Proceedings, 2003 Congress on Evolutionary Computation, Canberra, Australia, 8-12 December 2003, pp. 2322-2329

The Simple Supply Chain Model and Evolutionary Computation
Congress on Evolutionary Computation

Timothy Gosling
University of Essex
Wivenhoe Park, Essex
C04 3SQ, England
tdbgos@essex.ac.uk

Abstract- This paper provides an overview of the Simple Supply Chain Model (SSCM), scenarios derived from this model and the strategies being used to begin to tackle SSCM problems. The paper further provides details of how evolutionary computation is used (via Population Based Incremental Learning) to optimise parameters for the designed strategies.

1 Introduction

Electronic market places will, in future, allow full electronic supply chains to be developed and exploited [1][2][3]. While a human operator in this environment could provide good situational awareness and excellent negotiation capabilities the sheer amount of information available and number of transactions possible would be overwhelming. To solve this problem it is proposed that a computer based strategy is necessary to handle the situation, leaving the human operator to provide high level decision making. The computer based strategies success is not simply in terms of providing negotiating well with customers and suppliers but also in it's ability to handle a far larger number of transactions simultaneously. To develop a successful strategy it is first necessary to define the problem more precisely. To aid in this definition consider a simple supply chain that consists of customers, middlemen and suppliers. The strategy used (by the middleman) to tackle this supply chain must attempt to fulfill customer requirements and make a profit by negotiating with suppliers for the required goods or services. This must be accomplished within constraints placed on the starting knowledge, negotiation time and the communication budget and framework of each participant. Having developed a model of supply chains the next stage (by step wise increments in complexity) is to develop successful strategies to tackle the problem. The Simple Supply Chain Model (SSCM) described shortly and its related scenarios follow this approach to provide a flexible framework for describing and investigating solutions to realistic problems. To support the investigation of a strategy's effectiveness in different environments it is possible to use a simulation system. The same simulation system may also provide an opportunity to optimise a strategy's parameters for a specified environment or environments. Again following this approach a SSCM simulator and strategy optimiser has been created and various experiments carried out within its framework. The work to date has resulted in a strategy to tackle a particular sub-problem (or scenario) of the SSCM and promises to tackle further, increasingly complex, problems in the near future. The strategy developed has further been optimised via the developed experimental system using a population based incremental learning mechanism [8].

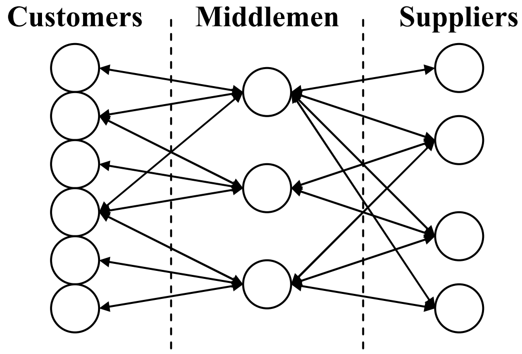

Figure 1: A simple supply chain

2 The SSCM

The SSCM provides a mechanism for describing supply chains containing three types of participants; customers, middlemen and suppliers. The model describes the starting information of each of these participants along with the goods/services available, timing information and the communication system each participant may use.

$Model = (S, M, C, P, T_{total}, T_{active}, Com)$
S – The suppliers (of products)
M – The middlemen (travel agents)
C – The customers (travellers)
P – Product types
T_{total} – The total time (in days) over which this simulation system runs
T_{active} – The length of activity time (in days) that occurs at the end of total time
Com – A communications protocol

Figure 2: Top-level SSCM definition

The SSCM provides a general, flexible framework for describing realistic business situations yet is precise enough to allow for rigorous study. Furthermore the SSCM is open to easy expansion and adaptation. In contrast to other work, which tends to focus on more specific or constrained negotiation or auctions situations ([4][5][7]), the SSCM approach takes a higher-level view and involves the participants in a more complex and dynamic environment.

To aid in the development of the model a specific business scenario was chosen. In this case a travel agency scenario was used, much like the TAC game [6], this scenario deals with middlemen providing travel packages to customers, obtaining the necessary services from a set of suppliers. There is only one location for travel in this case and limited accommodation options but various options for 'entertainment' while away. There is no restriction on the numbers of customer, middleman or suppliers. Each participant may use any strategy provided that the communication scheme is adhered too and that only deals the participant can fulfil are made.

The customer starting information includes information about travel preferences and budget, including earliest and start time, preferred duration and the sort of entertainment required. Further information includes the known middlemen and, as with all participants, the communication budget (maximum number of outbound messages per time slice).

Supplier information is limited to what product the supplier sells, the availability of products per day and the base cost of an item to the supplier. The information also includes the known middlemen and, again, the communication budget.

A middleman's starting information is considerably less than either the customers or suppliers. The middleman information includes only the known customers and suppliers along with the communication budget. In this sense the middleman is highly unconstrained.

A complete supply chain defined using the SSCM can be view as a constraint satisfaction or optimisation problem where the objective is to fulfil as many customer requirements as possible using the least funds from customers and both middlemen and suppliers leveraging at least a break even price for the goods they buy and sell.

In reality a problem defined via the SSCM would be distributed across the specified participants it is thus possible to view the SSCM as a distributed constraint satisfaction/optimisation problem. In this case the objectives are the same but no central solution could be found. If the further complication of purely self-interested, non-cooperative (in the sense that they won't do what you want simply because you want them to) participants is brought in the problem is made considerably more difficult. Tradition distributed constraint satisfaction techniques won't suffice.

Having defined the SSCM it is necessary to develop strategies that, using the specified communication system, are able to achieve reasonable global results via negotiation without the need for philanthropy or cooperation by individual participants. Developing these strategies for the SSCM from scratch would prove difficult, as the range of problems that may be represented is quite large. To this end certain SSCM Scenarios are used to restrict the SSCM's breadth.

3 SSCM Scenarios

SSCM Scenarios are designed to make the task of designing a strategy to tackle the SSCM more tractable. Since the range of situations the SSCM encompasses is large, immediately developing a strategy to tackle it in total would not be prudent. Instead we take the course of developing useful components and sub-components of a final strategy through the use of SSCM Scenarios. These Scenarios place additional constraints on the problems the SSCM may represent and the way in which participants may behave. By reducing the range of problems encapsulated by the SSCM and the situations participants may find themselves in it is possible to develop effective independent strategies that may be used later as components of a complete SSCM strategy. Having used an initial SSCM Scenario to develop a simple SSCM strategy, further Scenarios can be designed to increase the complexity of the task faced incrementally. Using the strategy developed from the previous Scenario as a foundation it is possible to develop a new more capable strategy. The ultimate aim is that a final Scenario would place no constraints on the SSCM and the final strategy would be capable of handling all SSCM representable problems.

The Scenario chosen to begin development of a strategy, Scenario One, is quite simple.

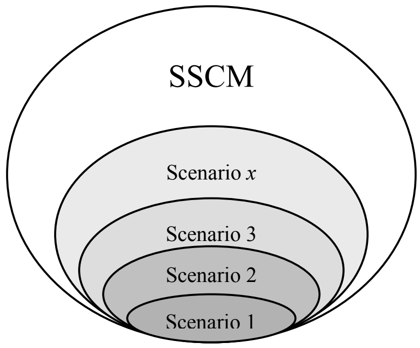

Figure 3: SSCM Scenarios increase in complexity

Scenario One restricts customers to only knowing one middleman each. Furthermore middlemen don't (initially) know any customers, this places the onus of making the first move on customers. A further restriction on customers is that they will not negotiate, having supplied an initial requirement the customer will be expecting a middleman to either accept or reject the trade, and any attempt at negotiation will be met by reaffirming the original requirement. This turns the customer-middleman side of the system into a satisfaction problem. Middlemen must simply decide if it is possible for them to fulfil a customer's requirements given what they know now, if not they should probably immediately reject the customer's requirements, if so they should probably start negotiations with the suppliers in an attempt to fulfil them. Of course, the middleman need not do any of these things but to be successful they probably should.

The restrictions on suppliers are in some respects similar. There is only one supplier per product. This prevents the middlemen from needing to decide with which supplier to deal. Suppliers furthermore have no knowledge of middlemen to begin with and as such should not initiate any negotiation. Suppliers are expected to attempt to fulfil a middleman's requirements as fully as possible but are free to negotiate on the issue of price including the ability to reject the requirement outright. The scenario details no specific mechanism for the suppliers negotiation system although it is assumed the mechanism would be a relatively simple one.

The restrictions on the middlemen are, as with the model, somewhat limited. A middleman is simply expected to try and make a profit from customer requirements. The Scenario doesn't say a middleman must do this, it doesn't for instance disallow a middleman buying everything it can without any customers to sell to but, the ways in which the middleman can behave are somewhat restricted by the customer and supplier constraints.

Further scenarios leading on from Scenario One have been developed but will not be discussed here.

Having developed a scenario the next step is to design a strategy for each participant.

4 Scenario One strategies

Scenario one places considerable constraints on the actions of customers and suppliers.

The operation of a customer is almost entirely described by Scenario One however exactly what requirements are initial sent to a middleman and when they should be sent are not. The suggested approach is that customers request the earliest possible travel arrangement that fit its parameters and send this requirement as soon as possible in order to give the middleman as much time as possible to negotiate with suppliers. This is not to say that customers are entirely cooperative with the middleman, some customers may have requirements that need to be dealt with much sooner than others providing the middleman with a more varied environment in which to work.

The supplier strategy is likewise quite simple. A supplier will immediately reject any requirement from the middleman it cannot fulfil in any way. If the requirement maybe completely fulfilled and the price offered is sufficiently high the supplier may immediately accept. If the requirement may not be fulfilled in total or the price suggested is too low a counter offer will be generated. The counter offer will fulfil the requirement as far as possible and include the lowest price the supplier is willing to accept at this time. The acceptance price is based on a per unit value generated by some algorithm. The one suggested appears in [7] (Page 2, Time-dependant tactics) and is controlled by a single parameter. Values for this control parameter below one make the supplier only concede value to the middleman as the end of the negotiation approaches, value above one make the supplier give ground early.

The middleman strategy is necessarily more complex. The basic mechanism is as follows.

The middleman maintains a variable number of requirement groups. Each requirement group is responsible for handling the negotiations for a number of different customer requirements. The requirements within the groups are placed there according to a similarity measure. Customer requirements are initially assessed for feasibility based upon the perceived value of the goods required and the amount the customer is willing to pay. Feasible requirements are added to a group based upon the similarity measure, unfeasible requirements will be immediately rejected.

The requirement groups themselves may be in one of three states; inactive, active or complete phase. Inactive groups are continually updated based upon the latest information available to the middleman and their membership is flexible with new requirements being added and older requirements perhaps being dropped if they are no longer considered feasible. Once an inactive group reaches its activation time (essentially the minimum amount of time the middleman requires to negotiate for goods) the state is changed to active.

When a group becomes active the middleman will begin negotiations with suppliers for the goods necessary to fulfil the customer requirements contained within that group[1]. As a result the membership of the group is locked

[1] The middlemen negotiation strategy is based on the full set of tactics presented in [7].

and no new requirements will be added. Requirements may be removed from the group if it becomes apparent that the middleman will be unable to acquire the goods necessary to fulfil them. Profitability estimates are maintained as with an inactive group but unprofitable requirements will not be dropped unless this can be done safely without compromising negotiations. Eventually time will run out for negotiations and the group will enter the final complete phase state.

The completion phase for a group sees the middleman providing a response to customers it felt it could deal with. If the necessary goods were obtained then a series of customer requirement accept messages will be sent, if not a series of reject messages will be sent instead. Ideally each group should complete successfully and make a profit in doing so.

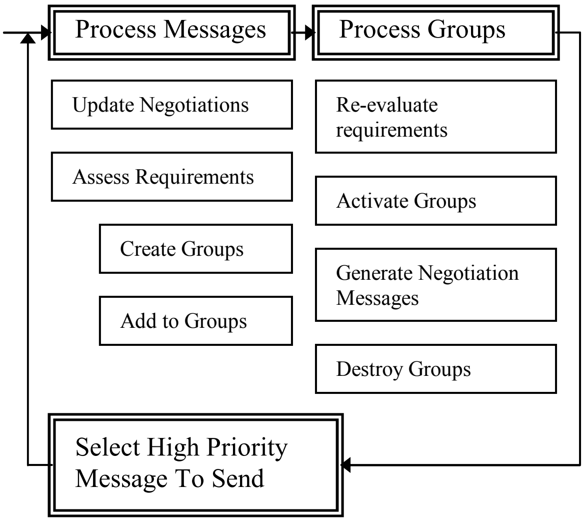

Figure 4 - The basic Middleman process

The strategy runs all the groups, assesses requirements, assigns requirements to groups (creating new groups if necessary) and handles the various communication traffic in a semblance of concurrently. To achieve this the strategy works in cycles, each cycle all the inbound messages are processed and negotiations updated. In the case of customer requirements each requirement is assessed and if worthy is assigned to an inactive group, unworthy requirements are added to a special failure group. Following message processing each group is processed in order of creation. Active and complete phase groups will generate messages, which are added to a pool; entirely complete groups will be removed as necessary. Having processed each group the pool of messages is assessed and the highest priority message (according to a measure) is selected and sent. The cycle can then return to the beginning.

5 The Experimental System

Having developed the strategies necessary to tackle SSCM Scenario One it is further necessary to develop a system within which the strategies can be tested and optimised. To this end an experimental system has been developed that allows the participants to operate within an SSCM market situation. This experimental system has further been designed to facilitate the testing and optimisation of new strategies and scenarios in the future.

The experimental system makes some small but important changes to the way the SSCM is viewed. The SSCM provides a description of a supply chain that is entirely static from the beginning with fixed information and fixed time in which negotiations must be accomplished. The experimental system allows these situations but is also capable of going further by essentially introducing new customers at intervals. This addition to the system has been made to aid in the optimisation of middleman strategy parameters through the use of a Population Based Incremental Learning [8] (PBIL) evolutionary algorithm. PBIL and the specifics of the systems PBIL component are described in the section below.

The experimental system is essentially split into three component parts[2], the communication center, the participants and the controller.

The communication center simply provides a mechanism for message redirection between participants and the controller but is important as it provides felxibility, allowing the distribution of the system over many experimental machines if necessary or desirable.

The participants (customers, middlemen and suppliers) are based around a series of core components to ensure that basic behaviors are the same. The individual customer, middleman and supplier components of this software implement their related strategy and can be configured separately. While the middleman and supplier components operate as per the described strategies the customers go a step further. Each customer is configured with a requirement template; this template is able to generate any number of customer requirements allowing each individual software customer to take on the role of many customers with respect to the SSCM. The customer is able to treat each individual requirement separately so that each customer may be able to deal with multiple middlemen but still retain the restrictions applied by SSCM Scenario One. This restriction can be easily relaxed for future scenarios. The flexibility of the customers to generate multiple requirements is also exploited by the controller to provide additional customer

[2] A further collection of analysis and experimental tools has also been developed but is not discussed here.

requirements over time when performing strategy optimisation.

The controller software provides a universal clock to the system's participants and is further able to request additional requirement generation, middleman performance evaluations and remotely reconfigure middlemen. For optimisation purposes the controller software maintains the PBIL information.

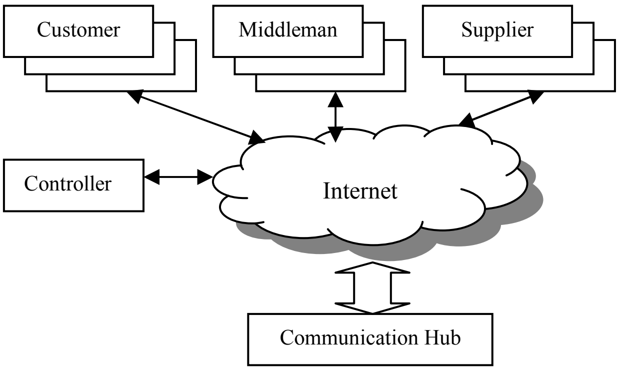

Figure 5: Experimental system operation

To run a standard SSCM experiment participants are configured and connected to the communication center. The controller software also connects to the communication center and is then able to issue the universal clock to participants. To start the system running the control software is then able to send requirement generation requests to the customers.

From this point on the participants are free to operate as they see fit and the controller is able to evaluate middleman performance at the end. Information about the controller and each participant's operations and communication are recorded for future reference.

Most experiments with the system have however focused on the optimisation of middleman strategy parameters. This mode of operation is discussed in more detail below.

6 Use of evolutionary computation (PBIL)

The evolutionary component of the experimental system provides a way of developing good parameter sets for a middleman strategy.

The development of good parameter sets is critical to the success of a strategy in a given environment as the potential problem space is large and diverse even given the limited complexity of Scenario One. Attempting to optmise the parameter set used by the Scenario One strategy allows the strategy to operate more effectively, as well as providing information on how it adapts and how it may be adapted to a given environment.

Optimising the middleman strategy is problematic as the resources required for testing a large number of different parameter sets in such a large environment are prohibitive. There is consequently a trade off between the number of samples that can be taken and the resources being used.

Population based incremental learning has been chosen as the optimisation mechanism as it has proven it self effective in a range of problems ([8][9][10][11]) and has the additional advantage of not requiring a large sample population. PBIL is as a result well suited to the problem of optimising the middleman strategy parameters and is subsequently explained below in this context.

Population based incremental learning is grounded in both the evolutionary and reinforcement learning domains. PBIL attempts to replace a GA's population of solutions with a single probability vector and update this vector using mechanisms similar to those used for neural networks. The probability vector includes the probability of each value for each symbol in the problems chromosome.

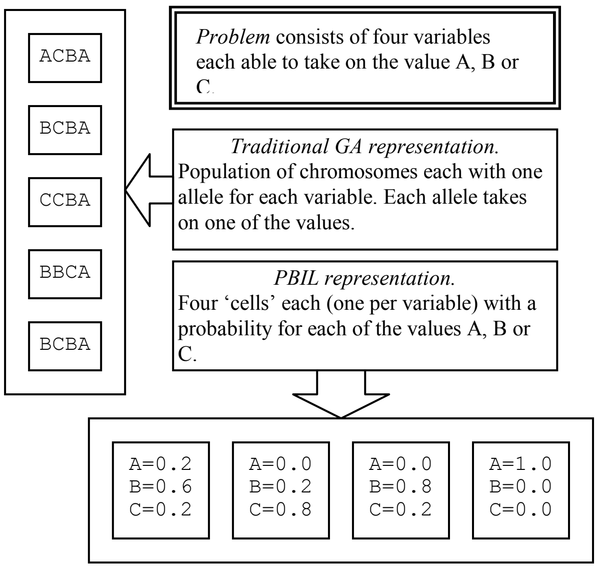

Figure 6: GA versus PBIL representation

Selecting the values of each symbol probabilistically generates a solution string. So for instance in the above example if the most probable value was selected in each case the resulting string would be {BCBA}.

Updating the PBIL is relatively straightforward. Positive reinforcement may be used to improve the chances of a given string occurring and negative reinforcement may be used to reduce the chances of a given string occurring. In both cases a learning rate is used (learning rates maybe between 0.0 and 1.0).

To perform positive reinforcement a good string (one you would want to repeat) must be found. This may be done by generating solutions from the probability vector and testing them against some criteria. Once a good string is found the probability of each value in the string is increased by the learning rate. The remaining values are multiplied by one minus the learning rate.

To perform negative reinforcement a similar process must be used to find a bad string (one you would not want to repeat). In this case the probability of each value in the string is reduced by the learning rate. A learning rate amount of probability is then redistributed amongst the remaining values for a symbol according to their existing probabilities.

This mechanism is used within the experimental system for the generation and update of strategy parameters that have obvious, discrete values. The criteria for whether a string is good or bad is based upon the middlemen evaluations that are received and both positive and negative reinforcement is used. This process will be discussed in more detail shortly.

While the above mechanism works well for variables with discrete values, variables that maintain continuous values cannot be optimised in this way. To address this issue and allow PBIL, and thus the experimental system, to optimise continuous ranges the following mechanism is applied.

The required value range is broken up into a number of discrete blocks; one block covers the wrap around from lowerest to highest value. A probability is assigned to each block in the same way that a probability is assigned to each discrete value for a discrete value variable. However the mechanism for generating values and updating the continuous blocks is different.

When generating a symbols value, one of the symbols value blocks is first selected according to its probability. Having done this a specific value is then chosen within that block with even probability.

When performing positive reinforcement the block to which the reinforcing value belongs has its probability increased in exactly the same manner as for a discrete variable however in addition, the boundaries of that block contract, focusing towards the given value causing the update. The boundary contraction reduces the distance between the focusing value and the boundaries of the block on either side. The reduction is performed by the same learning rate used for changing the probability (i.e. for a learning rate of 0.1, the boundaries either side of a value are reduced by 10%).

To perform negative reinforcement the block to which the reinforcing value belongs probability is reduced just as before but no change to the blocks boundaries are made.

This continuous value mechanism is again used by the experimental system this time for parameters that cannot be sensible described as discrete.

The experimental system uses PBIL to optimise the middleman strategy by providing probability information for each variable in the strategy whether that variable is discrete or continuous. For a typical optimisation experiment this would mean the use of seventy-seven variables.

To perform the optimisation, strategy parameter sets are created from the PBIL information probabilistically and used to remotely configure the middleman within the experimental system. The system is then run for a number of turns (typically fifty) with new customer requirements being requested periodically (typically every ten turns) to test the middlemen configurations. At the end of the test period the controller forces each middleman to perform an evaluation and return the results. If the evaluation of a given middleman is good its parameters will be used to positively reinforce the PBIL information (a learning rate of 0.1 has generally been used). If the evaluation returned by a middleman is bad, its parameters will be used to perform negative reinforcement on the PBIL information. Further more a new parameter set will be generated and sent back to reconfigure the middleman. The system can then resume as before testing the current set of middleman configurations. Periodically (usually every three test periods) the criteria by which middlemen are considered good or bad will be updated to help track improving middleman performance.

The evaluation mechanism being used by the system is thus. The controller requests and each supplier returns market information about its products. This information is in the form of the mean per unit sale price of a product within a specified window of transactions (typically five). This market information is then sent to the middlemen in order for them to calculate their current worth. The current worth of a middleman is defined as its current funds plus the total value of usable products it holds (hence the market information) minus the total of any past evaluations. This mechanism provides a middleman's performance over the current evaluation period and prevents previous good performance from providing cover for recent bad performance. This final value is sent to the controller for evaluation against a current set of criteria for success.

The criteria used to rate the success or failure of a middleman is based upon the mean evaluations of middlemen over a set number of test phases - these criteria are updated periodically. To stop inactive middlemen from become predominant (an initial problem) the experimental system can be set up to regard any zero valued middleman as bad.

The specific criteria come in three parts. The first is the actual mean value of previous middleman evaluations and is essentially used to provide a touchstone for future evaluations. A middleman achieving an evaluation within a set percentage of this mean (usually 10%) will neither cause positive or negative reinforcement to occur.

The second and third criteria relate to middleman evaluations that fall outside the above 'safety zone'. One of these is the upper-limit. The upper-limit is a specified proportion above the mean. Middlemen evaluations at or above this limit will cause positive reinforcement at the maximum learning rate being used. Middlemen between the mean and this limit cause positive updates with a learning rate linearly scaled from zero to the maximum learning rate used respectively. The final criterion is the lower-limit. This works in exactly the same way but for negative reinforcement and evaluation below the mean.

By using PBIL within the experimental system it has been possible to run experiments that optimise the middleman strategy parameters. This is discussed in the next section.

7 Experimentation and Results

Experiments run using the scenario one strategies within the experimental system were directed via four goals.

The first goal was to demonstrate that the experimental system (and its PBIL component) was capable of optimising the middleman strategy.

The second goal was to show that no universal solution existed to the problem of parameter set optimisation – that is to say that different environmental conditions would be more effectively tackled by different strategy parameter sets.

The third goal was to demonstrate that while the system is capable of adaptation, there are limits to the ability of the system to adapt.

The final goal was to show that the limits to adaptation could be overcome to some degree by making use of an initial PBIL configuration in line with successful parameter sets that were formed near to the adaptation boundaries.

This first of these goals was easy to achieve, if the experimental system did not perform any form of optimisation it would become obvious rapidly. Fortunately the system did optimise the middleman strategy parameters and further experimentation could be undertaken.

The second goal was tackled in the following way. Part of the supplier configuration is the harshness with which it negotiates with middlemen. Keeping all other factors the same this negotiation harshness was adjusted to provide the middlemen with a more or less difficult environment in which to operate. This was done with a view to forcing the evolution of differently optimised parameter sets. This proved to be the case and lent initial support to the notion that different environments require different parameter sets. To further reinforce this point PBIL initialisation data was formed from parameter sets created within a given environment and placed within another. In this set of experiments the expected result was that of seeing the PBIL configuration adjust towards values associated with parameter sets previously developed within that environment. This again appeared to be the case and further reinforced this goal.

The third goal was relatively simple to show to be true. During initial testing it became apparent that some customer or supplier configurations were too difficult for the system to be able to effectively adapt middleman behaviour. In order to be more precise about the boundaries of adaptation the aforementioned harshness value was again made use of. By making suppliers increasingly more difficult to deal with it was possible to determine at what point the ability to adapt broke down. In this particular situation it turned out to be at the point where suppliers start using a bulwark rather than conceder tactic.

Showing the fourth goal was a direct extension of work from the third. Taking optimised parameter sets from near the adaptation boundary PBIL initialisation data was formed. This information was then used in experiments beyond the boundary to try and give the system a head start in adapting to the new conditions.

8 Conclusions

A computer-based strategy is necessary to handle the large amount of information and transactions within a future electronic supply chain.

The SSCM provides a mechanism for the modelling of supply chains.

The SSCM represents a large problem space and therefore to tackle it effectively a strategy must be built out of known, succesful, components.

The SSCM Scenarios provide an incremental approach to developing strategies that tackle elements of SSCM and thus provide these components.

SSCM Scenario One provides a simple starting situation. A middleman strategy has been developed to tackle it.

Scenario One still allows for a wide range of environments and therefore it is necessary to optimise the strategy's parameters accordingly.

Scenario One's strategy parameters are optimised within an experimental system using a PBIL evolutionary algorithm.

Scenario One's strategy parameters are optimised differently according to starting conditions.

Optimised parameter sets perform best in their native environment.

There are constraints on how harsh initial starting conditions may be.

The boundaries of optimisation maybe pushed back by pre-configuring the PBIL from an optimised parameter set nearby within the boundaries.

The experimental system and its PBIL component provide a good mechanism to find the best way to tackle a given Scenario One situation.

Scenario One strategy provides a foundation for work on tackling more complex scenarios.

The experimental system provides a foundation for future experimentation with future scenarios and strategies.

The experimental system and PBIL component will provide a way to optimise parameters for future scenario strategies.

Acknowledgments

British Telecommunications PLC and the EPSRC jointly sponsor this work under a CASE studentship.

Bibliography

[1] M. He, N. R. Jennings and H. Leung, On Agent-Mediated Electronic Commerce, *IEEE Trans on Knowledge and Data Engineering 15 (4) 985-1003 (2003)*

[2] Sandholm, T.: Distributed Rational Decision Making. *In: Weiss, G. (ed.): Multiagent Systems A Modern Approach to Distributed Artificial Intelligence. MIT press 1999, pp. 201*

[3] W. E. Walsh, Market Protocols for Decentralized Supply Chain Formation, *Doctoral Thesis submitted to the University Of Michigan (2001)*

[4] P. Anthony and N. R. Jennings, A heuristic bidding strategy for multiple heterogeneous auctions, *Proc. 5th Int. Conf. on Electronic Commerce, Pittsburgh, USA (2003)*

[5] P. Faratin, C. Sierra and N. R. Jennings, Using similarity criteria to make negotiation trade-offs, *Proc. 4th Int. Conf. on Multi-Agent Systems (ICMAS-2000), Boston, USA, 119-126 (2000).*

[6] M. P. Wellman, A. Greenwald, P. Stone and P. R. Wurman, The 2001 Trading Agent Competition, *Fourteenth Conference on Innovative Applications of Artificial Intelligence (2000)*

[7] N. Matos, C. Sierra and N. Jennings, Determining Successful Negotiation Strategies: An Evolutionary Approach, *Proceedings of the 3rd International Conference on Multi-Agent Systems (ICMAS-98)*

[8] Shumeet Baluja, Population Based Incremental Learning – A Method for Integrating Genetic Search Based Function Optimisation and Competitive Learning, *(Tech. Rep. No. CMU-CS-94-163). Pittsburgh, PA: Carnegie Mellon University (1994)*

[9] R. Sukthankar, S. Baluja, J. Hancock, Multiple Adaptive Agents for Tactical Driving, *Applied Intelligence (1998)*

[10] M. Sebag, A. Ducoulombier, Extending population-based incremental learning to continuous search spaces, *In Th. Back, G. Eiben, M. Schoenauer, and H.-P. Schwefel, editors, Proceedings of the 5 th Conference on Parallel Problems Solving from Nature, pages 418--427. Springer Verlag, (1998)*

[11] I. Inza, M. Merino, P. Larra~naga, J. Quiroga, B. Sierra, M. Girala, Feature Subset Selection by Population-Based Incremental Learning. A case study in the survival of cirrhotic patients treated with TIPS, *Tehnical Report no. EHU-KZAA-IK-1/99, University of the Basque Country, Spain, (1999)*