

MULTI-OBJECTIVE CALIBRATION FOR AGENT-BASED MODELS

Alex Rogers
Intelligence, Agents, Multimedia Group
Electronics and Computer Science
Southampton, SO17 1BJ, UK
a.rogers@ecs.soton.ac.uk

Peter von Tessin
Eurobios UK
26 Farringdon Street
London EC4A 4AB, UK
peter.tessin@eurobios.com

KEYWORDS

Agent based model, calibration, pareto-optimality, genetic algorithm.

ABSTRACT

Agent-based modelling is already proving to be an immensely useful tool for scientific and industrial modelling applications. Whilst the building of such models will always be something between an art and a science, once a detailed model has been built, the process of parameter calibration should be performed as precisely as possible. This task is often made difficult by the proliferation of model parameters with non-linear interactions. In addition to this, these models generate a large number of outputs, and their ‘accuracy’ can be measured by many different, often conflicting, criteria. In this paper we demonstrate the use of multi-objective optimisation tools to calibrate just such an agent-based model. We use an agent-based model of a financial market as an exemplar and calibrate the model using a multi-objective genetic algorithm. The technique is automated and requires no explicit weighting of criteria prior to calibration. The final choice of parameter set can be made after calibration with the additional input of the domain expert.

INTRODUCTION

Agent-based modelling and simulation is one of the most useful tools for understanding complex dynamic systems. The properties of such systems, which are mostly the result of the agents actions and interactions, are much better understood when we build models of these systems from the bottom up. By such a procedure we gain insights into the working of the entire system which would not have been possible by concentrating our analysis on only the in and out flow of data, as would be the case with traditional statistical models.

However, this approach often leads to a concentration of the modelling efforts on the inner workings of the various agents and thus a proliferation of model parameters. These parameters often interact in an extremely complex, non-linear manner. Whilst the insights gained through this modelling work might be

very valuable, the usefulness of the resulting model is only assured if it can accurately reproduce phenomenon observed in the real system. Our model building often indicates how some of the model parameters must be set, however we are typically left with a number of parameters which must be *tuned* by comparing the predictions of the model against real world data.

The comparison of the model to real world data is often complicated by a number of factors. A sufficiently complex model produces a number of outputs and thus the ‘accuracy’ of the model can be measured by any number of different criteria. These criteria are often conflicting and thus, increasing the accuracy of one, results in the degrading of another. Additionally, there is often no clear *a priori* way of weighting one criteria against another. Also, the model often incorporate some form of stochastic events. Thus the model must not reproduce exact real world data, but it must be statistically similar to the real world data.

Despite the importance of this calibration task, there are few systematic methodologies and the calibration is typically done by hand (Manson, 2001; Palin, 2002). Where researchers have considered evolutionary algorithms to automatically update model parameters, there is typically only a single ‘accuracy’ criteria or a direct weighting over several criteria (Said et al., 2002).

In this paper, using a simple agent-based model of a financial market, we introduce the use of techniques from multi-objective optimisation to tune the parameters of the model. This technique is automated and does not require that we explicitly trade-off one criteria against another. We use a genetic algorithm to run the model repeatedly with different parameter setting and we evolve a population of pareto-optimal parameters sets. From this population, we can then select a single candidate as our final tuned parameter set. The generation of a population of pareto-optimal parameter sets, allows this final selection to be made in the presence of the domain expert without any prior explicit weighting of criteria.

The remainder of the paper is organised as follows: in the following section we give a short description of the exemplar agent-based model and the various

agents it incorporates. We then describe the use of the multi-objective genetic algorithm for calibrating this model against real world stock price data. We present the results of this calibration and in the conclusion discuss this the use of this technique in more general problems.

FINANCIAL MARKET MODEL

Our exemplar model is an agent-based model of a financial market (Farmer and Joshi, 2000). Unlike more complex models where agents bid on individual assets (Arthur et al., 1997; Darley and Brown, 1999), it incorporates a market maker who drives the price discovery process depending on the net orders placed by a population of trading agents. These trading agents are split in two different categories – trend followers who act solely on their current observation of the asset price and value investors who also consider a fundamental value for the asset. The intention of the Farmer and Joshi model is to show how simple trading strategies inside an agent-based model of a financial market can provide realistic looking asset price time-series. These time-series share some of the characteristics of real world asset prices (e.g. clustered volatility and unit roots). We go one step beyond the statement of such similarities and attempt to calibrate the parameters of the Farmer and Joshi model, in order to derive prices which are as statistically similar to real world prices as possible. We give a sketch of agents here, although we recommend Farmer and Joshi’s original paper for a full account.

Market Maker

The market maker in the Farmer and Joshi model is risk neutral and determines the price of the traded asset by looking only at the supply and demand generated by the population of N traders. A market impact function is used to describe the change in p_t (the logarithm of the asset price P_t) at each time step (Das, 2003). This market impact function is given by

$$p_{t+1} = p_t + \frac{1}{\lambda} \sum_{i=0}^N \omega_t^i + \xi_{t+1} \quad (1)$$

where ω_i is the net demand per agent (i.e. the number of shares of the asset that the agent wishes to buy or sell). The market maker has a liquidity parameter, λ , and in each period a small random term, ξ_{t+1} , drawn from a Gaussian, is used to capture all the other possible influences on the price of the asset.

Trend Follower

The trend following investor assumes that the asset price has inertia and thus always continues to move in its current direction (i.e. it will continue to rise if it rose in the past and it will continue to fall if it

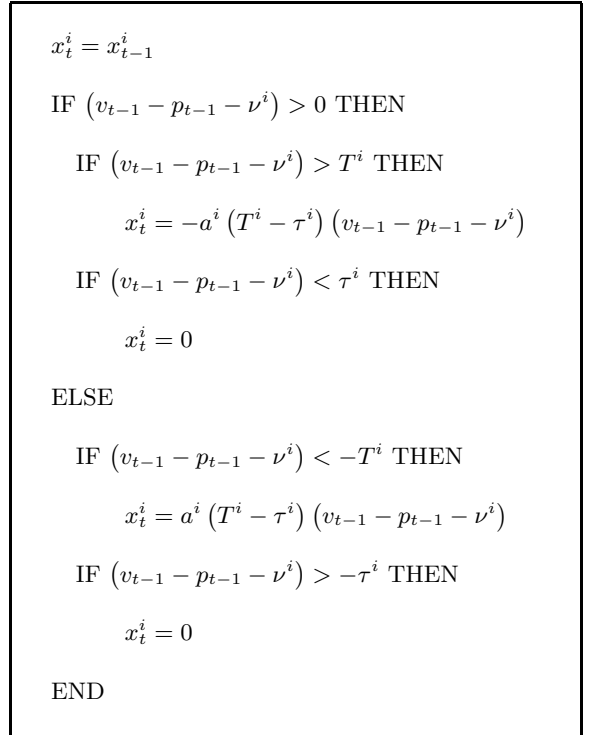


Figure 1: Algorithm for calculating the position of the value investor.

fell in the past). Therefore her position (the number of shares of the asset held) in each period is proportional to the change in asset price determined over a fixed period of time

$$x_t^i = a^i (p_{t-1} - p_{t-\theta^i}) \quad (2)$$

where θ^i represents the time-lag over which the price movement is determined and a^i represents a scale parameter for capital assignment. The demand, ω_t^i , of the agent at this time is simply the change in her position and is thus given by

$$\omega_t^i = x_t^i - x_{t-1}^i. \quad (3)$$

Value Investor

The value investor is a more sophisticated agent than the trend follower. She makes a subjective assessment of the value of the asset in relation to the current price of the asset. Her position is proportional to the difference between this fundamental value and the actual current market price

$$x_t^i \propto (v_{t-1} - p_{t-1} - \nu^i) \quad (4)$$

where v_{t-1} represents the logarithm of the asset value, V_{t-1} , and ν^i represents an offset in the perceived value of the asset which accounts for different agents having different perceptions of the actual asset value.

Number of value agents	N_{value}	100
Number of trend agents	N_{trend}	100
Min threshold for entering position	T_{min}^{value}	0.4
Max threshold for entering position	T_{max}^{value}	1.4
Min threshold for exiting position	τ_{min}^{value}	-1.4
Max threshold for exiting position	τ_{max}^{value}	-0.4
Scaling for capital assignment	a_{value}	0.001
Scaling for capital assignment	a_{trend}	0.001
Min offset for perceived value	ν_{min}	-0.10
Max offset for perceived value	ν_{max}	1.00
Min delay for trend followers	θ_{min}	20
Max delay for trend followers	θ_{max}	80
Price formation process noise	σ_{ξ}	0.01
Liquidity	λ	0.50
Starting price	p_0	0.20

Figure 2: Default model parameters.

Under this strategy, the value investors disregard their actual position. However, this disregard may lead to an unbounded value for this position; a situation which would be contrary to any real world investors behavior. For this reason Farmer and Joshi introduced thresholds into the value strategy, so that a value investor will only enter or exit a position when the difference between the perceived fundamental value and the current price is beyond that threshold; otherwise the position remains unchanged from the last time period. To quote Farmer and Joshi, “Assume that a short position is entered when the mispricing exceeds a threshold T^i and exited when it goes below a threshold τ^i . Similarly, a long position is entered when the mispricing drops below a threshold $-T^i$ and exited when it exceeds $-\tau^i$.”

Figure 1 shows the algorithm for the decision making when these thresholds are incorporated. Again, the demand, ω_t^i , of the agent at this time is simply the change in her position and is thus given by

$$\omega_t^i = x_t^i - x_{t-1}^i. \quad (5)$$

Running the Model

In order to run the model, the agents must be initialised with their relevant parameters. In the case of a value investor, for example, these are a^i, ν^i, T^i and τ^i . Rather than specify parameter values for every individual agent, Farmer and Joshi assign these values by drawing them from a uniform distribution where the model parameters specify the upper and lower bound of this uniform distribution. Figure 2 shows the default parameters which we are ultimately seeking to calibrate.

We do not have complete freedom over the choice of parameter values. As the correlation of log-returns (the change in asset price, $p_t - p_{t-1}$) is close to zero in real asset price time-series, the model should impose this feature. This is done by ensuring that the number of trend following agents matches the num-

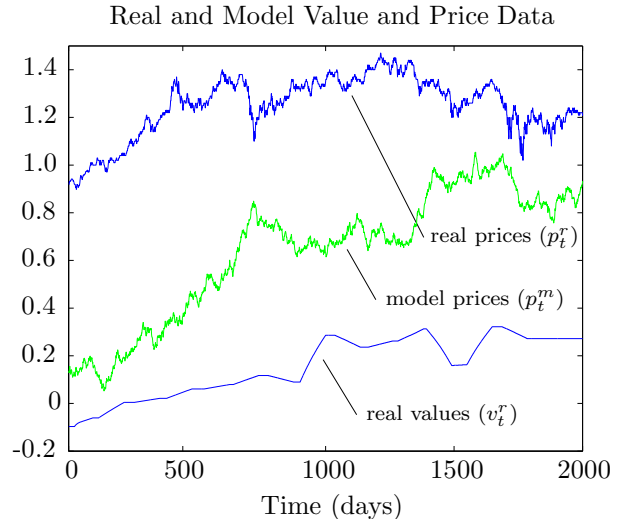


Figure 3: Real price and value data plotted with the model price output when run with default parameters.

ber of value investors and they share the same scaling constant for capital assignment.

In addition to initialising the agent parameters, we must provide a real asset value time-series, which acts as the driver for the value investors. As a proxy for this data, we use the earnings per share (EPS) of the asset, smoothed over a number of days to generate a continuous curve. In the example runs of the model presented here, we use historical data covering eight years of daily data taken from an asset in the Euro-Stoxx-50TM index ¹.

Figure 3 shows the results of running the model with this input data and the default parameters. The ‘real value’ curve is the smoothed real world earning per share data. The ‘real price’ data is the real world share price data and the ‘model price’ curve is the model output.

MULTI-OBJECTIVE CALIBRATION

In order to calibrate the model, we must tune the parameters to ensure that the output matches the real asset price data. As the model itself is stochastic, we can not simply match the time-series, but must match the statistical properties of the time-series. Our choice of measure is driven by a feature of the model, that is, the price and value time-series are cointegrated (Jasiak and Gouriou, 2002). Thus, whilst the price and value data may follow their own random walks, the value $[p_t - v_t]$ is drawn from a well defined distribution. Thus we attempt to match two criteria, the mean and variance of this $[p_t - v_t]$

¹We would like to thank George v. Wallwitz of Bankhaus Hauck und Aufhaeuser for making the data available to us

distribution. These two criteria are thus given by

$$\begin{aligned} \mathbb{E}[p_t - v_t] &= \frac{1}{T} \sum_{t=0}^T (p_t - v_t) \\ \sigma^2[p_t - v_t] &= \frac{1}{T} \sum_{t=0}^T (p_t - v_t)^2 - \left[\frac{1}{T} \sum_{t=0}^T (p_t - v_t) \right]^2. \end{aligned}$$

We compare $\mathbb{E}[p_t^m - v_t^r]$ and $\sigma^2[p_t^m - v_t^r]$ calculated using the model price time-series and the real value time-series with $\mathbb{E}[p_t^r - v_t^r]$ and $\sigma^2[p_t^r - v_t^r]$ calculated using the both the real price and real value time-series. As the model is driven by stochastic factors, we run the model a number of times (in this case, 100 times) in order to collect repeatable statistics.

We calculate the percentage error in the model price value compared to the real price values and these two criteria are used to judge the accuracy of the model output in our example. As it is not clear, *a priori*, how we should weight these two measures, we use multi-objective optimisation to find parameters sets which results in pareto-optimal combinations of two criteria.

There are a number of multi-objective optimisation techniques, but the one we present here uses a multi-objective genetic algorithm to search for a population parameter sets which yield pareto-optimal combinations to the accuracy criteria (Fleming, 1993; Horn et al., 1994). The chromosomes of the genetic algorithm encode the parameters of the model. The chromosomes are evaluated by running the model with the encoded parameters and calculating the accuracy criteria as described above.

The first two operators of the multi-objective genetic algorithm (i.e. mutation and crossover) are unchanged from the conventional genetic algorithm. We apply mutation to each individual by selecting one of the real coded model parameters and applying a small random percentage change. Uniform crossover is performed by dividing the population into pairs and exchanging parameters between each pair with probability 1/2.

The significant difference between the multi-objective genetic algorithm and the conventional genetic algorithm occurs when selection is considered. In the conventional genetic algorithm, selection to the next generation requires a single ‘explicit’ fitness to be evaluated (Davis, 1991). The multi-object genetic algorithm differs from this standard algorithm in two important ways. Firstly, there is no need for this single ‘fitness’ value. Rather the solutions are sorted in order of pareto-dominance, where one solution is dominated by another, if all the accuracy criteria of that solution are better. The next generation is selected on the basis of this ranking. Thus the population member which is not dominated by any other, is most likely to be selected for the next generation.

Whilst the population member which is dominated by all the other solutions, will almost certainly not be selected. Like the conventional genetic algorithm, this selection scheme is stochastic. This changes alleviates the need to explicitly weight different criteria in order to calculate a single ‘fitness’ value and also preserves diversity in the population, as two solutions with the same pareto-dominance ranking may actually encode very different parameter settings.

The second difference, is that all the solutions generated are stored in an archive at the end of each generation and solutions which are dominated by any others are pruned from this archive. Thus the archive represents a store of parameter sets which yield pareto-optimal solutions to the calibration problem and the current population of the genetic algorithm is the source of new solutions.

RESULTS

The multi-objective genetic algorithm was initialised with a population of 20 parameter sets and run for 100 generations. Each evaluation of a parameter set involves running the model 100 times in order to ensure that the values of $\mathbb{E}[p_t - v_t]$ and $\sigma^2[p_t - v_t]$ calculated, are repeatable. In general, there are no stopping criteria for the multi-objective genetic algorithm; whilst the population still shows some diversity, it is still effectively searching for new solutions. The limiting factor is the computational time available.

Figure 4 shows the initial population of parameter sets and the archive of pareto-optimal parameter sets after 20 generation and after 100 generations. The two plots of the archive, show how the pareto-optimal parameters sets form a line in the two dimensional space which gradually moves toward the bottom left corner, indicating that better and better solutions to the calibration problem are being found. The plots show some evidence of there being a trade-off in the accuracy criteria. It is possible to generate parameters sets which very accurately match the value of $\mathbb{E}[p_t - v_t]$ but at some sacrifice to the accuracy of matching the value of $\sigma^2[p_t - v_t]$.

The final contents of the archive is a population of parameter sets that yield non-dominated pareto-optimal solutions to the calibration problem. With no way to weight one criteria against the other, there is no explicit way to choose between them. However in sophisticated models, there are often other characteristics which have not been included as measurable criteria but allow the domain expert to choose between them. In our example, this is not the case and we simply select a parameter set in the middle of the range where both criteria are well satisfied. Figure 5 shows these parameter values.

We show the results of a single run of the model using these parameters in figure 6. Clearly the model prices show much better agreement than in the pre-

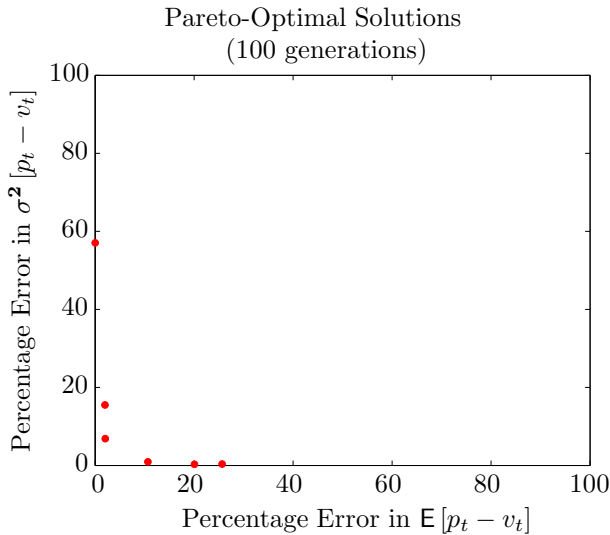
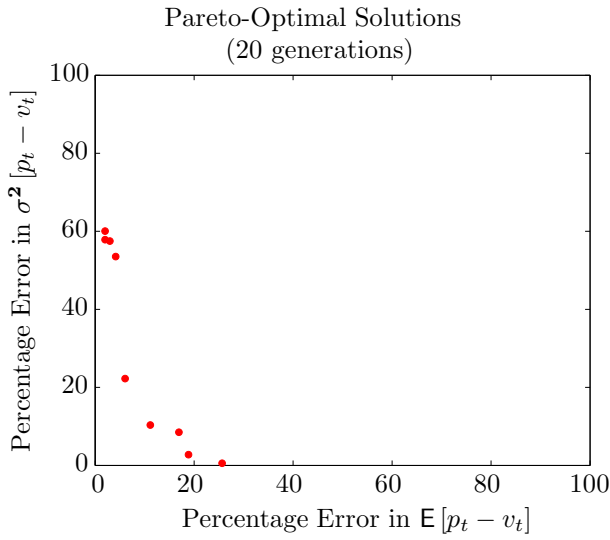
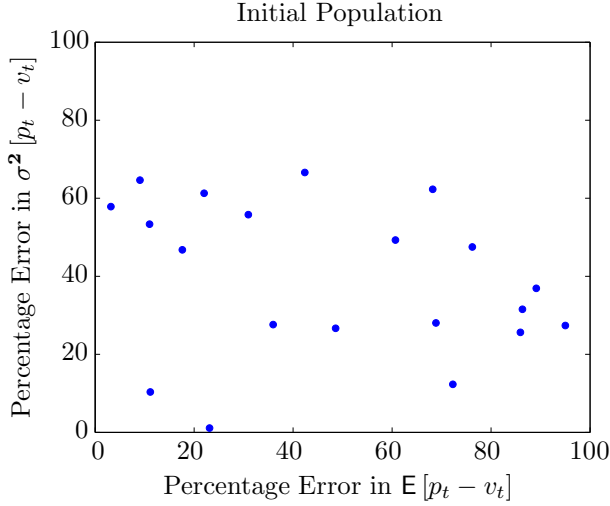


Figure 4: Sequence showing the development of pareto-optimal parameter sets during subsequent generations of the multi-objective genetic algorithm.

Number of value agents	N_{value}	120
Number of trend agents	N_{trend}	120
Min threshold for entering position	T_{min}^{value}	0.18
Max threshold for entering position	T_{max}^{value}	2.73
Min threshold for exiting position	τ_{min}^{value}	-0.32
Max threshold for exiting position	τ_{max}^{value}	-0.13
Scaling for capital assignment	a_{value}	0.0001
Scaling for capital assignment	a_{trend}	0.0001
Min offset for perceived value	v_{min}	-1.95
Max offset for perceived value	v_{max}	0.75
Min delay for trend followers	θ_{min}	1
Max delay for trend followers	θ_{max}	120
Price formation process noise	σ_{ξ}	0.01
Liquidity	λ	0.41
Starting price	p_0	1.08

Figure 5: Final calibrated model parameters.

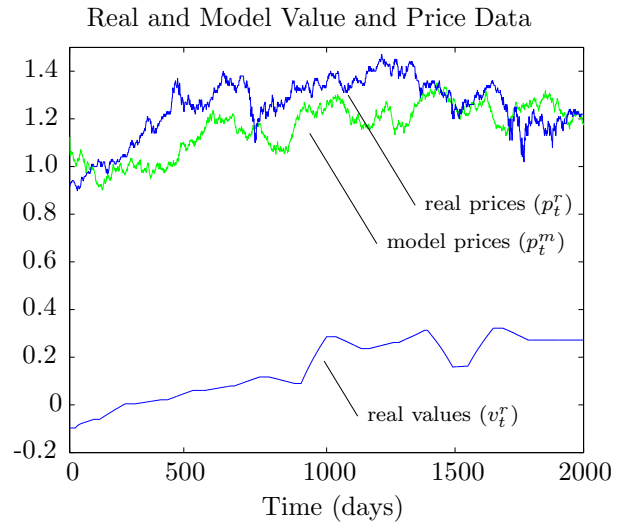


Figure 6: Real and model price data plotted against time. Value data also plotted.

vious run presented, where the default parameters were used. Whilst we have not attempted to match the actual time-series, the statistical properties of both the model and real price time series show good agreement.

CONCLUSIONS

In this paper we have addressed a common problem in the use of agent-based models; how to calibrate the final model to ensure that it accurately reflects the behaviour of the real system. As these models typically involve large numbers of parameters and their accuracy is measured with a number of different criteria, an automated calibration process is of great value. Here we have demonstrated the use of a multi-objective genetic algorithm to evolve a population of parameter sets which yield Pareto-optimal solutions to the calibration problem. The final choice of the single parameter set to be used, can be made without

the need to explicitly trade-off one criteria against another and this final decision can often be made with the assistance of the domain expert.

We demonstrated the process with a simple agent-based model of a financial market. This model shares many of the properties of larger more complex agent-based models used in real applications. It involves a large number of parameters that interact in a non-linear manner. As the model incorporates stochastic effects, we can not simply compare the model outputs to the real world equivalent, but must match the statistical properties of both. This typically involves running the model repeatedly to ensure that these statistical measures are repeatable. It is the computational time required to perform these runs which tends to be the limiting factor in the process.

In the example agent-based model presented, we derived two criteria for comparing the accuracy of the model output to the real world asset prices. The resulting population of pareto-optimal thus represents a line in the two dimension plot. Where there are more criteria, the population represents a surface in some high dimensional space. Visualising this result becomes more complex and we are currently investigating visualisation tools to aid in this process.

REFERENCES

- Arthur, B., Holland, J., LeBaron, B., Palmer, R., and Tayler, P. (1997). *Asset pricing under endogenous expectations in an artificial stock market*. Addison-Wesley.
- Darley, V. and Brown, M. (1999). The future of trading: Biology-based market modeling at Nasdaq. In *Perspectives in Business Innovation. Issue 4*. Ernst and Young Center for Business Innovation.
- Das, S. (2003). *Intelligent Market-Making in Artificial Financial Markets*. PhD thesis, Massachusetts Institute of Technology.
- Davis, L., editor (1991). *Handbook of Genetic Algorithms*. Van Nstrand Reinhold, New York.
- Farmer, J. and Joshi, S. (2000). The price dynamics of common trading strategies. Technical report, SFI Working Paper 00-12-069, Santa Fe Institute.
- Fleming, C. M. F. . P. J. (1993). Genetic Algorithms for Multiobjective Optimisation: Formulation, Discussion and Generalisation. In *Proceedings of 5th International Conference on Genetic Algorithms*, pages 416–423. Morgan Kaufmann.
- Horn, J., Nafpliotis, N., and Goldberg, D. E. (1994). A niched Pareto Genetic Algorithm for Multiobjective Optimisation. In *Proceedings of IEEE Conference on Evolutionary Computation*, volume 1, pages 82–87. IEEE.
- Jasiak, J. and Gouieroux, C. (2002). *Financial Econometrics*. Princeton University Press.
- Manson, S. (2001). Calibration, Verification, and Validation. In *Meeting the Challenge of Complexity. Proceedings of a Special Workshop on Land-Use/Land-Cover Change*. Center for the Study of Institutions, Population, and Environmental Change.
- Palin, J. (2002). *Agent-Based Stockmarket Models: Calibration Issues and Application*. PhD thesis, University of Sussex, UK.
- Said, L. B., Bouron, T., and Drogoul, A. (2002). Agent-based interaction analysis of consumer behavior. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 184–190. ACM Press.

AUTHOR BIOGRAPHIES

ALEX ROGERS is a Senior Research Fellow in the Intelligence, Agents, Multimedia Group at the University of Southampton. He gained a Ph.D. for work modelling the dynamics of genetic algorithms and has worked with Eurobios UK on a number of business consulting and agent-based modelling projects. His email address is a.rogers@ecs.soton.ac.uk.

PETER VON TESSIN is Senior Software Engineer at Eurobios UK. He gained a DEA in Mathematical Economics and Econometrics from Pantheon-Sorbonne University in Paris and a Ph.D. in Econometrics from Tuebingen University in Germany. Since graduation the main focus of his work has been in business consulting, quantitative and agent-based modelling and object-oriented software design. His email address is peter.tessin@eurobios.com.