

# GANETXL: A DSS GENERATOR FOR MULTIOBJECTIVE OPTIMISATION OF SPREADSHEET-BASED MODELS

**Dragan A. Savić<sup>1</sup>, Josef Bicik and Mark S. Morley**

The Centre for Water Systems, College of Engineering, Mathematics and Physical Sciences, University of Exeter,  
North Park Road, Exeter, Devon, EX4 4QF, United Kingdom

## Abstract

*Water management practice has benefited from the development of model-driven Decision Support Systems (DSS), and in particular those that combine simulation with single or multiple-objective optimisation tools. However, there are many performance, acceptance and adoption problems with these decision support tools caused mainly by misunderstandings between the communities of system developers and users. This paper presents a general-purpose decision-support system generator, GANetXL, for developing specific applications that require multiobjective optimisation of spreadsheet-based models. The system is developed as an Excel add-in that provides easy access to evolutionary multiobjective optimisation algorithms to non-specialists by incorporating an intuitive interactive graphical user interface that allows easy creation of specific decision-support applications. GANetXL's utility is demonstrated on two examples from water engineering practice, a simple water supply reservoir operation model with two objectives and a large combinatorial optimisation problem of pump scheduling in water distribution systems. The two examples show how GANetXL goes a long way toward closing the gap between the achievements in optimisation technology and the successful use of DSS in practice.*

---

<sup>1</sup> Corresponding author, e-mail: [d.savic@exeter.ac.uk](mailto:d.savic@exeter.ac.uk), Tel. +44 1392 723637, Fax: +44 1392 217965

## 1. INTRODUCTION

Nowadays computer-based modelling is used routinely as part of decision-support processes in many areas of business and engineering. Decision Support Systems (DSS), and model-driven DSS in particular (Power and Sharda, 2007), that could be defined simply as “computer-based models together with their interactive interfaces” (Loucks, 1995), aim to support the user in addressing and solving particular unstructured problems in a timely manner (Scott Morton, 1971). Although there is no widely accepted definition of DSS, these systems can integrate a number of technologies (tools), including optimisation and simulation models, geographic information systems, artificial intelligence algorithms, data-mining tools, expert or knowledge-based systems, statistical and graphing tools, etc. From the early beginnings when computers were utilised as mere calculators to the so-called ‘fourth generation models’, which are widely used commercial software that includes databases, models and policy analysis instruments (Abbott, 1991), water management practice has benefited from the development of DSS (Loucks, 1995). Most notably, DSS that integrate Evolutionary Multiobjective Optimisation (EMO) with simulation technologies, have gained in popularity in the water resources literature over the last few decades (Ritzel et al., 1994; Halhal et al., 1997; Reed et al., 2001; Farmani et al., 2005; Nicklow et al., 2010). The key advantage of EMO is that, unlike classical mathematical programming methods where objective functions may be optimised separately, these population-based methodologies have the ability to search effectively for many non-dominated (trade-off) solutions at the same time (in a single run), which makes them a very attractive tool for solving multiobjective optimisation problems (Fonseca and Fleming, 1995). However, despite the substantial growth in DSS development in the last few decades (Eom and Lee, 1990; Loucks, 1995; Choi et al., 2005; Berlekamp et al., 2007; Giupponi, 2007; de Kok et al., 2009; Hadihardaja et al., 2010), successful use of DSS in practice remains limited. The most often cited reasons for that are behavioural and technical issues, which impact DSS performance, acceptance and adoption (Power and Sharda, 2007; de Kok et al., 2009).

Most bottlenecks to DSS use in practice seem to stem from misunderstandings between the communities of developers and users (Borowski and Hare, 2007). This in turn is often a result of: (i) the lack of involvement of the potential users in the development of DSS, and (ii) the need to involve specialists in operating them. Savenije (1995) points out that in some cases the use of DSS was so dependent on the people who developed and operated them that “the tools became useless as soon as the operator left the office”. Loucks (1995) advocates that to build effective DSS, developers “must understand not only the technical aspects of the models and computer systems but also, and most importantly, the particular social and economic characteristics of the potential users and their institutions”. Along the same lines, Van Delden (2009) emphasises the development process and its role in the actual uptake of DSS by recommending it should: (i) be based on close interaction between the different parties involved, and (ii) provide prototypes throughout the development process. Van Delden (2009) also recommends using an iterative development process that leads to an improved uptake of the system in the organisations and overcomes problems related to the institutional acceptance and the willingness of people to use and adopt a DSS as part of their daily practice. The obvious solution to the abovementioned problems would be to allow those responsible for making water management decisions, i.e., potential users, to develop their own DSS. However, intended users (managers and policy planners) are rarely skilled in writing optimisation and simulation models or integrating the relevant technologies flexibly and quickly to obtain the necessary support for making decisions (Scott Morton, 1971; Loucks, 1995).

This paper introduces a tool that can help bring the power of evolutionary optimisation closer to users who may be knowledgeable about the practical decision-making problem, but are inexperienced in developing and using optimisation methods. It is envisaged that in turn this will reduce the gap between developers of optimisation methods and their users.

## 2. BACKGROUND AND REQUIREMENTS

### 2.1. Decision Support System Generators

A set of software that provides tools and capabilities to help developers build a specific DSS is called a “DSS generator” (Sprague, 1980). These tools could serve as a foundation for building multiple kinds of model-based DSS, including those incorporating multiobjective optimisation technologies. Power and Sharda (2007) list a number of commercially available DSS generators for optimisation applications (e.g., the AIMMS Modelling System, [www.aimms.com](http://www.aimms.com); IBM ILOG Optimization, [www.ibm.com](http://www.ibm.com); MPL Modelling Language, [www.maximalsoftware.com](http://www.maximalsoftware.com)). Evolutionary computing methods have been incorporated in several of those tools, e.g., Evolver ([www.palisade.com](http://www.palisade.com)) and GeneHunter ([www.wardsystems.com](http://www.wardsystems.com)). Many of these DSS generators include a number of optimisation techniques and tools, and are provided either as stand alone application environments and/or as spreadsheet add-ins (to allow easy DSS prototyping). However, what is missing in this commercial software landscape is a DSS generator that integrates EMO technologies flexibly and quickly to construct robust prototype applications.

A typical spreadsheet package, such as the ubiquitous Microsoft Excel, is the most familiar example of a DSS generator for creating model-driven DSS. The use of Excel by practicing engineers for everyday modelling tasks has become widespread over the last few decades because it provides virtually all of the graphical user interface, database, modelling, data analysis and programming tools required for creating small to medium size DSS with minimal effort. The introduction of a macro language and Visual Basic for Applications (VBA) together with add-in programs, makes Excel an even more powerful and useful generator of model-based DSS. Savenije (1995) lists twenty spreadsheet-based models for water resources modelling and planning, including groundwater modelling, reservoir routing, water resources analysis, unsteady flow modelling, modelling non-point source pollution in watersheds, decision-support tools for river basin planning, etc. More recently, de Condappa et al. (2008) report on a spreadsheet-based water balance model providing monthly estimates of major water movement, uses and losses within a

river basin. They conclude that spreadsheets can be powerful tools for supporting decision making for water allocation by identifying under-utilised resources and by allowing trade-offs to be evaluated in water scarce basins.

Thanks to the availability of the Solver add-in for Microsoft Excel, the popularity of spreadsheets for solving optimisation problems has increased to the point that Solver is probably the most widely used general purpose optimisation modelling system. The add-in employs a generalised reduced gradient algorithm (Lasdon et al., 1978) and can solve small linear and nonlinear optimisation problems as well as mixed integer programming problems. However, in addition to limits on the size of problems that can be tackled (200 decision variables, 100 explicit constraints and 400 bound constraints) and the relatively low execution speed (when compared to the compiled, stand-alone DSS), Solver does not provide any multiobjective optimisation capabilities. Recently, Balter and Fontane (2008) developed a multiobjective particle swarm optimisation algorithm for a reservoir operation planning problem with four objectives. Although they suggest that the algorithm, which was coded in VBA and linked to a spreadsheet-based simulation model, could be installed as an add-in to Excel, no indication was given on how this could be done to create a general purpose DSS generator.

## **2.2. Evolutionary Multiobjective Optimisation (EMO)**

Evolutionary algorithms, which are metaheuristics inspired by the natural evolution process involving *natural selection* and *population genetics*, have become the method of choice for optimisation problems that are too complex to be solved using classic mathematical optimisation techniques (Nicklow et al., 2010). The most well known of them, Genetic algorithms (GA), evolve a *population* of solutions to the optimisation problem through an iterative application of randomised processes of *selection*, *recombination* (also referred to as *crossover*) and *mutation* (Goldberg, 1989). Unlike traditional optimisation methods that may require simplification of the problem (e.g. linearisation), calculation of derivatives or matrix inversion, GAs require only that the *fitness* of a solution can be evaluated for a given set of decision variables. In addition, they are easy to implement, robust and inherently parallel. However,

there is no guarantee that the global optimum will be found using GA although the number of applications suggests a good rate of success in identifying good solutions (Nicklow et al., 2010). Their ability to handle both single and multiple objectives and constraints makes them attractive to the decision making processes in many areas of water engineering.

Most water engineering decision-making problems need to achieve multiple objectives, e.g., maximisation of benefits, minimisation of costs, minimisation of risks, maximisation of reliability, minimisation of deviations from desired performance levels, etc. (Haimes, 1998). The goal of multiobjective optimisation is to investigate trade-offs between the problem's conflicting objectives (Nicklow et al., 2010). Unlike single-objective optimisation, whose aim is to find the 'best solution' by aggregating all different objectives into one, multiobjective optimisation aims to find a set of compromised solutions, largely known as the Pareto-optimal solutions (Fonseca and Fleming, 1995). A Pareto-optimal solution is one that is better than any other solution in at least one objective. The entire set of such solutions is called a Pareto-optimal set whose 'front' is obtained by plotting solutions according to their objective values, yielding an  $M$ -dimensional surface, where  $M$  is equal to the total number of objectives (Nicklow et al., 2010). By running a population of solutions in parallel, multiobjective evolutionary algorithms can evolve entire trade-off (Pareto) surfaces within a single run even for large mixed-integer optimisation problems (Halhal et al., 1997).

However, developing multiobjective optimisation models or implementing optimisation tools in practice, requires a good level of programming skills and/or thorough understanding of the optimisation methodologies on which the tools are based. This often results in a disproportionate amount of time being spent on debugging and polishing models as compared to time spent on creativity (Savenije, 1995).

### **2.3. Requirements**

In order to bring the power of multiobjective optimisation closer and faster to the intended users, a new DSS generator called GANetXL, which will address some of the difficulties associated with the

development and use of model-based DSS in water engineering practice, has been envisaged. The basic requirements considered for the new DSS generator are:

1. To provide easy access to efficient EMO optimisation algorithms to users who are optimisation non-specialists in order that they could consistently find good Pareto solutions to problems. These solutions are expected to be better than those that could be found through trial-and-error experimentation (i.e., using simulation only);
2. To develop a new general-purpose DSS generator as an Excel add-in thereby allowing a large number of users to take advantage of integrating powerful optimisation with modelling capabilities of spreadsheet technology;
3. To provide an intuitive Graphical User Interface (GUI) that will allow the easy creation of specific DSS applications. The interface should allow inexperienced users to define an optimisation problem, configure and execute an optimisation run and analyse the obtained results through intuitive visualisation of Pareto-optimal solutions in both decision and objective spaces;
4. To minimise the need for complex coding of the interface between optimisation and any simulation routines that have to be used to evaluate potential solutions.

### **3. STRUCTURE AND FEATURES**

The above mentioned bottlenecks to model-based DSS development and use in practice were the principal motivation for the development of GANetXL (Bicik et al., 2008). Although a number of commercial (e.g., Evolver and GeneHunter) and non-commercial (Schreyer, 2006) tools exist that incorporate GAs as a global search technique into a spreadsheet environment, none of these tools employs evolutionary optimisation algorithms that tackle multiobjective problems.

GANetXL has been built upon a robust optimisation framework (Morley et al., 2001) developed within the Centre for Water Systems (CWS) at the University of Exeter for over a decade. The tool combines the strength of single-objective and multiobjective optimisation using a GA with an interface

that allows the easy creation of a specific DSS that uses a GA to formulate and optimise the problem at hand. For single-objective problems GANetXL provides a family of steady-state, generational and generational elitist genetic algorithms (Goldberg, 1989). In the domain of multiobjective problems, the NSGA-II algorithm (Deb et al., 2002) is currently supported. Since its launch in 2006, GANetXL has been used by more than 250 researchers from 35 countries all over the world. Several applications of the GANetXL in the water engineering area already exist, including the development of a model-based DSS for optimal management of groundwater contamination (Farmani et al., 2008), optimal design of water distribution systems (Deepthi et al., 2009; Čistý and Bajtek, 2009), integrated water resources management (Molina et al., 2010), planning renewal of water distribution systems (Nafi and Kleiner, 2010), and optimisation of water recycling schemes (Rozos et al., 2010).

Formulating a problem using the tool requires the user to create a spreadsheet using Microsoft Excel (a set of example problems and templates are provided with the tool's installation file) and to configure the GA parameters using the GANetXL interface. The software structure, which is implemented in GANetXL to provide an intuitive interface and allow the easy formulation of optimisation problems, is illustrated in Figure 1. There are four key components of the DSS generator: (1) the *Toolbar*, (2) *Configuration Wizard*, (3) *Excel spreadsheet model*, and (4) *Interactive visualisation interface*.

### **3.1. Toolbar**

The Toolbar contains five control buttons (*Configuration*, *Run*, *Resume*, *Results* and *About*), each of them controlling one of the key components of GANetXL:

- *Configuration Wizard* – used to configure the application.
- *Run* – starts the optimisation or attempts to recover crashed computation from a backup file (if automatic backup is enabled).
- *Resume* – resumes suspended optimisation.
- *Results* – displays the Pareto front in the EMO version.
- *About* – displays information about the software version, license, expiration date and limitations.

**(Figure 1 approximately here)**

The Configuration Wizard guides the user through the required steps and options to provide parameters required by any of the evolutionary algorithms implemented within GANetXL. After configuring all the parameters the optimisation algorithm can be executed by pressing the *Run* button on the Toolbar (Figure 1).

### **3.2. Configuration Wizard**

The Wizard has three main tabs for fulfilling the configuration tasks (*Genetic Algorithm*, *Excel Link* and *Options*). It is configured to provide default values for all EMO configurations and parameters, thus allowing even inexperienced users to set up the optimisation problem, but also to allow experts to fine-tune the algorithms and genetic operators to experiment with different setups. For example, in Figure 1, although there are six tabs for configuring the *Genetic Algorithm* parameters (*Type*, *Population*, *Algorithm*, *Crossover*, *Selector* and *Mutator*), the only essential information is whether the *Single Objective* or the *Multiple Objectives* option is required. An experienced user could in addition set the population size, select the type of GA, type and rate of crossover and mutation, as well as selector type. In the case of a steady-state GA the user can furthermore choose from a wide range of replacement operators.

**(Figure 2 approximately here)**

The *Excel Link* tab in the Configuration Wizard associates the selected GA optimisation algorithm with the Excel spreadsheet model by providing information on: (i) decision variables, i.e., *Chromosome* tab, (ii) objectives, and (iii) constraints (Figure 2). GANetXL supports up to 250 integer or real decision variables under Microsoft Excel 2000, XP and 2003. However, with the latest versions of Microsoft Excel 2007 and 2010, users can work with more than 16,000 decision variables. Internally, all decision variables are encoded as binary strings and genetic operators are applied while ensuring that their bounds are satisfied. In addition, the *Simulation* tab allows an external simulation model to be called for fitness evaluation (via a VBA macro), while the *Write Back* tab returns the current GA information into specific

spreadsheet cells to apply user defined penalty multipliers as a function of the optimisation progress. The possibility of calling a VBA macro to evaluate the fitness of a solution enables users to integrate easily other models into the optimisation process or perform multiple model runs. The interfacing can be carried out using a Dynamic-link library (DLL), Component Object Model (COM) or by invoking a standalone executable. A DLL can be written in, e.g., C, C++, Matlab or FORTRAN, whereas a COM object can be implemented in C++, VB or the Microsoft .NET family of languages (e.g., C#, VB.NET, etc.). The use of standalone executables, which typically read inputs from one file and write model outputs to another file, comes with a significant performance penalty unlike the two other approaches (i.e., DLL and COM). The decision variables are defined (e.g., cell B9 entered in *Genes Range*, in Figure 2) together with the gene type (e.g., Integer Bounded, Real Bounded, etc. in Figure 2) and lower and upper bounds on each of the decision variables. It is also necessary to provide the location where in the spreadsheet the formulae for objectives are to be found (e.g., cell range B12:B13 in Figure 1), the type of objective (i.e., minimisation or maximisation) and the name of the objective (for graphing purposes). If constraints are required, a spreadsheet cell that contains the penalty formula must also be specified.

Finally, the *Options* tab provides a host of advanced options for configuring an optimisation run, including the number of batch optimisation runs, number of generations in each run, ability to change the random generator seed value, etc. The full list of options is available in the GANetXL User Manual (CWS, 2010).

The configuration data is saved in a worksheet that is created automatically after the *Configuration* button is pressed. This worksheet is by default hidden as the user has the Wizard to modify the configuration of the DSS. However, experienced users are able to generate the configuration sheet dynamically using VBA.

### **3.3. Excel Spreadsheet Model**

The model of the optimisation problem in hand should define decision variables, objectives and constraints, as well as any other relationships between them. The only requirement a spreadsheet file must

meet to be compatible with GANetXL is that it contains one worksheet named “*Problem*”. This is the worksheet which defines the model, including the cell locations for decision variables, objectives and constraints. Modern spreadsheets, such as Microsoft Excel, provide a number of chart options to facilitate visualisation of the decision space.

Visualisation of the objective space during the optimisation run is achieved through an optimisation progress form shown in Figure 3. It displays the solutions which have been obtained in the form of a grid containing values of decision variables (genes, e.g. G1), objective functions, penalty/infeasibility and other statistical indicators. To visualise the progress of an optimisation run in the objective space, a chart is used to display the best Pareto front (in EMO) or just the fitness of the best solution (in single objective optimisation). The visualisation of a Pareto-optimal set is currently only possible in two dimensions (2D) by selecting which two objectives will be displayed, as shown in Figure 3. With more than two objectives this 2D representation can be changed by selecting different combinations of objectives from the drop-down menu at the top of the GUI form.

**(Figure 3 approximately here)**

### **3.4. Interactive visualisation interface**

The results obtained from the optimisation run(s) are automatically saved in a separate worksheet(s) of the same spreadsheet file after the end of computation. The Pareto-optimal front can then be visualised by activating the model worksheet and then by invoking the “Results” option in the GANetXL toolbar. The *Interactive visualisation interface* is also limited to 2D representation and allows easy changing of displayed objectives on the X and Y axes (Figure 4). In addition to manipulating the plotting axes, the user can choose whether the whole Pareto set should be displayed or only its subset by zooming into the desired area of the graph. If several runs of the same Excel model were carried out (e.g., using different optimisation parameters), the resulting Pareto-optimal sets can be plotted by selecting the name of the worksheet (check box on the left of the form) where the solutions of the particular run were saved. Figure

4 also illustrates how by moving the mouse over a point on the Pareto set the objective function values are displayed, e.g., “Yield: 40.00” and “Recreational Benefit: 200”.

**(Figure 4 approximately here)**

A user can interact with the solutions displayed by moving a mouse over an individual solution and clicking on it, which automatically updates the model in the Excel worksheet to show the change in decision variables and all other related model parameters. This interaction facilitates both objective and decision space probing (Kollat and Reed, 2007).

#### **4. RESERVOIR OPERATION APPLICATION**

The ability of GANetXL to quickly create a specific model-based DSS to optimise a water supply reservoir operation problem will be illustrated on a simple hypothetical example with two objectives. Figure 5 shows a spreadsheet created to perform the water balance computations over 10 time steps (column D) for a sequence of monthly inflows (column E). For simplicity, all the values in the spreadsheet are given in volumetric units. In this example, the goal is to find the maximum yield (amount of water) that the reservoir can supply constantly throughout the period (cell B9, which is then copied to cells F4:F13), while maximising the storage levels as a surrogate for recreational benefit (assumes that higher water levels are more desirable for recreational purposes). The water balance equation (cell range G3:G13) is expressed as:

$$\text{Storage}(t+1) = \text{Storage}(t) + \text{Inflow}(t) - \text{Supply}(t) \quad \text{Eq. 1}$$

where:

- $t$  – time step for water balance computations
- $\text{Storage}(t)$  – storage at the end of time step  $t$
- $\text{Inflow}(t)$  – inflow during time step  $t$
- $\text{Supply}(t)$  – volume supplied during time step  $t$

For example, the balance equation is implemented in the spreadsheet for cell G4 as:

$$\text{IF}(G3 + E4 - F4 > \$B\$5, \$B\$5, G3 + E4 - F4) \quad \text{Eq. 2}$$

To account for the constraint that the water stored cannot exceed the reservoir capacity (*Reservoir Capacity* in cell B5), the following equation is used to calculate the spilled volume (cell range H4:H13),  $\text{Spill}(t)$ :

$$\text{Spill}(t) = \begin{cases} 0, & \text{If } \text{Storage}(t) \leq \text{Reservoir Capacity} \\ \text{Storage}(t) - \text{Res Capacity}, & \text{If } \text{Storage}(t) > \text{Reservoir Capacity} \end{cases} \quad \text{Eq. 3}$$

The minimum storage constraint is defined through the water supply deficit, which occurs when reservoir storage falls below the *Dead Storage* (cell B4):

$$\text{Deficit}(t) = \begin{cases} 0, & \text{If } \text{Storage}(t) \geq \text{Dead Storage} \\ |\text{Storage}(t) - \text{Dead Storage}|, & \text{If } \text{Storage}(t) < \text{Dead Storage} \end{cases} \quad \text{Eq. 4}$$

An additional constraint is introduced to ensure continuity of operation, thus requiring that the reservoir has completed the cycle at the end of time period  $t=10$  at the same level from which it started. This is achieved by modifying the deficit constraint (Eq. 4) for that time period to:

$$\text{Deficit}(t = 10) = \begin{cases} 0, & \text{If } \text{Storage}(t = 10) \geq \text{Storage}(t = 0) \\ |\text{Storage}(t = 10) - \text{Storage}(t = 0)|, & \text{If } \text{Storage}(t = 10) < \text{Storage}(t = 0) \end{cases} \quad \text{Eq. 5}$$

The sum of all deficits (Total Penalty in cell I14) is then computed (Figure 5) and this is used to penalise solutions with supply deficit. In the example (Figure 5), the total deficit of 4 volumetric units contains the deficit calculated at the end of period  $t=2$  where the storage falls to zero (below the Dead Storage) and at the end of period  $t=10$  where the end storage level is below the starting storage level.

The two objective functions can be expressed as:

1. Maximise Yield (cell B12), which is calculated as the sum of supply volumes, i.e.,  $\text{SUM}(F4:F13)$ , and
2. Maximise Recreational Benefit (cell B13), which is calculated as the sum of all storage levels, i.e.,  $\text{SUM}(G4:G13)$ .

To optimise this model, one only needs to select the *Multiple Objectives* option in the Configuration Wizard and provide information where the decision variable, i.e., constant supply (cell B9), two

objectives (cells B12:B13) and the penalty (infeasibility) cell (I14) are located. In addition the type (e.g., integer, real, etc.) and bounds for the decision variables need also to be specified.

**(Figure 5 approximately here)**

First, the optimisation problem is solved under the assumption that the decision variables are discrete (Integer Bounded in Figure 2), i.e., the supplied amount during each time step can only take discrete values. This is done for illustrative purposes, i.e., to limit the number of solutions in the Pareto set and allow easier analysis of the trade-off solutions. The visualisation of the two-objective Pareto-optimal set is shown in Figure 4, with “Yield” and “Recreational Benefit” objectives plotted on X and Y axes, respectively. In Figure 4, the user is presented with seven discrete solutions showing various levels of trade-off between the two objectives, i.e., Yield in the range [0; 60] and Recreational Benefit in the range [166; 226]. For example, the highlighted point (red square) shows a compromise solution with Yield = 40 and Recreational Benefit = 200. It is unsurprising that the identified solutions correspond to the seven feasible discrete supply levels in the range of [0,6]. It is also interesting to note at this point that the same solutions could have been obtained by using a single objective GA, but that would have required at least seven independent optimisation runs, each constrained by the discrete level of yield required, i.e., 0, 1, ..., 6, whereas the solutions in Figure 4 were obtained in a single run of the EMO algorithm. By clicking on each of the solutions the value of the two objectives is displayed and the model in the Excel worksheet is automatically updated to show the corresponding reservoir storage levels and other variables for that particular solution over the entire time period considered.

The next run was performed by simply changing the type of the decision variable from integer to continuous. The resulting Pareto-optimal front is shown in Figure 6, which demonstrates that the discrete solutions identified in the previous run are the subset of the new Pareto front. It is interesting to note here that unlike in the case of discrete decision variables (Figure 4), these solutions cannot be easily obtained by running a single objective GA many times for different levels of Yield. The other benefit of visualising the entire Pareto-optimal set is that it is now more apparent that there is a considerable change in the slope of the trade-off curve beyond the point Yield = 40, thus resulting in a sharper decline in Recreational

Benefit for Yields > 40 when compared to its slope for Yields up to that level. This happened because of the maximum storage constraint, which limits the increase in recreational benefit that can be achieved. This is potentially a valuable piece of information for a decision maker needing to select one of the solutions from the Pareto set for implementation.

**(Figure 6 approximately here)**

Although this is admittedly a simple example, the application demonstrates how easy is to set up a model, optimise it using an EMO algorithm, and visualise solutions in the decision (Figure 5) and objective (e.g., Figure 6) spaces helping the decision maker to identify relevant trade-offs.

## **5. PUMP SCHEDULING OPTIMISATION**

The cost of pumping treated water is a major part of the total operating costs of water distribution systems (Bene et al., 2009). In England and Wales, the total direct water distribution costs were reported to be £458 million in the 2008/9 financial year, £130 million of which was attributed to water distribution power consumption (OFWAT, 2009). Therefore, even small improvements in how pumps are used could result in very high cost savings if proper optimisation methods are implemented.

Pump scheduling is the process of choosing which of the available pumps within a water distribution system are to be used and for which particular periods of the day the pumps are to be operated. A significant amount of research effort has been focused on optimising pump operation schedules with the aim of minimising the marginal cost of supplying water, whilst keeping within physical and operational constraints of the system (Jowitt and Germanopoulos, 1992; Mackle et al., 1995, Savić et al., 1997; Ostfeld and Tubaltzev, 2008; López-Ibáñez et al., 2008). This includes maintaining sufficient water within the systems storage tanks to meet the required time varying consumer demands while minimising the number of pump switches in an operational cycle to avoid excessive pump maintenance costs (Lansey and Awumah, 1994). As an optimisation problem, pump scheduling is difficult to solve due to the electricity tariff varying greatly through a typical operating cycle and due to the hydraulic behaviour

being highly nonlinear (Biscos et al., 2003), causing computer modelling to be a complex, computationally demanding and time consuming process (van Zyl et al., 2004).

The network example that is used in this study was first analysed by Pasha and Lansey (2009) and consists of 37 pipes, 19 nodes, 1 tank (node 21) and 1 source (node 20) with four pumps installed in the pump station at the source, as shown by a network model in Figure 7. The tank elevation and diameter are 65.53 m (215 ft) and 12.2 m (40 ft), respectively. Due to complex, nonlinear behaviour of the system, instead of developing a hydraulic simulator in Excel, EPANET software (Rossman, 2000) is used to compute the response of the system to changing pump operating schedules. The hydraulic time step and pattern time step used in EPANET were set to one hour each, while the tank level was permitted to fluctuate between 67.67 m (222 ft) to 76.2 m (250 ft). Hourly demand factors used by Pasha and Lansey (2009) range from 0.5 to 1.7, while in this work the base demand was doubled to allow better utilisation of the pump capacities and tank storage.

**(Figure 7 approximately here)**

In the case of pump scheduling, the problem can be posed as a two-objective optimisation problem with objectives being the minimisation of energy costs and the minimisation of pump switches:

$$\text{Minimise } TC = \sum_{t=1}^{24} C(t)E(Q_p(t), H(t), H(t-1)) \quad \text{Eq. 6}$$

$$\text{Minimise } TN_{ps} = \sum_{t=1}^{24} N_{ps}(t) \quad \text{Eq. 7}$$

where:

- $TC$  – total energy cost for a 24-hour period,
- $TN_{ps}$  – total number of pump switches in a 24-hour period,
- $t$  – hourly time step,
- $C(t)$  – unit cost energy during time step  $t$ ,
- $E(t)$  – energy consumed during time step  $t$ , which is as a function of pump flow,  $Q_p(t)$  and tank water level,  $H(t)$ ,

- $N_{ps}(t)$  – number of pump switches during time step  $t$ .

In this case, the total number of pump switches is always even as a pump has to be switched on and off at least once during the 24-hour period, unless it is always on or always off, in which case  $TN_{ps}=0$ . As there are four fixed speed pumps in the system, each of which can be run during any time interval, the total number of possible pump combinations is  $2^4 = 16$  during each hour of the day, or the total search space of  $7.92 \times 10^{28}$ . The main constraint is that the water level in the tank has to be kept between the allowable minimum,  $H_{\min}$ , and maximum,  $H_{\max}$ , storage levels and is a function of the previous hour tank water level,  $H(t-1)$ , and the pump station flow,  $Q_p(t)$  of that hour:

$$H_{\min} \leq H(t) \leq H_{\max} \quad \text{Eq. 8}$$

There is also a constraint on the flow each combination of pumps can deliver, which is a function of the pump characteristics and the tank water level:

$$0 \leq Q_p(t) \leq Q_{p,\max}(t) \quad \text{Eq. 9}$$

The final constraint ensures that the initial water level is reached or exceeded in the tank at the end of the optimisation period:

$$H(t = 24) \geq H(t = 0) \quad \text{Eq. 10}$$

The model is implemented as an Excel spreadsheet with EPANET initialisation/termination buttons (Figure 8-1), charts visualising key variables, i.e., tank levels, demands, pumps running (decision variables), energy tariff and consumption (Figure 8-2), and tabular representation of minimum and maximum tank levels (Figure 8-3), decision variables (Figure 8-4), model outputs (Figure 8-5), energy consumption (Figure 8-6), energy cost (Figure 8-7), and objective and penalty term values (Figure 8-8).

**(Figure 8 approximately here)**

The key point to note here is that GANetXL provides an easy way to connect the model spreadsheet (via a VBA code) to external simulators (such as EPANET). For example, in order to link EPANET and Excel spreadsheet the two VBA modules need to be developed: (1) EPANET\_DLL, which is imported simply from the EPANET programmers Toolkit (Rossman, 2000) and (2) EPANET\_Interface, which uses a set

of Toolkit functions to open the EPANET input file, read and modify pump schedules, run extended-period simulations and save the results back into the Excel spreadsheet for further analysis. An example with VBA code of how EPANET is linked to Excel for GANetXL optimisation is given in the User Manual (CWS, 2010). A model-based DSS run requiring 30,000 fitness evaluations (i.e., EPANET extended-period simulations over a 24-hour period with one hour time step), takes about 5 minutes on a Windows Vista computer with the Intel Core Quad CPU running at 2.40 GHz.

The results of the optimisation model run are shown in Figure 9. The four sets of trade-off solutions shown correspond to a single run of the multiobjective GA with a population of 150 solutions. The four sets of solutions consist of the best approximation of the Pareto set obtained after a total of 2,000 generations (30,000 solution evaluations) together with a snapshot of intermediate solutions at generations 50, 100 and 200. By running a single objective optimisation with a fixed number of pump switches the points on the best Pareto set have been confirmed.

**(Figure 9 approximately here)**

It can be observed that the Pareto-optimal sets identified pump schedules within the range of costs  $TC = [\$1,040; \$3,933]$  and pump switches,  $TN_{ps} = [0; 18]$ . The true Pareto-optimal set (i.e., for 30,000 EPANET fitness evaluation calls) achieved slightly better costs than the solution that allowed only 200 generations (i.e., 3,000 EPANET calls), but the shorter run does not provide the full coverage of the Pareto front, i.e., non-dominated solutions for 16 and 18 pump switches are missing. Furthermore, as the objective of selecting and comparing solutions would be to find a lower cost solution while keeping within an acceptably low number of pump switches, the visualisation of the Pareto set could indicate to the user that solutions with higher number of switches, e.g.,  $TN_{ps} > 6$ , provide marginal returns in terms of cost reduction.

By clicking on any of the solutions on the graphical interface showing the Pareto-optimal front, the user is able to analyse the behaviour of the system in the Excel model (e.g., the full pump schedule and tank levels over the 24-hour period, item 2 in Figure 8). For example, two pump schedules, A ( $TC=\$1,421$ ;  $TN_{ps}=6$ ) and B( $TC=\$1,263$ . $TN_{ps}=8$ ) both identify Pump1 as running continuously over 24 hours, with

Pump2 also running continuously in Schedule A and for 20 hours in Schedule B (which accounts for the additional pump switching). Pump3 runs for 8 hours in both schedules, while additionally Schedule A requires Pump3 to be switched on for an hour (at time step 20) as opposed to Schedule B, which requires Pump4 to be switched on during the same time step as in Schedule A. It is also interesting to note that the number of fitness function evaluations, e.g., ranging from 7,500 to 30,000, still represents only a minute proportion of the total search space ( $7.92 \times 10^{28}$ ).

The two case studies illustrate how GANetXL provided an easy to use framework that allows robust prototype applications to be constructed. Furthermore, the framework was easily customised to present specific solutions visually using readily available charting options within Excel.

## **6. CONCLUSIONS**

This study presents a general-purpose DSS generator, GANetXL, for developing specific DSS that require multiobjective optimisation of spreadsheet-based models. The DSS generator is a result of the desire to allow easier involvement of users in the development of DSS prototypes, which will lead to better DSS performance, acceptance and adoption in practice. It is argued here that this is made possible by using GANetXL since a large number of users (decision makers and analysts), many of whom may be inexperienced in developing and using complex optimisation methods, are able to develop or at least understand spreadsheet models. Spreadsheets, such as Microsoft Excel, are considered ideal for small to medium-size modelling applications due to their ubiquitous presence, transparency of spreadsheet-based models, ease of use, availability of ready-to-use graphical interfaces and the fact that they can be programmed by using macro language.

The DSS generator, GANetXL, which integrates evolutionary multiobjective optimisation (EMO) with spreadsheet modelling capabilities, was demonstrated on two examples from water engineering practice. The first example involves the optimisation of a simple water supply reservoir operation model with two objectives. The functionality of the DSS generator and its intuitive interface allowed rapid

implementation of the water supply reservoir simulation model within Excel, execution of the EMO algorithm and interactive visualisation of the objective and decision spaces thereby allowing a user to gain invaluable insights into the trade-offs between the two objectives. The second example deals with a much larger combinatorial optimisation problem of pump scheduling in water distribution systems and demonstrates how sophisticated Excel models (with simulation capabilities provided via an external simulation package, EPANET) can be optimised using the specific DSS developed. The two example applications demonstrated that GANetXL can be used to generate specific DSSs that are capable of effectively employing evolutionary algorithms to tackle multiobjective problems in water engineering practice and that they are intuitive enough to be comfortably used by non-expert optimisation users. It is argued here that such a tool goes a long way toward closing the gap between the achievements in optimisation technology and the successful use of DSS in practice.

The DSS generator has been released free of charge for non-commercial applications in order to support the adoption of EMO DSS among potential users. GANetXL can be downloaded from <http://www.exeter.ac.uk/cws/ganetxl> and new users can request a license through an online registration form. The example spreadsheets used in this paper have been made available at: <http://www.exeter.ac.uk/cws/ganetxl-envsoft-examples>. Further enhancements to the tool are currently being investigated, including a simple rule-based check to ensure that the GA parameters are within sensible limits, speed improvements through multi-processing technology (to enable multiple GA fitness evaluations to be run concurrently), improvements in objective space visualisation such as provided by VIDEO (Kollat and Reed, 2007) and improvements in the tool's interactive features.

## **ACKNOWLEDGEMENTS**

The authors of this work were partially supported by the *Platform Grant* (GR/T26054/01) awarded by the UK Engineering and Physical Sciences Research Council (EPSRC), which is gratefully acknowledged.

## REFERENCES

- Abbott, M.B., 1991. *Hydroinformatics: information technology and the aquatic environment*, Avebury.
- Balter, A.M., Fontane, D.G., 2008. Use of multi-objective particle swarm optimization in water resources management. *J Water Resour Plan Manage.* 134(3), 257–265.
- Bicik, J., Morley, M.S. and Savić, D.A., 2008. A rapid optimization prototyping tool for spreadsheet-based models, in: Van Zyl, J.E., Ilemobade, A.A., Jacobs, H.E. (Eds.), *Proceedings of the 10th Annual Water Distribution Systems Analysis Conference*, August 17-20, Kruger National Park, South Africa, 472-482.
- Bene, J.G., Selek, I. and Hos, C., 2010. Neutral search technique for short-term pump schedule optimization. *Journal of Water Resources Planning and Management*, 136(1), 133-137.
- Berlekamp, J., Lautenbach, S., Graf, N., Reimer, S., Matthies, M., 2007. Integration of MONERIS and GREAT-ER in the decision-support system for the Elbe river basin. *Environ Model Softw*, 22(2), 239-247.
- Biscos, C., Mulholland, M., Le Lann, M.-V., Buckley, C.A., Brouckaert, C.J., 2003. Optimal operation of water distribution networks by predictive control using MINLP. *Water SA* 29 (4), 393–403.
- Borowski, I. and Hare, M., 2007. Exploring the gap between water managers and researchers: difficulties of model-based tools to support practical water management. *Water Resour Manage*, 21,1049-1074.
- Choi, J-W., Engel, B.A. and Farnsworth, R.L., 2005. Web-based GIS and spatial decision support system for watershed management. *Journal of Hydroinformatics*, 7(3), 165-174.
- CWS, 2010. *GANetXL User Manual*. [http://www.ex.ac.uk/cws/downloads/doc\\_download/26-user-manual](http://www.ex.ac.uk/cws/downloads/doc_download/26-user-manual) (last accessed 19 August 2010).
- Čistý, M. and Bajtek, Z., 2009. Hybrid Method for Least-Cost Design of The Water Distribution Systems. *J. Hydrol. Hydromech.*, 57(2), 130–141 (in Slovak).
- Deb, K., Pratap, A, Agarwal, S., and Meyarivan, T., 2002. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transaction on Evolutionary Computation*, 6(2), 181-197.

- de Condappa D., Chaponnière, A., Andah, W., and Lemoalle, J., 2008. Decision-Tool for Water Allocation in the Volta Basin. International Forum for Water and Food, Addis Ababa, November, available online at [http://www.ifwf2.org/addons/download\\_presentation.php?fid=1135](http://www.ifwf2.org/addons/download_presentation.php?fid=1135) (last accessed 03 Aug 2010).
- de Kok, J-L., Kofalk, S., Berlekamp, J., Hahn, B., and Wind, H., 2009. From Design to Application of a Decision-support System for Integrated River-basin Management. *Water Resources Management*, 23(9), 1781-1811.
- Deepthi, N., Suja, R. and Letha, J., 2009. Multi-Objective Reliability Based Design of Water Distribution System, 10th National Conference on Technological Trends (NCTT09), 6-7 November 2009, available online at: <http://kbase.cet.ac.in:8180/jspui/handle/123456789/755> (last accessed 25 Aug 2010).
- Eom, H.B., and Lee, S.M., 1990. A Survey of Decision Support system Applications (1971-1998). *Interfaces*, 20(3), 65-79.
- Farmani, R., Savić, D.A. and Walters, G.A., 2005. Evolutionary multi-objective optimization in water distribution network design. *Eng Optim*, 37(2), 167-83.
- Farmani, R., Henriksen, H.J. and Savić, D.A., 2008. An evolutionary Bayesian belief network methodology for optimum management of groundwater contamination, *Environ. Model. & Softw.*, 24(3), 303-310.
- Fonseca, C.M. and Fleming, P.J., 1995. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1), 1-16.
- Giupponi, C., 2007. Decision-support systems for implementing the European water framework directive: the MULINO approach. *Environ. Model. & Softw.*, 22(2), 248–258.
- Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA.
- Haines, Y., 1998. *Risk Modeling, Assessment, and Management*. John Wiley & Sons, Inc., New York, NY.

- Halhal, D., Walters, G.A., Ouazar, D. and Savić, D.A., 1997. Water network rehabilitation with structured messy genetic algorithm. *J. Water Resour. Plann. Manage.*, 123(3), 137-46.
- Hadihardaja, I.K., Latief, H. and Mulia, I.E., 2010. Decision support system for predicting tsunami characteristics along coastline areas based on database modelling development. *J. of Hydroinformatics*, doi:10.2166/hydro.2010.001, available online at <http://www.iwaponline.com/jh/up/hydro2010001.htm> (last accessed 03 Aug 2010).
- Jowitt, P.W. and Germanopoulos, G., 1992. Optimal pump scheduling in water supply networks. *J. Water Resour. Plann. Manage.*, 118(4), 406–422.
- Kollat, J.B. and Reed, P., 2007. A framework for Visually Interactive Decision-making and Design using Evolutionary Multi-objective Optimization (VIDEO). *Environ. Model. & Softw.*, 22, 1691-704.
- Lansey, K.E., and Awumah, K., 1994. Optimal pump operations considering pump switches. *J. Water Resour. Plann. Manage.*, 120(1), 17-35.
- Lasdon, L.S., Waren, A.D., Jain, A., and Ratner, M., 1978. Design and testing of a generalized reduced gradient code for nonlinear programming. *ACM Transactions on Mathematical Software*, 4(1), 34-49.
- López-Ibáñez, M., Prasad, T.D. and Paechter, B., 2008. Ant Colony Optimisation for the Optimal Control of Pumps in Water Distribution Networks. *J. Water Resour. Plann. Manage.*, 134(4), 337-346.
- Loucks, D.P., 1995. Developing and implementing decision support systems: a critique and a challenge. *Water Resources Bulletin*, 31(4), 571–582.
- Mackle, G., Savić, D.A. and Walters, G.A., 1995. Application of Genetic Algorithms to Pump Scheduling for Water Supply. *GALESIA'95*, London, 1995.
- Molina, J.L., Bromley, J., García-Aróstegui, J.L., Sullivan, C. and Benavente, J., 2010. Integrated water resources management of overexploited hydrogeological systems using Object-Oriented Bayesian Networks: A case study of the Murcia's Altiplano SE Spain water system. *Environ. Model. & Softw.*, 25(4), 383-397.

- Morley, M.S., Atkinson, R.M., Savić, D.A. and Walters, G.A., 2001. GAnet: genetic algorithm platform for pipe network optimisation. *Advances in Engineering Software*, 32(6), 467-475.
- Nafi, A. and Kleiner, Y., 2010. Scheduling Renewal of Water Pipes While Considering Adjacency of Infrastructure Works and Economies of Scale, *J. Water Resour. Plann. Manage.*, 136(5), 519-530.
- Nicklow, J., Reed, P., Savić, D.A., Dessalegne, T., Harrell, L., Chan-Hilton, A., Karamouz, M., Minsker, B., Ostfeld, A., Singh, A. and Zechman, E., 2010. State of the Art for Genetic Algorithms and Beyond in Water Resources Planning and Management. *J. Water Resour. Plann. Manage.*, 136(4), 412-432.
- OFWAT, 2009. Financial performance and expenditure of the water companies in England and Wales 2008-09. available online: [http://www.ofwat.gov.uk/regulating/rpt\\_fpr\\_2008-09.pdf](http://www.ofwat.gov.uk/regulating/rpt_fpr_2008-09.pdf) (last accessed 26 Aug 2010).
- Ostfeld, A. and Tubaltzev, A. (2008) Ant Colony Optimization for Least-Cost Design and Operation of Pumping Water Distribution Systems, *J. Water Resour. Plann. Manage., ASCE*, 134(2), 107-118.
- Pasha, M.F.K., and Lansey, K., 2009. Optimal Pump Scheduling by Linear Programming. *World Environmental and Water Resources Congress 2009*, Kansas City, Missouri, 395-404.
- Power, D.J. and Sharda, R., 2007. Model-driven decision support systems: Concepts and research directions. *Decision Support Systems*, 43, 1044-1061.
- Reed, P., Minsker, B.S. and Goldberg, D.E., 2001. A multiobjective approach to cost effective long-term groundwater monitoring using an elitist nondominated sorted genetic algorithm with historical data. *J. Hydroinform*, 2001, 3(2), 71-90.
- Ritzel, B.J., Eheart, J.W. and Ranjithan, S.R., 1994 .Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. *J. Water Resour. Res.*, 30(5), 1589-603.
- Rossman, L.A., 2000. EPANET 2 USERS MANUAL. United States Environmental Protection Agency, Water Supply and Water Resources Division, National Risk Management Research Laboratory, Cincinnati, OH, 45268.

- Rozos, E., Makropoulos, C. and Butler, D., 2010. Design Robustness of Local Water-Recycling Schemes. *J. Water Resour. Plng. and Mgmt.*, 136(5), 531-538.
- Savenije, H.H.G., 1995, Spreadsheets: flexible tools for integrated management of water resources in river basins. In: *Modelling and management of sustainable basin-scale water resources systems*. IAHS Publications no. 231, 207-215.
- Savić, D.A., Walters, G.A., and Schwab, M., 1997. Multiobjective genetic algorithms for pump scheduling in water supply. *Lect. Notes Comput. Sci.*, 1305, 227–236.
- Schreyer, A. C., 2006. GA Optimization for Excel Version 1.2. Quick Start Manual, available from: <http://www.alexschreyer.net/projects/xloptim/> (last accessed on 10 August 2010).
- Scott Morton, M.S., 1971. *Management Support Systems, Computer-Based Support for Decision Making*. Division of Research, Harvard University, Cambridge, Massachusetts.
- Sprague, R.H., 1980. A Framework for the Development of Decision Support Systems. *MIS Quarterly*, 4(4), 1-26.
- Van Delden, H., 2009. Lessons learnt in the development, implementation and use of Integrated Spatial Decision Support Systems, 18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009.
- Van Zyl, J.E., Savić, D.A. and Walters, G.A., 2004. Operational Optimization of Water Distribution Systems Using a Hybrid Genetic Algorithm. *J. Water Resour. Plann. Manage.*, 130(2), 160-170.

Figure captions:

Figure 1. GANetXL components.

Figure 2 Selection and configuration of the GA chromosome (decision variables).

Figure 3 Optimisation run progress form.

Figure 4 An example of Pareto-optimal front.

Figure 5 Reservoir operation model.

Figure 6 Reservoir optimisation Pareto front (continuous decision variables).

Figure 7 Example water distribution network (Pasha and Lansey, 2009).

Figure 8 Pump scheduling model implemented in Excel.

Figure 9 Results of the pump scheduling optimisation.

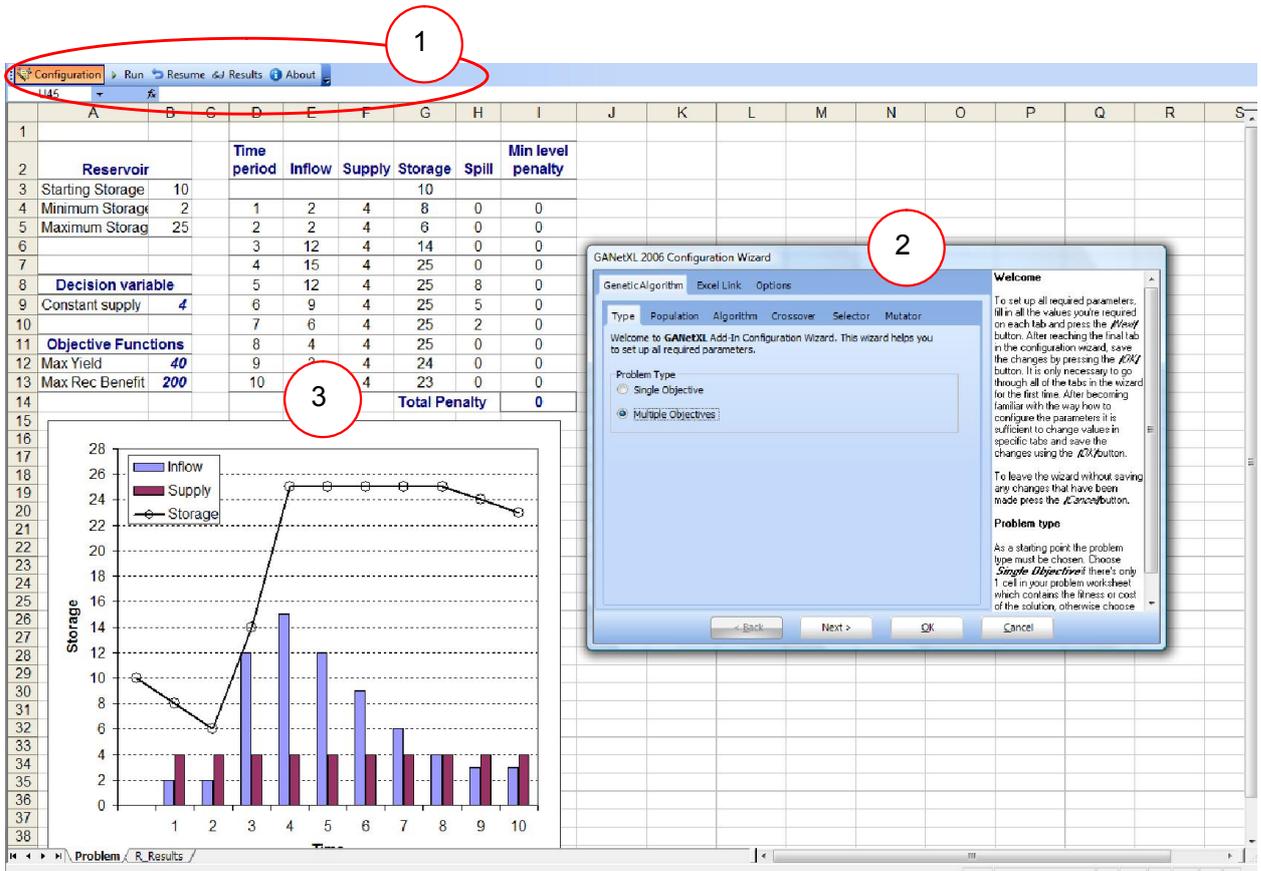


Figure 1. GANetXL components 1-3

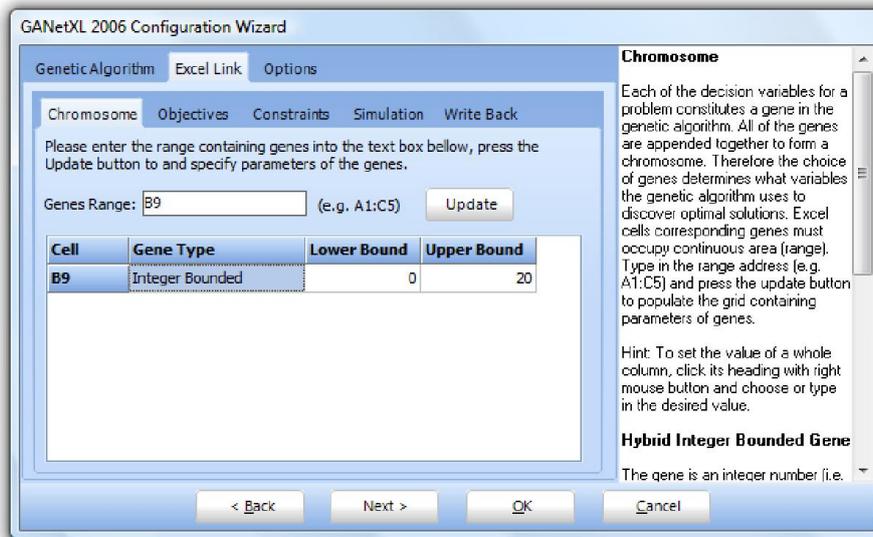


Figure 2 Selection and configuration of the GA chromosome (decision variables).

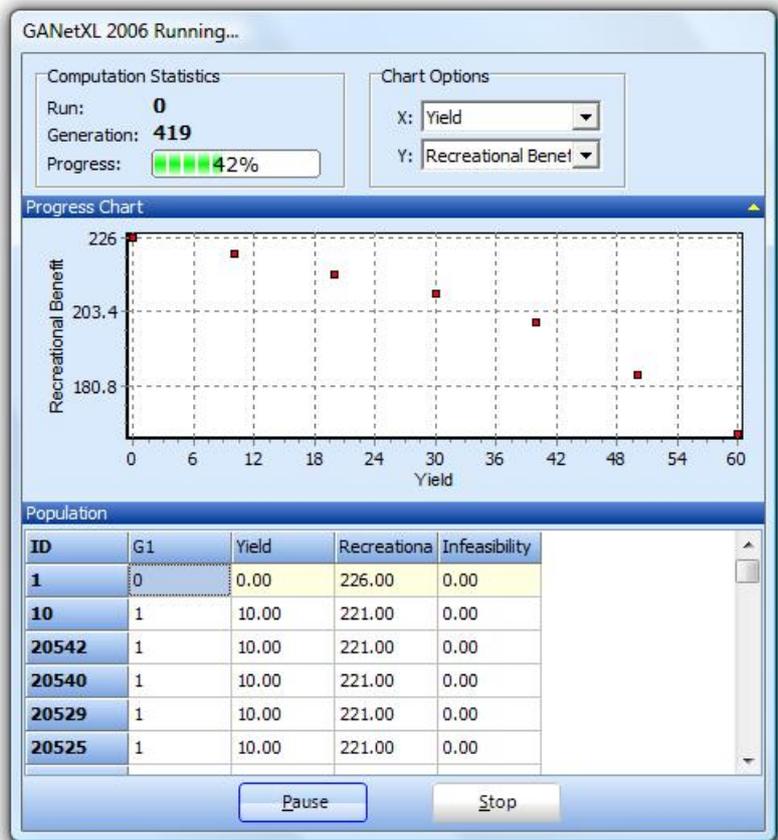


Figure 3 Optimisation run progress form

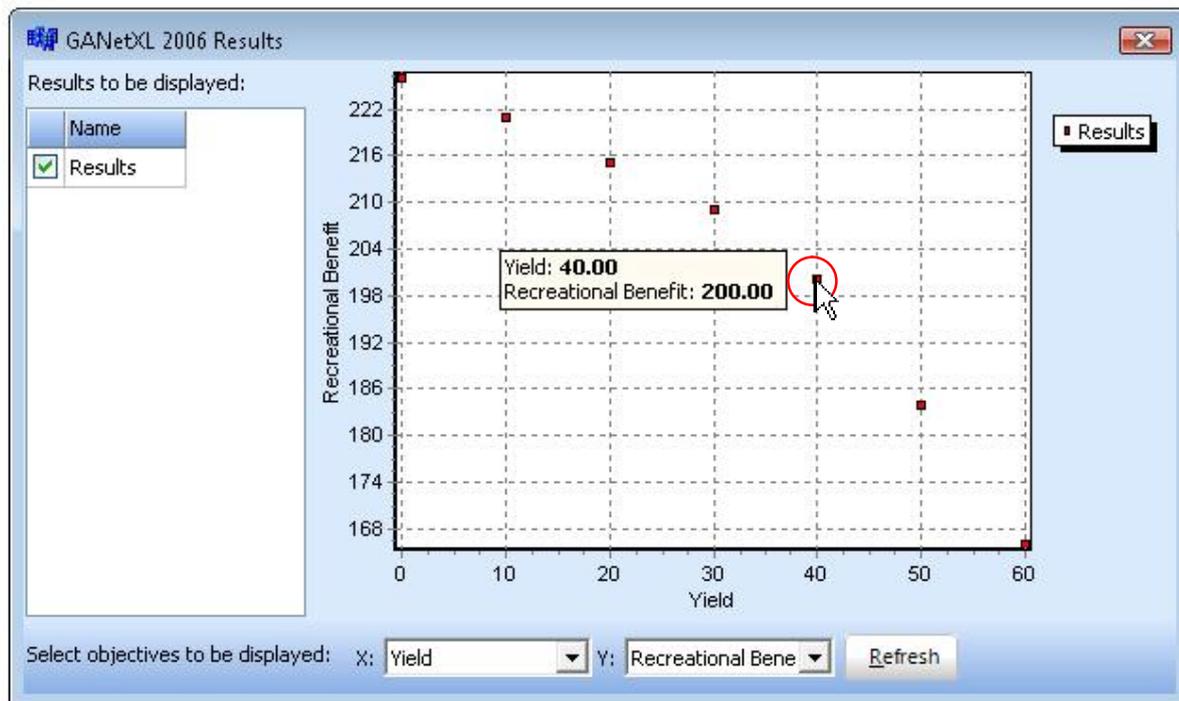


Figure 4 An example of Pareto-optimal front shown on the interactive visualisation interface

|    | A                          | B          | C | D                  | E             | F             | G                    | H            | I              |
|----|----------------------------|------------|---|--------------------|---------------|---------------|----------------------|--------------|----------------|
| 1  |                            |            |   |                    |               |               |                      |              |                |
| 2  | <b>Reservoir</b>           |            |   | <b>Time period</b> | <b>Inflow</b> | <b>Supply</b> | <b>Storage</b>       | <b>Spill</b> | <b>Deficit</b> |
| 3  | Starting Storage           | 10         |   |                    |               |               | 10                   |              |                |
| 4  | Dead Storage               | 2          |   | 1                  | 2             | 7             | 5                    | 0            | 0              |
| 5  | Reservoir Capacity         | 25         |   | 2                  | 2             | 7             | 0                    | 0            | 2              |
| 6  |                            |            |   | 3                  | 12            | 7             | 5                    | 0            | 0              |
| 7  |                            |            |   | 4                  | 15            | 7             | 13                   | 0            | 0              |
| 8  | <b>Decision variable</b>   |            |   | 5                  | 12            | 7             | 18                   | 0            | 0              |
| 9  | Constant supply            | <b>7</b>   |   | 6                  | 9             | 7             | 20                   | 0            | 0              |
| 10 |                            |            |   | 7                  | 6             | 7             | 19                   | 0            | 0              |
| 11 | <b>Objective Functions</b> |            |   | 8                  | 4             | 7             | 16                   | 0            | 0              |
| 12 | Max Yield                  | <b>70</b>  |   | 9                  | 3             | 7             | 12                   | 0            | 0              |
| 13 | Max Rec Benefit            | <b>116</b> |   | 10                 | 3             | 7             | 8                    | 0            | 2              |
| 14 |                            |            |   |                    |               |               | <b>Total Penalty</b> |              | <b>4</b>       |

Figure 5 Reservoir operation model



Figure 6 Reservoir optimisation Pareto front (continuous decision variables).

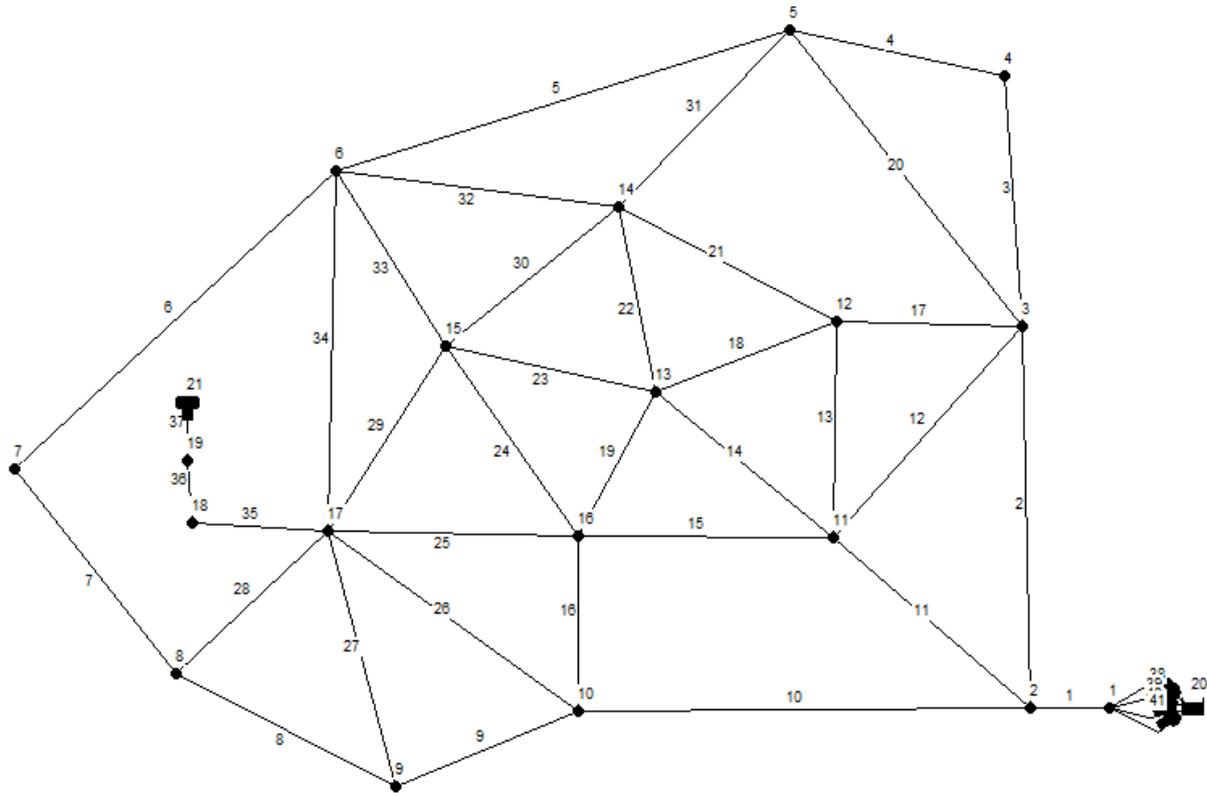


Figure 7 Example water distribution network (Pasha and Lansey, 2009).



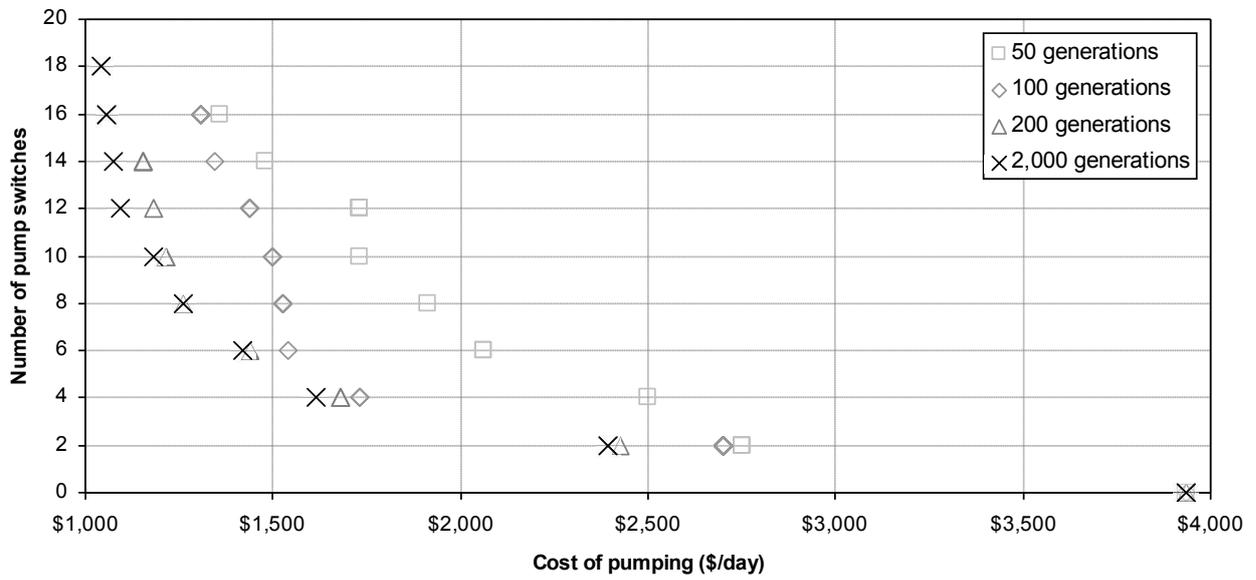


Figure 9 Results of the pump scheduling optimisation.