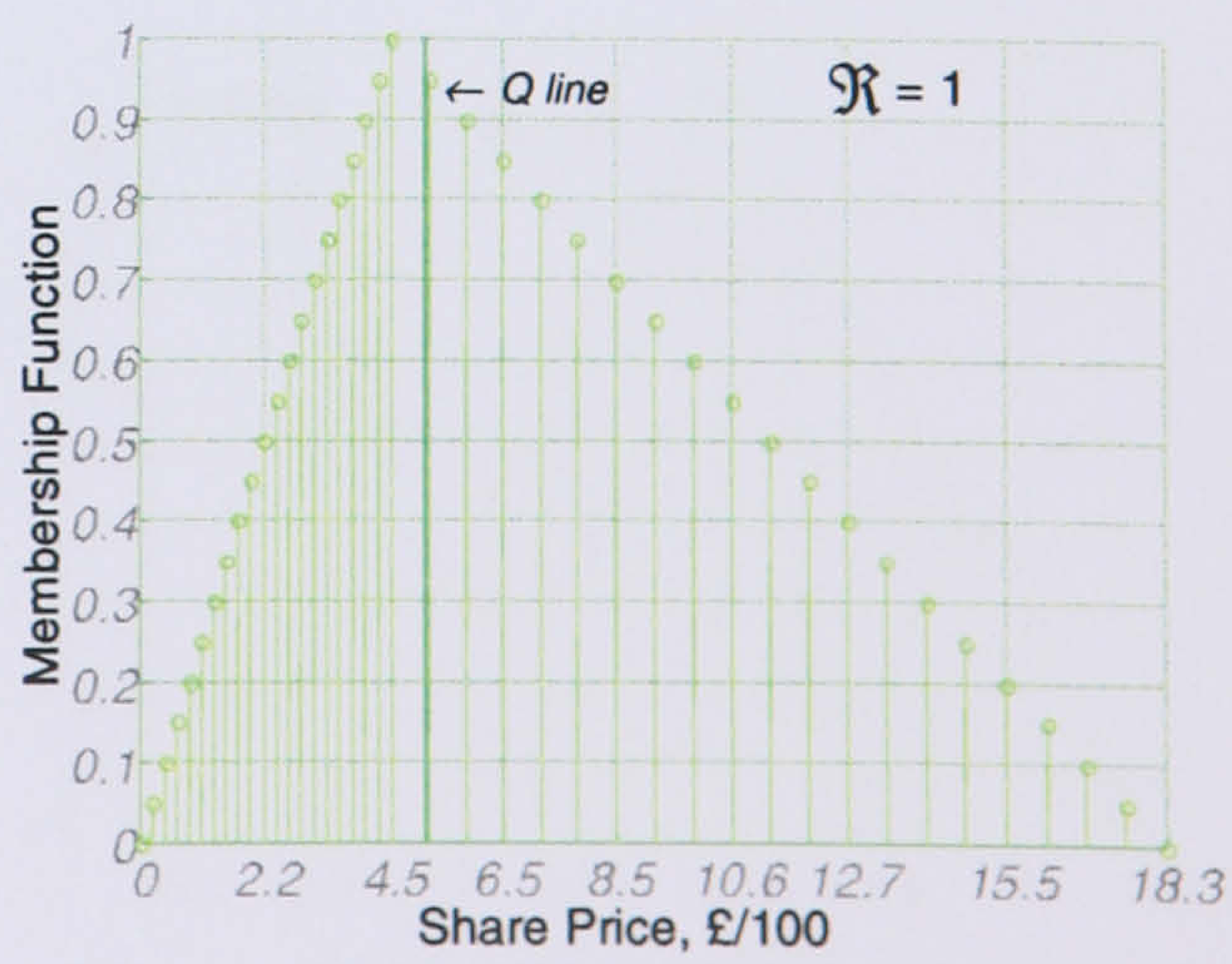


Figure A2.22: SINSBURY (J) - evaluated risk $\Re = 1$

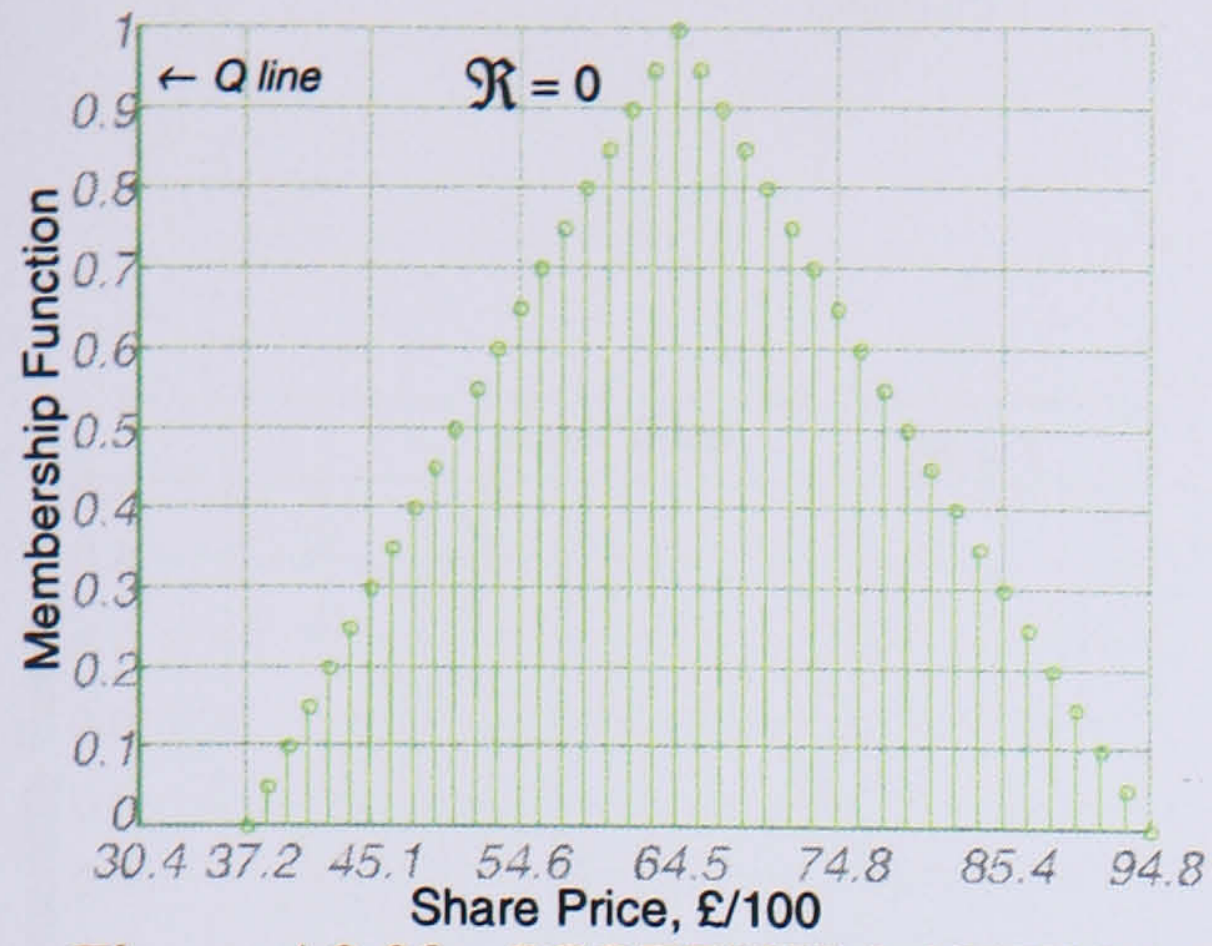


Figure A2.23: SCOTTISH & NEWCASTLE - evaluated risk $\Re = 0.487$

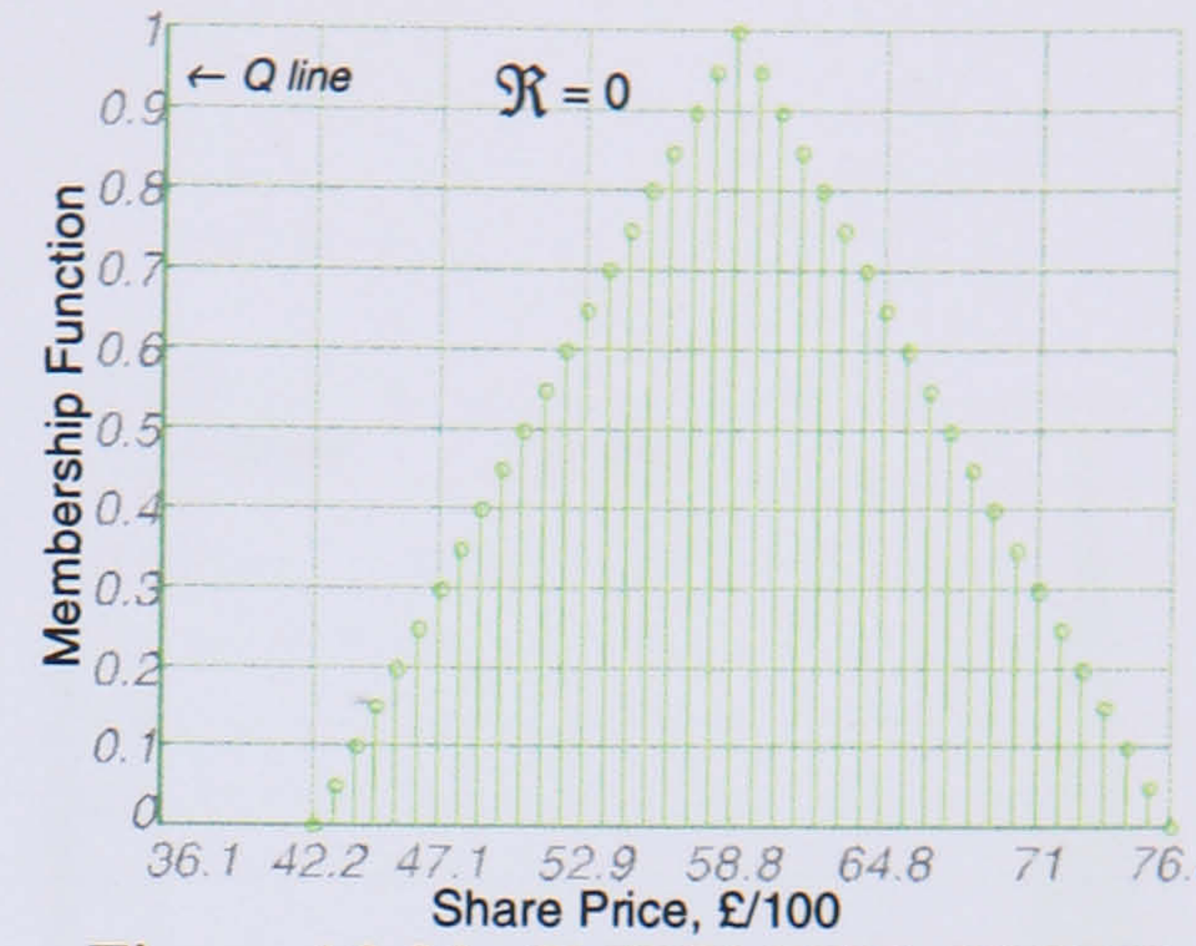


Figure A2.24: SMITH (WH) GROUP - evaluated risk $\Re = 0.986$

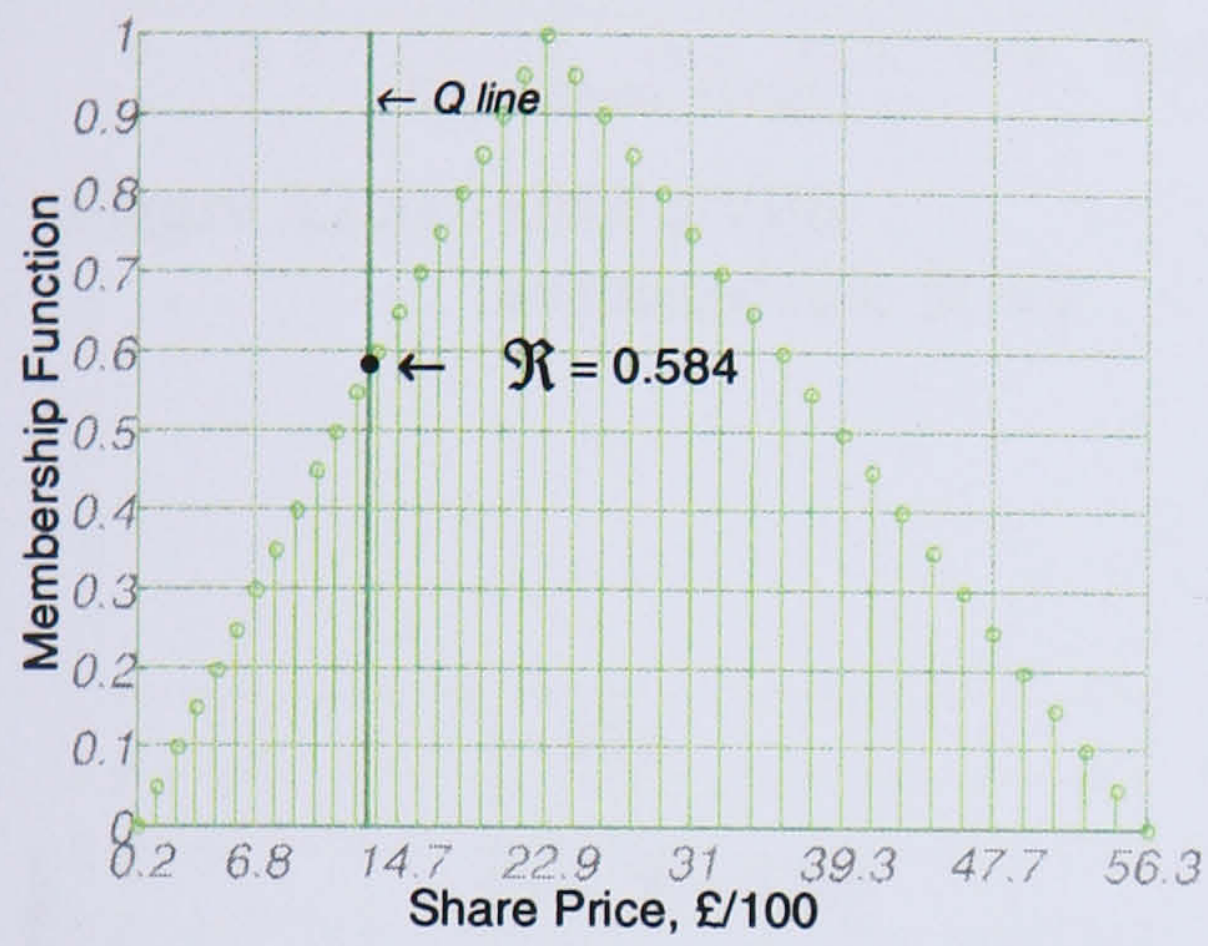


Figure A2.25: SMITHS INDUSTRIES - evaluated risk $\Re = 0.584$

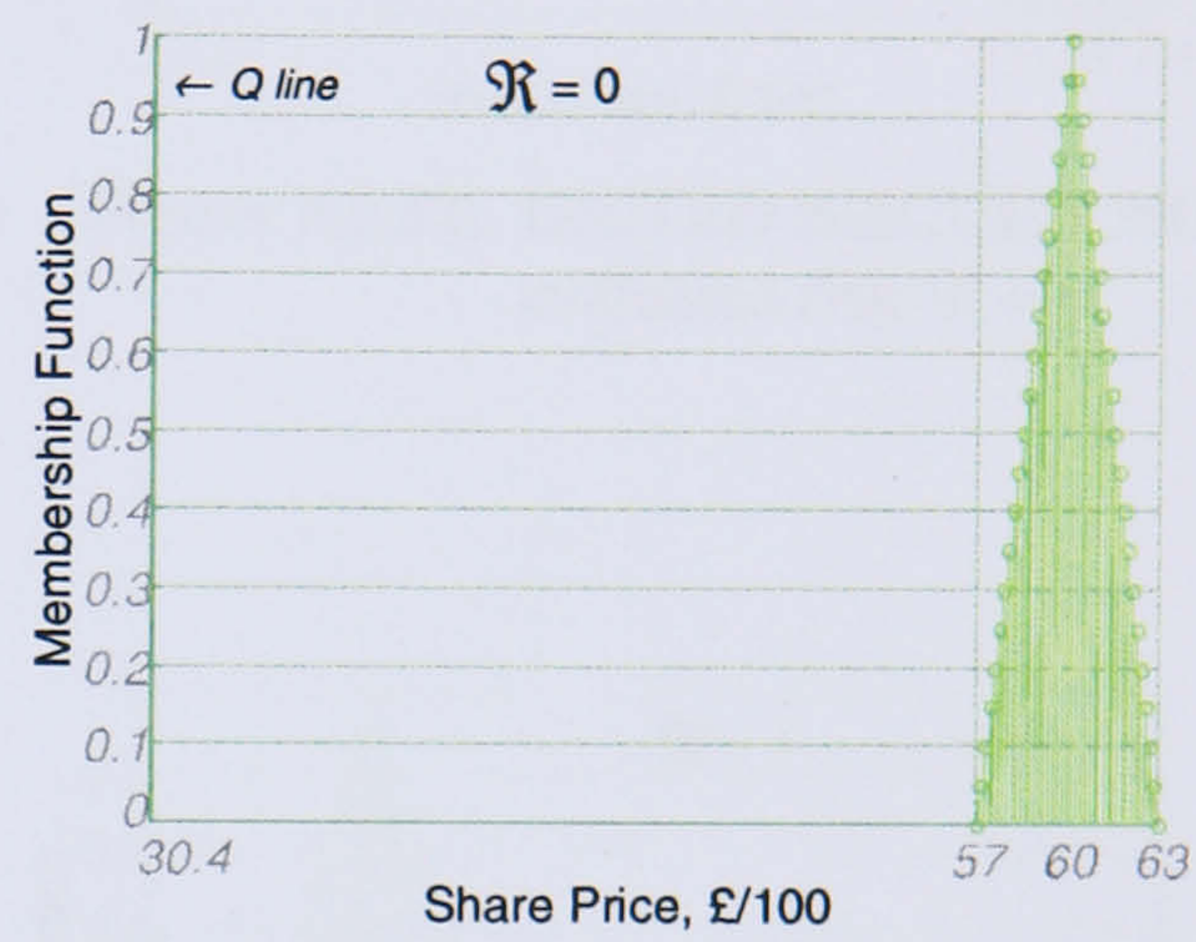


Figure A2.26: TARMAC - evaluated risk $\Re = 0$

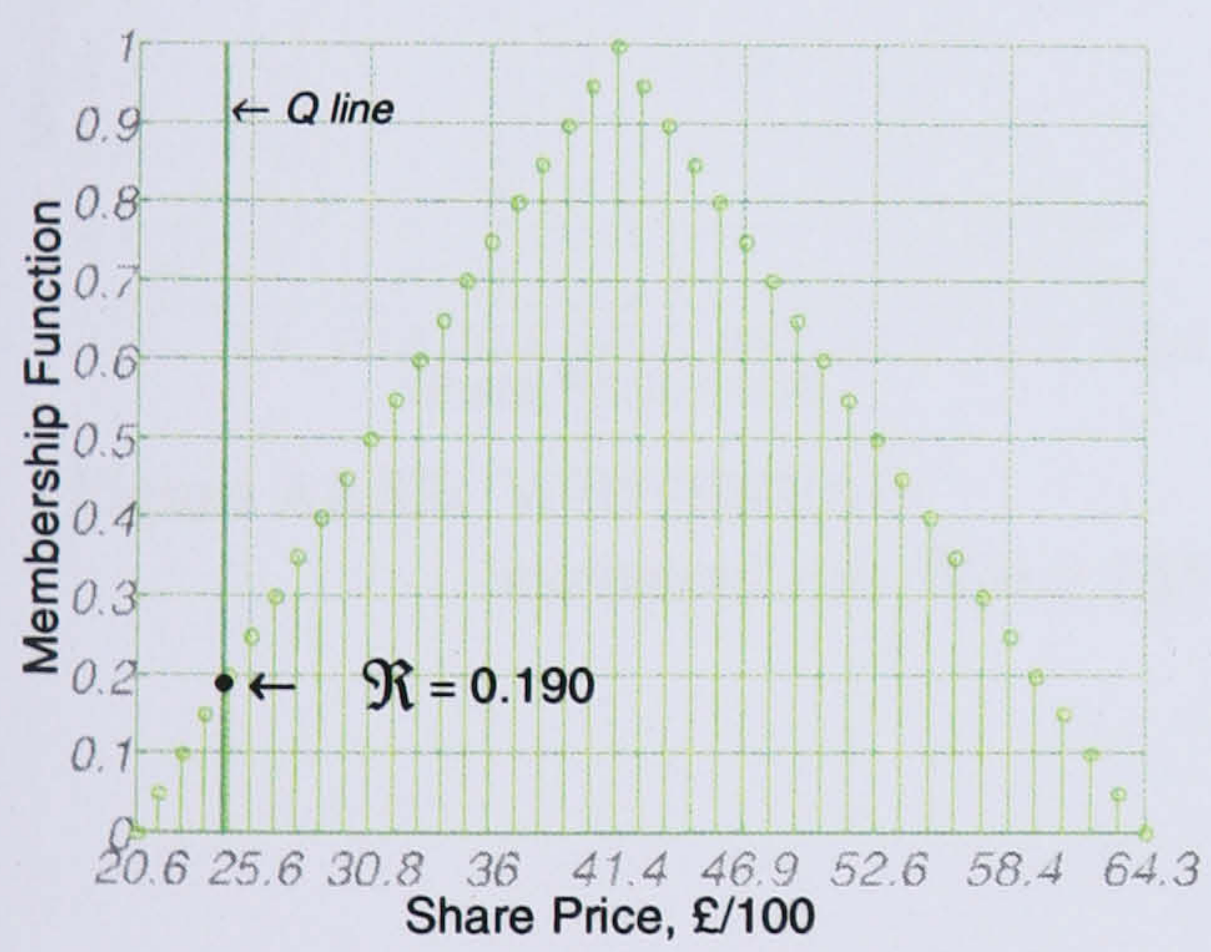


Figure A2.27: TATE & LYLE - evaluated risk $\Re = 0.190$

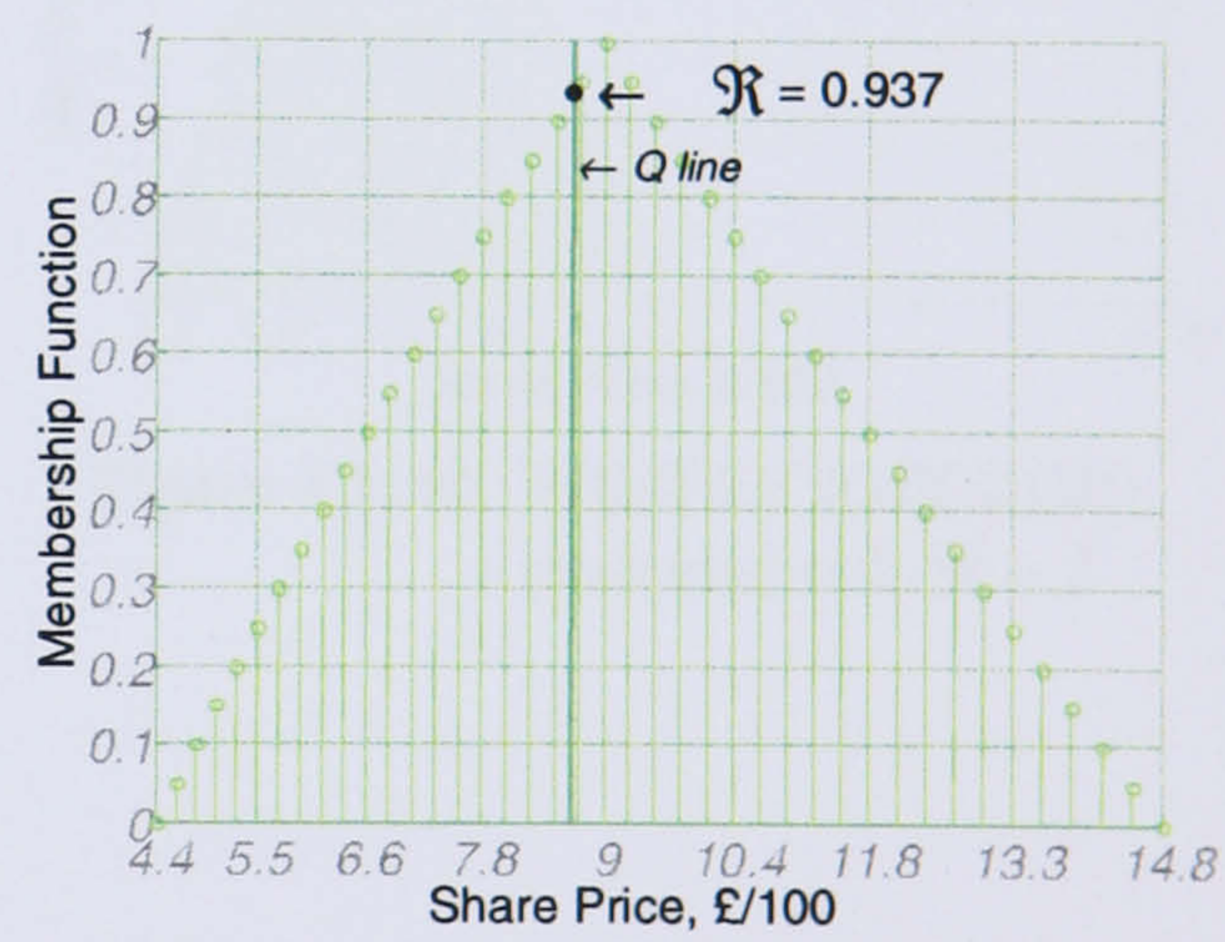


Figure A2.28: TAYLOR WOODROW - evaluated risk $\Re = 0.937$

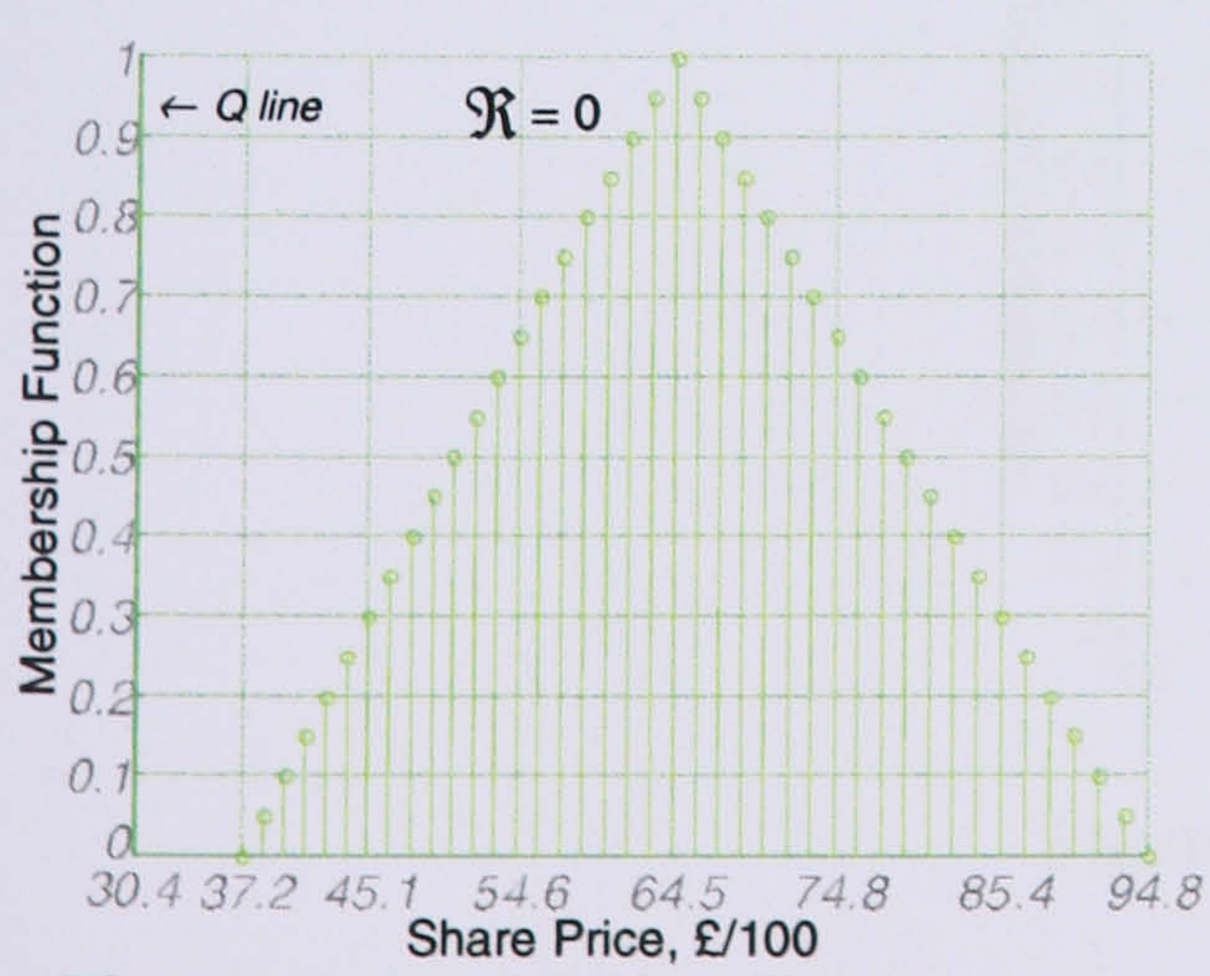


Figure A2.29: TI GROUP - evaluated risk $\Re = 0$

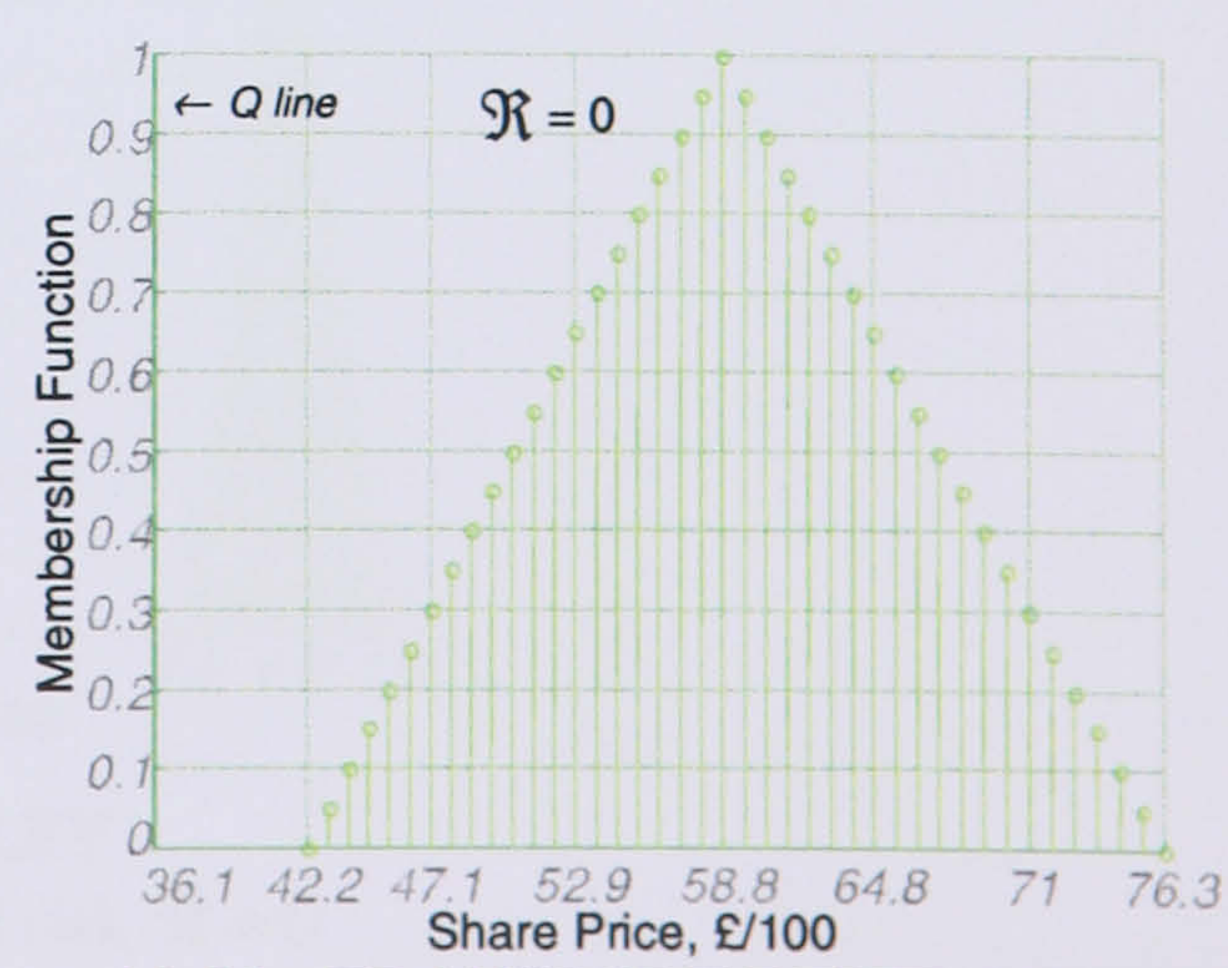


Figure A2.30: TRANSPORT DEVELOPMENT GROUP - evaluated risk $\Re = 0$

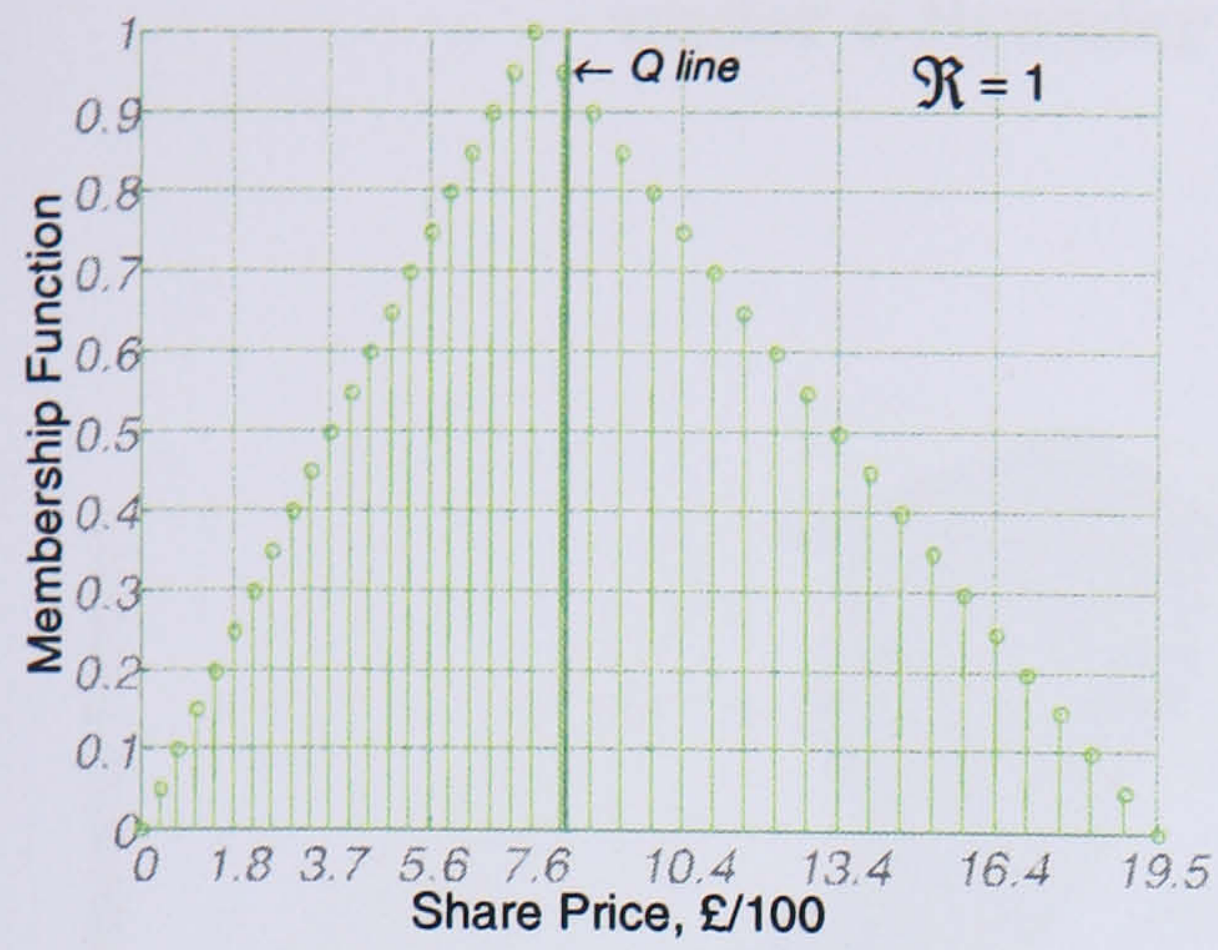


Figure A2.31: UNILEVER
evaluated risk $\Re = 1$

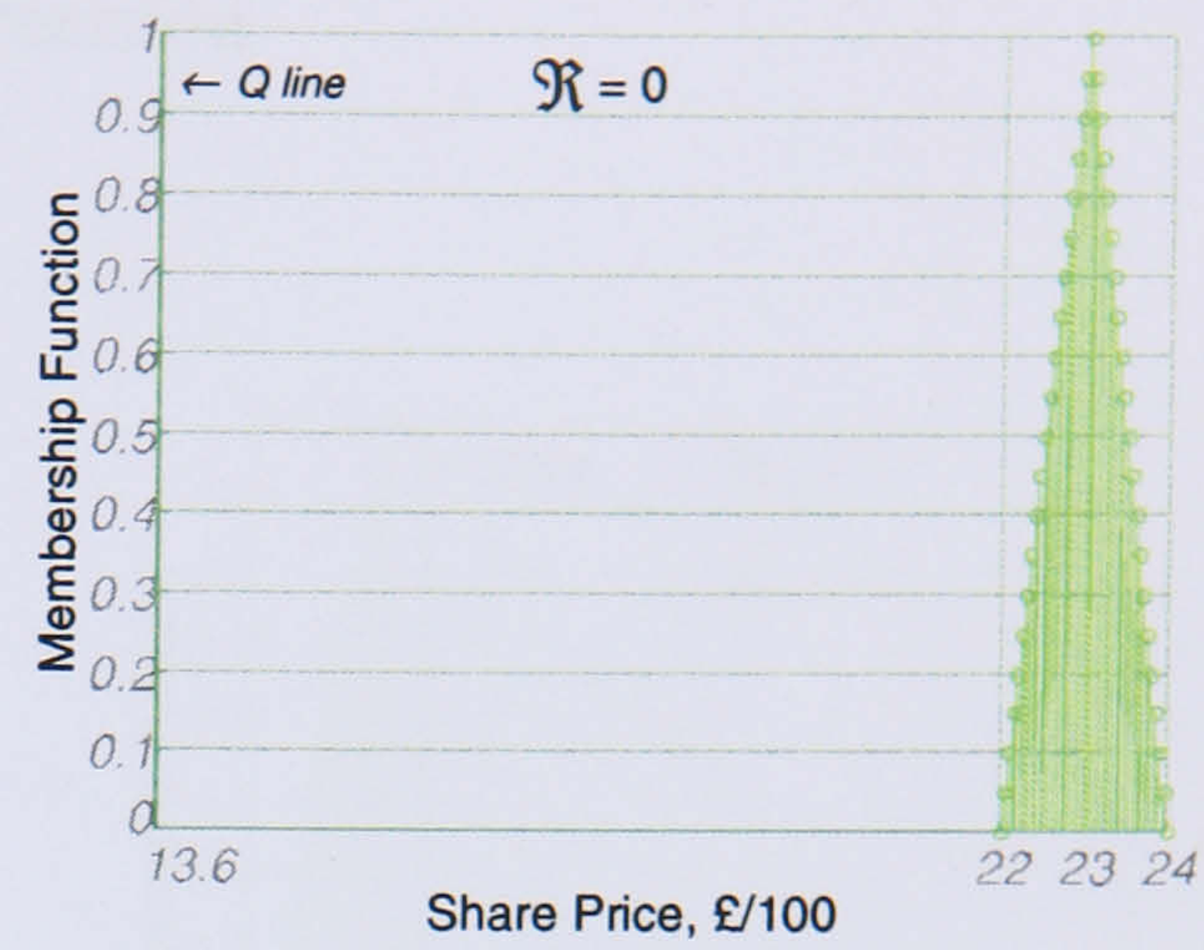


Figure A2.32: UNITED BISCUITS HOLDINGS
evaluated risk $\Re = 0$

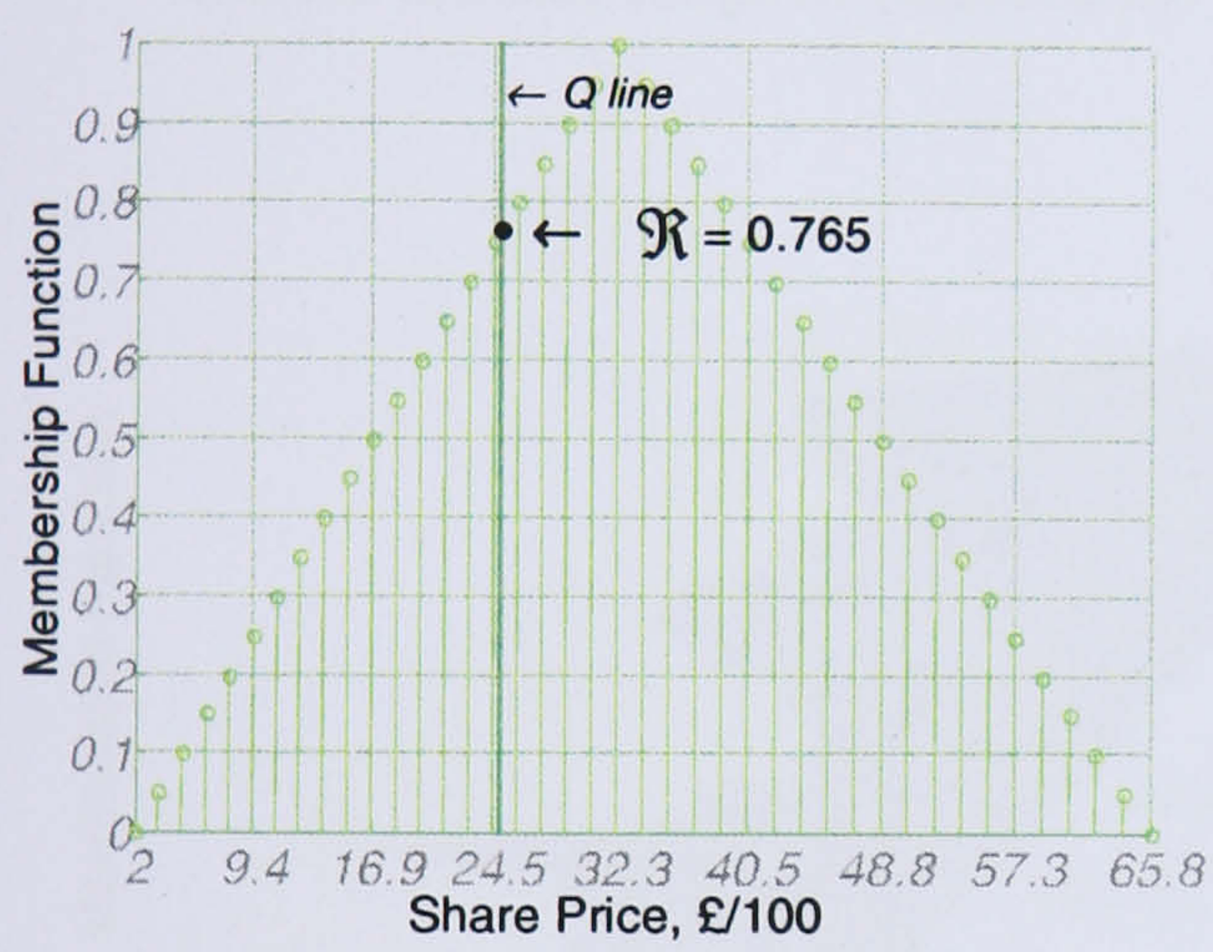


Figure A2.33: WHITBREAD
evaluated risk $\Re = 0.765$

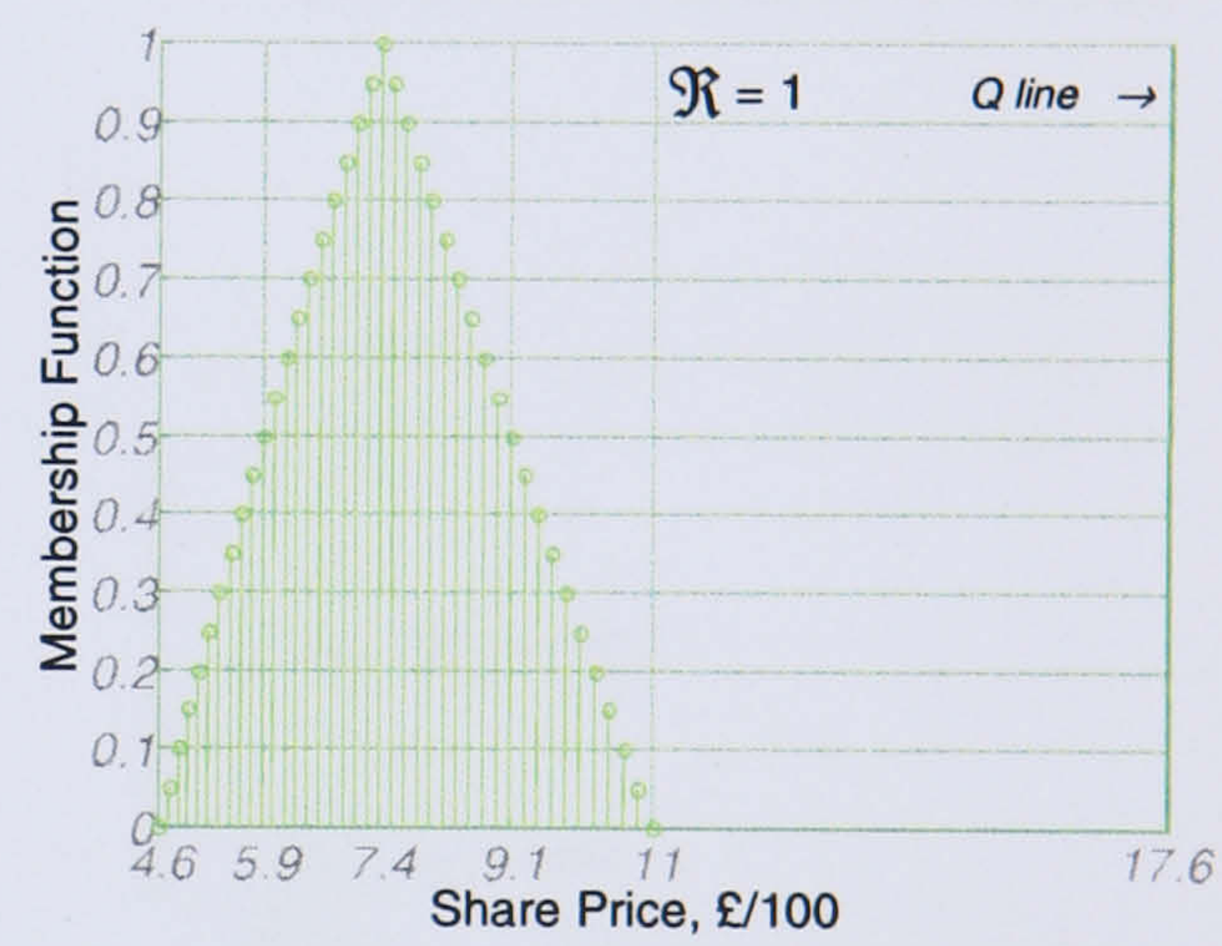


Figure 4.1.34: WIMPEY (GEORGE)
evaluated risk $\Re = 1$

Figure A2.35: WOLSELEY
evaluated risk $\Re = 0$

Appendix A3: Fuzzified Data and Evaluated Robustness by Company under a Broader Range of Imprecision

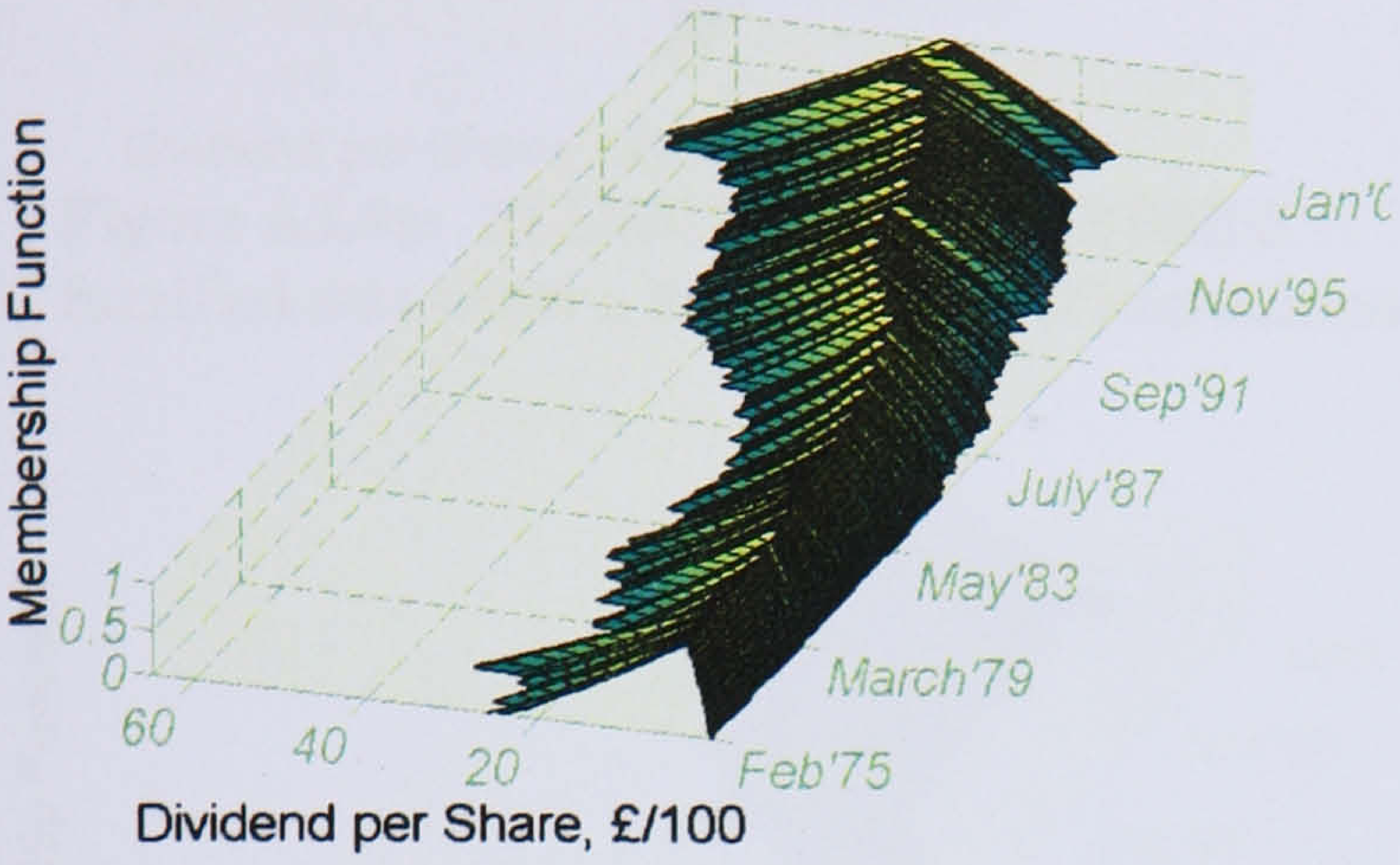


Figure A3.1a: BASS - fuzzified data under a broader range of imprecision

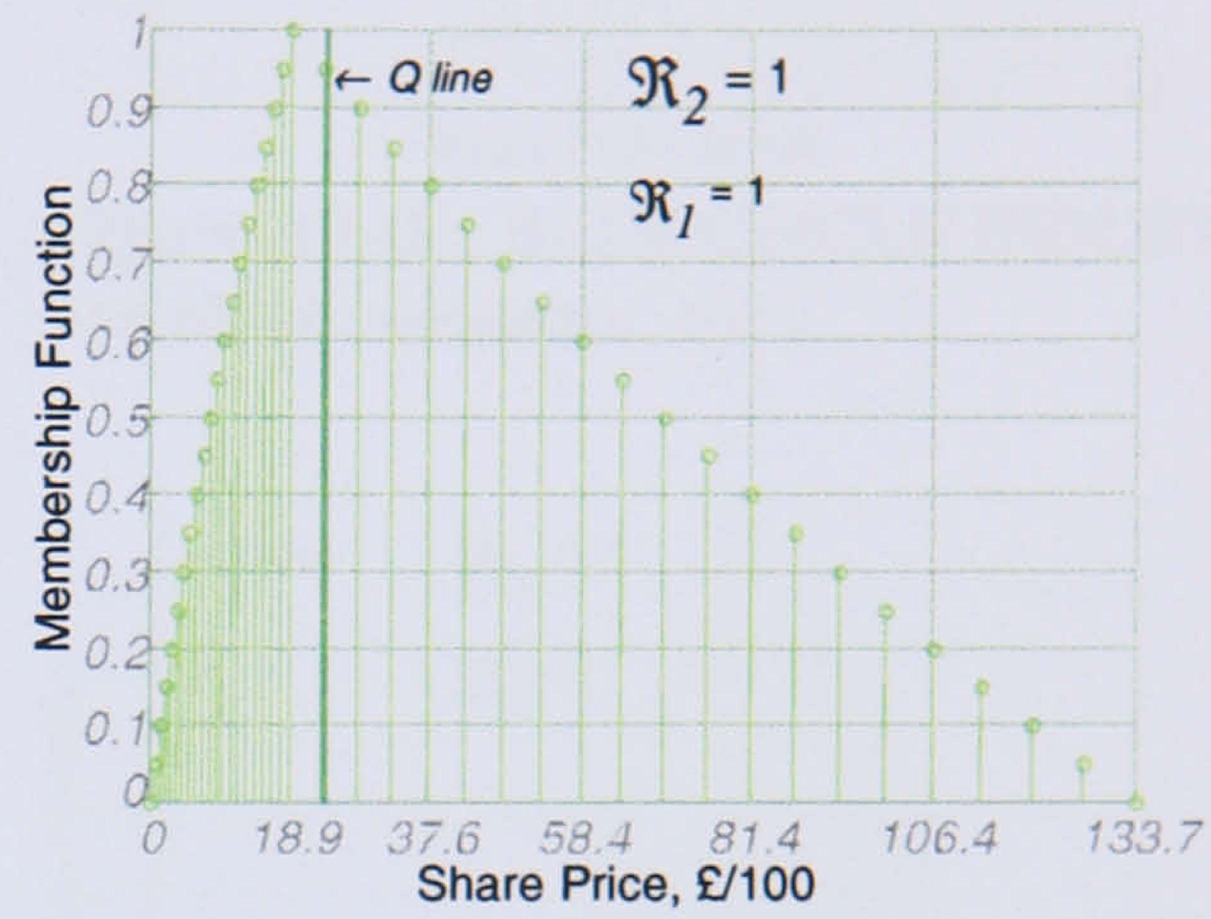


Figure A3.1b: BASS - evaluated robustness $\Delta = 0.952$

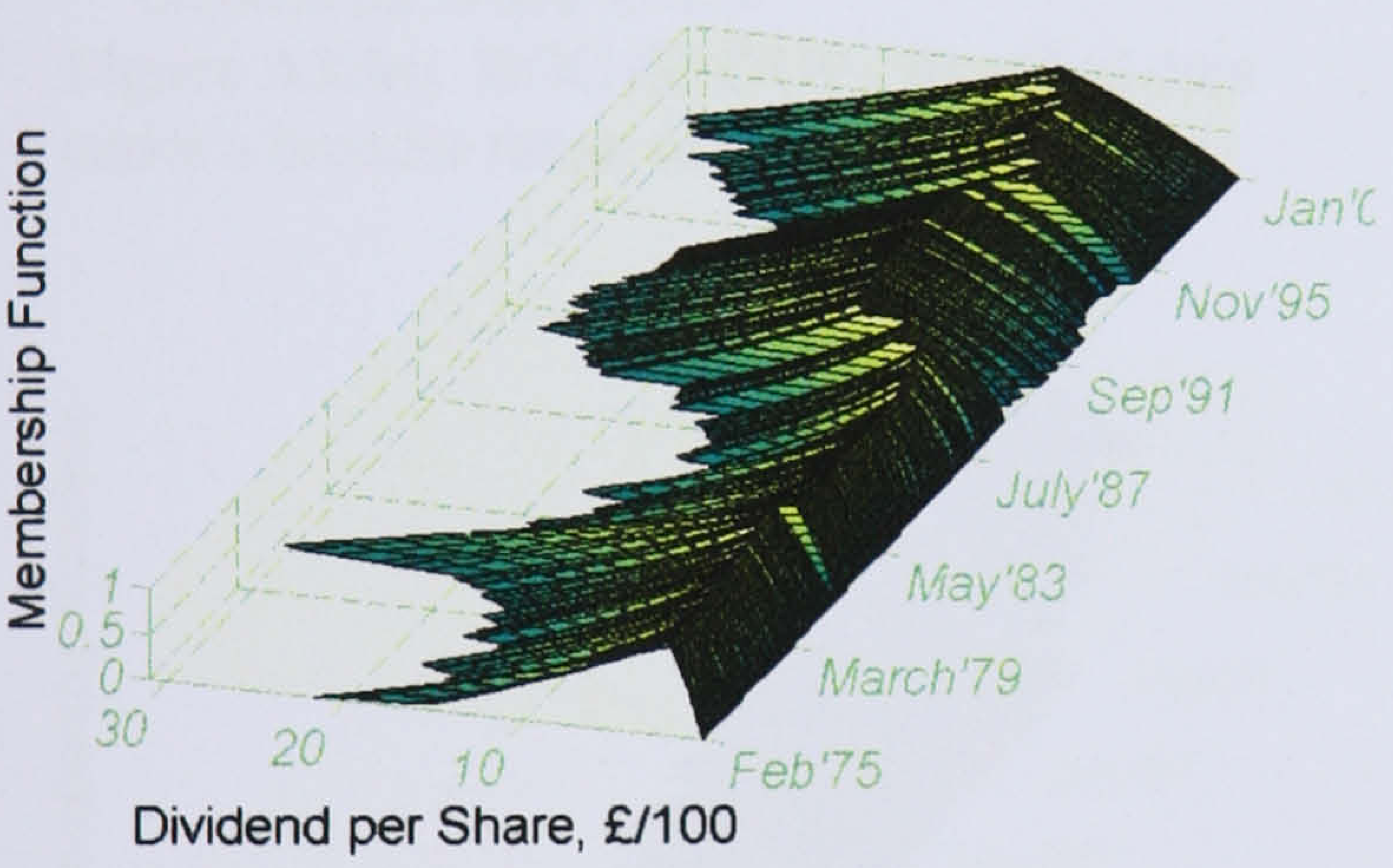


Figure A3.2a: BBA GROUP - fuzzified data under a broader range of imprecision

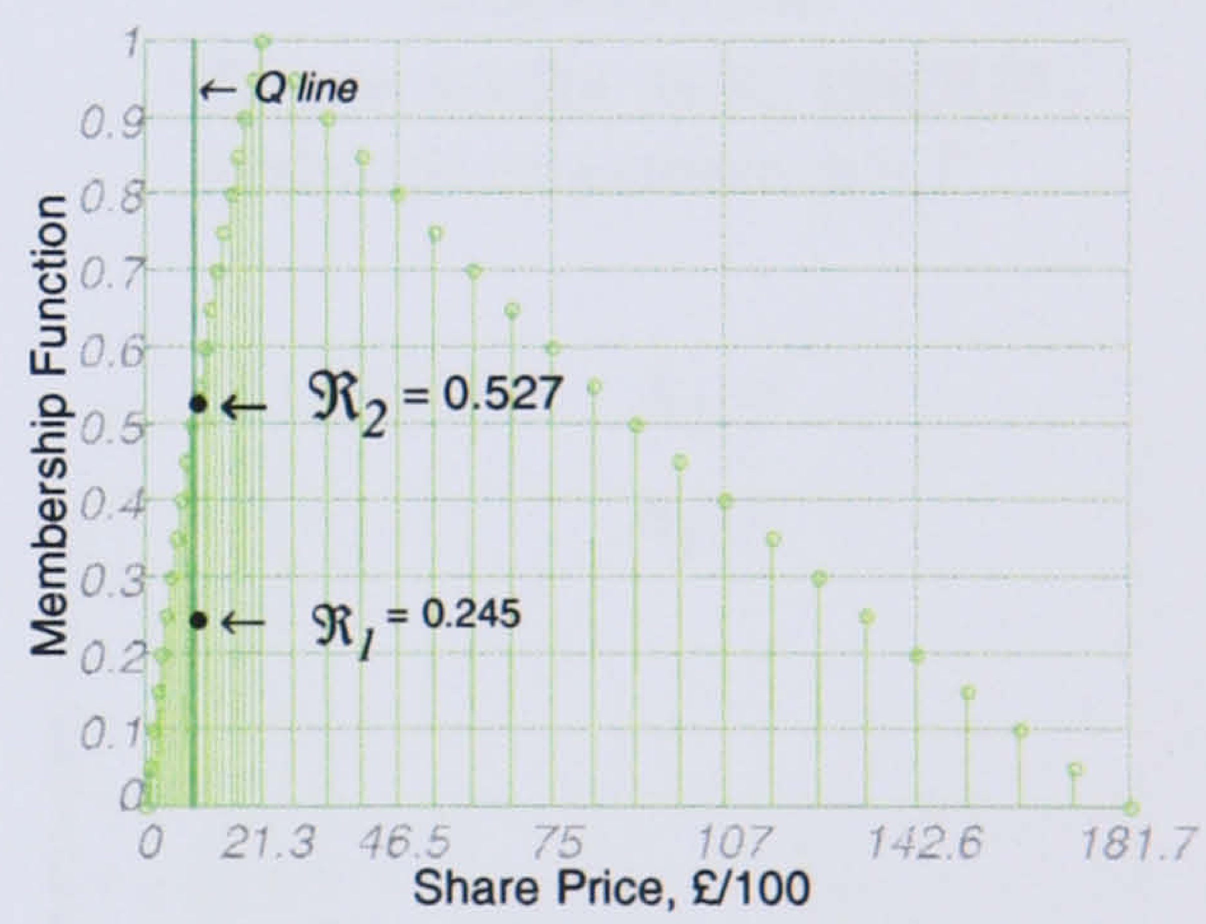


Figure A3.2b: BBA GROUP - evaluated robustness $\Delta = 0.718$

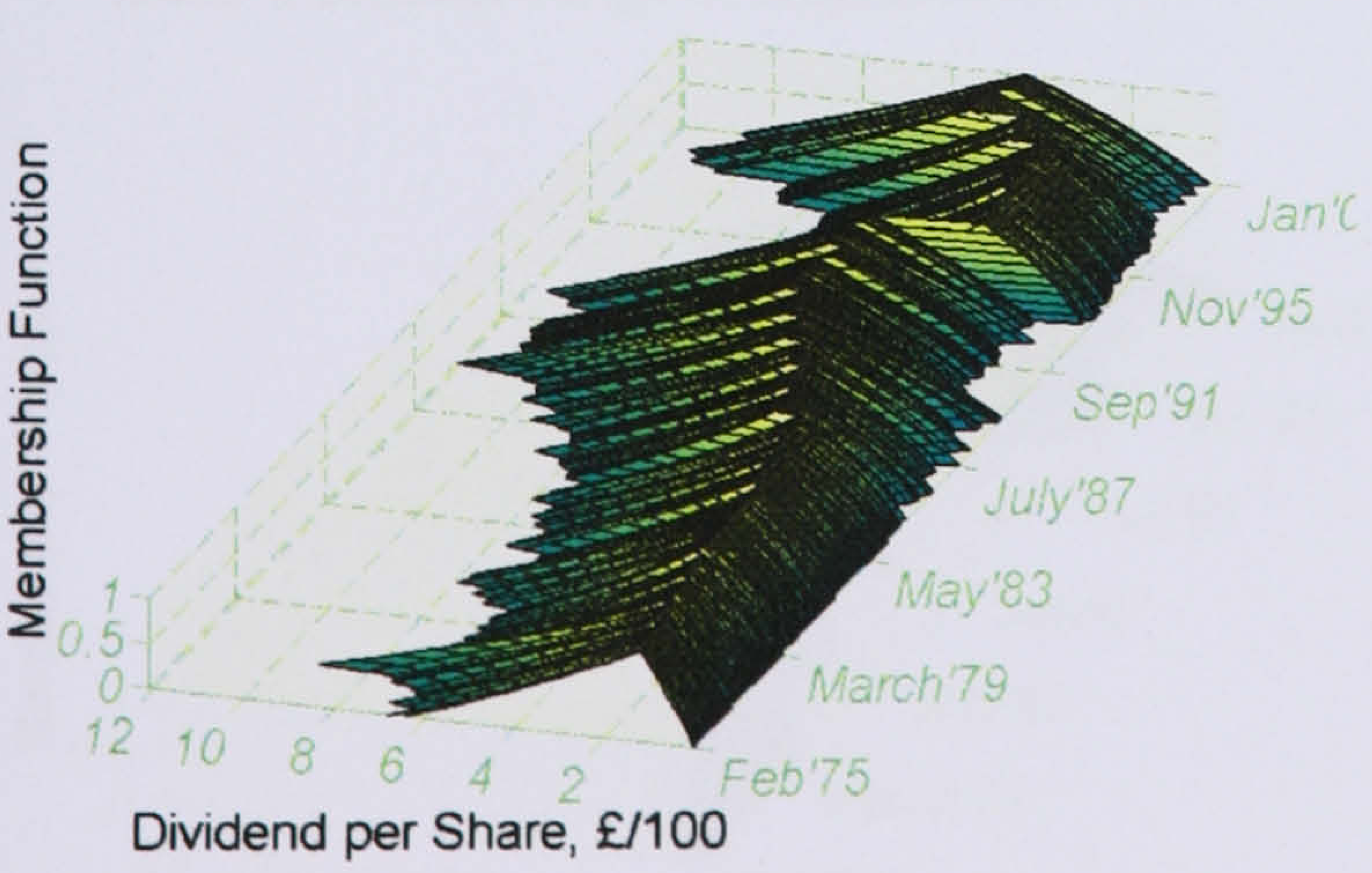


Figure A3.3a: BENTALLS - fuzzified data under a broader range of imprecision

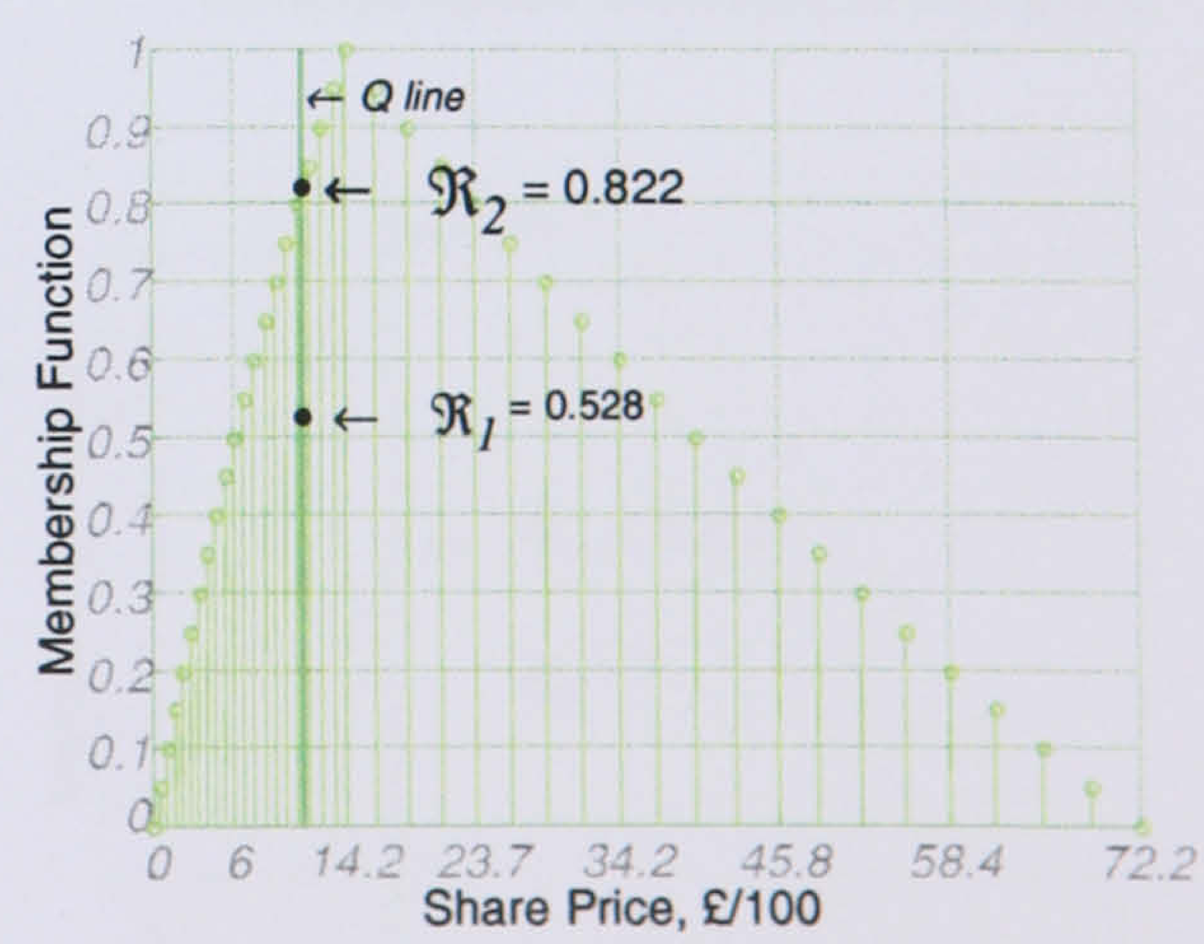


Figure A3.3b: BENTALLS - evaluated robustness $\Delta = 0.706$

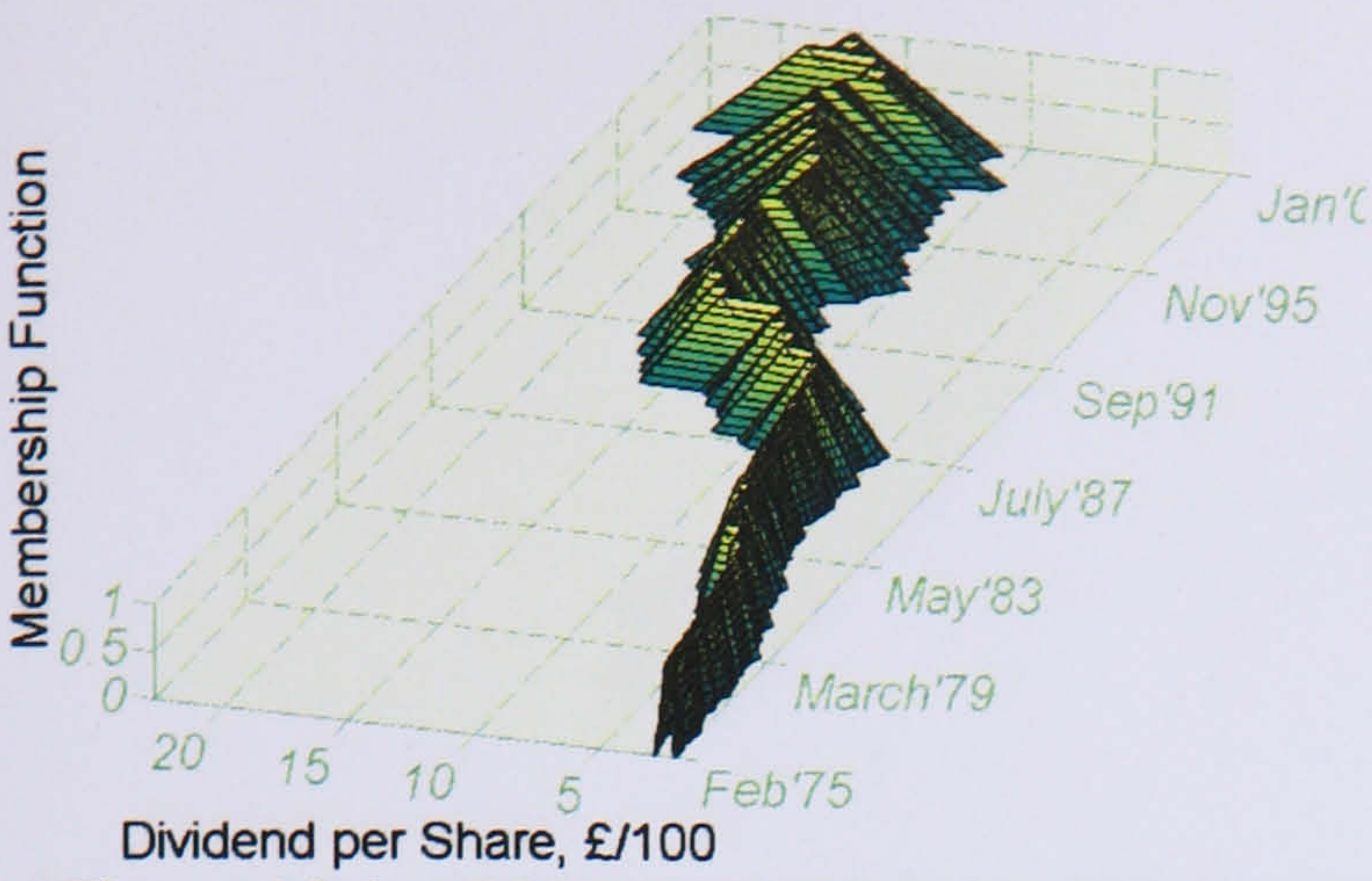


Figure A3.4a: BLUE CIRCLE INDUSTRIES - fuzzified data under a broader range of imprecision

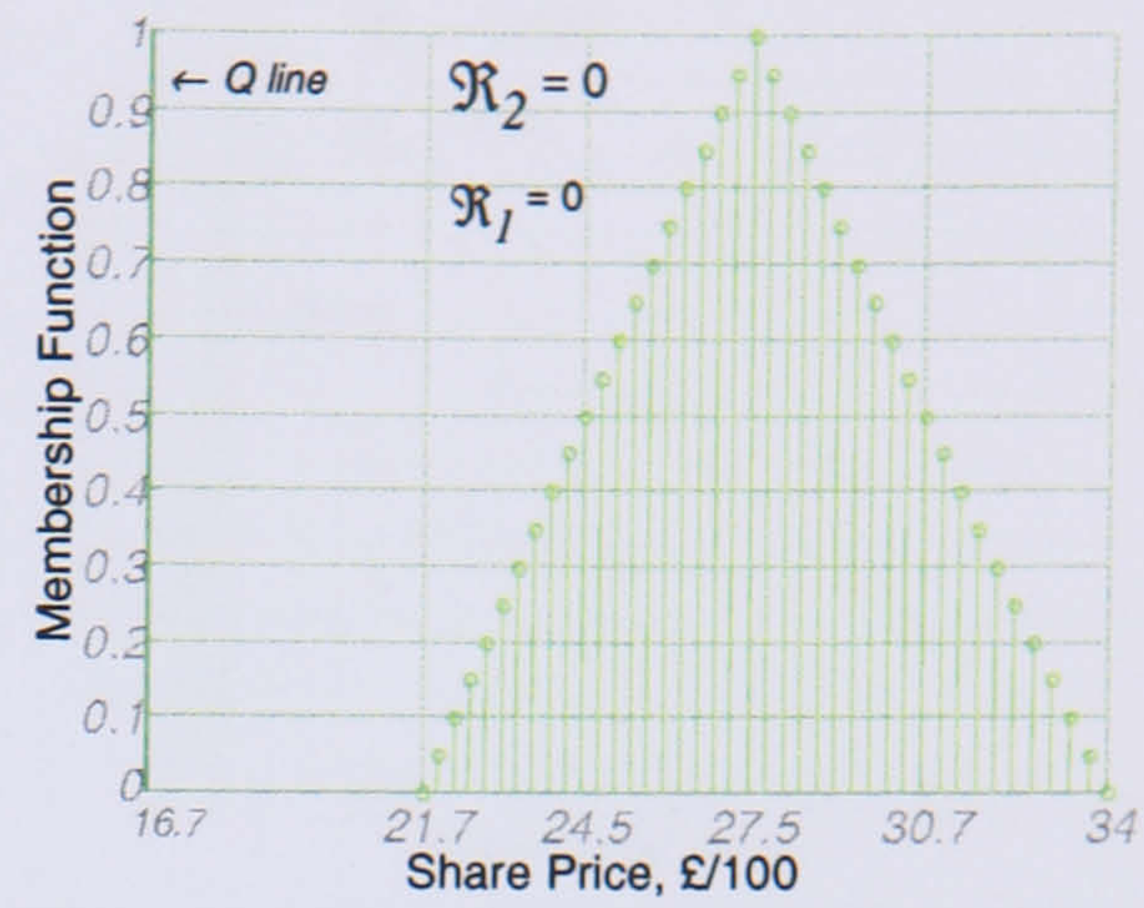


Figure A3.4b: BLUE CIRCLE INDUSTRIES - evaluated robustness $\Delta = 1$

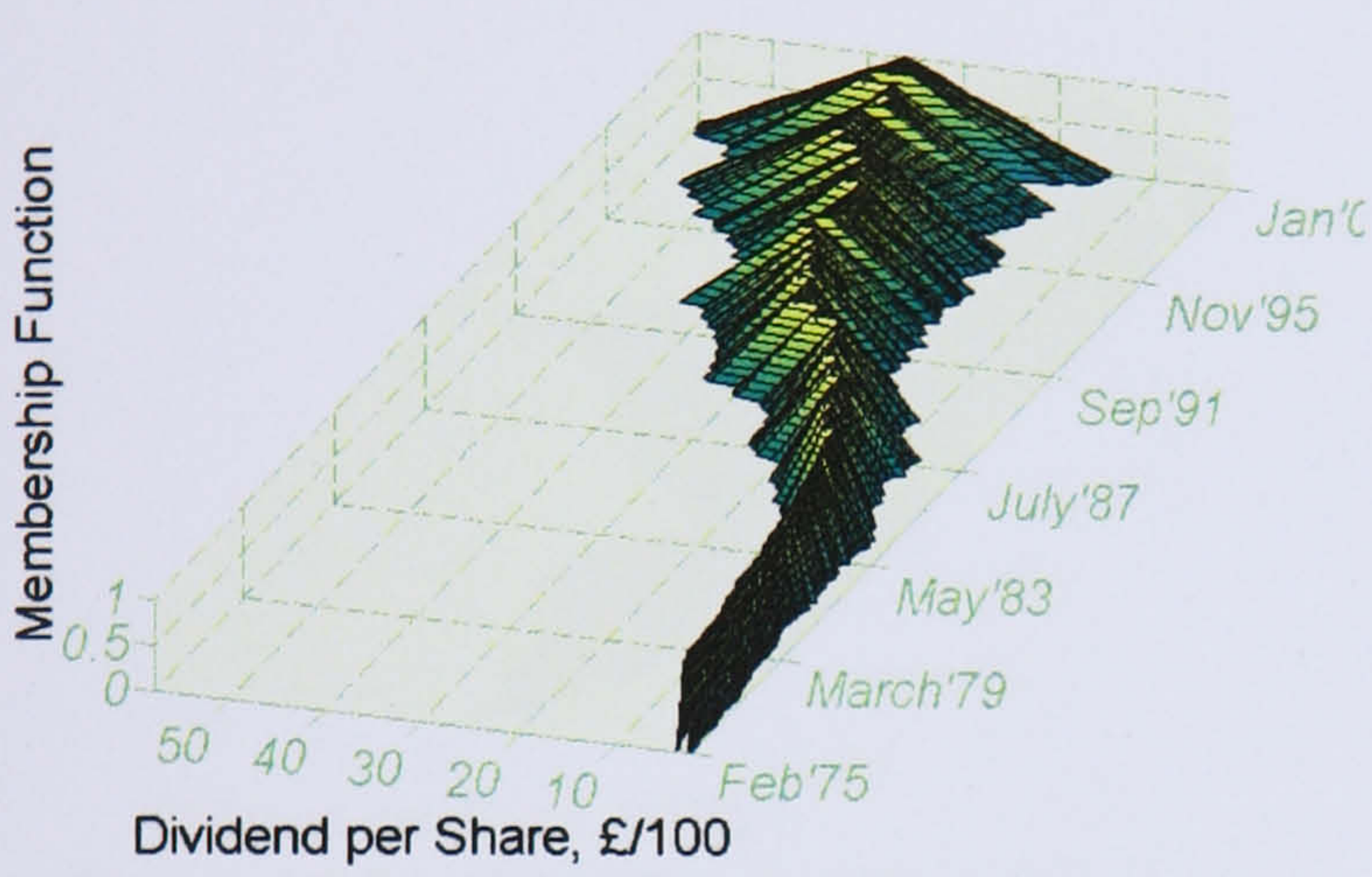


Figure A3.5a: BOC GROUP - fuzzified data under a broader range of imprecision

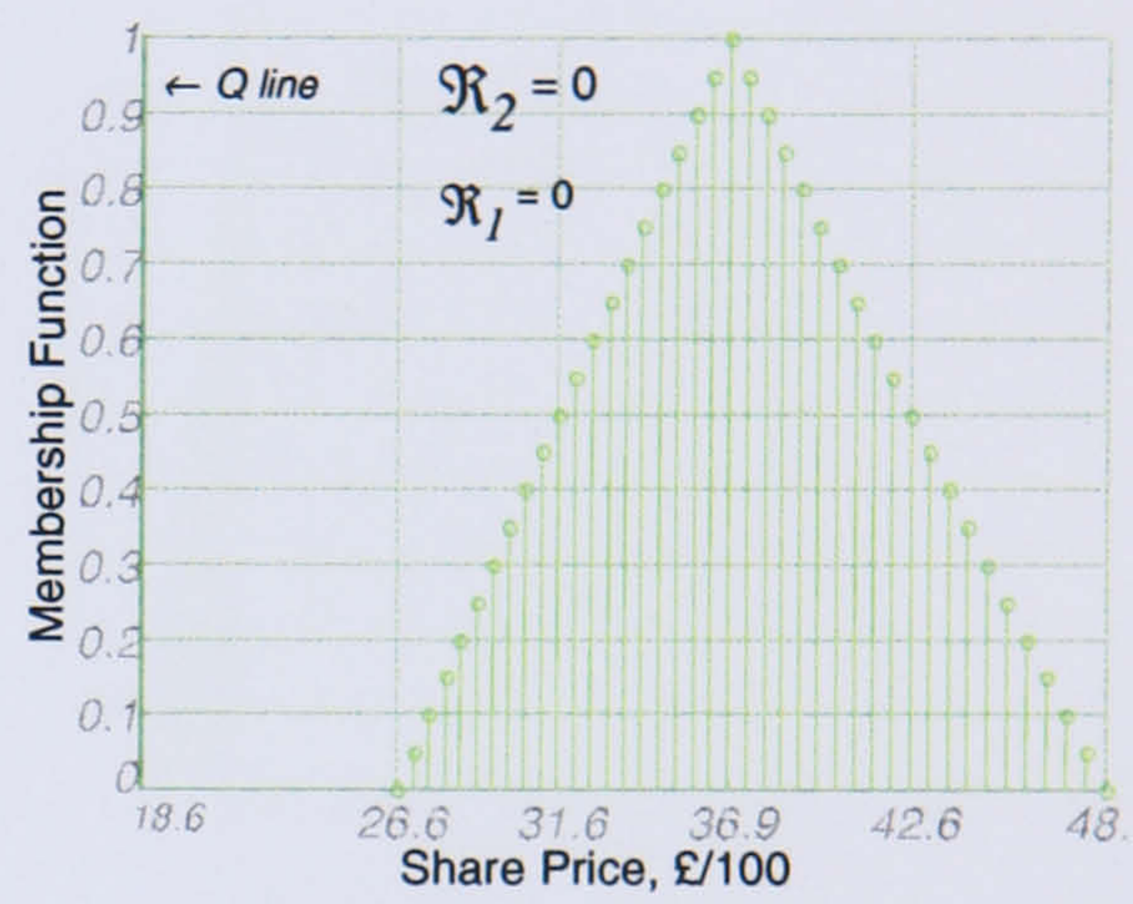


Figure A3.5b: BOC GROUP - evaluated robustness $\Delta = 1$

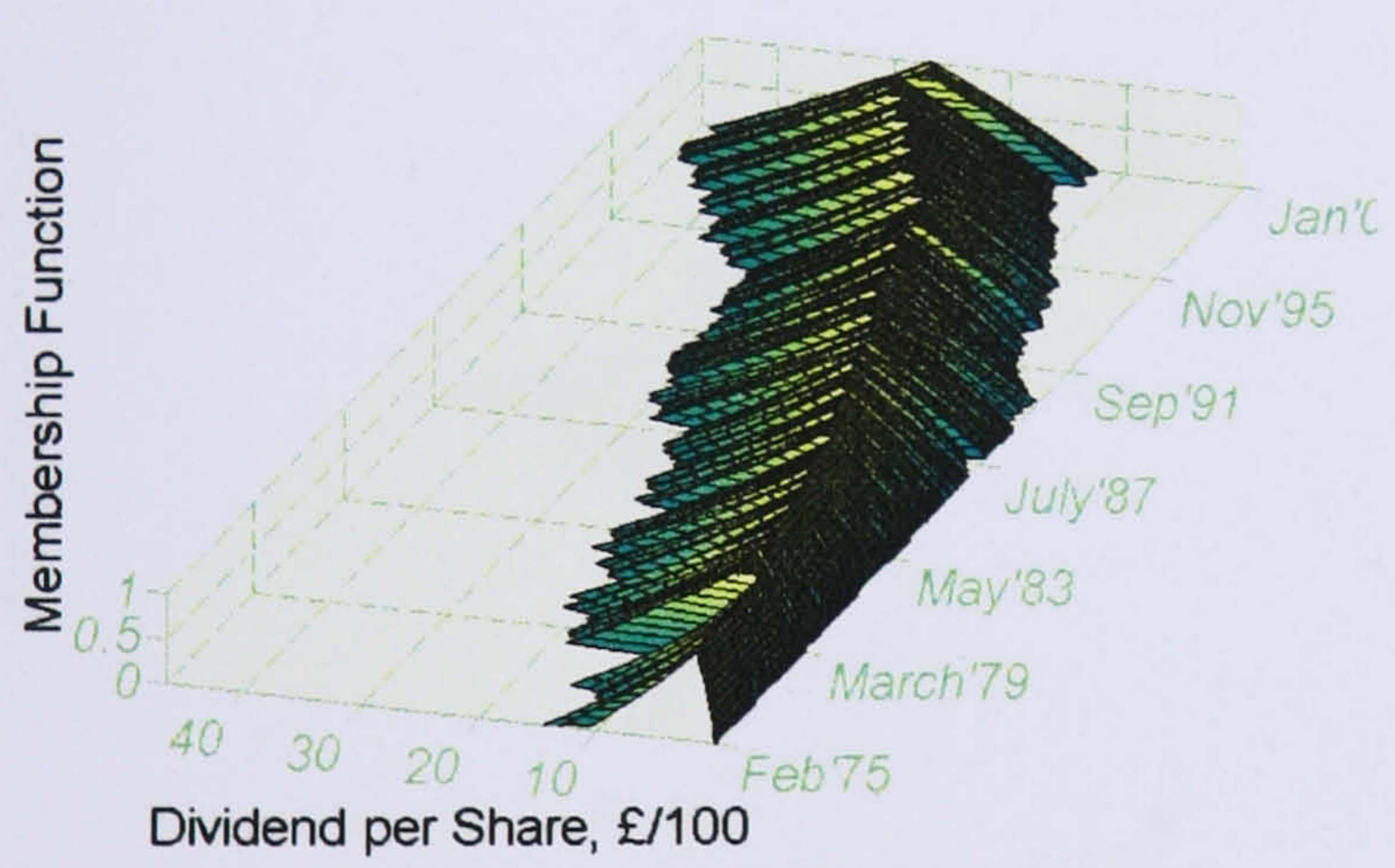


Figure A3.6a: BOOTS CO. - fuzzified data under a broader range of imprecision

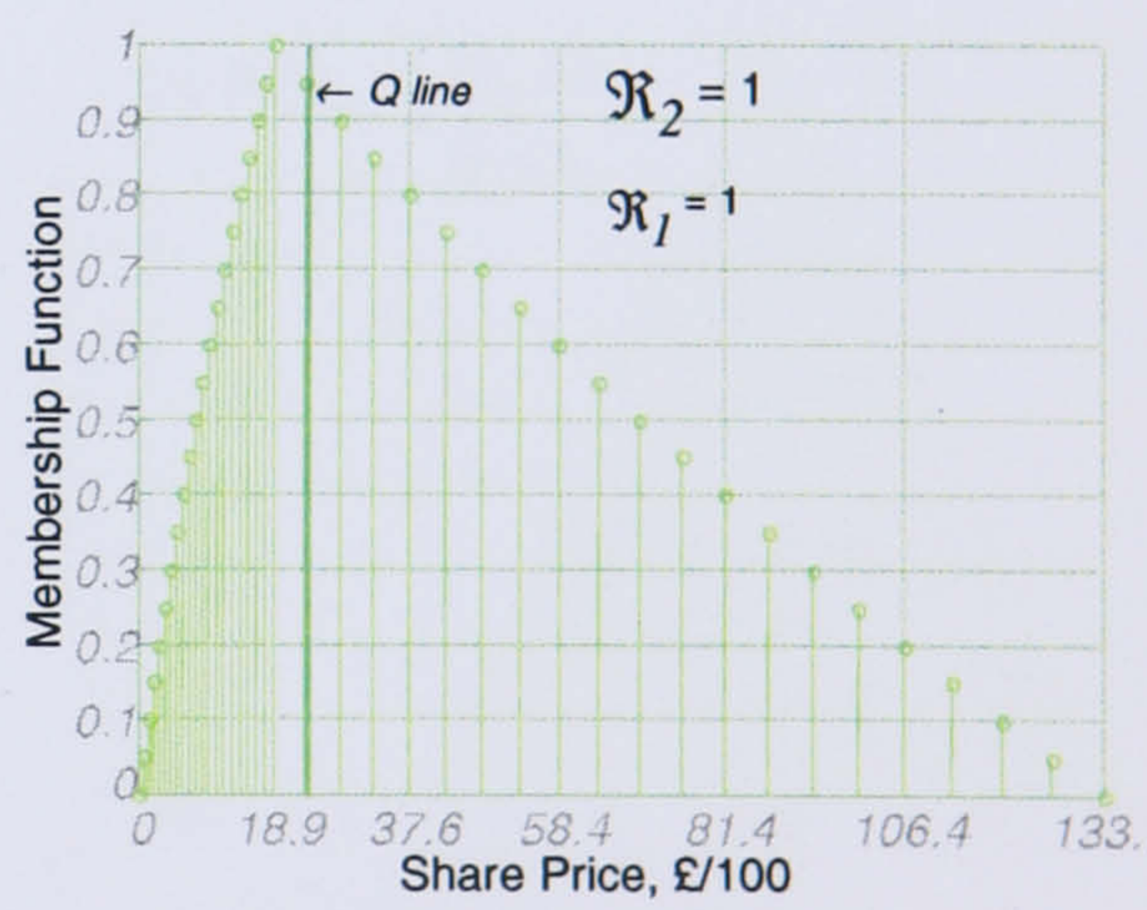


Figure A3.6b: BOOTS CO. - no robustness measure is assigned

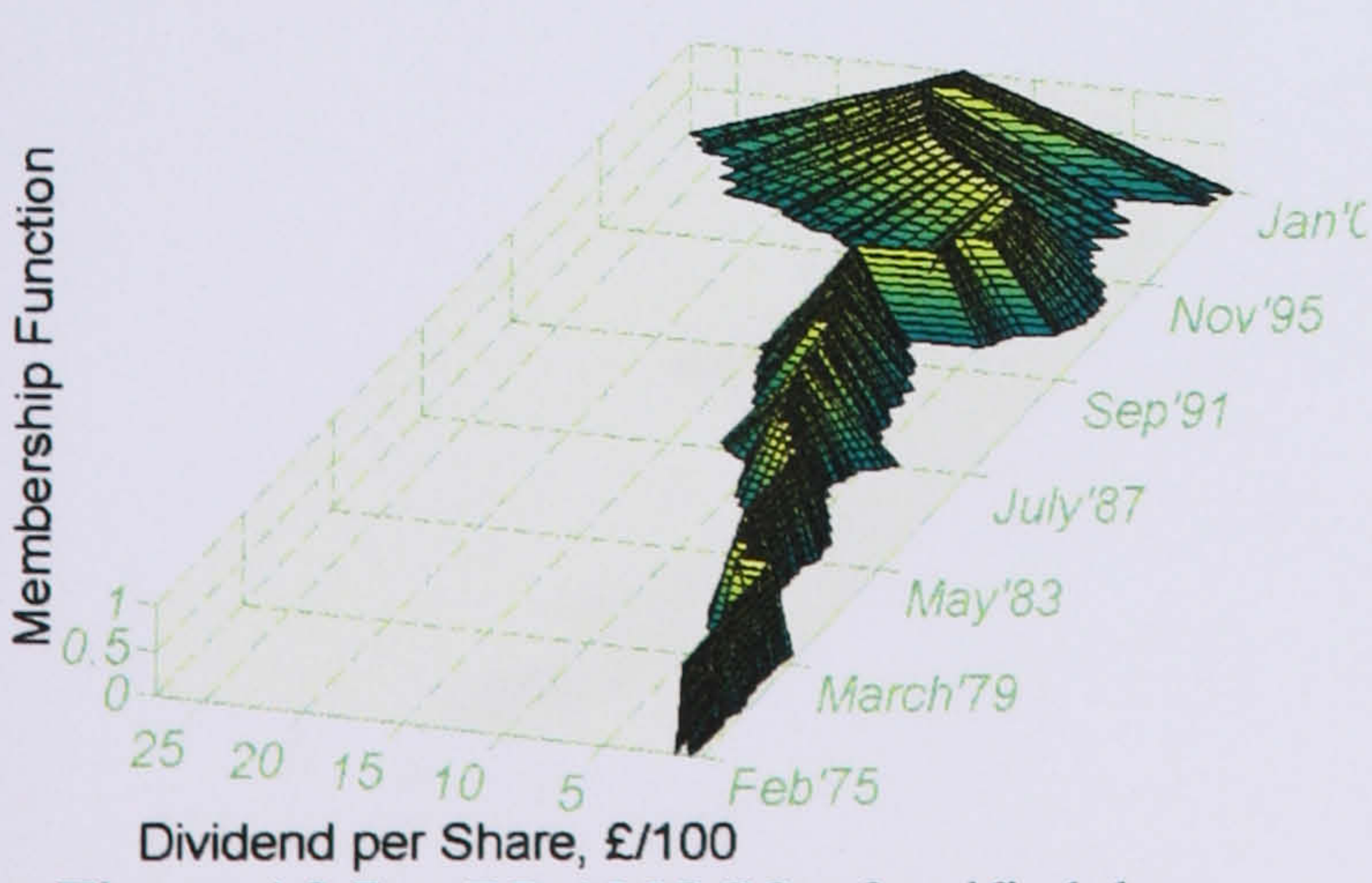


Figure A3.7a: BP AMOCO - fuzzified data under a broader range of imprecision

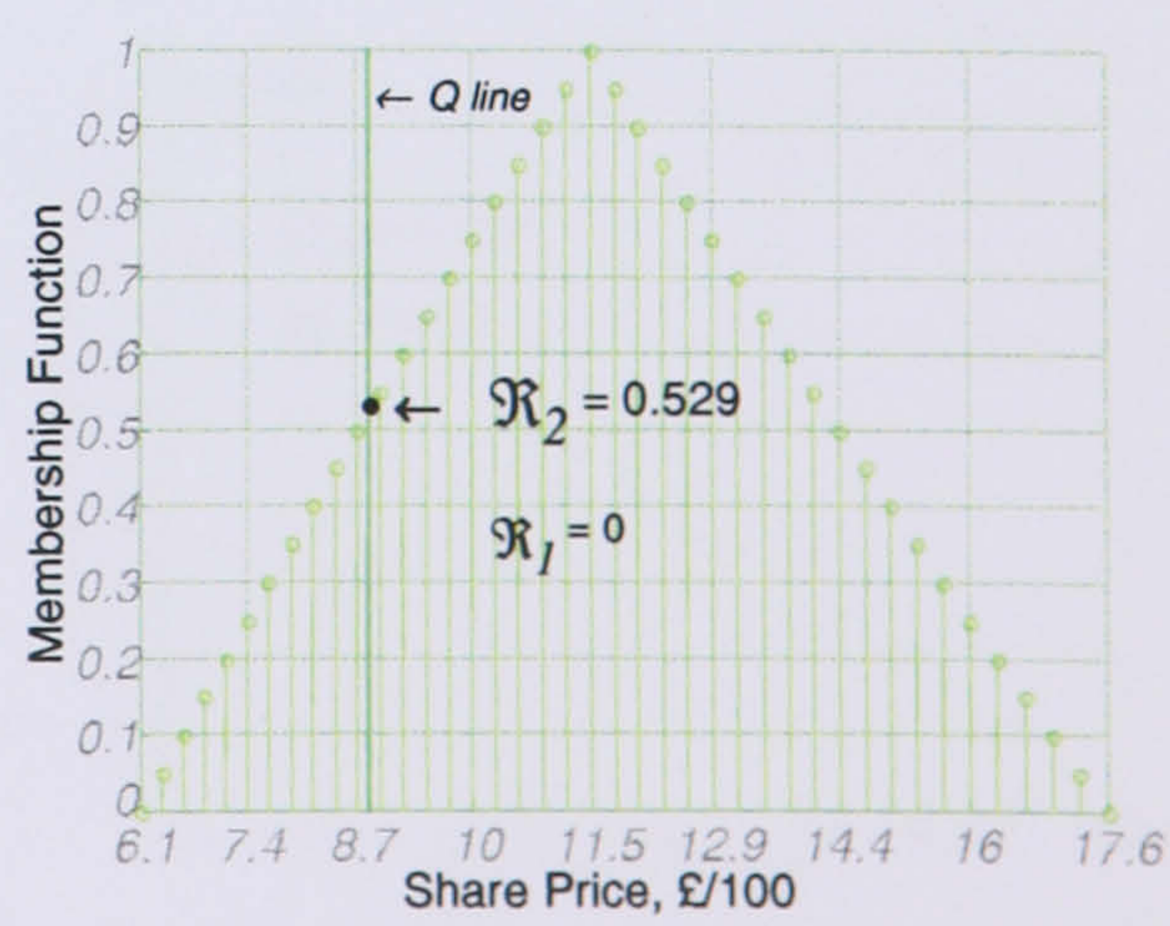


Figure A3.7b: BP AMOCO - evaluated robustness $\Delta = 0.471$

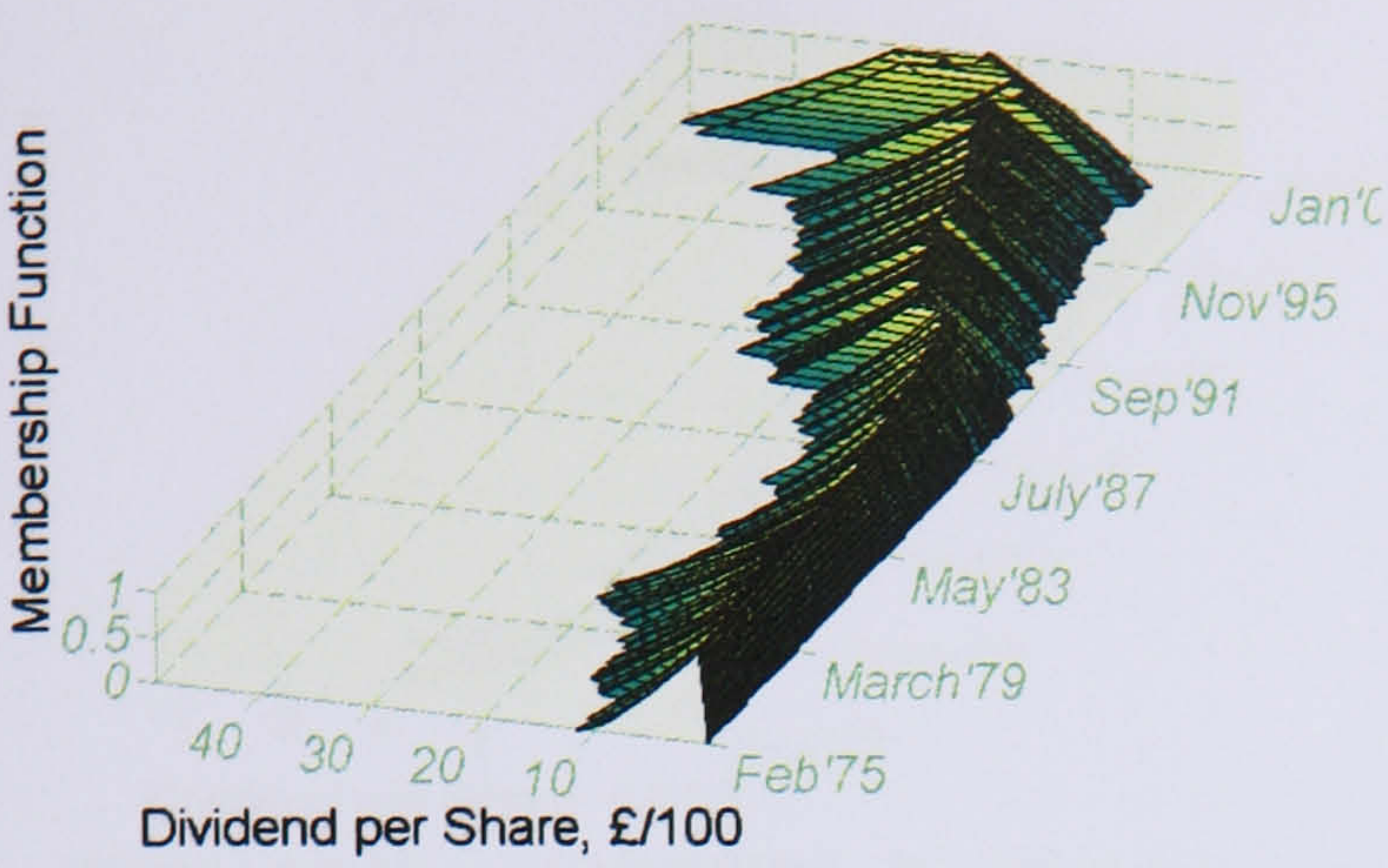


Figure A3.8a: BRITISH AMERICAN TOBACCO - fuzzified data under a broader range of imprecision

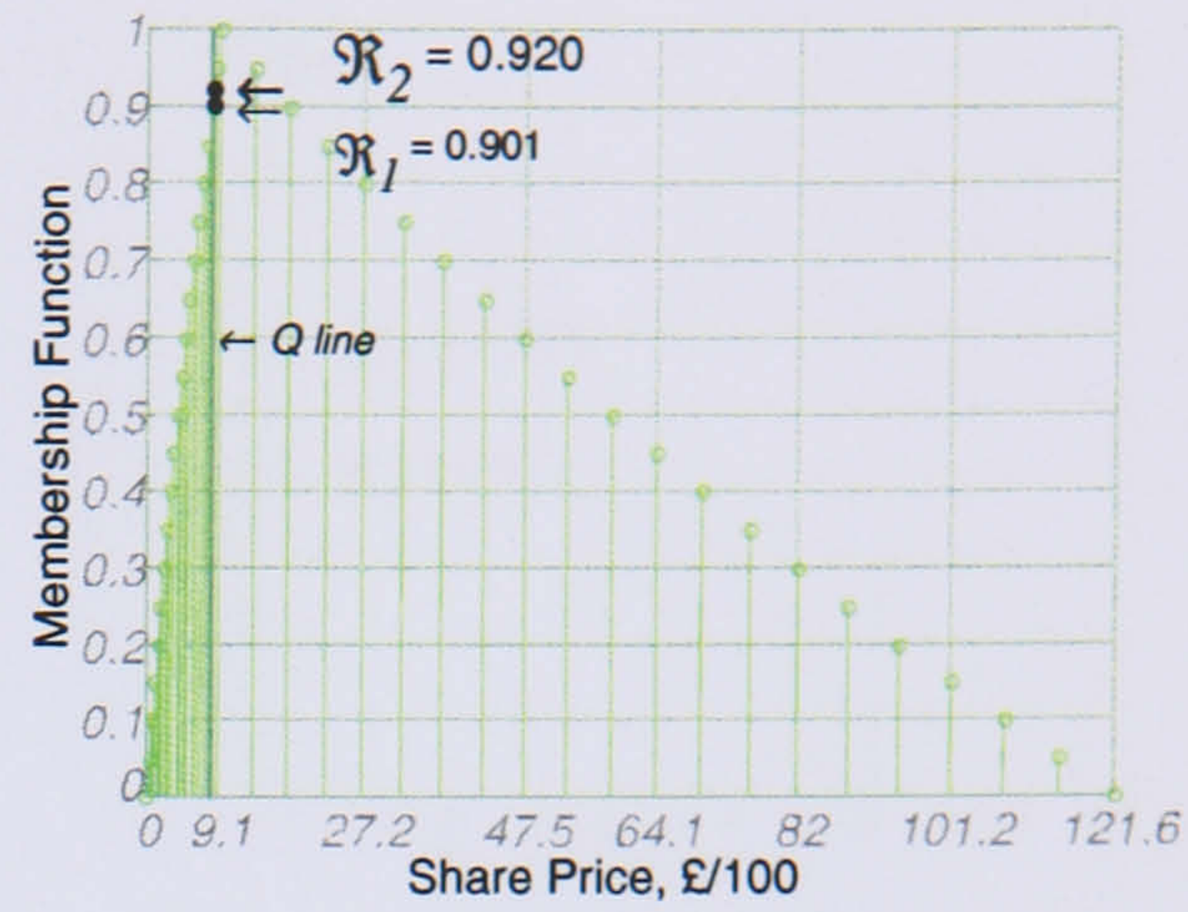


Figure A3.8b: BRITISH AMERICAN TOBACCO evaluated robustness $\Delta = 0.981$

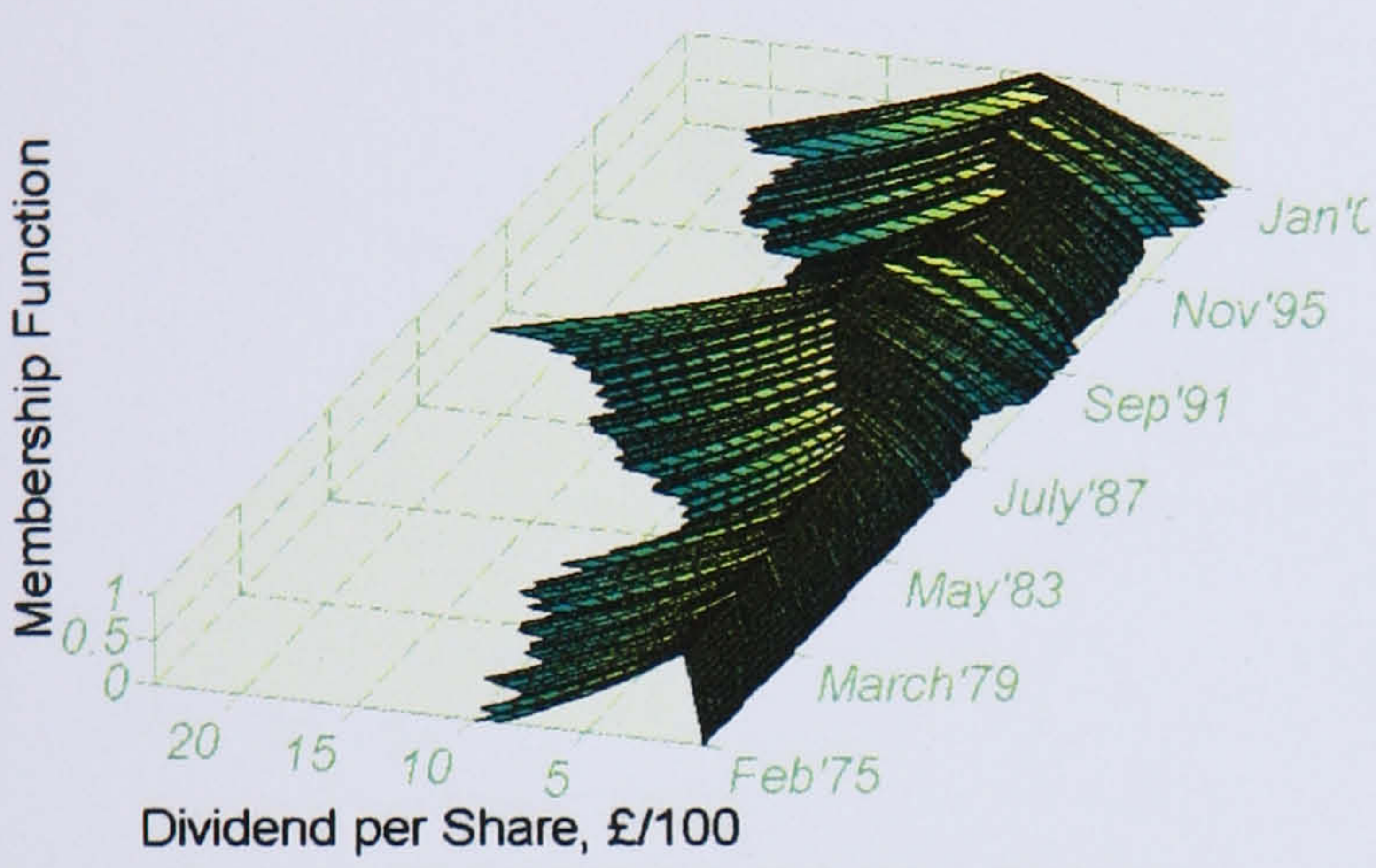


Figure A3.9a: BUNZL - fuzzified data under a broader range of imprecision

Figure A3.9b: BUNZL - evaluated robustness $\Delta = 0.959$

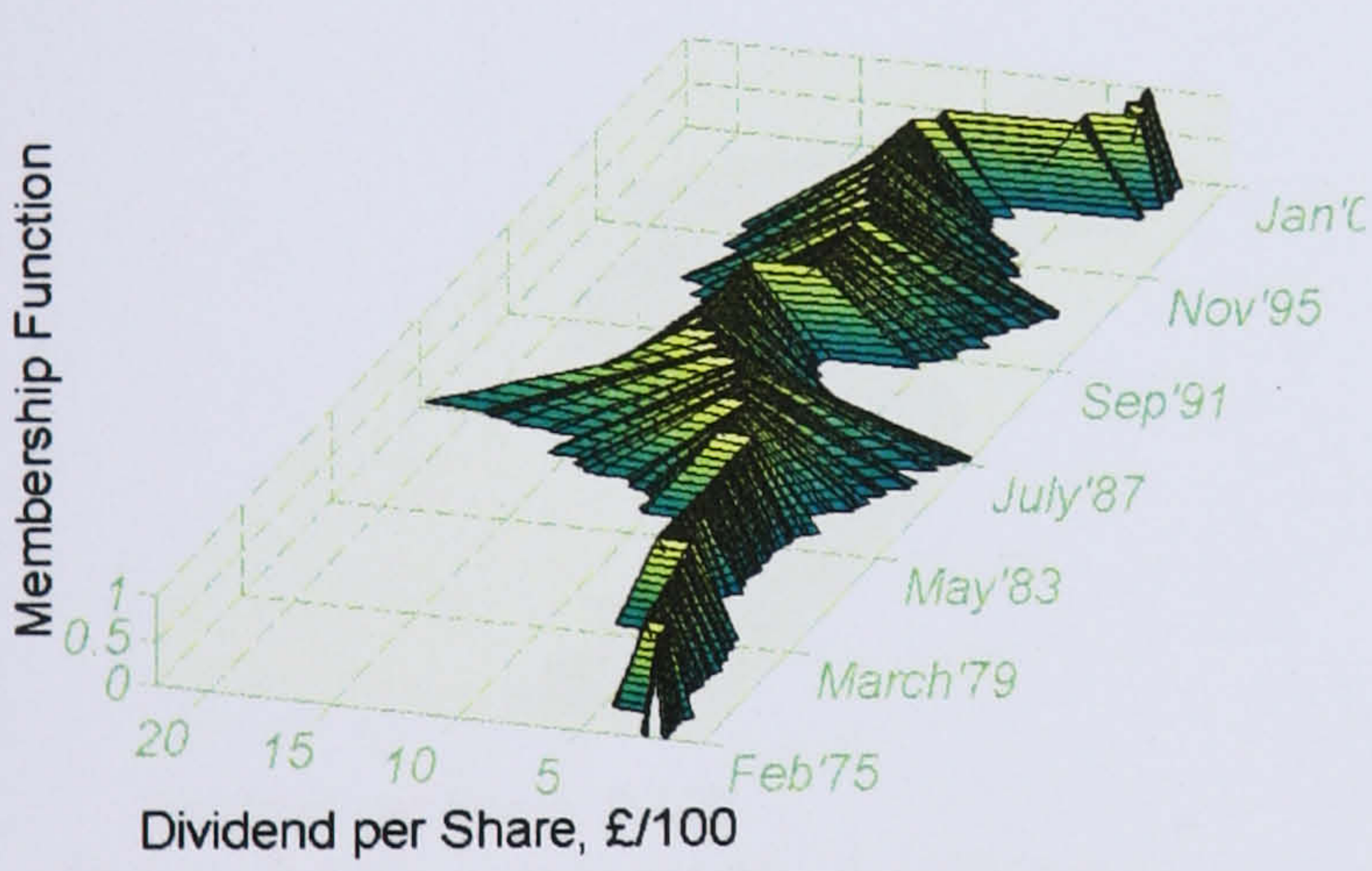


Figure A3.10a: COATS VIYELLA - fuzzified data under a broader range of imprecision

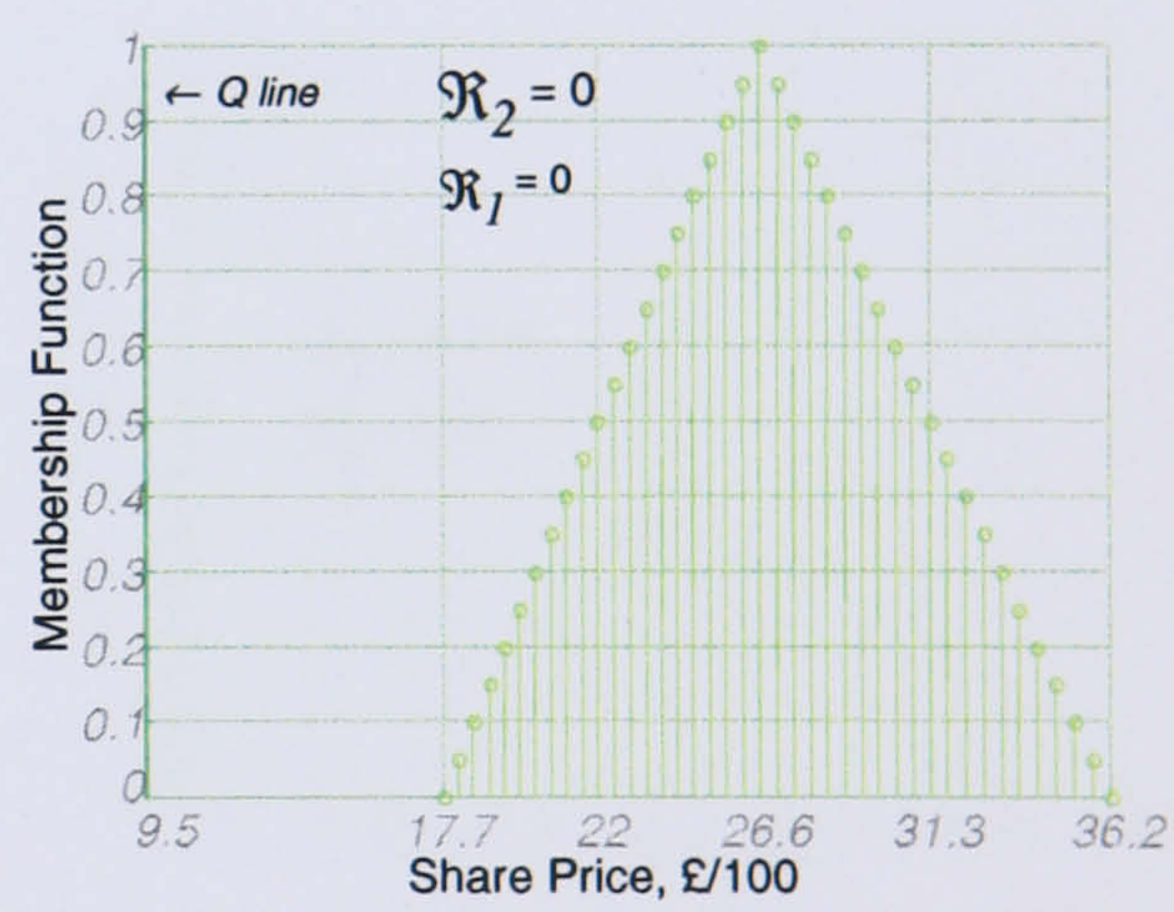


Figure A3.10b: COATS VIYELLA - evaluated robustness $\Delta = 1$

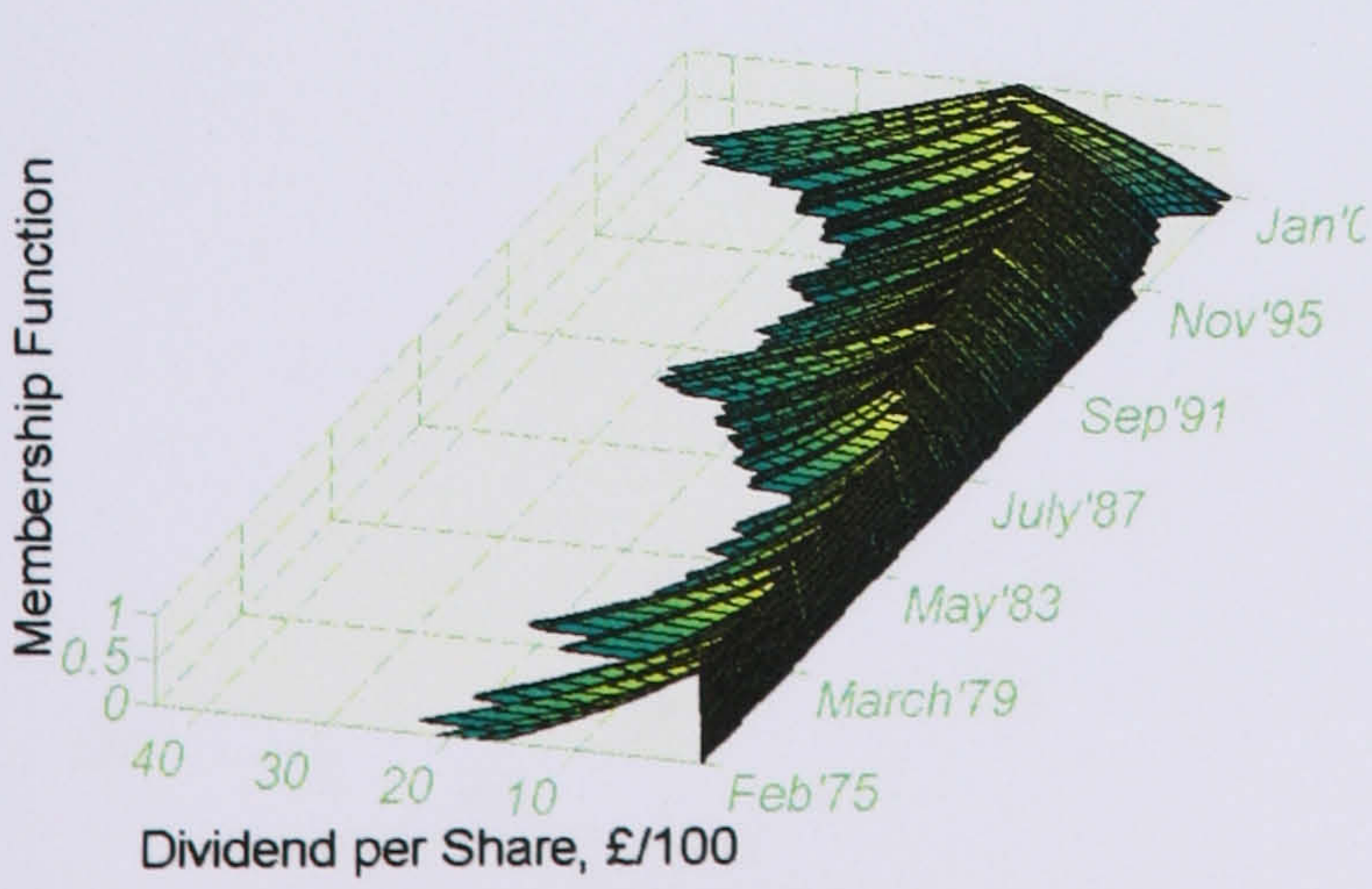


Figure A3.11a: DIXONS GROUP - fuzzified data under a broader range of imprecision

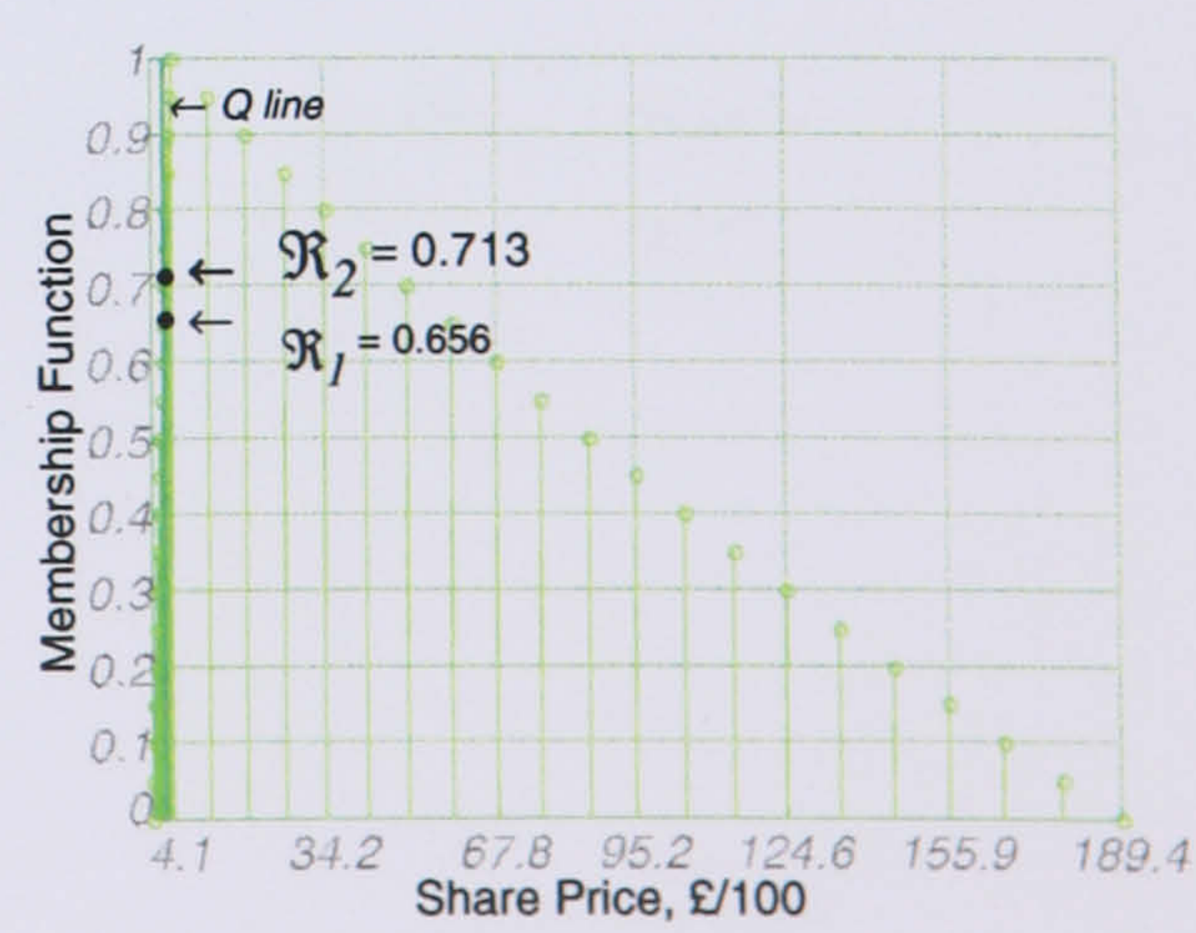


Figure A3.11b: DIXONS GROUP - evaluated robustness $\Delta = 0.943$

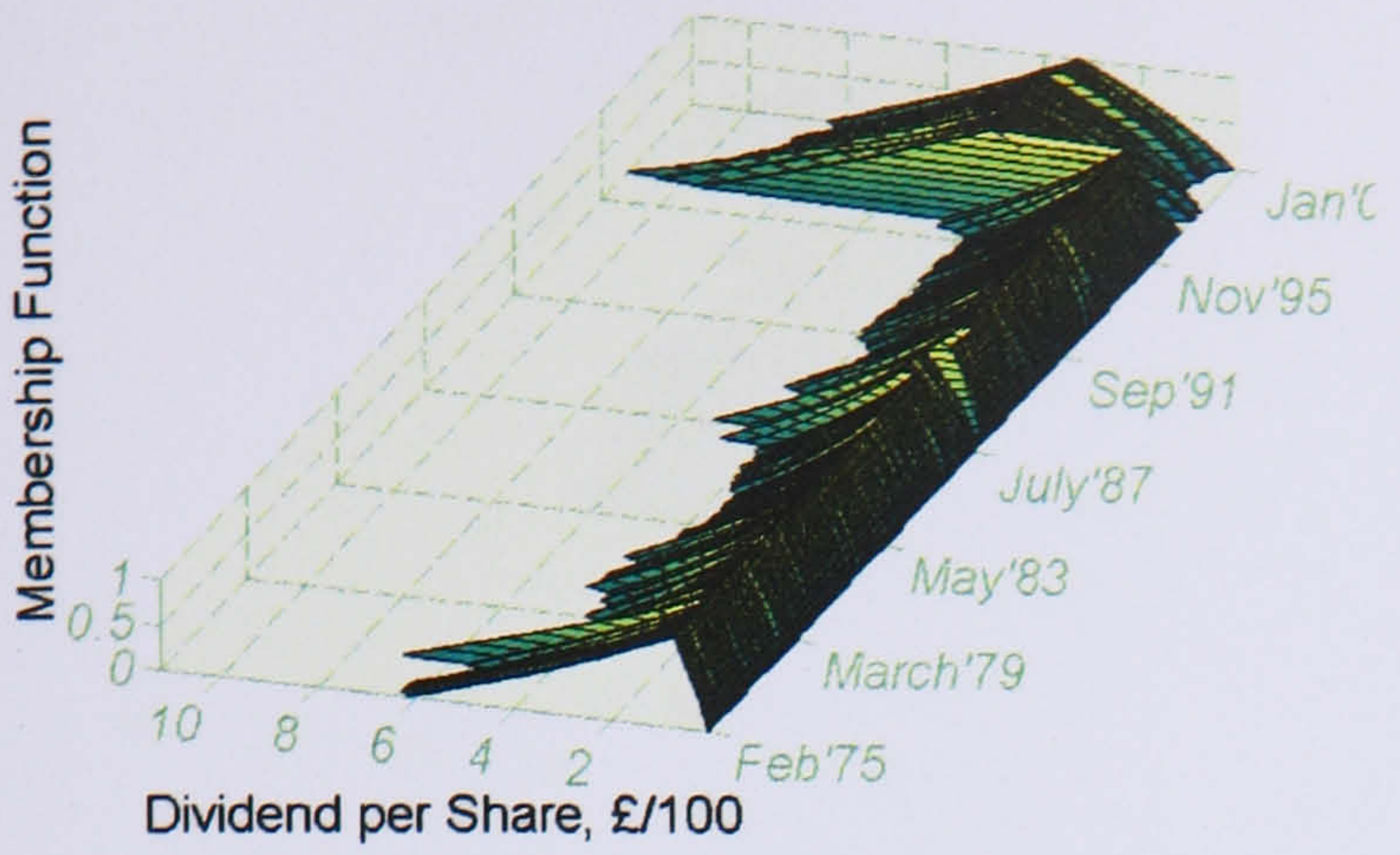


Figure A3.12a: GOODWIN - fuzzified data under a broader range of imprecision

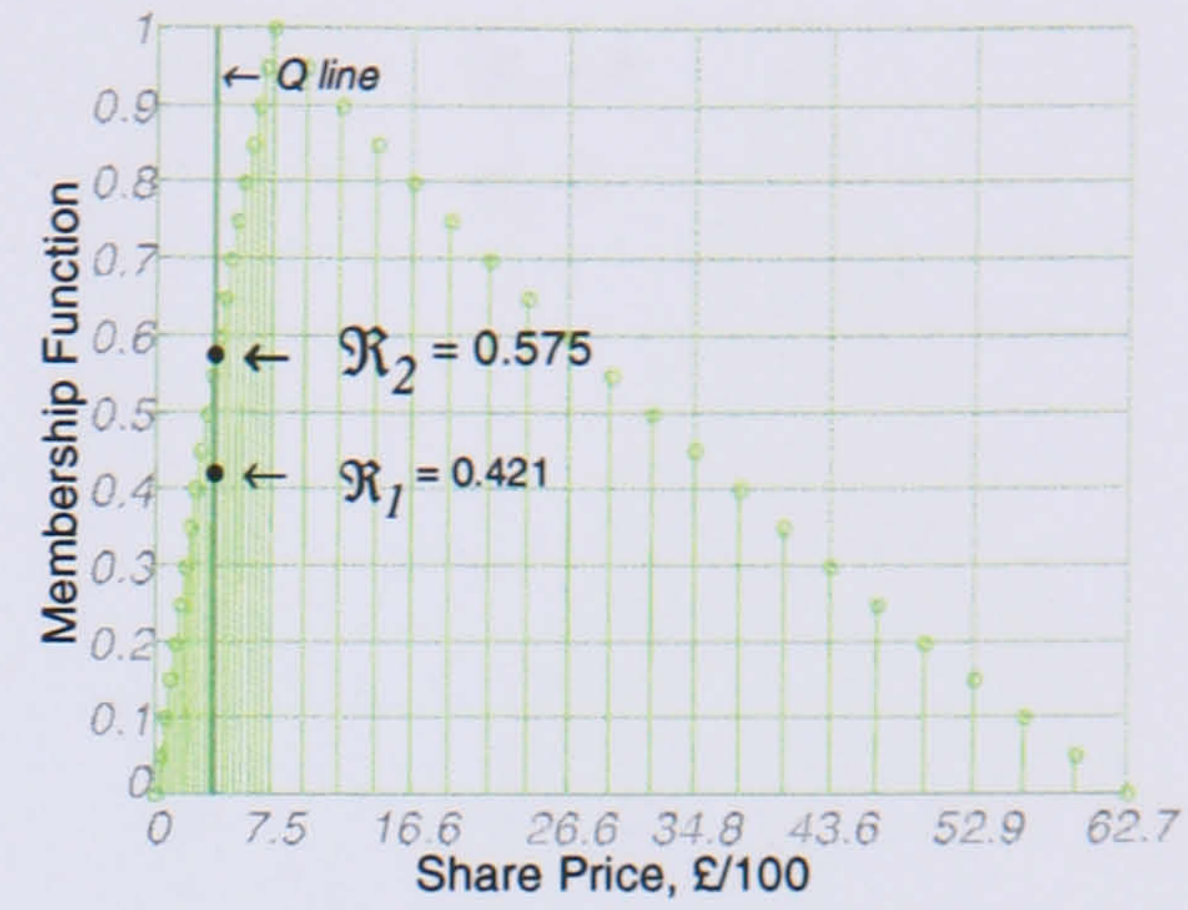


Figure A3.12b: GOODWIN - evaluated robustness $\Delta = 0.846$

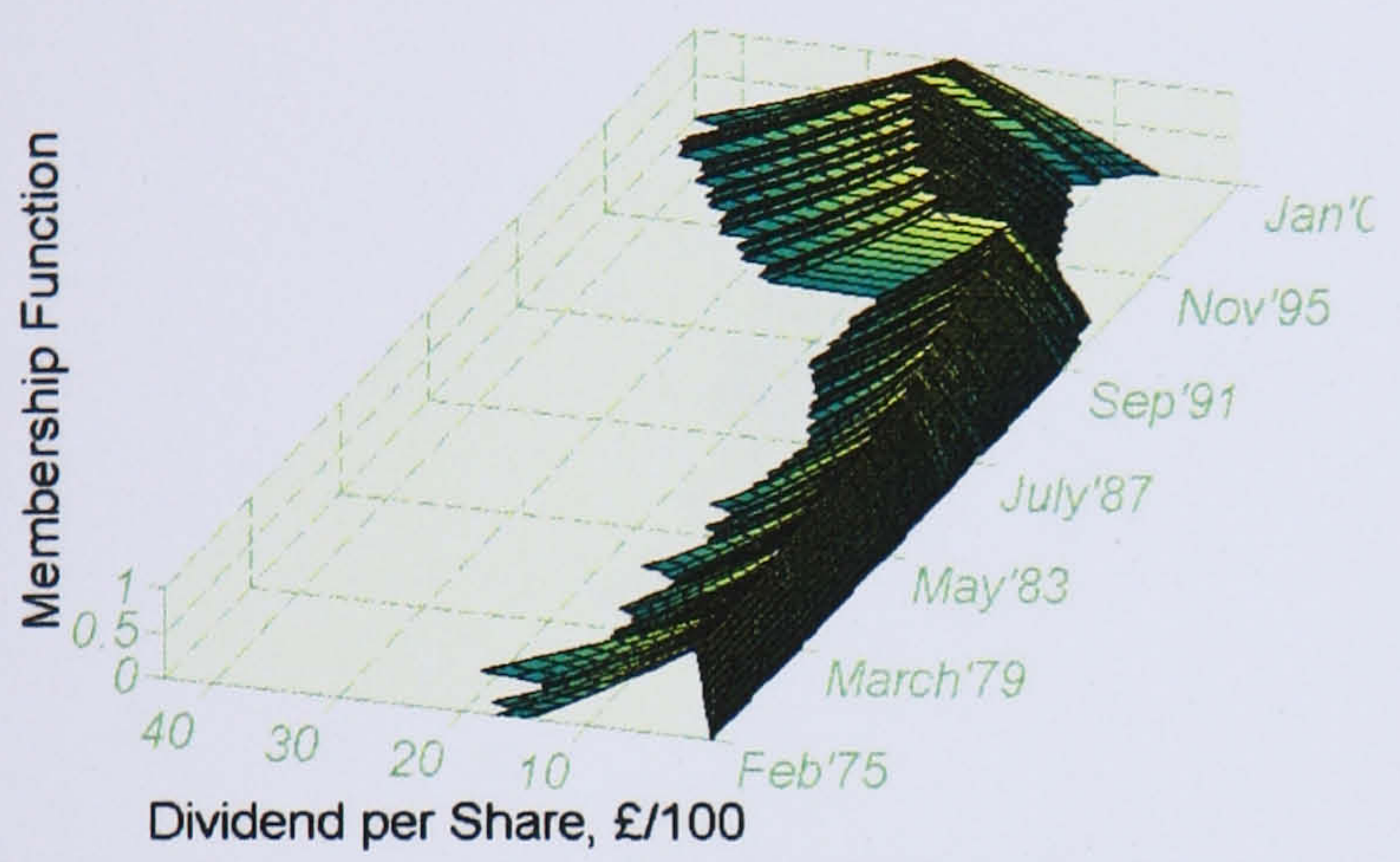


Figure A3.13a: GREAT UNIVERSAL STORES - fuzzified data under a broader range of imprecision

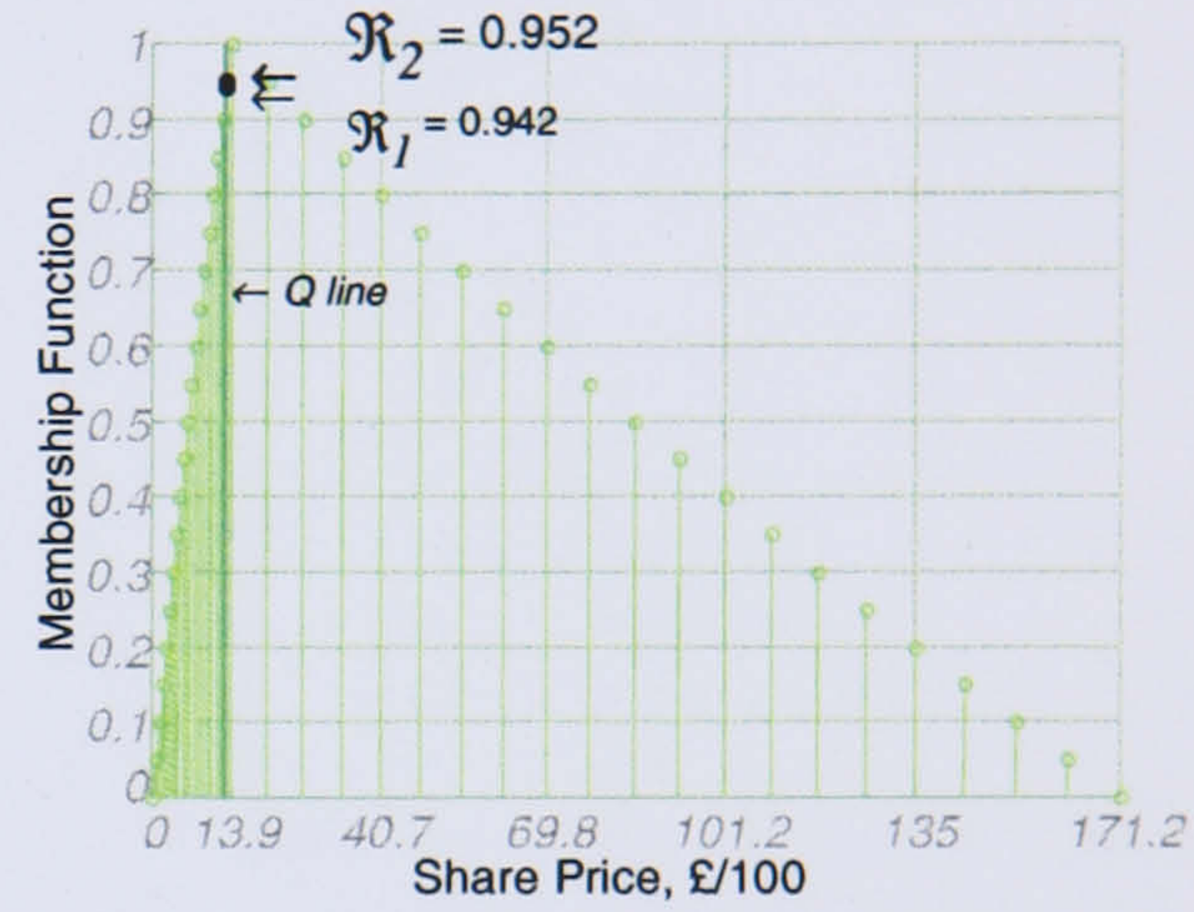


Figure A3.13b: GREAT UNIVERSAL STORES - evaluated robustness $\Delta = 0.990$

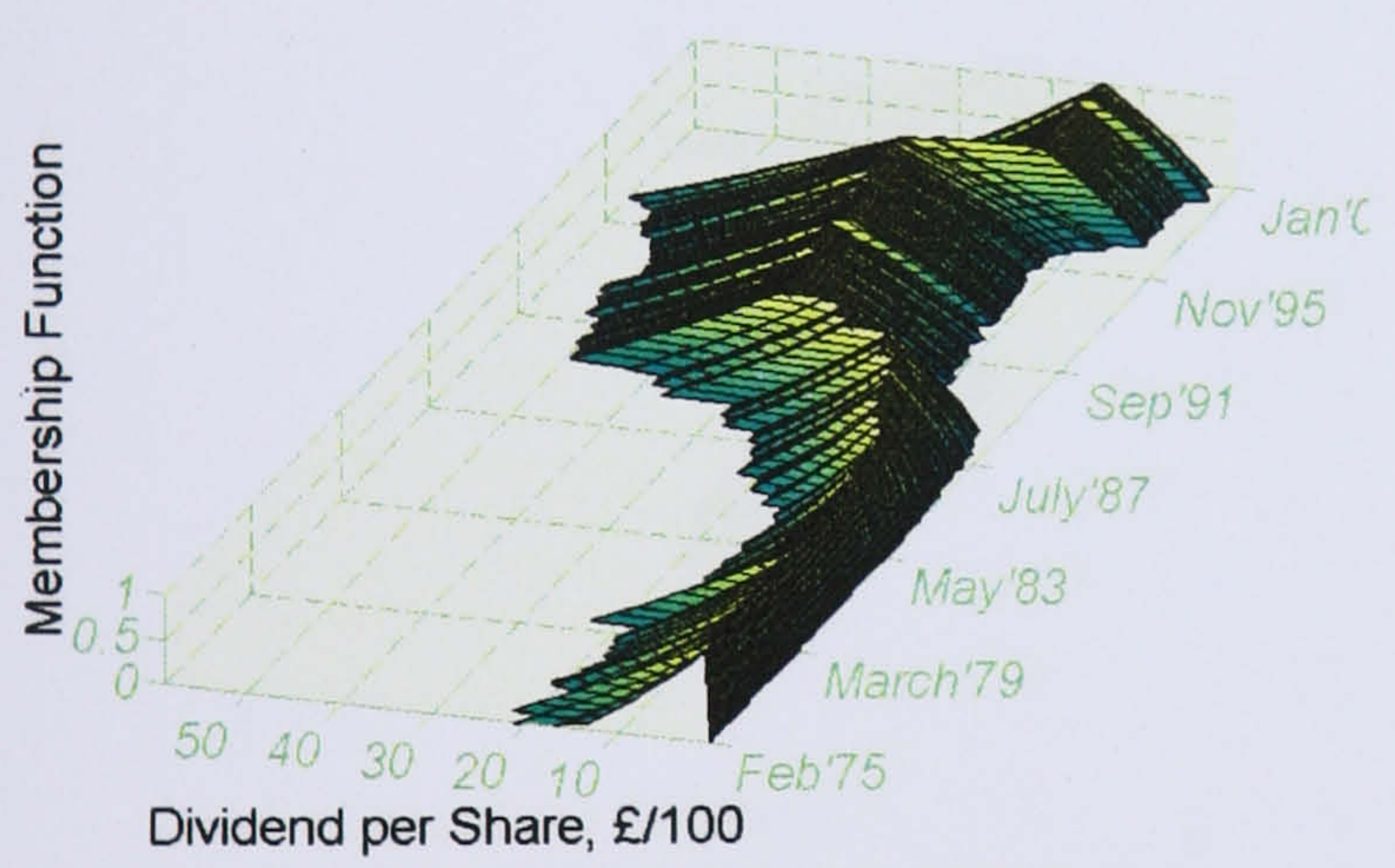


Figure A3.14a: HANSON - fuzzified data under a broader range of imprecision

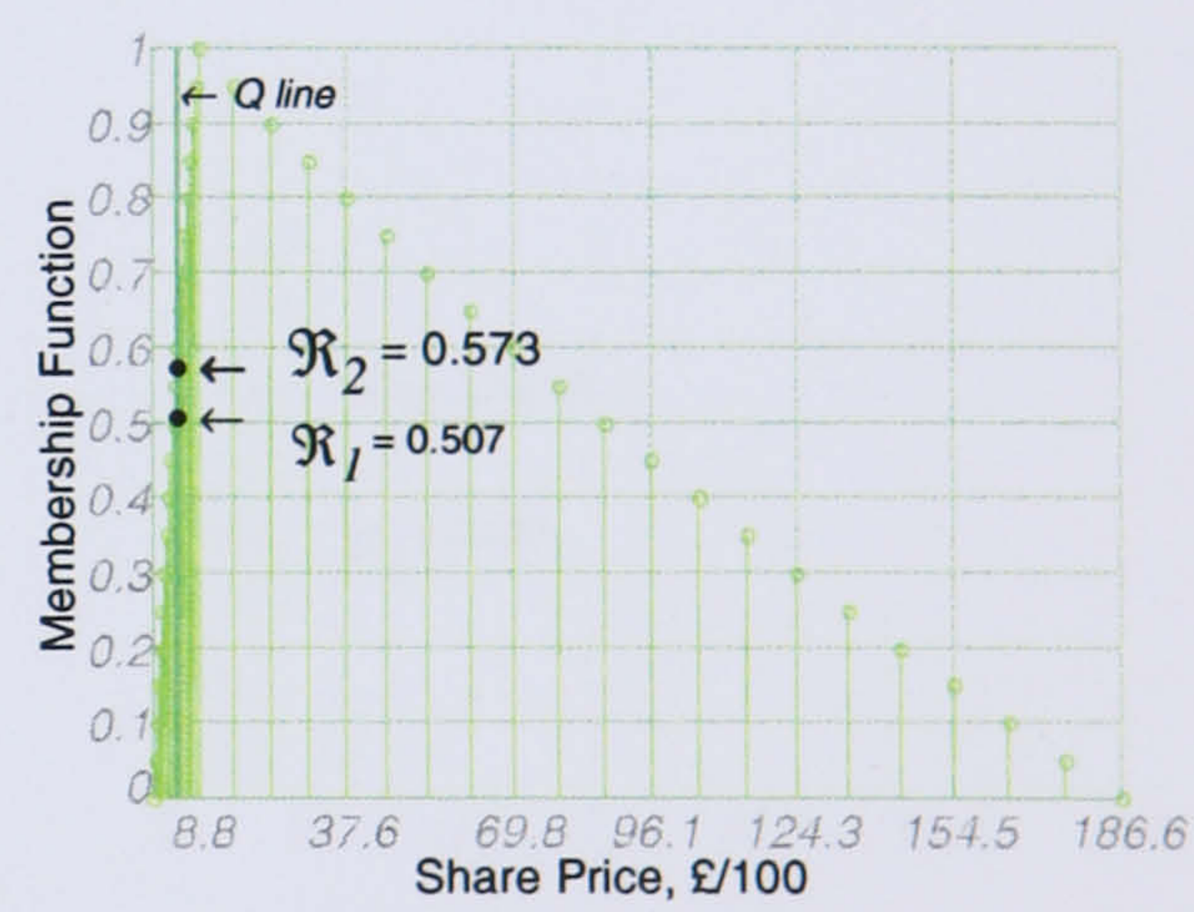


Figure A3.14b: HANSON - evaluated robustness $\Delta = 0.934$

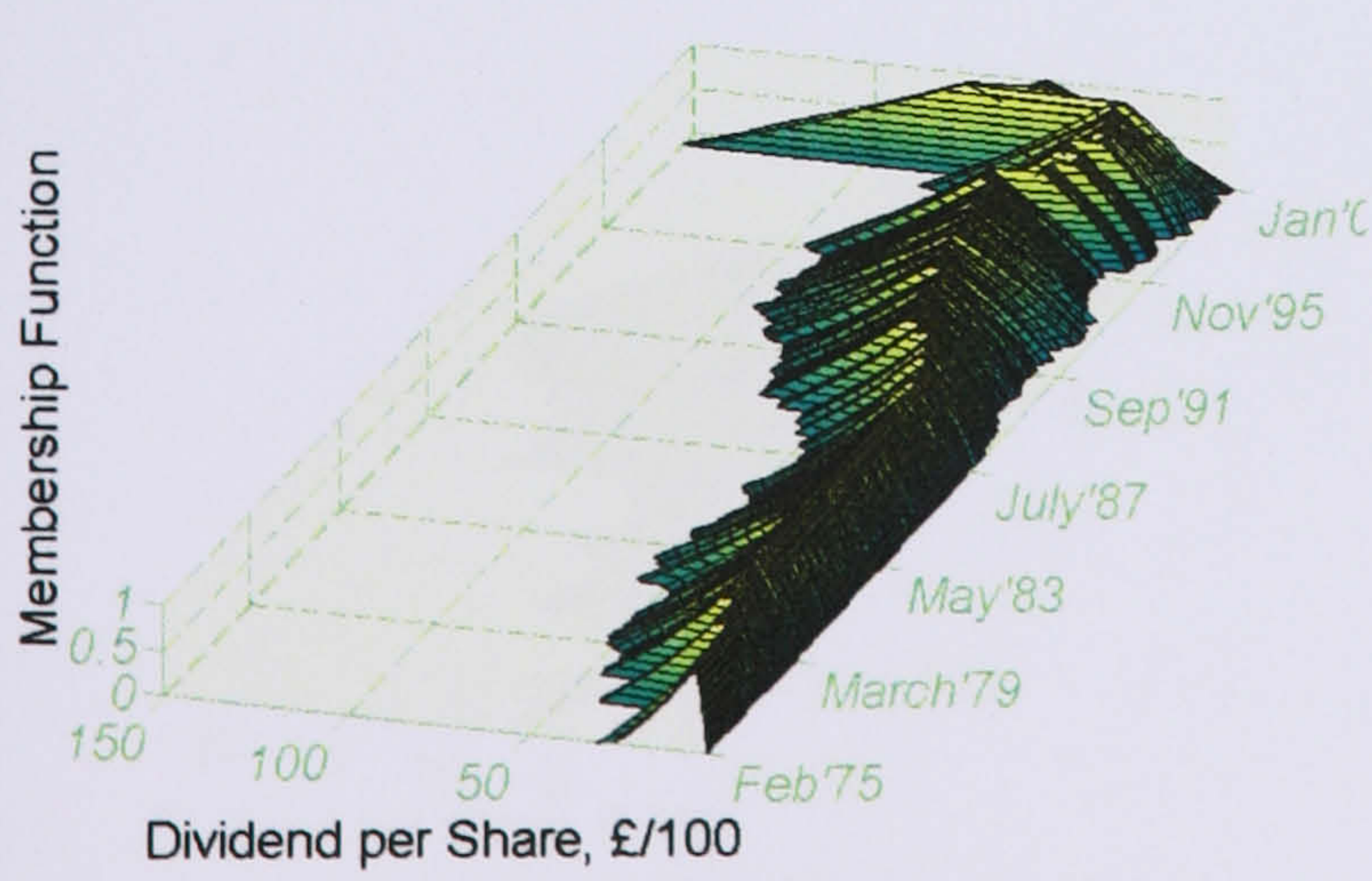


Figure A3.15a: INCHCAPE - fuzzified data under a broader range of imprecision

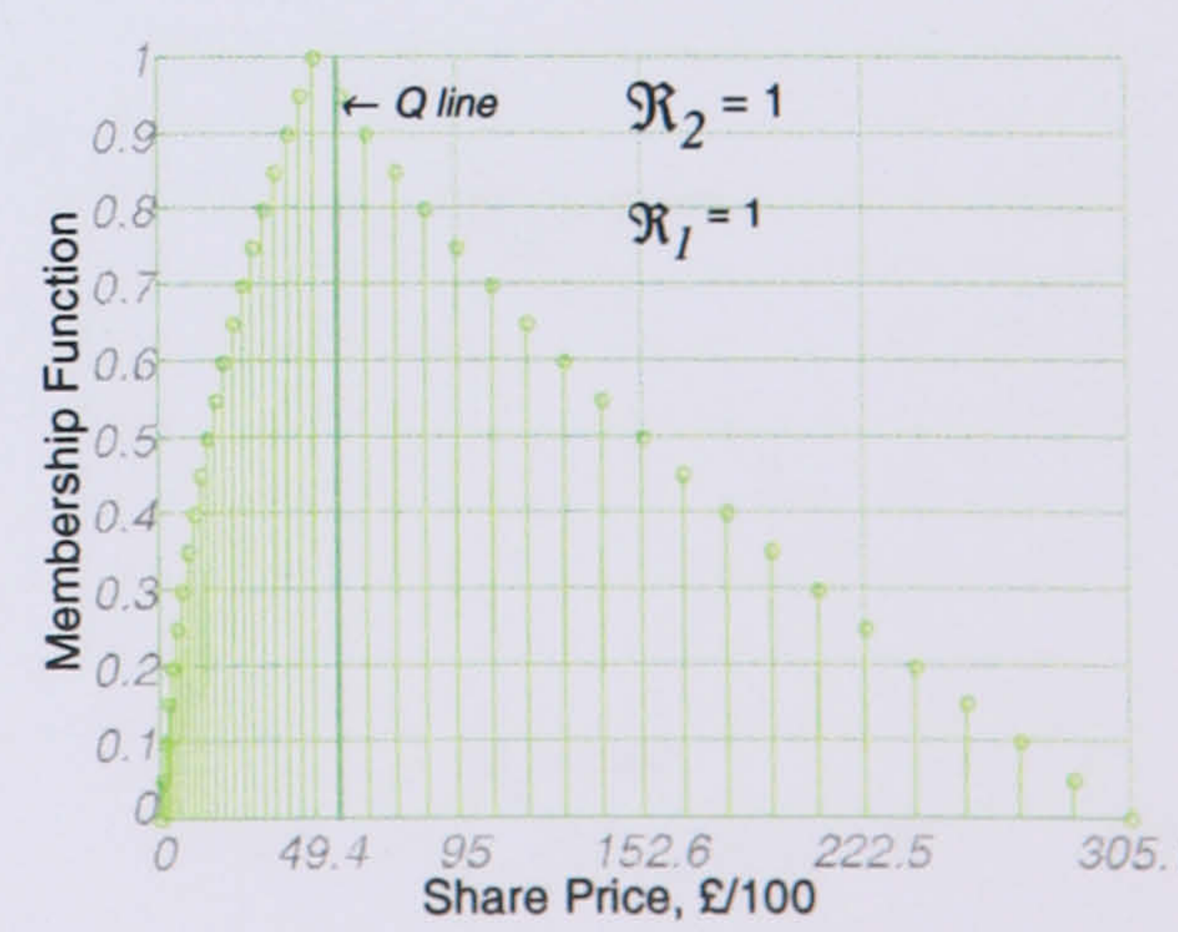


Figure A3.15b: INCHCAPE - no robustness measure is assigned

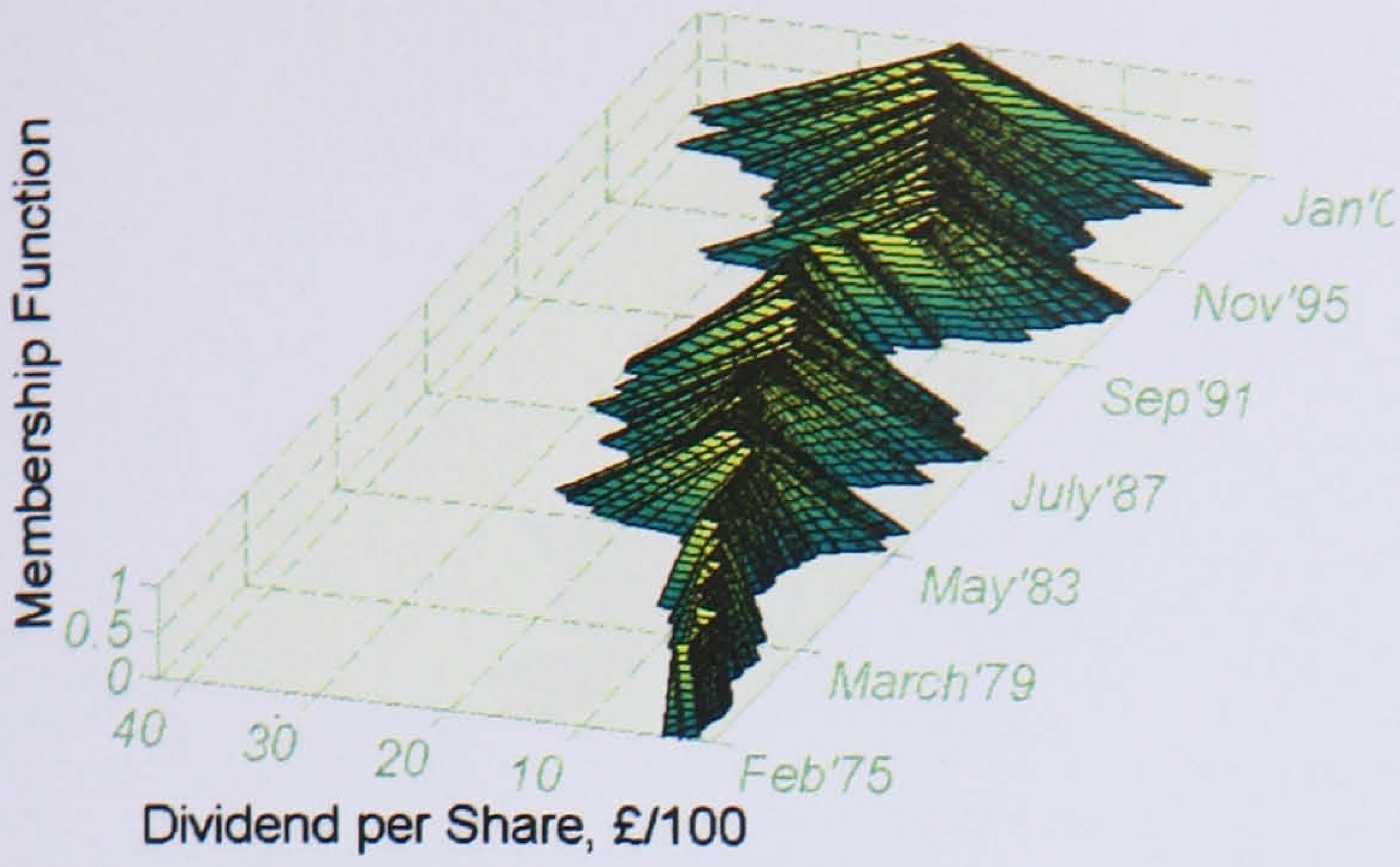


Figure A3.16a: LEX SERVICE - fuzzified data under a broader range of imprecision

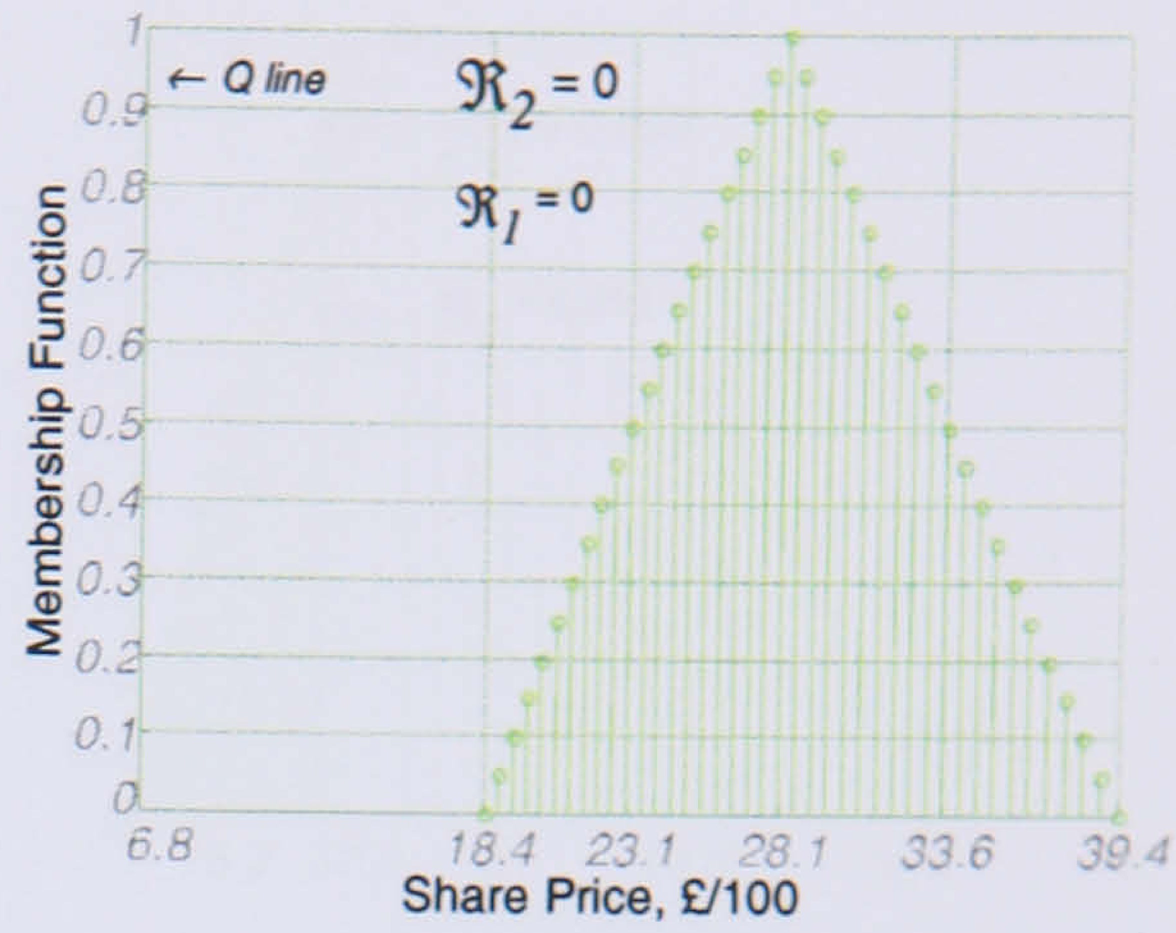


Figure A3.16b: LEX SERVICE - evaluated robustness $\Delta = 1$

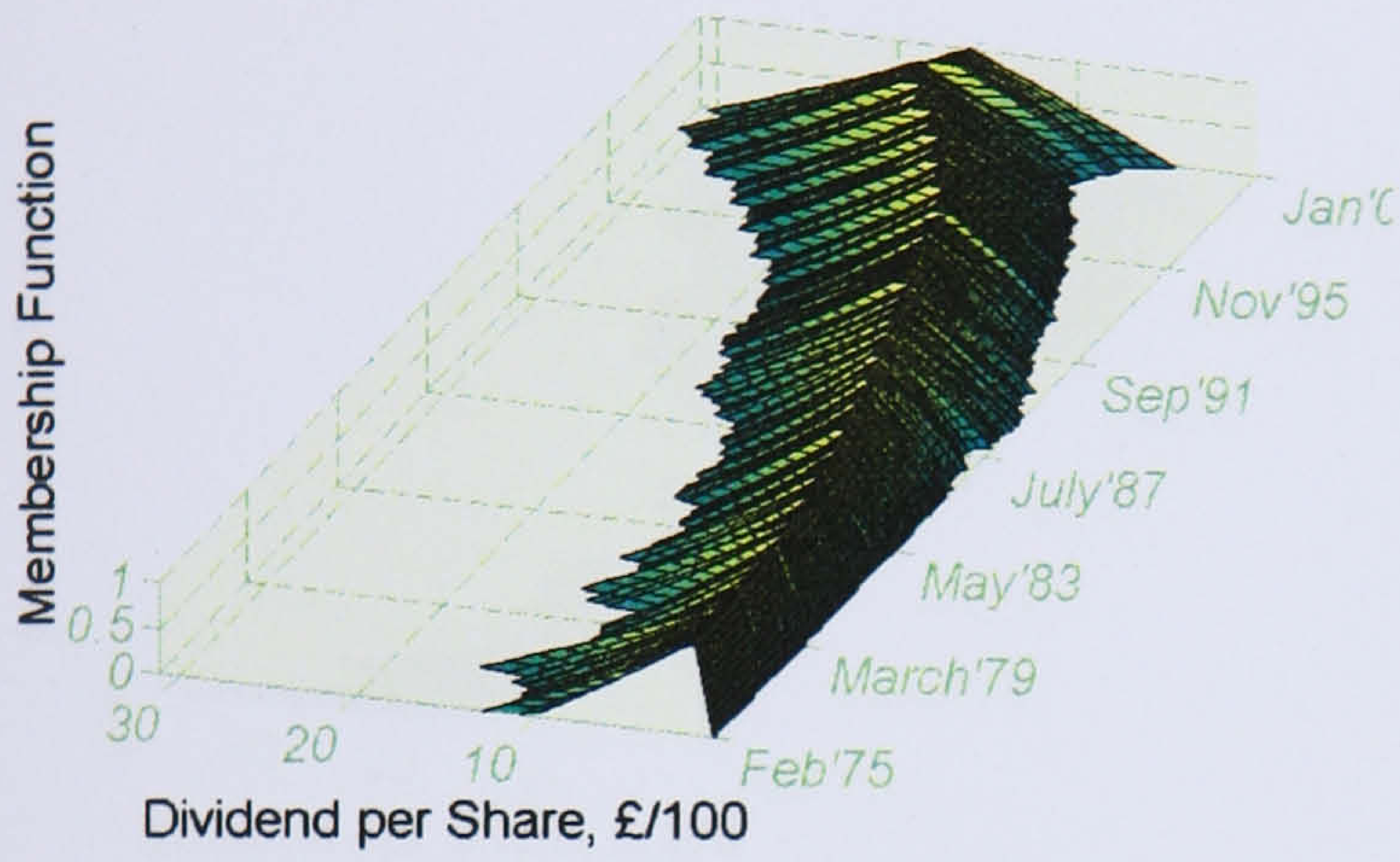


Figure A3.17a: MARKS & SPENCER - fuzzified data under a broader range of imprecision

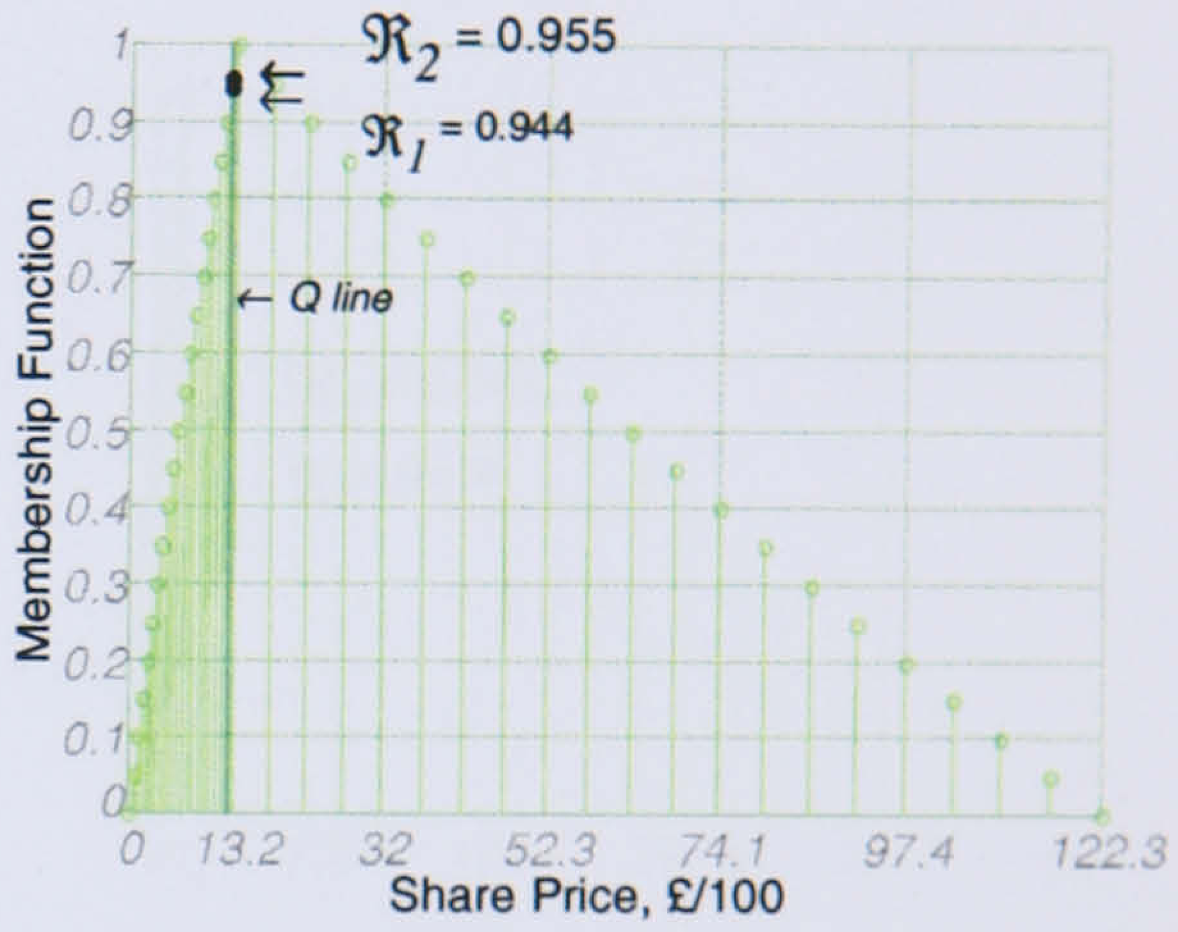


Figure A3.17b: MARKS & SPENCER - evaluated robustness $\Delta = 0.989$

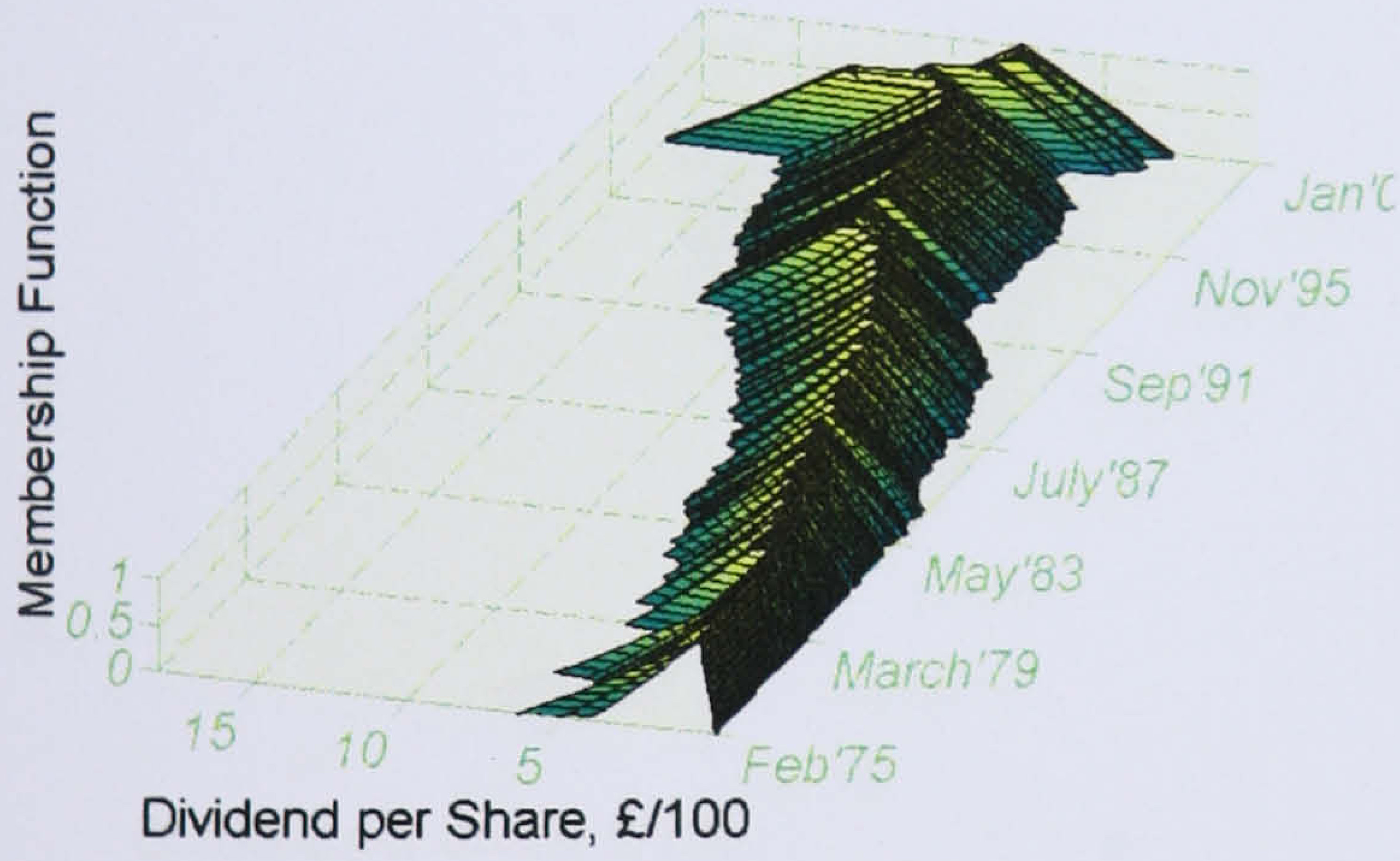


Figure A3.18a: NORTHERN FOODS - fuzzified data under a broader range of imprecision

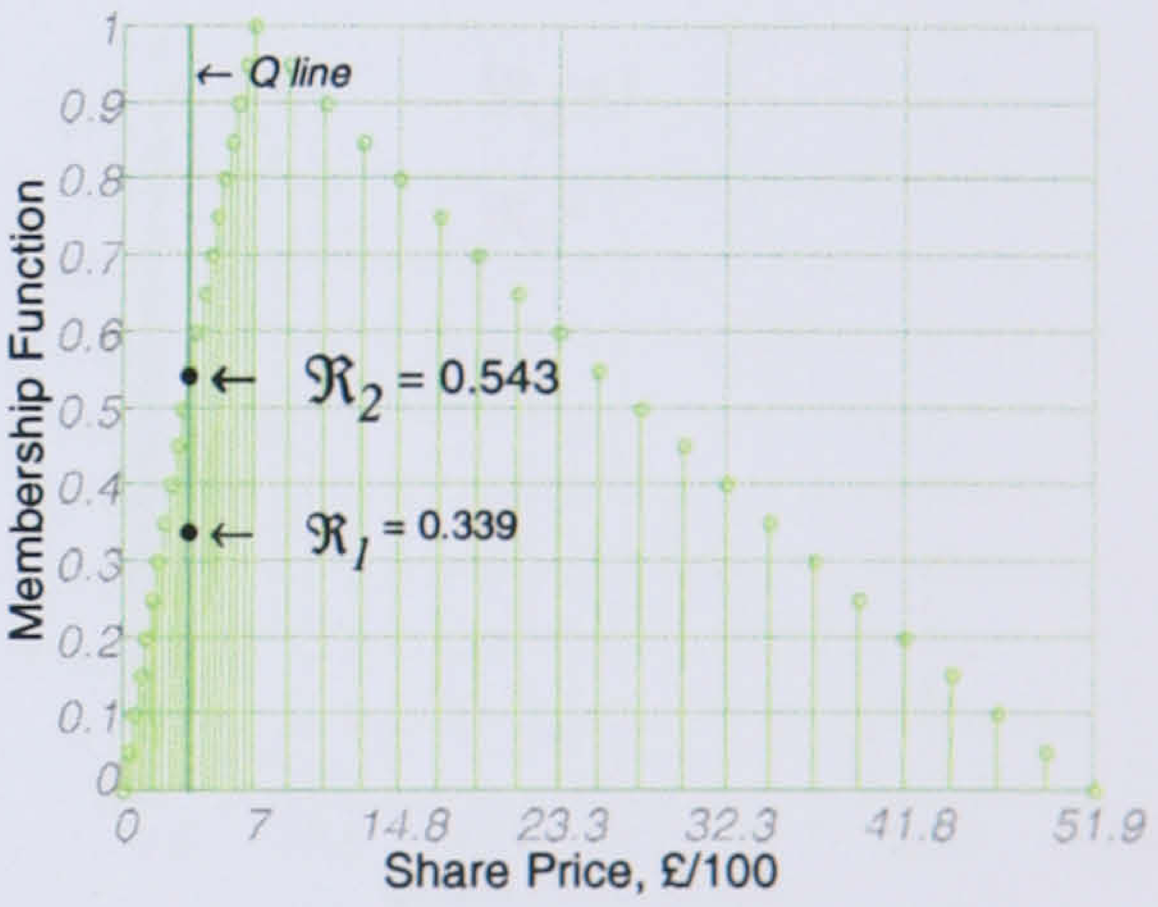


Figure A3.18b: NORTHERN FOODS - evaluated robustness $\Delta = 0.796$

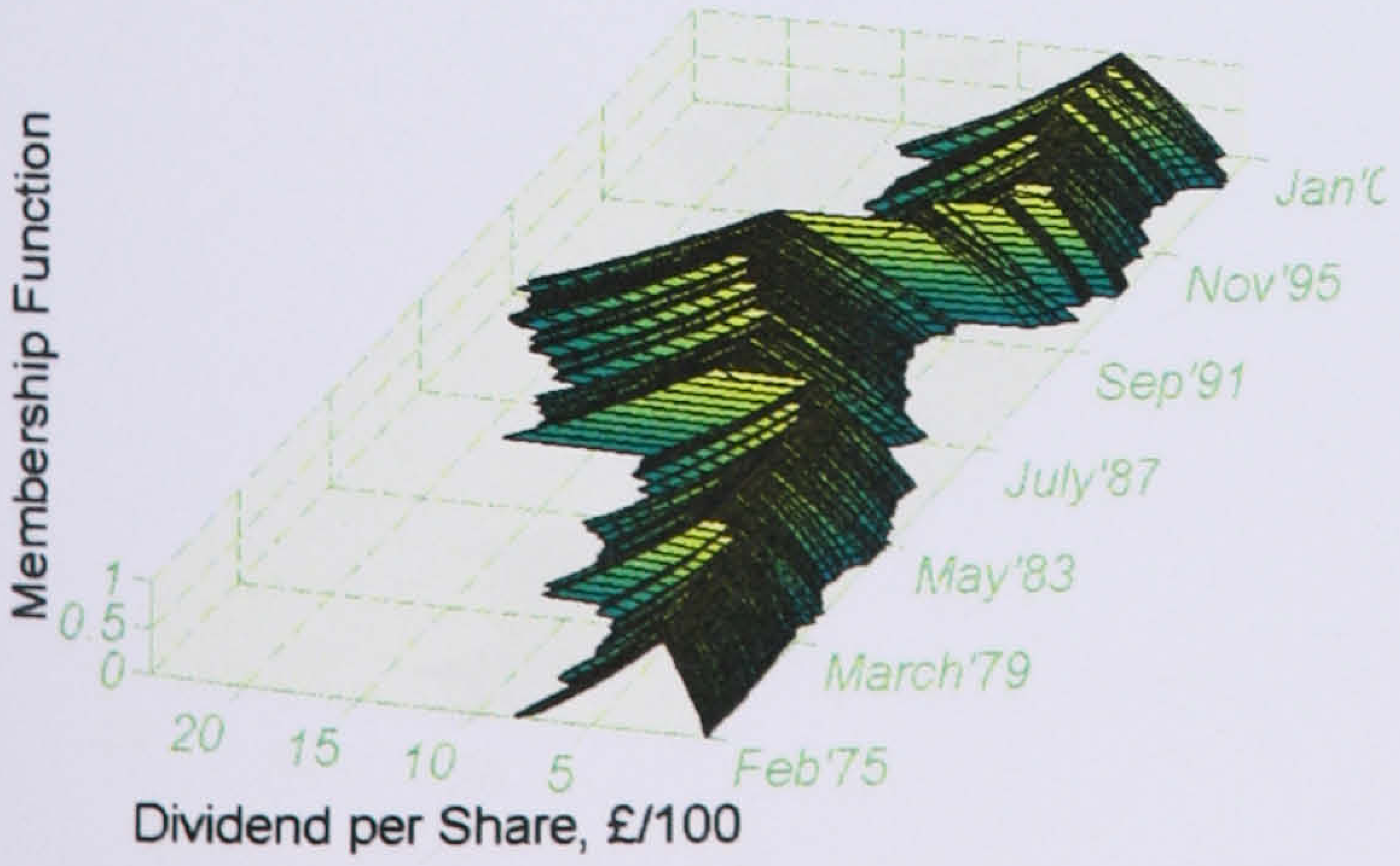


Figure A3.19a: PILKINGTON - fuzzified data under a broader range of imprecision

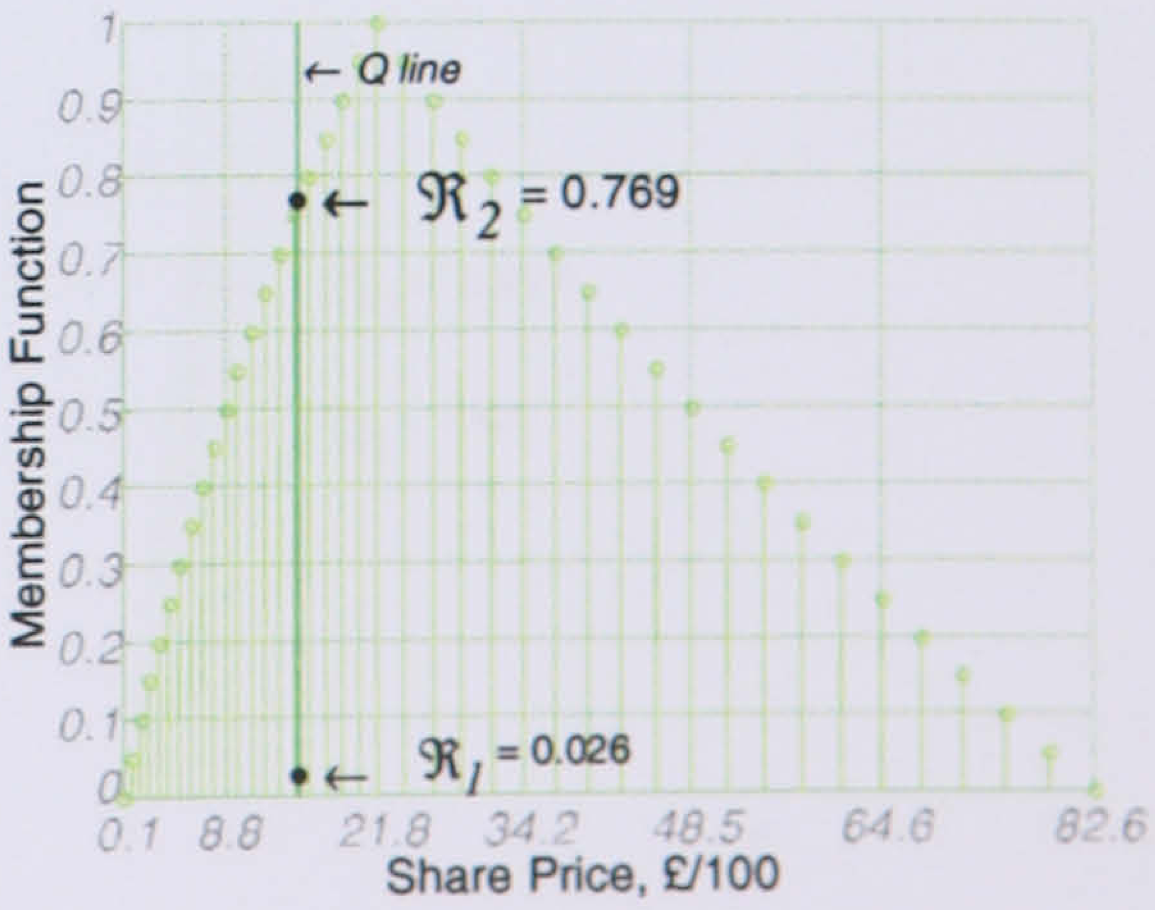


Figure A3.19b: PILKINGTON - evaluated robustness $\Delta = 0.257$

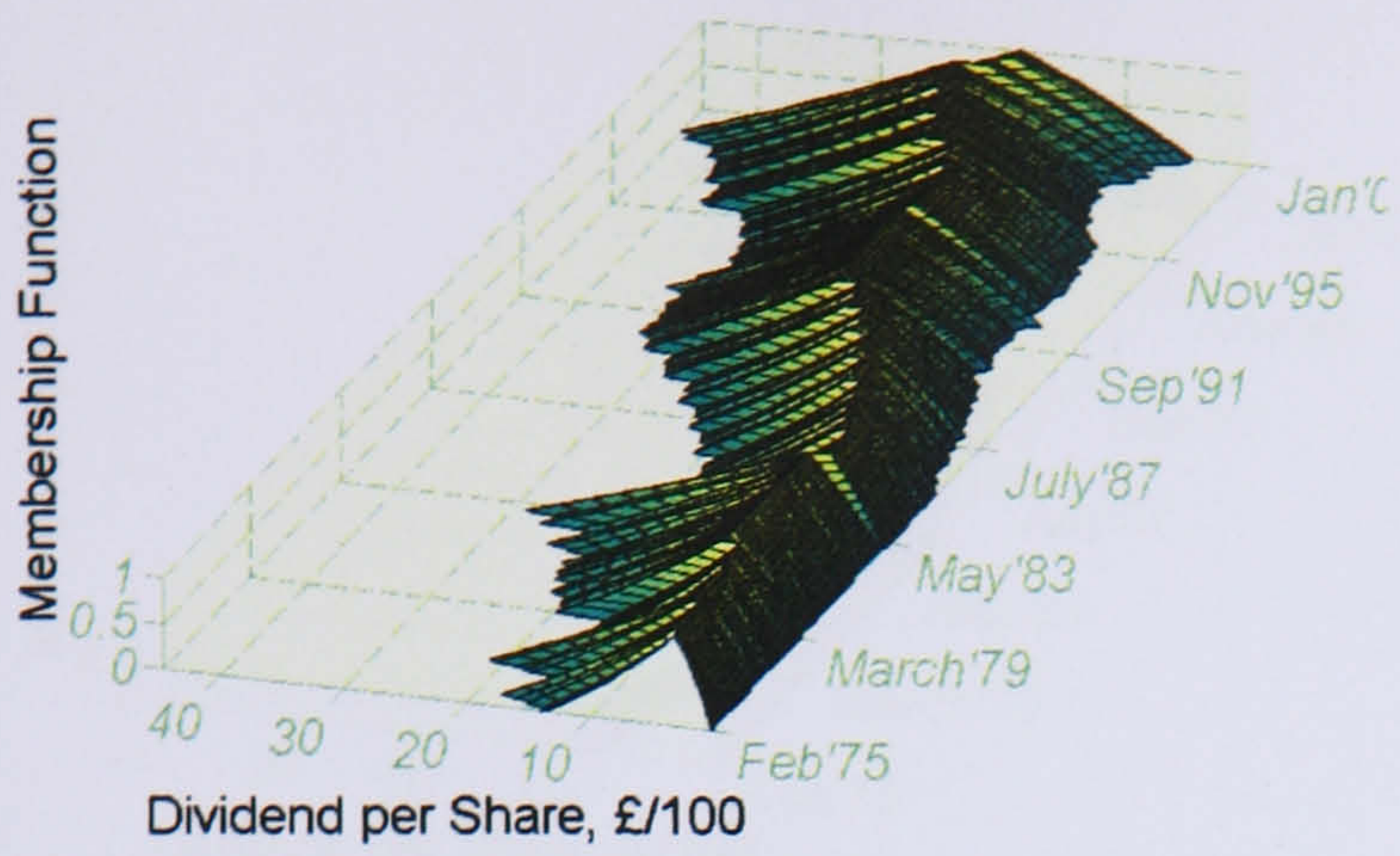


Figure A3.20a: RANK GROUP - fuzzified data under a broader range of imprecision

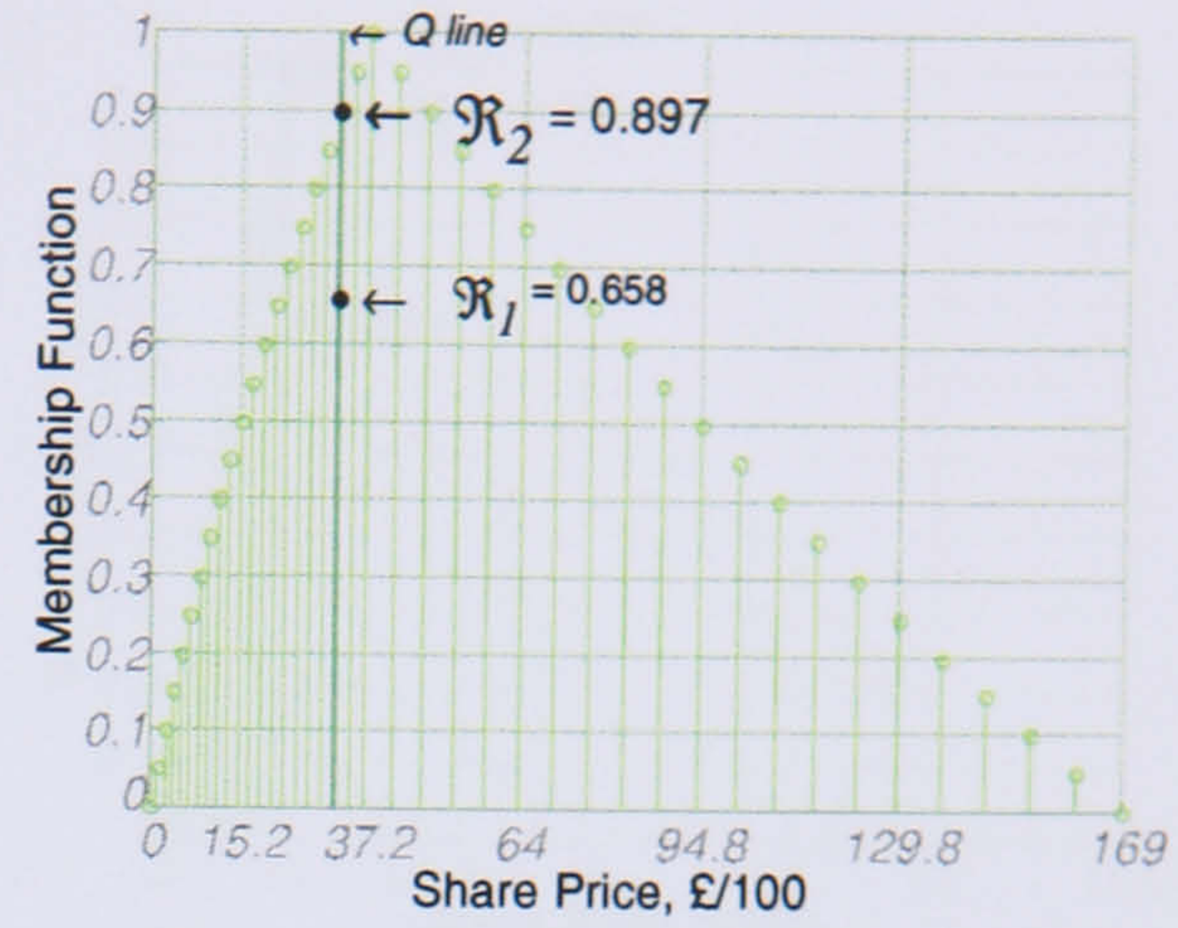


Figure A3.20b: RANK GROUP - evaluated robustness $\Delta = 0.761$

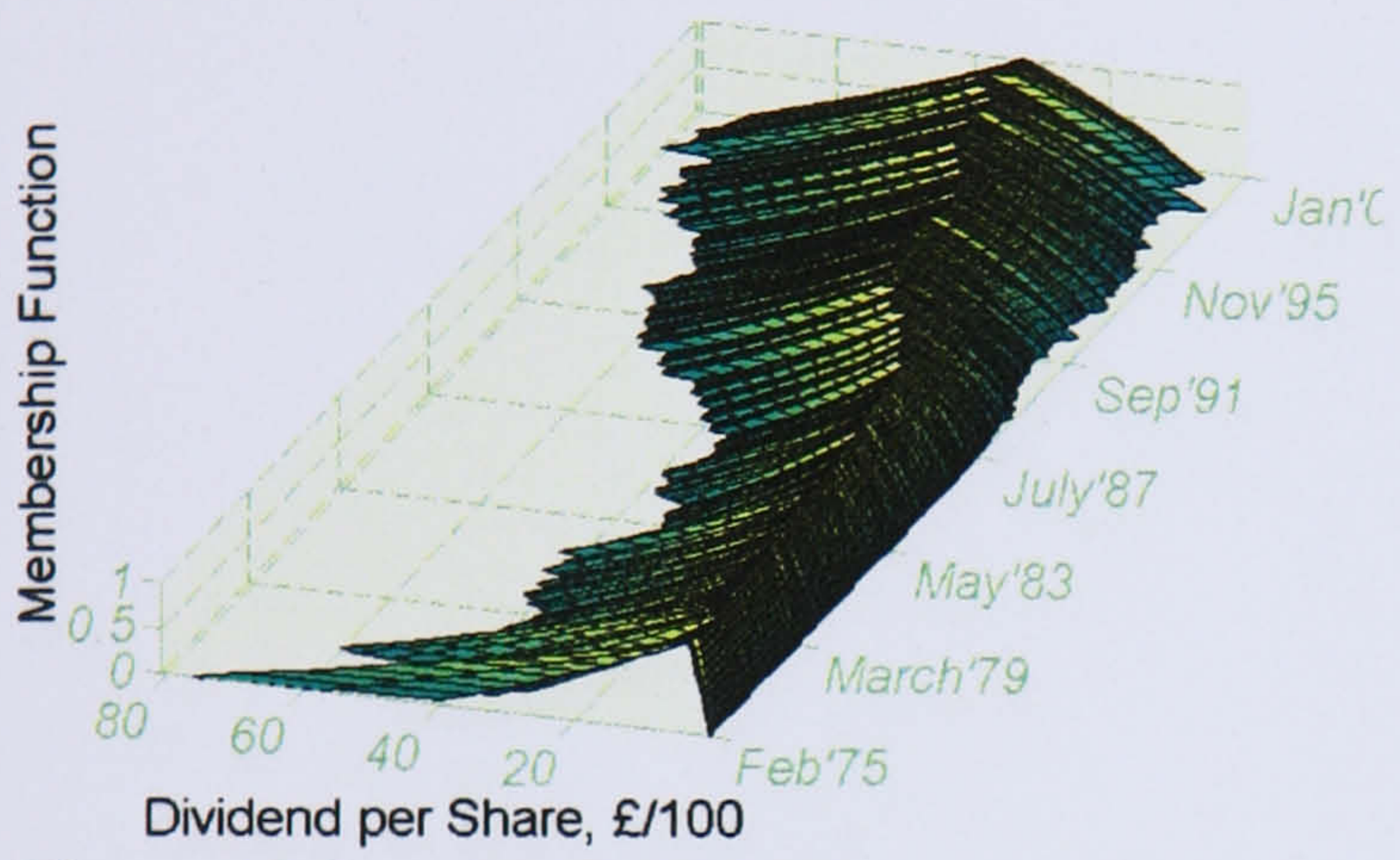


Figure A3.21a: RMC GROUP - fuzzified data under a broader range of imprecision

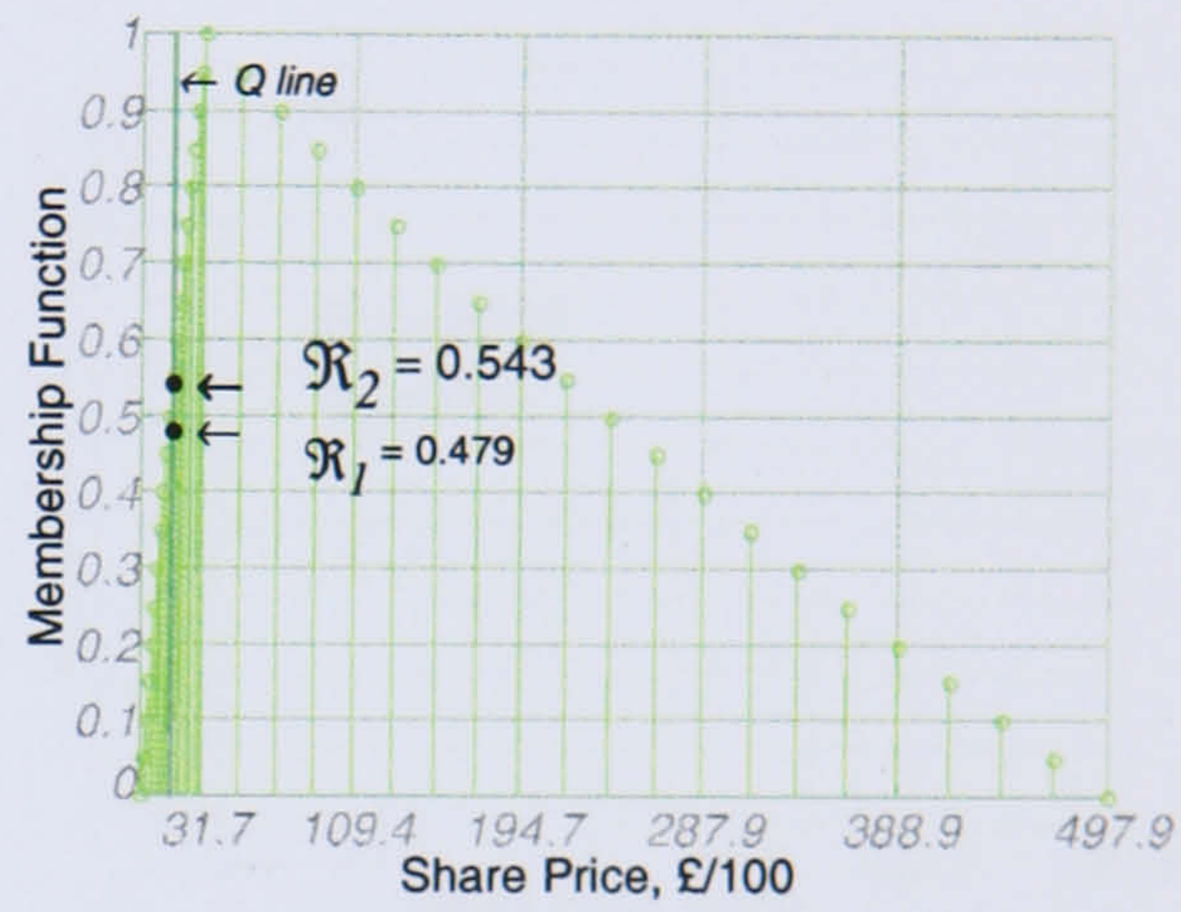


Figure A3.21b: RMC GROUP - evaluated robustness $\Delta = 0.936$

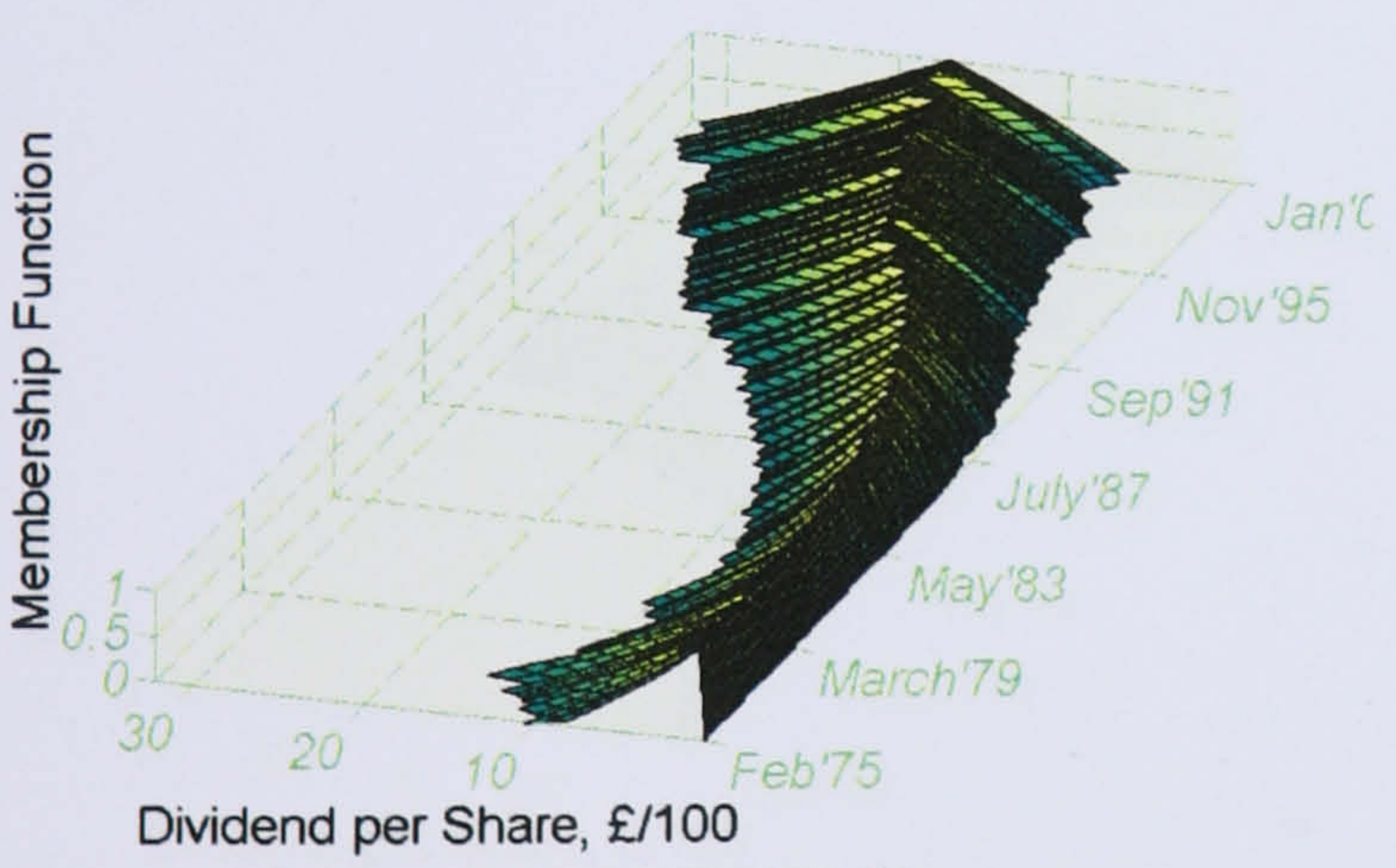


Figure A3.22a: SAINSBURY (J) - fuzzified data under a broader range of imprecision

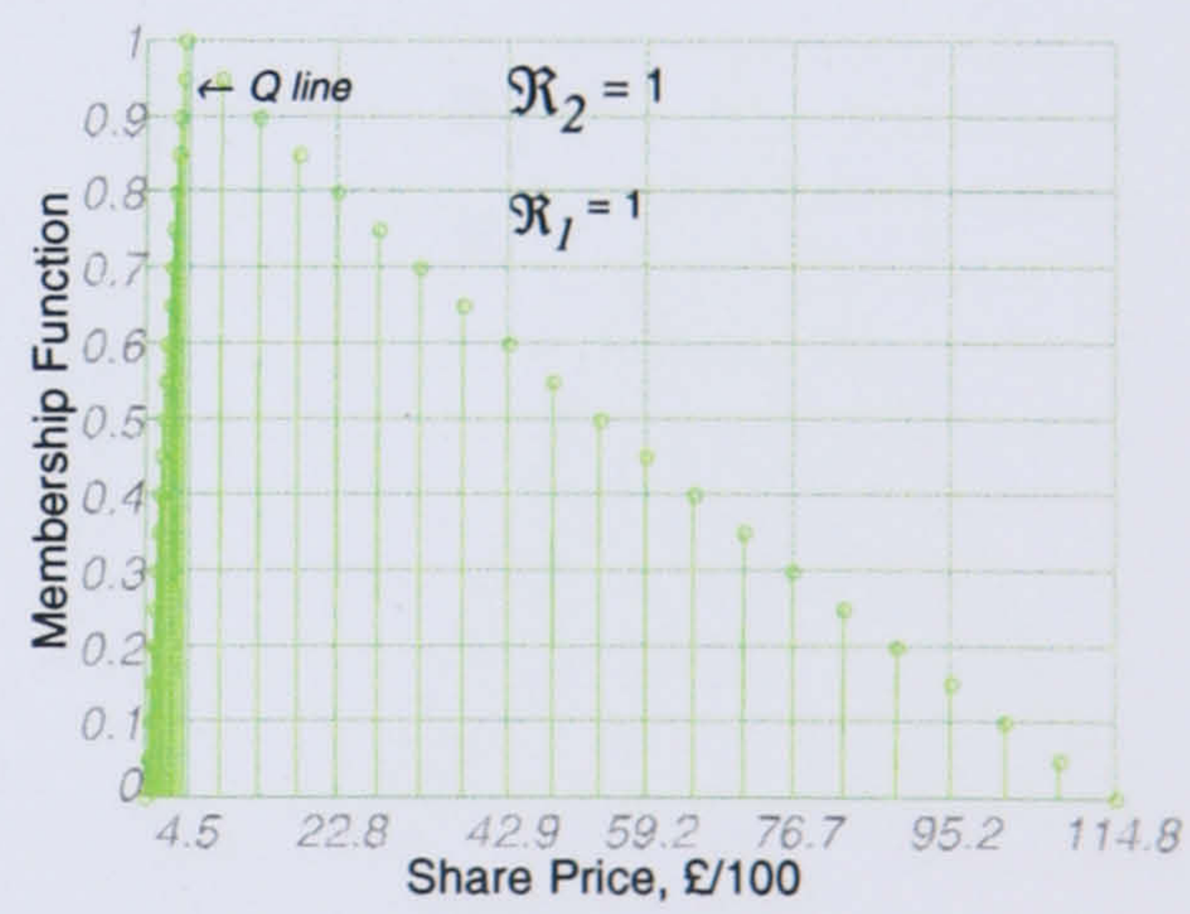


Figure A3.22b: SAINSBURY (J) - no robustness measure is assigned

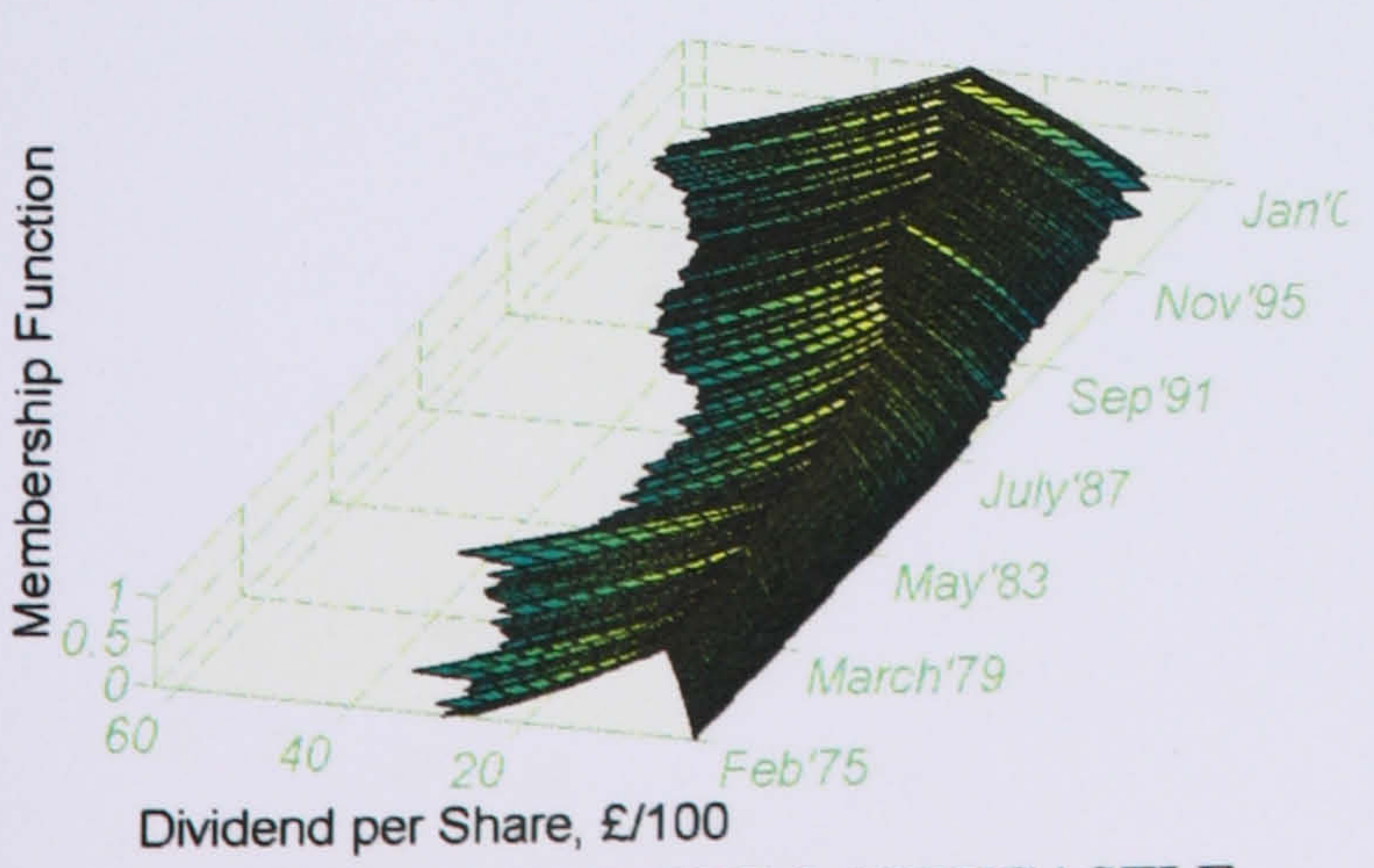


Figure A3.23a: SCOTTISH & NEWCASTLE - fuzzified data under a broader range of imprecision

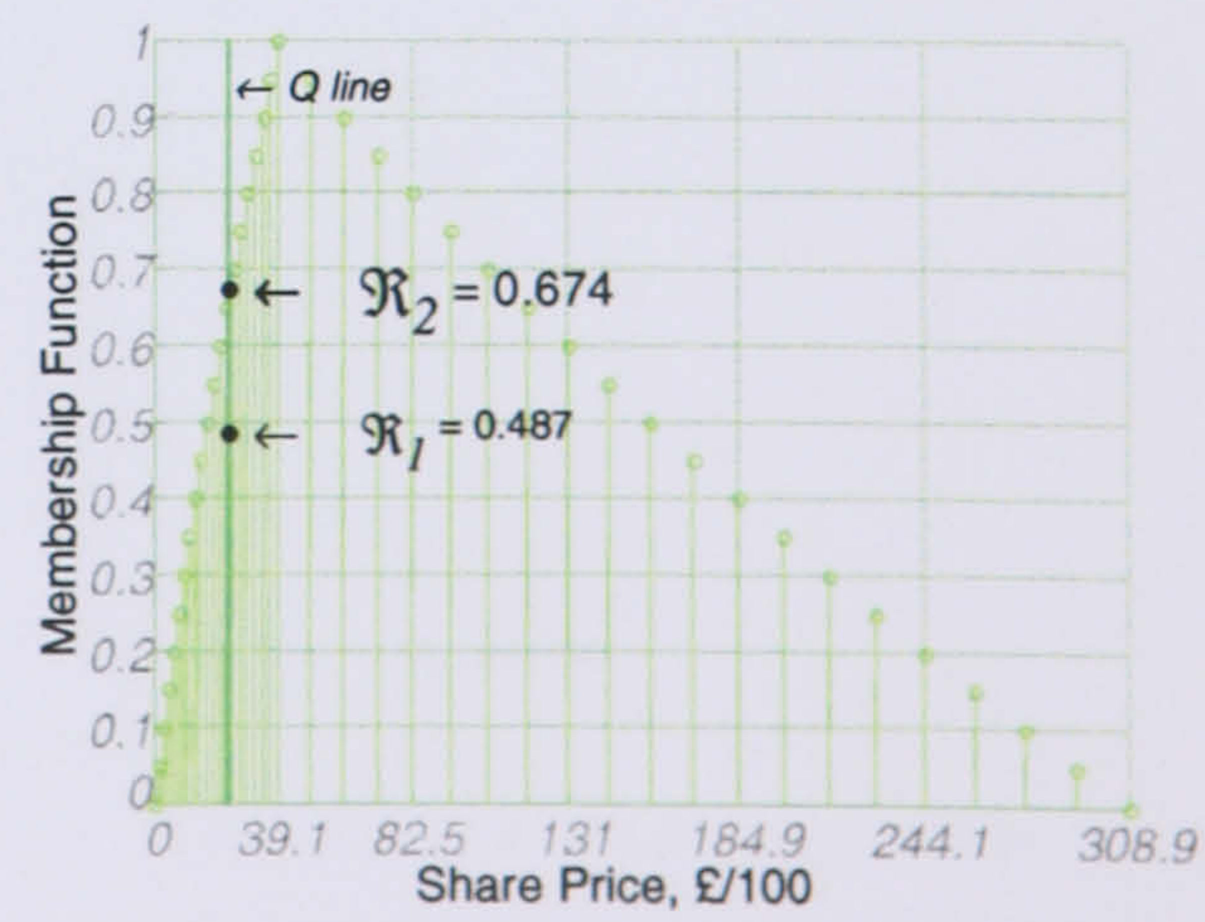


Figure A3.23b: SCOTTISH & NEWCASTLE - evaluated robustness $\Delta = 0.813$

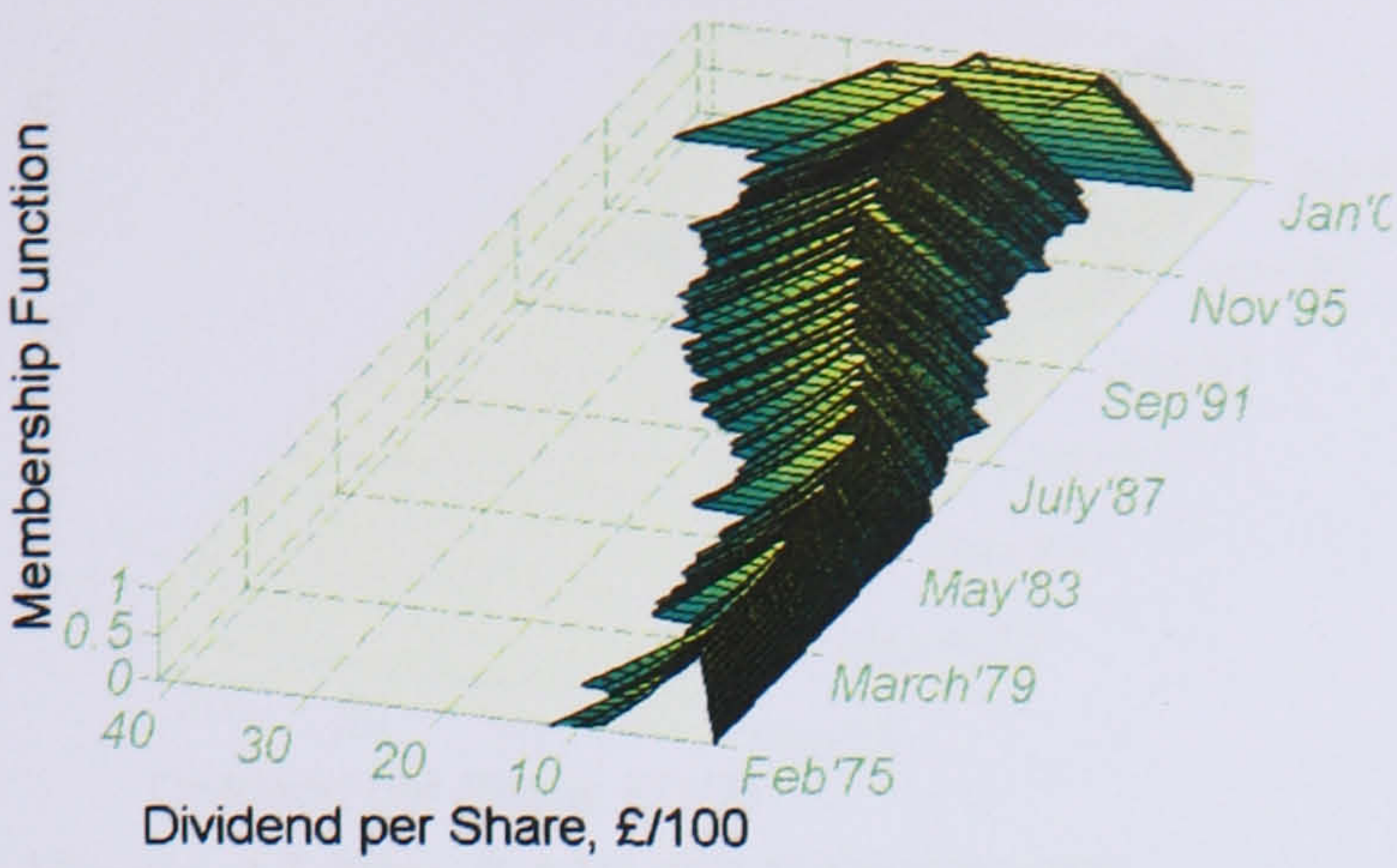


Figure A3.24a: SMITH (WH) GROUP - fuzzified data under a broader range of imprecision

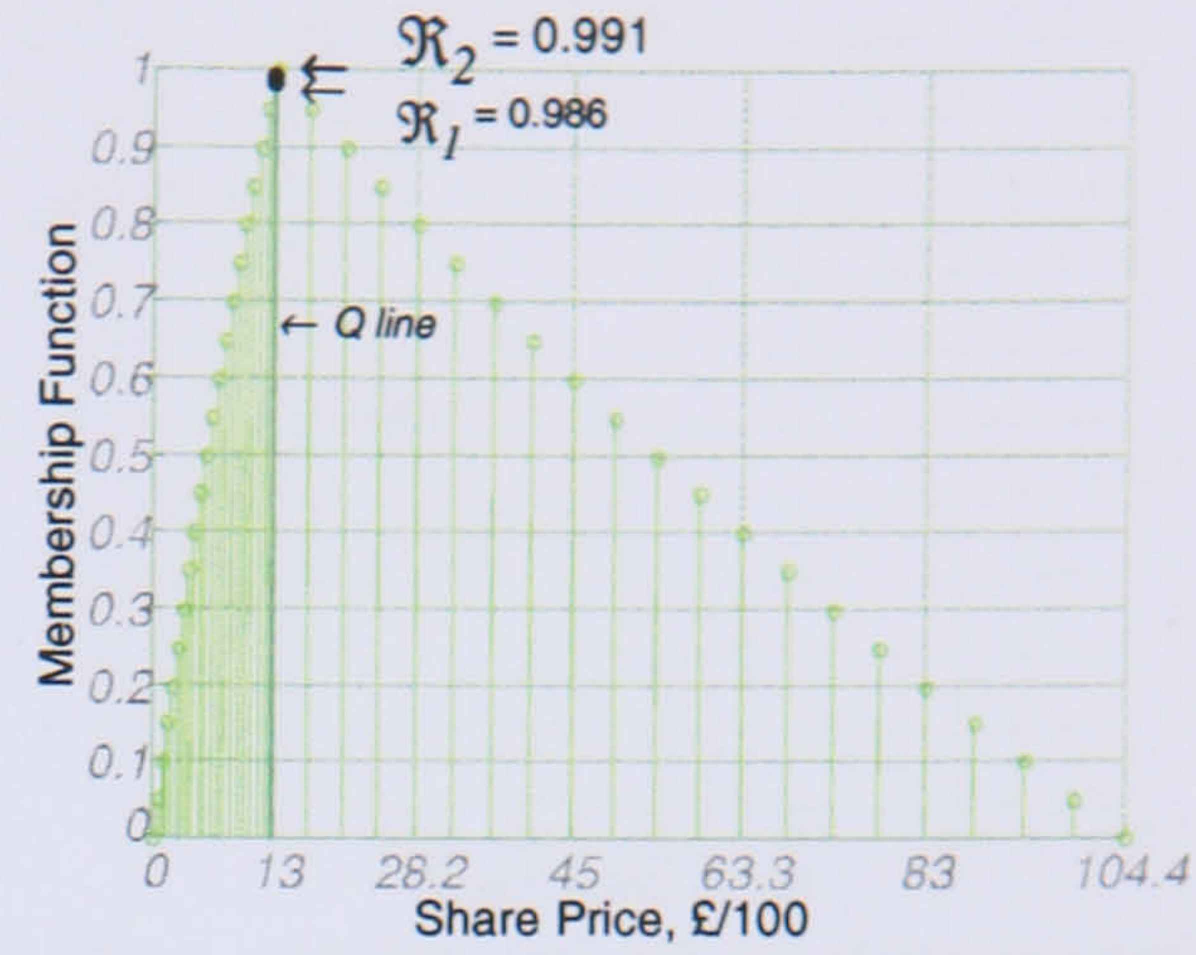


Figure A3.24b: SMITH (WH) GROUP - evaluated robustness $\Delta = 0.995$

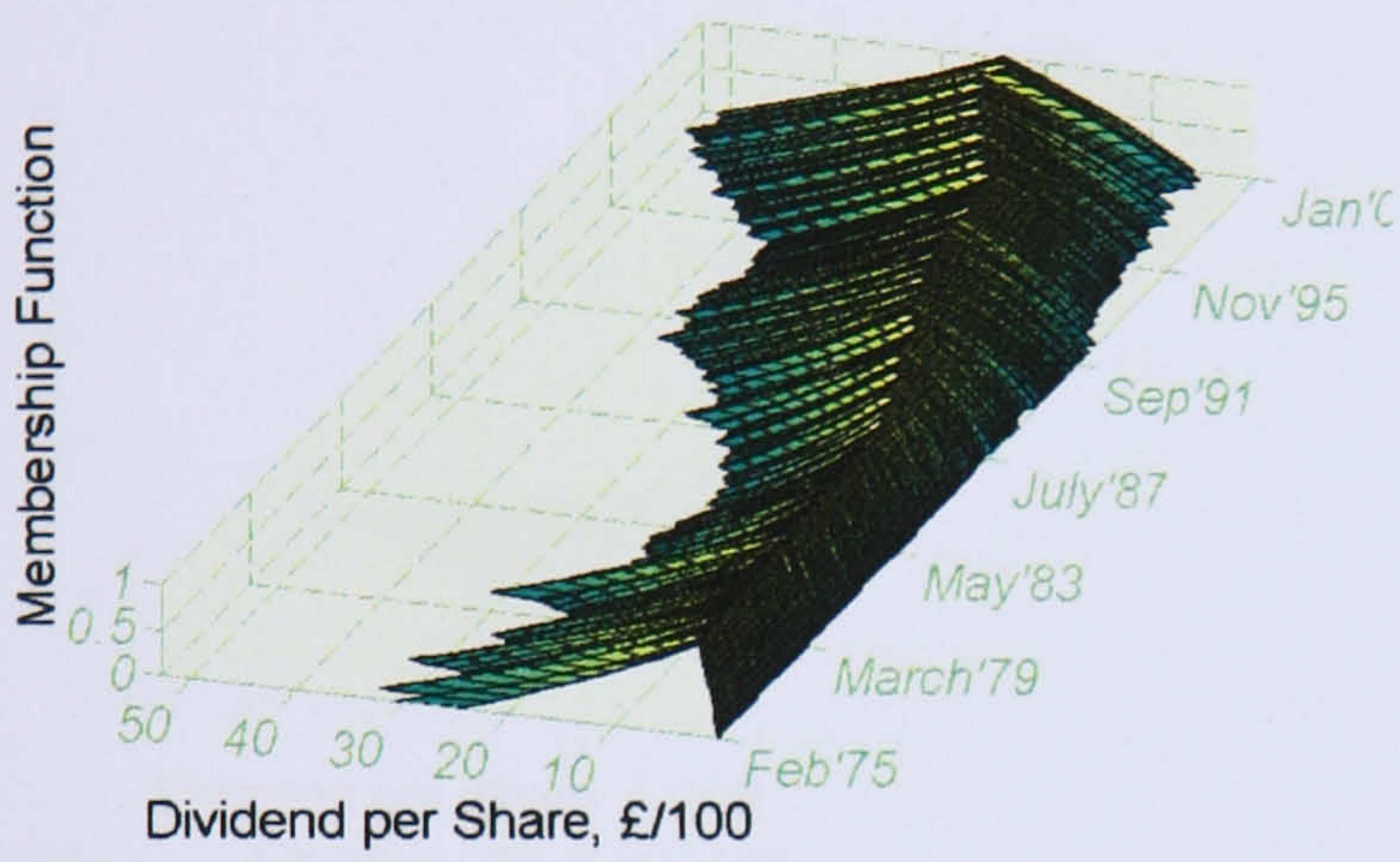


Figure A3.25a: SMITHS INDUSTRIES - fuzzified data under a broader range of imprecision

Figure A3.25b: SMITHS INDUSTRIES - evaluated robustness $\Delta = 0.945$

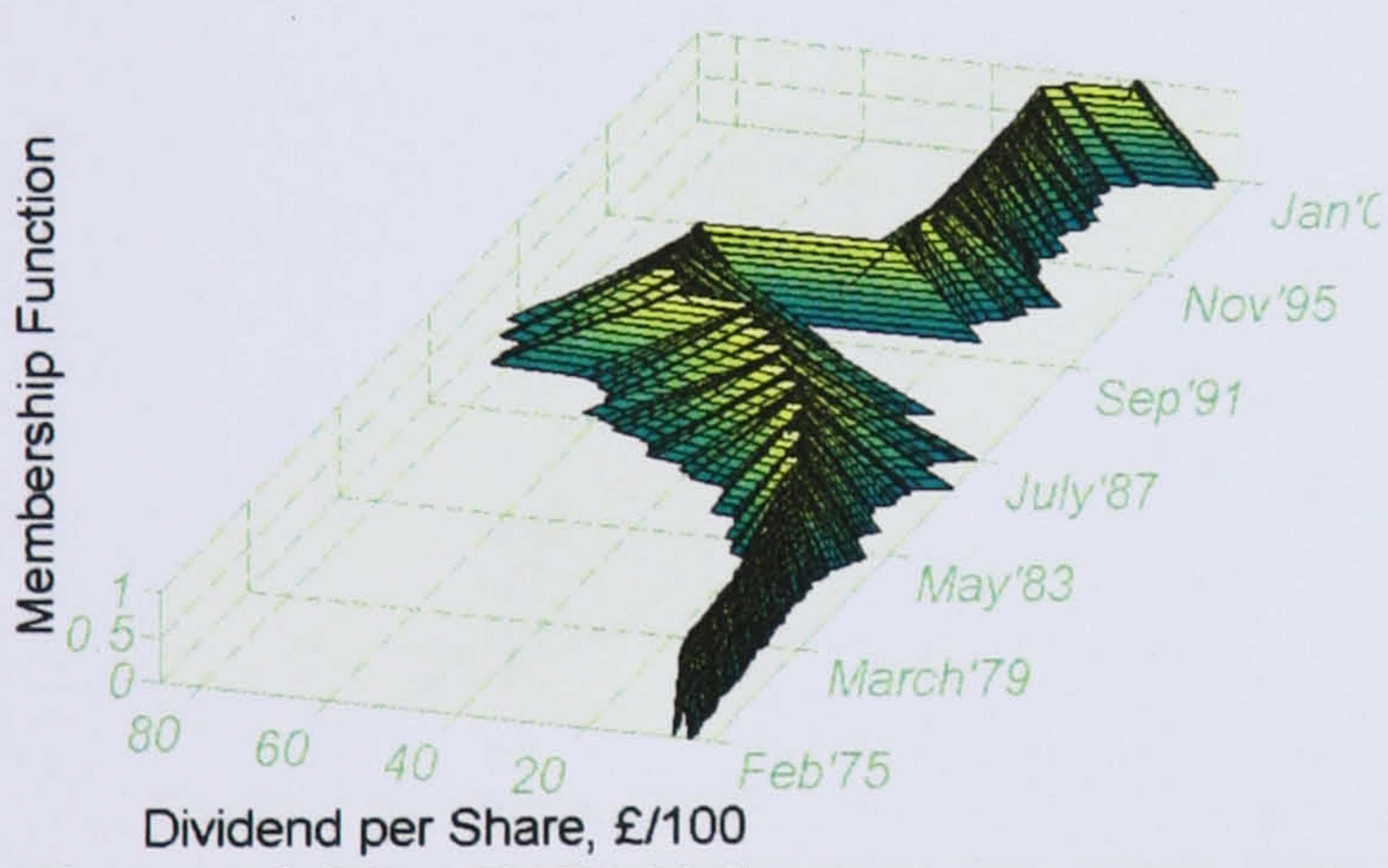


Figure A3.26a: TARMAC - fuzzified data under a broader range of imprecision

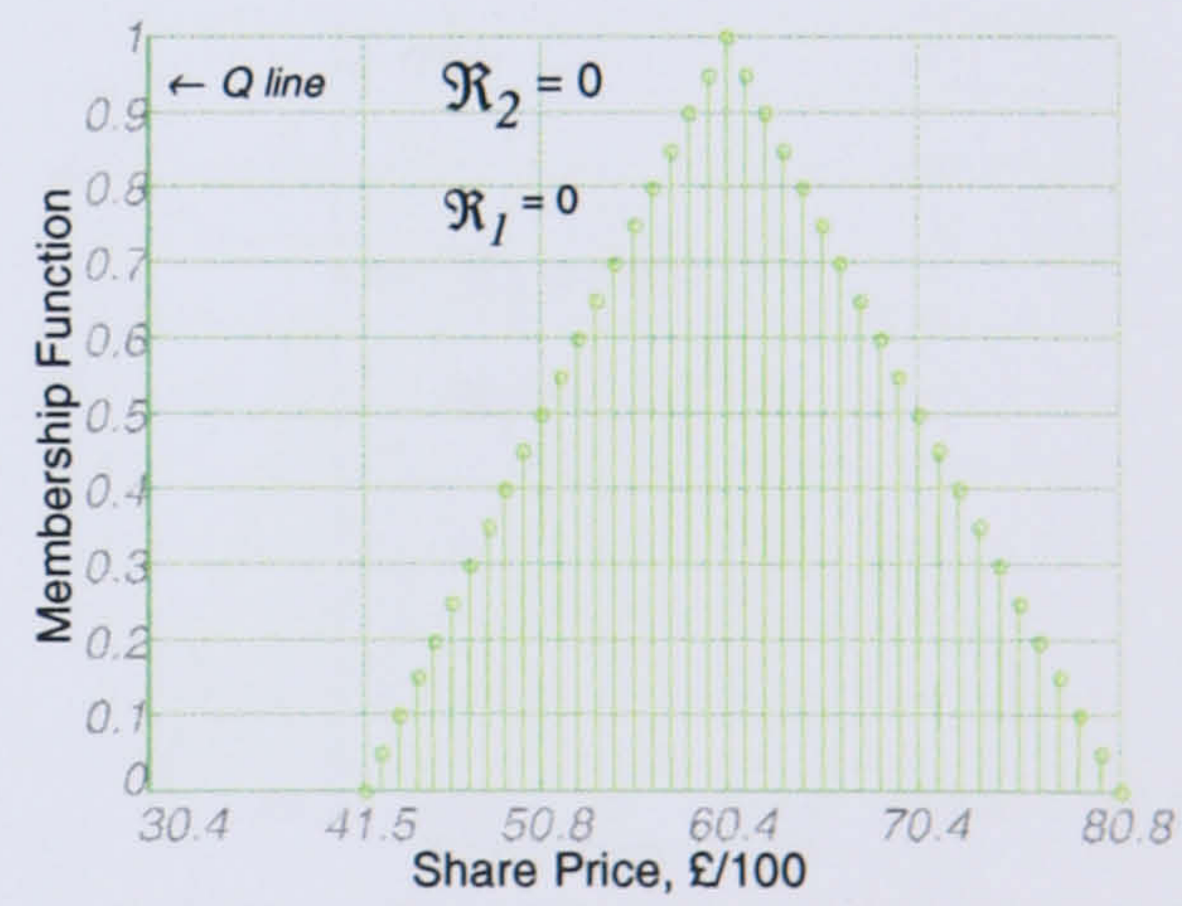


Figure A3.26b: TARMAC - evaluated robustness $\Delta = 1$

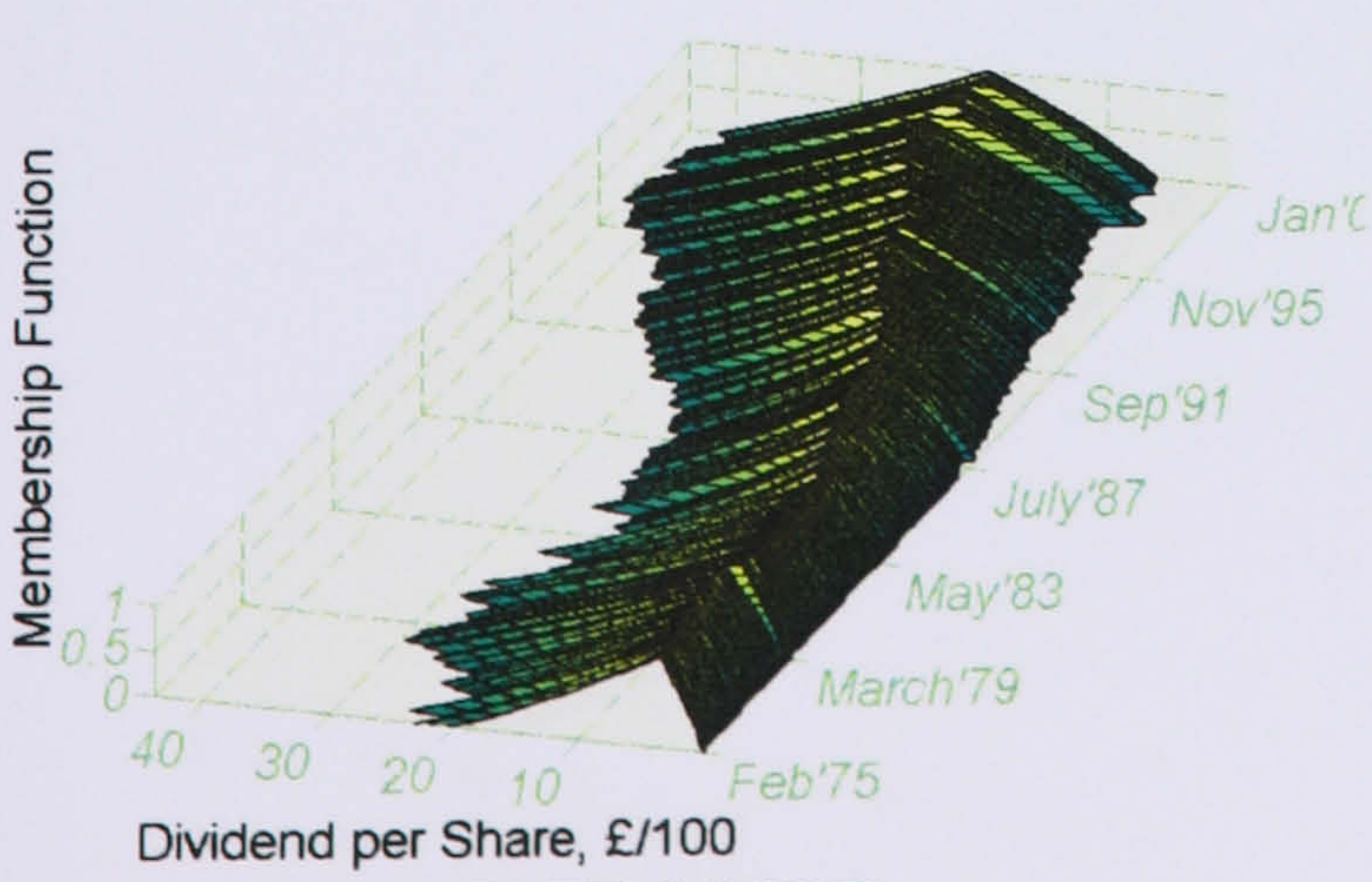


Figure A3.27a: TATE & LYLE - fuzzified data under a broader range of imprecision

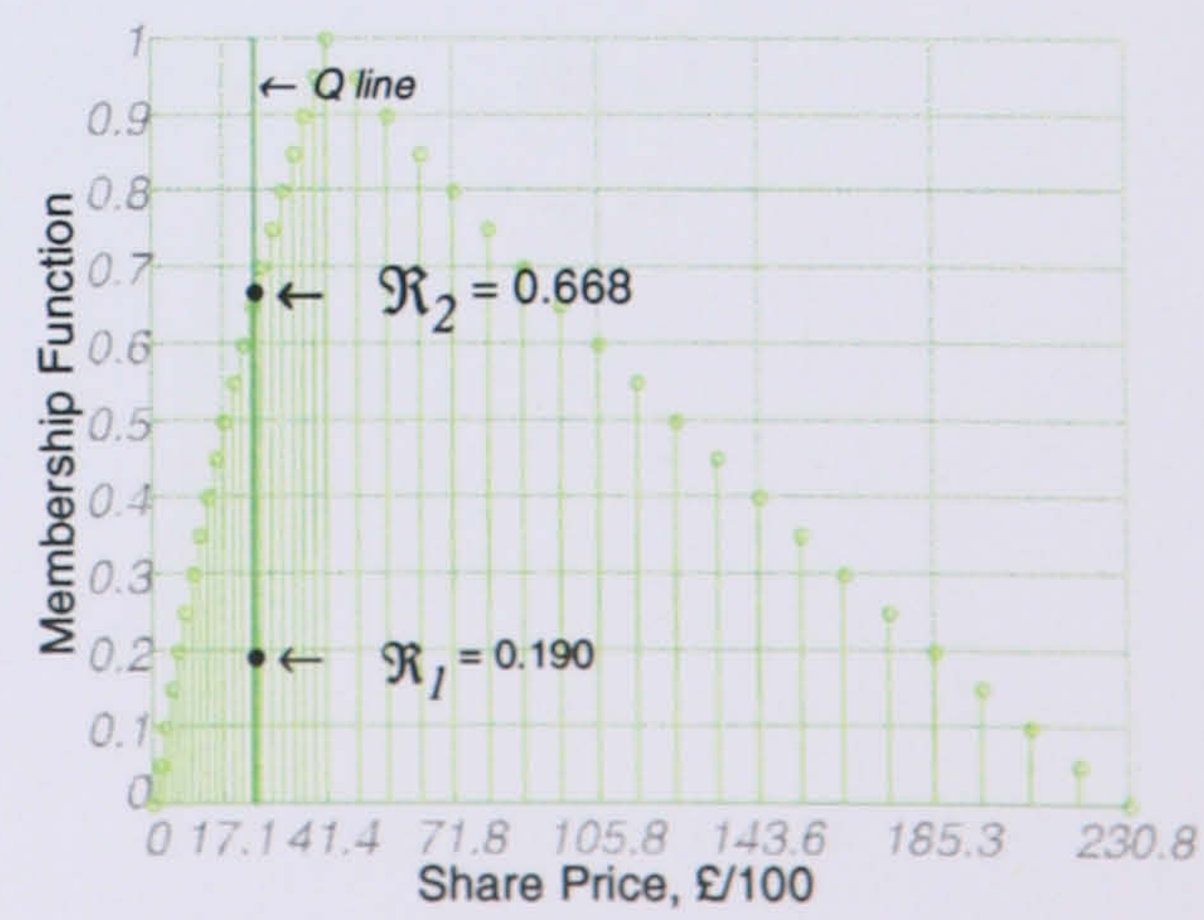


Figure A3.27b: TATE & LYLE - evaluated robustness $\Delta = 0.522$

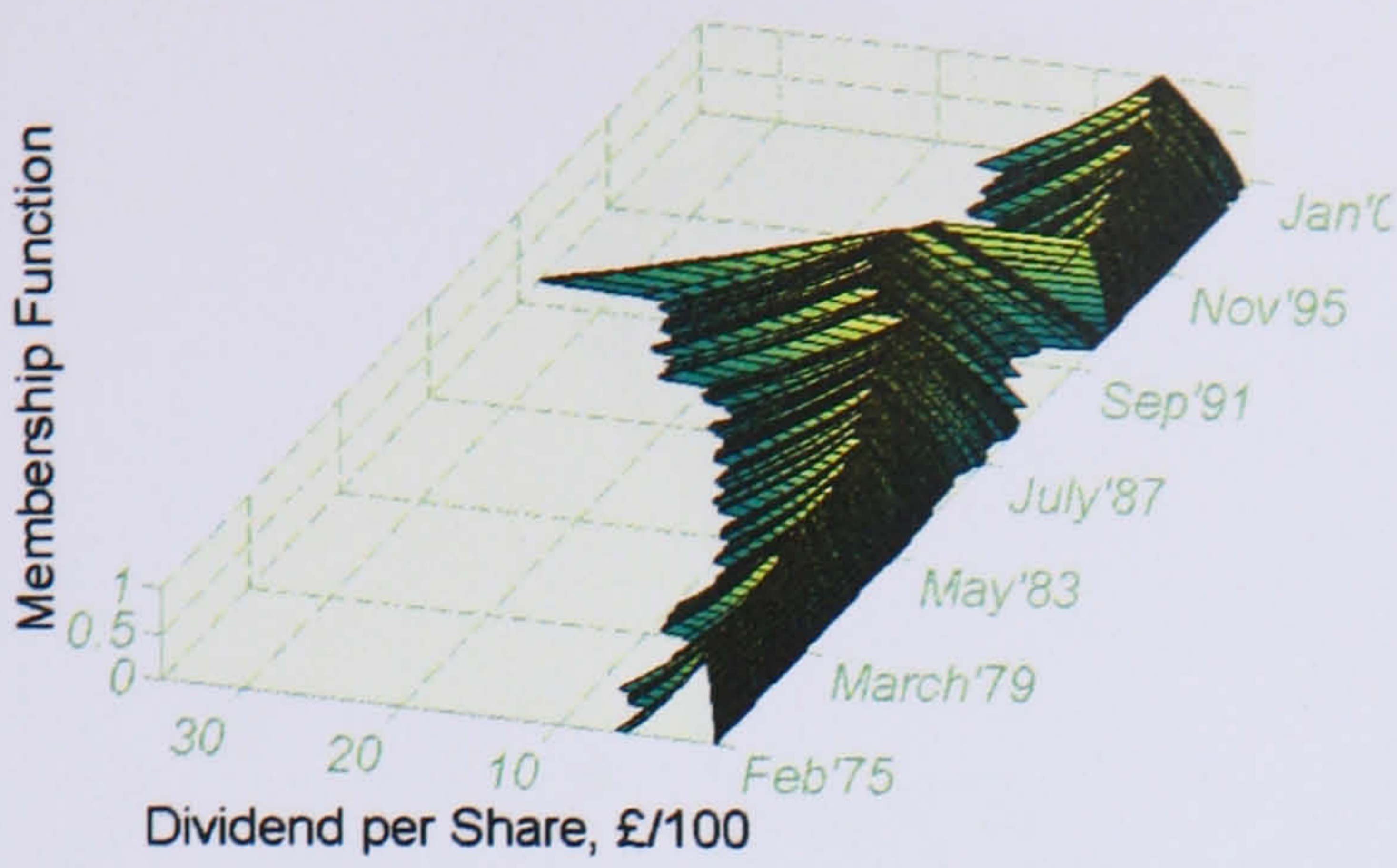


Figure A3.28a: TAYLOR WOODROW -
fuzzified data under a broader range of imprecision

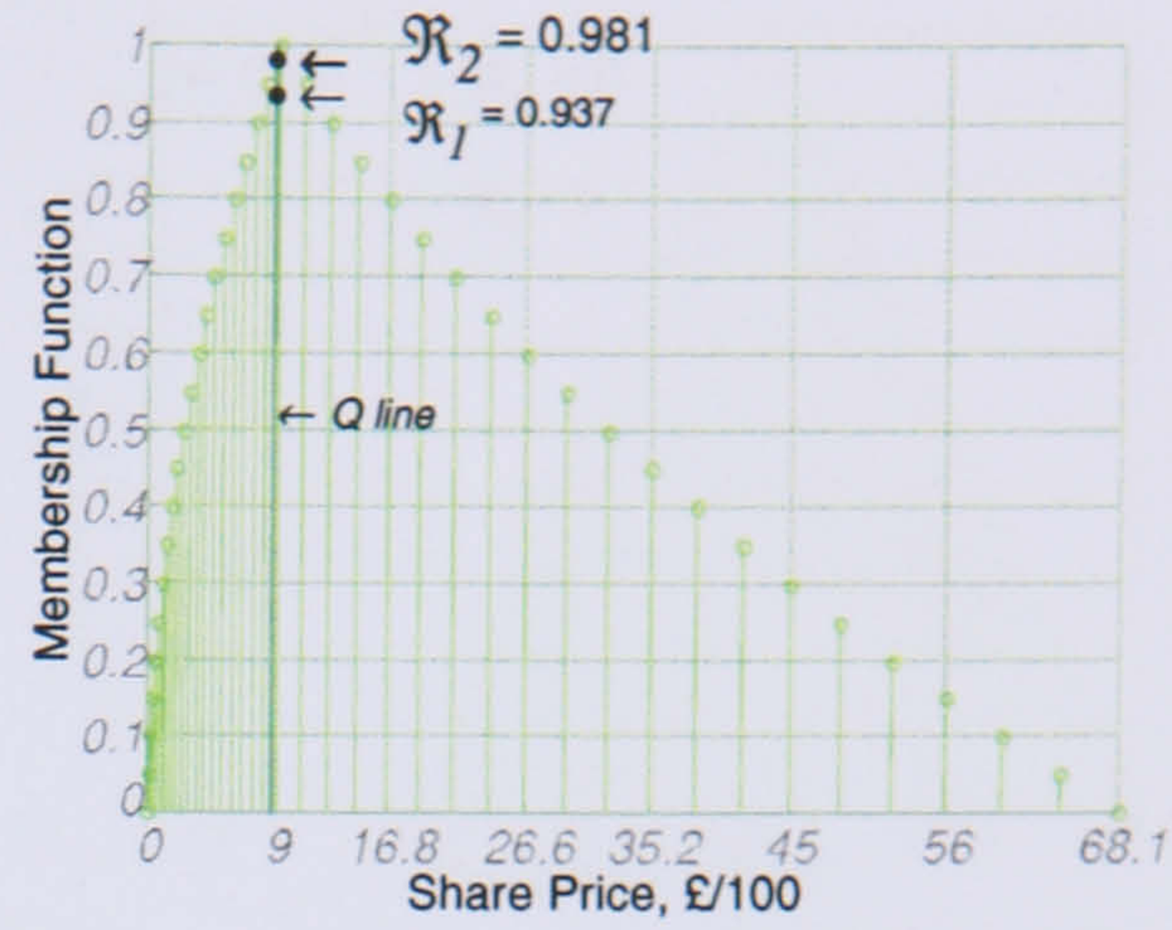


Figure A3.28b: TAYLOR WOODROW -
evaluated robustness $\Delta = 0.956$

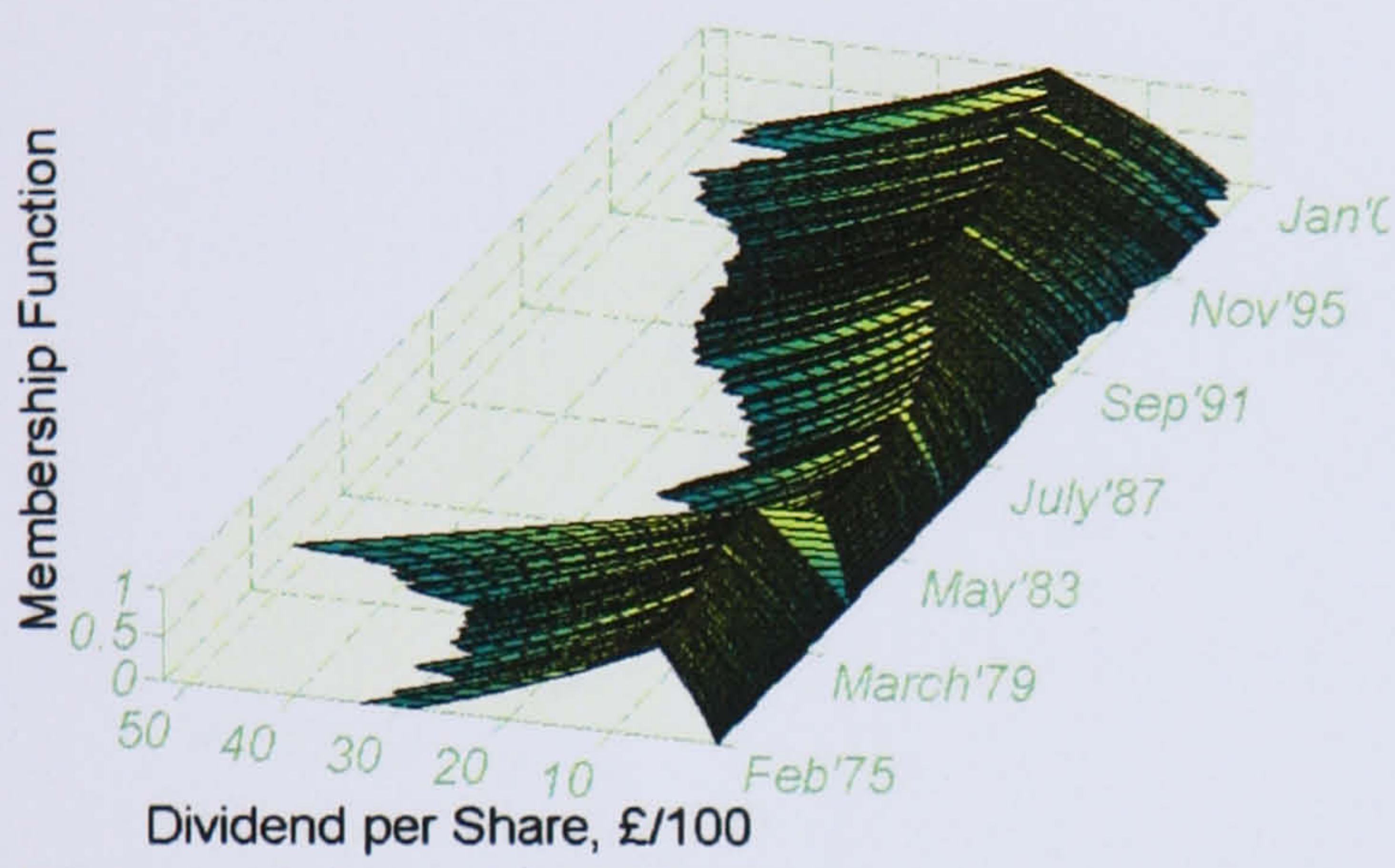


Figure A3.29a: TI GROUP -
fuzzified data under a broader range of imprecision

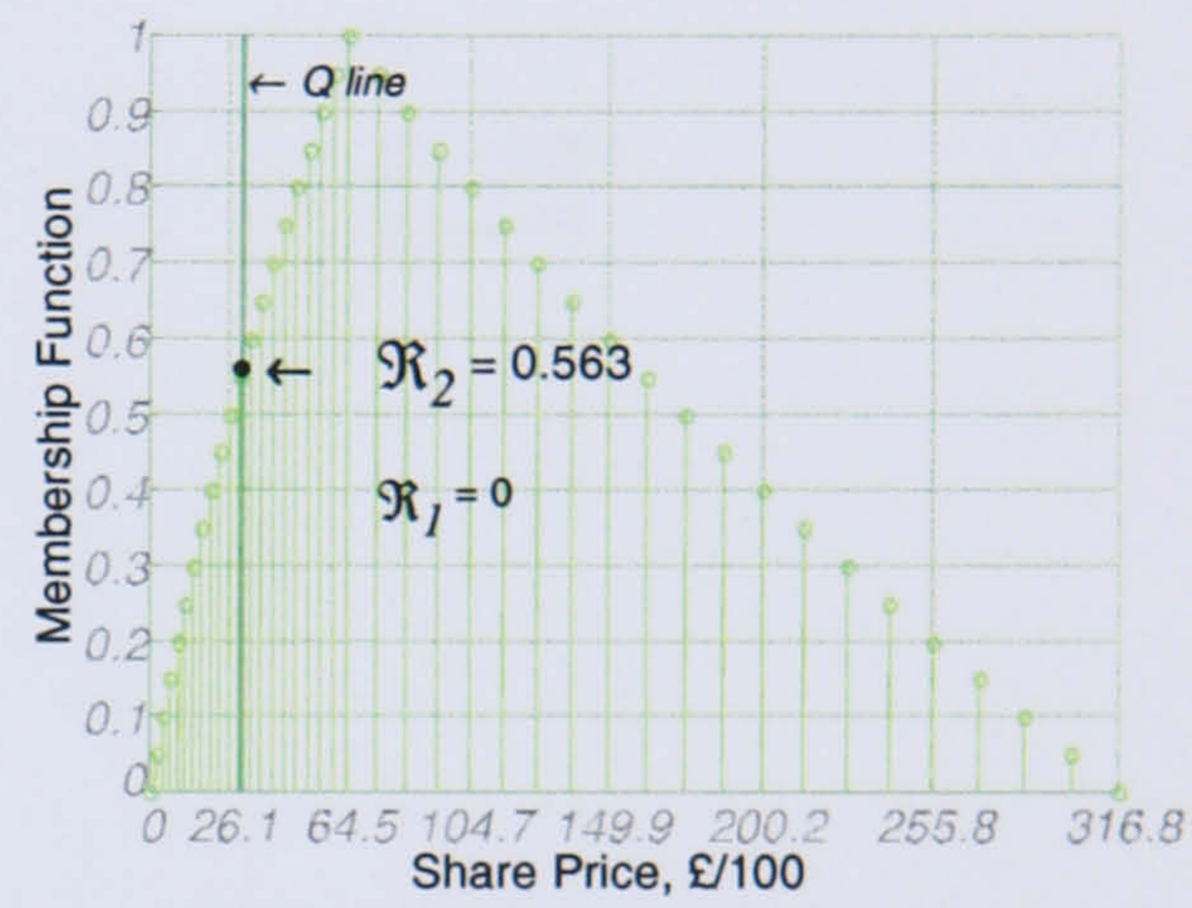


Figure A3.29b: TI GROUP -
evaluated robustness $\Delta = 0.437$

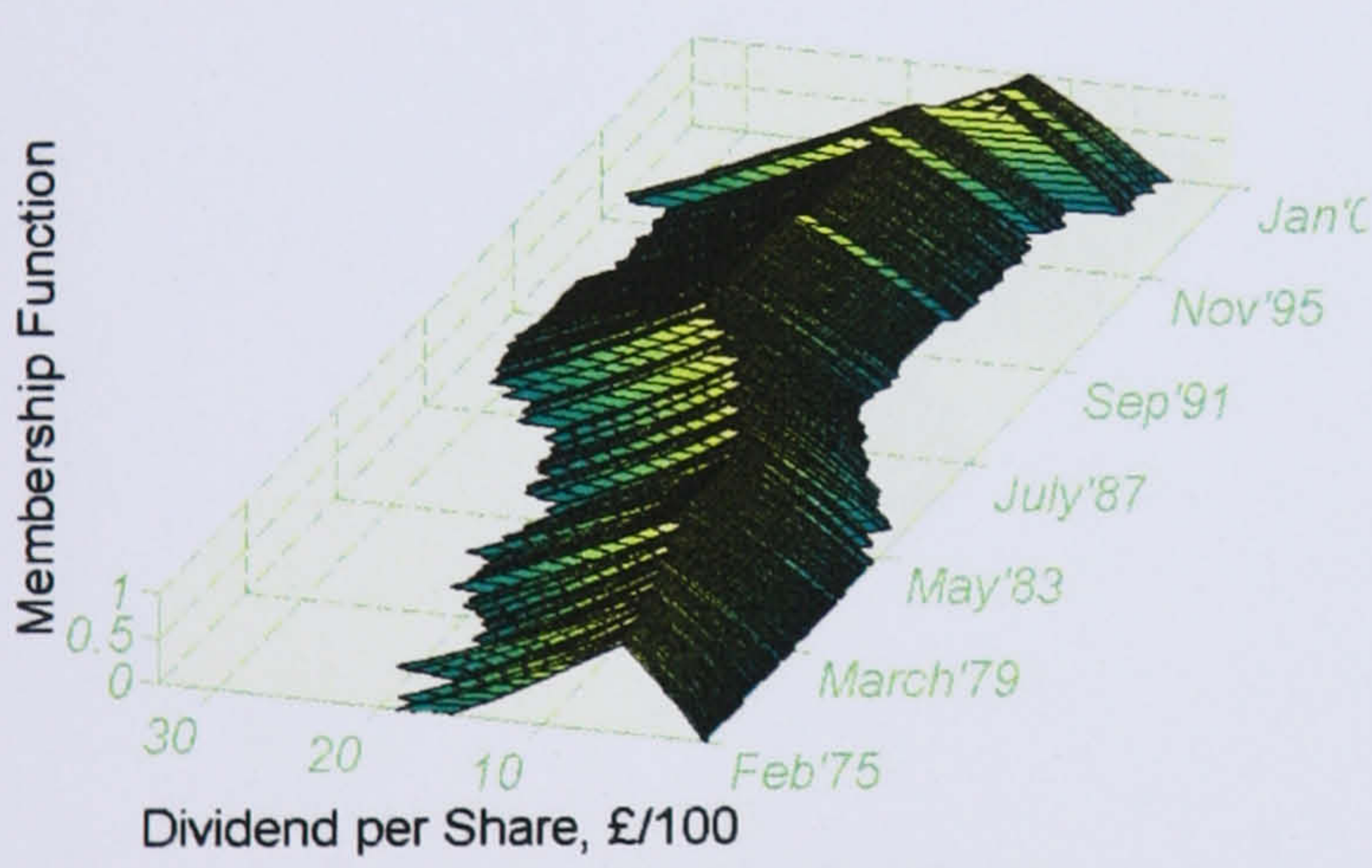


Figure A3.30a: TRANSPORT DEVELOPMENT GROUP -
fuzzified data under a broader range of imprecision

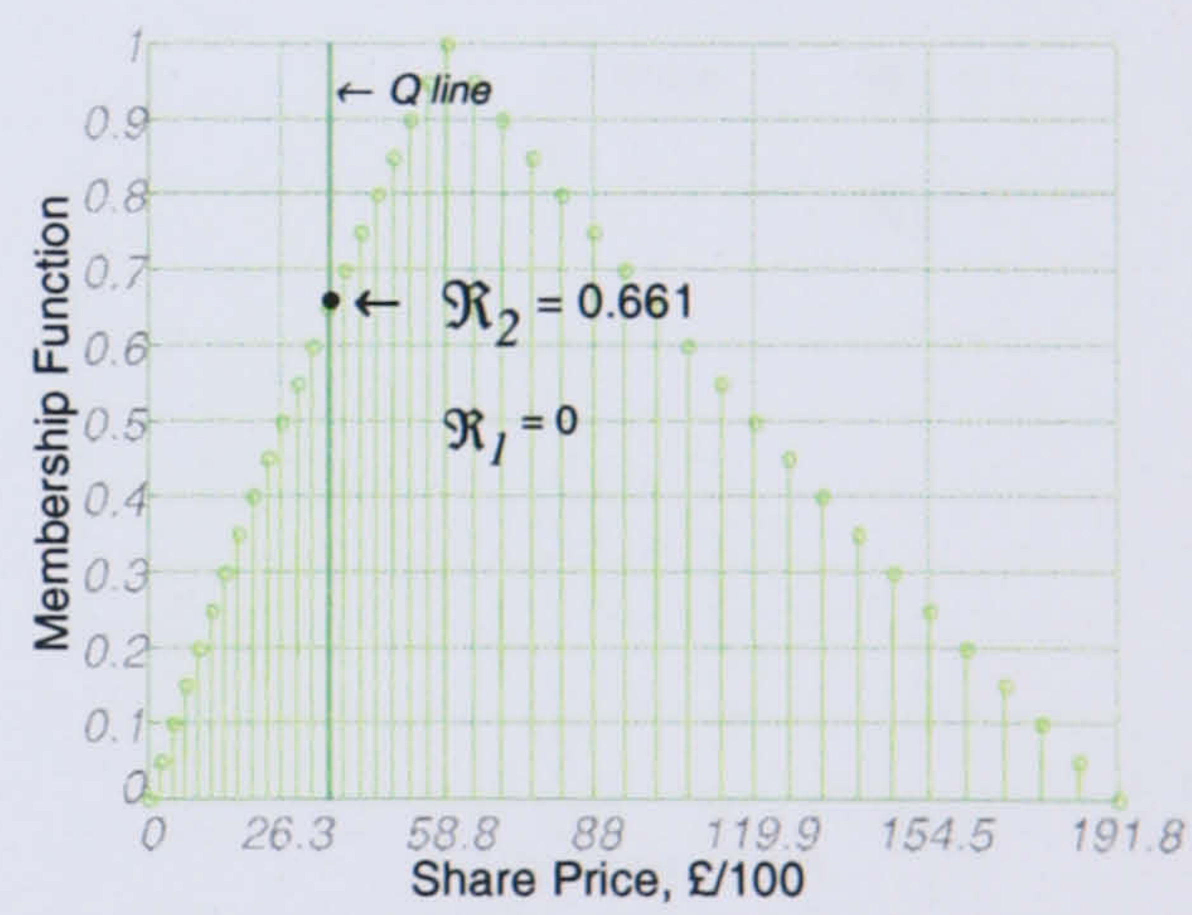


Figure A3.30b: TRANSPORT DEVELOPMENT GRC
evaluated robustness $\Delta = 0.339$

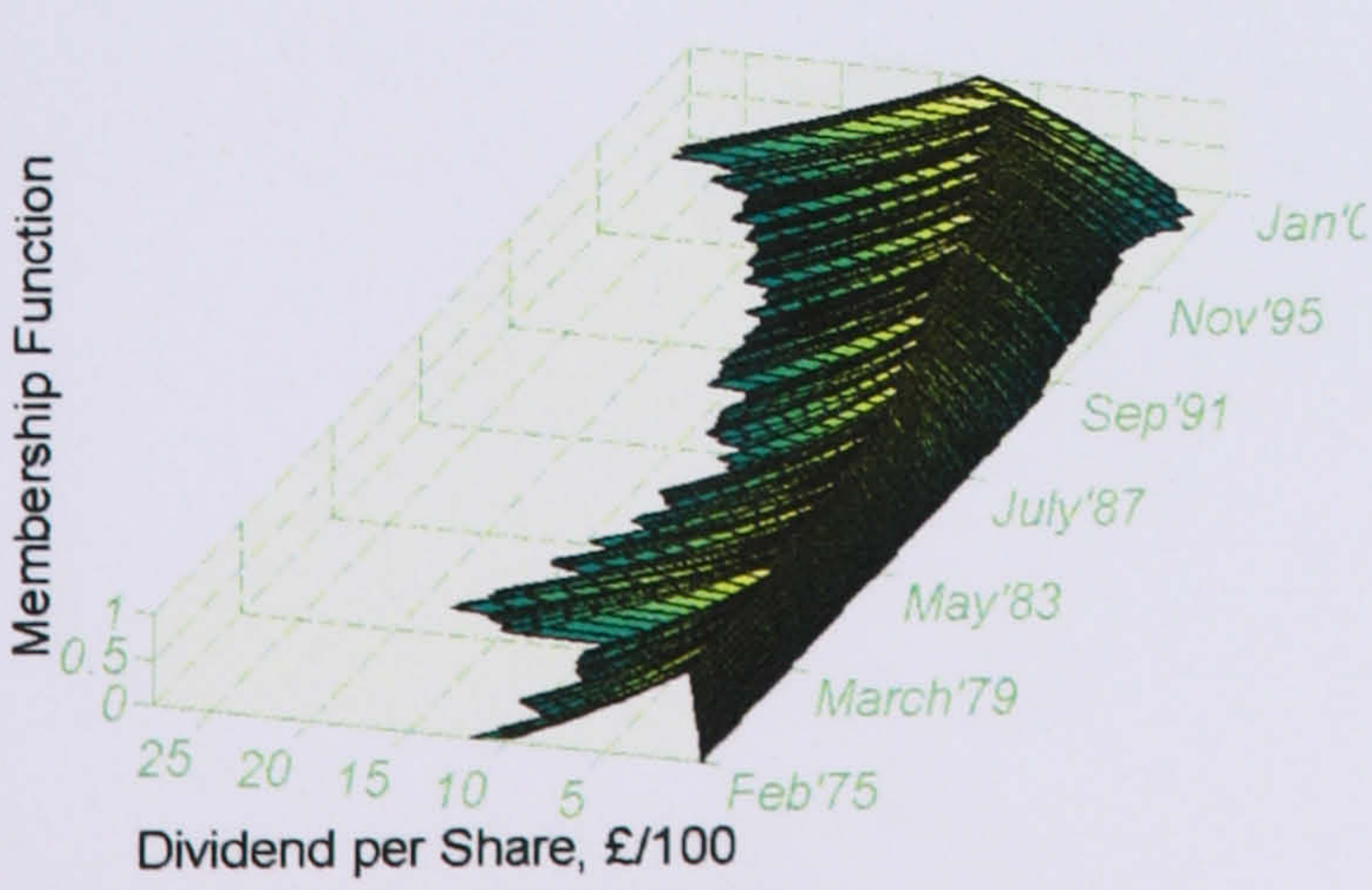


Figure A3.31a: UNILEVER -
fuzzified data under a broader range of imprecision

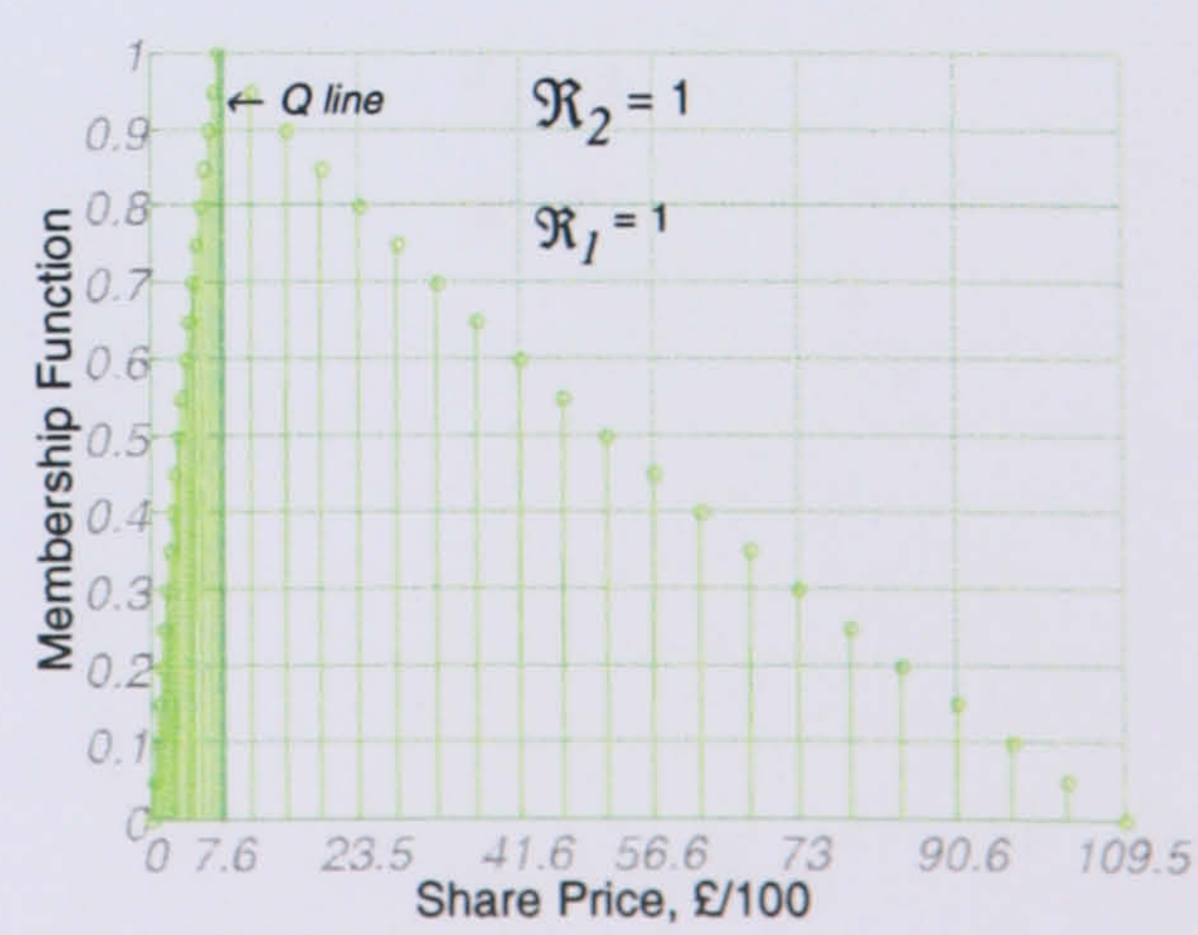


Figure A3.31b: UNILEVER -
no robustness measure is assigned

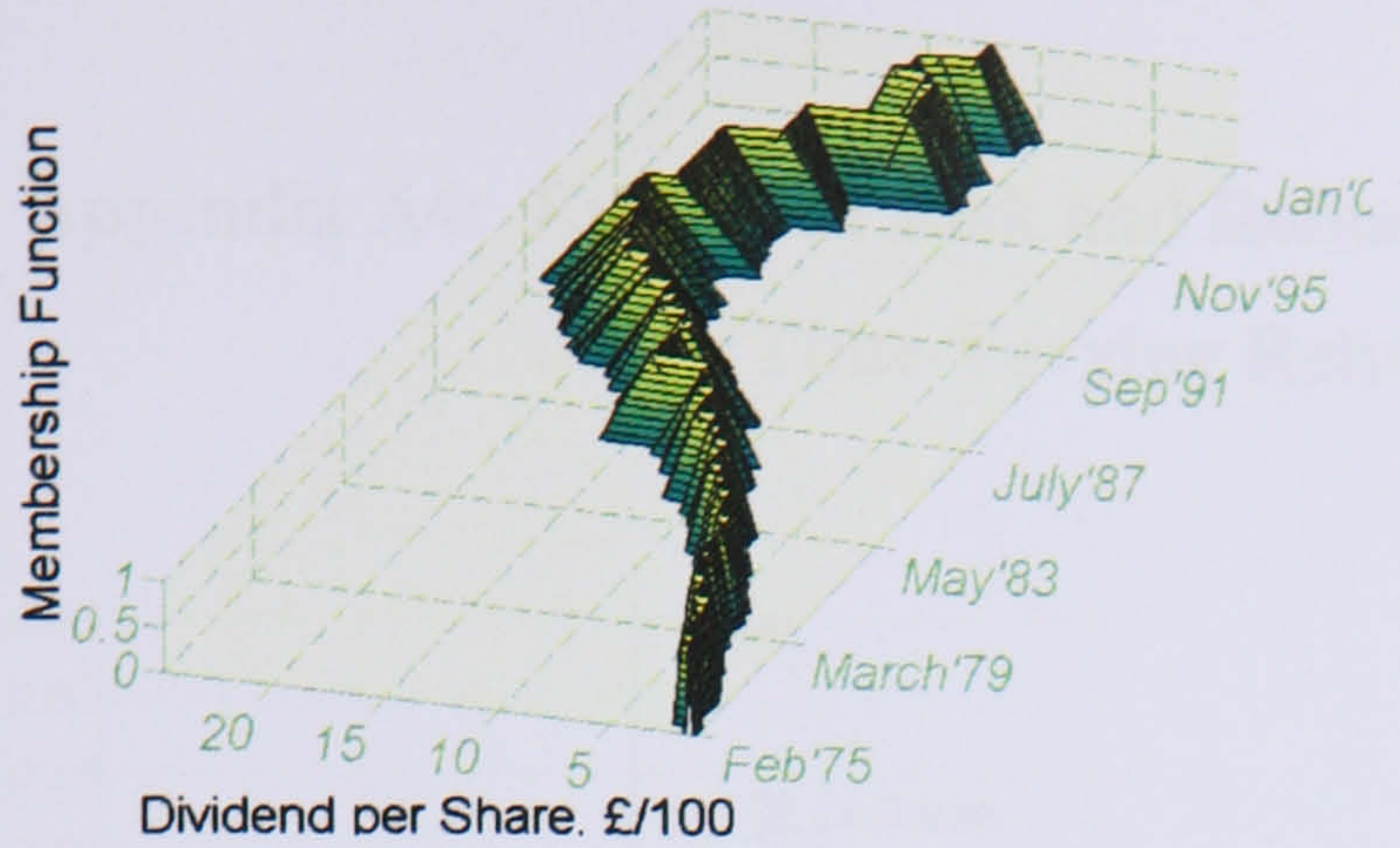


Figure A3.32a: UNITED BISCUITS HOLDINGS - fuzzified data under a broader range of imprecision

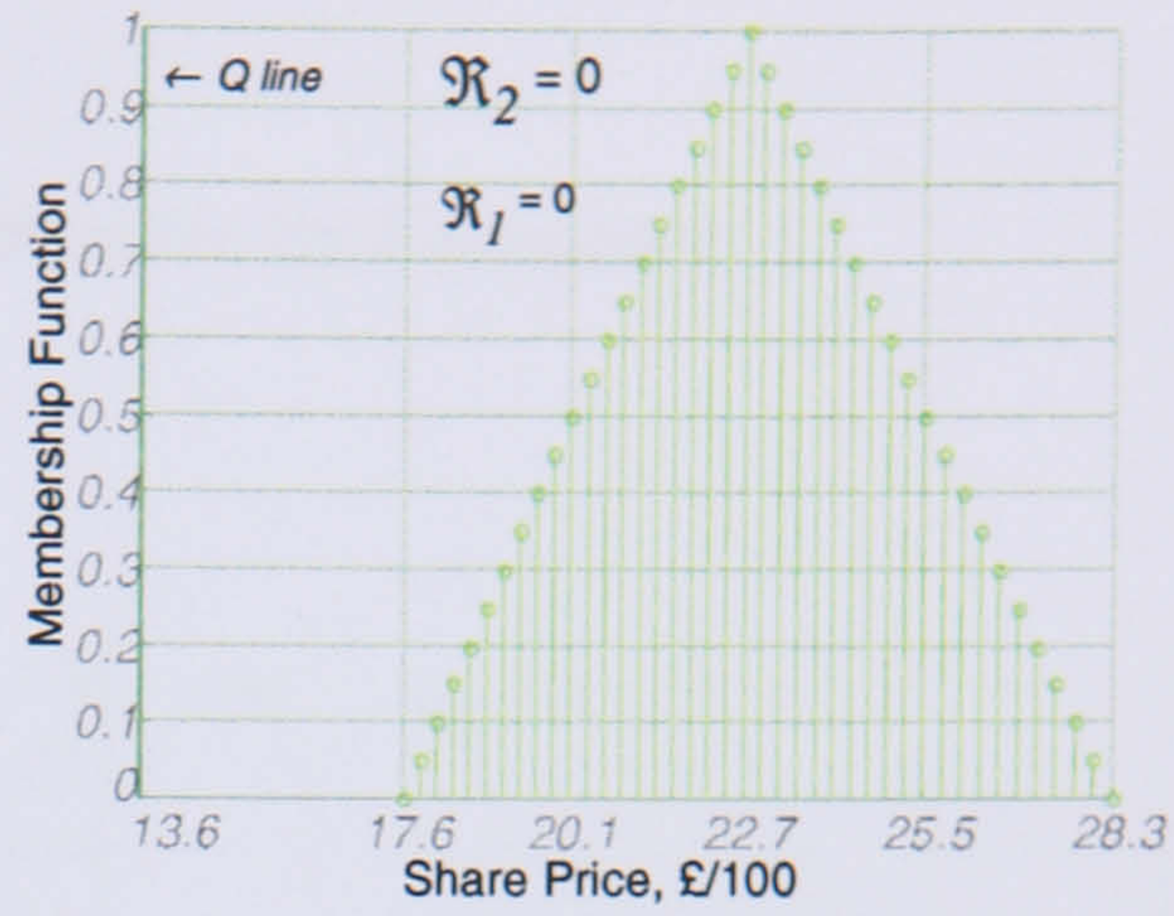


Figure A3.32b: UNITED BISCUITS HOLDINGS - evaluated robustness $\Delta = 1$

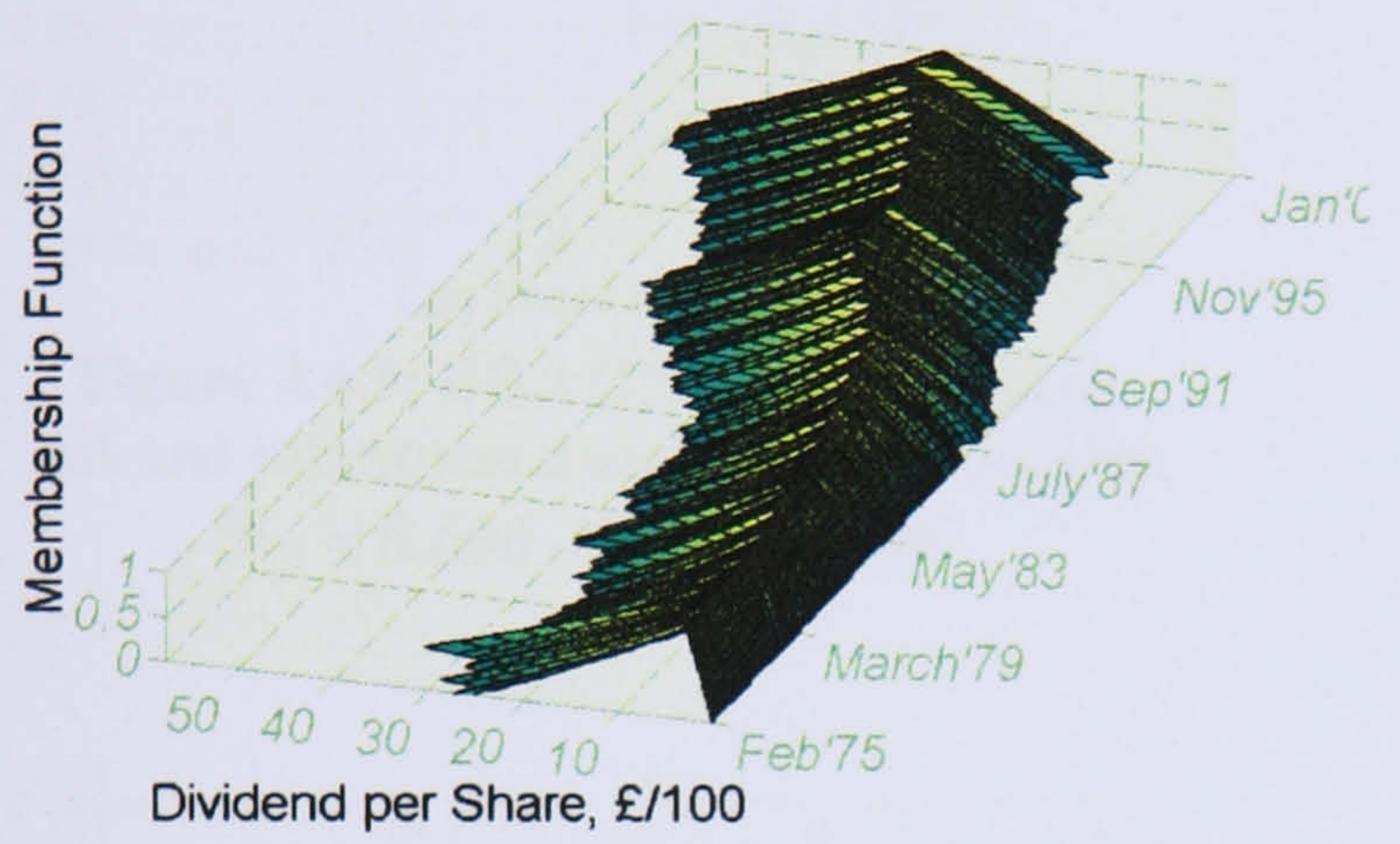


Figure A3.33a: WHITBREAD - fuzzified data under a broader range of imprecision

Figure A3.33b: WHITBREAD - evaluated robustness $\Delta = 0.959$

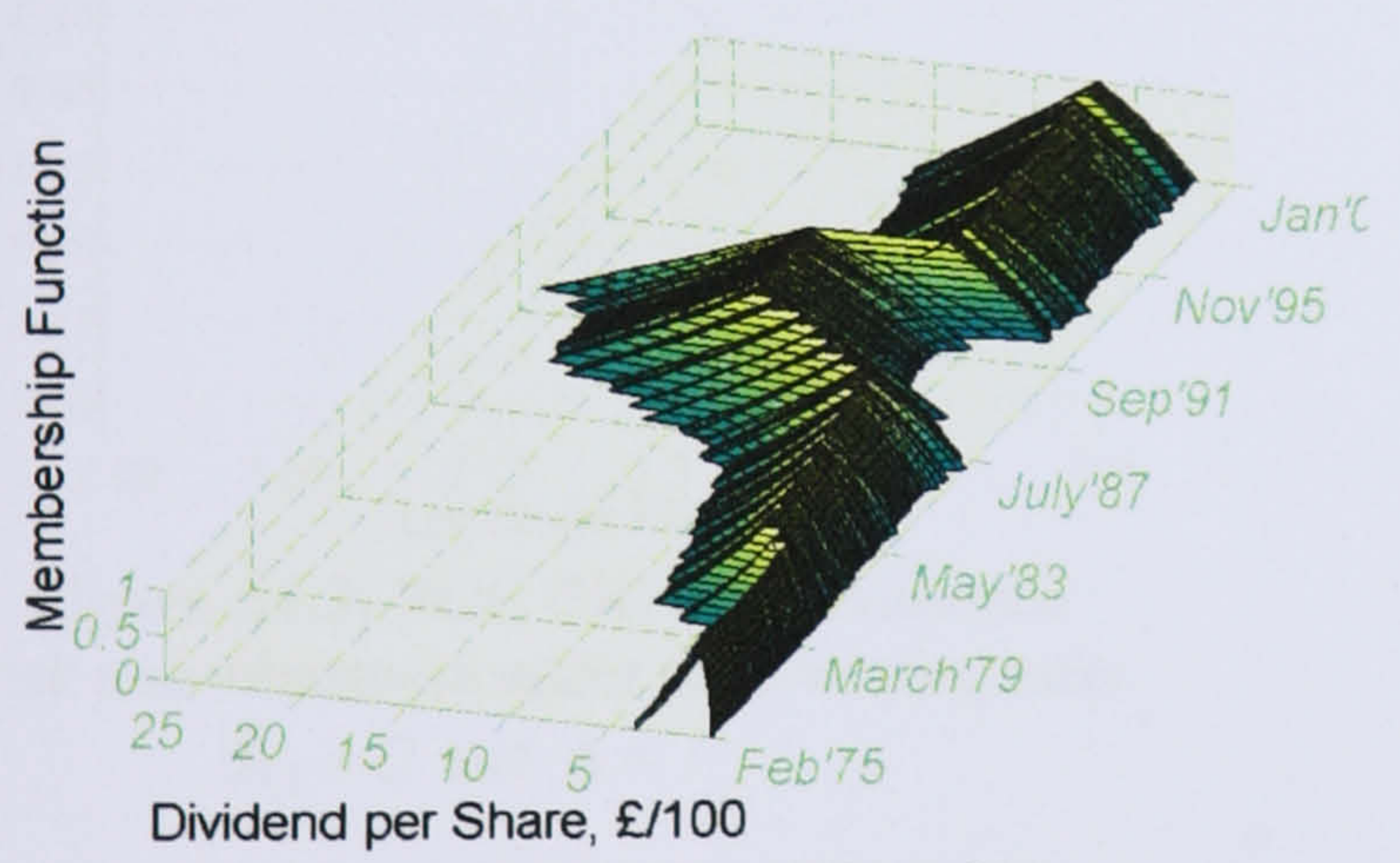


Figure A3.34a: WIMPEY (GEORGE) - fuzzified data under a broader range of imprecision

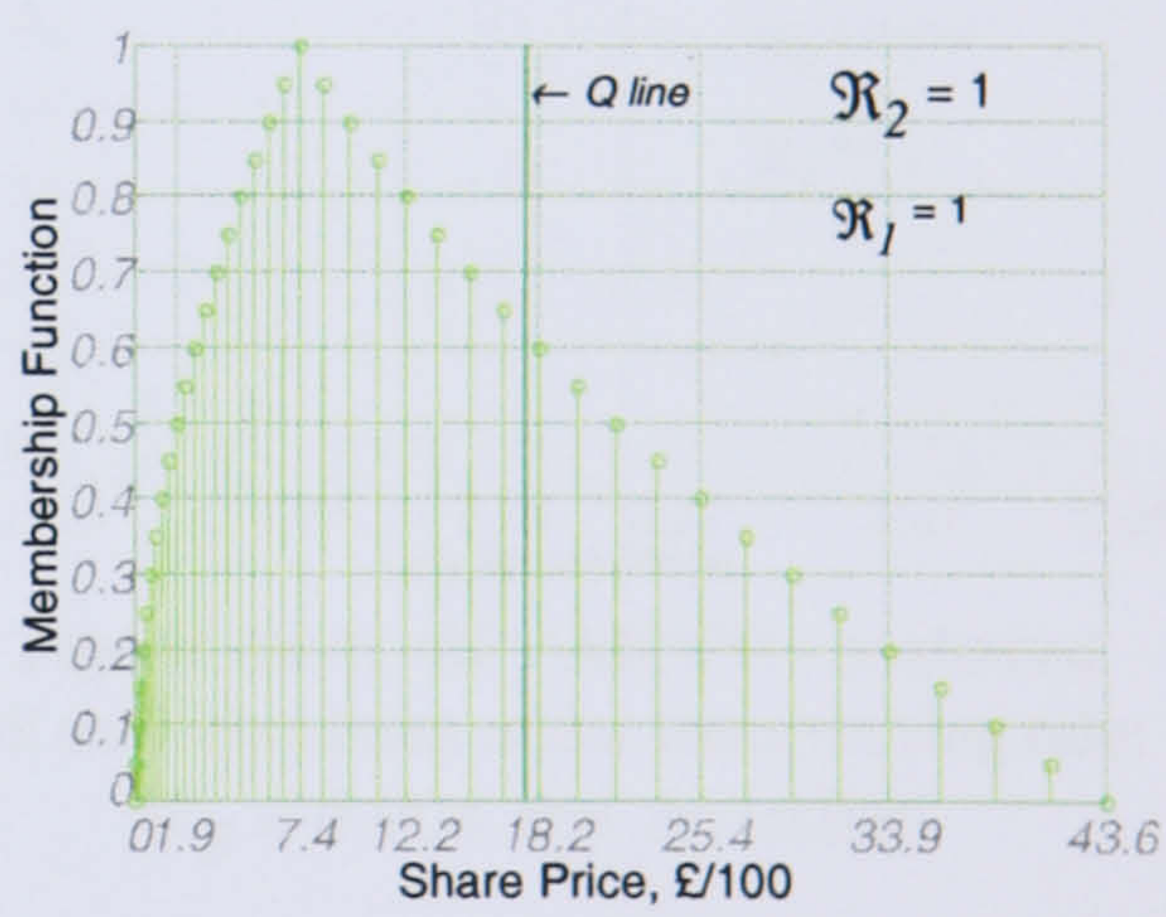


Figure A3.34b: WIMPEY (GEORGE) - no robustness measure is assigned

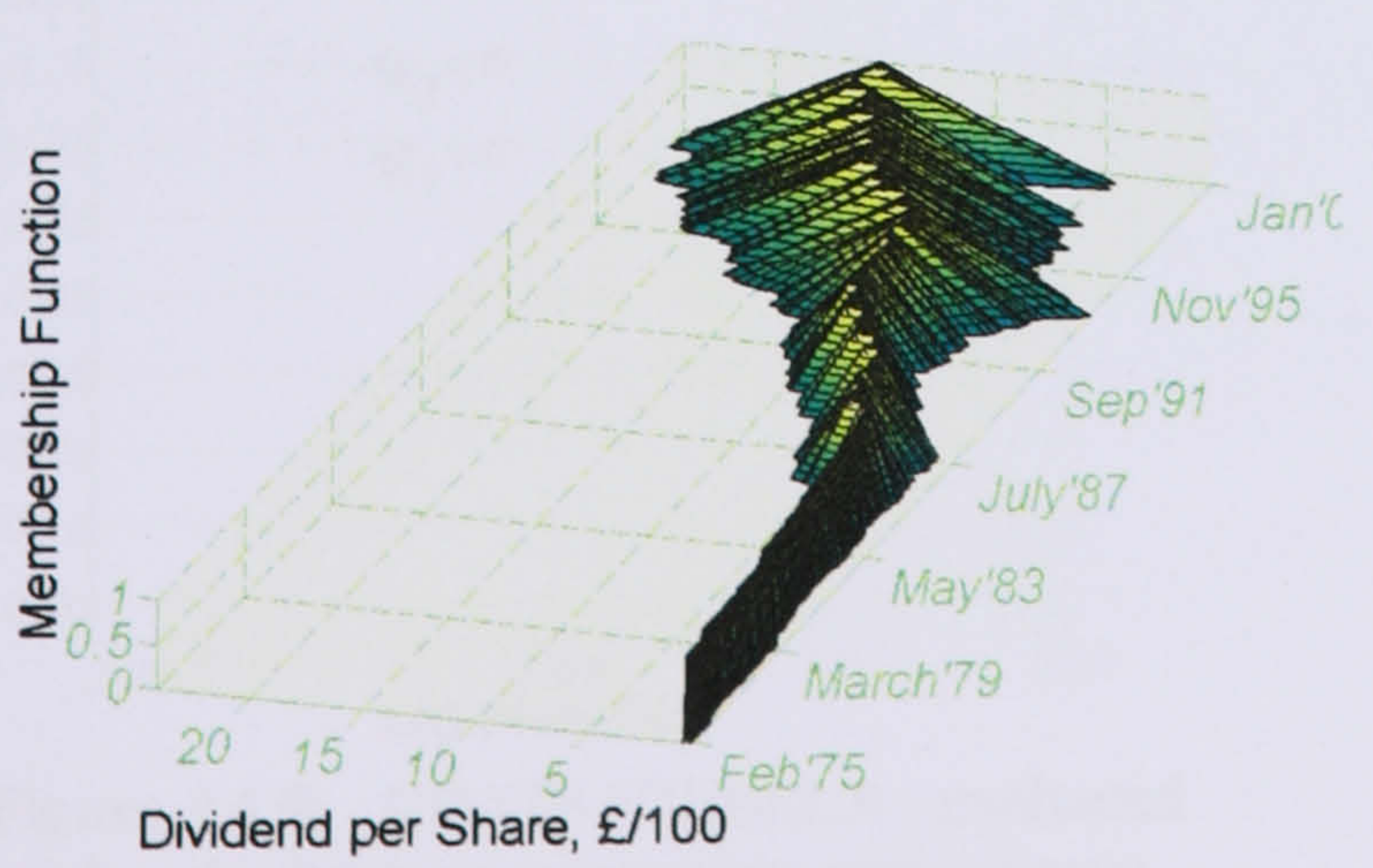


Figure A3.35a: WOLSELEY - fuzzified data under a broader range of imprecision

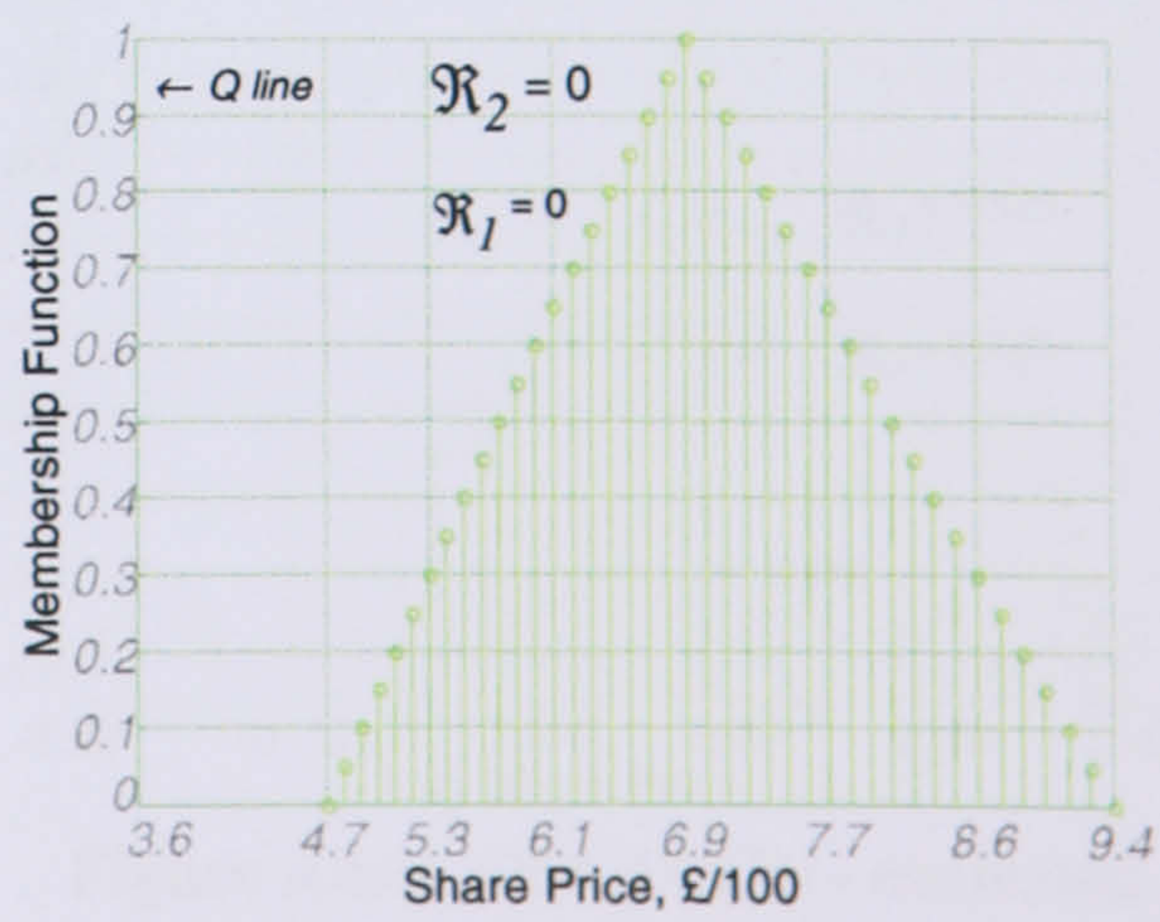


Figure A3.35b: WOLSELEY - evaluated robustness $\Delta = 1$

Appendix A4: Evaluated Risk and Robustness Measures by Company under Time-Varying Return

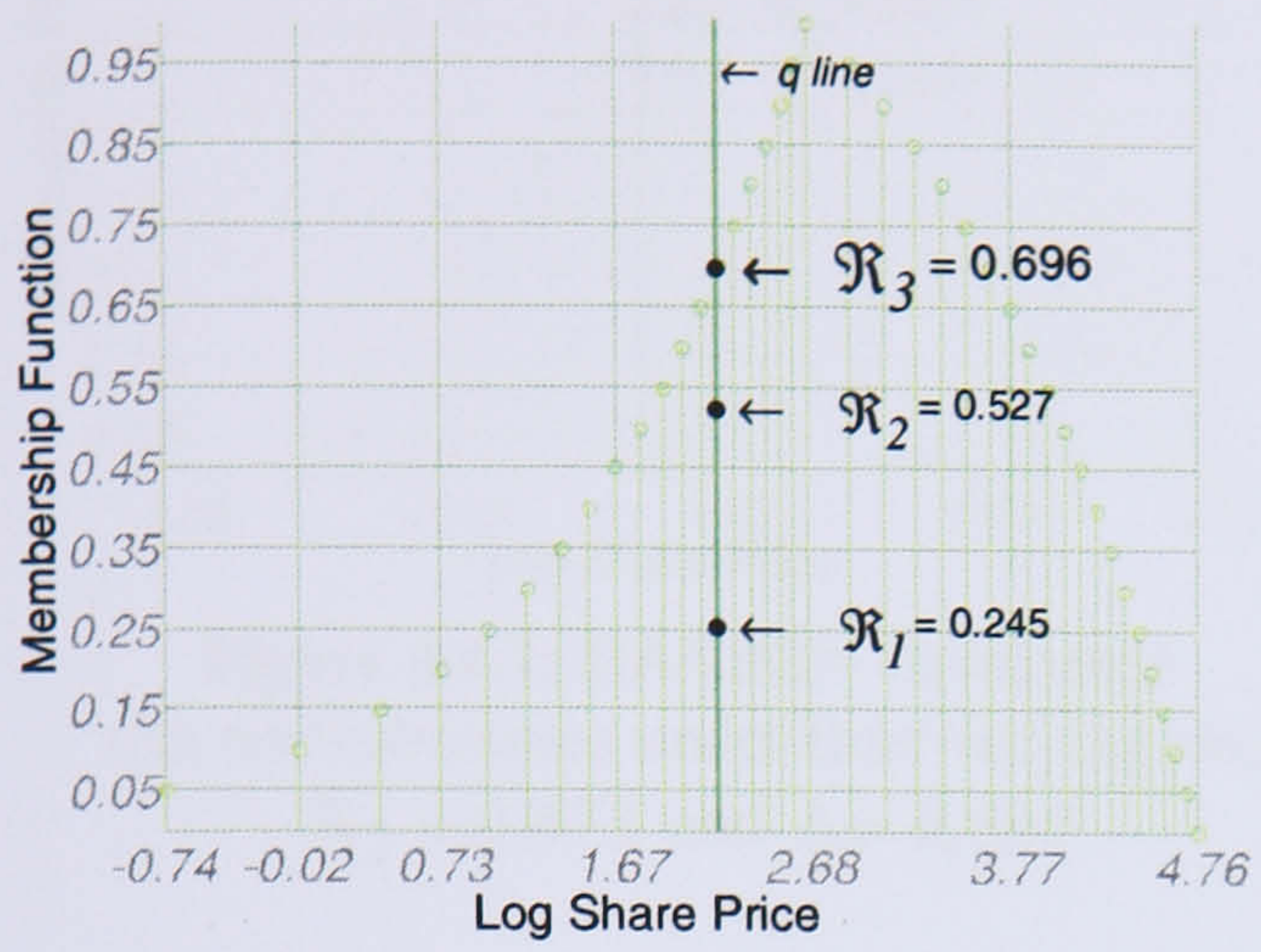


Figure A4.1: BBA GROUP - evaluated risk and robustness under time-varying rate: $\Re_3 = 0.696$ and $\Delta = 0.549$

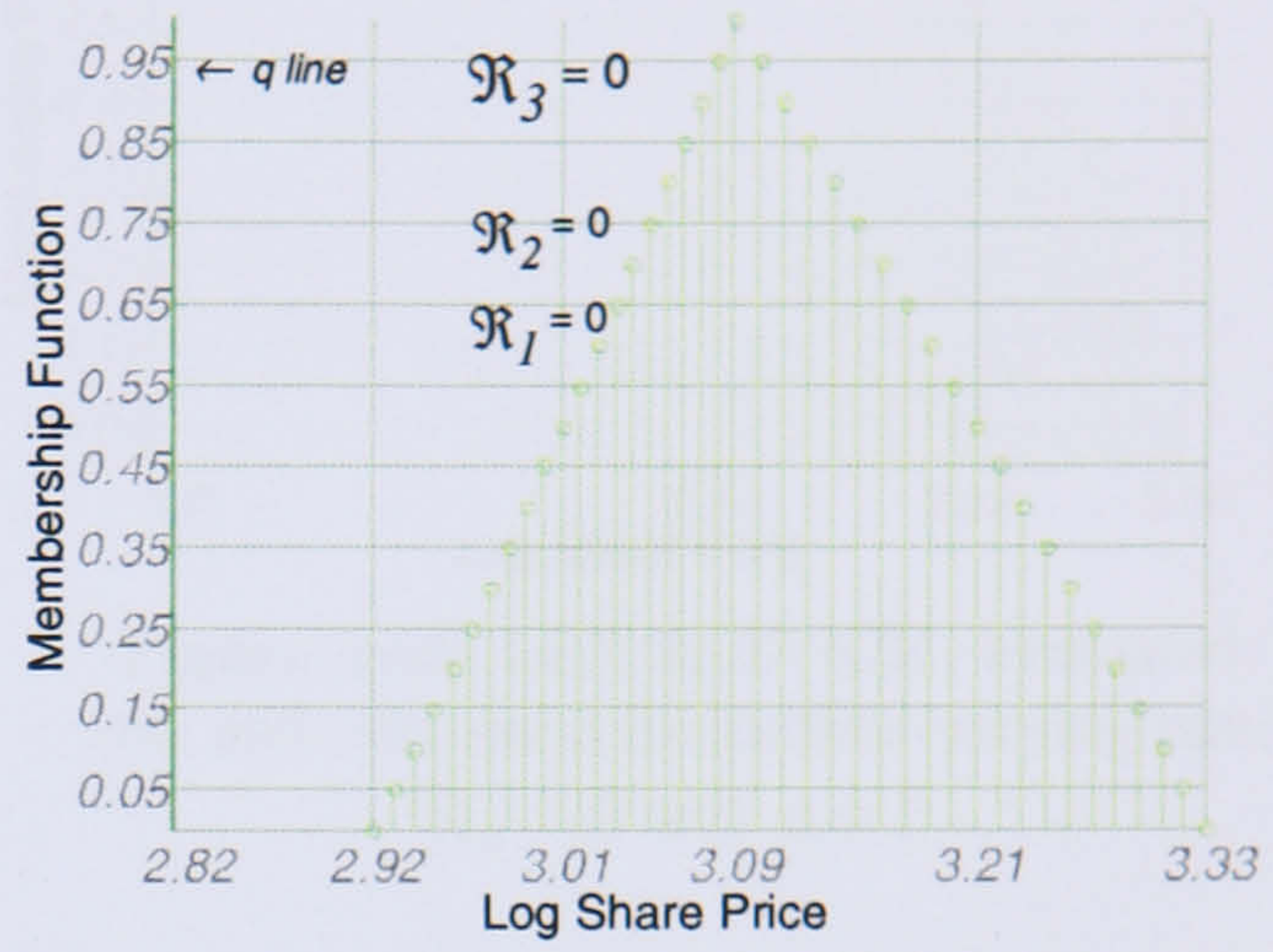


Figure A4.2: BLUE CIRCLE INDUSTRIES - evaluated risk and robustness under time-varying rate: $\Re_3 = 0$ and $\Delta = 1$

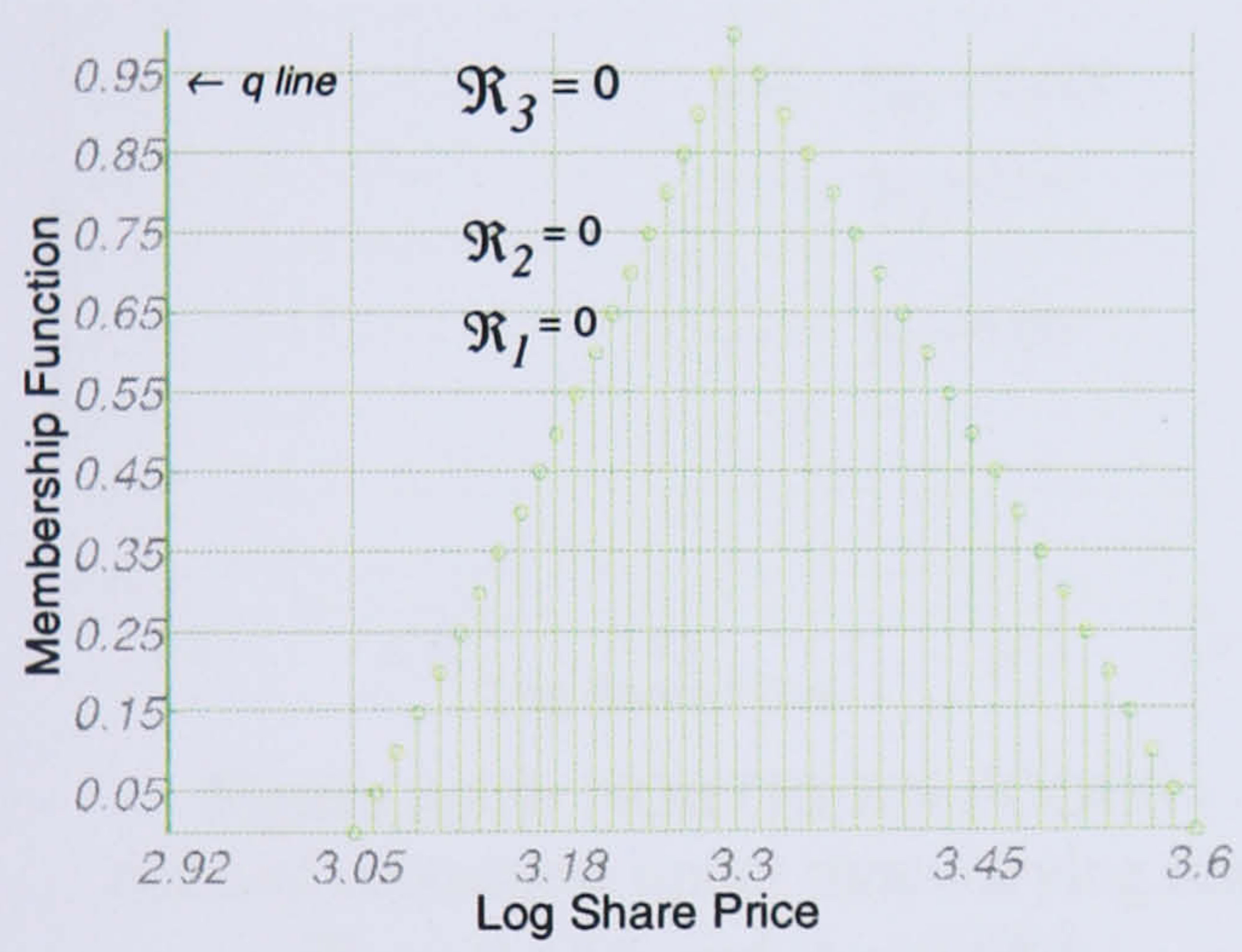


Figure A4.3: BOC GROUP - evaluated risk and robustness under time-varying rate: $\Re_3 = 0$ and $\Delta = 1$

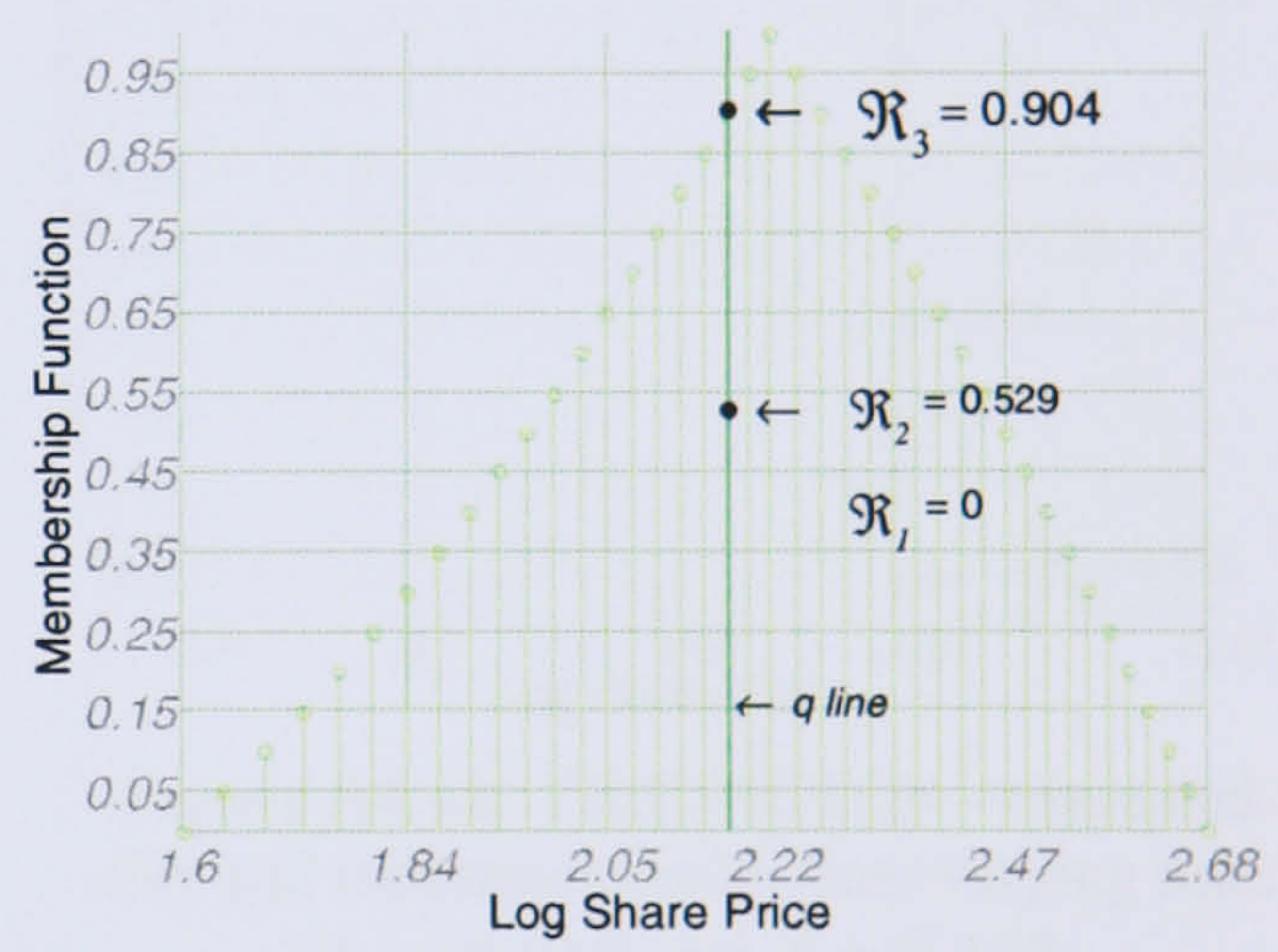


Figure A4.4: BP AMOCO - evaluated risk and robustness under time-varying rate: $\Re_3 = 0.904$ and $\Delta = 0.096$

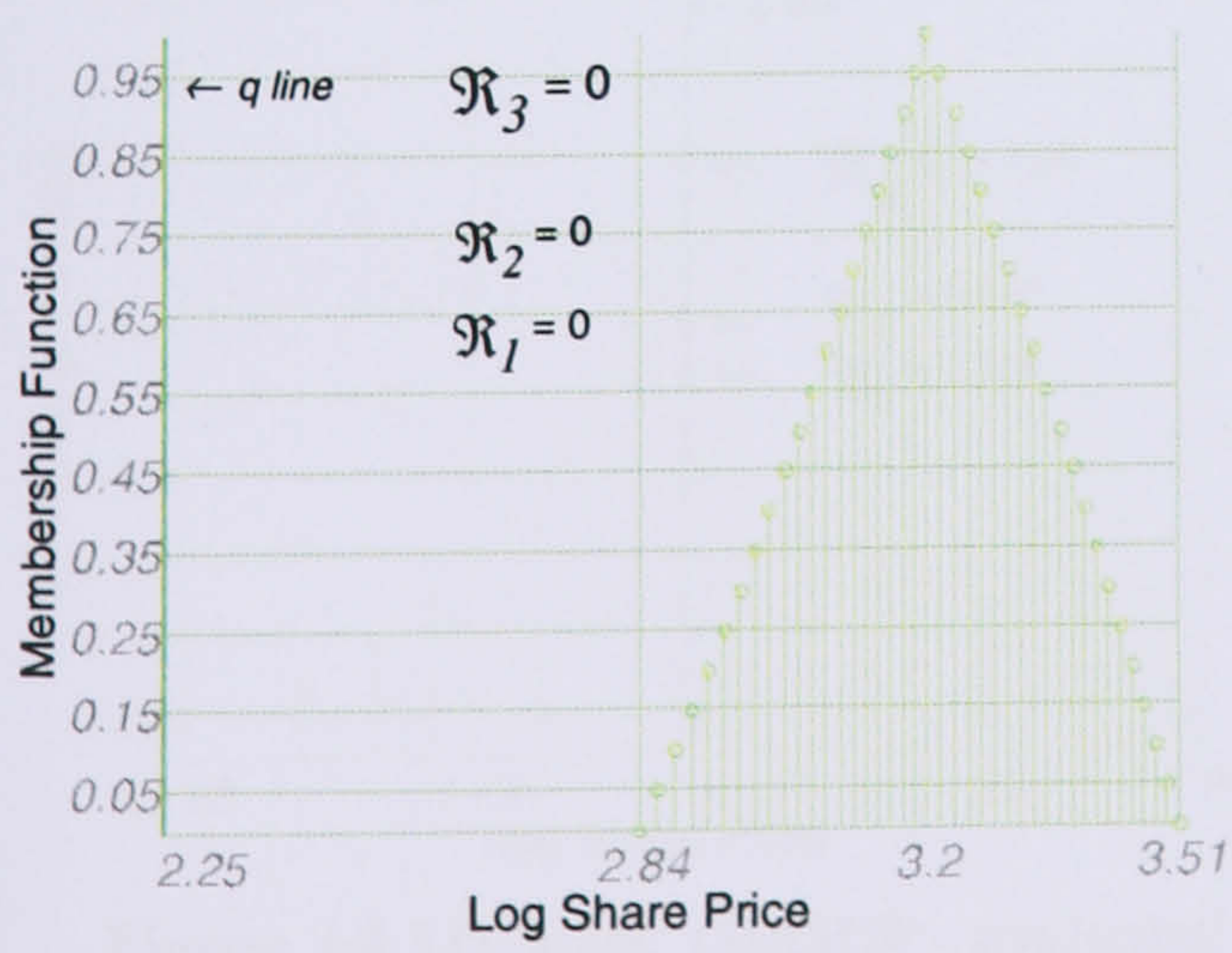


Figure A4.5: COATS VIYELLA - evaluated risk and robustness under time-varying rate: $\Re_3 = 0$ and $\Delta = 1$

Figure A4.6: GOODWIN - evaluated risk and robustness under time-varying rate: $\Re_3 = 0.925$ and $\Delta = 0.496$

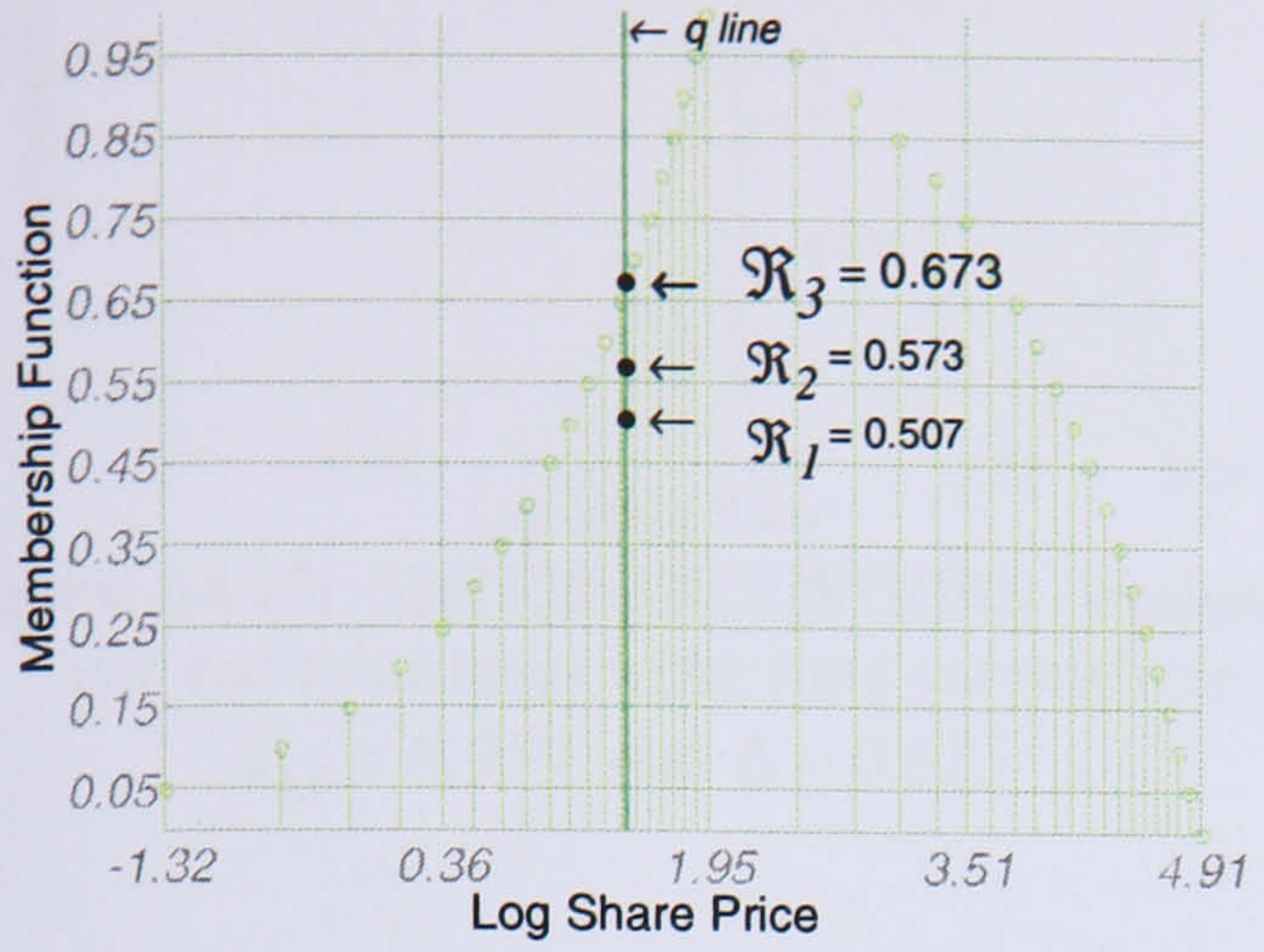


Figure A4.7: HANSON - evaluated risk and robustness under time-varying rate: $\Re_3 = 0.673$ and $\Delta = 0.834$

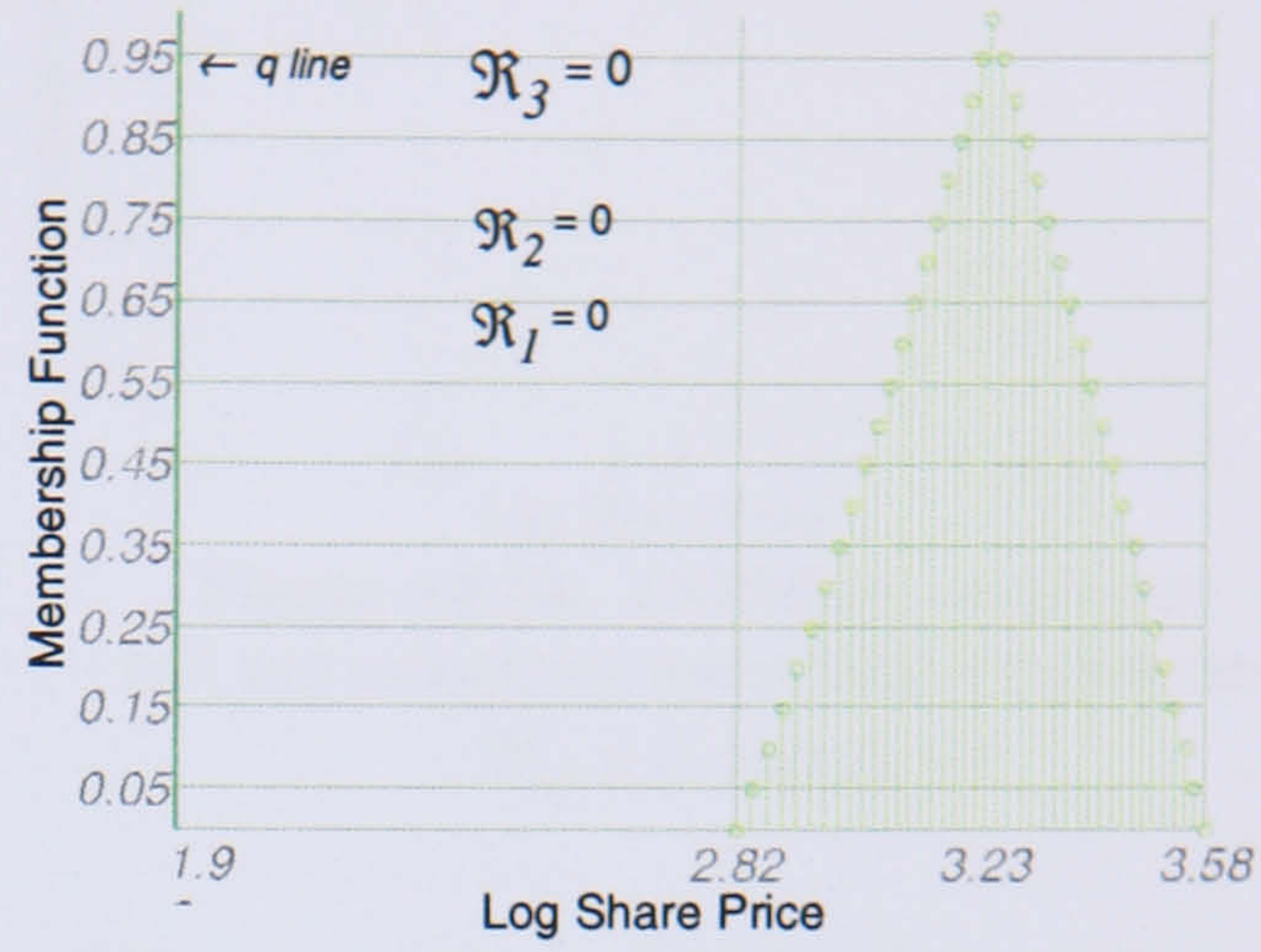


Figure A4.8: LEX SERVICE - evaluated risk and robustness under time-varying rate: $\Re_3 = 0$ and $\Delta = 1$

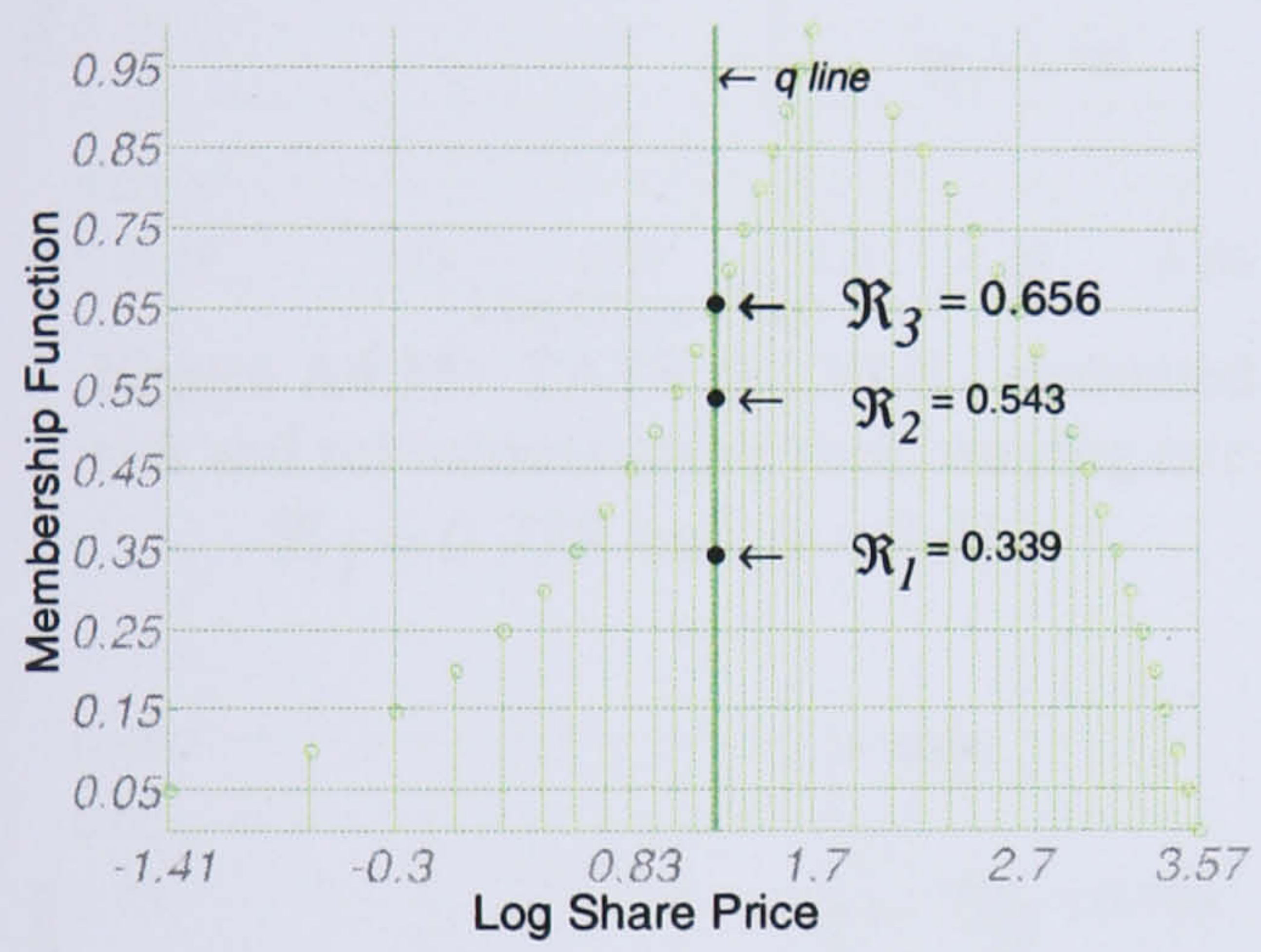


Figure A4.9: NORTHERN FOODS - risk and robustness under time-varying rate: $\Re_3 = 0.656$ and $\Delta = 0.683$

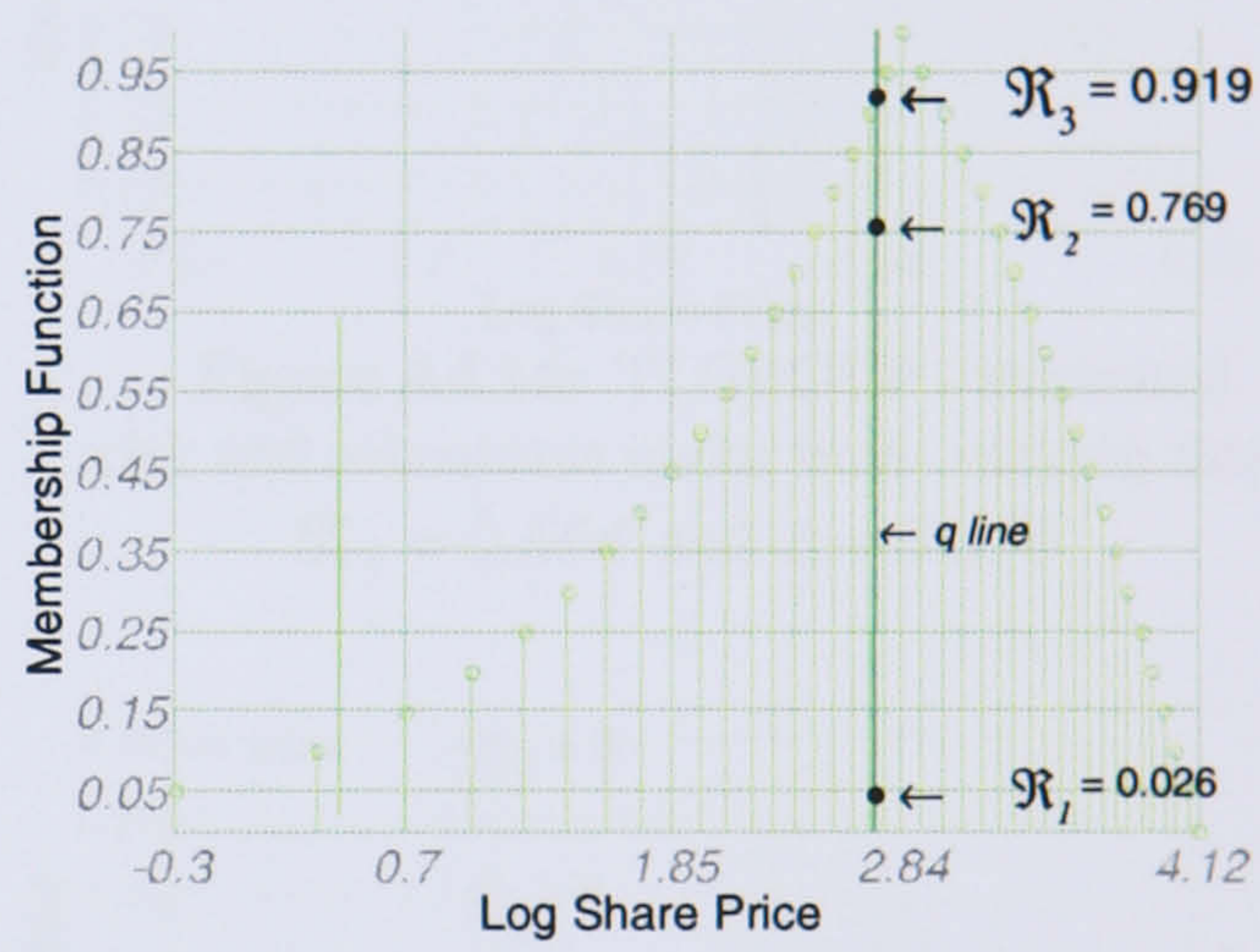


Figure A4.10: PILKINGTON - evaluated risk and robustness under time-varying rate: $\Re_3 = 0.919$ and $\Delta = 0.107$

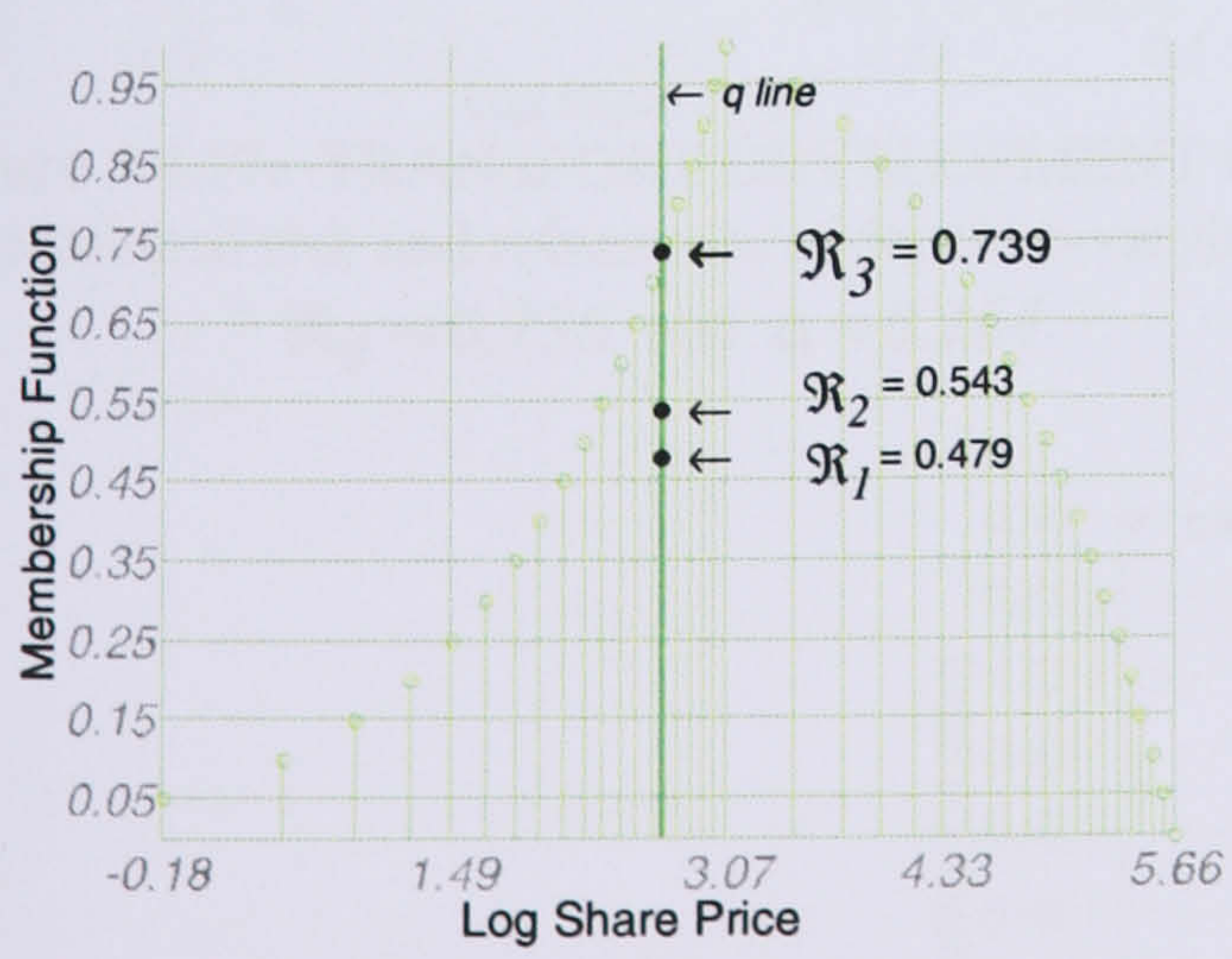


Figure A4.11: RMC GROUP - evaluated risk and robustness under time-varying rate: $\Re_3 = 0.739$ and $\Delta = 0.740$

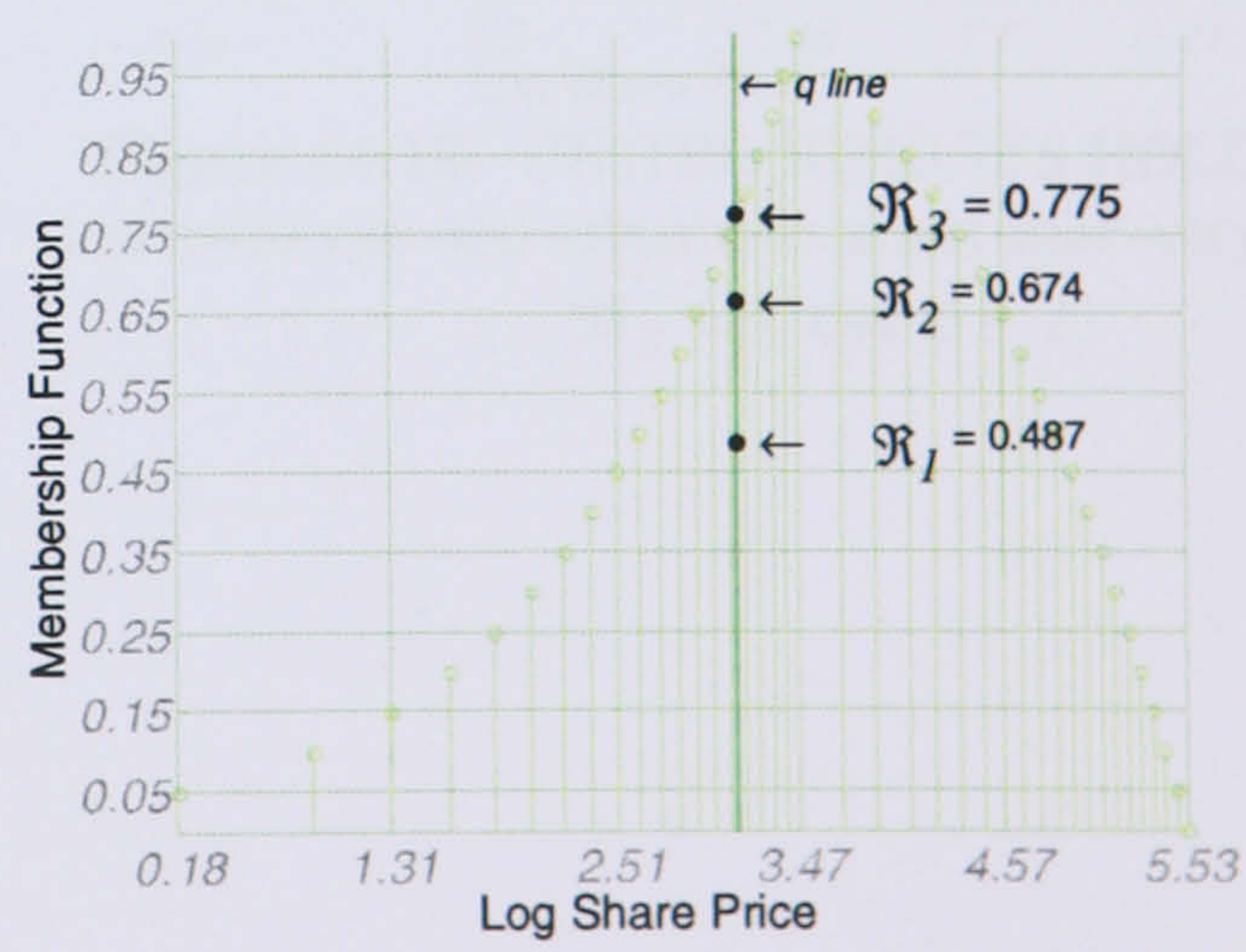


Figure A4.12: SCOTTISH & NEWCASTLE - evaluated risk and robustness under time-varying rate: $\Re_3 = 0.775$ and $\Delta = 0.712$

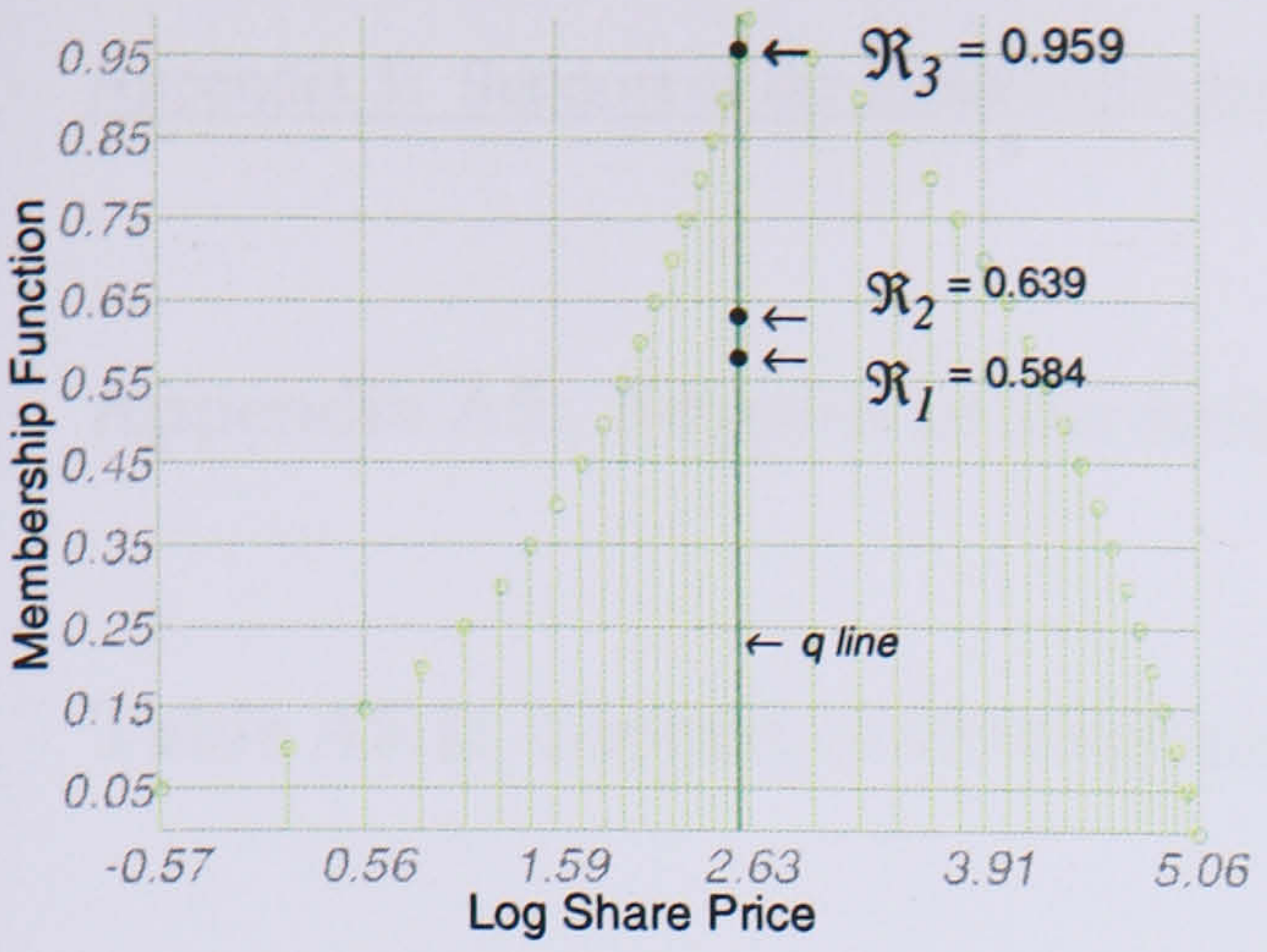


Figure A4.13: SMITHS INDUSTRIES - evaluated risk and robustness under time-varying rate: $\Re_3 = 0.959$ and $\Delta = 0.625$

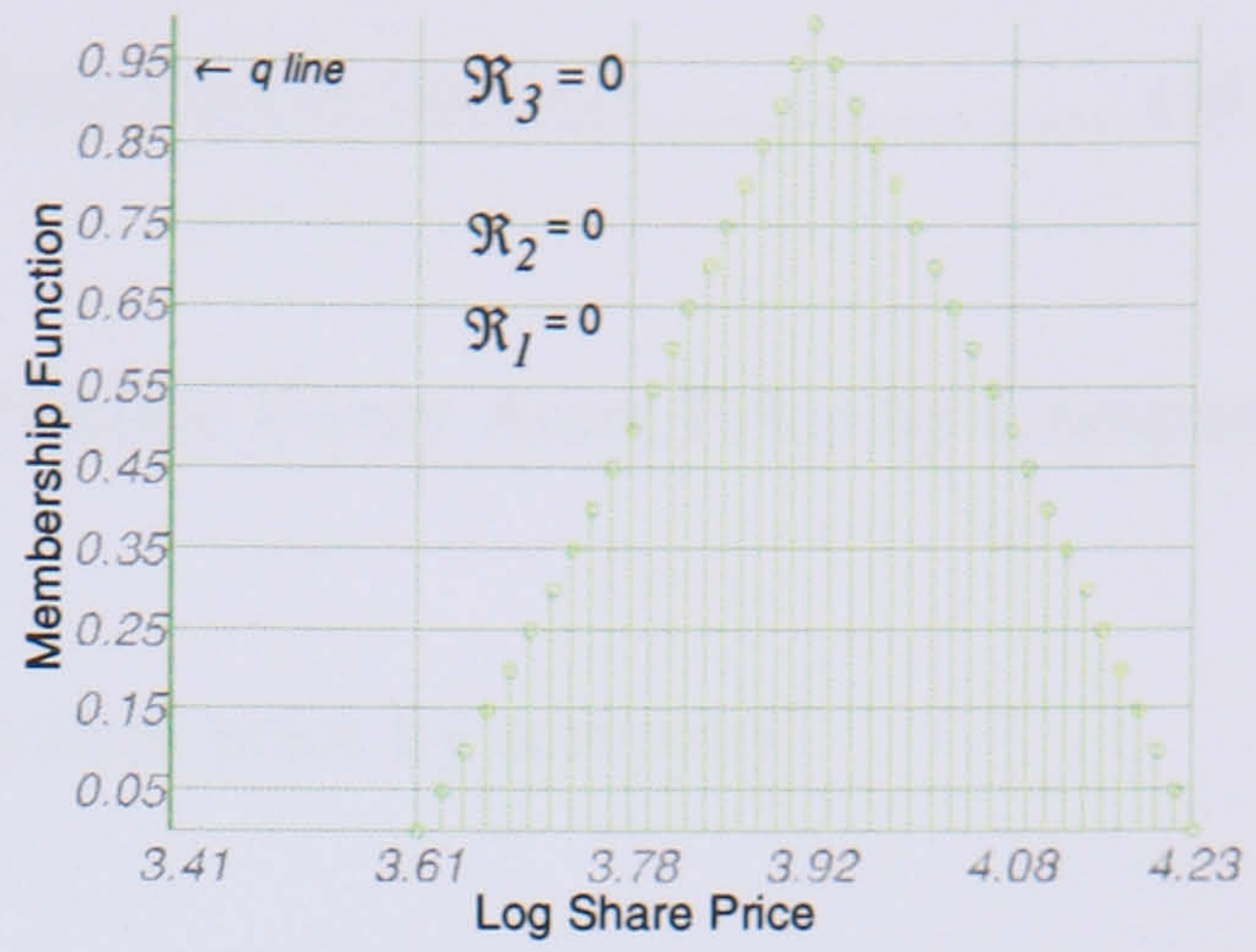


Figure A4.14: TARMAC - evaluated risk and robustness under time-varying rate: $\Re_3 = 0$ and $\Delta = 1$

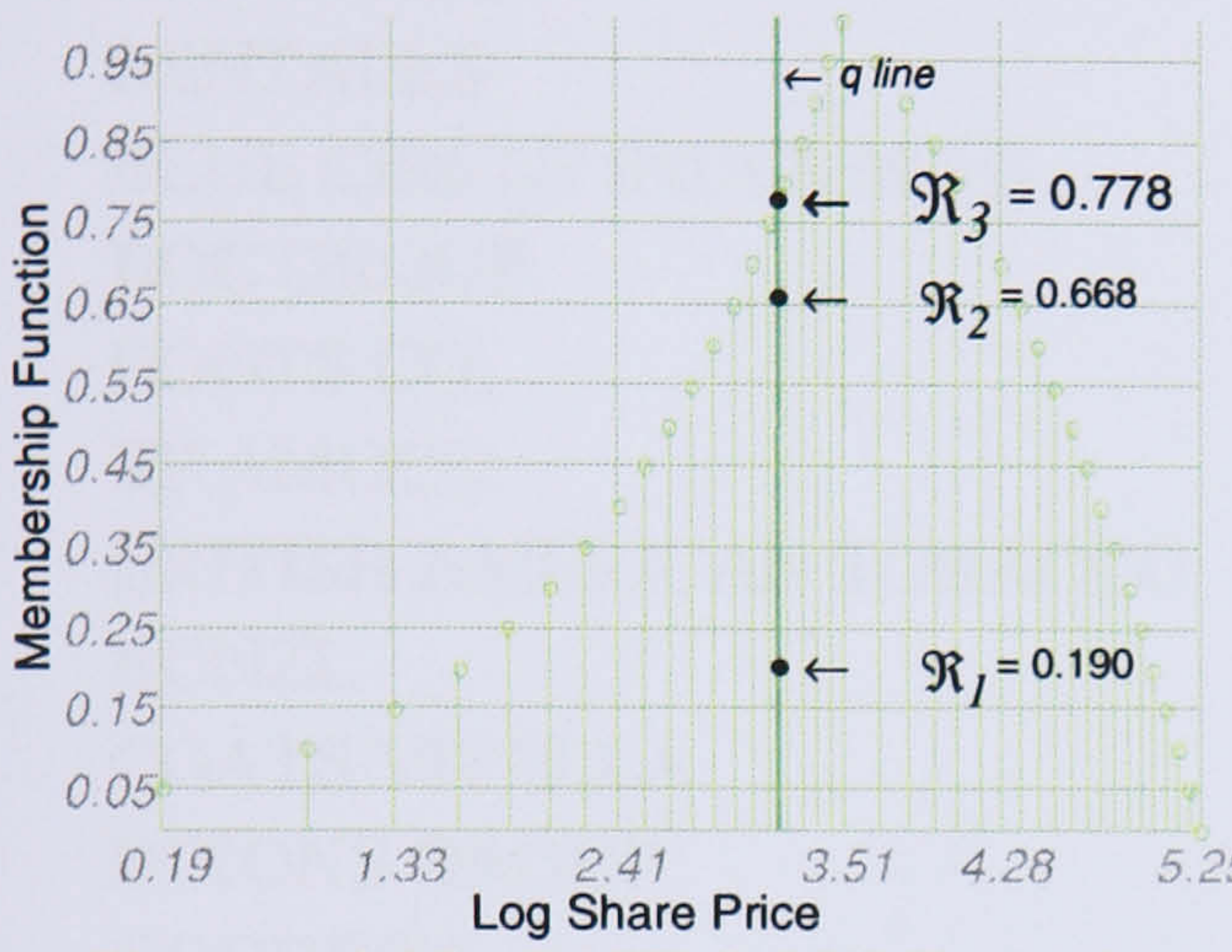


Figure A4.15: TATE & LYLE - evaluated risk and robustness under time-varying rate: $\Re_3 = 0.778$ and $\Delta = 0.412$

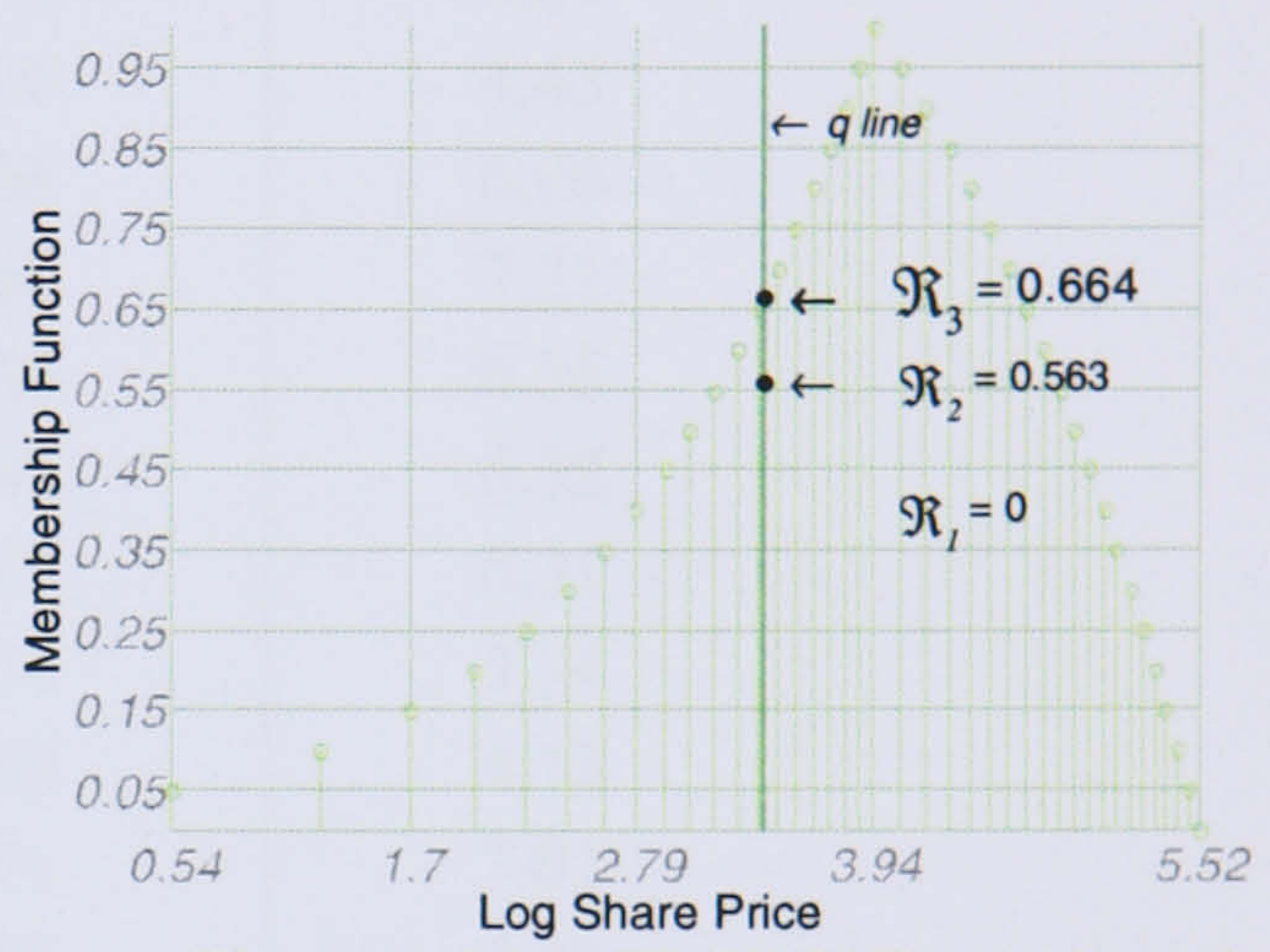


Figure A4.16: TI GROUP - evaluated risk and robustness under time- varying rate: $\Re_3 = 0.664$ and $\Delta = 0.336$

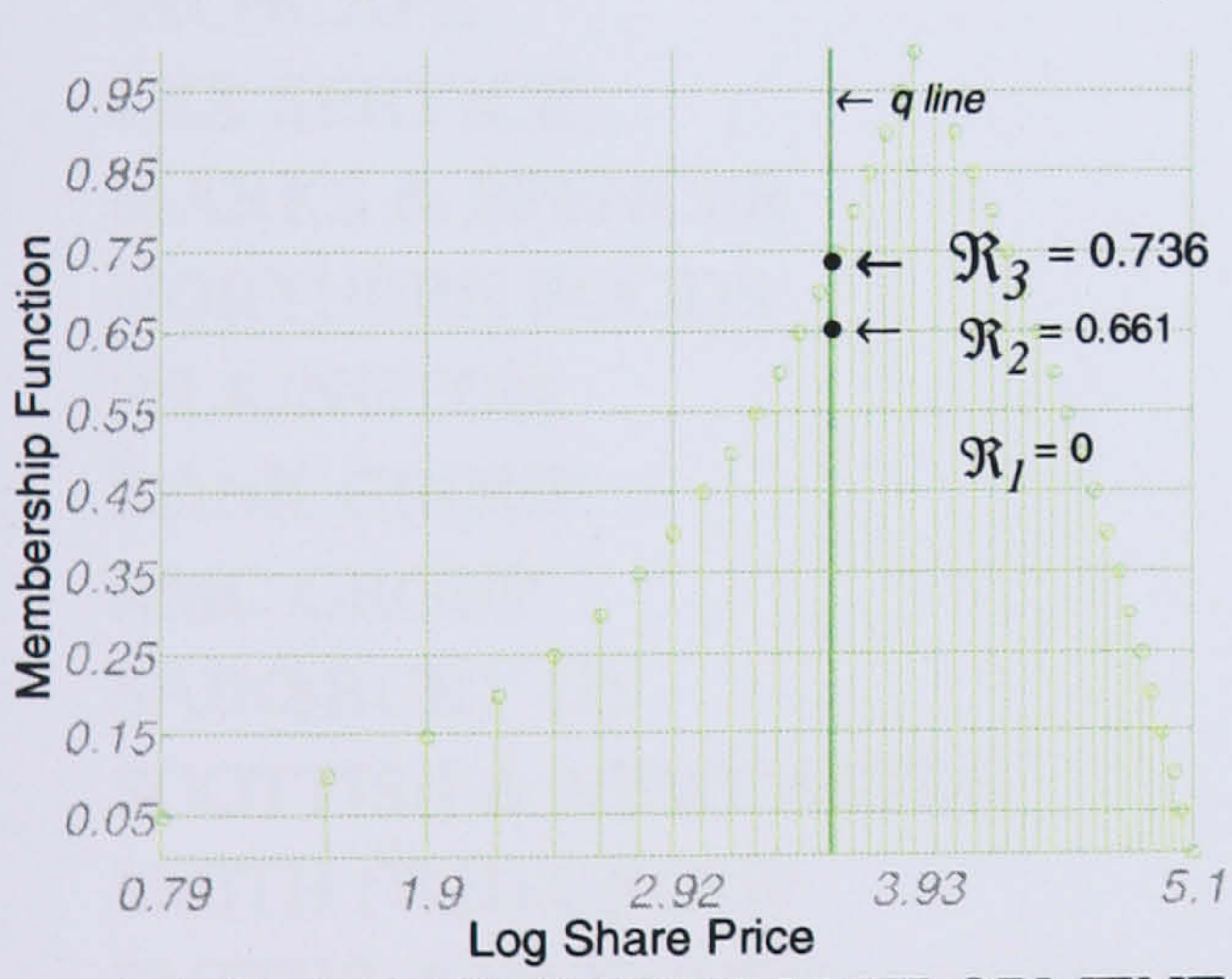


Figure A4.17: TRANSPORT DEVELOPMENT GROUP - evaluated risk and robustness under time-varying rate: $\Re_3 = 0.736$ and $\Delta = 0.264$

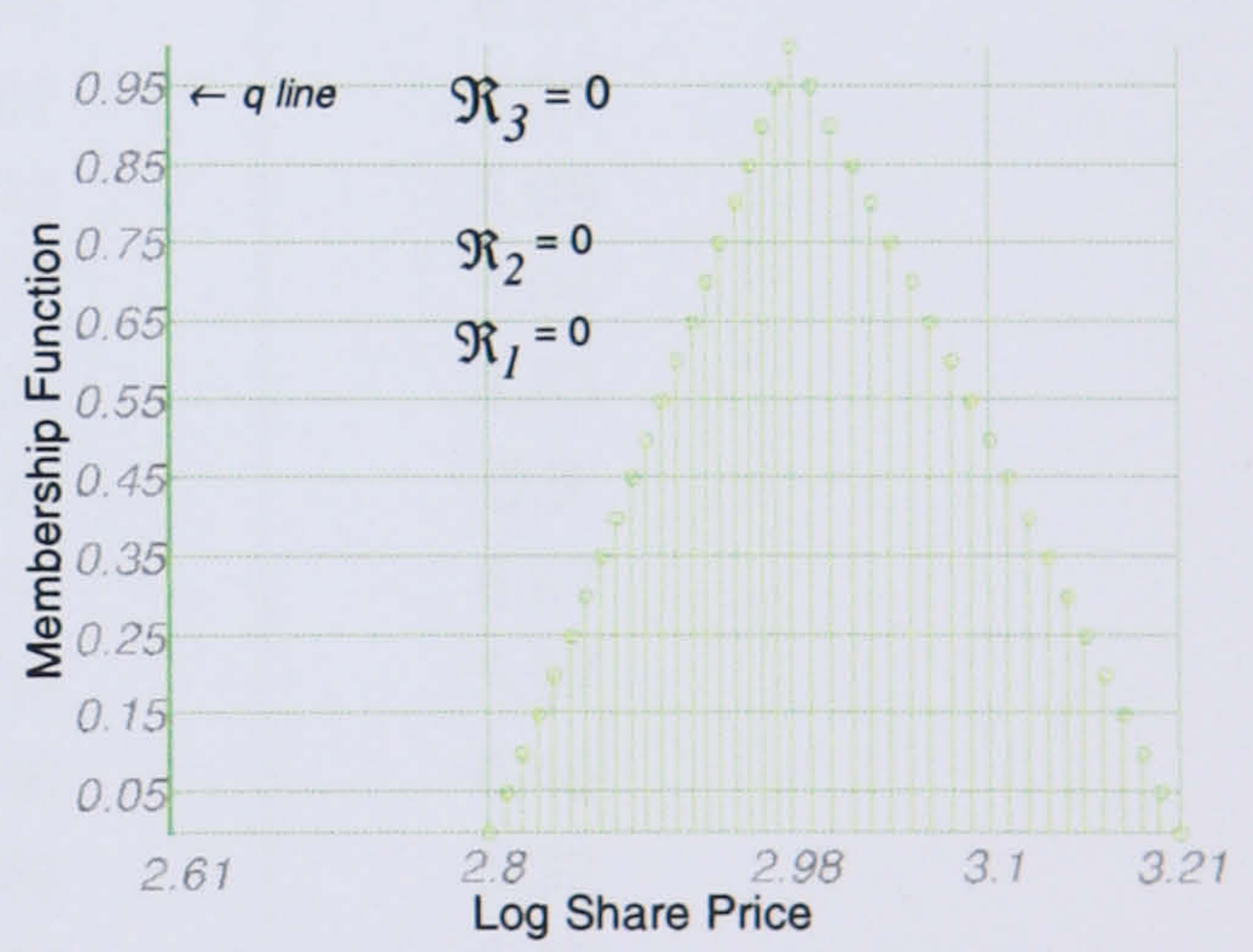


Figure A4.18: UNITED BISCUITS HOLDINGS - evaluated risk and robustness under time-varying rate: $\Re_3 = 0$ and $\Delta = 1$

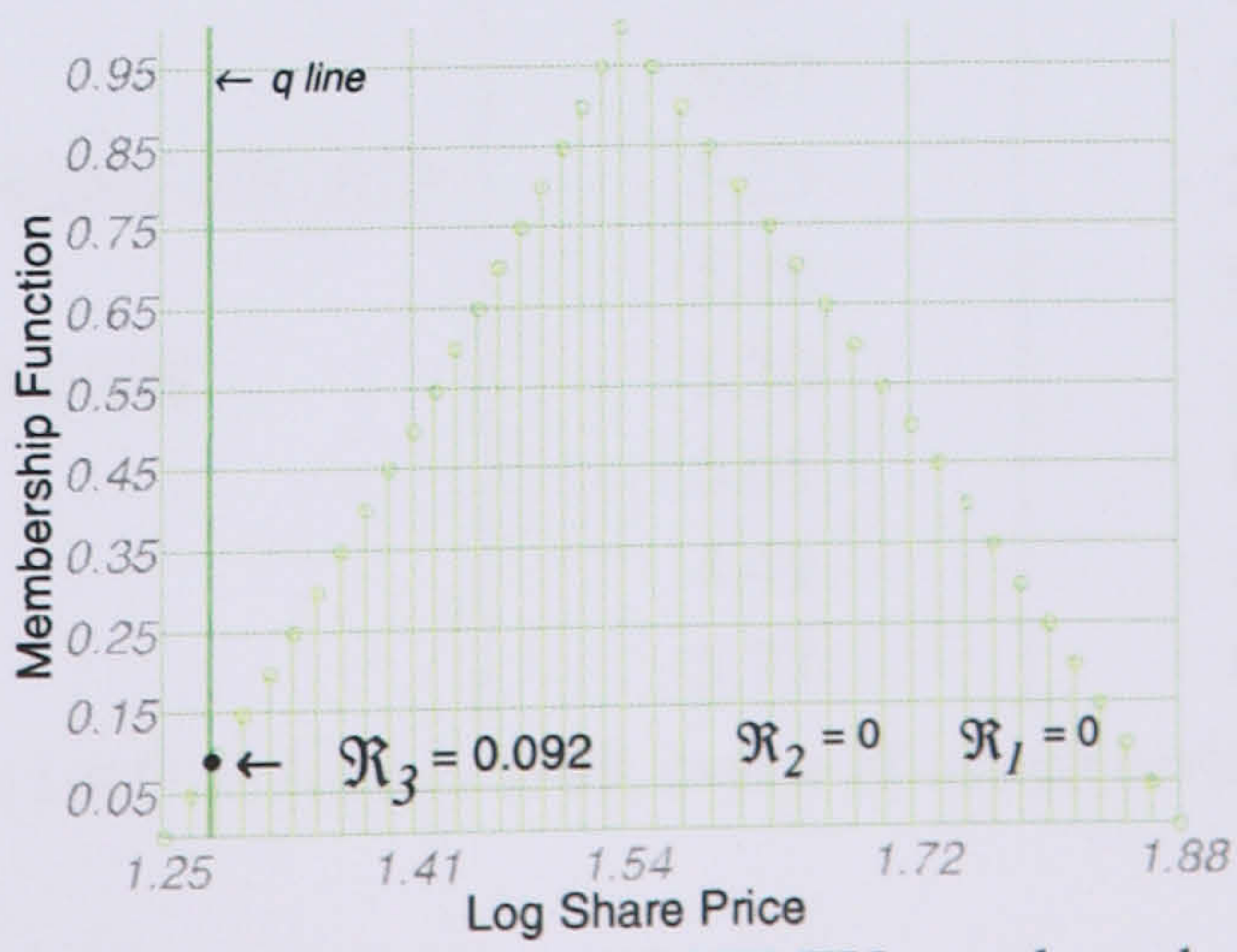


Figure A4.19: WOLSELEY - evaluated risk and robustness under time-varying rate: $\Re_3 = 0.092$ and $\Delta = 0.908$

Appendix A5: Support of the Evaluated Logarithmic Fuzzy Asset Price by Company

Table A5.1: Support of the evaluated logarithmic asset price by company*

company	$\underline{p_0(0)}$	$\overline{p_0(0)}$
BASS	6.38	6.79
BBA GROUP	5.72	6.03
BENTALLS	4.20	4.45
BLUE CIRCLE INDUSTRIES	5.58	6.06
BOC GROUP	6.66	7.21
BOOTS CO.	6.03	6.58
BP AMOCO	5.93	6.32
BRITISH AMERICAN TOBACCO	5.59	6.31
BUNZL	5.34	5.74
COATS VIYELLA	3.68	4.13
DIXONS GROUP	6.71	7.01
GOODWIN	4.13	4.43
GREAT UNIVERSAL STORES	5.4	6.37
HANSON	5.84	6.19
INCHCAPE	5.49	6.57
LEX SERVICE	5.61	6.31
MARKS & SPENCER	5.41	5.99
NORTHERN FOODS	4.57	5.04
PILKINGTON	4.24	4.83
RANK GROUP	5.39	5.9
RMC GROUP	6.36	6.87
SAINSBURY (J)	5.68	5.97
SCOTTISH & NEWCASTLE	5.80	6.46
SMITH (WH) GROUP	5.44	6.32
SMITHS INDUSTRIES	6.47	6.67
TARMAC	5.92	6.44
TATE & LYLE	5.87	6.06
TAYLOR WOODROW	4.57	5.01
TI GROUP	5.87	6.22
TRANSPORT DEVELOPMENT GROUP	5.25	5.58
UNILEVER	5.70	6.03
UNITED BISCUITS HOLDINGS	5.30	5.75
WHITBREAD	6.01	6.73
WIMPEY (GEORGE)	4.49	5.07
WOLSELEY	5.77	6.15

*The fuzzy price is evaluated at $t = 0$ in January 1999, over a 12-month horizon for the pricing factors.