

PhD in Open Constraint Satisfaction

Technical Report 6:

SSCM Experimental Results Analysis Systems

By

Timothy Gosling

Contents

1.1	INTRODUCTION.....	2
1.1.1	<i>Data Types</i>	2
1.1.2	<i>PBIL Distributions</i>	2
1.1.3	<i>Current Tools Summary</i>	3
1.1.3.1	PBIL Visualiser.....	3
1.1.3.2	PBIL Analyser.....	3
1.1.3.3	PBIL Entropy Measurer.....	3
1.1.3.4	SSCM Control Log Analyser.....	3
1.1.3.5	SSCM Integrity Checker.....	4
1.2	BASIC TOOLS.....	5
1.2.1	<i>The PBIL Visualiser</i>	5
1.2.2	<i>PBIL Entropy Measurement Tool</i>	6
1.2.3	<i>SSCM Integrity Checker</i>	6
1.3	MORE COMPLEX TOOLS.....	8
1.3.1	<i>The PBIL Analyser</i>	8
1.3.2	<i>Control Log Analyser</i>	10
1.4	FUTURE DIRECTIONS.....	13
1.4.1	<i>General Applicability Of Tools</i>	13
1.4.2	<i>New Tools – The SSCM Analyser</i>	13
1.4.3	<i>New Methods</i>	13
	REFERENCES.....	15

1.1 Introduction

At present various electronic market places, auctions and negotiation systems exist, in the near future full electronic supply chains will be possible and indeed desirable to improve efficiency [4]. This situation, however, presents a problem. While humans are good at negotiation and situation analysis there are less able to handle large volumes of information and numbers of transactions. What is needed is a computer-based strategy for handling these situations. The strategy does not need to be the perfect negotiation, although it must be competent, but it must be able to deal with more negotiations more rapidly than a human operator could. A core objective of this work is to develop strategies that are able to make a profit in a situation where customers are continually requesting bundles of products and may need to be negotiated with, suppliers must be negotiated with and there is a limit to both the communication capacity available and the amount of information about the market place.

The simple supply chain model (SSCM) was developed to allow the description of simple market scenarios with the aim of developing strategies to tackle such scenarios [5]. The SSCM however, while simple, is not trivial to tackle. As a result various scenarios have been developed to further restrict the use of the SSCM and so provide an incremental approach to tackling the problem [6]. A set of strategies has been developed to tackle these scenarios and a simulation system built to evolve these strategies in different environments [8][7].

Having run experiments and obtained a considerable amount of data the next problem is that of analysis and some consideration of what the results mean, if any trends can be discerned and so on. This document discusses the analysis tools that have been developed thus far and how they have been used; it further discusses future directions for the development of these analysis tools.

1.1.1 Data Types

The types of data available for analysis from experiments are as follows:

- Configuration Files; Customers, Suppliers and Middlemen – The customer and supplier configuration files essentially represent part of the experimental environment. These are stored along with the experiment generated data and may prove useful in later analysis. Middlemen configuration files are generated and stored for each middleman in the experiment each time an update is performed, these too might be helpful.
- Configuration Files; Controller – The controller configuration file is available for analysis in the same way the other configurations are. There is less variability between controller configurations in different experiments but it might yet prove valuable.
- PBIL distribution data – The PBIL distribution that is refined during the experimental process is stored each time it is changed. There is therefore a time-series of PBIL data available for review as well as the final PBIL vector.
- Agent Logs; Customer, Supplier and Middlemen – Each agent within the simulation maintains a running log of operation – this is primarily for debug purposes but also allows a view on what and when the agents did anything.
- Controller Logs – The central simulation controller generates all logs of its activity and records its interaction with the other agents. This information is particularly helpful as it effectively provides an overview of gross agent activity without resorting to the individual agent logs.
- Communication Log – The system can capture all communication messages generated during a simulation. This is normally never done since the information flow is extensive still it can be accomplished if required.

1.1.2 PBIL Distributions

The population based incremental learning (PBIL) distribution within the experimental system is based around the Flexi PIBL system developed during the course of this project. Flexi PIBL provides a class based highly configurable PBIL ([3]) distribution. For every variable in the problem to be solved, or in this case configuration parameter to be set, Flexi PIBL provides for the definition of one ‘cell’.

This cell can be of two types, symbolic or continuous. Symbolic cells are designed for variables that have one of a small set of discrete possible values; a probability is associated with each of these values in the cell. Continuous cells are designed for variables where a range is known (or can be guessed at) but there are no discrete steps between values. In continuous cells a set of ranges is defined with a probability associated with each range. The learning mechanism used updates the probabilities of associated cell values or ranges accordingly.

1.1.3 Current Tools Summary

The following is a complete summary of the tools currently in existence, the data they make use of and the output they generate. This is discussed in more detail within the remainder of the document.

1.1.3.1 PBIL Visualiser

Purpose	Data visualisation. Provides a convenient way to view PBIL distribution data.
Data Source	Single PBIL distribution file.
Output	'DNA finger print' style graphical display of the distribution. Text details for individual variables.
Applicability	Any system that uses PBIL could use this visualisation tool. Use of FlexiPIBL would make this even easier.
Possible Expansion	Ability to load and display multiple distributions for comparison. Inclusion of other PBIL data in display such as entropy.

1.1.3.2 PBIL Analyser

Purpose	Data visualisation. Comparison of PBIL distribution. Generation of aggregate PBIL distributions for experimental purposes. Correlation check of PBIL data.
Data Source	Single or multiple PBIL distribution files.
Output	Graphical display of the PBIL distribution on a per variable basis. Graphical and textual information about correlation between variables. PBIL distributions based upon multiple distributions loaded.
Applicability	Any system that uses PBIL could use this tool. Use of FlexiPIBL would again make this easier.
Possible Expansion	Limited. Improvement of aggregation mechanism. Some display improvement.

1.1.3.3 PBIL Entropy Measurer

Purpose	Test-bed for entropy measurement. Data visualisation. Secondary system for checking on experimental convergence.
Data Source	Single or multiple PBIL distribution files.
Output	For single distribution provides detailed entropy information on a per variable basis. For multiple distributions provides values plotted on a graph – this could therefore be used for time series display or inter-experiment comparison.
Applicability	Any system that uses PBIL could use this tool. Use of FlexiPIBL would again make this easier.
Possible Expansion	Limited. May prove useful as a test-bed for other measures.

1.1.3.4 SSCM Control Log Analyser

Purpose	Data Visualisation. Provides a simple way to view the complete history of an experimental run. Allows viewing of multiple runs for comparison. Allows observation of experiment convergence via PBIL entropy.
Data Source	Control Log files. Optional multiple PBIL distributions.
Output	Summary of experimental information such as start and end times etc. Graphical display of multiple data such as supplier goods price, middleman evaluations and PBIL entropy over time. All data can be saved to files.

<i>Applicability</i>	Only applicable to the SSCM experimental system.
<i>Possible Expansion</i>	Inclusion of other measures. Improvement of inter-experiment comparison. Addition of analysis tools. Possible use of experiment configuration files to elucidate results.

1.1.3.5 SSCM Integrity Checker

<i>Purpose</i>	Provides a quick way to check experiments for errors. Helps in tracking errors.
<i>Data Source</i>	Multiple Customer, Supplier and Middleman log files.
<i>Output</i>	Summary and list of all errors of different types detected in each file. Record reports to disk.
<i>Applicability</i>	Only applicable to the SSCM experimental system.
<i>Possible Expansion</i>	Improved error tracking with explanation and elimination of safe faults. Changes to make system more widely applicable.

1.2 Basic Tools

1.2.1 The PBIL Visualiser

The PBIL Visualiser is a tool designed for quickly assessing a PBIL vector visually. It provides a horizontal view of the PBIL vector in cell order. Each cell is of either symbolic or continuous type.

Symbolic cells are displayed graphically by having their values ordered lowest to highest from top to bottom – the values themselves are not displayed but each value is represented by a coloured horizontal bar. The bars colour represents the values probability – black being zero probability to white being a probability of one.

Continuous cells are displayed graphically in a similar manner to that of the symbolic cells but with important differences. The ranges of the continuous cell vary as do the associated probabilities with high probability ranges gradually becoming smaller. The display is organised in the same way as with symbolic cells but the range size is reflected along with the probability – larger ranges take up more of the display, smaller ones less. If the size of a range is small enough a marker is provided by the side of the display to indicate it's presence and probability.

There are few controls available over the display; it is however possible to scroll through the possible cells to view each and, by selecting a cell, a full account of a cell may be provided.

The overall use of the visualiser is to provide a quick view of PBIL information and to show where training has and hasn't occurred. The view is looks something akin to a DNA fingerprint and in this way can help with determining of certain trends or characteristics however no deeper analysis or comparison is allowed for.

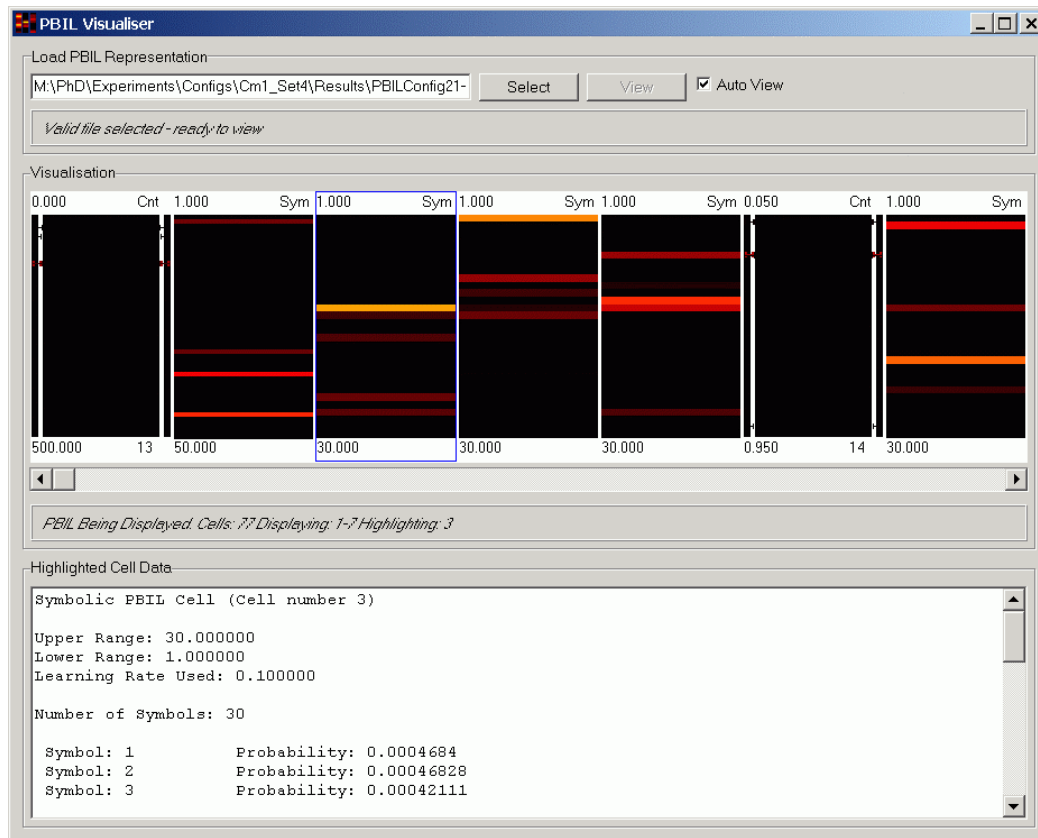


Figure 1: The PBIL Visualiser - The DNA fingerprint style can be seen. The third cell (a symbolic one) is highlighted.

1.2.2 PBIL Entropy Measurement Tool

The PBIL Entropy Measurement tool is a very simple analysis tool that acted as a source of additional functionality to other more complex tools. The tool supports two functions. The first function is to load and analyse a PBIL vector providing a entropy measurement of the PBIL vector as a single value (an largely untrained SSCM-S1 PBIL vector has an entropy value of , while a trained vector has an entropy around 330). The second function of the tool is to load multiple PBIL vectors and plot their entropies on a graph – if the PBIL vectors are ordered in time this allows the drop in entropy as training commences within the simulation system to be observed. Ordering is entirely up to the user however and any set of PBIL vectors can be selected.

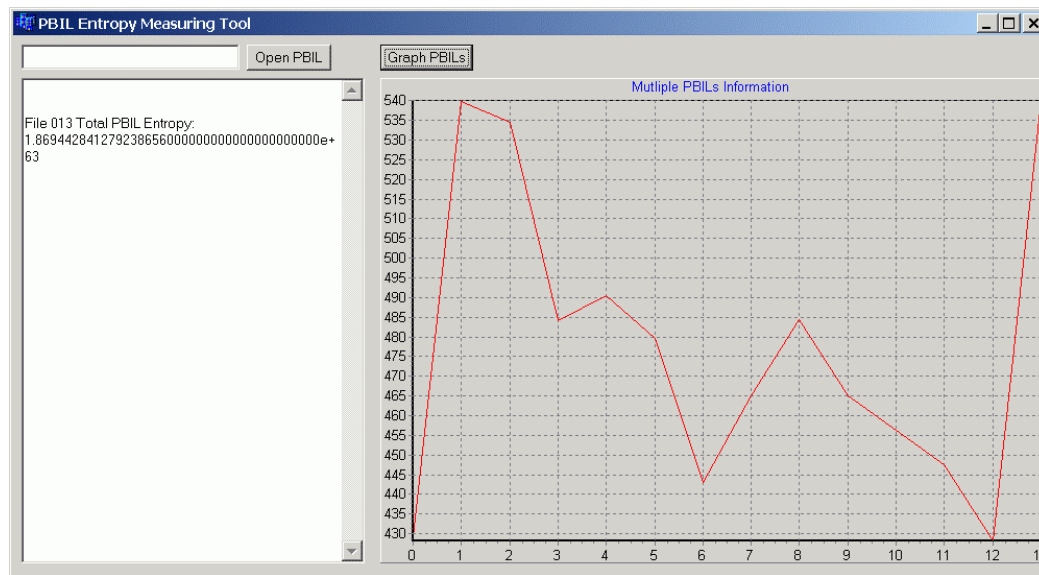


Figure 2: The PBIL Entropy Measurement Tool - A number of PBIL distributions have been loaded and their entropies displayed.

1.2.3 SSCM Integrity Checker

The SSCM Integrity Checker is a simple tool used for determining if an experiment has been running without error. The Checker makes use of the agent log files and, after extracting some limited information about the agents, basically performs a key word search on the log file looking for reported errors. The agents may generate four types of error message:

- **Warnings:** These may occur if something unfortunate but not critical occurs – for instance if an agent is unable to continue negotiations for some reason – this doesn't indicate a problem with the software as such.
- **Errors:** Errors indicate something has gone wrong that probably really shouldn't. It is still possible that the error was of a non-critical nature and all ought to be investigated.
- **Exceptions:** Exceptions are worse than errors and generally indicate some sort of critical failure within the software. These should generally speaking never occur but if they do it may indicate a fault in the software that was previously missed.
- **Unables:** Unables are a report of something an agent was unable to do for some reason, some of these may lead to and explain other reported errors. An unable report is not likely to represent a true error rather, an unfortunate circumstance the agent found itself in. They've been included to help diagnose true errors and track problem circumstances.
- **Fails:** Fails are similar to unables but might indicate a more serious problem.
- **Invalids:** Indicate errors associated with message sending.

The checker having performed its analysis provides output in a simple text report – these may be save for future reference.

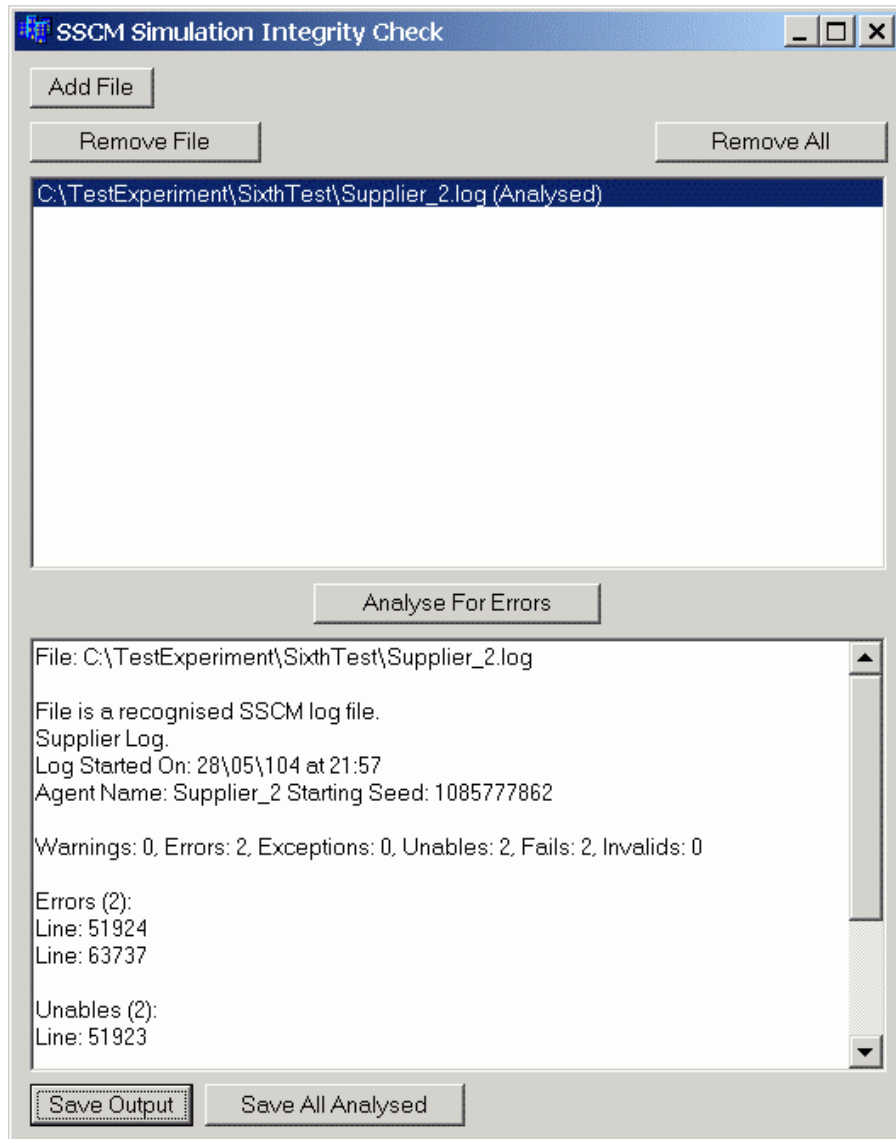


Figure 3: SSCM Integrity Checking Tool. Images shows the tool after analysing an Supplier file that contains some errors.

1.3 More Complex Tools

1.3.1 The PBIL Analyser

The PBIL Analyser tool provides three primary functions.

The first function is to allow multiple PBIL vectors to be displayed at one time. For the display to work all the PBIL vectors must be of the same length with cells of the same type and general configuration in each position. The analyser allows these cells to be viewed individually in aggregate form. If a single PBIL vector is viewed a more detailed graphical representation of each cell is thus available, if multiple PBIL vectors are used common features in each cell will stand out.

The second function of the analyser is to facilitate seeding of experiments with aggregate PBIL data. Some experiments have been run to push back the boundaries of agent adaptability by first training agents under less harsh conditions – these configurations can then be used as starting points for learning under more difficult conditions. Since the analyser outputs aggregate PBIL vector information it can be used to collect together training from multiple simulations in the same (or if necessary, different,) environments. For symbolic cells this is a simple matter of accumulation and normalisation, for continuous cells this is considerably more difficult.

The approach taken for determining aggregate continuous cell PBIL information is based around using normal PBIL learning. Aggregate information is generated by first finely slicing up the range space and determining the aggregate, normalised probabilities within those chunks. By using roulette wheel type selection on this information the PBIL continuous cell can be trained to mimic this data. Various parameters are available to control how this learning takes place and these can strongly influence the success of the mimicry.

The third function of the analyser is to provide variable correlation information. The correlation analysis is performed between all variables in all selected PBIL vectors in an attempt to determine which variables are connected in some way. This provides an interesting insight to the way the configurations respond to changes in the environment and in each other.

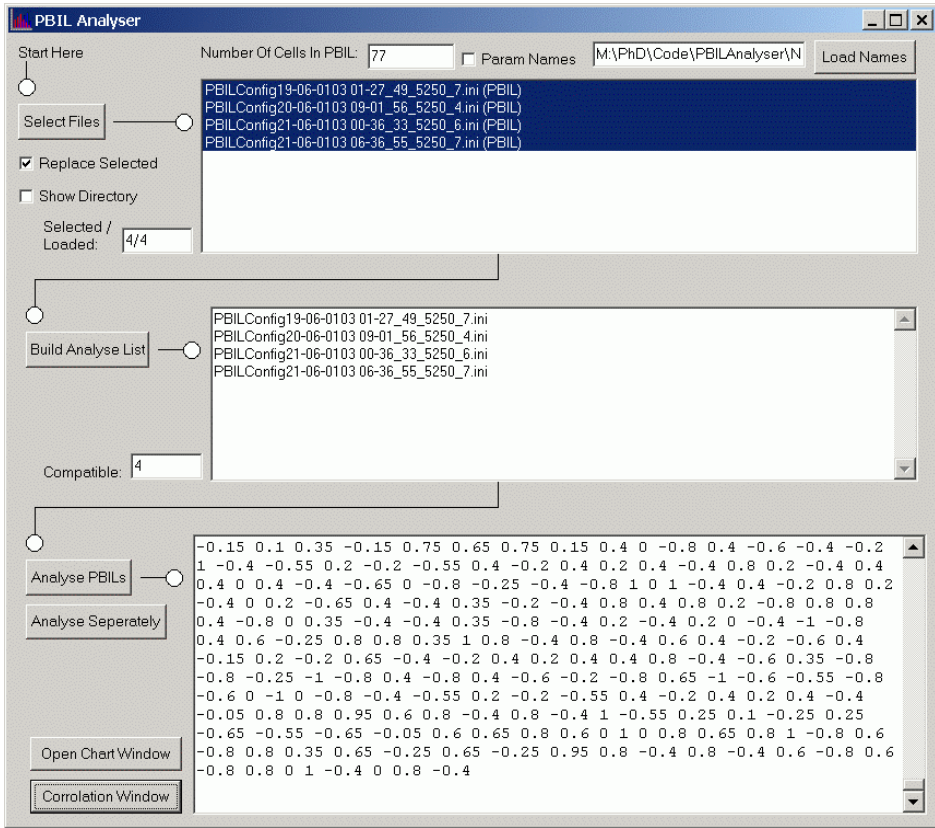


Figure 4: The PBIL Analyser main window - Four PBIL distributions have been loaded and analysis carried out.

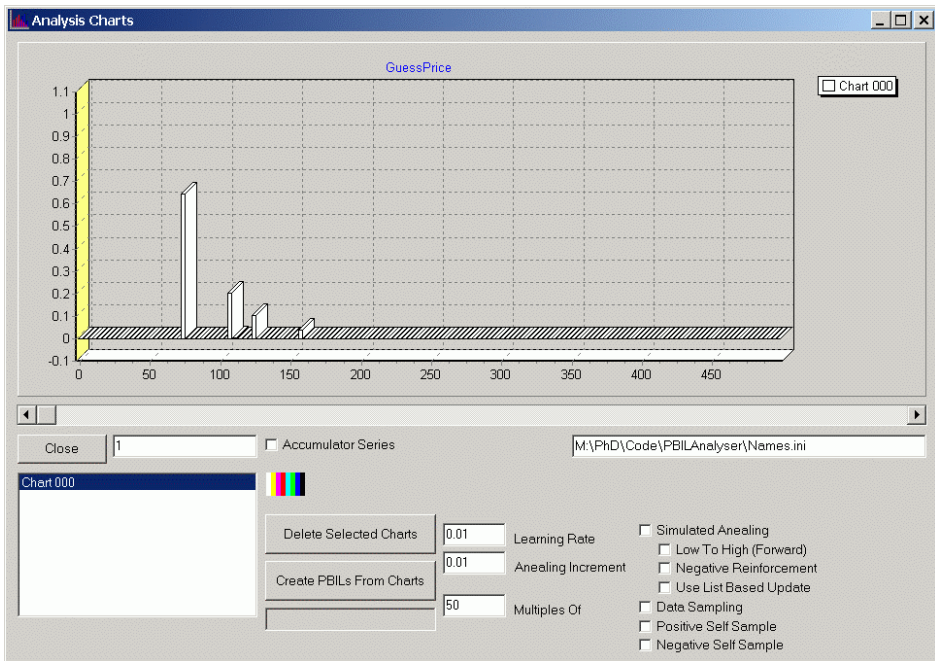


Figure 5: The PBIL Analyser cell display - The first (guess price) cell is displayed. It would appear values around 75 are favoured for this cell in this environment.

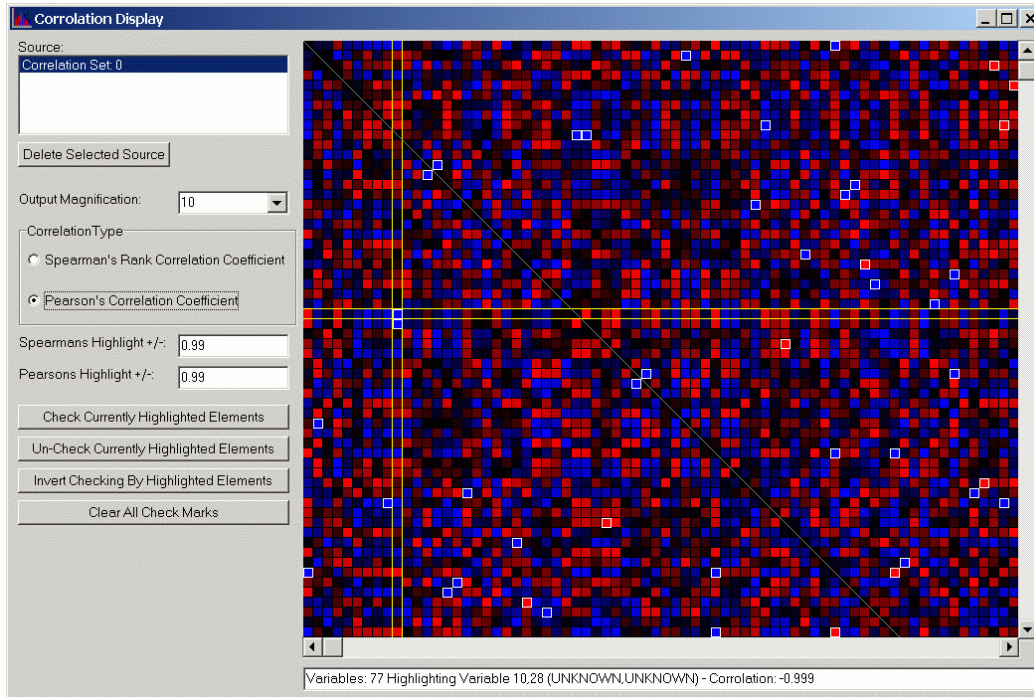


Figure 6: The PBIL Analyser correlation window. While not enough files have really been loaded here, correlation between some variables can be seen with a high threshold.

1.3.2 Control Log Analyser

The control log analyser is designed to provide a simple way to view information about an entire experimental run. The information it displays is based upon the Control Logs generated by the controller that houses the PBIL distribution, hence the name. As part of its operation the control log analyser is able to load all PBIL files related to the experiment in question if they are available – this allows for a plot of entropy over time as will be seen later.

The control log analyser may load information from any number of control logs but may only display information about any one at a time. The information displayed takes two forms, a simple summary of the experiments and more detailed graphical information about the different elements.

The summary display is shown below and contains the following elements:

- The list of middlemen and suppliers in the experiment (customers are not listed as analysis of these are not undertaken).
- Start and end times for the experiment along with the experiments own internal clock end time.
- Information about learning, learning rate, number of evaluations of different types and so on.
- A list of all related PBIL files.

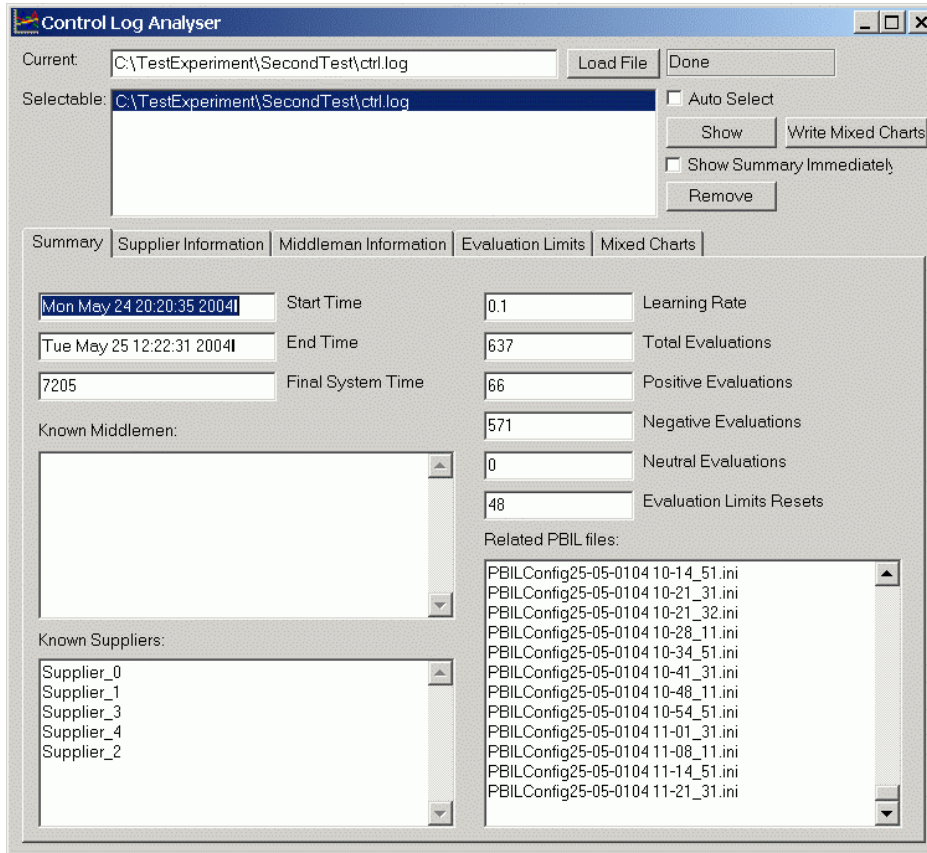


Figure 7: The Control Log Analyser. The summary display is shown.

The supplier, middleman and evaluation limits information is in the form of a series of selectable charts. All of these are combined under ‘mixed charts’ that allows viewing across all of these. The data available in these charts is all shown over the course of the experiment and is as follows:

- Supplier goods price – the average price paid by middlemen for goods from each supplier
- Number of middleman evaluations – this should read constant, if it doesn’t an error has occurred.
- The mean and median middleman evaluations
- The highest and lowest middleman evaluations
- The locally highest and lowest middleman evaluation (within that time period)
- The number of positive and negative evaluation results
- The upper and lower acceptability boundaries during evaluation
- The sample mean used for the ‘safe’ zone
- The PBIL entropy

The last of these ‘PBIL entropy’ is useful for showing that learning (of some form) is occurring and that the system is heading for convergence. Changing the learning parameters or the market environment changes the shape of the entropy drop curve.

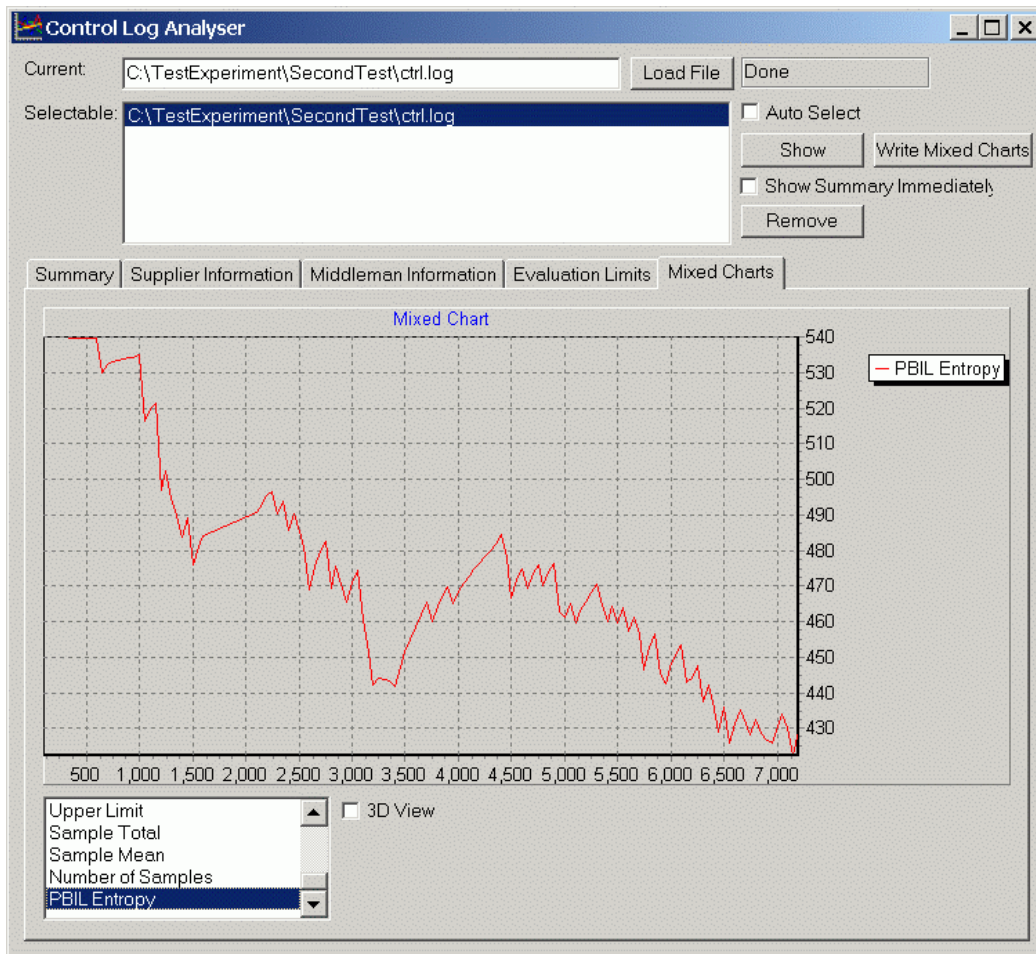


Figure 8: The Control Log Analyser Tool. A PBIL entropy curve over time is shown.

1.4 Future Directions

1.4.1 General Applicability Of Tools

The tools discussed in this document have been developed for the SSCM work but some are applicable and potentially useful beyond it.

The PBIL visualiser, analyser and entropy measurer all use simple PBIL distribution information as their input – this means they are readily applicable to any other project that makes use of PBIL and is capable of outputting the distribution in the required form. The developed FlexiPIBL component could be used by other projects easily and incorporates this ability. For this reason it may be worth making the FlexiPIBL component readily available.

The SSCM control log analyser and integrity checker are unlikely to be useful in other projects or areas of work. The control log analyser while useful for all foreseeable SSCM experiments is tied too closely to the agent logging format, the same could be said of the integrity checker. The integrity checker tool maybe more widely applicable for searching out key words within files but is not sophisticated enough to warrant any distribution at this stage.

1.4.2 New Tools & The SSCM Analyser

Currently under development is an SSCM analyser. This tool essentially acts to bring all available information about an experiment together in one place and make use of it. This tool is intended to act as the primary direction of expansion in the near future. The analyser makes use of components from several of the previous tools. For instance it makes use of and expands upon elements of the control log analyser. The control log essentially acts as a core data source of the new system but additional information can be made use of. Configuration files can be loaded and multiple control logs associated with them. Analysis can thus be performed across multiple experiments in a single given environment or indeed across multiple experiments in different environment. Correlation is the primary method of analysis used so far although other methods will be added at a later date.

Important in the development of the SSCM Analyser and other tools is the idea of reusable modular components. The FlexiPIBL component embodies this well and is easily reusable, for this reason some analysis elements have now been directly incorporated in FlexiPIBL. Future analysis components will adopt a similar approach hopefully providing general useful analysis tools that other projects might make use of.

The existing tools maybe further improved in the future. The visualiser could be improved with the ability to compare multiple PBIL vectors for instance. Most effort will however be directed into the new SSCM Analyser tool as this is more likely to aid in experimental analysis in both the short and long term.

1.4.3 New Methods

Currently PBIL entropy and variable correlation (across different data sets) are the main forms of analysis. This is combined with visualisation to aid in examining experimental results. This however is not enough and new techniques need to be sort. Some of these may be applicable to in-experiment operation.

Some of the questions that ought to be answered by these techniques are as follows:

- Cycles – Does the learning process include cycles of operation, do strategies evolve in one way only to change back again afterwards in a continual cycle? This is a possibility as competition effectively changes the landscape.
- Adaptation – How do the strategies evolve? Trends seem to be apparent in the data but how do we put numbers on this when the environment is complex, there are many potential solutions and not many experiments to work with.

- Convergence – Is entropy a good measure of convergence? It is possible for the system to learn without learning anything terribly useful.
- Clusters and group behaviour – Can we see how multiple elements of a strategy work together to perform effectively, can we determine what elements are unimportant.
- Range – Can we determine how widely applicable a strategy is? Experiments have already been run to ‘push back’ the adaptation boundary – can we determine how widely applicable a strategy might be to different environments without running a simulation? Can we determine what trade offs are need for wider applicability.

References

- [1] Simple Constrained Bargaining Game, Edward Tsang, Tim Gosling, 2002
- [2] Determining Successful Negotiation Strategies: An Evolutionary Approach, Noyda Matos, Carles Sierra, Nick Jennings
- [3] Population Based Incremental Learning – A Method for Integrating Genetic Search Based Function Optimisation and Competitive Learning, Shumeet Baluja
- [4] On Agent-Mediated Electronic Commerce, Minghua He, et al, 2003
- [5] Technical Report 1: The Simple Supply Chain Model (SSCM), Tim Gosling, 2003
- [6] Technical Report 2: SSCM Scenarios, Tim Gosling, 2--3
- [7] Technical Report 3: SSCM Scenario One Strategy, Tim Gosling, 2003
- [8] Technical Report 4: Implementing Scenario One Strategies And The Use Of Evolutionary Computation, Tim Gosling, 2003
- [9] PhD in Open Constraint Satisfaction: SSCM, A Position Statement, Tim Gosling, 2004