

# Application of genetic algorithms - Determination of the optimal pipe diameters

SJ van Vuuren

Department of Civil and Biosystem Engineering, University of Pretoria, Pretoria 0001, South Africa

## Abstract

The past decade has marked the development of computational capacity that far exceeds the capacity of the "instructor" to define options to be evaluated when optimisation has to be achieved. In the context of water supply, there are numerous variables that can influence the selection and, hence, the final cost of system implementation, improvement and operation. This article reflects some of the findings of a research project, funded by the Water Research Commission of South Africa to establish the potential use of GAs in the water industry in South Africa. A utility program (GAPOP) was developed to demonstrate the application of GAs in the determination of the optimal pipe diameter.

## Introduction

The past decade has marked the development of computational capacity that far exceeds the capacity of the "instructor" to define options to be evaluated when optimisation has to be achieved.

The Government's objective to provide "water for all" made it essential for the limited capital to be employed to provide the maximum benefit. The optimal investment decision for water infrastructure implementation, expansion, addition or rehabilitation has to be reviewed against the background of the conflicting demand for capital and the backlog of water service provision. This complicates the decision to select the cost-effective and efficient solution.

The determination of the optimal selection of system components requires techniques that can be employed to assist the decision-maker in finding the appropriate solution within the environment of all the possible solutions (solution space).

Genetic algorithms (GAs) have been developed (Holland, 1992) to assist in searching through complex solution spaces for the optimum solution. GAs have been applied as search techniques for various engineering problems such as structural design optimisation, water distribution network evaluation, pump scheduling, hydrological runoff predictions and resource utilisation. This technique has not been generally used in South Africa.

Within the context of water supply, there are numerous variables that can influence the selection and the final cost of system improvements. The high variance in rainfall and runoff, availability of alternative water supply with different reliabilities, demand pattern variability, operational complexity of the system, maintenance requirements, running cost (especially power cost), affordability and willingness to pay for services, will influence the decision on whether the water supply scheme should be implemented, refurbished, replaced, discarded or expanded.

Identification and analyses of the system components, which are required to optimise the solution, will have to employ techniques that can assist in finding the optimal solution. GAs have been developed as an optimisation search technique.

Based on the functioning of DNA in nature to produce a gene population with specified characteristics, a mathematical cloning of this process has been defined to produce outcomes with specific characteristics. If the objective outcome can be defined, be it the minimum cost solution or any other objective, the genetic algorithm process will "calculate" which gene pool will best approach the objective function (Goldberg, 1989).

The technique of GAs has been applied on a number of different real problems and has resulted in exciting, but not always straightforward solutions.

In complex water distribution systems, for instance, the alternative options when evaluating the extensions to water supply systems become numerous. GAs provide procedures for the evaluation of the optimal solutions in the solution space.

## The need for optimisation in the water sector

It has been indicated that the challenges in the water industry in South Africa and the world at large, together with the capital constraints and operational cost escalation, necessitate the evaluation of technical, economical and environmental parameters to reach an optimal solution.

In the water sector it has been indicated that large savings can be accomplished if optimal solutions are implemented (Walters et al., 1999), when new systems are designed or when existing systems are refurbished or extended.

The need for the application of optimisation techniques stems from the fact that:

- the selection of system components to be evaluated in a water system is dependent on a number of inter-dependent variables. For example, if an optimal diameter has to be determined it is known that by reducing the diameter the capital cost is reduced but the operating cost (pumping) will escalate and the possibility of pipe burst due to surge pressures associated with high-flow velocities will increase.
- a number of uncertainties exist when the optimisation of water infrastructure is considered, such as:
  - what would the influence of escalation be on operational and capital replacement cost;

---

☎ 012 420 2438; fax: 012 362 5218; e-mail: fvuuren@eng.up.ac.za  
Received 24 May 2001; accepted in revised form 14 January 2002.

- how would natural phenomena like floods and droughts influence the choice of the system components;
- how would the affordability of services to the consumer influence the design standards; and
- how would demand variance and increase influence the selection of the capacity of system components.

Planners, designers and operators are involved in the assessment of the following issues where GAs can be employed with great advantage:

- The identification of pipe segments in a distribution network that should be rehabilitated to improve the performance of the system.
- Determining the application of a phased development of infrastructure for different development horizons.
- Cost-effective development of infrastructure for alternative service levels ensuring an affordable service.
- Optimisation of reservoir sizes and determination of the required pump capacities.
- Optimisation of operational scheduling.

### Genetic algorithms (GAs) – An overview

An algorithm is any procedure that uses data and modifies it according to a set of instructions. Every structured calculation procedure is an example of an algorithm. GAs are programs that simulate the logic of Darwinian selection. If one appreciates that populations accumulate differences over time, due to the environmental conditions acting as a selective breeding mechanism, then you understand GAs. Putting it another way: Understanding a GA means understanding the simple, iterative process that underpins evolutionary change. The issue, of course, is how best to get that “selection process” translated into a program procedure and applied to your problem.

Holland (1992) the pioneer of GAs defines the evolutionary nature of the algorithm as follows:

- Start with an initial population that is randomly generated, but contains the parameter characteristic variability of the population.
- The fitness of each individual in the population is assessed according to a fitness function.
- The probability of each individual to survive is proportional to its fitness.
- The individuals of the next generation are selected on probability and through a genetic transformation process of cross-over and mutation, ensuring that the solution is not localised within the solution environment.

GAs are suited to solve problems that are not vulnerable to attack by enumerative methods because the sheer number of potential solutions defies the possibility of testing them all. Such problems are typically multi-constrained, that is the solution must be a balance of conflicting or synergistic properties. When considering a problem with multiple dependencies you are normally forced to admit the possibility of isomeric solutions, i.e. solutions that give the same result using different processing routes. So for some problems there is no such thing as the “best solution”, but instead one looks for members of a fuzzy set of solutions that can be defined as “good enough”.

Some problems have a “best” solution, but can be lost in a vast result space of complex problems. If the solution space is limited

then enumerative techniques can work. This is why machines can now beat chess masters. The problem with enumerative techniques is their almost complete lack of scalability. If you manage to build a computer large enough to solve a reasonably constrained problem, then adding just one more constraint will probably require another 10 similar computers to deal with additional calculations. Add another row of squares to a chessboard and the number of possible moves expands dramatically. One of the great strengths of GAs is that they do not have to evaluate all the possible solutions. This means that increasing the number of possible solutions has little impact on the running time of a GA.

Goldberg (1989) indicated that a GA differs from the traditional search methods in the following ways:

- GAs work with coding of the parameter set, not the parameters themselves.
- GAs evaluate a population of points, not a single point.
- GAs use objective function (payoff) information, not derivatives or other auxiliary knowledge, to determine the fitness of the solution.
- GAs use probabilistic transition rules, not deterministic rules in the generation of the new populations.

GAs use bit-strings to represent the state and characteristics of an object model. Changing the values of the bits in these bit-strings, can be translated back into changes to the associated objects’ data. Once the object has been converted (coded) into bit-strings, the GA-program (coder) can apply biologically analogous processes such as replication (or reproduction), cross-over and mutation to the bit-strings, which can then be translated back to the objects themselves. In this way the GA-coder can evolve the instance-state of the components within an object model to obtain a bit-string (solution) with a high fitness. Changes to the bit-string values can be accomplished through the process of reproduction, cross-over and mutation that will be discussed in more detail later.

Now that it is known that a GA is a search procedure, based on the mechanics of natural selection and natural genetics which is geared to handle complex multi-objective problems with a large solution space, a brief overview of the functioning of a GA will be provided through the discussion of an example.

### Example of a genetic algorithm

Goldberg (1989) provides the following examples, which have been adapted in this paper, to illustrate the working of a genetic algorithm. The objective is to maximise function  $f(x) = x^2$ , where  $x$  is permitted to vary between 0 and 31.

To use a genetic algorithm for the optimisation of this problem, the decision variables of the problem must first be coded in some finite-length string. In this case, the variable  $x$  will simply be coded as a binary unsigned integer of length 5. Before proceeding with the simulation, briefly review the notion of a binary integer. In Base 2 arithmetic, there are only two digits to work with, 0 and 1, and as an example for the number 10 the decoding will result in a string 01010.

$$0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 0 + 8 + 0 + 2 + 0 = 10.$$

Another example is given below for the coding of the value of 19 in Base 2 arithmetic (10011).

$$1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 16 + 0 + 0 + 2 + 1 = 19.$$

With a five-bit (**binary digit**) unsigned integer it is possible to obtain

numbers between 0 (00000) and 31 (11111). With a well-defined objective function and coding, a simulation of a single generation of a genetic algorithm with reproduction, cross-over, and mutation is conducted.

To start off, select an initial population at random with a population size 4 (4 strings or members in the population) by tossing a fair coin 20 times to obtain the bit value of the 4 times 5 = 20 digits in the four chromosome (bit) strings. Looking at this population, shown in Column 2 of Table 1, it can be observed that the decoded  $x$  values are presented along with the fitness or objective function values  $f(x)$ . To make sure that it is known how the fitness values  $f(x)$  are calculated from the string representation, take a look at the third string of the initial population, string 01000 (Column 2 of Table 1). Decoding this string as an unsigned binary integer, note that there is a single one in the  $2^3$ 's position. Hence for string 01000 the value of  $x = 0 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 8$  (Column 2 of Table 1). To calculate the fitness or objective function, simply square the  $x$  value and obtain the resulting fitness value  $f(x) = 64$  (Column 6 of Table 1). Other  $x$  and  $f(x)$  values may be obtained similarly.

A new generation (population) (Column 10 of Table 1) of the genetic algorithm begins with reproduction of new chromosomes from the mating pool. The strings that will be available in the next generation are selected on the basis of their fitness (Column 6 and 7 of Table 1). In this example one of string 1, two of string 2 and one of string 4 will be transferred to the mating pool, as shown in Column 7 of Table 1. This reflects that the "best" string gets more copies, the average string stays even (survives), and the worst dies off.

With an active pool of strings looking for mates, simple cross-over proceeds in two steps:

- strings are mated randomly, using a random process to pair off the happy couples; and
- mated string couples' cross-over position is also determined randomly.

Referring again to Table 1, random choice of mates has selected the second string in the mating pool to be mated with the first (Column 9 of Table 1). With a crossing site of 4, the two strings 01101 and 11000 cross and yield two new strings 01100 and 11001 (Column 11 of Table 1). The remaining two strings in the mating pool are crossed at Site 3; the resulting strings are shown in Column 11 of Table 1.

The last operator, mutation, is performed on a bit-by-bit basis. Assume that the probability of mutation in this example is 0.001. With 20 transferred bit positions it is expected that  $20 \times 0.001 = 0.02$  bits will undergo

TABLE 1  
A genetic algorithm simulation by hand

String No	Initial population (Randomly generated) <sup>(1)</sup>	X value	$f(x) = x^2$	$P_{select} = \frac{f_i}{\sum f}$	Expected count $\frac{f_i}{f}$	Number of strings to be transferred to the next generation	Strings available for reproduction, mating and cross-over	Pool mate (randomly selected)	Cross-over site (randomly selected)	New population	X value	$f(x) = x^2$
1	2	3	4	5	6	7	8	9	10	11	12	13
1	0 1 1 0 1	13	169	0.14	0.58	1	0 1 1 0 1	1 and 2	4	0 1 1 0 0	12	144
2	1 1 0 0 0	24	576	0.49	1.97	2	1 1 0 0 0	3 and 4	4	1 1 0 0 1	25	625
3	0 1 0 0 0	8	64	0.06	0.22	0	1 1 0 0 0		3	1 1 0 1 1	27	729
4	1 0 0 1 1	19	361	0.31	1.23	1	1 0 0 1 1		3	1 0 0 0 0	16	256
Sum $\sum f$			1170	1.00	4.00	4.0		Sum				1754
Average			293	0.25	1.00	1.0		Average				439
Max			576	0.49	1.97	2.0		Max				27

**Notes:**

- 1) Initial population chosen randomly by evaluating bit-by-bit. If the random generator provides a value greater than 0.5, then the bit will be 1. Else it will be 0.
- 2) Cross-over performed through binary decoding of 2 coin tosses.
- 3) Cross-over probability assumed to be unity  $p_c = 1.0$
- 4) Mutation probability assumed to be 0.001,  $p_m = 0.001$ , expected mutations =  $5 \times 4 \times 0.001 = 0.02\%$ . No mutations expected during a single generation. None simulated.

mutation during a given generation. Simulation of this process indicates that no bits undergo mutation for this low mutation probability. As a result, no bit positions are changed from 0 to 1 or vice versa during this generation.

Following reproduction, cross-over, and mutation, the new population is ready to be tested. To do this, simply decode the new strings created by the simple genetic algorithm and calculate the fitness function values from the  $x$  values thus decoded. The results of a single generation of the simulation are shown in Column 12 of Table 1. While drawing concrete conclusions from a single trial of a stochastic process is, at best, a risky business, it is clear to see how GAs combine high-performance solution to achieve even better performance. In Table 1, note how both the maximal and average performance has improved in the new population. The population average fitness has improved from 293 to 439 in one generation, while the maximum fitness has increased from 576 to 729 during that same period. Although random processes help cause these circumstances, it is evident that this improvement is no fluke. The best string of the first generation (11000) receives two copies because of its high, above-average performance. When this combines at random with the next highest string (10011) and is crossed at Location 3 (again at random), one of the resulting strings (11011) proves to be a very good solution indeed.

In any genetic algorithm the following functional procedures are used:

- Reproduction
  - copy according to objective function, maintaining the optimal strings for the next generation
- Cross-over
  - parent pairs exchange their gene bits creating new gene strings that contain the characteristics of the parent strings
- Mutation
  - protects against loss of useful genetic material and forces the search for the optimal solution to a different place in the solution space (secondary mechanisms)

Repetitive application of these functions of the genetic algorithm, has been used in the optimisation of a variety of complex problems.

### **Application of genetic algorithms (GAs) in water engineering**

Researchers, covering a wide range of water engineering-related topics have discussed various applications of GAs. A number of fields in the water industry, have been reviewed and the contributions of GAs can be summarised according to the following groupings:

- Runoff estimation
- Yield assessment of surface reservoirs
- Optimisation of the system components during the planning and design stage
- Operational optimisation
- Network rehabilitation
- Optimisation of operations of water purification plants and pump stations

Within the limited space available in this paper, reference is made only to some of the references on pipeline optimisation, operational optimisation and network optimisation.

Montesinos et al. (1999) investigated the New York water supply problem and found that by modifying the genetic algorithm used by previous researchers, the cheapest published solution for

this problem was obtained. The paper also highlighted the fact that there was no defined methodology for the evaluation of the penalty. A variable was defined to indicate the number of constrained failures. The variables were related directly to the cost of the selected solution. The authors also modified the mutation operation so that the mutation of a string occurs at a random position along the bits (individual elements of the gene) of the string.

Reis and Chaudhry (1997) used GAs to determine the location and setting of pressure control valves in a distribution system, the objective being to minimise leakage for certain reservoir levels.

Yeo and Agyei (1998) indicated the application of GAs on a number of engineering problems such as the finite element analyses of a structure, extraction of pollutants from an aquifer, positioning of wells for optimum abstraction and minimizing pumping rates and overall costs. The authors indicated the flexibility of GAs as an optimisation technique and its applicability in practical engineering problems.

Yagi and Shiba (1999) found that the utilization of GAs provided an effective way of establishing a reasonable fuzzy rule base for a pumping station handling combined sewage and natural flow.

Wardlaw and Sharif (1999) indicated that a four-reservoir problem could be handled effectively by utilizing GAs with real value coding with tournament selection, uniform cross-over and modified uniform mutation. The authors were the first to use uniform cross-over for this problem, which allowed each gene to cross-over. They also indicated that real value coding was significantly faster than binary coding and gave better results.

Morshed and Kaluarachchi (1998) indicated the complexities of flow and contaminant modeling of groundwater resources and reflected the usefulness of artificial neural networks in solving groundwater problems. Reference to the use of GAs as a robust mechanism in solving non-linear, optimisation problems was made.

Barros and Rutledge (1998) indicated the difficulty to establish which of the variables should be included in a multivariate regression model. The problem that arose here was the assessment of a large number of measurements taken from a set of samples with the independent variable not being mutually independent (linear dependent vectors or co-linearity). The authors showed that GAs provided a global optimal selection of variables in a reasonable time, but warned that the initial selection of strings was important to ensure convergence.

Olivera and Loucks (1997) proved in a very useful and descriptive way that GAs could be employed to optimise the operating rules of multi reservoir systems.

Huang and Mayer (1997) indicated that GAs could be used in the search for optimum pumping rates and discrete spacing of wells in an aquifer with homogeneous or non-homogeneous characteristics.

Gupta et al (1999) developed a methodology based on the use of GAs for the development of lower cost designs and augmentation of existing water distribution systems.

Halhal et al. (1997) indicated that in network rehabilitation, expansion and replacement it was difficult to determine the correct improvements to be achieved with a limited capital budget if conventional optimisation techniques (e.g. linear programming) were applied. A multi-objective function approach was applied by using capital cost and benefit as the objective function to determine the required changes to the system through the use of GAs as the search technique. The two examples that were discussed showed the value of this technique.

Walters et al. (1999) presented a procedure to assist with the decision in the rehabilitation and expansion of existing networks.

This procedure was tested against the benchmark problem (“Anytown”). In the evaluation of the benchmark problem, tank levels were defined as independent variables and all the nodes were specified as potential sites for new storage tanks. Allowance was also made for the pump sizes to vary. The solutions that were obtained by evaluating a 24 h extended period, resulted in a cheaper solution than determined before. It was found that the required storage was more than that, determined by other researchers, using other techniques.

The benefit of the optimisation has been proven and further application of GAs now remains the challenge. In the planning and design stages of any project, alternative solutions have to be compared and optimised. The optimisation problems frequently encountered are:

- The selection of pipelines in a distribution network to be incorporated for rehabilitation.
- Decision-support for abandoning or replacing pipelines.
- Determining the stages of a phased development of infrastructure for different development horizons.
- Cost-effective development of infrastructure, taking cognisance of alternative service levels and affordability of service.
- Optimisation of reservoir sizing and pump capacity.

The optimisation decision is always taken against the background of a number of uncertainties, such as :

- what will the influence of escalation be on operational and capital replacement cost;
- how will natural phenomena like floods and droughts influence the choice of the system components;
- how will affordability of services for the consumers influence the design standards; and
- how will demand variance and increase influence the selection of the capacity of system components.

In the next section the utility program that was developed to determine the optimal pipe diameter for a gravity or a pumping main will be described.

## Utility program for the determination of the optimum pipe diameter

### Introduction to the program features

The utility program can analyse two simple problems. One being a gravity system with off-takes between the supply and the end reservoir and the second is a pumping line with demands between the pump station and the end reservoir. In both these problems the objective function is to minimise the costs while the required residual pressure at all the nodes and the velocity is maintained below the maximum permissible flow velocity in the pipes. The constraints against which the solution is valued are:

- the residual pressure should be in excess of the minimum required residual pressure at all the demand nodes; and
- the flow velocity should be less or equal to the maximum allowable velocity in the system.

The designer can set these constraints.

## Short description of the program, “Genetic Algorithm Pipeline Optimisation Program” (GAPOP)

The objectives of GAPOP are to:

- provide access for the novice to the background, value and understanding of the use of GA techniques;
- determine an optimal solution for pipe diameters based on the life-cycle cost for a simple gravity or pumping system.

In the search for the optimal solution, the “fitness function” seeks the lowest cost solution. Cost comparison in the case of the pumping system is done, based on the net present value (NPV), (capital, maintenance and energy costs) calculated over the analysis period. In the case of the gravity system only the capital cost is optimised. Provision has been made for a weighting factor to penalise the solution by increasing the cost of the solution on the following two grounds:

- the residual pressure being less than the requirement; and
- the flow velocity being more than the acceptable maximum design flow velocity.

In the case of a pumping system the weighting factor, which is applied to increase the NPV of the cost of the solution (real cost), resulted in a new cost referred to as “comparative cost”. The “comparative cost” will always be higher than or equal to the real cost of the solution and is used to rank the alternatives according to the comparative cost. In the case of the gravity system, the weighting factor is applied to the capital cost of the pipeline section(s) where the specified residual pressure or the specified maximum flow velocity has been violated, to obtain the “comparative cost”.

The program can analyse up to 20 pipe sections and 8 different pipe sizes and types (1,15E18 possible solutions). Although the program requires the minimum input, it provides the user with an optimised set of solutions to be investigated in more detail. In the following paragraph the input data are discussed.

### Data input required

In the following section a brief summary of the input data is given. The input screens that reflect the required format for the data are attached as **Annexure 1**. The program has two options, a gravity system or a pumping system.

The system selection is made on the first screen:

- Gravity system or
- Pumping system

The data requirements for the gravity system and the pumping system differ and are reflected in Table 2, which reflects the details of the data that are required to optimise the pipeline diameters for the gravity and pumping pipelines. Table 3 reflects the output from the program. The data input screens and output screens are included in **Annexure 1**.

Copies of the GAPOP utility program can be obtained from the Water Research Commission’s website <http://www.wrc.org.za> or from the author.

The program uses the information to evaluate different pipe diameters, solving the objective function (lowest capital cost for a gravity system and the lowest total NPV of all the cost over the design period for the pumping system) that adheres to the constraints defined for the system. Once the system is analysed the 20 most

<b>TABLE 2</b> <b>Data input required for the pipeline optimisation program, GAPOP</b>		
<b>Data description and output details</b>	<b>Program screens containing the data for the following systems</b>	
	<b>Gravity</b>	<b>Pumping</b>
<b>Project data</b> : Project description, designer's name and date <b>General data</b> : Upstream reservoir level, upstream reservoir ground level and downstream reservoir level. Unit definition for the pipe length, elevation and demand. <b>Profile data</b> : Pipe length, end-point elevation and the demand at the end-point are required.	<b>G1</b>	<b>P1</b>
<b>Pipe material selection</b> The designer selects the pipe material, specifies the pipe diameter, pipe roughness parameter, cost of pipe per meter and selects whether the pipe should be included in the analysis. Units for the pipe diameter and roughness are selected.	<b>G2</b>	<b>P2</b>
<b>Design limitations</b> The minimum required residual pressure, maximum flow velocity and the mutation rate are defined on this screen.	<b>G3</b>	<b>P3</b>
<b>Energy details</b> Unit power consumption costs for the summer and winter season are defined as well as the daily pumping hours that should be considered in the analysis. The pump efficiency is also defined (%).		<b>P4</b>
<b>Rates</b> The cost functions for mechanical and electrical works associated with the pump station need to be defined as well as the % yearly allowance for maintenance on the pipeline (civil), mechanical and electrical works. Escalation rate for future expenses, energy escalation rate, the discount rate and the design life are defined.		<b>P5</b>
<b>Design limitations</b> The minimum required residual pressure, maximum flow velocity and the mutation rate are defined on this screen.	<b>G4</b>	<b>P6</b>

<b>TABLE 3</b> <b>Output for the pipeline optimisation program, GAPOP</b>		
<b>Output</b>	<b>G5</b>	<b>P7</b>
<b>Favourable solutions</b> The 20 best solutions are summarised – reflecting the real and total cost as well as the pipe diameters used in each of the solutions and the penalty that was imposed on the solution either as a result of too low residual pressures or too high flow velocities. <b>Graphical presentation</b> The 20 best solutions are ranked in order of the comparative costs. <b>Computed results</b> Details of the initial selection of the pipe diameters (chromosomes) are given as well as all the detail regarding subsequent generations.		

favourable results (if there are more than 20 possibilities) will be shown. Screen G5 and P7 reflects the graphical presentation of the best solutions respectively for the gravity and pumping system.

In the GAPOP program all the steps that are required for the application of GAs have been included with the objective to demonstrate the use and value of GAs in the water field.

The initial selection of the chromosome strings, defining the diameters for the first analysis, is done randomly on a bit-by-bit basis. Reproduction is done by transferring the best solutions to the mating pool for the next generation. Mating selection is done at random and the cross-over point is also selected randomly. Mutation is done on a uniform basis for all the bits.

### Conclusion

Based on the literature that has been reviewed, it can be stated that:

- GAs are not used to their full potential for optimisation in the water industry in South Africa.
- Potential applications of the technique are:
  - hydrology and water resource assessment;
  - network optimisation;
  - optimisation of rehabilitation, extension and upgrading of distribution networks; and

- operation and maintenance scheduling of pumps and purification plants.
- Little formal teaching on GAs is included in the curriculum of civil engineering courses in South Africa.
- The need for the development of software utility programs that can be used in practice and to stimulate further exploitation of this technique has been identified.

Hopefully this paper provides some insight into the use of GAs and stimulates further implementation of this useful optimisation procedure.

## References

- BARROS AS and RUTLEDGE (1998) Genetic algorithm applied to the selection of principal components – Chemometrics and Intelli. *Lab. Sys.* **40** 65- 81
- GOLDBERG DE (1989) *Genetic Algorithms in Search, Optimisation and Machine Learning*. Addison-Westley Press.
- GUPTA I, GUPTA A and KHANNA P (1999) - Genetic algorithm for optimisation of water distribution systems. *Environ. Modelling and Software* **14** 437-446
- HALHAL D, WALTERS GA, OUAZAR D and SAVIC DA (1997) Water network rehabilitation with structured messy genetic algorithm. *J. Water Resour. Plann. Manage.* **123** (3) 137-146
- HOLLAND JH (1992) *Adaptation in Natural and Artificial Systems* (2nd edn.) Univ. of Michigan Press.
- HUANG C and MAYER AS (1997) Pump-and-treat optimization using well locations and pumping rates as decision variables. *Water Resour. Res.* **33** (5) 1001-1012
- MONTESINOS P, CARCIA-GUZMAN A and AYUSO JL (1999) Water distribution network optimization using a modified genetic algorithm. *Water Resour. Res.* **35** (11) 3467- 3473.
- MORSHEJ J and KALUARACHCHI JJ (1998) Application of artificial neural network and genetic algorithm in flow and transport simulations. *Advances in Water Resour.* **22** (2) 145-158.
- OLIVEIRA R and LOUKS D P (1997) Operating rules for multi-reservoir systems. *Water Resour. Res.* **33** (4) 839-852.
- REIS LFR and CHAUDHRY FH (1997) Optimal location of control valves in pipe networks by genetic algorithm. *J. Water Resour. Manage.* **123** (6) 317-320.
- WALTERS G A, HALHAL D, SAVIC D and QUAZAR D (1999) Improved design of “Anytown” distribution network using structured messy genetic algorithms. *Urban Water* **1** (1) 23-38
- WARDLAW R and SHARIF M (1999) Evaluation of genetic algorithms for optimal reservoir system operation. *J. Water Resour. Plann. Manage.* **125** (1) 25-33.
- YAGI S and SHIBA S (1999) Application of genetic algorithms and fuzzy control to a combined sewer pumping station. *Water Sci. Technol.* **39** (9) 217-224.
- YEO MF and AGYEI EO (1998) Optimizing engineering problem using genetic algorithms. *Eng. Comput.* **15** (2-3) 268-280.

## Annexure 1



Opening screen

Gravitation pipeline

**Project Data**

Project description: This is an example of a gravity system

Designer: Prof SJ van Vuuren

Date: 2001-09-12

**General Data**

Upstream reservoir level: 120      Pipe length units: m

Upstream reservoir ground elevation: 115      Elevation/level units: m

Downstream reservoir level: 65      Demand units: lit

**Profile Data**

Pipe Number	Pipe length	Pipe end point elevation	Pipe end point demand
1	1000	105	0
2	1500	100	10
3	2000	75	15
4	2200	80	0
5	1800	75	0

Add    Delete      View Profile    Continue

Screen G1 or P1 - Project data, general data and profile data



**Pipe diameters to be analysed**

Nr	Pipe material	Pipe diameter	Pipeline roughness	Pipeline cost per meter (R)	Use this option in analysis
1	GRP	100	1	75	<input type="checkbox"/>
2	GRP	150	1	95	<input checked="" type="checkbox"/>
3	GRP	200	1	130	<input checked="" type="checkbox"/>
4	Steel	250	1	190	<input checked="" type="checkbox"/>
5	Steel	300	1	290	<input checked="" type="checkbox"/>
6	Steel	350	1	410	<input checked="" type="checkbox"/>
7	Steel	400	1	650	<input checked="" type="checkbox"/>
8	Steel	500	1	900	<input checked="" type="checkbox"/>

Units

Pipeline diameter units:

Pipeline absolute roughness ( $K_s$ ):


 


Screen G2 or P2 - Pipe material selection

**Design limitations**

Minimum required residual pressure:   m

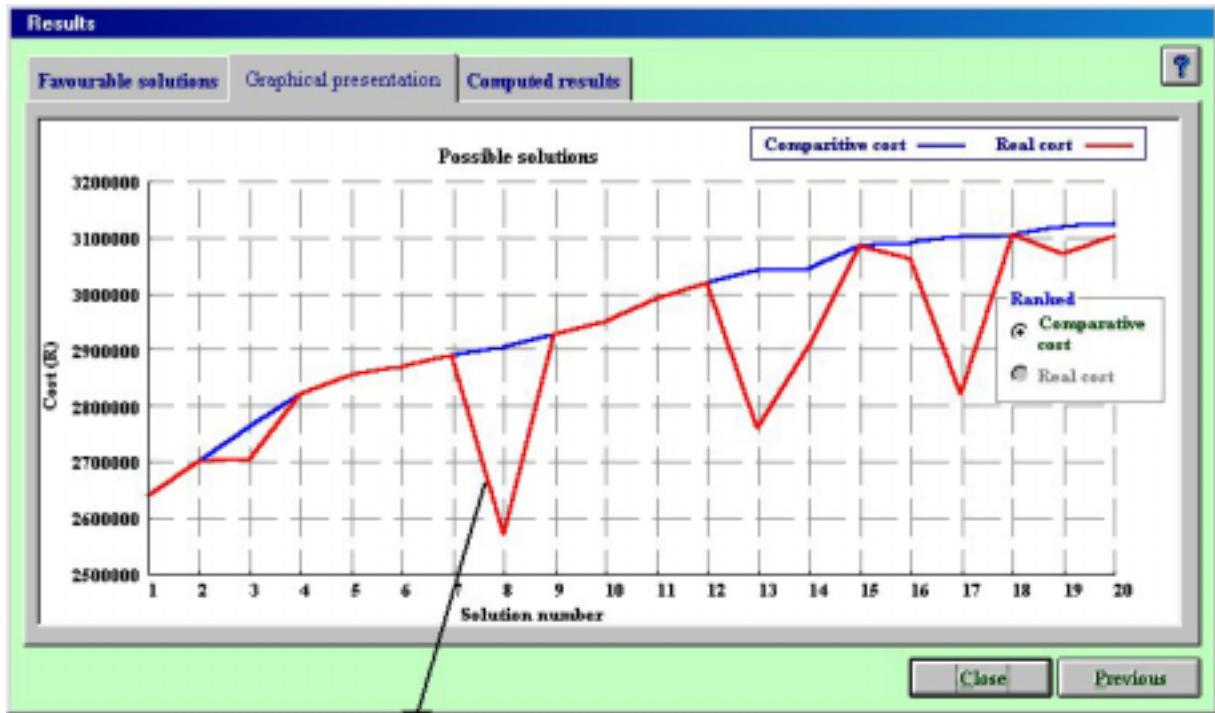
Maximum flow velocity:   m/s

Mutation percentage:  % 



Screen G3 or P3 - Design limitations





Favourable result details of number : 8

Pipe segment	Diameter	Flow	Velocity	Velocity factor	Pressure	Pressure factor
1	350	0.055	0.572	1.000	12.968	1.000
2	300	0.055	0.778	1.000	11.466	1.000
3	250	0.045	0.917	1.000	21.979	1.000
4	300	0.030	0.424	1.000	14.133	1.000
5	200	0.030	0.595	1.000	1.449	2.415
6	250	0.030	0.611	1.000	9.401	1.000

OK

Screen G4 or P6 - Graphical presentation of the 20 best solutions

**Energy details**

Summer

Click in the rate boxes to select the applicable rate

Hour	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Rate	Yellow	Yellow	Yellow	Yellow	Green	Red	Red	Green	Green	Yellow	Yellow	Yellow	Yellow	Yellow	Green	Green	Red	Red	Red	Green	Green	Yellow	Yellow	Yellow

Peak rate R 0.28 /kWh  
  Standard rate R 0.19 /kWh  
  Off-peak rate R 0.9 /kWh

Winter

Click in the rate boxes to select the applicable rate

Hour	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Rate	Yellow	Yellow	Yellow	Yellow	Yellow	Red	Red	Red	Green	Yellow	Yellow	Yellow	Yellow	Green	Green	Green	Red	Red	Red	Red	Green	Green	Yellow	Yellow

Peak rate R 0.32 /kWh  
  Standard rate R 0.21 /kWh  
  Off-peak rate R 0.12 /kWh

Pumping hours

10 hours  
  14 hours  
 18 hours  
 22 hours  
 Number of summer months: 6

12 hours  
 16 hours  
 20 hours  
 24 hours


Pumping efficiency

Pumping system efficiency: 70 %

Previous Continue

Screen P4 - Energy details

**Rates**

**Capital cost** 

**Mechanical works** Cost =  $aX^b$   
 a =  b =

**Electrical works** Cost =  $cX^d$   
 c =  d =

**Maintenance cost**

**Pipeline**

**Mechanical works**



**Electrical works**

**Escalation rate**  %

**Energy cost escalation rate**  %

**Discount rate**  %

**Design life**

Screen P5 - Rates