Recent applications of genetic algorithms to water system design
G.A. Walters, D.A. Savic
School of Engineering, University of Exeter, Exeter EX4 4QF, UK

Abstract

Genetic algorithms and other similar techniques based on mimicking the natural evolution process have been successfully developed over the last few years for the optimisation of complex systems in a wide variety of areas including image recognition, operations research and engineering design. Presented in this paper are recent developments by the authors and others of genetic algorithm techniques applied to water system analysis, design and operation, mainly in the specific area of water distribution.

Introduction

The use of Genetic Algorithms (GAs) and other similar Evolutionary Design techniques has caught the imagination of researchers in the engineering, operations research and mathematical fields on a world-wide basis over the last few years. This is evident from the seemingly exponential growth of papers relating to their use, and the never-ending variations on the basic idea that researchers are seeking to exploit. The basic principles of the GA are not new: being simulations of natural evolution, they are as old as life itself. What is new is the realisation that the natural process of evolution is a very effective optimisation technique for large, complex systems. Armed with an understanding of basic genetics and with modern computer processing power, researchers can now numerically model natural evolution, albeit very crudely, and apply these numerical models to “evolve” solutions to problems involving highly complex systems. Within the field of engineering, the best design for a
component or complete system may be obtained, the set of parameters that best match observed conditions may be derived, or the optimal operating policy for a system may be determined.

The present use of evolutionary based optimisation techniques stems largely from the development of evolution strategies by Rechenberg et al in Germany in the 1960’s and the development of genetic algorithms by Holland et al in the USA soon afterwards. Some of the rapidity of recent developments has come from the combination of ideas from the two separately developed strands of research. The basic ideas are briefly summarised in the following section, and a good introduction to the subject is given by Michalewicz.

Genetic Algorithms and Evolutionary Design

Evolution is the process whereby a species adapts generation by generation to suit its natural environment. A simple model of the evolutionary process will contain the following features:

i) a population, the individuals within which die off and are replaced by offspring

ii) a breeding process, involving the formation of offspring, by combining the genetic information (genes) of parents selected from the population

iii) a selection process by which fitter individuals in a population are more likely to breed (and successfully rear their young) than less fit individuals. Fit individuals are those well suited to their environment.

In the GA, the numerical optimisation process retains the above features. For example, if the optimal design for a scheme is required, a population of different designs is considered. Each design is defined by numerical values for each of the design parameters. Numerical values can be real numbers, integers or Boolean choices, the design being encoded, usually onto a binary string, in a form analogous to a chromosome in nature. The initial population of designs is normally selected at random within the search space defined by the limiting values of the variables.

Selection of parents for breeding can be accomplished by first evaluating the comparative fitness of each individual. Fitness will normally be a measure of how well the proposed design meets its objective. Parents can then be selected from the population with a probability of selection proportional to each individual’s fitness (Roulette-Wheel selection). Alternatively, two individuals are chosen at random and only the fitter becomes a parent (Tournament selection). With either method, the ability to produce offspring is strongly influenced by the fitness of an individual, relative to the other population members.

The breeding process involves the combination of the design parameters from the two parents to form new parameters for the one or more offspring. The aim is to enable potentially good features of the parental designs to be combined to form even better offspring. In a conventional GA, the binary
strings of the two parents, (the chromosomes, which define the designs), are split at a random location and the second parts of the strings are switched over to form two offspring, the process being known as cross-over. The offspring will therefore contain some design features from each parent. In a more general formulation, the genetic information (designs) of the two parents can be considered pooled together, with feasible offspring being grown from the pool of genetic possibilities (Walters and Smith\textsuperscript{4}). The latter approach is particularly useful if the problem is constrained such that the formation of feasible solutions by cross-over is unlikely.

A further mechanism called mutation is introduced into the breeding process, by which random changes are introduced with low probability into the design of an offspring. The simple GA handles mutation by flipping a binary digit (bit) from 0 to 1 or vice-versa. The pooling approach handles mutation by introducing extra possibilities into the pool of genetic material from which offspring are grown.

Using the above processes of selection and breeding, a new population of solutions can be created which replaces the original one, and will contain, on average, individuals which are better than those in the original population. The process can therefore be applied iteratively to produce successive populations, each generally better than its predecessor, the process terminating when no improvement is detected over a given number of generations.

That such a mathematically unsophisticated technique works, and works very well for large complex systems, may seem initially surprising. However, it should be remembered that the technique is mimicking a natural self regulating process which has itself evolved to be a robust and efficient method for producing designs for complex living organisms.

Applications to Water Engineering

As previously reported by the authors (Walters and Savic\textsuperscript{5}), early work using GAs in the water engineering field centred on pipe sizing and layout of water distribution networks (Cembrowicz and Krauter\textsuperscript{6}) with the use of similar algorithms for the design of sewer systems. The implementation of these methods in the design of new water supply and sewerage networks for a number of cities in developing countries is described by Cembrowicz\textsuperscript{7}.

Water distribution network design

Most recent work on GAs for water engineering has been on aspects of water distribution network optimisation, including new design, rehabilitation, modelling and operation. Although design of a complete new network is in practice rare, extension and reinforcement of existing systems occur frequently, particularly for rapidly growing conurbations in developing countries, where shortages of water and inadequate pressures are commonplace. In long established systems, poor water quality (often associated with old iron pipes)
and excessive leakage necessitate substantial investment in pipe replacement, often involving some quite complex decision making to obtain the best overall strategy for improvement.

**Pipe sizing**  Conceptually the simplest of the GA applications, pipe sizing is essentially about determining the set of pipe diameters from a selection of commercially available sizes which minimises the cost of construction for a network of given plan layout, whilst maintaining adequate pressure at all network nodes for one or more sets of specified nodal demands. As an alternative to GAs the basic problem can be solved by the Linear Programming (LP) approach of Labye\textsuperscript{8} for tree-like networks, and by the gradient descent method of Alperovits and Shamir\textsuperscript{9} for looped systems, although both generate solutions with pipelines split into two lengths of different diameter pipes. It should also be noted that, whatever the method used, looped networks will degenerate into tree systems if designed on a minimum cost basis with a single set of design flows (Quindry et al.\textsuperscript{10}).

In the GA approach, each pipe diameter is treated as an integer variable, 1, 2, 3 etc. corresponding to different pipe sizes. The chromosome can be a binary or integer string of these numbers, which thus defines the network design. A simulation routine (network solver) is required to analyse each candidate design in terms of pressure at the nodes, and a separate routine determines the construction cost. During the evolution, many solutions will be generated which are infeasible in terms of unsatisfactory nodal pressures. Rather than discarding these solutions, it is found advantageous to treat them as feasible and to add to the cost a penalty function which penalises inadequate pressures. Judicious choice of penalty function will lead the GA to converge on an optimal solution in which all pressures are adequate.

In addition to designing new networks, the method can readily be employed in designing extensions to existing systems, as existing pipes can be included with fixed diameters. The method has been used to design several water distribution network extensions in Australia (Simpson et al.\textsuperscript{11}) with large savings in capital expenditure over conventionally designed schemes. A comparison of the GA with other techniques for tackling previously published problems is given in Savic and Walters\textsuperscript{12}, in which the sensitivity of the solution to the design parameters is also considered.

**Layout**  The simple approach above requires that all pipelines must have at least a minimum diameter in any solution. A problem in which one or more pipes may have zero diameter (i.e. be excluded from the design) cannot be handled without substantial modifications to the algorithm, as the basic topology of the network is altered by excluding links. This requires reorganisation of the data input to the solver for network analysis, and more fundamentally, will give rise to the generation of disconnected networks (networks in which at least one node is isolated from the source). As the
network size increases, the chances of forming a connected network by random choice of diameters (from a set which includes a zero option) rapidly decreases, becoming negligible for networks involving hundreds of pipes. The simple GA approach becomes unworkable, and other methods have been adopted, based on the idea of finding the optimal layout and (non-zero) pipe sizes simultaneously. Cembrowicz describes a variety of approaches using different GA codings, with applications to real problems. Walters and Smith describe a very efficient evolutionary design algorithm for generating minimum cost tree-like networks, which can be used as the base for a looped distribution network.

**Rehabilitation** The basic algorithm for pipe sizing can, for simple cases, be used to determine how best to rehabilitate an existing network which is performing unsatisfactorily. Rehabilitation options can include pipe removal and renewal (with pipes of any one of a set of available diameters), pipe insertion with the same size or smaller inserts, pipe duplication with pipes of any diameter, pipe cleaning, pipe lining or taking no action. The decision for each pipe can be coded as an integer choice as previously.

If there is sufficient money available to upgrade the whole network to a satisfactory standard, the optimisation can be stated as: minimise the cost of upgrading subject to satisfactory network performance. For a small to medium sized network, the above approach will work well. However, several difficulties arise when attempting to apply the approach to large networks. First, the amount of money available will usually be less than that required for complete rehabilitation. Second, the number of variables will be very large, making numerical techniques difficult to implement. In practice, with restricted funds, it is inevitable that only a small subset of the network pipes will be selected for rehabilitation. On the one hand, defining a variable for every pipe in the network is very inefficient, and on the other, pre-selecting a set of candidate pipes to consider for improvement is unnecessarily restrictive.

The problems were overcome by Halhal et al. using a **multiobjective** approach, combined with a specially developed **Structured Messy Genetic Algorithm (SMGA)**. The SMGA was introduced to avoid handling the very large data strings necessary when there is one data item for each pipe in the network. Essentially the SMGA starts by considering simple one element strings, corresponding to a single decision variable (one pipe), and progressively builds up longer and longer strings of decision variables as the evolution proceeds. This parallels the natural process in which complex life-forms evolve from single cell structures. Only those pipes for which some action is specified are included in a string, and a maximum string length can be specified, corresponding to the largest number of pipe improvements that can be made from the available fund.

With a limited fund, the basic problem is to determine the set of improvements which will maximise the benefit to the system, in terms of reducing supply shortfall, inadequate pressure, leakage, breakages etc. Within
the SMGA, it is necessary to compare the fitnesses of the partial solutions that evolve. For instance, at the start of the process, all individual actions are considered for all pipes as potential one element strings. It is necessary to select a limited number of the single element strings, from which a second generation of two element strings are created. The problem of how to evaluate the fitness of the single elements is important. Neither cost nor benefit alone is sufficient, as low cost solutions may have little potential benefit and high benefit solutions will tend to have high cost. A multi-objective approach, using minimum cost and maximum benefit as the objectives, was therefore adopted.

The multi-objective approach fits in very well with the overall problem, which can be redefined as: find solutions which give the best possible benefits for a range of costs up to the maximum amount of money available. A set of non-inferior solutions is developed, (a non-inferior solution is one which cannot be bettered by any other solution on both criteria), during a single run of the SMGA, using techniques to ensure that solutions are spread over the full range of allowable costs.

**Pressure Regulation** In the UK, one of the major concerns that water companies face is leakage from water distribution networks. Reduction in pressure has a substantial influence on leakage, and one low-cost way of implementing pressure reduction is to alter the network configuration by resetting stop valves, present along most network links. This can have the effect of partially isolating higher level zones from low lying areas, thus reducing pressures in the latter. The on/off setting of a stop valve is simply represented by a binary digit, giving a very straightforward GA coding. However, the standard GA operators are ineffective, since crossover and mutation lead very frequently to disconnected, and hence infeasible, networks. A problem specific but much more efficient evolution program was developed by Savic and Walters** in which feasible (connected) networks are generated from the pooled designs of parents.

**Calibration** Simulation of system performance by mathematical modelling is a key tool in the efficient design and operation of distribution networks. However, models invariably require incorporation of numerical parameters, exact values of which are not known. For instance, pipe roughnesses can only be estimated, and even pipe diameters may not be recorded with any certainty or precision. The technique of selecting values for the unknown parameters, known as network calibration, can be treated as an optimisation process in which the difference between sets of observed and simulated readings of flows and pressures is minimised.

Savic and Walters** have shown that the network calibration can be handled effectively by a GA, with considerable improvement in the accuracy of the ensuing model compared with the traditional trial and error approach.
Pump Scheduling

One of the operational aspects of water supply that is amenable to optimisation is that of minimising the cost of pumping water. Normally a supply system will be fed by a number of electrically driven pumps, probably from several separate sources, which will feed water to a number of service reservoirs distributed throughout the system. Typically, electricity is available at different tariffs during peak and off-peak periods. Demand for water varies through the day, and the service reservoirs provide storage to supply peak demands.

The optimisation problem is to determine which pumps to operate at which times of day to meet the predicted demand for water at minimum cost for electricity. The normal scheduling horizon is one day, split at most into 24 one-hour periods during which pump settings remain constant. After one day, the schedule is repeated, or a revised schedule may be implemented if the network has changed (e.g., for repair or maintenance or supply interruption) or if predicted demands have altered.

At first sight, the problem appears to be well suited to solution by a GA approach. The decision on whether to operate a particular pump during a particular time interval can be coded as a binary choice corresponding to off or on. The complete 24-hour schedule can therefore be specified by a string with 24 bits for each pump in the system. Software is needed to simulate the pump, pipework, reservoir system for the 24-hour period for each schedule generated. If pumps are variable speed, rather than the fixed speed units assumed above, several bits are required to define operating speed, and so substantially longer strings are required.

The constraints on the schedule need to be carefully handled. The volume in any reservoir cannot be negative, and in practice should not be allowed to fall below a safe minimum at any time. Neither can a reservoir store more than its full volume, and it is clearly undesirable for any overspill to occur due to overpumping. Also, the total volume supplied to each reservoir should equal the total amount drawn from each reservoir over the 24-hour period, i.e., the reservoir volumes at the end of the 24-hour period should equal the volumes at the start. Penalty functions can be devised for the above constraints which ensure feasibility of the final optimal schedule.

A simple trial of the above procedure was reported by Mackle et al. Further work incorporating a multi-objective approach was carried out by Schwab et al. This considered not only the cost of electricity, but also the pump switching as objectives. Schedules involving numerous changes to a pump’s status are considered undesirable due to extra wear and tear on the pump unit, motor and switch-gear, and possible pressure surges in the pipe system. So a low-cost schedule that involves few pump switches is required. If the actual cost of pump switching were known precisely, then the electricity and pumping costs could be combined into a single objective. However, a satisfactory alternative is to present a range of non-inferior solutions, which will clearly show the trade-off between electricity cost and number of pump switches.
switches. As previously discussed, the GA provides an excellent vehicle for implementing a multi-objective search, as solutions within the population can evolve over the complete range of interest of the variables.

Discussion

The above developments show that GAs can be used for a wide variety of water engineering optimisation problems, but it is important to realise that in certain cases other methods may out-perform the GA. The size of the solution space (the number of possible discrete solutions to be searched) and the characteristics of the search space are important. GAs show their true value when applied to search spaces which are very large, and will typically require hundreds or thousands of generations to be formed before converging on the best solution. For small problems, a complete enumeration of all solutions may well be more efficient! Likewise problems that can be adequately solved by Linear Programming are unlikely to be solved more rapidly by GAs.

As the GA uses a population of solutions spread throughout the solution space, it has a tendency to converge onto the global optimum, rather than a local optimum. This is a very important advantage over other optimisation techniques, as most search spaces of practical interest contain many local minima. Conventional hill-climbing methods use only local information to guide the search, and hence will converge on the nearest local optimum.

Another advantage deriving from the use of a population of solutions is that a variety of near optimal solutions can be found and retained. In fact, in many examples, the authors have found radically different solutions with very similar cost. This gives the decision maker a range of schemes from which to choose.

GAs also work better for discrete valued variables, typical of many engineering design problems, than for continuous variables, which must be discretised to be in a form that the conventional GA can handle. Nevertheless, a development of GAs using continuous variables has been used with success, for example in the calibration model previously described.

In the authors’ experience, GAs combine very well with multi-objective optimisation, as seen in both the rehabilitation and the pump scheduling models. As a population of solutions is being processed, it is relatively easy to ensure the evolution of a set of non-inferior solutions, rather than a cluster of solutions which meet one objective well.

The authors see GAs as being a very valuable tool for practical multivariable engineering optimisation, being robust, relatively easy to apply and suitable for very large systems. Considerable increase in efficiency of the evolution can however be achieved if either the GA or the design problem can be structured such that infeasible solutions are not created by the conventional processes of crossover and mutation. A suite of programs is being developed by
the authors to handle all aspects of water distribution network design and operation.

Conclusion

The use of Genetic Algorithms and other Evolutionary Design techniques for the optimisation of large water systems is rapidly gaining acceptance, both in research and in practice. As an optimisation technique, Genetic Algorithms show considerable advantages over other techniques for large, complex engineering systems.

Acknowledgement

The authors gratefully acknowledge the support of the Engineering and Physical Sciences Research Council (Grant Reference: GR/J09796)

References


