8-1-2014

SWMM5-EA – A Tool For Learning Optimization Of Urban Drainage And Sewerage Systems With Genetic Algorithms

Assela Pathirana

SWMM5-EA – A TOOL FOR LEARNING OPTIMIZATION OF URBAN DRAINAGE AND SEWERAGE SYSTEMS WITH GENETIC ALGORITHMS

ASSELA PATHIRANA (1)
(1):Water Science and Engineering Department, UNESCO-IHE, Delft, The Netherlands

A software tool developed to help urban water engineers to lean the application of evolutionary computing techniques is presented. A popular urban drainage network modeler is coupled with an evolutionary computing library to create an optimization system driven by an accessible graphical user interface. The design approach results in software that does not sacrifice range of applicability while being user-friendly.

.

INTRODUCTION

Optimization problems arise widely in the urban water sector. They range from calibration of complex models [1] to least-cost design of urban water systems subject to various constraints [2]. These problems are formulated either as single objective [1,2] or multi-objective [3] . An example for a multi-objective optimization problem is the optimal relationship between investment in upgrading an urban drainage system and the reduction in number of flooding events.

Any optimization problem involves one or more `objective functions' (function whose maximum or minimum value is sought) and a number of decision variables, where the former can be specified as a unique function of the latter. In other words, once a set of values for the decision variables are known, the corresponding value of an objective function can be calculated. One common feature of many optimization problems occurring in the urban water sector is the computational complexity of calculating the value of objective function from a given set of decision variables, due to the use of sector specific modeling techniques (e.g. Hydrologic, Hydraulic and water quality models, and financial models). Metahuristic methods, particularly those based on evolutionary algorithms (EA) are popularly used to solve such problems. Some of the reasons for this popularity are discussed in the next section.

Examples of urban water problems that are known to work well with EA include, water conduit sizing problems [4], Planning flood control measures against future change [5], optimal planning of flood management with flexibility [6], optimal water quality management [7, 8] and model calibration [9]. While optimization is no panacea, properly applied with professional judgement, it can work as a useful tool for the practicing water engineer.

However, in spite of the developments in application of EA in the urban the water sector, it is not widely used by urban water engineers in their day to day work. A main reason for this underutilization is the lack of accessible optimization frameworks. Although there are accessible tools for very specific applications: e.g. optiDesigner (http://www.optiwater.com/optidesigner.html) for least-cost drinking water pipe network design and SewerGEMS (http://www.bentley.com/en-US/Products/SewerGEMS) optimizing Sustainable Drainage Systems. (Both these solutions are commercial systems.) While these provide useful optimization applications in specific domains, they are too restrictive in the

range of application. The other extreme is to `glue-together' a general purpose EA library and other required algorithms (e.g. an urban drainage simulation model) using computer programming languages [5,8,10]. While this approach provides flexibility to potentially implement any optimization scheme, the computer programming skills demanded from the user make it inaccessible for many.

Another reason for the limited proliferation of EA application in the domain of water engineering is the lack of adequate coverage of the topic in (postgraduate) engineering curricula. In many engineering programs, lectures on optimization and sometimes specifically on EA are provided. However, a major challenge faced by instructors is how to facilitate students to apply the knowledge provided in these lectures for practical problems. Such applications are crucial to internalize the knowledge and to provide viable ways of extending the knowledge to true real-world applications [11]. The biggest barrier to this is the lack of accessible tools to allow the student to experiment with. This is often not a big issue in disciplines like computer engineering, where the students are exposed to courses computer programming. They can be introduced to the approach of programmatically integrating EA libraries with necessary sector-specific routines. The author has attempted this approach in classes with students of largely Civil and Environmental Engineering backgrounds and in various training and capacity building programs involving a variety of users from the urban water sector in several countries. In the case of classroom teaching, a four day crash course on computer programming was delivered in a blended learning mode [12] to provide instructions in computer programming to students. Towards the end of this class some EA applications were introduced. The results were mixed. While some students respond well to learning computer programming, others found it difficult to master it adequately in four days. Achieving the learning objective of being able to apply EA to a practical water problem thus depended on successful achievement of a set of learning objectives related to computer programming. In the case of other training/capacity-building programs this was even more challenging due to the lack of scope/time to properly introduce the required programming skills. What was ideally needed was a software tool that does not demand computer programming skills – therefore accessible – but at the same time provides enough flexibility to apply to a variety of optimization problems in a sector.

This paper presents a tool that fulfills these needs to some degree. SWMM5-EA was created for applying EA in urban drainage networks. It is a tool designed primarily for engineering education, therefore it is designed to be `user-friendly' with a modern user interface, uncomplicated installation procedure and straight-forward and well-documented usage pattern. At the same time, it allows application of EA to a variety of optimization problems associated with urban drainage networks. This makes it suitable for diverse research/operational applications that the student may move into at later stages. SWMM5-EA is licensed under the version 3.0 of GNU General Public License and available for download (https://code.google.com/p/swmm5-ea/). Currently there are no user friendly multi-purpose multi-criteria optimization tools available in the urban water sector while many problems demand such a tool. (e.g. see the applications in references [1] though [9].) Therefore the objective of this paper is to introduce this tool to the engineering education community with the hope that it will be widely used to overcome the aforesaid challenge to promote 'learning-by-doing' in the subject of optimization in urban drainage.

The next section discusses the architecture of the tool: its constituent components and how the tool is integrated. Then the typical procedure for use is demonstrated followed by a list of possible uses. Finally conclusions are presented.

THE ARCHITECTURE OF THE TOOL

How EA work

As the name implies, Evolutionary Algorithms mimic the generational advancement of life as explained by Darwinian evolution. Both in natural evolution and EA, various properties are encoded by genes, which combined into a `genotype', fully express an individual. The individual that results in expressing of genes in a genotype is called 'phenotype'. As with natural evolution, EA exploits the process of a large number of iterations consisting of random mutations and crossover (mixing of genes) followed by non-random selection, generationally improve the phenotype. The crossover process makes various combinations of parent genes (hence properties expressed by phenotype) to make new offspring. Mutation introduces random changes in the individuals, opening up new possibilities. The selection favors the fit and offers them the possibility of contributing to the new generation by crossover. Like in natural evolution, this process results in preservation of `good' features (inheritance) as tested by the objective (or fitness) function (analogous to environment where species live), contributing to generational improvement. Figure 1 shows a flowchart with essential steps in an EA. The shaded box (Evaluate Fitness) is the only step that is strictly problem-specific – the rest of the algorithm is common. The versatility of EA lies in the fact that the problem-specific part (evaluator) and the common part (EA) are coupled loosely by two simple communication paths: evaluator receives a number of `solutions' or genotypes (represented by `genes' representing decision variables) collectively known as `the population' and provides the EA with the `fitness' of each member of the population. This separation of concerns (the evaluator and the rest of the scheme are effectively separated from each other, with simple well-defined communication between them, i.e., the EA provides genotypes to the evaluator and evaluator returns their fitness) makes it very easy to adapt EA to a variety of problems. The most challenging part of implementing an EA solution is the proper implementation of this problem-specific evaluator component. The evaluator should map genotypes to respective phenotypes and evaluate them to return the fitness values. For example, a problem involving the sizing of three pipes in drainage network, with the constraint that the system should not allow flooding under a specified rainfall event. In this case the EA could return three real-numbers which in combination represent the genotype of the problem. Typically each of these numbers could range from zero to one. The evaluator should first map these three numbers to their `real-world' value. Suppose we know that the range of diameter of each pipe will be between 10 cm to 500 cm. Then each number (between 0 and 1) should be mapped as $v_i = 10 + g_i(500 - 10)$, where g_i and v_i are the gene and the respective expression in the phenotype, respectively. Now the evaluator should build the phenotype by building a network model using values $v_1, v_2,$ and v_3. Then it should calculate the cost of pipe laying using engineering knowledge coded to it. It should also run the hydraulic simulation of the network to check if there is flooding, in order to test the constraint of no-flooding. In most cases the evaluator is by far the computationally most expensive part of an EA.

COMPONENTS USED IN SWMM5-EA

In SWMM5-EA the evaluator consists of the urban drainage network model (EPA-SWMM 5.0) with some supporting code. EPA-SWMM 5.0 [13] (hereafter SWMM) is a general purpose urban hydrology and conveyance system hydraulics software and a dynamic rainfall-runoff simulation model used for single event or long-term simulation of runoff quantity and quality primarily from urban areas. The software is maintained by the Water Supply and Water Resources Division of the EPA's Risk Management Research program with assistance from the consulting firm of CDM, Inc. SWMM represents an urban drainage network with all its information like physical objects (e.g. watersheds, conduits), their properties (e.g. dimensions of conduits, hydrological properties of catchments) in a plain text file, known as SWMM input file. The SWMM model has a graphical user interface (hereafter known as SWMM Desktop) that allows building, editing and visualizing of a drainage network graphically (largely by

pointing and clicking). The component that does the simulation of the network is called the `SWMM engine'. The engine is entirely written in standard C language and the source code (for both Desktop and Engine) is made available in the public domain.

There are numerous studies on optimal design and planning by coupling the SWMM model with EA [3, 55, 56, 65]. The typical approach is to link the model with an EA library available for C language. However, in the present case it was decided to use the Python programming language for the software integration due to several reasons as outlined below.

Python [16] is a general purpose programming language well-known for its code-readability and supporting better programmer productivity [17] compared with conventional languages like C and Java. Another useful feature of the language is the availability of a wide variety of library routines via its standard library and numerous third party packages. An advantage of using python for the present project is the public availability of standardized, cross-platform Graphical User Interface (GUI) toolkits. For implementation of SWMM5-EA we selected PyQt – bindings for the Qt GUI toolkit [18]. However, one of the foremost reason to choose Python for this project is the efficiency and ease in programming. Compared with conventional languages (e.g. C), python provides an easy learning curve and the possibility to write clear, concise code. These and many other advantages make python an excellent programming language for engineering and scientific education[20]. One downside of python is that it is slower in execution compared to C or FORTRAN. However, in many programming projects in python, computation-intensive parts are often offloaded to libraries written in conventional languages. The language provides tools to conveniently achieve this.

Since the SWMM model is written in C language it was necessary to develop an interface to call various functionalities of the SWMM model (e.g. running a simulation, retrieving various simulation results) from python. Using SWIG interfacing library, such an interface is developed (The interface package is available under GPL license at https://pypi.python.org/pypi/SWMM5/).

From the number of EA libraries available for python, we used inspyred (``inspyred: Bio-inspired Algorithms in Python'' http://inspyred.github.com/), a widely used, general purpose EA library, in this project.

ACCESSIBILITY AND VERSATILITY

As stated in the introduction the major design challenge was to make SWMM5-EA both accessible (user-friendly) and versatile (possible to solve a variety of optimization problems related to urban drainage networks). Care was taken to embed these requirements into the design of the software.

The key innovation used in achieving versatility in the software design is the introduction of `place-holders' in the SWMM input file. A place-holder is a special annotation denoted by **'@! g !@**' where **g** is any function of the decision variables of the problem. In many problems this place-holder could be straight forward entry like `@!v1!@' or `@!v2!@'. However, it is possible to have complex place-holders. For example, take the simple example of optimal sizing of a dendritic drainage network (circular conduits A and B making a confluence to conduit C). It is the usual design practice that the size of downstream conduits are designed to be as large as or larger than any of the upstream conduits. In this case we can represent the diameters of conduits A, B and C as `@!v1!@' or `@!v2!@' and like `@!v3+max(v1,v2)!@' , respectively. The third statement makes sure that size of C is always at least as large as the bigger of A and B. A SWMM input file with such place-holders introduced is referred to as a 'template' in the rest of the discussion.

When the optimization is running, the evaluator receives genotypes, each specified by number of (three in the current example) genes (g_1, g_2 and g_3). These are mapped to variables v_1, v_2 and v_3 and used to fill-in the place-holders in the template. This makes the template a legitimate SWMM5 input file. Then the evaluator uses it to perform hydraulic analysis and other required computations like cost calculations.

SWMM5-EA has an easy to use graphical dialog that displays the SWMM5 input file and allows users to introduce decision variables by inserting such place holders by mouse clicks. Therefore the process of doing this remains simple for the new user. At the same time, the approach allows the advanced users to utilize a large degree of freedom to create a variety of advanced optimization problems.

It is important to state that the number of decision variables in the EA is equal to the number of variables, v_i, used in the place-holders in the template, not to the number of place-holders. For example, suppose the user introduces three place holders: @!v1!@, @!v1+v2!@ and @!v2!@. In this case the number of decision variables is two and the EA should supply the evaluator with genotypes consisting of two numbers in each.

The following section explains the process of using SWMM5-EA for a typical optimization problem.

PROCESS OF USE

Figure 2 shows the process of running a simulation with the SWMM5-EA tool. The first essential step is the building of a model to represent the drainage network. The user employs SWMM desktop to achieve this task. Next is the identification of the objective function(s) and decision variables and possible constraints. For example, in the case of least-cost design of a drainage network: Objective function is the cost of building the network. Decision variables are size of conduits (e.g. Diameter for a closed circular shape). The constraint could be that the system should perform properly (should not surcharge and flood) for a given design rainfall event. Then it is necessary to express the objective function in terms of the decision variables. In the above example, this is the cost of the pipe laying. Obviously the cost will be a function of pipe size, but not necessarily a linear one.

Another example is the calculation of optimal sizing of detention tanks in urban drainage systems. Here the decision variables are the sizes of each detention tank. For example it could be that we fix the height of each tank and represent area as a variable. The objective function is the cost of detention tanks. Again the constraint could be no-flooding for a certain design rainfall event.

At this stage the user invokes SWMM5-EA. First the user creates a project and imports the SWMM input file to the project. The next step is to replace in the SWMM network file, the constant value referring to each decision variable by a 'place holder'. For simple cases (e.g. `!@v2!@') , this is achieved by just blocking the relevant constant by dragging cursor over it and then pressing the button called `insert var' (Figure 3). However, when a complicated entry is needed (e.g. `@!v3+max(v1,v2)!@' in the example given in the above section), The user can manually edit the entries within this window.

Then the user adjusts the optimization parameters in the `project parameters' window. `Cost Function 1' is the `investment' cost (e.g. construction of pipes). `Cost Function 2' represents cost associated with flooding of the system. In the least-cost design example above, this function will be used as a constraint to prevent flooding from occurring. To achieve this the cost function is expressed as the total flood volume (f) multiplied by a large number (1.e12 in this case). This approach, known generally as a `penalty function', is a typical technique used to implement constraints in evolutionary methods [18]. This window also allows specifying (a) the type of optimization; (b) Important EA parameters like mutation and crossover rates; (c) Range of each variable (v1,v2,..vn); (d) How many evaluations to be conducted in the optimization run and (e) How many parallel processors to launch (Number of CPUs).

Once the above steps are completed the user starts the optimization process by pressing `Initiate optimization' and 'Run' buttons subsequently. Figure 5 shows the initial stage of an optimization run.

TYPICAL USAGES

The current version of SWMM5-EA allows the student to implement and run a variety of optimization problems. Some common examples include:

- Conduit sizing
- Optimal design of detention tanks.
- Optimal design of other `Low Impact Development' interventions [19] like Green Roofs, Rain Barrels, Swales, etc, for flood control.
- Combining several approaches for a least cost design (e.g. pipe sizing + detention)
- Calibration of SWMM models against observed data.

SWMM5-EA also provides multi-objective optimization. A typical example is obtaining a Pareto-optimal relationship between Investment cost (e.g. Building detention tanks) and volume of flood water (Figure 6).
Another typical use of SWMM5-EA is for `staged' design problems [5], which involve upgrading of a drainage system over a number of stages in time in response to environmental change (e.g. Change of imperviousness due to increased build up). Given that the anticipated change at each stage is known a priori, it is possible to calculate optimal upgrade paths.
It is also possible to implement optimization problems that involve combining a number of features listed above. For example, it is possible to combine detention tanks with pipe sizing or setup staged-optimization involving multiple-objectives.
Several usage examples are provided with the software documentation. The documentation is installed with the software and is also available online at http://swmm5-ea.readthedocs.org/.
One of the obvious limitations of the current form of SWMM5-EA is the inability to compute financial damage due to flooding, which is a useful component of many optimal design problems associated with urban drainage. However, enabling this feature involves implementing a more complex hydrodynamic model (e.g. 1D-2D coupled model). The computational burden involved makes such a product less appropriate for educational use than current implementation.

DISCUSSION

SWMM5-EA's design goal was first to be `user-friendly' and then to be flexible. If it is the flexibility that was the prime objective, then perhaps the approach of ad-hoc`gluing-together' of various libraries (as discussed in the Introduction) would have been appropriate. However, in order to create a tool for the novice user, who does not have experience in programming we opted for a modern graphical user interface. As it stands, SWMM5-EA provides a tool that is easy and straight forward to: (a) download and install; (b) execute variety of basic urban drainage network optimization tasks; and (c) examine the results of such optimizations.

Advanced Use

The primary goal of the software is as outlined above. However, since the introduction of the tool, there have been a several cases where users expressed the desire to adapt it for advanced optimization tasks. One example was the explicit estimation of flood damage using an inundation model that computes flood depths on the land surface. Obviously the tool, without major modifications, cannot achieve this. While it is possible to modify and adapt SWMM5-EA by end-users to achieve such objectives. However, this is not optimal as then the end-user is faced with the complexity of managing the total software system that includes the graphical interface, various pre-processing and post-processing modules etc. Instead, SWMM5-EA exploits a powerful feature of Python language to provide an interface to do such major modifications without the complexity of dealing with the unnecessary software components. The software allows the advanced user to introduce a `drop-in' replacement for the evaluator. This is practically achieved by introducing a Python script file named `swmm5ec_custom.py' in the directory `customcode' under the SWMM5-EA installation file. The software recognizes this file and replaces the internal evaluator with the evaluator provided by this customized code. This allows the advanced-user to introduce arbitrarily complex evaluators to the software by means of writing a bespoke python script.

In the context of this advanced mode, SWMM5-EA provides almost unlimited flexibility to implement any type of optimization problem for advanced research use. Another application of this `drop-in evaluator' is the possibility of introducing distributed computing abilities. In its simple form, SWMM5-EA already provides the ability to fully utilize modern multi-processor/multi-core computers. Using the advanced approach outlined above, it is possible to extend the software to utilize multiple computers in a grid-computing framework.

Other Domains

SWMM5-EA was primarily designed for applying optimization techniques for design of drainage systems. However, the framework and concepts introduced (e.g. `place-holder' system) can equally be used in other application domains. One straightforward adaptation is to apply the framework to apply optimization techniques in design of drinking water networks. Using the popular EPANET2.0 model, whose architecture is very similar to that of SWMM5, it is possible to achieve this conveniently.

CONCLUSIONS

A software tool intended to help with the learning of evolutionary algorithm-based optimization of urban drainage networks has been presented. Being a desktop software tool with a graphical user interface, it helps overcoming the existing barrier in including drainage network optimization exercises in engineering curricula, which are often limited to covering the theoretical aspects of this useful technique. The design of the software makes it possible to apply it to a wide variety of optimization problems in the sector, without being difficult to use.

Note: SWMM5-EA is copyrighted by the author and is released under the terms of GNU General Public License version 3. Available for download at: https://code.google.com/p/swmm5-ea/. Currently it is used in several graduate level courses including Asset management course (face-to-face) and Urban Drainage urban drainage and sewerage (online course) offered by UNESCO-IHE institute for water education.

REFERENCE

[1] Di Pierro, F., Djordjevic, S., Kapelan, Z., Khu, S. T., Savic, D., & Walters, G. A. (2005). Automatic calibration of urban drainage model using a novel multi-objective genetic algorithm. *Water Science & Technology*, *52*(5), 43-52.

[2] Rauch, Wolfgang, and Poul Harremoes. "Genetic algorithms in real time control applied to minimize transient pollution from urban wastewater systems." *Water Research* 33.5 (1999): 1265-1277.

[3] Delelegn, S. W., Pathirana, A., Gersonius, B., Adeogun, A. G., & Vairavamoorthy, K. (2011). Multi-objective optimisation of cost-benefit of urban flood management using a 1 D 2 D coupled model. *Water Science and Technology*, *63*(5), 1054.

[4] Savic, D., & Walters, G. (1995). Genetic operators and constraint handling for pipe network optimization. *Evolutionary Computing*, 154-165.

[5] Maharjan, M., Pathirana, A., Gersonius, B., & Vairavamoorthy, K. (2009). Staged cost optimization of urban storm drainage systems based on hydraulic performance in a changing environment. *Hydrology and Earth System Sciences*,*13*(4), 481.

[6] Gersonius, B., Ashley, R., Pathirana, A., & Zevenbergen, C. (2012). Climate change uncertainty: building flexibility into water and flood risk infrastructure.*Climatic Change*, 1-13.

[7] Kuo, J. T., Wang, Y. Y., & Lung, W. S. (2006). A hybrid neural–genetic algorithm for reservoir water quality management. *Water research*, *40*(7), 1367-1376.

[8] Radhakrishnan, M., Pathirana, A., Ghebremichael, K., & Amy, G. (2012). Modelling formation of disinfection by-products in water distribution: optimisation using a multi-objective evolutionary algorithm. *Journal of water supply: research and technology. AQUA*, *61*(3), 176-188.

[9] Khu, S. T., di Pierro, F., Savić, D., Djordjević, S., & Walters, G. A. (2006). Incorporating spatial and temporal information for urban drainage model calibration: An approach using preference ordering genetic algorithm. *Advances in water resources*, *29*(8), 1168-1181.
[10] Solomatine, D.P. Two strategies of adaptive cluster covering with descent and their comparison to other algorithms. Journal of Global Optimization, 1999, vol. 14, No. 1, pp. 55-78.
[11] Pathirana, A., Koster, J. H., de Jong, E., and Uhlenbrook, S.: On teaching styles of water educators and the impact of didactic training, Hydrol. Earth Syst. Sci., 16, 3677-3688, doi:10.5194/hess-16-3677-2012, 2012.
[12] Pathirana, A., Gersonius, B., and Radhakrishnan, M.: Web 2.0 collaboration tool to support student research in hydrology – an opinion, Hydrol. Earth Syst. Sci., 16, 2499-2509, doi:10.5194/hess-16-2499-2012, 2012.
[13] Rossman, L. A.:(2010). *Storm water management model user's manual, version 5.0*. National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency.
[14] Wan, B., & James, W. (2004, October). SWMM calibration using genetic algorithms. ASCE.Proceedings of 9th conference on Urban Drainage.
[15] Balascio, C. C., D. J. Palmeri, and H. Gao. "Use of a genetic algorithm and multi-objective programming for calibration of a hydrologic model." *Transactions of the ASAE* 41.3 (1998): 615-619.
[16] Van Rossum, G., & Drake, F. L. (2003). *Python language reference manual*. Network Theory Limited.
[17] Prechelt, Lutz. "An empirical comparison of C, C++, Java, Perl, Python, Rexx and Tcl." *IEEE Computer* 33.10 (2000): 23-29.
[18] Coello Coello, C. A. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art.*Computer methods in applied mechanics and engineering*, *191*(11), 1245-1287.
[19] Dietz, M. E. (2007). Low impact development practices: A review of current research and recommendations for future directions. *Water, Air, & Soil Pollution*, *186*(1), 351-363.
[20] Radenski, Atanas (2006). "Python First: A lab-based digital introduction to computer science." *ACM SIGCSE Bulletin*. Vol. 38. No. 3. ACM.