10th International Conference on Hydroinformatics
HIC 2012, Hamburg, GERMANY

MULTI-OBJECTIVE OPTIMISATION OF WATER DISTRIBUTION SYSTEMS USING A GENETIC ALGORITHM EMBEDDED WITH AN ENHANCED EVOLUTIONARY DIRECTION CROSSOVER OPERATOR

E. BARLOW & T.T. TANYIMBOH
Department of Civil Engineering, University of Strathclyde, Glasgow, UK
euan.barlow@strath.ac.uk, tiku.tanyimboh@strath.ac.uk

The optimal design of a water distribution system (WDS) is a non-linear multi-modal multi-objective problem which generally involves an extremely large discrete decision space. Genetic algorithms (GAs) present an intuitive approach to solving such problems. These algorithms are particularly suited to searching large decision spaces and can avoid convergence to local optima. The construction of a GA is designed to mimic the process of natural evolution. A large population of random networks will evolve through successive generations towards the pareto-optimal front; however, due to the stochastic nature of a GA, the number of network evaluations required for convergence can be extremely large. For a large WDS, each network evaluation can be time-consuming and a standard GA may require millions of solutions to be evaluated. It is therefore desirable to speed up the convergence of the GA in order to make the solution of large networks feasible. The evolutionary direction crossover (EDC) operator is a mechanism which is capable of following the natural course of evolution inherent to a GA. At a given generation, the direction of evolution between parents and children is identified. Progressive evolutionary directions are explored further to determine whether additional improvements can be made. This process will potentially advance the evolution, and thus achieve accelerated convergence. An enhanced EDC operator (EEDC) is proposed here, which is simpler to implement than EDC and is more suitable for application in a multi-objective environment. A modified GA is employed, with the EEDC operator embedded within the framework of a standard GA. In this paper, the EEDC operator is used in conjunction with the non-dominated sorting genetic algorithm II (NSGA II), although any GA could be used alternatively. Once the child population is generated, EEDC is applied to each child with a fixed probability. The fitness of each child is not assessed until after EEDC has been applied, thus ensuring that no additional fitness evaluations are required. The enhanced GA therefore incorporates the EEDC operator without significant increase in complexity or computation time. The performances of the enhanced algorithm and the standard NSGA II are compared for the solution of the Hanoi network, a benchmark problem from the WDS literature. The enhanced algorithm is shown to substantially improve the convergence of the population, particularly in the early stages of the evolution. Applied to a large WDS this improved convergence will dramatically reduce the required computation time. This paper therefore presents a progression towards the tractable optimisation of real-life water distribution systems.

WATER DISTRIBUTION SYSTEMS

Water distribution systems (WDSs) are integral components to the effective design of urban areas, and the ever-increasing urbanisation in developed and developing countries world-wide establishes the problem of optimising WDS design as an important research area. These systems are the networks of pipes, pumps, reservoirs, tanks and nodes which transport drinking water from the supply source to the users. A detailed analysis and review of the modelling of WDSs is given by Haestad et al [7]. One goal of WDS design is to find the cheapest network configuration which provides an adequate pressure supply at all user outlets. Physical requirements which must be satisfied are that mass and energy are conserved throughout the network. Solutions were traditionally found by considering the single-objective optimisation problem of minimising the cost subject to the constraints of satisfying the nodal pressure requirements and the conservation of mass and energy. An increasing trend is to formulate the problem as a multi-objective optimisation, where the competing objectives are to minimise the cost and minimise the nodal pressure deficits throughout the network. The decision variables considered are then the diameters of the pipes. With the nodal deficit an objective to be minimised rather than a constraint which must be satisfied, a WDS designer is able to assess whether the extra cost required to achieve a zero pressure deficit is preferable to incurring a slight deficit for a cheaper cost. By analysing each network with a hydraulic solver, the remaining constraints are handled implicitly and there is no need to resort to penalty functions. In this setting, all solutions which lie within the bounds of the decision variables are therefore feasible as the hydraulic solver will ensure that the remaining constraints are satisfied. It is standard in the WDS literature, however, to measure the effectiveness of a solution technique by its ability to find the least-cost zero-deficit solution. In the results presented here, the focus is therefore on zero-deficit solutions, and these will be referred to as the feasible solutions for the remainder of the paper.

The problem can be formally expressed as

$$\min \ \boldsymbol{f} = (f_1, f_2),$$

where the network cost is

$$f_1 = \Sigma_{i \in NP} \, c(d_i, l_i)$$

and the total pressure deficit across the network is

$$f_2 = \Sigma_{j \in NN} \max(hr_j - hp_j, 0).$$

The cost of a specific pipe is given by the function c which is dependent on the length and diameter of the pipe. The required head and the nodal pressure at node j are given by hr_j and hp_j respectively. NP is the set of all pipes in the network and NN is the set of all demand nodes. The conservation of mass is satisfied by balancing the inflow and outflow at each node $\Sigma_{ij\in J}\ q_{ij} = Q_j$, where q_{ij} is the flow-rate in pipe ij linking nodes i and j, J is the set of all pipes incident on node j and Q_j is the inflow or outflow at node j. When node j is a demand node Q_j is equivalent to the nodal demand. Conservation of energy requires that the net head-loss around each closed loop L must equal zero: $\Sigma_{ij\in L}\ hf_{ij} = 0$, where the head-loss hf_{ij} in pipe ij is defined here by the Hazen—Williams equation $hf_{ij} = \omega\ l_{ij}\ d_{ij}^{\ b}(q_{ij}/c_{ij})^a$. The pipe length and diameter are given by l_{ij} and d_{ij} respectively, with C_{ij} representing the roughness co-efficient of the pipe. The dimensionless factor ω enables conversion between alternative units and the exponents a and b are constants.

There are two major categories of hydraulic solver [9] which are commonly used: demand driven analysis (DDA) and head/pressure driven analysis (PDA). The solver used here is Epanet 2.0 [8] which is a DDA solver, although any other DDA or PDA solver could be used alternatively.

GENETIC ALGORITHMS

Genetic Algorithms (GAs) are widely used in the solution of complex optimisation problems in many areas of science and engineering. A detailed introduction to GAs and their application in solving several real-world problems is given by Deb [3]. The strengths of a GA are well suited to the optimal design of a WDS; however, their stochastic nature can result in slow, unreliable convergence to the pareto-optimal front. As the hydraulic evaluation of a WDS requires an external solver, each fitness evaluation is important.

The GA employed here is the non-dominated sorting genetic algorithm II (NSGA II) [4] which has been applied to a vast array of problems. In this implementation of NSGA II, the decision variables are integer-coded, where each integer represents a specific pipe-size. A population of 200 individuals is maintained, with 200 children created at each generation. Individuals are selected for crossover through a randomly populated binary tournament; one-point crossover is then applied to the two selected individuals with the probability of crossover equal to 1.0. The crossover-site is randomly selected and two children are produced from each pair of parents selected. Mutation is applied to each decision variable with a probability of $1/N$, where N is the number of decision variables or pipes. If a particular decision variable is selected for mutation either a random mutation or a creeping mutation is applied, each type with a probability of 0.5.

EVOLUTIONARY DIRECTIONAL CROSSOVER

The evolutionary directional crossover (EDC) operator [10] is designed to manipulate the direction of evolution between a child and its parents in order to produce a fitter individual. The new individual generated through EDC, C^*, is defined by Yamamoto and Inoue as

$$C^* = C + \varphi_1 d_1 + \varphi_2 d_2,$$

where C represents the child in decision space, d_i represents the evolution from the parent to the child in decision space, and φ_i is a multiplicative factor which identifies the progressive direction of this evolution. The direction of evolution between a parent and a child can be represented by the vector defining the difference between the child and the parent in decision space, $d_i = C - P_i$, where P_i represents the parent in decision space. The multiplicative factor φ_i is defined by $\varphi_i = \text{sign}(F_C - F_{Pi})R_i$, where F_C and F_{Pi} are the fitness of the child and parent respectively and R_i is a random real number between 0 and 1. The combined term $\varphi_i d_i$ therefore promotes exploitation in the progressive direction of evolution between the child and the parent.

In its original implementation, the application of the EDC operator to the multi-objective WDS design optimisation is not straightforward. A bounded decision space and multiple objectives raise problems with fitness comparisons and the feasibility of new individuals generated. Additionally, EDC incurs additional fitness evaluations in every generation in comparison with a standard GA, as well as an increase in computer processing unit (CPU). These issues reduce the effectiveness and increase the complexity of the EDC operator for WDS problems.

ENHANCED EVOLUTIONARY DIRECTIONAL CROSSOVER

The underlying principle behind the method proposed here is that the children generated through the standard selection, crossover and mutation operators do not necessarily improve the population. The exploration potential of a GA can often result in fast convergence to a population of non-dominating solutions which occupy promising areas of the objective space, but are not pareto-optimal. With no means to exploit these promising areas, the evolution stagnates and the GA has to rely on chance to finally converge to the pareto-optimal solutions. Depending on the problem, this can take many generations. When the population of an elitist GA remains largely unchanged over many generations this indicates that the children produced are not improving the population from generation to generation. The children are therefore dominated by one or more individuals from the parent population.

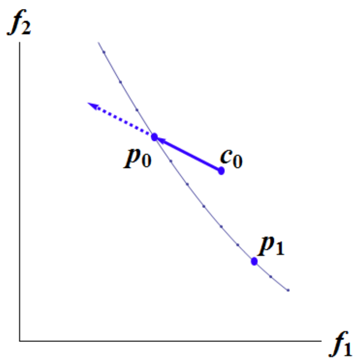


Figure 1: A sample portion of the population – each individual in this region is represented by a dot and the current non-dominated front is represented by the curved line. The three individuals highlighted by large dots are the parents, P_0 and P_1, and the child, C_0. The solid arrow indicates the progressive evolutionary direction from the child to one of its parents, the dashed arrow indicates the direction of evolution searched by the EEDC operator from the parent.

If a parent lies on the leading non-dominated front and a generated child is dominated by some member of the population, then the evolutionary direction from the child to the parent can be considered as a progressive direction. That is, the parent lies in a better region of objective space and so a move from the child to the parent is an improvement. Further exploitation of this evolutionary direction from the position of the parent may lead to additional improvements; this reasoning is embodied in the enhanced evolutionary directional crossover (EEDC) operator and is displayed in Figure 1. To apply EEDC to a child, one parent is randomly selected (each parent has equal probability of selection), and the child is assumed to be poorer than this parent with respect to the current population. The progressive evolutionary direction is therefore assumed to be the change from the child to the parent and the new individual generated by EEDC is given by

$$C^* = 2P_i - C.$$

Any infeasible decision variables are truncated to the appropriate boundary values. The EEDC operator therefore exploits the (assumed) progressive evolutionary direction from the position of the parent in decision space, with the possibility of identifying a more promising individual. The EEDC operator is applied within the framework of a standard GA: once the child population is generated through selection, crossover and mutation, each child is then selected for EEDC with a fixed probability, p_ε. The EEDC operator is then applied and the child is replaced with the new individual, C^*. The fitness of the child population is only assessed after EEDC has been applied, thus no additional fitness evaluations are required in comparison with the standard GA.

As the fitness of the child and parent are compared with respect to the current population rather than directly with each other, no surrogate fitness measure is required. The burden of identifying the dominance relationship between the child and each member of the population is negated however by simply assuming that the child is dominated. The implementation of EEDC given above is a simple subtraction of two vectors which is straight-forward to apply and the increase in CPU time in comparison with a child generated only by crossover and mutation is minimal. Additionally, as it is unnecessary to evaluate the fitness of the child prior to application of the EEDC operator, generating the new individual C^* requires no extra fitness evaluations than required by a standard GA. The EEDC operator therefore draws on the strengths of EDC in exploiting progressive directions of evolution, while addressing several of the difficulties discussed above; however, a move in the progressive evolutionary direction from the starting point of the parent may not be as effective as the local improvement from the child to the parent. This will depend on, among other things, the position of the child and parent with respect to the bounds of the decision space and the position of the child relative to the current non-dominated front.

The EEDC operator is presented here in an extremely simple form in order to avoid additional fitness evaluations in each generation, and to minimise the additional CPU time required per application of the operator. The implementation of EEDC could be more refined in principle, which is one aspect of ongoing investigations.

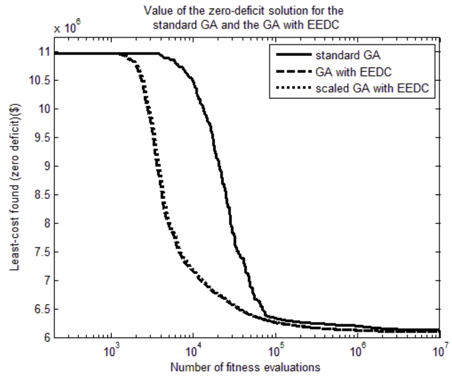
Table 1: Performance measures of the GA, the GA with EEDC and the scaled version of the GA with EEDC. Rows 1-5 are averaged over 100 runs for each algorithm. **Averaged over converging runs, no method showed 100% convergence.*

Measures	GA	GA with EEDC	Scaled GA with EEDC
Average final cost (at 10^7 FEs)	6.13 M$	6.10 M$	6.10 M$
Average number of FEs required to find a feasible solution within 1% of optimal cost	1,356,600	201,000	211,860
Average number of FEs required to find a feasible solution within 5% of optimal cost	80,400	52,000	54,811
Average number of FEs required to find a feasible solution	25,444	3,178	3,350
Average number of FEs required to find the optimal solution*	1,621,600	1,060,700	1,118,000
Smallest number of FEs required to find the optimal solution	158,400	203,600	214,610

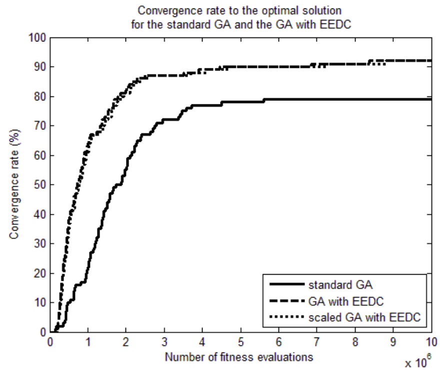
RESULTS

The EEDC operator was tested on the solution to the Hanoi network [6], a benchmark problem in the WDS literature. This single-reservoir network consists of 32 nodes connected by 34 pipes. The six commercially available pipe-sizes are {12, 16, 20, 24, 30, 40} inches, giving a solution space comprising 6^{34}, or 2.87×10^{26}, possible network configurations. The cost associated with each pipe is given by $c(d_{ij},l_{ij})=1.1d_{ij}^{1.5}l_{ij}$ and the required pressure at each node is 30 m. The demand pattern and pipe-lengths can be found in [6].

This problem has been widely studied in the literature with a variety of techniques applied, see Bolognesi et al. [1] and references therein. The poor performance of GAs in finding the optimal solution to this problem has been reported by various authors [2,5]. To address the poor convergence of the GA, each algorithm is allowed to run here for 10 million function evaluations (FEs) and each algorithm is tested on 100 randomly initiated runs. This marks a substantial increase in analysis in comparison with other studies in the literature, which tend to use between 100,000-500,00 FEs and 1-50 runs; however the poor convergence of the GA indicates that a much more robust investigation is required. In addition to comparing the performance of the GA and the GA with EEDC, a scaled version of the GA with EEDC is included which address the increased CPU time incurred by the GA with EEDC. The ratio of the average CPU time per FE for the GA with EEDC, T_E, to the average CPU time per FE for the GA, T_G, is calculated, and this ratio is used to scale the FEs required by the GA with EEDC. That is $FE_S = FE_E\, T_E\, /T_G$, where FE_E is the number of FEs incurred by the GA with EEDC and FE_S is the number of FEs incurred by the scaled version of this algorithm. Table 1 and Figure 2 display the results of the GA, the GA with EEDC and the scaled GA with EEDC applied to the Hanoi network with the hydraulic co-efficients as standard in Epanet 2.0 [8], that is a = 1.852, b = 4.871 and ω = 10.6744 when the units of the head-loss equation are (m, m^3s^{-1}) (ω = 4.727 when the units are (ft, cfs)). The least-cost feasible solution found in the literature to date for the standard Epanet 2.0 head-loss parameters is 6.081 M$. The probability of EEDC applied to a child here was set at p_{ε}= 0.5. This probability was found



(a)



(b)

Figure 2 (a): The average least-cost feasible solution found per number of fitness evaluations, over 100 runs for each algorithm. For each run, the least-cost solution is assumed to be the maximum-cost solution (10.97 M$) until a feasible solution is identified. (b) The percentage of 100 runs converging to the optimal solution per number of fitness evaluations for each algorithm. The standard GA is represented by the solid lines and the GA with EEDC is represented by the dashed lines. The dotted lines represent the GA with EEDC where each function evaluation is scaled with respect to the percentage increase in CPU time in comparison with the standard GA.

to perform well, although no rigorous tests were performed to optimise this value; a range of probabilities between 0.05-0.6 were investigated, and the results were similar.

The results clearly indicate that the GA with EEDC represents a marked improvement over the standard GA in terms of convergence speed and convergence rate. Figure 2(a) shows the average least-cost feasible solution found per FE over the 100 runs for each algorithm. The GA with EEDC finds a lower cost per FE throughout the entire evolution, with a quicker, smoother convergence towards the region of the optimal solution. In Figure 2(b) the number of runs that converge per FE is shown, and again the GA with EEDC is superior. The convergence rate of the GA improves slowly as the FEs increase, reaching a maximum of 79% convergence after 10 million FEs. In comparison the convergence rate of the GA with EEDC increases rapidly over the initial stages, and reaches the GA's final convergence rate in under 1.8 million FEs, less than one fifth required for the standard GA. After roughly 2 million FEs the rate slows slightly, but it is evident that runs are still converging over the entire evolution, whereas the GA shows no new converging runs over the last 4.5 million FEs. Both figures clearly show that the increased CPU time incurred by the GA with EEDC is minimal in comparison with the standard GA, and the behaviour of the GA with EEDC and its scaled version are very similar.

CONCLUSIONS

A new operator, the enhanced evolutionary directional crossover (EEDC) is proposed here, which fits seamlessly into the operation of any standard genetic algorithm (GA). This operator is demonstrated on the Hanoi water distribution system (WDS), a benchmark optimisation problem which has been widely studied. The GA with EEDC substantially improved the performance of the standard GA, demonstrating improved convergence speeds and rates. The EEDC incurs a minimal increase in CPU time on comparison with the standard GA, and is straightforward to implement. Improved GA performance may provide a positive step towards the ultimate goal of achieving real-time optimisation of real-world WDSs, and the results presented here indicate that the GA with EEDC could make an important contribution to this goal.

REFERENCES

[1] A. Bolognesi, C. Bragalli, A. Marchi, S. Artina, Genetic heritage evolution by stochastic transmission in the optimal design of water distribution netwroks, Adv. Eng. Soft., 41 792-801 (2010).

[2] G.Dandy, A. Wilkins and H. Rohrlach, A methodology for comparing evolutionary algorithms for optimising water distribution systems, Proc. WDSA 2010, 12-15 Sept. 2010, Tuscon, AZ, USA (2010).

[3] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, John Wiley & Sons Ltd, New York (2001).

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE T. Evolut. Comput., 6 (2), 182 – 197 (2002).

[5] R. Farmani, D.A. Savis and G.A. Walters, Evolutionary multi-objective optimization in water distribution network design, Eng. Opt., 37 (2) 167-183 (2005).

[6] O. Fujiwara and D.B. Khang, A two-phase decomposition method for optimal design of looped water distribution networks, Water Resour. Res., 26 (4), 539–549 (1990).

[7] Haestad Methods, T.M. Walski, D.V. Chase, D.A. Savic, W. Grayman, S. Beckwith and E. Koelle, Advanced water distribution modeling and management, Haestad Press, Waterbury CT (2003).

[8] L.A. Rossman, Epanet 2 user’s manual, U.S. Environmental Protection Agency, Cincinnati OH (2000).

[9] T.T Tanyimboh and A.B. Templeman, Seamless pressure-deficient water distribution system model, J. Water Management, ICE, 163 (8), 389-396 (2010).

[10] K. Yamamoto amd O. Inoue, New evolutionary direction operator for genetic algorithms, AIAA J., 33 (10) 1990-1993 (1995).